British
Geological Survey
NATURAL ENVIRONMENT RESEARCH COUNCIL

The Digital Energy Atlas and Library (DEAL) Database

Information Management Programme

Commercial Report CR/01/266

BRITISH GEOLOGICAL SURVEY

COMMERCIAL REPORT CR/01/266

The Digital Energy Atlas and Library (DEAL) Database

R Heaven

Contributor

R Pedley

The National Grid and other Ordnance Survey data are used with the permission of the Controller of Her Majesty's Stationery Office.
Ordnance Survey licence number GD 272191/1999

Key words

database, oil, gas, exploration and production, offshore, website, GIS

Front cover

Cover picture details, delete if no cover picture.

Bibliographical reference

HEAVEN, R. 2001. The Digital Energy Atlas and Library (DEAL) Database. *British Geological Survey Commercial Report*, CR/01/266. 96pp.

Keyworth, Nottingham British Geological Survey 2001

BRITISH GEOLOGICAL SURVEY

The full range of Survey publications is available from the BGS Sales Desks at Nottingham and Edinburgh; see contact details below or shop online at www.thebgs.co.uk

The London Information Office maintains a reference collection of BGS publications including maps for consultation.

The Survey publishes an annual catalogue of its maps and other publications; this catalogue is available from any of the BGS Sales Desks.

The British Geological Survey carries out the geological survey of Great Britain and Northern Ireland (the latter as an agency service for the government of Northern Ireland), and of the surrounding continental shelf, as well as its basic research projects. It also undertakes programmes of British technical aid in geology in developing countries as arranged by the Department for International Development and other agencies.

The British Geological Survey is a component body of the Natural Environment Research Council.

Keyworth, Nottingham NG12 5GG

☎ 0115-936 3241 Fax 0115-936 3488
e-mail: sales@bgs.ac.uk
www.bgs.ac.uk
Shop online at: www.thebgs.co.uk

Murchison House, West Mains Road, Edinburgh EH9 3LA

☎ 0131-667 1000 Fax 0131-668 2683
e-mail: scotsales@bgs.ac.uk

London Information Office at the Natural History Museum (Earth Galleries), Exhibition Road, South Kensington, London SW7 2DE

☎ 020-7589 4090 Fax 020-7584 8270
☎ 020-7942 5344/45 email: bgslondon@bgs.ac.uk

Forde House, Park Five Business Centre, Harrier Way, Sowton, Exeter, Devon EX2 7HU

☎ 01392-445271 Fax 01392-445371

Geological Survey of Northern Ireland, 20 College Gardens, Belfast BT9 6BS

☎ 028-9066 6595 Fax 028-9066 2835

Maclean Building, Crowmarsh Gifford, Wallingford, Oxfordshire OX10 8BB

☎ 01491-838800 Fax 01491-692345

Parent Body

Natural Environment Research Council, Polaris House, North Star Avenue, Swindon, Wiltshire SN2 1EU

☎ 01793-411500 Fax 01793-411501
www.nerc.ac.uk

Foreword

This report describes the relational database designed and implemented by BGS for the Digital Energy Atlas and Library (DEAL) project.

Acknowledgements

With thanks to Keith Adlam and Jason Careless for their knowledge of ESRI ArcSDE and ArcIMS.

Contents

FIGURES

Summary

This is a technical report describing the relational database designed and implemented for phases 1 and 2 of the DEAL project (Digital Energy Atlas and Library).

The first part of the report gives a brief description of the DEAL project. This is followed by the project scope and business rules that governed the database design: the data types to be stored, and how the entities should be related and attributes should be constrained. Several entity relationship diagrams of the logical model are included. The resulting Oracle implementation is described by a full table and column listing; database constraints and other database objects are listed in an appendix.

1 Introduction

This report describes the relational database designed and implemented for phases 1 and 2 of the DEAL project (Digital Energy Atlas and Library).

DEAL is a free, publicly available web based service developed and operated by BGS on behalf of Common Data Access Ltd (CDA) that promotes and facilitates access to data and information relevant to the exploration and production of hydrocarbons on the UK Continental Shelf (UKCS). The service consists of an ArcIMS interactive map and ColdFusion forms, which query a database of reference information and a catalogue of vendor products. Development started in June 2000; the website was publicly released in September 2000; upgrades for phase 2 were made live on 22 January 2001.

See Appendix 1 ("About Deal") or the DEAL website (http://www.ukdeal.co.uk) for a full description.

2 Database Design

The data was to be stored in an Oracle (version 8.1.6) relational database, and an ESRI ArcSDE service would be used to convey the spatial data to a geographical information system (the ESRI ArcIMS application).

2.1 DATABASE SCOPE

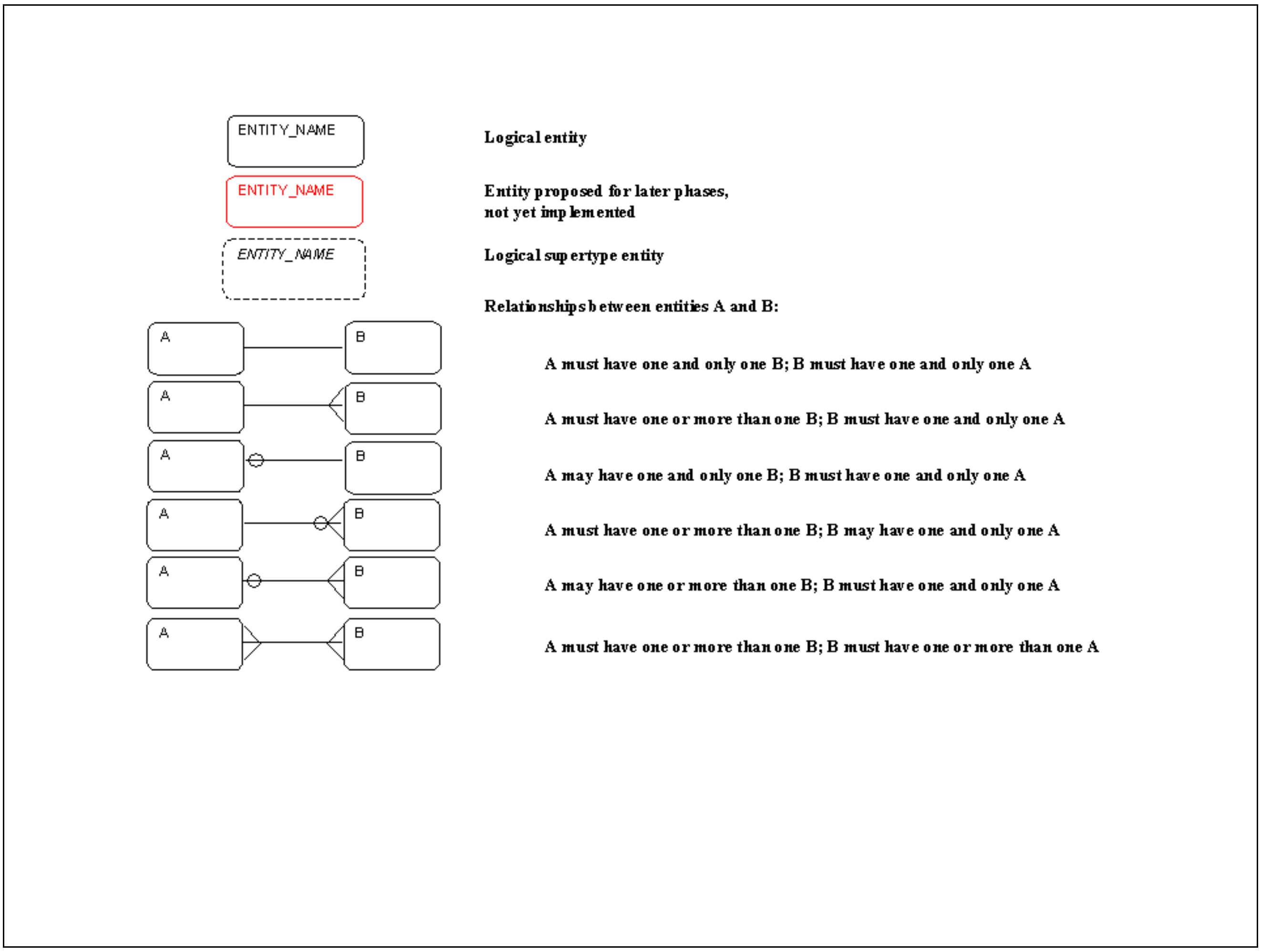
The database should store the following data groups:

- Spatial information and associated attributes of offshore features related to the oil and gas exploration and production industry. This includes seismic surveys, exploration wells, exploration and production infrastructure, and relevant political and administrative boundaries. The required entities and attributes for offshore features were detailed in the invitation to tender; a description of the datasets currently stored at the release of phase 2 can be found in Appendix 3.
- A multi-vendor catalogue of products associated with well and seismic features. Product catalogue categories were decided in consultation with product vendors.
- Details about companies and website users.

DEAL data modelling standards (Appendix 2) and a model for the management of spatial data (Appendix 4) were both followed.

2.2 LOGICAL DATA MODEL

Data modelling of the above datasets resulted in a logical data model, which can be described in terms of related entities and their attributes. This section contains overview entity relationship (ER) diagrams of the logical data model. The full set of diagrams of detailed parts of the logical model and the implemented oracle table relationships can be found in Appendix 6 at the end of this report.



Logical entity

Entity proposed for later phases,
not yet implemented

Logical supertype entity

Relationships between entities A and B:

A must have one and only one B; B must have one and only one A

A must have one or more than one B; B must have one and only one A

A may have one and only one B; B must have one and only one A

A must have one or more than one B; B may have one and only one A

A may have one or more than one B; B must have one and only one A

A must have one or more than one B; B must have one or more than one A

Figure 2-1 : Key to Logical Data Model Diagrams

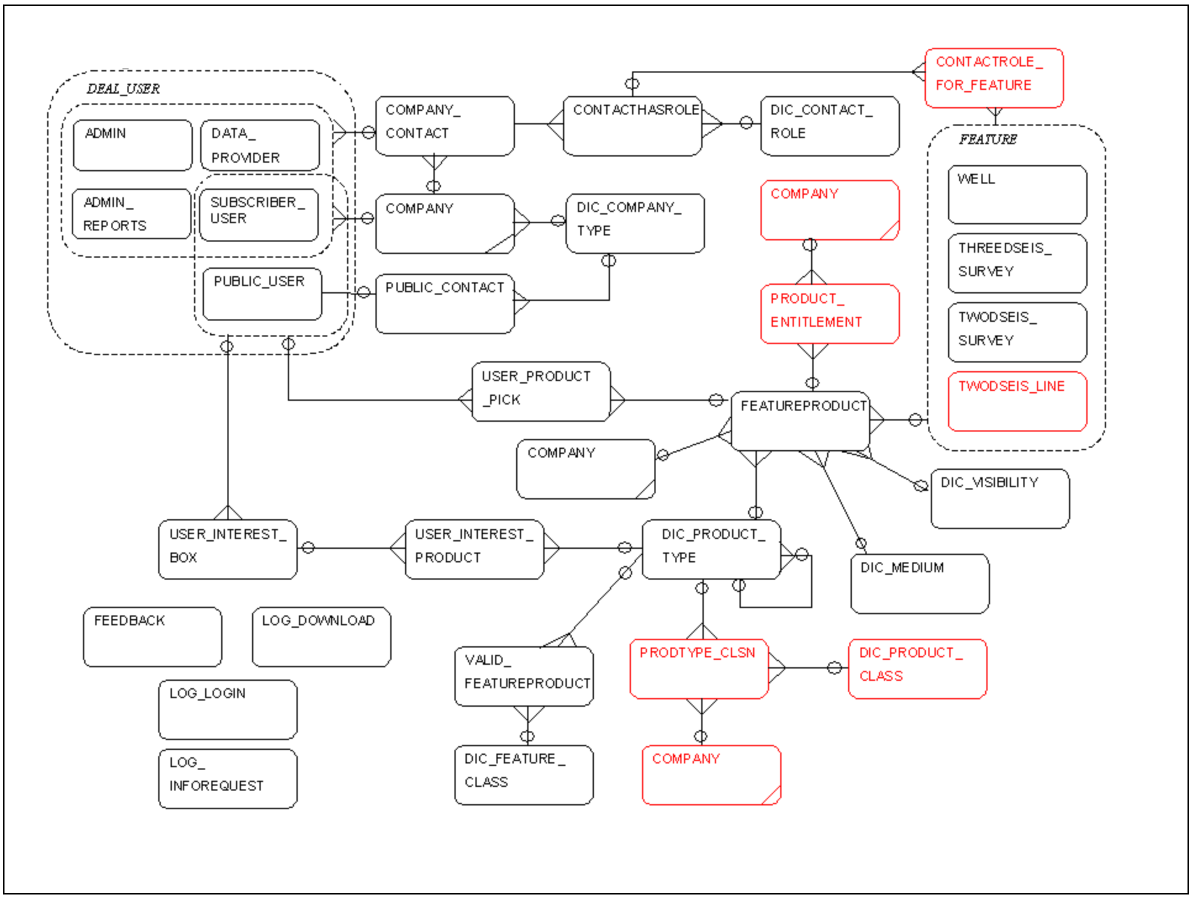


Figure 2-2 : Logical Model A : Users and Products

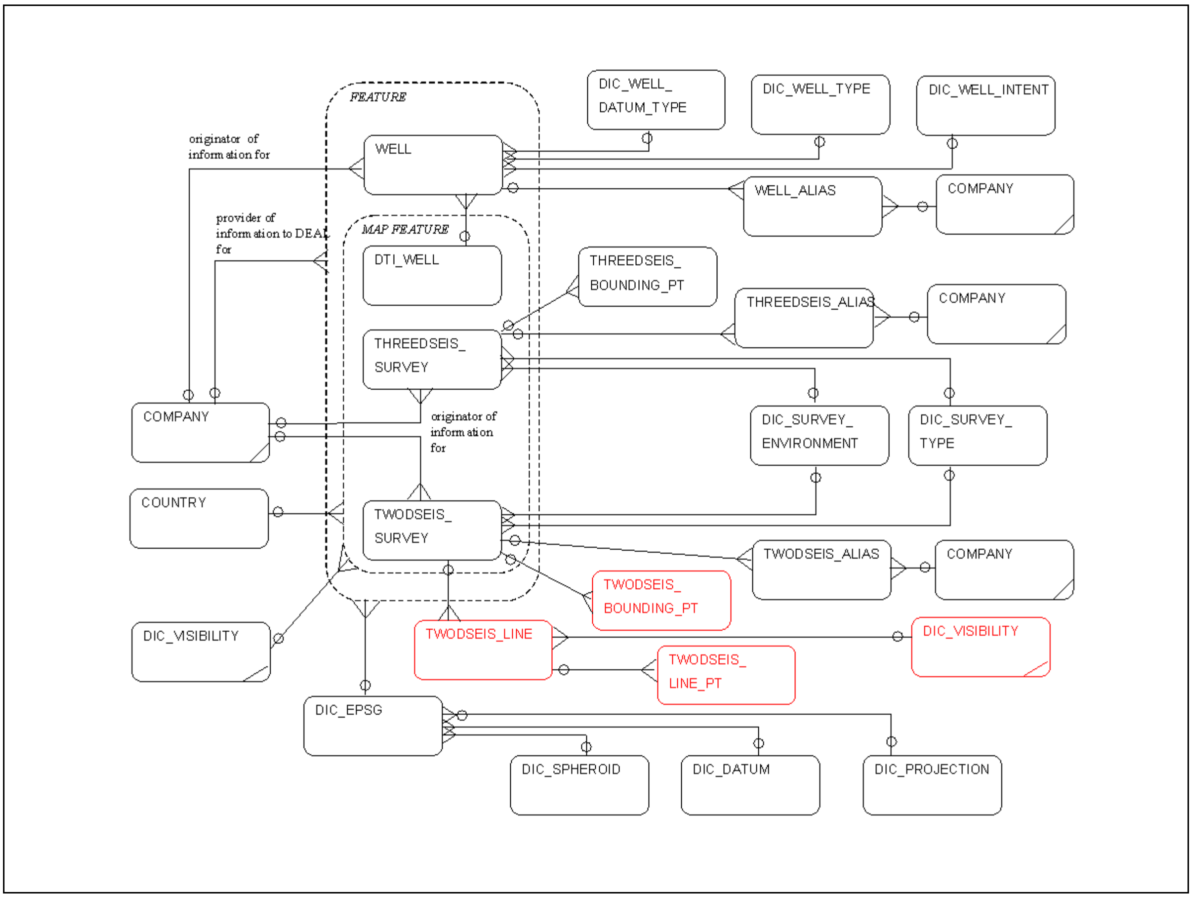


Figure 2-3 : Logical Model B : Wells and Seismic Surveys

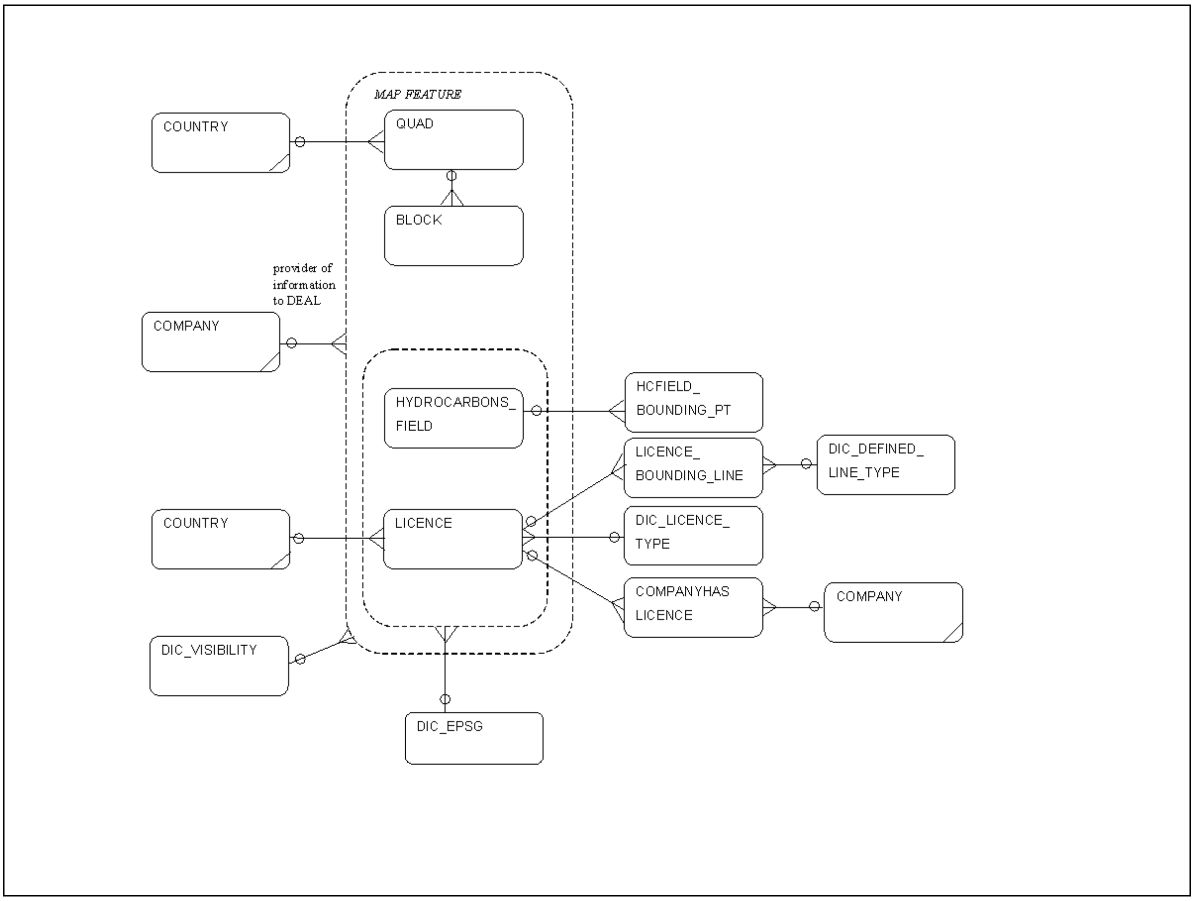


Figure 2-4 : Logical Model C : Quads, Licences, Fields

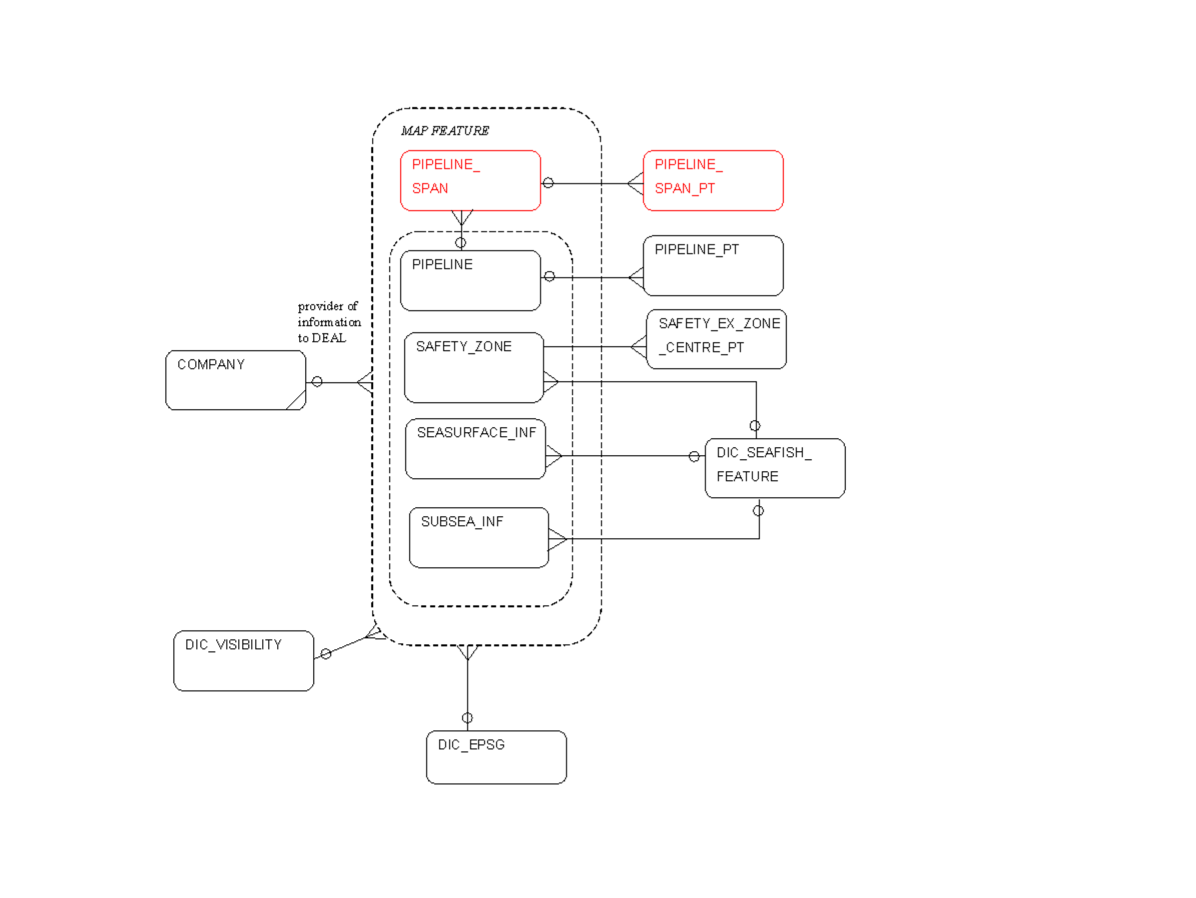


Figure 2-5 : Logical Model D : Infrastructure

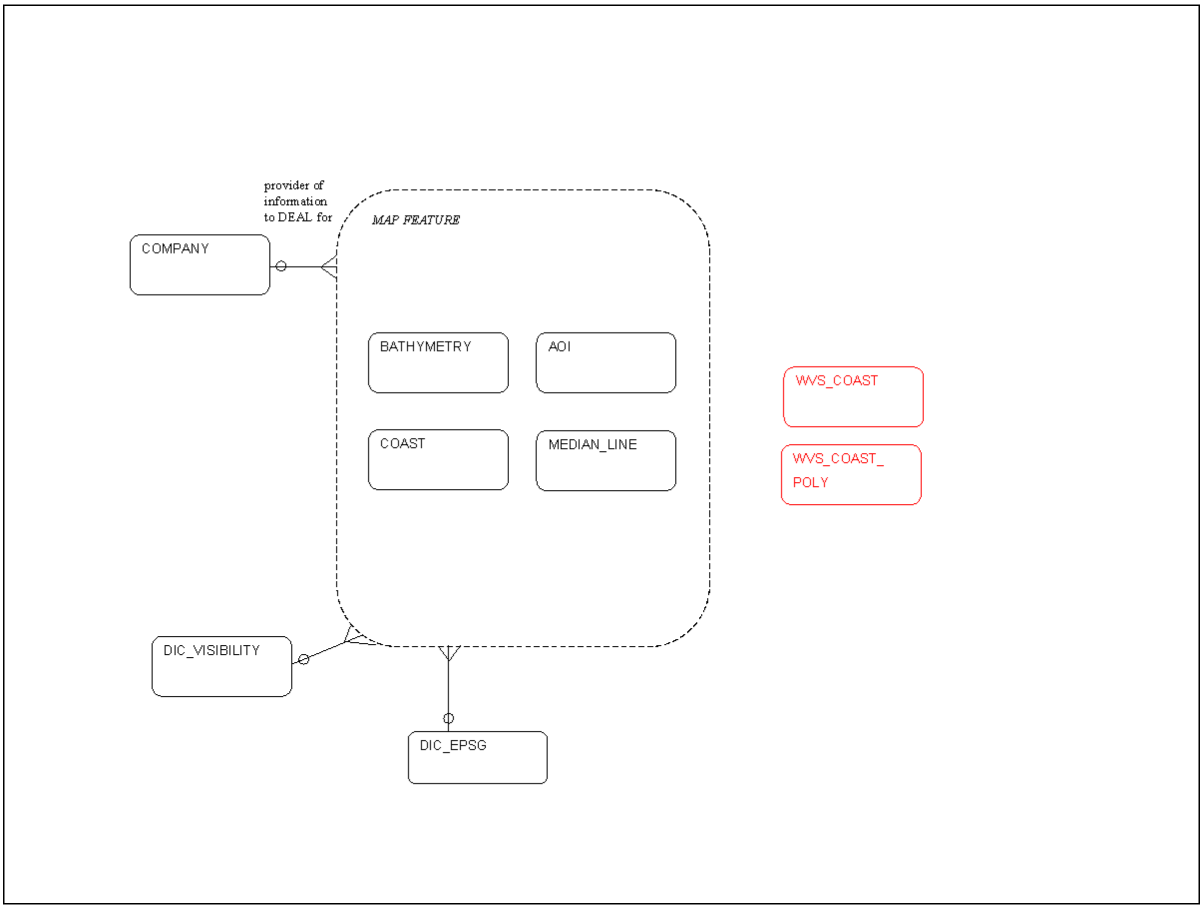


Figure 2-6 : Logical Model E : Other Cultural Features

2.3 BUSINESS RULES AND IMPLEMENTATION

The following section lists the principal business rules of the DEAL project that guided the design of the implemented database. Some of these rules were set out in the invitation to tender (ITT), but most were decided during the data modelling process, development of the website applications and on close inspection of the data to be loaded.

The method of implementation of each rule is shown. In most cases this is internal to the database, but a few rules are implemented only by the front-end applications.

N.B. Subheadings match the headings of the data model diagrams in Appendix 6.

2.3.1 General

Business Rule	Implementation
The source of each data item must be stored	All feature tables have an attribute DATA_SOURCE (or DATA_PROVIDER) which is a foreign key to COMPANY. This is the company that provided the data to DEAL, and the company that can be given permission to update details of the features. In some cases, the provider of the data to DEAL and the original source of the data have been identified separately. In these cases the single attribute of DATA_SOURCE has been replaced by two attributes of DATA_PROVIDER and DATA_ORIGIN. E.g. for 3D seismic surveys in THREEDSEIS_SURVEY, DATA_PROVIDER is CDA but DATA_ORIGIN is usually the owner of the survey. N.B. This is subject to revision to standardise column names.
Each table must have data insert and update user and dates	All tables have attributes of DATE_ENTERED, USER_ENTERED, DATE_UPDATED and USER_UPDATED. The user name should be user initials (e.g. REH) rather than the oracle user id. Triggers on each table insert date_entered and date_updated automatically
Each record should be uniquely identified in the relational database	Integer number identifiers have been used as the primary keys on many tables; the natural identifier of the entity has been enforced by a unique constraint where such a natural identifier exists in the data (not the case for infrastructure features). Oracle sequences have been defined for each table and triggers automatically use the next value in the sequence to populate the primary key field on insert.
Date fields should be constrained to avoid errors with incorrect date masks or Y2K issues	All fields containing dates that should be in the past e.g. WELL.SPUD_DATE, are constrained by triggers (*TABLE_COL_*…LT_SYSDATE) to be less than system date on insert

2.3.2 Users

Business Rule	Implementation
All deal users must have login name, type, date_created and date_lastused	Not null constraints

User must be of type ‘ADMIN’, ‘ADMIN_REPORTS’, ‘DATAPROVIDER’, ‘SUBSCRIBER’ or ’PUBLIC’	Check constraint on DEAL_USER.TYPE
User login name must be unique	Primary key on DEAL_USER.LOGIN
User must belong to one registered company unless user is of type ‘PUBLIC’	Check constraint and foreign key to COMPANY
User must be associated with one contact name within a registered company unless user is of type ‘PUBLIC’	Check constraint and foreign key to COMPANY_CONTACT
‘SUBSCRIBER’ and ‘PUBLIC’ type users must register whether they want to allow adverts	Check constraint
‘SUBSCRIBER’ and ‘PUBLIC’ type users must register whether they want to be notified of new products	Check constraint
‘SUBSCRIBER’ and ‘PUBLIC’ type users must have date of when they last used what’s new service	Check constraint
‘SUBSCRIBER’ and ‘PUBLIC’ type users must have date of when they were last notified of new products	Check constraint
‘PUBLIC’ type users must have one set of contact details; a set of public contact details can be associated with one user	Trigger on DEAL_USER to not allow a user of type ‘PUBLIC’ unless contact details exist in PUBLIC_CONTACT Trigger on PUBLIC_CONTACT to not allow delete of a row while the user is of type ‘PUBLIC’
Public contact must have a login name and password	Not null constraints
Public contact details may include a company type from a DEAL dictionary	Foreign key
A company may be of a type from a DEAL dictionary	Foreign key

A company contact must belong to one registered company	Foreign key
A company contact can perform one or more than one type of role; one type of role can be performed by one or more than one contact name.	Foreign keys from CONTACTHASROLE

2.3.3 Products

Business Rule	Implementation
A featureproduct is an instance of a type of product available in a type of medium from a particular data vendor for a particular exploration feature	Unique key Foreign keys
A featureproduct must have an attribute of visibility which describes its confidentiality status	Foreign key to DIC_VISIBILITY
A featureproduct must refer to one exploration feature, either a well or a seismic survey or seismic line.	Check constraint to ensure featureproduct has one and only one foreign key to an exploration feature (from WELL, THREEDSEIS_SURVEY, TWODSEIS_SURVEY, TWODSEIS_LINE)
Data vendor for the featureproduct must be registered as a company in DEAL	Foreign key to COMPANY
A product type may be a subtype of another product type; a product type can have one or more than one subtypes. Currently only two levels in this hierarchy will be supported.	Recursive foreign key from DIC_PRODUCT_TYPE.TYPE_CODE to DIC_PRODUCT_TYPE.SUPER_TYPE_CODE
A company may have entitlements to one or more than one featureproduct; a featureproduct may have entitlements to it from one or more than one company	Foreign keys from PRODUCT_ENTITLEMENT to COMPANY and FEATUREPRODUCT

2.3.4 User-Product Interactions

Business Rule	Implementation
Users of type ‘PUBLIC’ or ‘SUBSCRIBER’ can register an interest in one or more than one geographical areas. (only one geographical area is allowed currently; the default is the whole world)	Trigger to only allow foreign key from USER_INTEREST_BOX to DEAL_USER where users are of type 'PUBLIC' or 'SUBSCRIBER'

Users of type ‘PUBLIC’ or ‘SUBSCRIBER’ can register an interest in one or more than one product types for each of their chosen areas of interest. A product type can be of interest in one or more than one user interest area.	Foreign keys from USER_INTEREST_PRODUCT to USER_INTEREST_BOX and DIC_PRODUCT_TYPE
Users of type ‘PUBLIC’ or ‘SUBSCRIBER’ can make selections of featureproducts	Trigger to only allow foreign key from USER_PRODUCT_PICK to DEAL_USER where users are of type 'PUBLIC' or 'SUBSCRIBER'
A featureproduct can be selected by one or more than one user; a user can select one or more than one featureproduct	Foreign keys from USER_PRODUCT_PICK to FEATUREPRODUCT and DEAL_USER
A data provider has permission to update the attributes of featureproducts, but only those that do not form part of the unique key.	ColdFusion scripts
A data provider can delete and insert featureproducts interactively as long as he has at least one featureproduct in the DEAL catalogue. Featureproducts can be deleted by the data provider even if they are on another user’s pick list	ColdFusion scripts. A warning is flashed to the user prior to deletion of featureproducts that have been picked; if the deletion is confirmed rows are deleted from USER_PRODUCT_PICK and from FEATUREPRODUCT.

2.3.5 Product Classifications

Business Rule	Implementation
A class of feature (well, 3D seismic or 2D seismic) may have one or more than one valid product types; a product type may be associated with one or more than one feature class	Foreign keys from VALID_FEATUREPRODUCT to DIC_PRODUCT_TYPE and DIC_FEATURE_CLASS
A company may register its own classification scheme for product types (not implemented for phase 1 or 2 - possible future development)	Foreign keys from prodtype_clsn to dic_product_type, dic_product_class and company

2.3.6 Features with SDE geometry

Business Rule	Implementation

A record in a SDE geometry table must reference one record in a DEAL feature table; a deal feature table may have an associated SDE geometry.	Trigger on DEAL table (e.g. SPCOL_DEL_CASCADE_*n*) to enforce referential integrity between a record with a not-null SHAPE attribute and the records in SDE feature tables F*n* and S*n*, where SHAPE=F*n*.FID and SHAPE=S*n*.SP_FID Delete or update to NULL of the SHAPE attribute is cascaded to delete the associated records in F*n* and S*n* A value of SHAPE cannot be inserted or updated unless there already exists an associated record in F*n* This trigger is created by SDE during upload of data from SDE to Oracle attribute tables. It may need to be re-compiled to cope with table name changes since upload.
SDE table SDE.geometry_columns maps the name of the SDE geometry table Fn to the DEAL attribute table. Index number also held in SDE.LAYERS.layerid	e.g. F_TABLE_NAME = 'DTI_WELL' G_TABLE_NAME = 'F7'
All features associated with SDE geometry can also have spatial information and projection details stored separately from the SDE.	Features with a point shape (e.g. well) have LATITUDE, LONGITUDE, EASTING, NORTHING attributes in the feature table. Features with polygon or line shapes have these attributes in a related bounding_pt or line_pt table. NB DEAL.WELL also has attributes to store spatial data, but has no data in the SDE.

Any features with stored spatial attributes must also store co-ordinate system description of the stored attributes (this is not necessarily the same as the co-ordinate system of the map)	All DEAL feature tables have foreign keys to a dictionary table for EPSG codes, which uniquely defines a combination of spheroid, datum and projection. Foreign keys from DIC_EPSG point to dictionary tables DIC_SPHEROID, DIC_DATUM and DIC_PROJECTION. N.B. the co-ordinate system details are replicated as attributes in the feature table (originally so they could also be retrieved by the ArcIMS map reporting tool), but triggers (*TABLE_NAME*_EPSG_RULES_CSYS and *TABLE_NAME*_CSYS_FROM_EPSG) on update and insert ensure that they always match the dictionary values for the EPSG_CODE. In effect they are "dumb" attributes, an automatic expansion of the EPSG_CODE within the feature table. This is subject to revision now (phase 2 and onwards) that coordinate system attributes are not required for the map report – columns SPHEROID_NAME,DATUM_NAME,PROJECTION_NAME could be removed from the feature tables, leaving just EPSG_CODE for a normalized design.
Stored spatial attributes must be within the limits of the area of interest for DEAL.	Limits of the DEAL area of interest are stored in table AOI where AOI.aoi_name='DEAL'. Triggers *TABLE_NAME*_IN_AOI ensure stored attributes are within the DEAL AOI limits
All DEAL features have an attribute of visibility	Mandatory foreign key to dic_visibility on all DEAL feature tables
All DEAL features have one and only one company who provided the data to DEAL and who have the right to update that data	Mandatory foreign key to COMPANY table. See general rules for all DEAL entities

2.3.7 Wells

Business Rule	Implementation
A DTI registered well is identified uniquely by its well registration number	Unique key on dti_well_reg_no
The map will be labelled with an upper case version of the well registration, to make map searches easier	Unique key; check constraint to ensure it matches upper case dti_well_reg_no
A dti_well must be registered in one country	Foreign key to country

A dti_well must have an attribute of visibility	Foreign key to dic_visibility
A dti_well may have one geometry which is stored in SDE tables; the SDE geometry for the map feature of dti_well must be associated with one instance of DTI_WELL	See implementation for all SDE features
A dti_well may provide the surface location for one or more than one well bores	Mandatory foreign key from WELL to DTI_WELL
The dti_well it is derived from and its wellname uniquely identify a well bore. These wells might be re-spuds or side tracks, which have the same DTI well registration but may have different data products associated with them. N.B. Subject to revision post phase 2	Unique key on WELL(dti_wellid, wellname)
A well has its own version of its spatial information which may be different from the spatial information registered with the dti_well. N.B. Subject to revision post phase 2	WELL has attributes of latitude, longitude, easting, northing and foreign keys to projection dictionaries
A well may have one or more than one alias names. A company may have one or more than one alias name by which they can identify a well. N.B. Subject to possible revision to only allow one alias name per company for each well.	WELL_ALIAS table, composite primary key of WELLID, ALIAS_COMPANY, WELL_ALIAS_NAME.

2.3.8 2D and 3D Seismic Surveys

Business Rule	Implementation
A seismic survey is uniquely identified by its CS9 name; a naming system developed for CDA.	Unique key on THREEDSEIS_SURVEY.THREEDSEIS_CS9NAM E Unique key on TWODSEIS_SURVEY.TWODSEIS_CS9NAME

A seismic survey may have one or more than one alias names. A company may have one or more than one alias name by which they can identify a survey. N.B. Subject to possible revision to only allow one alias name per company for each survey.	Foreign keys from threedseis_alias to threedseis_survey and company Unique primary key on THREEDSEIS_ALIAS (THREEDSEIS_SURVEYID, ALIAS_COMPANY, THREEDSEIS_ALIAS_NAME) Similarly, foreign keys from twodseis_alias to twodseis_survey and company Unique primary key on TWODSEIS_ALIAS (TWODSEIS_SURVEYID, ALIAS_COMPANY, TWODSEIS_ALIAS_NAME)
2D seismic surveys will be shown on 2 map layers, depending on the lineage of the data. For phase 2 there will be one layer for surveys originally stored in the DTI database and now archived by CDA. A second layer will be for surveys in the CDA repository that were supplied to CDA from their clients. N.B. Revision post phase 2 will allow further map layers for survey data that have been provided directly to DEAL independently of CDA; probably one new map layer for each provider company.	TWODSEIS_SURVEY has column DATA_PROVIDER_SUPPLIER which is constrained to be in ('DTI','CDA Repository'). Map layers are distingushed using this column. N.B. Revision post phase 2 will have to expand the list for this check constraint to accommodate the additional map layer names, or change to make the column a foreign key to COMPANY table or other dictionary.

2.3.9 Infrastructure

Business Rule	**Implementation**
A safety zone can be defined by a distance around a point, line or polygon feature (only data so far is for zones defined around points, which are mostly sub-surface infrastructure features)	Attribute SAFETY_EX_ZONE.CENTRE_GEOMETRY_TYPE must be point, line or polygon. Table SAFETY_EX_ZONE_CENTRE_PT contains the location of a point around which the safety zone is defined; this may be a single point or one in a sequence along a line or polygon boundary. Foreign key from SAFETY_EX_ZONE_CENTRE_PT to SAFETY_EX_ZONE
Sub-seasurface infrastructures are point features and include suspended wells. Seasurface infrastructures are point features. There is some overlap between these two sets due to structures that extend from the sea surface to the sea floor, e.g. platforms and buoys	Tables SUBSEA_INF and SEASURFACE_INF

A pipeline may have one or more than one sections that span sea floor valleys	Foreign key from pipeline_span to PIPELINE

2.3.10 Quad and Blocks

Business Rule	Implementation
A quad is uniquely identified by its quad number and country	Unique key on QUADNO,COUNTRY Foreign key to COUNTRY.GMICODE
The quad may or may not be labelled on the map	Optional attribute QUADLABEL; check constraint ensures this is NULL or is equal to QUADNO
A block is uniquely defined by the block number and the quad it is contained in	Unique key on BLOCKNO, QUADNO, COUNTRY. Mandatory foreign key to QUAD
A block may or may not be labelled on the map with its full identifier	Optional attribute BLOCKLABEL; check constraints ensures this is NULL or is equal to “QUADNO/BLOCKNO”

3 Database Description

The tables are described in full in the following section. Appendix 5 “Database Constraints and Other Database Objects” contains the SQL commands used to create the primary key, unique key, foreign key, and check constraints; and for the database sequences, indexes, functions and triggers. Table relationships can be seen in diagrams of the implemented model in Appendix 6 “Data Model Diagrams”.

3.1 DESCRIPTION OF ORACLE TABLES

Tables are listed in alphabetical order.

AOI

spatial feature of an area of interest

COLUMN	FORMAT	NULL?	Description
AOI_NAME	VARCHAR2(4)	NOT NULL	PK, name to identify the area of interest. This is ‘DEAL’ for the AOI of the whole project.
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
MIN_LAT	NUMBER(10,7)		minimum latitude, decimal degrees
MIN_LON	NUMBER(11,7)		minimum longitude, decimal degrees, negative west
MAX_LAT	NUMBER(10,7)		maximum latitude, decimal degrees
MAX_LON	NUMBER(11,7)		maximum longitude, decimal degrees, negative west

MIN_EASTING	NUMBER(10,2)		minimum easting, metres. Only allowed not null value if PROJECTION_NAME is not null
MIN_NORTHING	NUMBER(10,2)		minimum northing, metres. Only allowed not null value if PROJECTION_NAME is not null
MAX_EASTING	NUMBER(10,2)		maximum easting, metres. Only allowed not null value if PROJECTION_NAME is not null
MAX_NORTHING	NUMBER(10,2)		maximum northing, metres. Only allowed not null value if PROJECTION_NAME is not null
EPSG_CODE	NUMBER(5)	NOT NULL	EPSG code for co-ordinate system of stored positional attributes, FK to DIC_EPSG
DATUM_NAME	VARCHAR2(10)	NOT NULL	Datum for co-ordinate system of stored positional attributes, FK to DIC_DATUM
SPHEROID_NAME	VARCHAR2(10)	NOT NULL	spheroid name for co-ordinate system of stored positional attributes, FK to DIC_SPHEROID
PROJECTION_NAME	VARCHAR2(10)		projection name for co-ordinate system if stored in easting northings, FK to DIC_PROJECTION
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		
SHAPE	NUMBER(38)		Must reference a feature id number stored in geometry table. Trigger to prevent update to or insert of a non-existent geometry id number

BATHYMETRY

No data received yet - required post phase 2. Table design may have to be altered

COLUMN	FORMAT	NULL?	Description
BATHYMETRYID	NUMBER(38)	NOT NULL	PK, database unique identifier of the safety zone feature. Integer number generated automatically by a sequence, has no meaning outside of the DEAL database
BATHYMETRY_LABEL	VARCHAR2(80)		Map label and unique identifier of the feature
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
EPSG_CODE	NUMBER(5)	NOT NULL	EPSG code for co-ordinate system of stored positional attributes, FK to DIC_EPSG
DATUM_NAME	VARCHAR2(10)	NOT NULL	Datum for co-ordinate system of stored positional attributes, FK to DIC_DATUM
SPHEROID_NAME	VARCHAR2(10)	NOT NULL	spheroid name for co-ordinate system of stored positional attributes, FK to DIC_SPHEROID
PROJECTION_NAME	VARCHAR2(10)		
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

BLOCK

spatial feature of administration block area. Uniquely identified by (COUNTRY, QUADNO, BLOCKNO). (COUNTRY,QUADNO) must already exist in table QUAD

COLUMN	FORMAT	NULL?	Description
BLOCKID	NUMBER(37)	NOT NULL	PK, database unique identifier of the block. Integer number generated automatically by a sequence, has no meaning outside of the DEAL database
COUNTRY	CHAR(3)	NOT NULL	gmi code of country for which quad and block is defined. Part of FK to QUAD. Part of unique key
QUADNO	VARCHAR2(5)	NOT NULL	the real world identifier of the quad for which the block is defined, integer number but stored as char field to enable map finds. Part of FK

		NULL	to QUAD. Part of unique key.
BLOCKNO	VARCHAR2(5)	NOT NULL	the real world identifier of the block, usually an integer number. Part of unique key.
BLOCKLABEL	VARCHAR2(11)		label for the block which will appear on the map. Constraint checks this is "quadno/blockno" or null
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		
VERIFIED	CHAR(1)		Indicates if feature has been verified by the data source. Constraint checks this is 'Y' or 'N'. Null indicates the field has not been verified yet, 'N' indicates it has been marked as suspect
SHAPE	NUMBER(38)		Must reference a feature id number stored in geometry table. Trigger to prevent update to or insert of a non-existent geometry id number

COAST

spatial feature of a section of coast line. No natural identifier for each section of coast line

COLUMN	FORMAT	NULL?	Description
COASTID	NUMBER(38)	NOT NULL	PK, database unique identifier of the coast line feature. Integer number generated automatically by a sequence, has no meaning outside of the DEAL database
FEATURE_LABEL	VARCHAR2(40)		the map label for the line section, usually the country name that the section of coast belongs to
FEATURE_TYPE	VARCHAR2(40)		the type of feature that the line section is defining, eg mainland or island belonging to a country
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
EPSG_CODE	NUMBER(5)	NOT NULL	EPSG code for co-ordinate system of stored positional attributes, FK to DIC_EPSG
DATUM_NAME	VARCHAR2(10)	NOT NULL	Datum for co-ordinate system of stored positional attributes, FK to DIC_DATUM
SPHEROID_NAME	VARCHAR2(10)	NOT NULL	spheroid name for co-ordinate system of stored positional attributes, FK to DIC_SPHEROID
PROJECTION_NAME	VARCHAR2(10)		projection name for co-ordinate system if stored in easting northings, FK to DIC_PROJECTION
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		
VERIFIED	CHAR(1)		Indicates if feature has been verified by the data source. Constraint checks this is 'Y' or 'N'. Null indicates the field has not been verified yet, 'N' indicates it has been marked as suspect
SHAPE	NUMBER(38)		Must reference a feature id number stored in geometry table. Trigger to prevent update to or insert of a non-existent geometry id number

COMPANY

COLUMN	FORMAT	NULL?	Description
COMPANYID	VARCHAR2(40)	NOT NULL	PK, identifier of the company. A shortened or coded version of the full company name

COMPANYNAME	VARCHAR2(80)	NOT NULL	full company name
COMPANY_TYPE	VARCHAR2(20)		company type, FK to DIC_COMPANY_TYPE
WEBSITEURL	VARCHAR2(80)		home page of company website
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

COMPANY_CONTACT

Individual or department in a company that can serve as a contact

COLUMN	FORMAT	NULL?	Description
COMPANYID	VARCHAR2(40)	NOT NULL	Part PK, company to which this contact belongs, FK to COMPANY
CONTACTREF	VARCHAR2(10)	NOT NULL	Part PK, identifier of the contact within the company eg userid, department name
NAME	VARCHAR2(80)		full name of contact
EMAIL	VARCHAR2(80)		email address for the contact
STREETADDRESS	VARCHAR2(80)		building number and street name of postal address
TOWN	VARCHAR2(80)		postal address town
POSTCODE	VARCHAR2(20)		postcode
COUNTRY	VARCHAR2(40)		freely entered country name
TELEPHONE	VARCHAR2(20)		contact telephone number
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

COMPANYHASLICENCE

intersection table that resolves many to many relationship between company and licence

COLUMN	FORMAT	NULL?	Description
OWNER_COMPID	VARCHAR2(40)	NOT NULL	part PK, identifier of the company which owns part or all of the licence, FK to COMPANY
LICENCEID	NUMBER(38)	NOT NULL	part PK, identifier of the licence, FK to LICENCE
PERCENTAGE	NUMBER(6,3)		percentage of the total licence that this partner owns. Constraint checks this is between 0 and 100
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

CONTACTHASROLE

the DEAL business function or role that a company_contact serves as a contact for

COLUMN	FORMAT	NULL?	Description
COMPANYID	VARCHAR2(40)	NOT NULL	Part PK, part FK to COMPANY_CONTACT
CONTACTREF	VARCHAR2(10)	NOT NULL	Part PK, part FK to COMPANY_CONTACT
CONTACTROLE	VARCHAR2(20)	NOT NULL	Part PK, function or role that this contact serves, FK to DIC_CONTACT_ROLE

DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

COUNTRY

COLUMN	**FORMAT**	**NULL?**	**Description**
GMICODE	CHAR(3)	NOT NULL	PK, GMI country code
FIPSCODE	CHAR(2)		FIPS country code
NAME	VARCHAR2(40)	NOT NULL	full country name
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DEAL_USER

A registered user of the DEAL service

COLUMN	**FORMAT**	**NULL?**	**Description**
LOGIN	VARCHAR2(20)	NOT NULL	PK, login and user id
TYPE	VARCHAR2(13)	NOT NULL	type of user in 'PUBLIC','DATAPROVIDER','SUBSCRIBER','ADMIN','ADMIN_REPORTS'
DATE_CREATED	DATE	NOT NULL	date user id was created
DATE_LASTUSED	DATE	NOT NULL	date user last logged in to DEAL website
COMPANYID	VARCHAR2(40)		company which this user belongs to, FK to COMPANY. Must be not null if TYPE!='PUBLIC'
CONTACTREF	VARCHAR2(10)		contact for this user id, FK to COMPANY_CONTACT. Must be not null if TYPE !='PUBLIC'
NOTIFYNEWPRODUCTS	CHAR(1)		'Y' or 'N'. Must be not null if TYPE in 'PUBLIC','SUBSCRIBER'
ALLOWADVERTS	CHAR(1)		'Y' or 'N'. Must be not null if TYPE in 'PUBLIC','SUBSCRIBER'
DATE_LASTWHATSNEW	DATE		Date the user last used the whatsnew script. Must be not null if TYPE in 'PUBLIC','SUBSCRIBER'
DATE_LASTNOTIFYCHECK	DATE		Must be not null if TYPE in 'PUBLIC','SUBSCRIBER'
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_COMPANY_TYPE

dictionary of code for company types for the purpose of admin reports and usage statistics, e.g. exploration, academic

COLUMN	**FORMAT**	**NULL?**	**Description**
TYPE_CODE	VARCHAR2(20)	NOT NULL	PK
TRANSLATION	VARCHAR2(40)	NOT	

		NULL	
DESCRIPTION	VARCHAR2(240)		
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_CONTACT_ROLE

dictionary of roles or responsibilities which a named contact can have

COLUMN	FORMAT	NULL?	Description
ROLE_CODE	VARCHAR2(20)	NOT NULL	PK
TRANSLATION	VARCHAR2(40)	NOT NULL	
DESCRIPTION	VARCHAR2(240)		
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_DATUM

Dictionary of codes for co-ordinate system datum

COLUMN	FORMAT	NULL?	Description
DATUM_CODE	VARCHAR2(10)	NOT NULL	PK
TRANSLATION	VARCHAR2(40)	NOT NULL	
DESCRIPTION	VARCHAR2(240)		
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_DEFINED_LINE_TYPE

dictionary of types of line by which a spatial feature can be defined e.g. constant bearing, great circle

COLUMN	FORMAT	NULL?	Description
TYPE_CODE	VARCHAR2(10)	NOT NULL	PK
TRANSLATION	VARCHAR2(40)	NOT NULL	
DESCRIPTION	VARCHAR2(240)		
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	

USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_DOWNLOAD_TYPE

dictionary of data types that can be downloaded - used in the logging of downloads in LOG_DOWNLOAD

COLUMN	FORMAT	NULL?	Description
TYPE	VARCHAR2(40)	NOT NULL	type of dataset that can be downloaded
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_EPSG

Dictionary of codes defined by the European Petroleum Survey Group which precisely specify all details of a co-ordinate system. Populated with just those of use to DEAL

COLUMN	FORMAT	NULL?	Description
EPSG_CODE	NUMBER(5)	NOT NULL	PK
TRANSLATION	VARCHAR2(40)	NOT NULL	
DESCRIPTION	VARCHAR2(240)		
DATUM_NAME	VARCHAR2(10)	NOT NULL	Code for the datum, FK to DIC_DATUM
SPHEROID_NAME	VARCHAR2(10)	NOT NULL	Code for the spheroid, FK to DIC_SPHEROID
PROJECTION_NAME	VARCHAR2(10)		Code for the projection, FK to DIC_PROJECTION
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_FEATURE_CLASS

Dictionary of codes for classes of features that may have product types in the DEAL catalogue

COLUMN	FORMAT	NULL?	Description
FEATURE_CLASS_CODE	CHAR(2)	NOT NULL	PK, code for the class e.g. 'WL' for WELL,'3D','2D' for three d and two d seismic
TRANSLATION	VARCHAR2(40)	NOT NULL	
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_LICENCE_TYPE

Dictionary of codes for types of licence e.g. hydrocarbons exploration licence

COLUMN	FORMAT	NULL?	Description
TYPE_CODE	VARCHAR2(10)	NOT NULL	PK

TRANSLATION	VARCHAR2(40)	NOT NULL	
DESCRIPTION	VARCHAR2(240)		
DATA_SOURCE	VARCHAR2(40)		company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_MEDIUM

Dictionary of codes for a medium on which a product type can be available

COLUMN	FORMAT	NULL?	Description
MEDIUM_CODE	VARCHAR2(20)	NOT NULL	PK
TRANSLATION	VARCHAR2(40)	NOT NULL	
DESCRIPTION	VARCHAR2(240)		
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_PRODUCT_CLASS

Dictionary of codes for classes that products may be grouped in using a different system from the DEAL catalogue. Not in use - post phase 2 possibility.

COLUMN	FORMAT	NULL?	Description
CLASS_CODE	VARCHAR2(20)	NOT NULL	PK
SUPER_CLASS_CODE	VARCHAR2(20)		
TRANSLATION	VARCHAR2(40)	NOT NULL	
DESCRIPTION	VARCHAR2(240)		
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company originally naming this class. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_PRODUCT_TYPE

Dictionary of codes for product types that products must be grouped in within the DEAL catalogue; hierarchical structure of supertypes and subtypes, only populated for 2 levels in the hierarchy for phase 1

COLUMN	FORMAT	NULL?	Description
TYPE_CODE	VARCHAR2(20)	NOT NULL	PK
SUPER_TYPE_CODE	VARCHAR2(20)		product type containing this type, self referencing FK to DIC_PRODUCT_TYPE.TYPE_CODE
TRANSLATION	VARCHAR2(40)	NOT NULL	
DESCRIPTION	VARCHAR2(240)		
DATA_SOURCE	VARCHAR2(40)	NOT	company that supplied this information to DEAL. FK to COMPANY

		NULL	
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_PROJECTION

Dictionary of codes for co-ordinate system projections

COLUMN	FORMAT	NULL?	Description
PROJECTION_CODE	VARCHAR2(10)	NOT NULL	PK
TRANSLATION	VARCHAR2(40)	NOT NULL	
DESCRIPTION	VARCHAR2(240)		
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_SEAFISH_FEATURE

Dictionary of codes used by SeaFish to classify surface and subsurface infrastructure features

COLUMN	FORMAT	NULL?	Description
FEATURE_CODE	CHAR(4)	NOT NULL	PK, code for the feature type
FEATURE_GROUP	VARCHAR2(20)		grouping that this feature code is in, constraint checks this is 'PIPELINE FEATURE','CABLE FEATURE','PIPELINE STATUS','SURFACE FEATURE','SUBSURFACE FEATURE','SUB AND SURFACE'
TRANSLATION	VARCHAR2(40)	NOT NULL	
DESCRIPTION	VARCHAR2(240)		
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_SPHEROID

Dictionary of codes for co-ordinate system spheroids

COLUMN	FORMAT	NULL?	Description
SPHEROID_CODE	VARCHAR2(10)	NOT NULL	PK
TRANSLATION	VARCHAR2(40)	NOT NULL	
DESCRIPTION	VARCHAR2(240)		
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	

USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_SURVEY_ENVIRONMENT

Dictionary of codes for survey environments e.g. land, air, marine. Common dictionary for 2d and 3d seismic surveys

COLUMN	FORMAT	NULL?	Description
ENVIRONMENT_CODE	VARCHAR2(10)	NOT NULL	PK
TRANSLATION	VARCHAR2(40)	NOT NULL	
DESCRIPTION	VARCHAR2(240)		
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_SURVEY_TYPE

Dictionary of codes for types of survey, e.g. high resolution, reconnaissance. Common dictionary for 2d and 3d seismic surveys

COLUMN	FORMAT	NULL?	Description
TYPE_CODE	VARCHAR2(10)	NOT NULL	PK
TRANSLATION	VARCHAR2(40)	NOT NULL	
DESCRIPTION	VARCHAR2(240)		
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_VISIBILITY

Dictionary of codes to indicate the restrictions that may apply to the visibility of a feature on a map

COLUMN	FORMAT	NULL?	Description
VISIBILITY_CODE	CHAR(1)	NOT NULL	PK
TRANSLATION	VARCHAR2(40)	NOT NULL	
DESCRIPTION	VARCHAR2(240)		
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_WELL_DATUM_TYPE

Dictionary of codes for well vertical datum e.g. Kelly Bushing, Ground Level

COLUMN	FORMAT	NULL?	Description
TYPE_CODE	VARCHAR2(3)	NOT NULL	PK
TRANSLATION	VARCHAR2(40)	NOT NULL	
DESCRIPTION	VARCHAR2(240)		
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_WELL_INTENT

Dictionary of codes for well intent e.g. appraisal, development, exploration

COLUMN	FORMAT	NULL?	Description
INTENT_CODE	CHAR(1)	NOT NULL	PK
TRANSLATION	VARCHAR2(40)	NOT NULL	
DESCRIPTION	VARCHAR2(240)		
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DIC_WELL_TYPE

Dictionary of codes for well types e.g. injector or producer

COLUMN	FORMAT	NULL?	Description
TYPE_CODE	VARCHAR2(10)	NOT NULL	PK
TRANSLATION	VARCHAR2(40)	NOT NULL	
DESCRIPTION	VARCHAR2(240)		
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

DTI_WELL

Spatial feature of a sea floor well position that has been registered with DTI. Labelled and identified by the DTI well registration number assigned to it. Uniquely identified by (DTI_WELL_REG_NO) or (DTI_WELL_REG_UPPER)

COLUMN	FORMAT	NULL?	Description
DTI_WELLID	NUMBER(38)	NOT NULL	PK, database unique identifier of the dti_well. Integer number generated automatically by a sequence, has no meaning outside of the

			DEAL database
DTI_WELL_REG_NO	VARCHAR2(40)	NOT NULL	mixed case DTI registration number of the well, used as a map label. Unique key.
DTI_WELL_REG_UPPER	VARCHAR2(40)	NOT NULL	upper case DTI registration number of the well, used for searches. Constraint checks DTI_WELL_REG_UPPER=upper(DTI_WELL_REG_NO). Second unique key.
DTI_DEN_NO	NUMBER(7)		DTI Department of Energy number identifier of the well
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
COUNTRY	CHAR(3)	NOT NULL	gmi code of the country in which the well is registered. FK to COUNTRY
FIELDNAME	VARCHAR2(40)		name of the hydrocarbons field containing this well. Constraint checks this is upper case. Does not reference HYDROCARBONS_FIELD
VERTICAL_UNITS	CHAR(1)		measurement units that depth and elevation fields are stored in. Must be 'F' for feet or 'M' for metres
WATER_DEPTH	NUMBER(7,2)		depth of the water column at the well location in VERTICAL_UNITS.
EPSG_CODE	NUMBER(5)	NOT NULL	EPSG code for co-ordinate system of stored positional attributes, FK to DIC_EPSG
DATUM_NAME	VARCHAR2(10)	NOT NULL	Datum for co-ordinate system of stored positional attributes, FK to DIC_DATUM
SPHEROID_NAME	VARCHAR2(10)	NOT NULL	spheroid name for co-ordinate system of stored positional attributes, FK to DIC_SPHEROID
LATITUDE	NUMBER(10,7)		latitude of the well location, decimal degrees
LONGITUDE	NUMBER(11,7)		longitude of the well location, decimal degrees, negative west
PROJECTION_NAME	VARCHAR2(10)		projection name for co-ordinate system if stored in eastings northings, FK to DIC_PROJECTION. Must be not null if EASTING or NORTHING are not null.
EASTING	NUMBER(10,2)		easting of the well location, metres. PROJECTION_NAME must be not null if this is not null
NORTHING	NUMBER(10,2)		northing of the well location, metres. PROJECTION_NAME must be not null if this is not null
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		
VERIFIED	CHAR(1)		Indicates if feature has been verified by the data source. Constraint checks this is 'Y' or 'N'. Null indicates the field has not been verified yet, 'N' indicates it has been marked as suspect
SHAPE	NUMBER(38)		Must reference a feature id number stored in geometry table. Trigger to prevent update to or insert of a non-existent geometry id number

FEATUREPRODUCT

A type of data product that is available in a type of medium from a particular data vendor for a particular feature. Uniquely identified by (WELLID,THREEDSEIS_SURVEYID,TWODSEIS_SURVEYID,TWODSEIS_LINEID,PROVIDER_COMPID,PRODUCT_TYPE,PRODUCT_MEDIUM)

COLUMN	FORMAT	NULL?	Description
FEATUREPRODUCTID	NUMBER(38)	NOT NULL	PK, integer number generated automatically by a sequence, has no meaning outside of the DEAL database.
WELLID	NUMBER(38)		FK to WELL if the product is attached to a well feature. Part of unique key
THREEDSEIS_SURVEYID	NUMBER(38)		FK to THREEDSEIS_SURVEY if the product is attached to a 3d seismic feature. Part of unique key
TWODSEIS_SURVEYID	NUMBER(38)		FK to TWODSEIS_SURVEY if the product is attached to a 2d seismic feature. Part of unique key
TWODSEIS_LINEID	NUMBER(38)		FK to TWODSEIS_LINE if the product is attached to a 2d line feature. Part of unique key

PROVIDER_COMPID	VARCHAR2(40)	NOT NULL	data provider company, FK to COMPANY. Part of unique key
PRODUCT_TYPE	VARCHAR2(20)	NOT NULL	type of product, FK to DIC_PRODUCT_TYPE. Part of unique key
PRODUCT_MEDIUM	VARCHAR2(20)	NOT NULL	medium that that product is stored on e.g. paper, tape, FK to DIC_MEDIUM. Part of unique key
PRODUCT_DETAILS_URL	VARCHAR2(80)	NOT NULL	URL to follow to get details of the product
PRODUCT_DETAILS_URLMETHOD	VARCHAR2(6)	NOT NULL	method for the PRODUCT_DETAILS_URL - 'get', 'post' or 'mailto'
ENTITLEMENT_URL	VARCHAR2(80)	NOT NULL	URL to follow for entitlement details
ENTITLEMENT_URLMETHOD	VARCHAR2(6)	NOT NULL	method for the ENTITLEMENT_URL - 'get', 'post' or 'mailto'
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

FEEDBACK

a feedback report received from website form. No PK or other constraints

COLUMN	FORMAT	NULL?	Description
LOGIN	VARCHAR2(20)		user login if the user has registered
DATE_ENTERED	DATE	NOT NULL	date of the feedback
IPADDRESS	VARCHAR2(80)	NOT NULL	IP address of the client
NAME	VARCHAR2(80)	NOT NULL	freely entered name of the user
COMPANY	VARCHAR2(80)		freely entered company name
EMAIL	VARCHAR2(80)	NOT NULL	freely entered email address
SUBJECT	VARCHAR2(80)	NOT NULL	subject area of the feedback comments
FEEDBACK_COMMENT	VARCHAR2(2000)	NOT NULL	comments made

HCFIELD_BOUNDING_PT

location of a point on the bounding polygon of a hydrocarbons field

COLUMN	FORMAT	NULL?	Description
HCFIELDID	NUMBER(38)	NOT NULL	Part PK. Identifier of the hydrocarbons field, FK to HYDROCARBONS_FIELD
SEQ_NO	NUMBER(10)	NOT NULL	Part PK, order number of this point around the boundary of the field
LATITUDE	NUMBER(10,7)		latitude of the point on the boundary, decimal degrees
LONGITUDE	NUMBER(11,7)		longitude of the point on the boundary, decimal degrees, negative west
EASTING	NUMBER(10,2)		easting of the point on the boundary, metres. Only allowed not null value if HYDROCARBONS_FIELD.PROJECTION_NAME is not null
NORTHING	NUMBER(10,2)		northing of the point on the boundary, metres. Only allowed not null value if HYDROCARBONS_FIELD.PROJECTION_NAME is not null
DATE_ENTERED	DATE	NOT NULL	

USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

HYDROCARBONS_FIELD

spatial feature of a named hydrocarbon field area. Uniquely identified by FIELDNAME

COLUMN	FORMAT	NULL?	Description
HCFIELDID	NUMBER(38)	NOT NULL	PK, database unique identifier of the hydrocarbons field. Integer number generated automatically by a sequence, has no meaning outside of the DEAL database
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
FIELDNAME	VARCHAR2(40)	NOT NULL	Freely entered name of the field. Unique key
HYDROCARBON_TYPE	VARCHAR2(4)	NOT NULL	type of hydrocarbon present in the field. Check constraint checks this is in ('OIL','GAS','OIL,GAS','COND').
OPERATOR	VARCHAR2(40)		freely entered name of company operating the field
EPSG_CODE	NUMBER(5)	NOT NULL	EPSG code for co-ordinate system of stored positional attributes, FK to DIC_EPSG
DATUM_NAME	VARCHAR2(10)	NOT NULL	Datum for co-ordinate system of stored positional attributes, FK to DIC_DATUM
SPHEROID_NAME	VARCHAR2(10)	NOT NULL	spheroid name for co-ordinate system of stored positional attributes, FK to DIC_SPHEROID
PROJECTION_NAME	VARCHAR2(10)		projection name for co-ordinate system if stored in eastings northings, FK to DIC_PROJECTION. Must be not null if HCFIELD_BOUNDING_PT.EASTING or NORTHING is not null
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		
VERIFIED	CHAR(1)		Indicates if feature has been verified by the data source. Constraint checks this is 'Y' or 'N'. Null indicates the field has not been verified yet, 'N' indicates it has been marked as suspect
SHAPE	NUMBER(38)		Must reference a feature id number stored in geometry table. Trigger to prevent update to or insert of a non-existent geometry id number

LICENCE

spatial feature of a licence area. Uniquely identified by (COUNTRY,LICENCE_DTINO). Subject to change post phase 2 to become uniquely identified by (COUNTRY,LICENCE_BLOCK)

COLUMN	FORMAT	NULL?	Description
LICENCEID	NUMBER(38)	NOT NULL	PK, database unique identifier of the licence. Integer number generated automatically by a sequence, has no meaning outside of the DEAL database
COUNTRY	CHAR(3)	NOT NULL	gmi code of country for which licence is defined, FK to COUNTRY. Part of unique key
LICENCE_LABEL	VARCHAR2(40)		the recognised label for the licence which will appear on the map. Consists of case-sensitive block, sub-block and suffix. Check constraint for case insensitive equality to LICENCE_BLOCK or null
LICENCE_DTINO	VARCHAR2(40)	NOT NULL	the DTI identifier for the licence - usually an integer number. Part of unique key. Subject to change post phase 2 to remove from unique key (this is not a widely recognised identifier, just one that exists for every record in the current dataset)
LICENCE_BLOCK	VARCHAR2(40)		licence block/sub-block and suffix stored in upper case to make it easily searchable. Constraint checks upper case. Subject to change post phase 2 (and when revised dataset is received in which this field is fully populated) to become not null and part of unique key with

			COUNTRY.
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
START_DATE	DATE		start date of the licence
END_DATE	DATE		end date of the licence - if null then the licence is current and can be seen as a map feature
LICENCE_TYPE	VARCHAR2(10)		type of licence, FK to DIC_LICENCE_TYPE
OPERATOR	VARCHAR2(40)		licence operator company. Freely entered company name
LICENCE_ROUND	NUMBER(4)		licence round during which licence was issued
EPSG_CODE	NUMBER(5)	NOT NULL	EPSG code for co-ordinate system of stored positional attributes, FK to DIC_EPSG
DATUM_NAME	VARCHAR2(10)	NOT NULL	Datum for co-ordinate system of stored positional attributes, FK to DIC_DATUM
SPHEROID_NAME	VARCHAR2(10)	NOT NULL	spheroid name for co-ordinate system of stored positional attributes, FK to DIC_SPHEROID
PROJECTION_NAME	VARCHAR2(10)		projection name for co-ordinate system if stored in eastings northings, FK to DIC_PROJECTION. Cannot be null if LICENCE_BOUNDING_LINE.START_EASTING or START_NORTHING or END_EASTING or END_NORTHING are not null
LICENCE_ISSUED_FLAG	CHAR(1)	NOT NULL	flag to indicate whether licence has been issued. Constraint checks this is 'Y' or 'N'
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		
VERIFIED	CHAR(1)		Indicates if feature has been verified by the data source. Constraint checks this is 'Y' or 'N'. Null indicates the field has not been verified yet, 'N' indicates it has been marked as suspect
SHAPE	NUMBER(38)		Must reference a feature id number stored in geometry table. Trigger to prevent update to or insert of a non-existent geometry id number

LICENCE_BOUNDING_LINE

A line which defines part of a licence boundary

COLUMN	FORMAT	NULL?	Description
LICENCEID	NUMBER(38)	NOT NULL	PK, identifier of licence feature which this bounding line describes
LINE_SEQUENCE	NUMBER(10)	NOT NULL	order number of this line section in the sequence of line sections that define the licence boundary
LINE_TYPE	VARCHAR2(10)	NOT NULL	type of bounding line, FK to DIC_DEFINED_LINE_TYPE
START_LATITUDE	NUMBER(10,7)		latitude of start point of the bounding line, decimal degrees
START_LONGITUDE	NUMBER(11,7)		longitude of start point of the bounding line, decimal degrees, negative west
END_LATITUDE	NUMBER(10,7)		latitude of the end point of the bounding line, decimal degrees
END_LONGITUDE	NUMBER(11,7)		longitude of the end point of the bounding line, decimal degrees, negative west
START_EASTING	NUMBER(10,2)		easting of start point of the bounding line, in metres. Trigger to prevent not null values here if LICENCE.PROJECTION_NAME is null
START_NORTHING	NUMBER(10,2)		northing of start point of the bounding line, in metres. Trigger to prevent not null values here if LICENCE.PROJECTION_NAME is null
END_EASTING	NUMBER(10,2)		easting of the end point of the bounding line, in metres. Trigger to prevent not null values here if LICENCE.PROJECTION_NAME is

			null
END_NORTHING	NUMBER(10,2)		northing of the end point of the bounding line, in metres. Trigger to prevent not null values here if LICENCE.PROJECTION_NAME is null
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

LOG_DOWNLOAD

An instance of a client downloading a set of data from the deal website. No PK

COLUMN	FORMAT	NULL?	Description
LOGIN	VARCHAR2(20)		user login name if user is registered
TIME	DATE	NOT NULL	time and date stamp of the download
IPADDRESS	VARCHAR2(80)	NOT NULL	IP address of the client
TYPE	VARCHAR2(40)	NOT NULL	type of dataset downloaded, FK to DIC_DOWNLOAD_TYPE

LOG_INFOREQUEST

a product information request made by a user to a data provider through the DEAL website. No PK and no constraints

COLUMN	FORMAT	NULL?	Description
LOGIN	VARCHAR2(20)		login name of the user if they are registered
TIME	DATE	NOT NULL	time and date stamp of the request
IPADDRESS	VARCHAR2(80)	NOT NULL	IP address of the client
FEATUREPRODUCTID	NUMBER(38)	NOT NULL	identifier of the featureproduct that has been requested

LOG_LOGIN

An instance of a client logging on to the DEAL website, with username or anonymously. No PK or other constraints

COLUMN	FORMAT	NULL?	Description
LOGIN	VARCHAR2(20)		user login name if login has been by a registered user
TIME	DATE	NOT NULL	time and date stamp of the login
IPADDRESS	VARCHAR2(80)	NOT NULL	IP address of the client

MEDIAN_LINE

spatial feature of a median line separating offshore areas belonging to different nations. No natural unique identifier in current data

COLUMN	FORMAT	NULL?	Description
MEDIAN_LINEID	NUMBER(38)	NOT NULL	PK, database unique identifier of the median line feature. Integer number generated automatically by a sequence, has no meaning outside of the DEAL database
MEDIAN_LINE_LABEL	VARCHAR2(80)		map label for the feature
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
EPSG_CODE	NUMBER(5)	NOT NULL	EPSG code for co-ordinate system of stored positional attributes, FK to DIC_EPSG
DATUM_NAME	VARCHAR2(10)	NOT NULL	Datum for co-ordinate system of stored positional attributes, FK to DIC_DATUM
SPHEROID_NAME	VARCHAR2(10)	NOT NULL	spheroid name for co-ordinate system of stored positional attributes, FK to DIC_SPHEROID

PROJECTION_NAME	VARCHAR2(10)		projection name for co-ordinate system if stored in easting northings, FK to DIC_PROJECTION
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		
VERIFIED	CHAR(1)		Indicates if feature has been verified by the data source. Constraint checks this is 'Y' or 'N'. Null indicates the field has not been verified yet, 'N' indicates it has been marked as suspect
SHAPE	NUMBER(38)		Must reference a feature id number stored in geometry table. Trigger to prevent update to or insert of a non-existent geometry id number

PIPELINE

spatial feature of a section of pipeline or umbilical line. No natural unique identifier - a single pipeline_name may be attributed to more than one pipeline section

COLUMN	FORMAT	NULL?	Description
PIPELINEID	NUMBER(38)	NOT NULL	PK, database unique identifier of the pipeline section. Primary key of the table. Integer number generated automatically by a sequence, has no meaning outside of the DEAL database
PIPELINE_NAME	VARCHAR2(40)	NOT NULL	Name of the pipeline. Check constraint so this doesn't contain double quote character - replace with inch.
DESCRIPTION	VARCHAR2(80)		Description of the pipeline, NB diameter, type of fluid, operating company name are usually found embedded in the name. Check constraint so this doesn't contain double quote character - replace with inch.
OPERATOR	VARCHAR2(40)		Freely entered name of company operating the pipeline
FLUID_CONVEYED	VARCHAR2(30)		Freely entered field indicating what the pipeline is used for e.g. Water, POWER.
PHYSICAL_STATUS	VARCHAR2(10)		Codes for physical pipeline status, may be a combination of more than one code from DIC_SEAFISH_FEATURE where FEATURE_GROUP='PIPELINE STATUS'
STATUS	VARCHAR2(11)		operational status of the feature, check constraint checks in 'ACTIVE','PROPOSED','DISUSED'
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
EPSG_CODE	NUMBER(5)	NOT NULL	EPSG code for co-ordinate system of stored positional attributes, FK to DIC_EPSG
DATUM_NAME	VARCHAR2(10)	NOT NULL	Datum for co-ordinate system of stored positional attributes, FK to DIC_DATUM
SPHEROID_NAME	VARCHAR2(10)	NOT NULL	spheroid name for co-ordinate system of stored positional attributes, FK to DIC_SPHEROID
PROJECTION_NAME	VARCHAR2(10)		projection name for co-ordinate system if stored in eastings northings, FK to DIC_PROJECTION. Must be not null if PIPELINE_PT.EASTING or NORTHING or PIPELINE_SPAN_PT.EASTING or NORTHING are not null
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
VERIFIED	CHAR(1)		Indicates if feature has been verified by the data source. Constraint checks this is 'Y' or 'N'. Null indicates the field has not been verified yet, 'N' indicates it has been marked as suspect
SHAPE	NUMBER(38)		Must reference a feature id number stored in geometry table. Trigger to prevent update to or insert of a non-existent geometry id number

PIPELINE_PT

location of a point along a pipeline feature

COLUMN	FORMAT	NULL?	Description
PIPELINEID	NUMBER(38)	NOT NULL	Part PK, identifier of pipeline feature, FK to PIPELINE
SEQ_NO	NUMBER(10)	NOT NULL	Part PK, order number of this located point along the pipeline
LATITUDE	NUMBER(10,7)		latitude of the point on the line, decimal degrees
LONGITUDE	NUMBER(11,7)		longitude of the point on the line, decimal degrees, negative west
EASTING	NUMBER(10,2)		easting of the point on the line, metres. Only allowed not null value if PIPELINE.PROJECTION_NAME is not null
NORTHING	NUMBER(10,2)		northing of the point on the line, metres. Only allowed not null value if PIPELINE.PROJECTION_NAME is not null
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

PIPELINE_SPAN

spatial feature of section of a pipeline which is elevated above the sea floor. Post phase 2 data - unique identifier not yet determined

COLUMN	FORMAT	NULL?	Description
PIPELINE_SPANID	NUMBER(38)	NOT NULL	PK, database unique identifier of the pipeline span feature. Integer number generated automatically by a sequence, has no meaning outside of the DEAL database
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
PIPELINEID	NUMBER(38)	NOT NULL	pipeline which this span is part of, FK to PIPELINE
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		
VERIFIED	CHAR(1)		Indicates if feature has been verified by the data source. Constraint checks this is 'Y' or 'N'. Null indicates the field has not been verified yet, 'N' indicates it has been marked as suspect
SHAPE	NUMBER (38)		Must reference a feature id number stored in geometry table. Trigger to prevent update to or insert of a non-existent geometry id number

PIPELINE_SPAN_PT

location of a point on the line of a pipeline_span feature

COLUMN	FORMAT	NULL?	Description
PIPELINE_SPANID	NUMBER(38)	NOT NULL	Part PK, identifier of the pipeline span, FK to PIPELINE_SPAN
SEQ_NO	NUMBER(10)	NOT NULL	Part PK, order number of this located point along the pipeline span line
LATITUDE	NUMBER(10,7)		latitude of the point on the line, decimal degrees
LONGITUDE	NUMBER(11,7)		longitude of the point on the line, decimal degrees, negative west
EASTING	NUMBER(10,2)		easting of the point on the line, metres. Only allowed not null value if PIPELINE.PROJECTION_NAME is not null
NORTHING	NUMBER(10,2)		northing of the point on the line, metres. Only allowed not null value if PIPELINE.PROJECTION_NAME is not null
DATE_ENTERED	DATE	NOT NULL	

		NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

PRODTYPE_CLSN

Post phase 2 possibility: a company-defined classification of a product type into a product class

COLUMN	FORMAT	NULL?	Description
CLASSIFIER_COMPID	VARCHAR2(40)	NOT NULL	Part PK, company which provided this classification, FK to COMPANY
PRODUCT_TYPE	VARCHAR2(20)	NOT NULL	Part PK, the DEAL product grouping, FK to DIC_PRODUCT_TYPE
PRODUCT_CLASS	VARCHAR2(20)	NOT NULL	Part PK, the non-DEAL product grouping, FK to DIC_PRODUCT_CLASS
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

PRODUCT_ENTITLEMENT

The entitlement that a company has for a data vendor's product type of a particular feature. Not in use yet, development for post phase 2 - attributes may need to be added

COLUMN	FORMAT	NULL?	Description
FEATUREPRODUCTID	NUMBER(38)	NOT NULL	Part PK. Identifier of a (product type, data provider, feature, product medium) grouping, FK to FEATUREPRODUCT
SUBSCRIBER_COMPID	VARCHAR2(40)	NOT NULL	Part PK. Company which has an entitlement to this featureproductid, FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

PUBLIC_CONTACT

A contact name and other details for a registered user of type ‘PUBLIC’

COLUMN	FORMAT	NULL?	Description
LOGIN	VARCHAR2(20)	NOT NULL	PK, user not a data provider or subscriber but registered to pick products and store preferences, FK to DEAL_USER
PASSWORD	VARCHAR2(20)	NOT NULL	
CONTACT_NAME	VARCHAR2(80)		
EMAIL	VARCHAR2(80)		
COMPANY_NAME	VARCHAR2(80)		freely entered company name
COMPANY_TYPE	VARCHAR2(20)		foreign key to DIC_COMPANY_TYPE
STREETADDRESS	VARCHAR2(80)		
TOWN	VARCHAR2(80)		
POSTCODE	VARCHAR2(20)		
COUNTRY	VARCHAR2(40)		freely entered country name
TELEPHONE	VARCHAR2(20)		contact telephone number
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT	

		NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

QUAD

spatial feature of administration quad area. Uniquely identified by (COUNTRY,QUADNO)

COLUMN	FORMAT	NULL?	Description
QUADID	NUMBER(37)	NOT NULL	PK, database unique identifier of the quad. Integer number generated automatically by a sequence, has no meaning outside of the DEAL database
COUNTRY	CHAR(3)	NOT NULL	gmi code of country for which quad is defined. FK to COUNTRY. Part of unique key
QUADNO	VARCHAR2(5)	NOT NULL	the real world identifier for the quad, integer number but stored as char field to enable map finds. Part of unique key
QUADLABEL	VARCHAR2(5)		the label for the quad which will appear on the map. Constraint checks QUADLABEL= QUADNO or is null
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		
VERIFIED	CHAR(1)		Indicates if feature has been verified by the data source. Constraint checks this is 'Y' or 'N'. Null indicates the field has not been verified yet, 'N' indicates it has been marked as suspect
SHAPE	NUMBER(38)		Must reference a feature id number stored in geometry table. Trigger to prevent update to or insert of a non-existent geometry id number

SAFETY_EX_ZONE

spatial feature of an offshore safety or exclusion zone area, defined by the distance from a central point, line or polygon structure. Uniquely identified by SAFETY_EX_ZONE_LABEL

COLUMN	FORMAT	NULL?	Description
SAFETY_EX_ZONEID	NUMBER(38)	NOT NULL	PK, database unique identifier of the safety zone feature. Integer number generated automatically by a sequence, has no meaning outside of the DEAL database
SAFETY_EX_ZONE_LABEL	VARCHAR2(80)		map label for the feature. Unique key - made by adding rowid number in Arcview to the POINTTITLE
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
DESCRIPTION	VARCHAR2(500)		
OPERATOR	VARCHAR2(40)	NOT NULL	freely entered name of company operating the feature at the centre of the safety zone
STATUS	VARCHAR2(11)		operational status of the safety zone, constraint checks in ('ACTIVE','DISUSED','PROPOSED')
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ISSUED	DATE		Date of issue of the safety zone feature
CENTRE_FEATURE_TYPE	CHAR(4)		type of structure or feature around which safety zone is defined. FK to DIC_SEAFISH_FEATURE
CENTRE_GEOMETRY_TYPE	VARCHAR2(10)		the type of geometry from which the safety zone distance is measured. Constraint checks this is POINT, LINE or POLYGON
DISTANCE_FROM_CENTRE	NUMBER(12,4)	NOT NULL	the distance from the centre geometry which defines the limit of the safety zone area
DISTANCE_UNITS	CHAR(1)	NOT NULL	units of DISTANCE_FROM_CENTRE value
EPSG_CODE	NUMBER(5)	NOT NULL	EPSG code for co-ordinate system of stored positional attributes, FK to DIC_EPSG

		NULL	DIC_EPSG
DATUM_NAME	VARCHAR2(10)	NOT NULL	Datum for co-ordinate system of stored positional attributes, FK to DIC_DATUM
SPHEROID_NAME	VARCHAR2(10)	NOT NULL	spheroid name for co-ordinate system of stored positional attributes, FK to DIC_SPHEROID
PROJECTION_NAME	VARCHAR2(10)		projection name for co-ordinate system if stored in easting northings, FK to DIC_PROJECTION
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		
VERIFIED	CHAR(1)		Indicates if feature has been verified by the data provider. Constraint checks this is 'Y' or 'N'. Null indicates the field has not been verified yet, 'N' indicates it has been marked as suspect
SHAPE	NUMBER(38)		Must reference a feature id number stored in geometry table. Trigger to prevent update to or insert of a non-existent geometry id number

SAFETY_EX_ZONE_CENTRE_PT

location of a point that defines a single point, a line or a polygon around which a safety zone is defined

COLUMN	FORMAT	NULL?	Description
SAFETY_EX_ZONEID	NUMBER(38)	NOT NULL	Part PK, database unique identifier of the safety zone feature. FK to SAFETY_EX_ZONE
SEQ_NO	NUMBER(10)	NOT NULL	Part PK. Sequence number of this located point in the definition of the safety zone centre
LATITUDE	NUMBER(10,7)		latitude of the point on the boundary, decimal degrees
LONGITUDE	NUMBER(11,7)		longitude of the point on the boundary, decimal degrees, negative west
EASTING	NUMBER(10,2)		easting of the point on the boundary, metres. Only allowed not null value if SAFETY_EX_ZONE.PROJECTION_NAME is not null
NORTHING	NUMBER(10,2)		northing of the point on the boundary, metres. Only allowed not null value if SAFETY_EX_ZONE.PROJECTION_NAME is not null
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

SEASURFACE_INF

spatial feature of offshore infrastructure on the sea surface. May coincide with some SURFACE infrastructure, e.g. platforms, moored buoys. Uniquely identified by SEASURFACE_INF_LABEL

COLUMN	FORMAT	NULL?	Description
SEASURFACE_INFID	NUMBER(38)	NOT NULL	PK, database unique identifier of the surface feature. Integer number generated automatically by a sequence, has no meaning outside of the DEAL database
SEASURFACE_INF_LABEL	VARCHAR2(80)	NOT NULL	map label for the feature. Unique key - made by adding rowid number in Arcview to the POINTTITLE supplied by Seafish
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
DESCRIPTION	VARCHAR2(500)		full name of the feature
OPERATOR	VARCHAR2(80)		freely entered name of the company operating the infrastructure feature
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ISSUED	DATE		Date of issue of the safety zone feature
STATUS	VARCHAR2(11)		operational status of the feature. Constraint checks this is in 'ACTIVE','DISUSED','PROPOSED'
FEATURE_TYPE	CHAR(4)		type of structure or feature. FK to DIC_SEAFISH_FEATURE

EPSG_CODE	NUMBER(5)	NOT NULL	EPSG code for co-ordinate system of stored positional attributes, FK to DIC_EPSG
DATUM_NAME	VARCHAR2(10)	NOT NULL	Datum for co-ordinate system of stored positional attributes, FK to DIC_DATUM
SPHEROID_NAME	VARCHAR2(10)	NOT NULL	spheroid name for co-ordinate system of stored positional attributes, FK to DIC_SPHEROID
PROJECTION_NAME	VARCHAR2(10)		projection name for co-ordinate system if stored in easting northings, FK to DIC_PROJECTION
LATITUDE	NUMBER(10,7)		latitude of feature point location, decimal degrees
LONGITUDE	NUMBER(11,7)		longitude of feature point location, decimal degrees, negative west
EASTING	NUMBER(10,2)		easting of feature point location, metres. PROJECTION_NAME must be not null if this is not null
NORTHING	NUMBER(10,2)		northing of feature point location, metres. PROJECTION_NAME must be not null if this is not null
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		
VERIFIED	CHAR(1)		Indicates if feature has been verified by the data source. Constraint checks this is 'Y' or 'N'. Null indicates the field has not been verified yet, 'N' indicates it has been marked as suspect
PLATFORM_FLAG	CHAR(1)		Indicates if feature is a platform or not - required for setting map symbol. Constraint checks this is 'Y' or 'N'
SHAPE	NUMBER(38)		Must reference a feature id number stored in geometry table. Trigger to prevent update to or insert of a non-existent geometry id number

SUBSEA_INF

spatial feature of offshore infrastructure beneath the sea surface, which includes pipeline and cable features, suspended wells, wrecks etc. Uniquely identified by SUBSEA_INF_LABEL

COLUMN	FORMAT	NULL?	Description
SUBSEA_INFID	NUMBER(38)	NOT NULL	PK, database unique identifier of the subsurface feature. Integer number generated automatically by a sequence, has no meaning outside of the DEAL database
SUBSEA_INF_LABEL	VARCHAR2(80)	NOT NULL	map label for the feature. Unique key - made by adding rowid number in Arcview to the POINTTITLE supplied by Seafish
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
DESCRIPTION	VARCHAR2(500)		full name of the feature
OPERATOR	VARCHAR2(80)		freely entered name of the company operating the infrastructure feature
DATA_SOURCE	VARCHAR2(40)	NOT NULL	company that supplied this information to DEAL. FK to COMPANY
DATE_ISSUED	DATE		Date of issue of the subsurface infrastructure feature
STATUS	VARCHAR2(11)		operational status of the feature. Constraint checks this is in 'ACTIVE','DISUSED','PROPOSED'
FEATURE_TYPE	CHAR(4)		type of structure or feature. FK to DIC_SEAFISH_FEATURE
EPSG_CODE	NUMBER(5)	NOT NULL	EPSG code for co-ordinate system of stored positional attributes, FK to DIC_EPSG
DATUM_NAME	VARCHAR2(10)	NOT NULL	Datum for co-ordinate system of stored positional attributes, FK to DIC_DATUM
SPHEROID_NAME	VARCHAR2(10)	NOT NULL	spheroid name for co-ordinate system of stored positional attributes, FK to DIC_SPHEROID
PROJECTION_NAME	VARCHAR2(10)		projection name for co-ordinate system if stored in eastings northings, FK to DIC_PROJECTION
LATITUDE	NUMBER(10,7)		latitude of feature point location, decimal degrees
LONGITUDE	NUMBER(11,7)		longitude of feature point location, decimal degrees, negative west
EASTING	NUMBER(10,2)		easting of feature point location, metres. PROJECTION_NAME must be not null if this is not null

NORTHING	NUMBER(10,2)		northing of feature point location, metres. PROJECTION_NAME must be not null if this is not null
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		
VERIFIED	CHAR(1)		Indicates if feature has been verified by the data source. Constraint checks this is 'Y' or 'N'. Null indicates the field has not been verified yet, 'N' indicates it has been marked as suspect
SUSWELL_FLAG	CHAR(1)		Indicates if feature is a suspended well or not - required for setting map symbol. Constraint checks this is 'Y' or 'N'
SHAPE	NUMBER(38)		Must reference a feature id number stored in geometry table. Trigger to prevent update to or insert of a non-existent geometry id number

THREEDSEIS_ALIAS

A company's alternative name for a 3D seismic survey. Uniquely identified by (THREEDSEIS_SURVEYID,ALIAS_COMPANY,THREEDSEIS_ALIAS_NAME)

COLUMN	FORMAT	NULL?	Description
THREEDSEIS_SURVEYID	NUMBER(38)	NOT NULL	Part PK, identifier of 3D seismic survey which has an alternative name, FK to THREEDSEIS_SURVEY.
ALIAS_COMPANY	VARCHAR2(40)	NOT NULL	Part PK; company which recognises the alternative survey name, FK to COMPANY.
THREEDSEIS_ALIAS_NAME	VARCHAR2(40)	NOT NULL	Part PK, alternative name for the survey.
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

THREEDSEIS_BOUNDING_PT

Location of a point on the bounding polygon of a three D seismic survey

COLUMN	FORMAT	NULL?	Description
THREEDSEIS_SURVEYID	NUMBER(38)	NOT NULL	Part PK, identifier of three d seismic survey which this point makes bounding polygon for, FK to THREEDSEIS_SURVEY
SEQ_NO	NUMBER(10)	NOT NULL	Part PK. Sequence number of this located point around the survey boundary
LATITUDE	NUMBER(10,7)		latitude of the point on the boundary, decimal degrees
LONGITUDE	NUMBER(11,7)		longitude of the point on the boundary, decimal degrees, negative west
EASTING	NUMBER(10,2)		easting of the point on the boundary, metres. Only allowed not null value if THREEDSEIS_SURVEY.PROJECTION_NAME is not null
NORTHING	NUMBER(10,2)		northing of the point on the boundary, metres. Only allowed not null value if THREEDSEIS_SURVEY.PROJECTION_NAME is not null
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

THREEDSEIS_SURVEY

A spatial feature of a 3D seismic survey area. Uniquely identified by (THREEDSEIS_CS9NAME)

COLUMN	FORMAT	NULL?	Description
THREEDSEIS_SURVEYID	NUMBER(38)	NOT NULL	PK, database unique identifier of the survey. Primary key of the table. Integer number generated automatically by a sequence, has no meaning outside of the DEAL database

THREEDSEIS_CS9NAME	CHAR(10)	NOT NULL	CS9 standardised survey name as defined by CDA. Must be upper case. Unique key
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
COUNTRY	CHAR(3)	NOT NULL	GMI code of country containing survey, FK to COUNTRY
DATA_PROVIDER	VARCHAR2(40)	NOT NULL	company which provided the survey information to DEAL, FK to COMPANY
DATA_ORIGIN	VARCHAR2(40)		company originating the survey information, FK to COMPANY
OPERATOR	VARCHAR2(40)		freely entered name of company or companies operating the survey. Must be upper case
CONTRACTOR	VARCHAR2(40)		freely entered name of contractor company or companies for the survey
START_DATE_SHOT	DATE		start date of survey
END_DATE_SHOT	DATE		end date of survey
SURVEY_TYPE	VARCHAR2(10)		type of survey, FK to DIC_SURVEY_TYPE
SURVEY_ENVIRONMENT	VARCHAR2(10)		enviroment that the survey was conducted in, FK to DIC_SURVEY_ENVIRONMENT
RECORD_LENGTH	NUMBER(7,4)		length of the seismic record produced, in seconds two way time
SOURCE_TYPE	VARCHAR2(30)		type of seismic source used e.g. Air gun, Water Gun
EPSG_CODE	NUMBER(5)	NOT NULL	EPSG code for co-ordinate system of stored positional attributes, FK to DIC_EPSG
DATUM_NAME	VARCHAR2(10)	NOT NULL	Datum for co-ordinate system of stored positional attributes, FK to DIC_DATUM
SPHEROID_NAME	VARCHAR2(10)	NOT NULL	spheroid name for co-ordinate system of stored positional attributes, FK to DIC_SPHEROID
PROJECTION_NAME	VARCHAR2(10)		projection name for co-ordinate system if stored in eastings northings, FK to DIC_PROJECTION. Must be not null if THREEDSEIS_BOUNDING_PT.EASTING or NORTHING is not null
COMMENTS	VARCHAR2(500)		
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		
VERIFIED	CHAR(1)		Indicates if feature has been verified by the data provider. Constraint checks this is 'Y' or 'N'. Null indicates the field has not been verified yet, 'N' indicates it has been marked as suspect
SHAPE	NUMBER(38)		Must reference a feature id number stored in geometry table. Trigger to prevent update to or insert of a non-existent geometry id number

TWODSEIS_ALIAS

A company's alternative name for a 2D seismic survey. Uniquely identified by (TWODSEIS_SURVEYID,ALIAS_COMPANY,TWODSEIS_ALIAS_NAME)

COLUMN	FORMAT	NULL?	Description
TWODSEIS_SURVEYID	NUMBER(38)	NOT NULL	Part PK, identifier of 2D seismic survey which has an alternative name, FK to TWODSEIS_SURVEY.
ALIAS_COMPANY	VARCHAR2(40)	NOT NULL	Part PK; company which recognises the alternative survey name, FK to COMPANY.
TWODSEIS_ALIAS_NAME	VARCHAR2(50)	NOT NULL	Part PK, alternative name for the survey.
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

TWODSEIS_BOUNDING_PT

As for threedseis_bounding_pt. Not in use for phase 2.

COLUMN	FORMAT	NULL?	Description
TWODSEIS_SURVEYID	NUMBER(38)	NOT NULL	
SEQ_NO	NUMBER(10)	NOT NULL	
LATITUDE	NUMBER(10,7)		
LONGITUDE	NUMBER(11,7)		
EASTING	NUMBER(10,2)		
NORTHING	NUMBER(10,2)		
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

TWODSEIS_LINE

Survey line of a 2d seismic survey. Not in use for phase 2.

COLUMN	FORMAT	NULL?	Description
TWODSEIS_LINEID	NUMBER(38)	NOT NULL	PK
TWODSEIS_SURVEYID	NUMBER(38)	NOT NULL	FK to TWOSEIS_SURVEY
TWODLINE_CS9NAME	VARCHAR2(14)	NOT NULL	
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

TWODSEIS_LINE_PT

Location of a point along a twodseis_line. Not in use for phase 2.

COLUMN	FORMAT	NULL?	Description
TWODSEIS_LINEID	NUMBER(38)	NOT NULL	Part PK, FK to TWODSEIS_LINE
SEQ_NO	NUMBER(10)	NOT NULL	Part PK; identifying number of this point along the line
LATITUDE	NUMBER(10,7)		
LONGITUDE	NUMBER(11,7)		
EASTING	NUMBER(10,2)		
NORTHING	NUMBER(10,2)		
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

TWODSEIS_SURVEY

A spatial feature of a 2D seismic survey area. Uniquely identified by (TWODSEIS_CS9NAME). Phase 2 data

COLUMN	FORMAT	NULL?	Description
TWODSEIS_SURVEYID	NUMBER(38)	NOT NULL	PK, database unique identifier of the survey. Integer number generated automatically by a sequence, has no meaning outside of the DEAL database
TWODSEIS_CS9NAME	CHAR(10)	NOT NULL	CS9 standardised survey name. Must be upper case. Unique key
COUNTRY	CHAR(3)	NOT NULL	GMI code of country containing survey, FK to COUNTRY
VISIBILITY	CHAR(1)	NOT NULL	visibility of map feature, FK to DIC_VISIBILITY
DATA_PROVIDER	VARCHAR2(40)	NOT NULL	company which provided the survey information to DEAL, FK to COMPANY
DATA_PROVIDER_SUPPLIER	VARCHAR2(14)	NOT NULL	Supplier or data store source of data used by the DATA_PROVIDER. Determines the map layer that the surveys are displayed on. Constraint checks this is 'DTI' for surveys originally stored in the DTI database and then archived by CDA or 'CDA Repository' for surveys stored by CDA that were supplied to them by their clients. Subject to change post phase 2 to allow further values for surveys coming directly to DEAL from operator companies.
DATA_ORIGIN	VARCHAR2(40)		company originating the survey information, FK to COMPANY
OPERATOR	VARCHAR2(40)		freely entered name of company or companies operating the survey.
CONTRACTOR	VARCHAR2(40)		freely entered name of contractor or contractors for the survey
START_DATE_SHOT	DATE		start date of survey
END_DATE_SHOT	DATE		end date of survey
NO_OF_LINES	NUMBER		number of lines in the survey
RECORD_LENGTH	NUMBER(7,4)		length of the seismic record produced, in seconds two way time
SOURCE_TYPE	VARCHAR2(30)		type of seismic energy source used
EPSG_CODE	NUMBER(5)	NOT NULL	EPSG code for co-ordinate system of stored positional attributes, FK to DIC_EPSG
DATUM_NAME	VARCHAR2(10)	NOT NULL	Datum for co-ordinate system of stored positional attributes, FK to DIC_DATUM
SPHEROID_NAME	VARCHAR2(10)	NOT NULL	spheroid name for co-ordinate system of stored positional attributes, FK to DIC_SPHEROID
PROJECTION_NAME	VARCHAR2(10)		projection name for co-ordinate system if stored in eastings northings, FK to DIC_PROJECTION
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		
VERIFIED	CHAR(1)		Indicates if feature has been verified by the data provider. Constraint checks this is 'Y' or 'N'. Null indicates the field has not been verified yet, 'N' indicates it has been marked as suspect
SHAPE	NUMBER(38)		Must reference a feature id number stored in geometry table. Trigger to prevent update to or insert of a non-existent geometry id number

USER_INTEREST_BOX

A rectangular area that a user has defined as an area of interest to them, uniquely identified by LOGIN and BOX_NO

COLUMN	FORMAT	NULL?	Description
LOGIN	VARCHAR2(20)	NOT NULL	Part PK, user login, FK to DEAL_USER
BOX_NO	NUMBER(4)	NOT NULL	Part PK, integer number assigned to the box to uniquely identify it from other boxes that this user may define - website scripts only allow one box per login at the moment so this number is always 1
LONGITUDE_EAST	NUMBER(11,7)		eastern limit defining the box, decimal degrees longitude, negative west - null means whole world

LONGITUDE_WEST	NUMBER(11,7)		western limit defining the box , decimal degrees longitude, negative west- null means whole world
LATITUDE_NORTH	NUMBER(10,7)		northern limit defining the box, decimal degrees latitude - null means whole world
LATITUDE_SOUTH	NUMBER(10,7)		southern limit defining the box, decimal degrees latitude - null means whole world
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

USER_INTEREST_PRODUCT

A product type that is of interest to a user within one of his interest areas, uniquely identified by (LOGIN,BOX_NO,PRODUCT_TYPE)

COLUMN	FORMAT	NULL?	Description
LOGIN	VARCHAR2(20)	NOT NULL	Part PK, composite FK to USER_INTEREST_BOX
BOX_NO	NUMBER(4)	NOT NULL	Part PK, composite FK to USER_INTEREST_BOX
PRODUCT_TYPE	VARCHAR2(20)	NOT NULL	Part PK, product type of interest, FK to DIC_PRODUCT_TYPE
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

USER_PRODUCT_PICK

A featureproduct that has been picked to be of interest by a user

COLUMN	FORMAT	NULL?	Description
LOGIN	VARCHAR2(20)	NOT NULL	part PK, user login, FK to DEAL_USER
FEATUREPRODUCTID	NUMBER(38)	NOT NULL	part PK, identifier of the feature-product-provider combination, FK to FEATUREPRODUCT
DATEPICKED	DATE	NOT NULL	
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

VALID_FEATUREPRODUCT

classification of a product type code as a valid DEAL product type for a class of feature

COLUMN	FORMAT	NULL?	Description
FEATURE_CLASS_CODE	CHAR(2)	NOT NULL	Part PK, class of feature that has valid product types, FK to DIC_FEATURE_CLASS
PRODUCT_TYPE	VARCHAR2(20)	NOT NULL	Part PK, product type that can be attached to a feature of this class, FK to DIC_PRODUCT_TYPE
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	

DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

WELL

The details about a drilled well bore, which must have a reference to a DTI registered well position. The wellname may match the DTI well registration if it is the main or only well bore for the registration, or it may be a new name, e.g. with suffixes, if it is an additional well bore. Each well bore for one DTI_WELL may have its own version of the well information. Uniquely identified by (DTI_WELLID,WELLNAME)

COLUMN	FORMAT	NULL?	Description
WELLID	NUMBER(38)	NOT NULL	PK, database unique identifier of the well. Integer number generated automatically by a sequence, has no meaning outside of the DEAL database
DTI_WELLID	NUMBER(38)	NOT NULL	database identifier of the DTI defined well which is a SDE geometry feature. Part of unique key
WELLNAME	VARCHAR2(40)	NOT NULL	name by which the originating company knows this well. Part of unique key
DATA_PROVIDER	VARCHAR2(40)	NOT NULL	company which provided the well information to DEAL, FK to COMPANY
DATA_ORIGIN	VARCHAR2(40)		company originating the well information. Should be DTI where the wellname is equal to the dti well registration number, otherwise CDA or the operator company. FK to COMPANY. Subject to change post phase 2 to be made a mandatory column.
OPERATOR	VARCHAR2(40)		freely entered company name of the well operator. Must be upper case
WELL_INTENT	CHAR(1)		FK to DIC_WELL_INTENT
WELL_TYPE	VARCHAR2(3)		FK to DIC_WELL_TYPE
MECHANICAL_RESULT	VARCHAR2(10)		Result or output of well e.g. Oil, Gas, Dry
SPUD_DATE	DATE		date that drilling started
END_OF_OP_DATE	DATE		date that operations ended for this well
RELEASE_STATUS	CHAR(1)		Code to indicate if the well information is commercial in confidence or has been publicly released. Must be 'Y' (confidential) or 'N' (not confidential, released)
VERTICAL_UNITS	CHAR(1)		Units that vertical measurements (depth, datum elevation) are stored in. Must be 'F' for feet or 'M' for metres
WELL_DATUM_TYPE	VARCHAR2(3)		type of elevation datum used to measure well depth from, FK to DIC_WELL_DATUM_TYPE
DATUM_ELEVATION	NUMBER(7,2)		elevation of well depth datum in VERTICAL_UNITS
TOTAL_DEPTH	NUMBER(7,2)		total drilled depth of the well in VERTICAL_UNITS
EPSG_CODE	NUMBER(5)	NOT NULL	EPSG code for co-ordinate system of stored positional attributes, FK to DIC_EPSG
DATUM_NAME	VARCHAR2(10)	NOT NULL	Datum for co-ordinate system of stored positional attributes, FK to DIC_DATUM
SPHEROID_NAME	VARCHAR2(10)	NOT NULL	spheroid name for co-ordinate system of stored positional attributes, FK to DIC_SPHEROID
LATITUDE	NUMBER(10,7)		latitude of well location, decimal degrees
LONGITUDE	NUMBER(11,7)		longitude of well location, decimal degrees, negative west
PROJECTION_NAME	VARCHAR2(10)		projection name for co-ordinate system if stored in eastings northings, FK to DIC_PROJECTION. Must be not null if EASTING or NORTHING are not null
EASTING	NUMBER(10,2)		easting of well location, metres. PROJECTION_NAME must be not null if this is not null
NORTHING	NUMBER(10,2)		northing of well location, metres. PROJECTION_NAME must be not null if this is not null
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

WELL_ALIAS

A company's alternative name for a well bore. Uniquely identified by (WELLID,ALIAS_COMPANY,WELL_ALIAS_NAME)

COLUMN	FORMAT	NULL?	Description
WELLID	NUMBER(38)	NOT NULL	Part PK, identifier of well bore which has an alternative name, FK to WELL.
ALIAS_COMPANY	VARCHAR2(40)	NOT NULL	Part PK; company which recognises the alternative well name, FK to COMPANY.
WELL_ALIAS_NAME	VARCHAR2(40)	NOT NULL	Part PK, alternative name for the well bore.
DATE_ENTERED	DATE	NOT NULL	
USER_ENTERED	VARCHAR2(10)	NOT NULL	
DATE_UPDATED	DATE		
USER_UPDATED	VARCHAR2(10)		

Appendix 1 About DEAL

Contents

DEAL Summary

DEAL promotes and facilitates access to data and information relevant to the offshore exploration and production of hydrocarbons on the United Kingdom Continental Shelf. DEAL offers access to reference information and a multi-vendor data products catalogue.

DEAL is a free service open to all users and all data vendors.

DEAL will continue to develop, adding new data and facilities, steered by user feedback.

DEAL is owned by Common data Access Ltd (CDA), a subsidiary of the United Kingdom Offshore Operators Association (UKOOA).

The DEAL web site is designed and operated by the British Geological Survey (BGS).

The DEAL web site address is www.ukdeal.co.uk

DEAL Background

In December 1998 the Department of Trade and Industry (DTI) created a joint Oil and Gas Industry Task Force (OGITF) to explore opportunities and initiatives to encourage continued investment in the UK Oil and Gas Industry, to protect jobs and to facilitate operating in a low oil price environment. A number of work-groups were established which explored various suggestions and ideas, some of which are still ongoing. One of these work-groups was charged with reviewing Regulations and Licensing.

As a result of the work done by this work-group a new web based facility, code-named DEAL (Digital Energy Atlas & Library), has been commissioned by CDA. The DEAL site, launched on 15th September 2000, promotes and facilitates access to data and information relevant to the exploration and production of hydrocarbons on the UKCS.

Who's Behind DEAL?

Stakeholders

DEAL is a PILOT initiative and enjoys broad, pan-industry support.

CDA is a wholly-owned subsidiary of UKOOA. The members that form UKOOA are working through its committee-structure to support DEAL's objectives.

The **DTI** is a strong supporter of DEAL and intends increasingly to rely on DEAL as its principal means of distributing quality-assured UKCS data.

Many important Data Vendors have already committed to using the DEAL Data Market Place to promote their data products.

CDA participants are funding DEAL and are committed to improving and maintaining the quality of the service.

UKOOA

The **UK Offshore Operators Association (UKOOA)** is the representative organisation for the UK offshore oil and gas industry. Its thirty-one members are licensed by the British Government to explore for and produce hydrocarbons in UK waters.

The UKOOA web site address is www.oilandgas.org.uk

PILOT

PILOT was established on January 1 2000 and replaces the OGITF (see above). Its goal is to strive to ensure the Task Force vision and deliverables are achieved. PILOT is made up of twenty-three key Government representatives and recognised leaders from Industry and meets on a quarterly basis.

CDA

Common Data Access Limited (CDA) was established as a not-for-profit limited company by a consortium of UK based exploration and production companies in 1995. The company was purchased by UKOOA in June 2000 and is now a wholly-owned subsidiary of UKOOA.

In addition to being the contract holder for DEAL, CDA operates a data repository on behalf of a consortium of twenty-one oil companies and eight service companies. The data repository services are outsourced.

CDA has two full-time employees based in London – a Chief Executive Officer and a Commercial Manager.

The CDA web site address is www.cdal.com

BGS

The British Geological Survey is the UK's national centre for earth science information and its foremost supplier of geoscience solutions. It acquires and maintains up-to-date knowledge of the UK and its continental shelf by means of systematic geological, geophysical, geochemical, hydrogeological and geotechnical surveys underpinned by high quality research.

BGS has been awarded the contract to design, build and operate the DEAL services.

The BGS web site address is www.bgs.ac.uk

DEAL Services

DEAL provides three related services:

DEAL Information Services	- i*DEAL*	Released on 15th September 2000
DEAL Data Market Place	- u*DEAL*	Released on 15th September 2000
DEAL Unified Data Network	- e*DEAL*	Due for release in 2001

These are described in detail below.

i*DEAL* (DEAL Information Services)

Service Scope

Despite the maturity of the UKCS, little basic spatial information of quality is available publicly. In the past, there have been a number of initiatives to gather information together but these have generally neither been coordinated nor maintained. i***DEAL*** meets the demand for quick and simple access to a single, complete and reliable reference set of basic spatial and attribute data over the UKCS. See also DEAL and Data Quality.

CDA is working with the various industry and non-industry bodies that have sets of such information and, through BGS (the DEAL Contractor), is loading it to DEAL. BGS will put practices and processes in place to maintain and quality control the information.

Service Description

CDA is working with UKOOA and directly with operators and other acquisition companies to load complete and accurate sets of the agreed information to the DEAL index database, both as an initial legacy capture exercise and as an ongoing data loading service.

DEAL users can view (and eventually download) this information over the Internet in convenient formats.

Service Development

New types of information will be added to DEAL as it becomes more mature.

In the future, CDA intends to extend the scope of DEAL to encompass information from neighbouring countries, either by extending DEAL itself or by interfacing DEAL with other systems.

Users are encouraged to feed back their comments on DEAL. These comments are an important influence for system modifications and improvements in subsequent releases and for improving data quality and completeness.

Service Benefits

i*DEAL* brings many benefits:

- It provides a single up-to-date reference set of quality coordinate and attribute information sourced from the oil companies themselves
- Users no longer need to maintain their own data sets
- All information is available through one ‘portal’ for immediate access and (eventually) download
- It promotes naming consistency and standards
- It offers a simple way for operators to provide meta data to the DTI and related industries; and
- It provides a sound foundation upon which data vendors may build value-added products and services

These benefits in turn will enable:

- Cost savings
- Regulatory compliance
- Value-add opportunities
- Risk reduction
- Work process improvement
- Data quality and completeness

u*DEAL* (DEAL Data Market Place)

Service Scope

It is generally difficult for users to find all relevant data products over a given area of interest on the UKCS. They must explore along a number of different pathways in turn (their own data sets, other oil companies and the DTI or its recognised release agents) but they can rarely be confident to have unearthed all available data products. Moreover, users are invariably under time pressure and can only afford to explore the most popular and readily available sources.

With u***DEAL***, CDA is creating a comprehensive catalogue of all UKCS geotechnical data products in one place. The service represents a market place for users and vendors alike.

Service Description

Data vendors may publish details of their data products on DEAL by associating them with an existing spatial object (a well for example) or by creating objects on an existing map layer (creating an outline polygon showing the areal extent of say a regional geological report). In some cases entirely new map layers will be created to allow new data product groups to be shown.

For their part, prospective users, who are ‘in the market’ for data, can see and compare the various products on offer within their area of interest. They may then contact the vendor for further technical and commercial information via email/web site links.

Service Development

In the future, CDA intends to bring several developments to the Data Market Place:

- Expansion of the system to allow for B2B e-commerce, including electronic procurement
- Expansion to include other geographical areas
- On-line Data Product delivery options including viewing of sample data
- Interfacing DEAL with other data sources
- Seamless interfacing of data vendors' databases with DEAL

The user community will be closely associated with the direction of DEAL development via user forums.

Service Benefits

The u***DEAL*** service offers a number of significant benefits:

- It offers vendors a single 'route-to-market'
- It reduces search time for buyers (fewer places to look)
- It offers users a choice of data products
- It improves work processes and yields resource and productivity savings; and
- It enables licensees to meet some of their obligations under the seismic data release guidelines

These benefits in turn will enable:

- Work process improvement
- Value-add opportunities
- Risk reduction
- Regulatory compliance (for data release agents)
- Improved information access

e*DEAL* (DEAL Unified Data Network)

Service Scope

In contrast to the other two DEAL services, e***DEAL*** will be a subscriber-only service.

Under the Phase 2 DEAL development plan (to be delivered in 2001), DEAL will act as a portal, or gateway, to a network of distributed data repositories. By extending the Data Market Place to include entitlement and ordering functionality, DEAL will draw these distributed data stores together into one unified virtual system. Users will be able to see what data exists and what data they are entitled to.

The first major use of the system is likely to be in relation to post-stack seismic trace data. No matter where or with whom the data is stored, an inventory may be viewed through and orders for the data (or for entitlement to the data) may be placed via DEAL.

For economic reasons, it is improbable that the large file sizes associated with seismic trace data will actually be delivered electronically in the short term. This is however a medium term goal and will become commercially feasible as the requisite technology becomes affordable.

Service Description

Data owners will continue to store their data products in a repository of choice. They will be able to set entitlement to their data at this local level such that it is visible to their license and trade partners in e***DEAL***.

License and trade partners will no longer need to keep their own duplicated copies of data but will be able to access their data on an 'as-needed' basis via entitlement set by the data owner (the operator in most cases).

Changes to data ownership and entitlement brought about by asset sales and company acquisitions may be effected via e***DEAL*** simply by amending the entitlements to reflect the changes.

Data users will be able to see details of their entitlements for various types of data in one place through e***DEAL***. Seismic contractors will be able to set entitlement to licensed customers for their speculative data.

Data owners (whether oil companies or data vendors) will need to continue to make their own arrangements for the storage of their proprietary data. They will however need to consider a number of issues, such as connectivity and semantic standards, to ensure that they may link to e***DEAL*** effectively.

Service Development

e***DEAL*** service subscribers will have a strong influence over its developmental direction.

As telecommunications costs fall, wide bandwidth networks will become more affordable and are likely to be adopted as the preferred delivery mechanism across the e***DEAL*** network.

Seismic trace data is the priority data type and will occupy much of the operational effort over the first two or three years of the plan period. There is however interest in the inclusion of other data types at the early stages.

Service Benefits

The e***DEAL*** network service offers a number of significant benefits:

- It eliminates the cost of duplicated data storage
- It improves and simplifies distribution
- It yields resource and productivity savings
- It yields data quality and data preservation gains
- It creates 'workflow compression'
- It allows rapid access to all data relevant to investment decisions
- It allows licensees to meet their data obligations via the CDA/DTI Deed

- It offers interoperability and format options
- It prepares the way for high speed on-line access, ‘on-demand’, via secure networks; and
- It enables data owners to choose where they wish to store their data

These benefits in turn will enable:

- Significant cost reduction
- Value-add opportunities
- Substantial work process improvement
- Regulatory compliance (the deed and data release)
- Improved information access
- Data quality and completeness gains
- Inter-operability benefits

Appendix 2 DEAL Data Model Standards

DEAL Data Model Standards

Rob Pedley 23/06/00

1. Table names will be singular, not plural. E.g. WELL not WELLS
2. Attributes with the same name in different tables should contain the same data.
   E.g. a well name is not the same thing as a 2D seismic survey name, so don't call them both NAME, call them WELLNAME and 2DSSURVEYNAME. A loading date is the same kind of thing for every table, so do give it the same attribute name in all tables.
3. Dictionary tables are those which exist only to define a valid set of values for the attributes of other tables and to provide descriptions. They should be named with prefix DIC_
4. All tables will include the audit attributes:
   DATE_ENTERED type DATE
   USER_ENTERED type VARCHAR2(10)
   DATE_UPDATED type DATE
   USER_UPDATED type VARCHAR2(10)
5. Geographical Locations
   Refer to the document *GIS Related Data Model* for the standard way of storing spatial positions.

Appendix 3 DEAL Data Descriptions

DEAL's Data Policy

Rob Pedley, BGS

DEAL aims to:

Display on web maps and reports, the best data sets which are either in the public domain, or for which we have negotiated display permission.

Download free from our web site, the best data sets which are either in the public domain, or for which we have negotiated download permission.

Link to definitive sources for statutory definitions.

Link to commercial data providers offering enhanced data sets.

The geographical extent of DEAL's data coverage is the UK continental shelf (UKCS) for offshore data only.
Data downloads direct from the DEAL site are free, linked source sites may charge for data.

Data Set	*Information*	*Access*
Wells	**DEAL Wells Description**	Download from DEAL
3D Seismic Surveys	**DEAL 3D Seismic Surveys Description**	Download from DEAL
2D Seismic Surveys	**DEAL 2D Seismic Surveys Description**	DEAL download not available
Geotechnical Data Products	**DEAL Geotechnical Data Products Catalogue Description**	DEAL download not available
Other Sources:	**CGG** 3D seismic survey data products.	mailto:tabouzakhm@cgg.com
	Geco-Prakla 3D seismic survey data products.	mailto:gibbom@gatwick.geco-prakla.slb.com
	IHS Energy Well data products.	mailto:data.services@ihsenergy.com
	Western Geophysical 3D seismic survey data products.	www.bakerhughes.com/westerngeo/
Coastline	**DEAL Coastline Description**	Download from DEAL

Definitive Source:	**Ordnance Survey** UK maps at all scales.	www.ordnancesurvey.co.uk
Median lines	**DEAL Median Lines Description**	DEAL download not available
Definitive Source:	**HMSO** Defined as text in Statutory Instruments	www.hmso.gov.uk Suggested search terms *statutory instrument median*
Other Sources:	**ECL** NW Europe shelf concession blocks and international boundaries database.	www.ecqc.com
Quads & Blocks	**DEAL Quads & Blocks Description**	DEAL download not available
Definitive Source:	**DTI 'Brown Book'**	www.dbd-data.co.uk/bbonline/book.htm
Other Sources:	**ECL** NW Europe shelf concession blocks and international boundaries database.	www.ecqc.com
Licences	**DEAL Licences Description**	DEAL download not available
Definitive Source:	**DTI 'Brown Book'**	www.dbd-data.co.uk/bbonline/book.htm
Hydrocarbon Fields	**DEAL Hydrocarbon Fields Description**	DEAL download not available
Surface Infrastructure	**DEAL Surface Infrastructure Description**	DEAL download not available
Subsurface Infrastructure	**DEAL Sub-surface Infrastructure Description**	DEAL download not available
Pipelines	**DEAL Pipelines Description**	DEAL download not available
Safety Zones	**DEAL Safety Zones Description**	DEAL download not available

Appendix 4 DEAL GIS Related Data Model

DEAL GIS Related Data Model

Rob Pedley, BGS

23/06/00

1. Introduction

This is a data model for the entities and attributes used by the GIS. It covers Phase 1 requirements and addresses the anticipated method for data product entitlements for Phase 2.

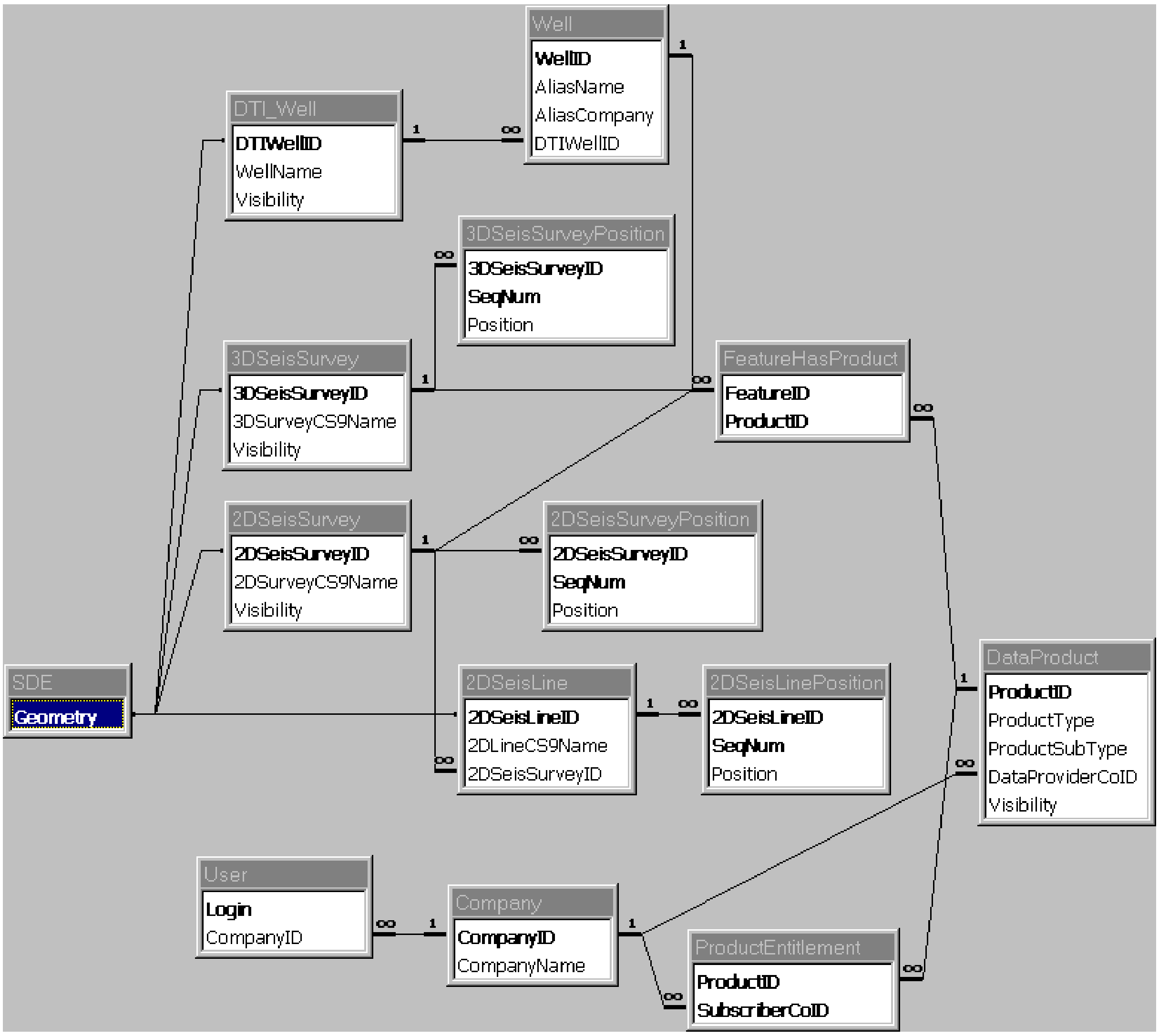
The main issues are :

- How geographical positions are to be stored
- What map projection is to be used
- How changes in feature geometry will be handled
- How subscribers will be given a map view of their entitlements in Phase 2

2. GIS Data Model ERD (Simplified)

The diagram shows the principles of relationships between the SDE, feature Oracle tables, Oracle position attributes, data products, data provider and entitled subscribers.

Several entities are omitted (coastline, licence etc.) and attribute lists are simplified.



3. Storage of Geographical Positions

3.1. Positional information used by the web maps will be stored in SDE, possibly with some static layers in shape files if this is required for performance (e.g. coastline).

3.2. Positions in the SDE will be loaded as decimal degrees of latitude and longitude to a precision equivalent to 0.001 sec of arc (7 decimal places for degrees).

3.3. The SDE co-ordinate system will be ED50. The ellipsoid and geodetic datum definitions for this system will be as defined by the EPSG.

3.4. The original positions and co-ordinate systems of data as provided to DEAL will be stored in Oracle table attributes. This will be modelled for all features except Coastline, Median lines, and Bathymetry. In the first release, these attributes will only be populated for Wells because the others will not be visible to users until data downloads are permitted.

3.5. 2D Seismic positions will be provided as 2D lines. These will be stored in a generalised form.

3.6. Original position attributes will be :

EPSG Code	Datum Name	Spheroid Name	Latitude	Longitude	Projection Name	Easting	Northing
number(5)	varchar2(40)	varchar2(40)	number(11,7)	number(11,7)	varchar2(40)	number(7)	number(7)

4230	ED50	International 1924	123.1234567	123.1234567			
23030	ED50	International 1924			UTM 30	1234567	1234567

Either (Latitude, Longitude) or (Projection, Easting, Northing) will be populated in any one row, not both.

EPSG Code, Datum Name, Spheroid Name, and Projection Name will be held as attributes of the highest possible entity, not for every value of Latitude, Longitude or Easting, Northing. E.g. for 2D seismic data they will be attributes of 2DseismicSurvey, but not attributes of 2DseismicPolygon or 2DseismicLine or 2DseismicLinePosition, because the same co-ordinate system is used for all navigation throughout a survey.

Given the EPSG code, the Datum, Spheroid and Projection names are redundant, but user will not be very familiar with EPSG codes, so it will be helpful to show them.

Latitude and longitude require 7 decimal places.

3.7. Co-ordinate conversion audit trail

When co-ordinate conversions are made to data, the seven transformation parameters are to be recorded in an audit log. This will mainly apply when loading data which was not provided in our internal unified co-ordinate system.

4. Map Projection and Unified Co-ordinate System

4.1. Phase 1 map projection will be ED50 / Lambert. This will be Lambert Azimuthal Equidistant, scale factor 1.0 at 6˚ West, 56˚ North (the centre of the area of interest). This is equivalent to Mercator.

4.2. Cursor co-ordinate display for Phase 1 should be in Latitude, Longitude if possible.

4.3. Map resolution will be sufficient to resolve spatial separation between features of 10 metres.

4.4. In Phase 2, subscribers will be able to select from a menu of predefined map projections.

5. Unique Feature Identifiers

5.1. The unique keys to all map feature entities will be 32 bit integer numbers (giving a range of 2,147,483,647 positive values). The numbers will be assigned from an Oracle SEQUENCE to ensure that they are unique and never re-used if a feature is deleted.

5.2. This system will allow values of the ‘natural’ identifiers for a feature (e.g. Well Registration) to be corrected after data loading, without deleting and reloading the record and modifying all related records, or relaxing the Oracle table integrity constraints.

6. Subscriber Map View of Entitlements and Visibility

6.1. In Phase 2, subscribers will be able to distinguish on the map, the features which have data products to which they are entitled, from the features which do not. E.g. if the subscriber is entitled to some well data products, those wells will have a different colour or symbol from the other wells. The above diagram shows the relationships involved.

6.2. Data products are associated only with Wells, 3D Seismic Polygons and 2D Seismic Polygons. Not with 2D Seismic Lines, nor with any other mapped features.

6.3. DataProduct has a many-to-many relationship with map features via table FeatureHasProduct *(this may be implemented as four separate tables WellHasProduct, 3DseisSurveyHasProduct etc. depending on results of performance tests)*

6.4. DataProduct has a many-to-many relationship with a subscriber Company via table ProductEntitlement. One ProductEntitlement entitles one subscriber company to one DataProduct.

6.5. A User belongs to one company and therefore has a one-to-many relationship with ProductEntitlement.

6.6. The web server software stores the authenticated subscriber's User.Login id as a session variable called CGI.REMOTE_USER. This is accessible to server side scripts such as ColdFusion to use in constructing database queries, giving a user specific view of the data.

6.7. The attribute tables for map features Well, 3DseisSurvey and 2DseisSurvey, and for DataProduct, all have a VISIBILITY attribute.

VISIBILITY takes the following values

Visibility Value	**Meaning**
UNRESTRICTED	This item is visible to public users
SUBSCRIBERS ONLY	Visible to all subscribers but not public users
BY ENTITLEMENT	Visible only to entitled subscribers (and to Vendor in the case of DataProduct)

For Phase 1, all VISIBILITY attributes will be UNRESTRICTED and map filtering will not be implemented. In Phase 2, the maps will be filtered to show only the features which should be visible to the current user according to these criteria.

The DataProduct Visibility attribute set to BY ENTITLEMENT can be used to withdraw products from a vendor's current catalogue, while still making them visible to entitled subscribers.

Appendix 5 Database Constraints and Other Database Objects

PRIMARY KEY AND UNIQUE KEY CONSTRAINTS

```
ALTER TABLE AOI ADD CONSTRAINT AOI_PK PRIMARY KEY (AOI_NAME);
ALTER TABLE BATHYMETRY ADD CONSTRAINT BATHYMETRY_PK PRIMARY KEY (BATHYMETRYID);
ALTER TABLE BLOCK ADD CONSTRAINT BLOCK_PK PRIMARY KEY (BLOCKID);
ALTER TABLE BLOCK ADD CONSTRAINT BLOCK_UK UNIQUE (COUNTRY,QUADNO,BLOCKNO);
ALTER TABLE COAST ADD CONSTRAINT COAST_PK PRIMARY KEY (COASTID);
ALTER TABLE COMPANY ADD CONSTRAINT COMPANY_PK PRIMARY KEY (COMPANYID);
ALTER TABLE COMPANY_CONTACT ADD CONSTRAINT COMPANY_CONTACT_PK PRIMARY KEY (COMPANYID,CONTACTREF);
ALTER TABLE CONTACTHASROLE ADD CONSTRAINT CONTACTHASROLE_PK PRIMARY KEY
      (COMPANYID,CONTACTREF,CONTACTROLE);
ALTER TABLE COUNTRY ADD CONSTRAINT COUNTRY_PK PRIMARY KEY (GMICODE);
ALTER TABLE DEAL_USER ADD CONSTRAINT DEAL_USER_PK PRIMARY KEY (LOGIN);
ALTER TABLE DIC_COMPANY_TYPE ADD CONSTRAINT DIC_COMPANY_TYPE_PK PRIMARY KEY (TYPE_CODE);
ALTER TABLE DIC_CONTACT_ROLE ADD CONSTRAINT DIC_CONTACT_ROLE_PK PRIMARY KEY (ROLE_CODE);
ALTER TABLE DIC_DATUM ADD CONSTRAINT DIC_DATUM_PK PRIMARY KEY (DATUM_CODE);
ALTER TABLE DIC_DEFINED_LINE_TYPE ADD CONSTRAINT DIC_DEFINED_LINE_TYPE_PK PRIMARY KEY (TYPE_CODE);
ALTER TABLE DIC_DOWNLOAD_TYPE ADD CONSTRAINT DIC_DOWNLOAD_TYPE_PK PRIMARY KEY (TYPE);
ALTER TABLE DIC_EPSG ADD CONSTRAINT DIC_EPSG_PK PRIMARY KEY (EPSG_CODE);
ALTER TABLE DIC_FEATURE_CLASS ADD CONSTRAINT DIC_FEATURE_CLASS_PK PRIMARY KEY (FEATURE_CLASS_CODE);
ALTER TABLE DIC_LICENCE_TYPE ADD CONSTRAINT DIC_LICENCE_TYPE_PK PRIMARY KEY (TYPE_CODE);
ALTER TABLE DIC_MEDIUM ADD CONSTRAINT DIC_MEDIUM_PK PRIMARY KEY (MEDIUM_CODE);
ALTER TABLE DIC_PRODUCT_CLASS ADD CONSTRAINT DIC_PRODUCT_CLASS_PK PRIMARY KEY (CLASS_CODE);
ALTER TABLE DIC_PRODUCT_TYPE ADD CONSTRAINT DIC_PRODUCT_TYPE_PK PRIMARY KEY (TYPE_CODE);
ALTER TABLE DIC_PROJECTION ADD CONSTRAINT DIC_PROJECTION_PK PRIMARY KEY (PROJECTION_CODE);
ALTER TABLE DIC_SEAFISH_FEATURE ADD CONSTRAINT DIC_SEAFISH_FEATURE_PK PRIMARY KEY (FEATURE_CODE);
ALTER TABLE DIC_SPHEROID ADD CONSTRAINT DIC_SPHEROID_PK PRIMARY KEY (SPHEROID_CODE);
ALTER TABLE DIC_SURVEY_ENVIRONMENT ADD CONSTRAINT DIC_SURVEY_ENVIRONMENT_PK PRIMARY KEY
      (ENVIRONMENT_CODE);
ALTER TABLE DIC_SURVEY_TYPE ADD CONSTRAINT DIC_SURVEY_TYPE_PK PRIMARY KEY (TYPE_CODE);
ALTER TABLE DIC_VISIBILITY ADD CONSTRAINT DIC_VISIBILITY_PK PRIMARY KEY (VISIBILITY_CODE);
ALTER TABLE DIC_WELL_DATUM_TYPE ADD CONSTRAINT DIC_WELL_DATUM_TYPE_PK PRIMARY KEY (TYPE_CODE);
ALTER TABLE DIC_WELL_INTENT ADD CONSTRAINT DIC_WELL_INTENT_PK PRIMARY KEY (INTENT_CODE);
ALTER TABLE DIC_WELL_TYPE ADD CONSTRAINT DIC_WELL_TYPE_PK PRIMARY KEY (TYPE_CODE);
ALTER TABLE DTI_WELL ADD CONSTRAINT DTI_WELL_PK PRIMARY KEY (DTI_WELLID);
ALTER TABLE DTI_WELL ADD CONSTRAINT DTI_WELL_UK UNIQUE (DTI_WELL_REG_NO);
ALTER TABLE DTI_WELL ADD CONSTRAINT DTI_WELL_UK2 UNIQUE (DTI_WELL_REG_UPPER);
ALTER TABLE FEATUREPRODUCT ADD CONSTRAINT FEATUREPRODUCT_PK PRIMARY KEY (FEATUREPRODUCTID);
ALTER TABLE FEATUREPRODUCT ADD CONSTRAINT FEATUREPRODUCT_UK UNIQUE
      (WELLID,THREEDSEIS_SURVEYID,TWODSEIS_SURVEYID,TWODSEIS_LINEID,PROVIDER_COMPID,PRODUCT_TYPE,PRODUCT
      _MEDIUM);
ALTER TABLE HCFIELD_BOUNDING_PT ADD CONSTRAINT HCFIELD_BOUNDING_PT_PK PRIMARY KEY (HCFIELDID,SEQ_NO);
ALTER TABLE HYDROCARBONS_FIELD ADD CONSTRAINT HYDROCARBONS_FIELD_PK PRIMARY KEY (HCFIELDID);
ALTER TABLE HYDROCARBONS_FIELD ADD CONSTRAINT HYDROCARBONS_FIELD_UK UNIQUE (FIELDNAME);
ALTER TABLE LICENCE ADD CONSTRAINT LICENCE_PK PRIMARY KEY (LICENCEID);
ALTER TABLE LICENCE ADD CONSTRAINT LICENCE_UK UNIQUE (COUNTRY,LICENCE_DTINO);
ALTER TABLE LICENCE_BOUNDING_LINE ADD CONSTRAINT LICENCE_BND_LINE_PK PRIMARY KEY
      (LICENCEID,LINE_SEQUENCE);
ALTER TABLE COMPANYHASLICENCE ADD CONSTRAINT COMPANYHASLICENCE_PK PRIMARY KEY (OWNER_COMPID,LICENCEID);
ALTER TABLE MEDIAN_LINE ADD CONSTRAINT MEDIAN_LINE_PK PRIMARY KEY (MEDIAN_LINEID);
ALTER TABLE PIPELINE ADD CONSTRAINT PIPELINE_PK PRIMARY KEY (PIPELINEID);
ALTER TABLE PIPELINE_PT ADD CONSTRAINT PIPELINE_PT_PK PRIMARY KEY (PIPELINEID,SEQ_NO);
ALTER TABLE PIPELINE_SPAN ADD CONSTRAINT PIPELINE_SPAN_PK PRIMARY KEY (PIPELINE_SPANID);
ALTER TABLE PIPELINE_SPAN_PT ADD CONSTRAINT PIPELINE_SPAN_PT_PK PRIMARY KEY (PIPELINE_SPANID,SEQ_NO);
ALTER TABLE PRODTYPE_CLSN ADD CONSTRAINT PRODTYPE_CLSN_PK PRIMARY KEY
      (CLASSIFIER_COMPID,PRODUCT_TYPE,PRODUCT_CLASS);
ALTER TABLE PRODUCT_ENTITLEMENT ADD CONSTRAINT PRODUCT_ENTITLEMENT_PK PRIMARY KEY
      (FEATUREPRODUCTID,SUBSCRIBER_COMPID);
ALTER TABLE PUBLIC_CONTACT ADD CONSTRAINT PUBLIC_CONTACT_PK PRIMARY KEY (LOGIN);
ALTER TABLE QUAD ADD CONSTRAINT QUAD_PK PRIMARY KEY (QUADID);
ALTER TABLE QUAD ADD CONSTRAINT QUAD_UK UNIQUE (COUNTRY,QUADNO);
ALTER TABLE SAFETY_EX_ZONE ADD CONSTRAINT SAFETY_EX_ZONE_PK PRIMARY KEY (SAFETY_EX_ZONEID);
ALTER TABLE SAFETY_EX_ZONE ADD CONSTRAINT SAFETY_EX_ZONE_UK UNIQUE (SAFETY_EX_ZONE_LABEL);
ALTER TABLE SAFETY_EX_ZONE_CENTRE_PT ADD CONSTRAINT SAFETY_EX_ZONE_CENTRE_PT_PK PRIMARY KEY
      (SAFETY_EX_ZONEID,SEQ_NO);
ALTER TABLE SUBSEA_INF ADD CONSTRAINT SUBSEA_INF_PK PRIMARY KEY (SUBSEA_INFID);
ALTER TABLE SUBSEA_INF ADD CONSTRAINT SUBSEA_INF_UK UNIQUE (SUBSEA_INF_LABEL);
ALTER TABLE SEASURFACE_INF ADD CONSTRAINT SEASURFACE_INF_PK PRIMARY KEY (SEASURFACE_INFID);
ALTER TABLE SEASURFACE_INF ADD CONSTRAINT SEASURFACE_INF_UK UNIQUE (SEASURFACE_INF_LABEL);
ALTER TABLE THREEDSEIS_ALIAS ADD CONSTRAINT THREEDSEIS_ALIAS_PK PRIMARY KEY
      (THREEDSEIS_SURVEYID,ALIAS_COMPANY,THREEDSEIS_ALIAS_NAME);
ALTER TABLE THREEDSEIS_BOUNDING_PT ADD CONSTRAINT THREEDSEIS_BND_PT_PK PRIMARY KEY
      (THREEDSEIS_SURVEYID,SEQ_NO);
ALTER TABLE THREEDSEIS_SURVEY ADD CONSTRAINT THREEDSEIS_SURVEY_PK PRIMARY KEY (THREEDSEIS_SURVEYID);
ALTER TABLE THREEDSEIS_SURVEY ADD CONSTRAINT THREEDSEIS_SURVEY_UK UNIQUE (THREEDSEIS_CS9NAME);
ALTER TABLE TWODSEIS_ALIAS ADD CONSTRAINT TWODSEIS_ALIAS_PK PRIMARY KEY
      (TWODSEIS_SURVEYID,ALIAS_COMPANY,TWODSEIS_ALIAS_NAME);
```

```
ALTER TABLE TWODSEIS_BOUNDING_PT ADD CONSTRAINT TWODSEIS_BND_PT_PK PRIMARY KEY
      (TWODSEIS_SURVEYID,SEQ_NO);
ALTER TABLE TWODSEIS_LINE ADD CONSTRAINT TWODSEIS_LINE_PK PRIMARY KEY (TWODSEIS_LINEID);
ALTER TABLE TWODSEIS_LINE ADD CONSTRAINT TWODSEIS_LINE_UK UNIQUE (TWODSEIS_SURVEYID,TWODLINE_CS9NAME);
ALTER TABLE TWODSEIS_LINE_PT ADD CONSTRAINT TWODSEIS_LINE_PT_PK PRIMARY KEY (TWODSEIS_LINEID,SEQ_NO);
ALTER TABLE TWODSEIS_SURVEY ADD CONSTRAINT TWODSEIS_SURVEY_PK PRIMARY KEY (TWODSEIS_SURVEYID);
ALTER TABLE TWODSEIS_SURVEY ADD CONSTRAINT TWODSEIS_SURVEY_UK UNIQUE (TWODSEIS_CS9NAME);
ALTER TABLE USER_INTEREST_BOX ADD CONSTRAINT USER_INTEREST_BOX_PK PRIMARY KEY (LOGIN,BOX_NO);
ALTER TABLE USER_INTEREST_PRODUCT ADD CONSTRAINT USER_INTEREST_PRODUCT_PK PRIMARY KEY
      (LOGIN,BOX_NO,PRODUCT_TYPE);
ALTER TABLE USER_PRODUCT_PICK ADD CONSTRAINT USER_PRODUCT_PICK_PK PRIMARY KEY (LOGIN,FEATUREPRODUCTID);
ALTER TABLE VALID_FEATUREPRODUCT ADD CONSTRAINT VALID_FEATUREPRODUCT_PK PRIMARY KEY
      (FEATURE_CLASS_CODE,PRODUCT_TYPE);
ALTER TABLE WELL ADD CONSTRAINT WELL_PK PRIMARY KEY (WELLID);
ALTER TABLE WELL ADD CONSTRAINT WELL_UK UNIQUE (DTI_WELLID,WELLNAME);
ALTER TABLE WELL_ALIAS ADD CONSTRAINT WELL_ALIAS_PK PRIMARY KEY (WELLID,ALIAS_COMPANY,WELL_ALIAS_NAME);
```

FOREIGN KEY CONSTRAINTS

```
ALTER TABLE AOI ADD CONSTRAINT AOI_DICDATUM_FK FOREIGN KEY (DATUM_NAME) REFERENCES
      DIC_DATUM(DATUM_CODE);
ALTER TABLE AOI ADD CONSTRAINT AOI_DICEPSG_FK FOREIGN KEY (EPSG_CODE) REFERENCES DIC_EPSG(EPSG_CODE);
ALTER TABLE AOI ADD CONSTRAINT AOI_DICPROJ_FK FOREIGN KEY (PROJECTION_NAME) REFERENCES
      DIC_PROJECTION(PROJECTION_CODE);
ALTER TABLE AOI ADD CONSTRAINT AOI_DICSPHEROID_FK FOREIGN KEY (SPHEROID_NAME) REFERENCES
      DIC_SPHEROID(SPHEROID_CODE);
ALTER TABLE AOI ADD CONSTRAINT AOI_VISIBILITY_FK FOREIGN KEY (VISIBILITY) REFERENCES
      DIC_VISIBILITY(VISIBILITY_CODE);
ALTER TABLE BATHYMETRY ADD CONSTRAINT BATHYMETRY_DATASOURCEFK FOREIGN KEY (DATA_SOURCE) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE BATHYMETRY ADD CONSTRAINT BATHYMETRY_DICDATUM_FK FOREIGN KEY (DATUM_NAME) REFERENCES
      DIC_DATUM(DATUM_CODE);
ALTER TABLE BATHYMETRY ADD CONSTRAINT BATHYMETRY_DICEPSG_FK FOREIGN KEY (EPSG_CODE) REFERENCES
      DIC_EPSG(EPSG_CODE);
ALTER TABLE BATHYMETRY ADD CONSTRAINT BATHYMETRY_DICPROJ_FK FOREIGN KEY (PROJECTION_NAME) REFERENCES
      DIC_PROJECTION(PROJECTION_CODE);
ALTER TABLE BATHYMETRY ADD CONSTRAINT BATHYMETRY_DICSPHEROID_FK FOREIGN KEY (SPHEROID_NAME) REFERENCES
      DIC_SPHEROID(SPHEROID_CODE);
ALTER TABLE BATHYMETRY ADD CONSTRAINT BATHYMETRY_VISIBILITY_FK FOREIGN KEY (VISIBILITY) REFERENCES
      DIC_VISIBILITY(VISIBILITY_CODE);
ALTER TABLE BLOCK ADD CONSTRAINT BLOCK_DATASOURCEFK FOREIGN KEY (DATA_SOURCE) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE BLOCK ADD CONSTRAINT BLOCK_QUAD_FK FOREIGN KEY (COUNTRY,QUADNO) REFERENCES
      QUAD(COUNTRY,QUADNO);
ALTER TABLE BLOCK ADD CONSTRAINT BLOCK_VISIBILITY_FK FOREIGN KEY (VISIBILITY) REFERENCES
      DIC_VISIBILITY(VISIBILITY_CODE);
ALTER TABLE COAST ADD CONSTRAINT COAST_DATASOURCEFK FOREIGN KEY (DATA_SOURCE) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE COAST ADD CONSTRAINT COAST_DICDATUM_FK FOREIGN KEY (DATUM_NAME) REFERENCES
      DIC_DATUM(DATUM_CODE);
ALTER TABLE COAST ADD CONSTRAINT COAST_DICEPSG_FK FOREIGN KEY (EPSG_CODE) REFERENCES
      DIC_EPSG(EPSG_CODE);
ALTER TABLE COAST ADD CONSTRAINT COAST_DICPROJ_FK FOREIGN KEY (PROJECTION_NAME) REFERENCES
      DIC_PROJECTION(PROJECTION_CODE);
ALTER TABLE COAST ADD CONSTRAINT COAST_DICSPHEROID_FK FOREIGN KEY (SPHEROID_NAME) REFERENCES
      DIC_SPHEROID(SPHEROID_CODE);
ALTER TABLE COAST ADD CONSTRAINT COAST_VISIBILITY_FK FOREIGN KEY (VISIBILITY) REFERENCES
      DIC_VISIBILITY(VISIBILITY_CODE);
ALTER TABLE COMPANY ADD CONSTRAINT COMPANY_COMPTYPE_FK FOREIGN KEY (COMPANY_TYPE) REFERENCES
      DIC_COMPANY_TYPE(TYPE_CODE);
ALTER TABLE COMPANY ADD CONSTRAINT COMPANY_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE COMPANY_CONTACT ADD CONSTRAINT COMPANY_CONTACT_COMP_FK FOREIGN KEY (COMPANYID) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE COMPANYHASLICENCE ADD CONSTRAINT COMPANYHASLICENCE_COMP_FK FOREIGN KEY (OWNER_COMPID)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE COMPANYHASLICENCE ADD CONSTRAINT COMPANYHASLICENCE_LICEN_FK FOREIGN KEY (LICENCEID)
      REFERENCES LICENCE(LICENCEID);
ALTER TABLE CONTACTHASROLE ADD CONSTRAINT CONTACTHASROLE_CONT_FK FOREIGN KEY (COMPANYID,CONTACTREF)
      REFERENCES COMPANY_CONTACT(COMPANYID,CONTACTREF);
ALTER TABLE CONTACTHASROLE ADD CONSTRAINT CONTACTHASROLE_ROLE_FK FOREIGN KEY (CONTACTROLE) REFERENCES
      DIC_CONTACT_ROLE(ROLE_CODE);
ALTER TABLE COUNTRY ADD CONSTRAINT COUNTRY_DATASOURCEFK FOREIGN KEY (DATA_SOURCE) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE DEAL_USER ADD CONSTRAINT DEAL_USER_COMP_FK FOREIGN KEY (COMPANYID) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE DEAL_USER ADD CONSTRAINT DEAL_USER_COMPCONT_FK FOREIGN KEY (COMPANYID,CONTACTREF) REFERENCES
      COMPANY_CONTACT(COMPANYID,CONTACTREF);
ALTER TABLE DIC_COMPANY_TYPE ADD CONSTRAINT DIC_COMPANY_TYPE_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE DIC_CONTACT_ROLE ADD CONSTRAINT DIC_CONTACT_ROLE_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE DIC_DEFINED_LINE_TYPE ADD CONSTRAINT DIC_DEFLINE_TYPE_DATASOURCE_FK FOREIGN KEY
      (DATA_SOURCE) REFERENCES COMPANY(COMPANYID);
ALTER TABLE DIC_EPSG ADD CONSTRAINT DIC_EPSG_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE DIC_EPSG ADD CONSTRAINT DIC_EPSG_DATUM_FK FOREIGN KEY (DATUM_NAME) REFERENCES
      DIC_DATUM(DATUM_CODE);
```

```
ALTER TABLE DIC_EPSG ADD CONSTRAINT DIC_EPSG_SPHEROID_FK FOREIGN KEY (SPHEROID_NAME) REFERENCES
      DIC_SPHEROID(SPHEROID_CODE);
ALTER TABLE DIC_EPSG ADD CONSTRAINT DIC_EPSG_PROJ_FK FOREIGN KEY (PROJECTION_NAME) REFERENCES
      DIC_PROJECTION(PROJECTION_CODE);
ALTER TABLE DIC_LICENCE_TYPE ADD CONSTRAINT DIC_LICENCE_TYPE_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE DIC_MEDIUM ADD CONSTRAINT DIC_MEDIUM_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE DIC_PRODUCT_CLASS ADD CONSTRAINT DIC_PRODCLS_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE DIC_PRODUCT_TYPE ADD CONSTRAINT DIC_PRODUCT_TYPE_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE DIC_PRODUCT_TYPE ADD CONSTRAINT DIC_PRODUCT_TYPE_SUPERTYPE_FK FOREIGN KEY (SUPER_TYPE_CODE)
      REFERENCES DIC_PRODUCT_TYPE(TYPE_CODE);
ALTER TABLE DIC_PROJECTION ADD CONSTRAINT DIC_PROJECTION_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE DIC_SEAFISH_FEATURE ADD CONSTRAINT DIC_SEAFISH_FT_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE DIC_SPHEROID ADD CONSTRAINT DIC_SPHEROID_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE DIC_SURVEY_ENVIRONMENT ADD CONSTRAINT DIC_SURVEY_ENV_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE DIC_SURVEY_TYPE ADD CONSTRAINT DIC_SURVEY_TYPE_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE DIC_VISIBILITY ADD CONSTRAINT DIC_VISIBILITY_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE DIC_WELL_DATUM_TYPE ADD CONSTRAINT DIC_WELLDATUM_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE DIC_WELL_INTENT ADD CONSTRAINT DIC_WELL_INTENT_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE DIC_WELL_TYPE ADD CONSTRAINT DIC_WELL_TYPE_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE DTI_WELL ADD CONSTRAINT DTI_WELL_COUNTRY_FK FOREIGN KEY (COUNTRY) REFERENCES
      COUNTRY(GMICODE);
ALTER TABLE DTI_WELL ADD CONSTRAINT DTI_WELL_DATASOURCEFK FOREIGN KEY (DATA_SOURCE) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE DTI_WELL ADD CONSTRAINT DTI_WELL_DICDATUM_FK FOREIGN KEY (DATUM_NAME) REFERENCES
      DIC_DATUM(DATUM_CODE);
ALTER TABLE DTI_WELL ADD CONSTRAINT DTI_WELL_DICEPSG_FK FOREIGN KEY (EPSG_CODE) REFERENCES
      DIC_EPSG(EPSG_CODE);
ALTER TABLE DTI_WELL ADD CONSTRAINT DTI_WELL_DICPROJ_FK FOREIGN KEY (PROJECTION_NAME) REFERENCES
      DIC_PROJECTION(PROJECTION_CODE);
ALTER TABLE DTI_WELL ADD CONSTRAINT DTI_WELL_DICSPHEROID_FK FOREIGN KEY (SPHEROID_NAME) REFERENCES
      DIC_SPHEROID(SPHEROID_CODE);
ALTER TABLE DTI_WELL ADD CONSTRAINT DTI_WELL_VISIBILITY_FK FOREIGN KEY (VISIBILITY) REFERENCES
      DIC_VISIBILITY(VISIBILITY_CODE);
ALTER TABLE FEATUREPRODUCT ADD CONSTRAINT FEATUREPRODUCT_MEDIUM_FK FOREIGN KEY (PRODUCT_MEDIUM)
      REFERENCES DIC_MEDIUM(MEDIUM_CODE);
ALTER TABLE FEATUREPRODUCT ADD CONSTRAINT FEATUREPRODUCT_PROV_FK FOREIGN KEY (PROVIDER_COMPID)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE FEATUREPRODUCT ADD CONSTRAINT FEATUREPRODUCT_THREEDSEIS_FK FOREIGN KEY (THREEDSEIS_SURVEYID)
      REFERENCES THREEDSEIS_SURVEY(THREEDSEIS_SURVEYID);
ALTER TABLE FEATUREPRODUCT ADD CONSTRAINT FEATUREPRODUCT_TWODLINE_FK FOREIGN KEY (TWODSEIS_LINEID)
      REFERENCES TWODSEIS_LINE(TWODSEIS_LINEID);
ALTER TABLE FEATUREPRODUCT ADD CONSTRAINT FEATUREPRODUCT_TWODSEIS_FK FOREIGN KEY (TWODSEIS_SURVEYID)
      REFERENCES TWODSEIS_SURVEY(TWODSEIS_SURVEYID);
ALTER TABLE FEATUREPRODUCT ADD CONSTRAINT FEATUREPRODUCT_TYPE_FK FOREIGN KEY (PRODUCT_TYPE) REFERENCES
      DIC_PRODUCT_TYPE(TYPE_CODE);
ALTER TABLE FEATUREPRODUCT ADD CONSTRAINT FEATUREPRODUCT_VISIBILITY_FK FOREIGN KEY (VISIBILITY)
      REFERENCES DIC_VISIBILITY(VISIBILITY_CODE);
ALTER TABLE FEATUREPRODUCT ADD CONSTRAINT FEATUREPRODUCT_WELL_FK FOREIGN KEY (WELLID) REFERENCES
      WELL(WELLID);
ALTER TABLE HCFIELD_BOUNDING_PT ADD CONSTRAINT HCFIELD_BND_PT_FIELD_FK FOREIGN KEY (HCFIELDID)
      REFERENCES HYDROCARBONS_FIELD(HCFIELDID);
ALTER TABLE HYDROCARBONS_FIELD ADD CONSTRAINT HCFIELD_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE HYDROCARBONS_FIELD ADD CONSTRAINT HCFIELD_DICDATUM_FK FOREIGN KEY (DATUM_NAME) REFERENCES
      DIC_DATUM(DATUM_CODE);
ALTER TABLE HYDROCARBONS_FIELD ADD CONSTRAINT HCFIELD_DICEPSG_FK FOREIGN KEY (EPSG_CODE) REFERENCES
      DIC_EPSG(EPSG_CODE);
ALTER TABLE HYDROCARBONS_FIELD ADD CONSTRAINT HCFIELD_DICPROJ_FK FOREIGN KEY (PROJECTION_NAME)
      REFERENCES DIC_PROJECTION(PROJECTION_CODE);
ALTER TABLE HYDROCARBONS_FIELD ADD CONSTRAINT HCFIELD_DICSPHEROID_FK FOREIGN KEY (SPHEROID_NAME)
      REFERENCES DIC_SPHEROID(SPHEROID_CODE);
ALTER TABLE HYDROCARBONS_FIELD ADD CONSTRAINT HCFIELD_VISIBILITY_FK FOREIGN KEY (VISIBILITY) REFERENCES
      DIC_VISIBILITY(VISIBILITY_CODE);
ALTER TABLE LICENCE ADD CONSTRAINT LICENCE_COUNTRY_FK FOREIGN KEY (COUNTRY) REFERENCES COUNTRY(GMICODE);
ALTER TABLE LICENCE ADD CONSTRAINT LICENCE_DATASOURCEFK FOREIGN KEY (DATA_SOURCE) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE licence ADD CONSTRAINT licence_DICDATUM_FK FOREIGN KEY (DATUM_NAME) REFERENCES
      DIC_DATUM(DATUM_CODE);
ALTER TABLE licence ADD CONSTRAINT licence_DICEPSG_FK FOREIGN KEY (EPSG_CODE) REFERENCES
      DIC_EPSG(EPSG_CODE);
ALTER TABLE licence ADD CONSTRAINT licence_DICPROJ_FK FOREIGN KEY (PROJECTION_NAME) REFERENCES
      DIC_PROJECTION(PROJECTION_CODE);
ALTER TABLE licence ADD CONSTRAINT licence_DICSPHEROID_FK FOREIGN KEY (SPHEROID_NAME) REFERENCES
      DIC_SPHEROID(SPHEROID_CODE);
ALTER TABLE LICENCE ADD CONSTRAINT LICENCE_LICTYPE_FK FOREIGN KEY (LICENCE_TYPE) REFERENCES
      DIC_LICENCE_TYPE(TYPE_CODE);
ALTER TABLE LICENCE ADD CONSTRAINT LICENCE_VISIBILITY_FK FOREIGN KEY (VISIBILITY) REFERENCES
      DIC_VISIBILITY(VISIBILITY_CODE);
```

```
ALTER TABLE LICENCE_BOUNDING_LINE ADD CONSTRAINT LICENCE_BND_LINE_TYPE_FK FOREIGN KEY (LINE_TYPE)
      REFERENCES DIC_DEFINED_LINE_TYPE(TYPE_CODE);
ALTER TABLE LICENCE_BOUNDING_LINE ADD CONSTRAINT LICENCE_BNDLINE_LICID_FK FOREIGN KEY (LICENCEID)
      REFERENCES LICENCE(LICENCEID);
alter table LOG_DOWNLOAD add constraint LOG_DOWNLOAD_TYPE_FK foreign key (TYPE) references
      DIC_DOWNLOAD_TYPE(TYPE);
ALTER TABLE MEDIAN_LINE ADD CONSTRAINT MEDIAN_LINE_DATASOURCEFK FOREIGN KEY (DATA_SOURCE) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE MEDIAN_LINE ADD CONSTRAINT MEDIAN_LINE_DICDATUM_FK FOREIGN KEY (DATUM_NAME) REFERENCES
      DIC_DATUM(DATUM_CODE);
ALTER TABLE MEDIAN_LINE ADD CONSTRAINT MEDIAN_LINE_DICEPSG_FK FOREIGN KEY (EPSG_CODE) REFERENCES
      DIC_EPSG(EPSG_CODE);
ALTER TABLE MEDIAN_LINE ADD CONSTRAINT MEDIAN_LINE_DICPROJ_FK FOREIGN KEY (PROJECTION_NAME) REFERENCES
      DIC_PROJECTION(PROJECTION_CODE);
ALTER TABLE MEDIAN_LINE ADD CONSTRAINT MEDIAN_LINE_DICSPHEROID_FK FOREIGN KEY (SPHEROID_NAME) REFERENCES
      DIC_SPHEROID(SPHEROID_CODE);
ALTER TABLE MEDIAN_LINE ADD CONSTRAINT MEDIAN_LINE_VISIBILITY_FK FOREIGN KEY (VISIBILITY) REFERENCES
      DIC_VISIBILITY(VISIBILITY_CODE);
ALTER TABLE pipeline ADD CONSTRAINT PIPELINE_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE pipeline ADD CONSTRAINT PIPELINE_DICDATUM_FK FOREIGN KEY (DATUM_NAME) REFERENCES
      DIC_DATUM(DATUM_CODE);
ALTER TABLE pipeline ADD CONSTRAINT PIPELINE_DICEPSG_FK FOREIGN KEY (EPSG_CODE) REFERENCES
      DIC_EPSG(EPSG_CODE);
ALTER TABLE pipeline ADD CONSTRAINT PIPELINE_DICPROJ_FK FOREIGN KEY (PROJECTION_NAME) REFERENCES
      DIC_PROJECTION(PROJECTION_CODE);
ALTER TABLE pipeline ADD CONSTRAINT PIPELINE_DICSPHEROID_FK FOREIGN KEY (SPHEROID_NAME) REFERENCES
      DIC_SPHEROID(SPHEROID_CODE);
ALTER TABLE pipeline ADD CONSTRAINT PIPELINE_VISIBILITY_FK FOREIGN KEY (VISIBILITY) REFERENCES
      DIC_VISIBILITY(VISIBILITY_CODE);
ALTER TABLE PIPELINE_PT ADD CONSTRAINT PIPELINE_PT_PIPE_FK FOREIGN KEY (PIPELINEID) REFERENCES
      pipeline(PIPELINEID);
ALTER TABLE PIPELINE_SPAN ADD CONSTRAINT PIPELINE_SPAN_DATASOURCEFK FOREIGN KEY (DATA_SOURCE) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE PIPELINE_SPAN ADD CONSTRAINT PIPELINE_SPAN_PIPE_FK FOREIGN KEY (PIPELINEID) REFERENCES
      pipeline(PIPELINEID);
ALTER TABLE PIPELINE_SPAN ADD CONSTRAINT PIPELINE_SPAN_VISIBILITY_FK FOREIGN KEY (VISIBILITY) REFERENCES
      DIC_VISIBILITY(VISIBILITY_CODE);
ALTER TABLE PRODTYPE_CLSN ADD CONSTRAINT PRODTYPE_CLSN_CLASS_FK FOREIGN KEY (PRODUCT_CLASS) REFERENCES
      DIC_PRODUCT_CLASS(CLASS_CODE);
ALTER TABLE PRODTYPE_CLSN ADD CONSTRAINT PRODTYPE_CLSN_COMP_FK FOREIGN KEY (CLASSIFIER_COMPID)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE PRODTYPE_CLSN ADD CONSTRAINT PRODTYPE_CLSN_TYPE_FK FOREIGN KEY (PRODUCT_TYPE) REFERENCES
      DIC_PRODUCT_TYPE(TYPE_CODE);
ALTER TABLE PRODUCT_ENTITLEMENT ADD CONSTRAINT PRODUCT_ENT_PRODUCT_FK FOREIGN KEY (FEATUREPRODUCTID)
      REFERENCES FEATUREPRODUCT(FEATUREPRODUCTID);
ALTER TABLE PRODUCT_ENTITLEMENT ADD CONSTRAINT PRODUCT_ENT_SUBSC_FK FOREIGN KEY (SUBSCRIBER_COMPID)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE PUBLIC_CONTACT ADD CONSTRAINT PUBLIC_CONTACT_COMPTYPE_FK FOREIGN KEY (COMPANY_TYPE)
      REFERENCES DIC_COMPANY_TYPE(TYPE_CODE);
ALTER TABLE QUAD ADD CONSTRAINT QUAD_COUNTRY_FK FOREIGN KEY (COUNTRY) REFERENCES COUNTRY(GMICODE);
ALTER TABLE QUAD ADD CONSTRAINT QUAD_DATASOURCEFK FOREIGN KEY (DATA_SOURCE) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE QUAD ADD CONSTRAINT QUAD_VISIBILITY_FK FOREIGN KEY (VISIBILITY) REFERENCES
      DIC_VISIBILITY(VISIBILITY_CODE);
ALTER TABLE SAFETY_EX_ZONE ADD CONSTRAINT SAFETY_EX_ZONE_DATASOURCEFK FOREIGN KEY (DATA_SOURCE)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE SAFETY_EX_ZONE ADD CONSTRAINT SAFETY_EX_ZONE_DICDATUM_FK FOREIGN KEY (DATUM_NAME) REFERENCES
      DIC_DATUM(DATUM_CODE);
ALTER TABLE SAFETY_EX_ZONE ADD CONSTRAINT SAFETY_EX_ZONE_DICEPSG_FK FOREIGN KEY (EPSG_CODE) REFERENCES
      DIC_EPSG(EPSG_CODE);
ALTER TABLE SAFETY_EX_ZONE ADD CONSTRAINT SAFETY_EX_ZONE_DICPROJ_FK FOREIGN KEY (PROJECTION_NAME)
      REFERENCES DIC_PROJECTION(PROJECTION_CODE);
ALTER TABLE SAFETY_EX_ZONE ADD CONSTRAINT SAFETY_EX_ZONE_DICSPHEROID_FK FOREIGN KEY (SPHEROID_NAME)
      REFERENCES DIC_SPHEROID(SPHEROID_CODE);
ALTER TABLE SAFETY_EX_ZONE ADD CONSTRAINT SAFETY_EX_ZONE_VISIBILITY_FK FOREIGN KEY (VISIBILITY)
      REFERENCES DIC_VISIBILITY(VISIBILITY_CODE);
ALTER TABLE SAFETY_EX_ZONE_CENTRE_PT ADD CONSTRAINT SAFETY_EX_ZONE_CNRTPT_ZONE_FK FOREIGN KEY
      (SAFETY_EX_ZONEID) REFERENCES SAFETY_EX_ZONE(SAFETY_EX_ZONEID);
ALTER TABLE SEASURFACE_INF ADD CONSTRAINT SEASURFACE_INF_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE SEASURFACE_INF ADD CONSTRAINT SEASURFACE_INF_DICDATUM_FK FOREIGN KEY (DATUM_NAME) REFERENCES
      DIC_DATUM(DATUM_CODE);
ALTER TABLE SEASURFACE_INF ADD CONSTRAINT SEASURFACE_INF_DICEPSG_FK FOREIGN KEY (EPSG_CODE) REFERENCES
      DIC_EPSG(EPSG_CODE);
ALTER TABLE SEASURFACE_INF ADD CONSTRAINT SEASURFACE_INF_DICPROJ_FK FOREIGN KEY (PROJECTION_NAME)
      REFERENCES DIC_PROJECTION(PROJECTION_CODE);
ALTER TABLE SEASURFACE_INF ADD CONSTRAINT SEASURFACE_INF_DICSPHEROID_FK FOREIGN KEY (SPHEROID_NAME)
      REFERENCES DIC_SPHEROID(SPHEROID_CODE);
ALTER TABLE SEASURFACE_INF ADD CONSTRAINT SEASURFACE_INF_VISIBILITY_FK FOREIGN KEY (VISIBILITY)
      REFERENCES DIC_VISIBILITY(VISIBILITY_CODE);
ALTER TABLE SUBSEA_INF ADD CONSTRAINT SUBSEA_INF_DATASOURCE_FK FOREIGN KEY (DATA_SOURCE) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE SUBSEA_INF ADD CONSTRAINT SUBSEA_INF_DICDATUM_FK FOREIGN KEY (DATUM_NAME) REFERENCES
      DIC_DATUM(DATUM_CODE);
ALTER TABLE SUBSEA_INF ADD CONSTRAINT SUBSEA_INF_DICEPSG_FK FOREIGN KEY (EPSG_CODE) REFERENCES
      DIC_EPSG(EPSG_CODE);
ALTER TABLE SUBSEA_INF ADD CONSTRAINT SUBSEA_INF_DICPROJ_FK FOREIGN KEY (PROJECTION_NAME) REFERENCES
      DIC_PROJECTION(PROJECTION_CODE);
ALTER TABLE SUBSEA_INF ADD CONSTRAINT SUBSEA_INF_DICSPHEROID_FK FOREIGN KEY (SPHEROID_NAME) REFERENCES
      DIC_SPHEROID(SPHEROID_CODE);
```

```
ALTER TABLE SUBSEA_INF ADD CONSTRAINT SUBSEA_INF_VISIBILITY_FK FOREIGN KEY (VISIBILITY) REFERENCES
      DIC_VISIBILITY(VISIBILITY_CODE);
ALTER TABLE THREEDSEIS_ALIAS ADD CONSTRAINT THREEDALIAS_CO_FK FOREIGN KEY (ALIAS_COMPANY) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE THREEDSEIS_ALIAS ADD CONSTRAINT THREEDALIAS_SURVEY_FK FOREIGN KEY (THREEDSEIS_SURVEYID)
      REFERENCES THREEDSEIS_SURVEY(THREEDSEIS_SURVEYID);
ALTER TABLE THREEDSEIS_BOUNDING_PT ADD CONSTRAINT THREEDSEIS_BND_PT_SURVEY_FK FOREIGN KEY
      (THREEDSEIS_SURVEYID) REFERENCES THREEDSEIS_SURVEY(THREEDSEIS_SURVEYID);
ALTER TABLE THREEDSEIS_SURVEY ADD CONSTRAINT THREEDSEIS_COUNTRYFK FOREIGN KEY (COUNTRY) REFERENCES
      COUNTRY(GMICODE);
ALTER TABLE THREEDSEIS_SURVEY ADD CONSTRAINT THREEDSEIS_DATAORIGFK FOREIGN KEY (DATA_ORIGIN) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE THREEDSEIS_SURVEY ADD CONSTRAINT THREEDSEIS_DATAPROVFK FOREIGN KEY (DATA_PROVIDER)
      REFERENCES COMPANY(COMPANYID);
ALTER TABLE THREEDSEIS_SURVEY ADD CONSTRAINT THREEDSEIS_DICDATUM_FK FOREIGN KEY (DATUM_NAME) REFERENCES
      DIC_DATUM(DATUM_CODE);
ALTER TABLE THREEDSEIS_SURVEY ADD CONSTRAINT THREEDSEIS_DICENVFK FOREIGN KEY (SURVEY_ENVIRONMENT)
      REFERENCES DIC_SURVEY_ENVIRONMENT(ENVIRONMENT_CODE);
ALTER TABLE THREEDSEIS_SURVEY ADD CONSTRAINT THREEDSEIS_DICEPSG_FK FOREIGN KEY (EPSG_CODE) REFERENCES
      DIC_EPSG(EPSG_CODE);
ALTER TABLE THREEDSEIS_SURVEY ADD CONSTRAINT THREEDSEIS_DICPROJ_FK FOREIGN KEY (PROJECTION_NAME)
      REFERENCES DIC_PROJECTION(PROJECTION_CODE);
ALTER TABLE THREEDSEIS_SURVEY ADD CONSTRAINT THREEDSEIS_DICSPHEROID_FK FOREIGN KEY (SPHEROID_NAME)
      REFERENCES DIC_SPHEROID(SPHEROID_CODE);
ALTER TABLE THREEDSEIS_SURVEY ADD CONSTRAINT THREEDSEIS_DICTYPEFK FOREIGN KEY (SURVEY_TYPE) REFERENCES
      DIC_SURVEY_TYPE(TYPE_CODE);
ALTER TABLE THREEDSEIS_SURVEY ADD CONSTRAINT THREEDSEIS_VISIBILITY_FK FOREIGN KEY (VISIBILITY)
      REFERENCES DIC_VISIBILITY(VISIBILITY_CODE);
ALTER TABLE TWODSEIS_ALIAS ADD CONSTRAINT TWODALIAS_CO_FK FOREIGN KEY (ALIAS_COMPANY) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE TWODSEIS_ALIAS ADD CONSTRAINT TWODALIAS_SURVEY_FK FOREIGN KEY (TWODSEIS_SURVEYID) REFERENCES
      TWODSEIS_SURVEY(TWODSEIS_SURVEYID);
ALTER TABLE TWODSEIS_BOUNDING_PT ADD CONSTRAINT TWODSEIS_BND_PT_SURVEY_FK FOREIGN KEY
      (TWODSEIS_SURVEYID) REFERENCES TWODSEIS_SURVEY(TWODSEIS_SURVEYID);
ALTER TABLE TWODSEIS_LINE ADD CONSTRAINT TWODSEIS_LINE_SURVEY_FK FOREIGN KEY (TWODSEIS_SURVEYID)
      REFERENCES TWODSEIS_SURVEY(TWODSEIS_SURVEYID);
ALTER TABLE TWODSEIS_LINE ADD CONSTRAINT TWODSEIS_LINE_VISIBILITY_FK FOREIGN KEY (VISIBILITY) REFERENCES
      DIC_VISIBILITY(VISIBILITY_CODE);
ALTER TABLE TWODSEIS_LINE_PT ADD CONSTRAINT TWODSEIS_LINE_PT_LINE_FK FOREIGN KEY (TWODSEIS_LINEID)
      REFERENCES TWODSEIS_LINE(TWODSEIS_LINEID);
ALTER TABLE TWODSEIS_SURVEY ADD CONSTRAINT TWODSEIS_COUNTRY_FK FOREIGN KEY (COUNTRY) REFERENCES
      COUNTRY(GMICODE);
ALTER TABLE TWODSEIS_SURVEY ADD CONSTRAINT TWODSEIS_DATAORIG_FK FOREIGN KEY (DATA_ORIGIN) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE TWODSEIS_SURVEY ADD CONSTRAINT TWODSEIS_DATAPROV_FK FOREIGN KEY (DATA_PROVIDER) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE TWODSEIS_SURVEY ADD CONSTRAINT TWODSEIS_DICDATUM_FK FOREIGN KEY (DATUM_NAME) REFERENCES
      DIC_DATUM(DATUM_CODE);
ALTER TABLE TWODSEIS_SURVEY ADD CONSTRAINT TWODSEIS_DICEPSG_FK FOREIGN KEY (EPSG_CODE) REFERENCES
      DIC_EPSG(EPSG_CODE);
ALTER TABLE TWODSEIS_SURVEY ADD CONSTRAINT TWODSEIS_DICPROJ_FK FOREIGN KEY (PROJECTION_NAME) REFERENCES
      DIC_PROJECTION(PROJECTION_CODE);
ALTER TABLE TWODSEIS_SURVEY ADD CONSTRAINT TWODSEIS_DICSPHEROID_FK FOREIGN KEY (SPHEROID_NAME)
      REFERENCES DIC_SPHEROID(SPHEROID_CODE);
ALTER TABLE TWODSEIS_SURVEY ADD CONSTRAINT TWODSEIS_VISIBILITY_FK FOREIGN KEY (VISIBILITY) REFERENCES
      DIC_VISIBILITY(VISIBILITY_CODE);
ALTER TABLE USER_INTEREST_BOX ADD CONSTRAINT USER_INTEREST_BOX_LOGIN_FK FOREIGN KEY (LOGIN) REFERENCES
      DEAL_USER(LOGIN);
ALTER TABLE USER_INTEREST_PRODUCT ADD CONSTRAINT USER_INTEREST_PROD_BOX_FK FOREIGN KEY (LOGIN,BOX_NO)
      REFERENCES USER_INTEREST_BOX(LOGIN,BOX_NO);
ALTER TABLE USER_INTEREST_PRODUCT ADD CONSTRAINT USER_INTEREST_PROD_TYPE_FK FOREIGN KEY (PRODUCT_TYPE)
      REFERENCES DIC_PRODUCT_TYPE(TYPE_CODE);
ALTER TABLE USER_PRODUCT_PICK ADD CONSTRAINT USER_PRODUCT_PICK_LOGIN_FK FOREIGN KEY (LOGIN) REFERENCES
      DEAL_USER(LOGIN);
ALTER TABLE USER_PRODUCT_PICK ADD CONSTRAINT USER_PRODUCT_PICK_PROD_FK FOREIGN KEY (FEATUREPRODUCTID)
      REFERENCES FEATUREPRODUCT(FEATUREPRODUCTID);
ALTER TABLE VALID_FEATUREPRODUCT ADD CONSTRAINT VALID_FEATUREPRODUCT_FEAT_FK FOREIGN KEY
      (FEATURE_CLASS_CODE) REFERENCES DIC_FEATURE_CLASS(FEATURE_CLASS_CODE);
ALTER TABLE VALID_FEATUREPRODUCT ADD CONSTRAINT VALID_FEATUREPRODUCT_PROD_FK FOREIGN KEY (PRODUCT_TYPE)
      REFERENCES DIC_PRODUCT_TYPE(TYPE_CODE);
ALTER TABLE WELL ADD CONSTRAINT WELL_DATAORIG_FK FOREIGN KEY (DATA_ORIGIN) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE WELL ADD CONSTRAINT WELL_DATAPROV_FK FOREIGN KEY (DATA_PROVIDER) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE WELL ADD CONSTRAINT WELL_DICDATUM_FK FOREIGN KEY (DATUM_NAME) REFERENCES
      DIC_DATUM(DATUM_CODE);
ALTER TABLE WELL ADD CONSTRAINT WELL_DICEPSG_FK FOREIGN KEY (EPSG_CODE) REFERENCES DIC_EPSG(EPSG_CODE);
ALTER TABLE WELL ADD CONSTRAINT WELL_DICINTENT_FK FOREIGN KEY (WELL_INTENT) REFERENCES
      DIC_WELL_INTENT(INTENT_CODE);
ALTER TABLE WELL ADD CONSTRAINT WELL_DICPROJ_FK FOREIGN KEY (PROJECTION_NAME) REFERENCES
      DIC_PROJECTION(PROJECTION_CODE);
ALTER TABLE WELL ADD CONSTRAINT WELL_DICSPHEROID_FK FOREIGN KEY (SPHEROID_NAME) REFERENCES
      DIC_SPHEROID(SPHEROID_CODE);
ALTER TABLE WELL ADD CONSTRAINT WELL_DICTYPE_FK FOREIGN KEY (WELL_TYPE) REFERENCES
      DIC_WELL_TYPE(TYPE_CODE);
ALTER TABLE WELL ADD CONSTRAINT WELL_DICWELL_DATUM_FK FOREIGN KEY (WELL_DATUM_TYPE) REFERENCES
      DIC_WELL_DATUM_TYPE(TYPE_CODE);
ALTER TABLE WELL ADD CONSTRAINT WELL_DTIWELL_FK FOREIGN KEY (DTI_WELLID) REFERENCES
      DTI_WELL(DTI_WELLID);
ALTER TABLE WELL_ALIAS ADD CONSTRAINT WELL_ALIAS_CO_FK FOREIGN KEY (ALIAS_COMPANY) REFERENCES
      COMPANY(COMPANYID);
ALTER TABLE WELL_ALIAS ADD CONSTRAINT WELL_ALIAS_SURVEY_FK FOREIGN KEY (WELLID) REFERENCES WELL(WELLID);
```

CHECK CONSTRAINTS

```
ALTER TABLE AOI ADD CONSTRAINT AOI_LOC_CK CHECK ((MIN_LAT IS NOT NULL AND MIN_LON IS NOT NULL  AND
      MAX_LAT IS NOT NULL AND MAX_LON IS NOT NULL) OR   (MIN_EASTING IS NOT NULL AND MIN_NORTHING IS NOT
      NULL   AND MAX_EASTING IS NOT NULL AND MAX_NORTHING IS NOT NULL AND PROJECTION_NAME IS NOT NULL));
ALTER TABLE AOI ADD CONSTRAINT AOI_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE BLOCK ADD CONSTRAINT BLOCK_NOLABEL_CK CHECK (BLOCKLABEL = QUADNO||'/'||BLOCKNO OR BLOCKLABEL
      IS NULL);
ALTER TABLE BLOCK ADD CONSTRAINT BLOCK_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE BOUNDARY ADD CONSTRAINT BOUNDARY_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE COAST ADD CONSTRAINT COAST_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE COMPANYHASLICENCE ADD CONSTRAINT COMPANYHASLICENCE_PC_CK CHECK (PERCENTAGE BETWEEN 0 AND
      100);
ALTER TABLE DEAL_USER ADD CONSTRAINT DEAL_USER_ADVERTS_CK CHECK (ALLOWADVERTS IN ('Y','N'));
ALTER TABLE DEAL_USER ADD CONSTRAINT DEAL_USER_COMP_CK CHECK (    (TYPE IN
      ('DATAPROVIDER','SUBSCRIBER','ADMIN_REPORTS','ADMIN' ) AND COMPANYID IS NOT NULL AND     CONTACTREF
      IS NOT NULL ) OR  TYPE='PUBLIC' );
ALTER TABLE DEAL_USER ADD CONSTRAINT DEAL_USER_LOGINUPPER_CK CHECK (LOGIN=UPPER(LOGIN));
ALTER TABLE DEAL_USER ADD CONSTRAINT DEAL_USER_NOTIFY_CK CHECK (NOTIFYNEWPRODUCTS IN ('Y','N'));
ALTER TABLE DEAL_USER ADD CONSTRAINT DEAL_USER_NULLS_CK CHECK ( (TYPE IN ('SUBSCRIBER','PUBLIC') AND
      NOTIFYNEWPRODUCTS IS NOT NULL AND ALLOWADVERTS IS NOT NULL AND DATE_LASTWHATSNEW IS NOT NULL AND
      DATE_LASTNOTIFY CHECK IS NOT NULL) OR (TYPE IN ('DATAPROVIDER','ADMIN','ADMIN_REPORTS')));
ALTER TABLE DEAL_USER ADD CONSTRAINT DEAL_USER_TYPE_CK CHECK (TYPE IN
      ('ADMIN','ADMIN_REPORTS','DATAPROVIDER','SUBSCRIBER','PUBLIC'));
ALTER TABLE DIC_SEAFISH_FEATURE ADD CONSTRAINT DIC_SEAFISH_FEATURE_GRP_CK CHECK (FEATURE_GROUP IN
      ('PIPELINE FEATURE','CABLE FEATURE','PIPELINE STATUS','SEASURFACE FEATURE','SUBSURFACE
      FEATURE','SUB AND SURFACE'));
ALTER TABLE DTI_QUAD ADD CONSTRAINT DTI_QUAD_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE DTI_WELL ADD CONSTRAINT DTI_WELL_FIELDUPPER_CK CHECK (FIELDNAME=UPPER(FIELDNAME));
ALTER TABLE DTI_WELL ADD CONSTRAINT DTI_WELL_LOC_CK CHECK ((LATITUDE IS NOT NULL AND LONGITUDE IS NOT
      NULL) OR  (EASTING IS NOT NULL AND NORTHING IS NOT NULL AND PROJECTION_NAME IS NOT NULL));
ALTER TABLE DTI_WELL ADD CONSTRAINT DTI_WELL_REGUPPER_CK CHECK
      (DTI_WELL_REG_UPPER=UPPER(DTI_WELL_REG_NO));
ALTER TABLE DTI_WELL ADD CONSTRAINT DTI_WELL_UNITS_CK CHECK (VERTICAL_UNITS IN ('F','M'));
ALTER TABLE DTI_WELL ADD CONSTRAINT DTI_WELL_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE FEATUREPRODUCT ADD CONSTRAINT FEATUREPRODUCT_DURLMETH_CK CHECK (PRODUCT_DETAILS_URLMETHOD IN
      ('GET','POST','MAILTO'));
ALTER TABLE FEATUREPRODUCT ADD CONSTRAINT FEATUREPRODUCT_EURLMETH_CK CHECK (ENTITLEMENT_URLMETHOD IN
      ('GET','POST','MAILTO'));
ALTER TABLE FEATUREPRODUCT ADD CONSTRAINT FEATUREPRODUCT_FEAT_CK CHECK  (   (WELLID IS NOT NULL AND
      (THREEDSEIS_SURVEYID IS NULL AND TWODSEIS_SURVEYID IS NULL AND TWODSEIS_LINEID IS NULL)  )  OR
      (THREEDSEIS_SURVEYID IS NOT NULL AND    (WELLID IS NULL AND TWODSEIS_SURVEYID IS NULL AND
      TWODSEIS_LINEID IS NULL)  )  OR  (TWODSEIS_SURVEYID IS NOT NULL AND   (WELLID IS NULL AND
      THREEDSEIS_SURVEYID IS NULL AND TWODSEIS_LINEID IS NULL)  )  OR  (TWODSEIS_LINEID IS NOT NULL AND
      (WELLID IS NULL AND THREEDSEIS_SURVEYID IS NULL AND TWODSEIS_SURVEYID IS NULL)  ));
ALTER TABLE HCFIELD_BOUNDING_PT ADD CONSTRAINT HCFIELD_BND_PT_LOC_CK CHECK ((LATITUDE IS NOT NULL AND
      LONGITUDE IS NOT NULL) OR   (EASTING IS NOT NULL AND NORTHING IS NOT NULL ));
ALTER TABLE HYDROCARBONS_FIELD ADD CONSTRAINT HCFIELD_TYPECK CHECK (HYDROCARBON_TYPE IN
      ('OIL','GAS','OIL,GAS','COND'));
ALTER TABLE HYDROCARBONS_FIELD ADD CONSTRAINT HYDROCARBONS_FIELD_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE LICENCE ADD CONSTRAINT LICENCE_BLOCKLABEL_CK  CHECK (UPPER(LICENCE_LABEL) =  LICENCE_BLOCK
      OR LICENCE_LABEL IS NULL);
ALTER TABLE LICENCE ADD CONSTRAINT LICENCE_BLOCKUPPER_CK CHECK (LICENCE_BLOCK = UPPER(LICENCE_BLOCK));
ALTER TABLE LICENCE ADD CONSTRAINT LICENCE_ISSUEDFLAG_CK CHECK ( LICENCE_ISSUED_FLAG IN ('Y','N'));
ALTER TABLE LICENCE ADD CONSTRAINT LICENCE_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE LICENCE_BOUNDING_LINE ADD CONSTRAINT LICENCE_BOUND_XY_CK CHECK ((START_LATITUDE IS NOT NULL
      AND START_LONGITUDE IS NOT NULL AND END_LATITUDE IS NOT NULL AND END_LONGITUDE IS NOT NULL) OR
      (START_EASTING IS NOT NULL AND START_NORTHING IS NOT NULL  AND END_EASTING IS NOT NULL AND
      END_NORTHING IS NOT NULL ));
ALTER TABLE MEDIAN_LINE ADD CONSTRAINT MEDIAN_LINE_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE PIPELINE ADD CONSTRAINT PIPELINE_NOQUOTES_CK CHECK (PIPELINE_NAME NOT LIKE '%"%' AND
      DESCRIPTION NOT LIKE '%"%' AND FLUID_CONVEYED NOT LIKE '%"%');
ALTER TABLE PIPELINE ADD CONSTRAINT PIPELINE_STATUS_CK CHECK (STATUS IN
      ('ACTIVE','DISUSED','PROPOSED'));
ALTER TABLE PIPELINE ADD CONSTRAINT PIPELINE_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE PIPELINE_PT ADD CONSTRAINT PIPELINE_PT_LOC_CK CHECK ((LATITUDE IS NOT NULL AND LONGITUDE IS
      NOT NULL) OR   (EASTING IS NOT NULL AND NORTHING IS NOT NULL ));
ALTER TABLE PIPELINE_SPAN ADD CONSTRAINT PIPELINE_SPAN_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE PIPELINE_SPAN_PT ADD CONSTRAINT PIPELINE_SPAN_PT_LOC_CK CHECK ((LATITUDE IS NOT NULL AND
      LONGITUDE IS NOT NULL) OR   (EASTING IS NOT NULL AND NORTHING IS NOT NULL ));
ALTER TABLE PUBLIC_CONTACT ADD CONSTRAINT PUBLIC_CONT_COMPUPPER_CK CHECK
      (COMPANY_NAME=UPPER(COMPANY_NAME));
ALTER TABLE PUBLIC_CONTACT ADD CONSTRAINT PUBLIC_CONT_COUNTUPPER_CK CHECK (COUNTRY=UPPER(COUNTRY));
ALTER TABLE PUBLIC_CONTACT ADD CONSTRAINT PUBLIC_CONT_PASSUPPER_CK CHECK (PASSWORD=UPPER(PASSWORD));
ALTER TABLE PUBLIC_CONTACT ADD CONSTRAINT PUBLIC_CONT_POSTUPPER_CK CHECK (POSTCODE=UPPER(POSTCODE));
ALTER TABLE PUBLIC_CONTACT ADD CONSTRAINT PUBLIC_CONT_STREETUPPER_CK CHECK
      (STREETADDRESS=UPPER(STREETADDRESS));
ALTER TABLE PUBLIC_CONTACT ADD CONSTRAINT PUBLIC_CONT_TOWNUPPER_CK CHECK (TOWN=UPPER(TOWN));
ALTER TABLE QUAD ADD CONSTRAINT QUAD_NOLABEL_CK CHECK (QUADLABEL = QUADNO OR QUADLABEL IS NULL);
ALTER TABLE QUAD ADD CONSTRAINT QUAD_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE SAFETY_EX_ZONE ADD CONSTRAINT SAFETY_EX_ZONE_STATUS_CK CHECK (STATUS IN
      ('ACTIVE','DISUSED','PROPOSED'));
ALTER TABLE SAFETY_EX_ZONE ADD CONSTRAINT SAFETY_EX_ZONE_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE SAFETY_EX_ZONE ADD CONSTRAINT SAFETY_ZEX_ONE_CENTRE_GEOM_CK CHECK (CENTRE_GEOMETRY_TYPE IN
      ('POINT','LINE','POLYGON'));
ALTER TABLE SAFETY_EX_ZONE_CENTRE_PT ADD CONSTRAINT SAFETY_EX_ZONE_CNTRPT_LOC_CK CHECK ((LATITUDE IS NOT
      NULL AND LONGITUDE IS NOT NULL) OR   (EASTING IS NOT NULL AND NORTHING IS NOT NULL ));
ALTER TABLE SEASURFACE_INF ADD CONSTRAINT SEASURFACE_INF_FLAG_CK CHECK (PLATFORM_FLAG IN ('Y','N'));
ALTER TABLE SEASURFACE_INF ADD CONSTRAINT SEASURFACE_INF_LOC_CK CHECK ((LATITUDE IS NOT NULL AND
      LONGITUDE IS NOT NULL) OR   (EASTING IS NOT NULL AND NORTHING IS NOT NULL AND PROJECTION_NAME IS
      NOT NULL));
ALTER TABLE SEASURFACE_INF ADD CONSTRAINT SEASURFACE_INF_STATUS_CK CHECK (STATUS IN
      ('ACTIVE','DISUSED','PROPOSED'));
ALTER TABLE SEASURFACE_INF ADD CONSTRAINT SEASURFACE_INF_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE SUBSEA_INF ADD CONSTRAINT SUBSEA_INF_FLAG_CK CHECK (SUSWELL_FLAG IN ('Y','N'));
ALTER TABLE SUBSEA_INF ADD CONSTRAINT SUBSEA_INF_LOC_CK CHECK ((LATITUDE IS NOT NULL AND LONGITUDE IS
      NOT NULL) OR   (EASTING IS NOT NULL AND NORTHING IS NOT NULL AND PROJECTION_NAME IS NOT NULL));
ALTER TABLE SUBSEA_INF ADD CONSTRAINT SUBSEA_INF_STATUS_CK CHECK (STATUS IN
      ('ACTIVE','DISUSED','PROPOSED'));
ALTER TABLE SUBSEA_INF ADD CONSTRAINT SUBSEA_INF_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE THREEDSEIS_BOUNDING_PT ADD CONSTRAINT THREEDSEIS_BND_PT_LOC_CK CHECK ((LATITUDE IS NOT NULL
      AND LONGITUDE IS NOT NULL) OR   (EASTING IS NOT NULL AND NORTHING IS NOT NULL ));
ALTER TABLE THREEDSEIS_SURVEY ADD CONSTRAINT THREEDSEIS_CS9UPPER_CK CHECK
      (THREEDSEIS_CS9NAME=UPPER(THREEDSEIS_CS9NAME));
ALTER TABLE THREEDSEIS_SURVEY ADD CONSTRAINT THREEDSEIS_OPERATORUPPER_CK CHECK
      (OPERATOR=UPPER(OPERATOR));
ALTER TABLE THREEDSEIS_SURVEY ADD CONSTRAINT THREEDSEIS_SURVEY_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE TWODSEIS_BOUNDING_PT ADD CONSTRAINT TWODSEIS_BND_PT_LOC_CK CHECK ((LATITUDE IS NOT NULL AND
      LONGITUDE IS NOT NULL) OR   (EASTING IS NOT NULL AND NORTHING IS NOT NULL ));
ALTER TABLE TWODSEIS_LINE_PT ADD CONSTRAINT TWODSEIS_LINE_PT_LOC_CK CHECK ((LATITUDE IS NOT NULL AND
      LONGITUDE IS NOT NULL) OR   (EASTING IS NOT NULL AND NORTHING IS NOT NULL ));
ALTER TABLE TWODSEIS_SURVEY ADD CONSTRAINT TWODSEIS_CS9UPPER_CK CHECK
      (TWODSEIS_CS9NAME=UPPER(TWODSEIS_CS9NAME));
ALTER TABLE TWODSEIS_SURVEY ADD CONSTRAINT TWODSEIS_SURVEY_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE WELL ADD CONSTRAINT WELL_LOC_CK CHECK ((LATITUDE IS NOT NULL AND LONGITUDE IS NOT NULL) OR
      (EASTING IS NOT NULL AND NORTHING IS NOT NULL AND PROJECTION_NAME IS NOT NULL));
ALTER TABLE WELL ADD CONSTRAINT WELL_OPERATORUPPER_CK CHECK (OPERATOR=UPPER(OPERATOR));
ALTER TABLE WELL ADD CONSTRAINT WELL_RELEASESTATUS_CK CHECK (RELEASE_STATUS IN ('Y','N'));
ALTER TABLE WELL ADD CONSTRAINT WELL_UNITS_CK CHECK (VERTICAL_UNITS IN ('F','M'));
ALTER TABLE WELL ADD CONSTRAINT WELL_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE WVS_COAST ADD CONSTRAINT WVS_COAST_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE WVS_POLY_COAST ADD CONSTRAINT WVS_POLY_COAST_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
ALTER TABLE WVSCOAST ADD CONSTRAINT WVSCOAST_VERIF_CK CHECK (VERIFIED IN ('Y','N'));
```

SEQUENCES

```
CREATE SEQUENCE BLOCKID_SEQNO START WITH             010000001 INCREMENT BY 1;
CREATE SEQUENCE QUADID_SEQNO START WITH              020000001 INCREMENT BY 1;
CREATE SEQUENCE THREEDSEIS_SURVEYID_SEQNO START WITH 030000001 INCREMENT BY 1;
CREATE SEQUENCE TWODSEIS_SURVEYID_SEQNO START WITH   040000001 INCREMENT BY 1;
CREATE SEQUENCE HCFIELDID_SEQNO START WITH           050000001 INCREMENT BY 1;
CREATE SEQUENCE MEDIAN_LINEID_SEQNO START WITH       060000001 INCREMENT BY 1;
CREATE SEQUENCE COASTID_SEQNO START WITH             070000001 INCREMENT BY 1;
CREATE SEQUENCE LICENCEID_SEQNO START WITH           100000001 INCREMENT BY 1;
CREATE SEQUENCE DTI_WELLID_SEQNO START WITH          200000001 INCREMENT BY 1;
CREATE SEQUENCE WELLID_SEQNO START WITH              300000001 INCREMENT BY 1;
CREATE SEQUENCE TWODSEIS_LINEID_SEQNO START WITH     400000001 INCREMENT BY 1;
CREATE SEQUENCE PIPELINEID_SEQNO START WITH          500000001 INCREMENT BY 1;
CREATE SEQUENCE PIPELINE_SPANID_SEQNO START WITH     600000001 INCREMENT BY 1;
CREATE SEQUENCE SEASURFACE_INFID_SEQNO START WITH    700000001 INCREMENT BY 1;
CREATE SEQUENCE SUBSEA_INFID_SEQNO START WITH        800000001 INCREMENT BY 1;
CREATE SEQUENCE SAFETY_EX_ZONEID_SEQNO START WITH    900000001 INCREMENT BY 1;
CREATE SEQUENCE BATHYMETRYID_SEQNO START WITH       1000000001 INCREMENT BY 1;
CREATE SEQUENCE FEATUREPRODUCTID_SEQNO START WITH   2000000001 INCREMENT BY 1;
```

INDEXES

-- NB also have index on each SHAPE column - created automatically by SDE

```
CREATE INDEX COMPANY_NAME_INX ON COMPANY(COMPANYNAME);
CREATE INDEX COMPANY_TYPE_INX ON COMPANY(COMPANY_TYPE);
CREATE INDEX COUNTRY_FIPS_INX ON COUNTRY(FIPSCODE);
CREATE INDEX COUNTRY_NAME_INX ON COUNTRY(NAME);
CREATE INDEX DEAL_USER_COMP_INX ON DEAL_USER(COMPANYID);
CREATE INDEX DIC_PRODUCT_SUPERTYPE_INX ON DIC_PRODUCT_TYPE(SUPER_TYPE_CODE);
CREATE INDEX DTI_WELL_DTIDEN_INX ON DTI_WELL(DTI_DEN_NO);
CREATE INDEX FEATUREPRODUCT_DATELOADED_INX ON FEATUREPRODUCT(DATE_ENTERED);
CREATE INDEX FEATUREPRODUCT_PRODTYPE_INX ON FEATUREPRODUCT(PRODUCT_TYPE);
CREATE INDEX FEATUREPRODUCT_PROV_INX ON FEATUREPRODUCT(PROVIDER_COMPID);
CREATE INDEX FEATUREPRODUCT_THREEDSEIS_INX ON FEATUREPRODUCT(THREEDSEIS_SURVEYID);
CREATE INDEX FEATUREPRODUCT_TWODLINE_INX ON FEATUREPRODUCT(TWODSEIS_LINEID);
CREATE INDEX FEATUREPRODUCT_TWODSEIS_INX ON FEATUREPRODUCT(TWODSEIS_SURVEYID);
CREATE INDEX FEATUREPRODUCT_WELLID_INX ON FEATUREPRODUCT(WELLID);
CREATE INDEX HYDROCARBONS_FIELD_NAME_INX ON HYDROCARBONS_FIELD(FIELDNAME);
CREATE INDEX LICENCE_BLOCK_INX ON LICENCE(LICENCE_BLOCK);
CREATE INDEX LICENCE_OPERATOR_INX ON LICENCE(OPERATOR);
CREATE INDEX LICENCE_ROUND_INX ON LICENCE(LICENCE_ROUND);
CREATE INDEX PIPELINE_NAME_INX ON PIPELINE(PIPELINE_NAME);
CREATE INDEX PIPELINE_OPER_INX ON PIPELINE(OPERATOR);
CREATE INDEX SAFETY_EX_ZONE_FEAT_IND ON SAFETY_EX_ZONE(CENTRE_FEATURE_TYPE);
CREATE INDEX SAFETY_EX_ZONE_OPER_IND ON SAFETY_EX_ZONE(OPERATOR);
CREATE INDEX SEASURFACE_INF_FEAT_IND ON SEASURFACE_INF(FEATURE_TYPE);
CREATE INDEX SEASURFACE_INF_OPER_IND ON SEASURFACE_INF(OPERATOR);
CREATE INDEX SUBSEA_INF_FEAT_IND ON SUBSEA_INF(FEATURE_TYPE);
CREATE INDEX SUBSEA_INF_OPER_IND ON SUBSEA_INF(OPERATOR);
CREATE INDEX THREEDSEIS_ALIAS_INX ON THREEDSEIS_ALIAS(ALIAS_COMPANY);
CREATE INDEX THREEDSEIS_COUNTRY_INX ON THREEDSEIS_SURVEY(COUNTRY);
CREATE INDEX THREEDSEIS_OPERATOR_INX ON THREEDSEIS_SURVEY(OPERATOR);
CREATE INDEX THREEDSEIS_RECORDLEN_INX ON THREEDSEIS_SURVEY(RECORD_LENGTH);
CREATE INDEX TWODSEIS_COUNTRY_INX ON TWODSEIS_SURVEY(COUNTRY);
CREATE INDEX TWODSEIS_OPERATOR_INX ON TWODSEIS_SURVEY(OPERATOR);
CREATE INDEX TWODSEIS_RECORDLEN_INX ON TWODSEIS_SURVEY(RECORD_LENGTH);
CREATE INDEX WELL_ALIAS_COMP_INX ON WELL_ALIAS(ALIAS_COMPANY);
CREATE INDEX WELL_DTIWELL_INX ON WELL(DTI_WELLID);
CREATE INDEX WELL_OPERATOR_INX ON WELL(OPERATOR);
CREATE INDEX WELL_TOTALDEPTH_INX ON WELL(TOTAL_DEPTH);
CREATE INDEX WELL_WELLNAME_INX ON WELL(WELLNAME);
```

FUNCTIONS

WITHIN_AOI
function takes a value of latitude and longitude and compares with the min and max stored in AOI table. Returns Y if input lat and long are within the AOI limits. Returns N if input lat and long are outside AOI limits

TO_EPSG
function takes datum code, spheroid code and projection code and returns the correct EPSG code (read from DIC_EPSG)

EPSG_TO_SPHEROID
function takes epsg code and returns the spheroid code (read from DIC_EPSG)

EPSG_TO_PROJECTION
function takes epsg code and returns the projection code (read from DIC_EPSG)

EPSG_TO_DATUM
function takes epsg code and returns the datum code (read from DIC_EPSG)

DEGMIN_TO_DECIMALDEGS
function reads a latitude or longitude value from character string of degrees [space] decimal minutes [space] and N/S/E/W e.g. 50 02.445 S converts to signed decimal degrees for data loading

TRIGGERS

TABLE_NAME	TRIGGER_NAME	COMMENTS
AOI	AOI_CSYS_FROM_EPSG	INSERTS OR UPDATES TO CORRECT SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME ON INSERT OR UPDATE OF EPSG_CODE
AOI	AOI_EPSG_RULES_CSYS	SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME NOT ALLOWED INSERT OR UPDATE TO VALUES OTHER THAN THOSE THAT AGREE WITH THE EPSG_CODE
AOI	AOI_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
AOI	SPCOL_DEL_CASCADE_2	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
BATHYMETRY	BATHYMETRY_CSYS_FROM_EPSG	INSERTS OR UPDATES TO CORRECT SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME ON INSERT OR UPDATE OF EPSG_CODE
BATHYMETRY	BATHYMETRY_EPSG_RULES_CSYS	SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME NOT ALLOWED INSERT OR UPDATE TO VALUES OTHER THAN THOSE THAT AGREE WITH THE EPSG_CODE
BATHYMETRY	BATHYMETRY_SEQNO_ONINSERT	INSERTS NEXT NUMBER IN BATHYMETRYID_SEQNO SEQUENCE FOR BATHYMETRYID VALUE
BATHYMETRY	BATHYMETRY_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE

BLOCK	BLOCK_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
BLOCK	SPCOL_DEL_CASCADE_3	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
BOUNDARY	SPCOL_DEL_CASCADE_4	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
COAST	COAST_CSYS_FROM_EPSG	INSERTS OR UPDATES TO CORRECT SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME ON INSERT OR UPDATE OF EPSG_CODE
COAST	COAST_EPSG_RULES_CSYS	SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME NOT ALLOWED INSERT OR UPDATE TO VALUES OTHER THAN THOSE THAT AGREE WITH THE EPSG_CODE
COAST	COAST_SEQNO_ONINSERT	INSERTS NEXT NUMBER IN COASTID_SEQNO SEQUENCE FOR COASTID VALUE
COAST	COAST_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
COAST	SPCOL_DEL_CASCADE_9	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
COMPANY	COMPANY_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
COMPANY_CONTACT	COMPANY_CONTACT_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
CONTACTHASROLE	CONTACTHASROLE_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
COUNTRY	COUNTRY_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DATA_SAFETY_EX_ZONE_PT	SPCOL_DEL_CASCADE_33	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
DEAL_USER	DEAL_USER_PUBLICCONTACT	INSERT OF DEAL_USER.TYPE='PUBLIC' IS NOT ALLOWED UNLESS THERE IS A RECORD IN PUBLIC_CONTACT FOR THE SAME LOGIN
DEAL_USER	DEAL_USER_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_COMPANY_TYPE	DIC_COMPANY_TYPE_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_CONTACT_ROLE	DIC_CONTACT_ROLE_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_DATUM	DIC_DATUM_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_DEFINED_LINE_TYPE	DIC_DEF_LINE_TYPE_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_DOWNLOAD_TYPE	DIC_DOWNLOAD_TYPE_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_EPSG	DIC_EPSG_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_FEATURE_CLASS	DIC_FEATURE_CLASS_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_MEDIUM	DIC_MEDIUM_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_PRODUCT_CLASS	DIC_PRODUCT_CLASS_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_PRODUCT_TYPE	DIC_PRODUCT_TYPE_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_PROJECTION	DIC_PROJECTION_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_SEAFISH_FEATURE	DIC_SEAFISH_FEATURE_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_SPHEROID	DIC_SPHEROID_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_SURVEY_TYPE	DIC_SURVEY_TYPE_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_VISIBILITY	DIC_VISIBILITY_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_WELL_DATUM_TYPE	DIC_WELL_DATUM_TYPE_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_WELL_INTENT	DIC_WELL_INTENT_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DIC_WELL_TYPE	DIC_WELL_TYPE_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED

		AUTOMATICALLY ON INSERT OR UPDATE
DTI_QUAD	SPCOL_DEL_CASCADE_12	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
DTI_WELL	DTI_WELL_CSYS_FROM_EPSG	INSERTS OR UPDATES TO CORRECT SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME ON INSERT OR UPDATE OF EPSG_CODE
DTI_WELL	DTI_WELL_EPSG_RULES_CSYS	SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME NOT ALLOWED INSERT OR UPDATE TO VALUES OTHER THAN THOSE THAT AGREE WITH THE EPSG_CODE
DTI_WELL	DTI_WELL_IN_AOI	INSERT OR UPDATE OF LATITUDE, LONGITUDE NOT ALLOWED IF VALUES ARE OUTSIDE THE LIMITS OF THE DEAL AREA OF INTEREST
DTI_WELL	DTI_WELL_SEQNO_ONINSERT	INSERTS NEXT NUMBER IN DTI_WELLID_SEQNO SEQUENCE FOR DTI_WELLID VALUE
DTI_WELL	DTI_WELL_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
DTI_WELL	SPCOL_DEL_CASCADE_7	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
FEATUREPRODUCT	FEATUREPRODUCT_SEQNO_ONINSERT	INSERTS NEXT NUMBER IN FEATUREPODUCTID_SEQNO SEQUENCE FOR FEATUREPRODUCTID VALUE
FEATUREPRODUCT	FEATUREPRODUCT_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
FEEDBACK	FEEDBACK_DATE_ONINSERT	DATE_ENTERED ENTERED AUTOMATICALLY ON INSERT
HCFIELD_BOUNDING_PT	HCFIELD_BNDPT_PROJ	EASTING AND NORTHING NOT ALLOWED TO BE INSERTED OR UPDATED TO VALUES OTHER THAN NULL IF PARENT HYDROCARBONS_FIELD.PROJECTION_NAME IS NULL
HCFIELD_BOUNDING_PT	HCFIELD_BOUNDING_PT_IN_AOI	INSERT OR UPDATE OF LATITUDE, LONGITUDE NOT ALLOWED IF VALUES ARE OUTSIDE THE LIMITS OF THE DEAL AREA OF INTEREST
HCFIELD_BOUNDING_PT	HCFIELD_BOUNDING_PT_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
HYDROCARBONS_FIELD	HCFIELD_CSYS_FROM_EPSG	INSERTS OR UPDATES TO CORRECT SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME ON INSERT OR UPDATE OF EPSG_CODE
HYDROCARBONS_FIELD	HCFIELD_EPSG_RULES_CSYS	SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME NOT ALLOWED INSERT OR UPDATE TO VALUES OTHER THAN THOSE THAT AGREE WITH THE EPSG_CODE
HYDROCARBONS_FIELD	HCFIELD_PROJ	PROJECTION_NAME NOT ALLOWED TO BE NULL IF CHILD HCFIELD_BOUNDING_PT.EASTING OR NORTHING IS NOT NULL
HYDROCARBONS_FIELD	HYDROCARBONS_FIELD_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
HYDROCARBONS_FIELD	HYDROCARBONS_SEQNO_ONINSERT	INSERTS NEXT NUMBER IN HCFIELDID_SEQNO SEQUENCE FOR HCFIELDID VALUE
HYDROCARBONS_FIELD	SPCOL_DEL_CASCADE_8	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
KA_TEST_SOLID	SPCOL_DEL_CASCADE_1	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
LICENCE	LICENCE_CSYS_FROM_EPSG	INSERTS OR UPDATES TO CORRECT SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME ON INSERT OR UPDATE OF EPSG_CODE
LICENCE	LICENCE_DATES_LT_SYSDATE	START_DATE MUST BE LESS THAN SYSDATE ON INSERT OR UPDATE
LICENCE	LICENCE_EPSG_RULES_CSYS	SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME NOT ALLOWED INSERT OR UPDATE TO VALUES OTHER THAN THOSE THAT AGREE WITH THE EPSG_CODE
LICENCE	LICENCE_PROJ	PROJECTION_NAME NOT ALLOWED TO BE NULL IF CHILD LICENCE_BOUNDING_PT.EASTING OR NORTHING IS NOT NULL
LICENCE	LICENCE_SEQNO_ONINSERT	INSERTS NEXT NUMBER IN LICENCEID_SEQNO

		SEQUENCE FOR LICENCEID VALUE
LICENCE	LICENCE_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
LICENCE	SPCOL_DEL_CASCADE_10	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
LICENCE_BOUNDING_LINE	LICENCE_BNDLINE_PROJ	EASTING AND NORTHING NOT ALLOWED TO BE INSERTED OR UPDATED TO VALUES OTHER THAN NULL IF PARENT LICENCE.PROJECTION_NAME IS NULL
LICENCE_BOUNDING_LINE	LICENCE_BNDLN_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
MEDIAN_LINE	MEDIAN_LINE_CSYS_FROM_EPSG	INSERTS OR UPDATES TO CORRECT SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME ON INSERT OR UPDATE OF EPSG_CODE
MEDIAN_LINE	MEDIAN_LINE_EPSG_RULES_CSYS	SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME NOT ALLOWED INSERT OR UPDATE TO VALUES OTHER THAN THOSE THAT AGREE WITH THE EPSG_CODE
MEDIAN_LINE	MEDIAN_LINE_SEQNO_ONINSERT	INSERTS NEXT NUMBER IN MEDIAN_LINEID_SEQNO SEQUENCE FOR MEDIAN_LINEID VALUE
MEDIAN_LINE	MEDIAN_LINE_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
MEDIAN_LINE	SPCOL_DEL_CASCADE_11	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
PIPELINE	PIPELINE_CSYS_FROM_EPSG	INSERTS OR UPDATES TO CORRECT SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME ON INSERT OR UPDATE OF EPSG_CODE
PIPELINE	PIPELINE_EPSG_RULES_CSYS	SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME NOT ALLOWED INSERT OR UPDATE TO VALUES OTHER THAN THOSE THAT AGREE WITH THE EPSG_CODE
PIPELINE	PIPELINE_PROJ	PROJECTION_NAME NOT ALLOWED TO BE NULL IF CHILD PIPELINE_PT.EASTING OR NORTHING OR PIPELINE_SPAN_PT.EASTING OR NORTHING IS NOT NULL
PIPELINE	PIPELINE_SEQNO_ONINSERT	INSERTS NEXT NUMBER IN PIPELINEID_SEQNO SEQUENCE FOR PIPELINEID VALUE
PIPELINE	PIPELINE_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
PIPELINE	SPCOL_DEL_CASCADE_23	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
PIPELINE_PT	PIPELINE_PT_IN_AOI	INSERT OR UPDATE OF LATITUDE, LONGITUDE NOT ALLOWED IF VALUES ARE OUTSIDE THE LIMITS OF THE DEAL AREA OF INTEREST
PIPELINE_PT	PIPELINE_PT_PROJ	EASTING AND NORTHING NOT ALLOWED TO BE INSERTED OR UPDATED TO VALUES OTHER THAN NULL IF PARENT PIPELINE.PROJECTION_NAME IS NULL
PIPELINE_PT	PIPELINE_PT_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
PIPELINE_SPAN	PIPELINE_SPAN_SEQNO_ONINSERT	INSERTS NEXT NUMBER IN PIPELINE_SPANID_SEQNO SEQUENCE FOR PIELINE_SPANID VALUE
PIPELINE_SPAN	PIPELINE_SPAN_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
PIPELINE_SPAN	PIPESPAN_SEQNO_ONINSERT	INSERTS NEXT NUMBER IN PIPELINE_SPANID_SEQNO SEQUENCE FOR PIPELINE_SPANID VALUE
PIPELINE_SPAN_PT	PIPELINE_SPAN_PT_IN_AOI	INSERT OR UPDATE OF LATITUDE, LONGITUDE NOT ALLOWED IF VALUES ARE OUTSIDE THE LIMITS OF THE DEAL AREA OF INTEREST
PIPELINE_SPAN_PT	PIPELINE_SPAN_PT_PROJ	EASTING AND NORTHING NOT ALLOWED TO BE INSERTED OR UPDATED TO VALUES OTHER THAN NULL IF PARENT PIPELINE.PROJECTION_NAME IS NULL
PIPELINE_SPAN_PT	PIPELINE_SPAN_PT_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
PRODTYPE_CLSN	PRODTYPE_CLSN_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE

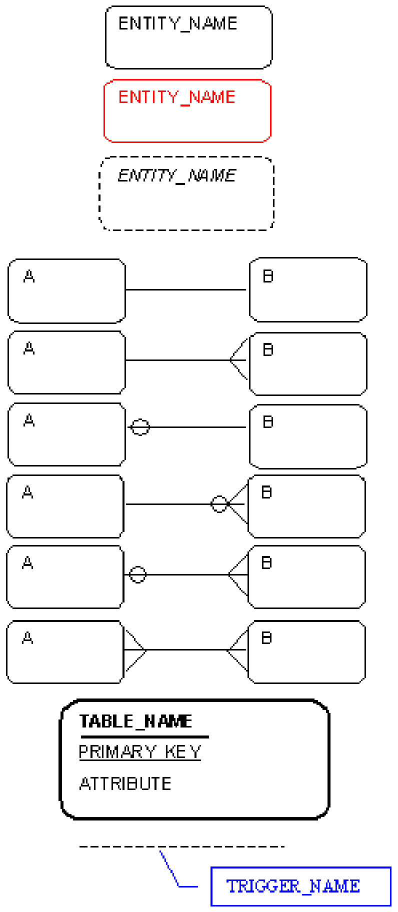
PRODUCT_ENTITLEMENT	PRODUCT_ENTITLEMENT_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
PUBLIC_CONTACT	PUBLIC_CONTACT_DELETE	DELETE OF RECORD NOT ALLOWED WHILE THERE EXISTS A RECORD IN DEAL_USER FOR THE SAME LOGIN WHERE TYPE='PUBLIC'
PUBLIC_CONTACT	PUBLIC_CONTACT_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
QUAD	QUAD_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
QUAD	SPCOL_DEL_CASCADE_21	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
SAFETY_EX_ZONE	SAFETY_EX_ZONE_PROJ	PROJECTION_NAME NOT ALLOWED TO BE NULL IF CHILD SAFETY_EX_ZONE_CENTRE_PT.EASTING OR NORTHING IS NOT NULL
SAFETY_EX_ZONE	SAFETY_EX_ZONE_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
SAFETY_EX_ZONE	SAFETY_ZONE_CSYS_FROM_EPSG	INSERTS OR UPDATES TO CORRECT SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME ON INSERT OR UPDATE OF EPSG_CODE
SAFETY_EX_ZONE	SAFETY_ZONE_DATE_LT_SYSDATE	DATE_ISSUED MUST BE LESS THAN SYSDATE ON INSERT OR UPDATE
SAFETY_EX_ZONE	SAFETY_ZONE_EPSG_RULES_CSYS	SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME NOT ALLOWED INSERT OR UPDATE TO VALUES OTHER THAN THOSE THAT AGREE WITH THE EPSG_CODE
SAFETY_EX_ZONE	SAFETY_ZONE_SEQNO_ONINSERT	INSERTS NEXT NUMBER IN SAFETY_EX_ZONEID_SEQNO SEQUENCE FOR SAFETY_EX_ZONEID VALUE
SAFETY_EX_ZONE_CENTRE_PT	SAFETY_ZONE_CENTRE_PT_IN_AOI	INSERT OR UPDATE OF LATITUDE, LONGITUDE NOT ALLOWED IF VALUES ARE OUTSIDE THE LIMITS OF THE DEAL AREA OF INTEREST
SAFETY_EX_ZONE_CENTRE_PT	SAFETY_ZONE_CENTRE_PT_PROJ	EASTING AND NORTHING NOT ALLOWED TO BE INSERTED OR UPDATED TO VALUES OTHER THAN NULL IF PARENT SAFETY_EX_ZONE.PROJECTION_NAME IS NULL
SAFETY_EX_ZONE_CENTRE_PT	SAFETY_ZONE_CTR_PT_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
SEASURFACE_INF	SEASURFACE_INF_CSYS_FROM_EPSG	INSERTS OR UPDATES TO CORRECT SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME ON INSERT OR UPDATE OF EPSG_CODE
SEASURFACE_INF	SEASURFACE_INF_DATE_LT_SYSDATE	DATE_ISSUED MUST BE LESS THAN SYSDATE ON INSERT OR UPDATE
SEASURFACE_INF	SEASURFACE_INF_EPSG_RULES_CSYS	SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME NOT ALLOWED INSERT OR UPDATE TO VALUES OTHER THAN THOSE THAT AGREE WITH THE EPSG_CODE
SEASURFACE_INF	SEASURFACE_INF_IN_AOI	INSERT OR UPDATE OF LATITUDE, LONGITUDE NOT ALLOWED IF VALUES ARE OUTSIDE THE LIMITS OF THE DEAL AREA OF INTEREST
SEASURFACE_INF	SEASURFACE_INF_SEQNO_ONINSERT	INSERTS NEXT NUMBER IN SEASURFACE_INFID_SEQNO SEQUENCE FOR SEASURFACE_INFID VALUE
SEASURFACE_INF	SEASURFACE_INF_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
SEASURFACE_INF	SPCOL_DEL_CASCADE_30	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
SUBSEA_INF	SPCOL_DEL_CASCADE_29	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
SUBSEA_INF	SUBSEA_INF_CSYS_FROM_EPSG	INSERTS OR UPDATES TO CORRECT SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME ON INSERT OR UPDATE OF EPSG_CODE
SUBSEA_INF	SUBSEA_INF_DATE_LT_SYSDATE	DATE_ISSUED MUST BE LESS THAN SYSDATE ON INSERT OR UPDATE
SUBSEA_INF	SUBSEA_INF_EPSG_RULES_CSYS	SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME NOT ALLOWED INSERT OR UPDATE TO VALUES OTHER THAN THOSE THAT AGREE WITH THE EPSG_CODE
SUBSEA_INF	SUBSEA_INF_IN_AOI	INSERT OR UPDATE OF LATITUDE, LONGITUDE NOT ALLOWED IF VALUES ARE OUTSIDE THE

		LIMITS OF THE DEAL AREA OF INTEREST
SUBSEA_INF	SUBSEA_INF_SEQNO_ONINSERT	INSERTS NEXT NUMBER IN SUBSEA_INFID_SEQNO SEQUENCE FOR SUBSEA_INFID VALUE
SUBSEA_INF	SUBSEA_INF_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
THREEDSEIS_ALIAS	THREEDSEIS_ALIAS_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
THREEDSEIS_BOUNDING_PT	THREEDSEIS_BNDPT_IN_AOI	INSERT OR UPDATE OF LATITUDE, LONGITUDE NOT ALLOWED IF VALUES ARE OUTSIDE THE LIMITS OF THE DEAL AREA OF INTEREST
THREEDSEIS_BOUNDING_PT	THREEDSEIS_BNDPT_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
THREEDSEIS_BOUNDING_PT	THREEDSEIS_SURVEY_BNDLINE_PROJ	EASTING AND NORTHING NOT ALLOWED TO BE INSERTED OR UPDATED TO VALUES OTHER THAN NULL IF PARENT THREEDSEIS_SURVEY.PROJECTION_NAME IS NULL
THREEDSEIS_SURVEY	SPCOL_DEL_CASCADE_6	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
THREEDSEIS_SURVEY	THREEDSEIS_CSYS_FROM_EPSG	INSERTS OR UPDATES TO CORRECT SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME ON INSERT OR UPDATE OF EPSG_CODE
THREEDSEIS_SURVEY	THREEDSEIS_DATES_LT_SYSDATE	START_DATE_SHOT AND END_DATE_SHOT MUST BE LESS THAN SYSDATE ON INSERT OR UPDATE
THREEDSEIS_SURVEY	THREEDSEIS_EPSG_RULES_CSYS	SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME NOT ALLOWED INSERT OR UPDATE TO VALUES OTHER THAN THOSE THAT AGREE WITH THE EPSG_CODE
THREEDSEIS_SURVEY	THREEDSEIS_PROJ	PROJECTION_NAME NOT ALLOWED TO BE NULL IF CHILD THREEDSEIS_BOUNDING_PT.EASTING OR NORTHING IS NOT NULL
THREEDSEIS_SURVEY	THREEDSEIS_SEQNO_ONINSERT	INSERTS NEXT NUMBER IN THREEDSEIS_SURVEYID_SEQNO SEQUENCE FOR THREEDSEIS_SURVEYID VALUE
THREEDSEIS_SURVEY	THREEDSEIS_SURVEY_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
TWODSEIS_ALIAS	TWODSEIS_ALIAS_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
TWODSEIS_BOUNDING_PT	TWODSEIS_BNDPT_IN_AOI	INSERT OR UPDATE OF LATITUDE, LONGITUDE NOT ALLOWED IF VALUES ARE OUTSIDE THE LIMITS OF THE DEAL AREA OF INTEREST
TWODSEIS_BOUNDING_PT	TWODSEIS_BNDPT_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
TWODSEIS_BOUNDING_PT	TWODSEIS_SURVEY_BNDLINE_PROJ	EASTING AND NORTHING NOT ALLOWED TO BE INSERTED OR UPDATED TO VALUES OTHER THAN NULL IF PARENT TWODSEIS_SURVEY.PROJECTION_NAME IS NULL
TWODSEIS_LINE	TWODSEIS_LINE_SEQNO_ONINSERT	INSERTS NEXT NUMBER IN TWODSEIS_LINEID_SEQNO SEQUENCE FOR TWOSEIS_LINEID VALUE
TWODSEIS_LINE	TWODSEIS_LINE_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
TWODSEIS_LINE_PT	TWODSEIS_LINE_PT_IN_AOI	INSERT OR UPDATE OF LATITUDE, LONGITUDE NOT ALLOWED IF VALUES ARE OUTSIDE THE LIMITS OF THE DEAL AREA OF INTEREST
TWODSEIS_LINE_PT	TWODSEIS_LINE_PT_PROJ	EASTING AND NORTHING NOT ALLOWED TO BE INSERTED OR UPDATED TO VALUES OTHER THAN NULL IF PARENT TWODSEIS_SURVEY.PROJECTION_NAME IS NULL
TWODSEIS_LINE_PT	TWODSEIS_LINE_PT_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
TWODSEIS_SURVEY	SPCOL_DEL_CASCADE_26	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
TWODSEIS_SURVEY	TWODSEIS_CSYS_FROM_EPSG	INSERTS OR UPDATES TO CORRECT SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME ON INSERT OR UPDATE OF EPSG_CODE
TWODSEIS_SURVEY	TWODSEIS_DATES_LT_SYSDATE	START_DATE_SHOT AND END_DATE_SHOT MUST BE LESS THAN SYSDATE ON INSERT OR UPDATE
TWODSEIS_SURVEY	TWODSEIS_EPSG_RULES_CSYS	SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME NOT ALLOWED INSERT OR UPDATE TO VALUES OTHER THAN THOSE THAT AGREE WITH THE EPSG_CODE
TWODSEIS_SURVEY	TWODSEIS_PROJ	PROJECTION_NAME NOT ALLOWED TO BE NULL IF

		CHILD TWODSEIS_BOUNDING_PT.EASTING OR NORTHING OR TWODSEIS_LINE_PT.EASTING OR NORTHING IS NOT NULL
TWODSEIS_SURVEY	TWODSEIS_SEQNO_ONINSERT	INSERTS NEXT NUMBER IN TWODSEIS_SURVEYID_SEQNO SEQUENCE FOR TWODSEIS_SURVEYID VALUE
TWODSEIS_SURVEY	TWODSEIS_SURVEY_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
USER_INTEREST_BOX	USER_INTEREST_BOX_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
USER_INTEREST_BOX	USER_INTEREST_BOX_USERTYPE	LOGIN MUST MATCH A LOGIN IN DEAL_USER WHERE DEAL_USER.TYPE='PUBLIC' OR 'SUBSCRIBER'
USER_INTEREST_PRODUCT	USER_INTR_PROD_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
USER_PRODUCT_PICK	USER_PRODUCT_PICK_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
VALID_FEATUREPRODUCT	VALID_FEATUREPROD_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
WELL	WELL_CSYS_FROM_EPSG	INSERTS OR UPDATES TO CORRECT SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME ON INSERT OR UPDATE OF EPSG_CODE
WELL	WELL_EPSG_RULES_CSYS	SPHEROID_NAME, DATUM_NAME, PROJECTION_NAME NOT ALLOWED INSERT OR UPDATE TO VALUES OTHER THAN THOSE THAT AGREE WITH THE EPSG_CODE
WELL	WELL_IN_AOI	INSERT OR UPDATE OF LATITUDE, LONGITUDE NOT ALLOWED IF VALUES ARE OUTSIDE THE LIMITS OF THE DEAL AREA OF INTEREST
WELL	WELL_OPDATES_LT_SYSDATE	SPUD_DATE,END_OF_OP_DATE MUST BE LESS THAN SYSDATE ON INSERT OR UPDATE
WELL	WELL_SEQNO_ONINSERT	INSERTS NEXT NUMBER IN WELLID_SEQNO SEQUENCE FOR WELLID VALUE
WELL	WELL_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
WELL_ALIAS	WELL_ALIAS_UPDINSDATE	DATE_ENTERED, DATE_UPDATED ENTERED AUTOMATICALLY ON INSERT OR UPDATE
WVS_COAST	SPCOL_DEL_CASCADE_18	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)
WVS_POLY_COAST	SPCOL_DEL_CASCADE_22	CASCADES DELETE OF FEATURE TO ROWS IN SDE TABLES STORING THE FEATURE GEOMETRY (N.B. SUFFIX NUMBER IN TRIGGER NAME IS SDE LAYER_ID AND IS LIABLE TO CHANGE)

Appendix 6 Data Model Diagrams

Logical and Oracle Implementation diagrams : Key



Logical entity

Entity proposed for later phases,
not yet implemented

Logical supertype entity

Relationships between entities A and B:

A must have one and only one B; B must have one and only one A

A must have one or more than one B; B must have one and only one A

A may have one and only one B; B must have one and only one A

A must have one or more than one B; B may have one and only one A

A may have one or more than one B; B must have one and only one A

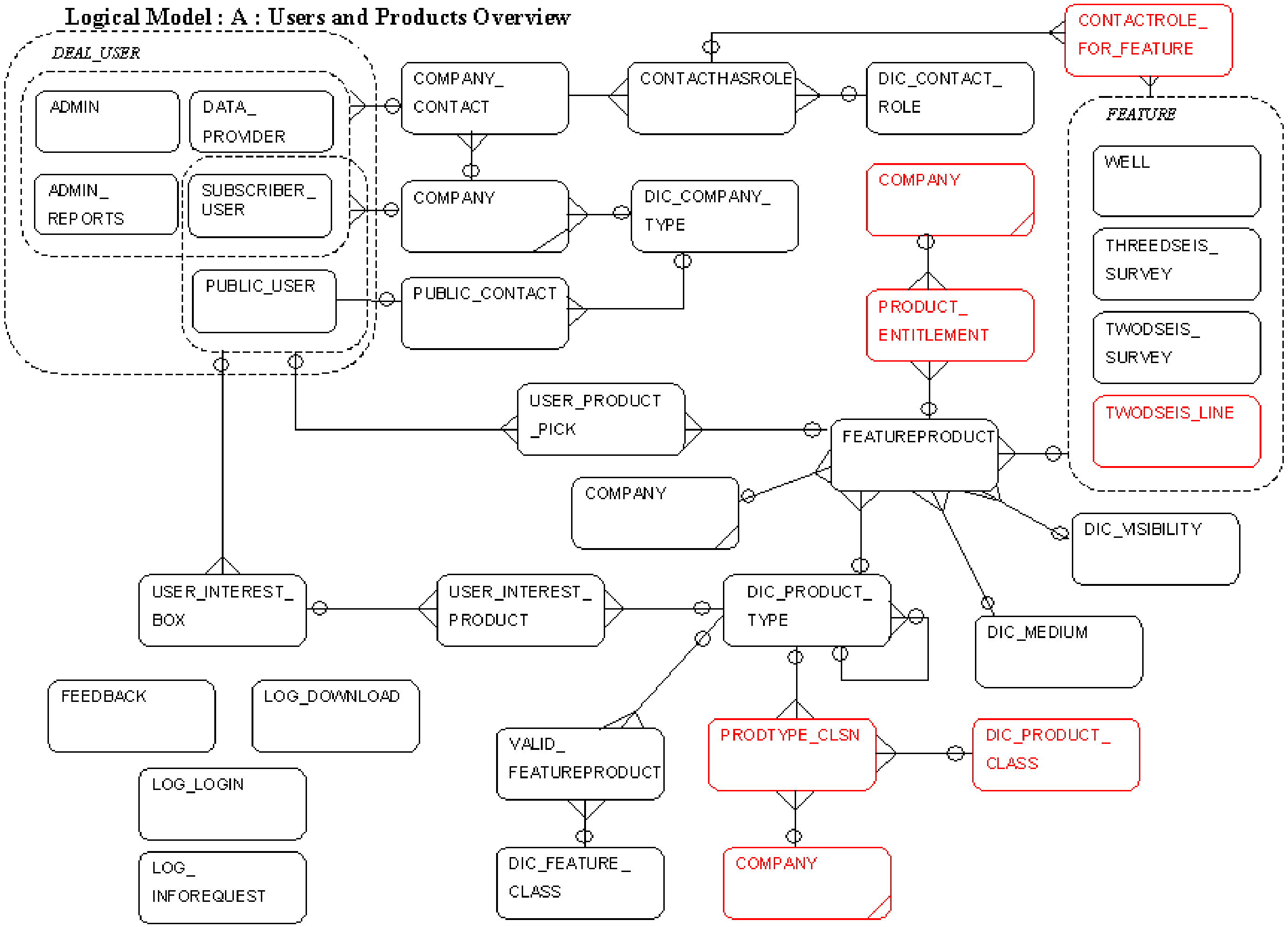
A must have one or more than one B; B must have one or more than one A

Table in Oracle implementation of logical model

Relationship constrained by a trigger rather than with a foreign key

1/31/01 DEAL data model

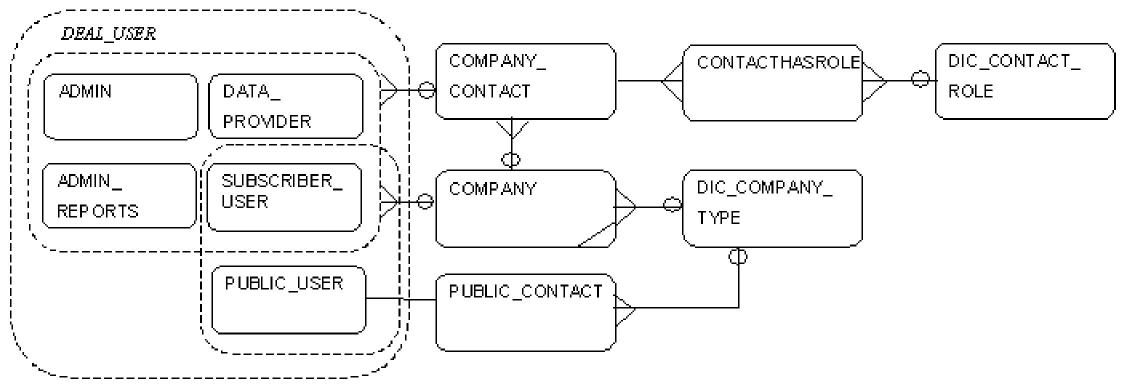
Logical Model : A : Users and Products Overview



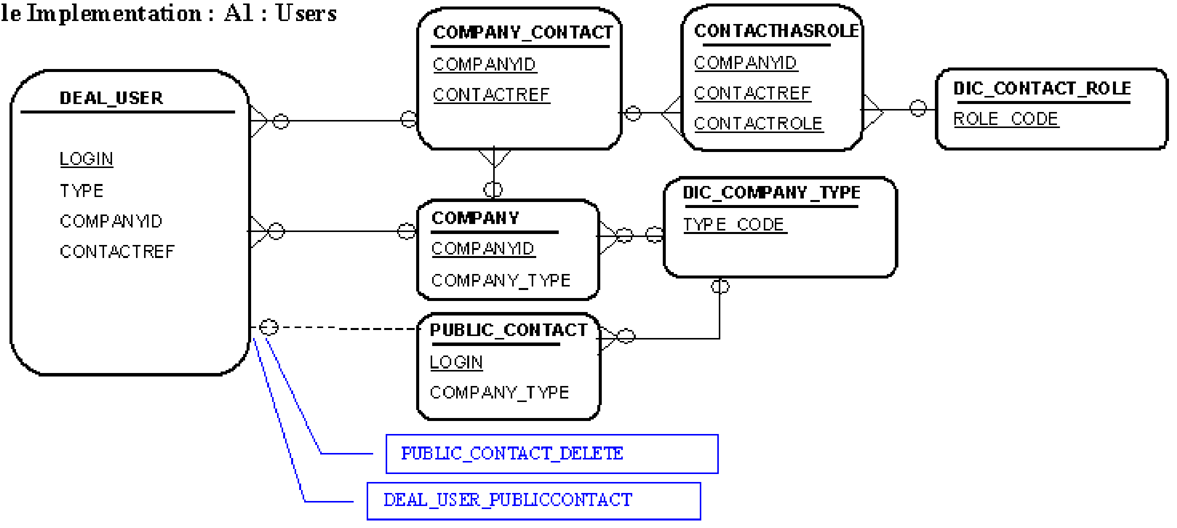
09/02/01

DEAL data model

Logical Model : A1 : Users

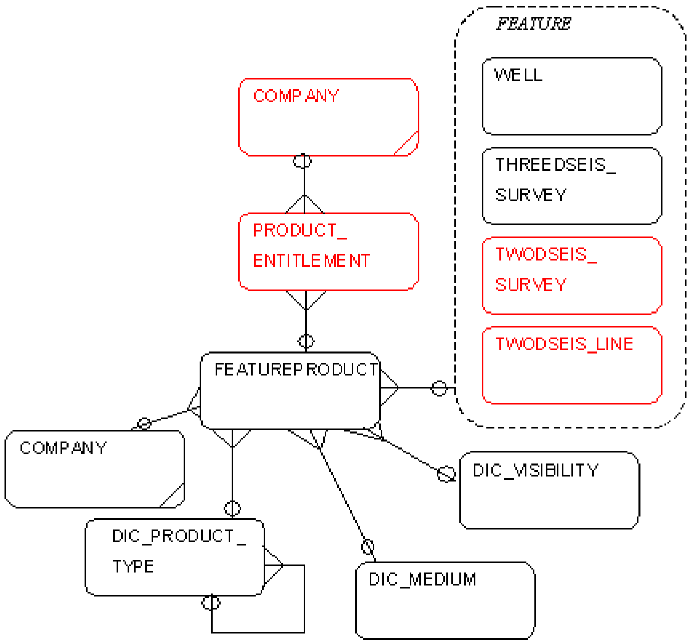


Oracle Implementation : A1 : Users

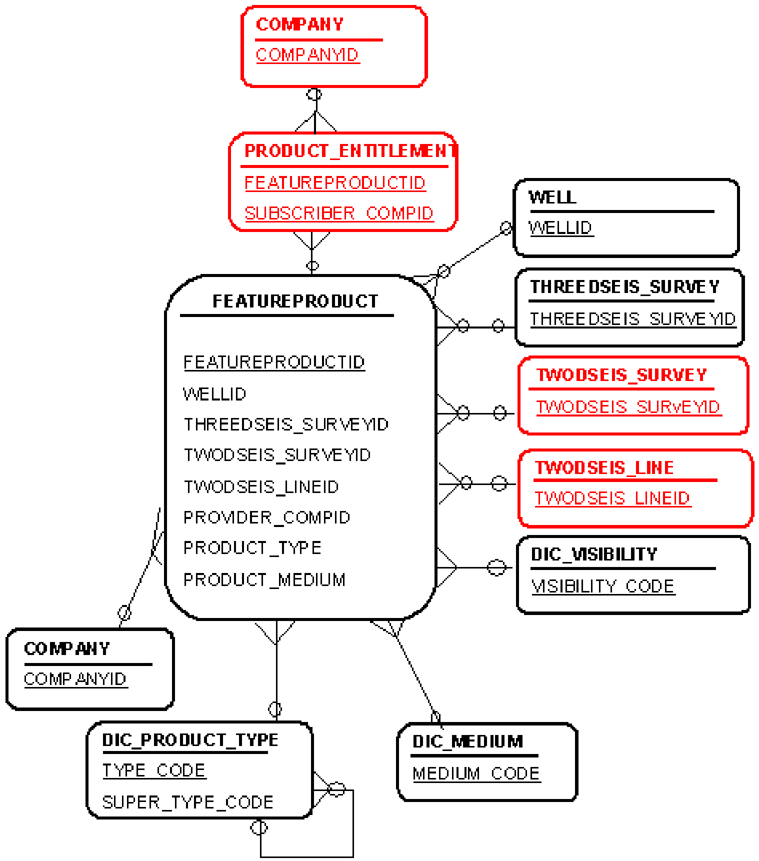


1/31/01 DEAL data model

Logical Model : A2: Products

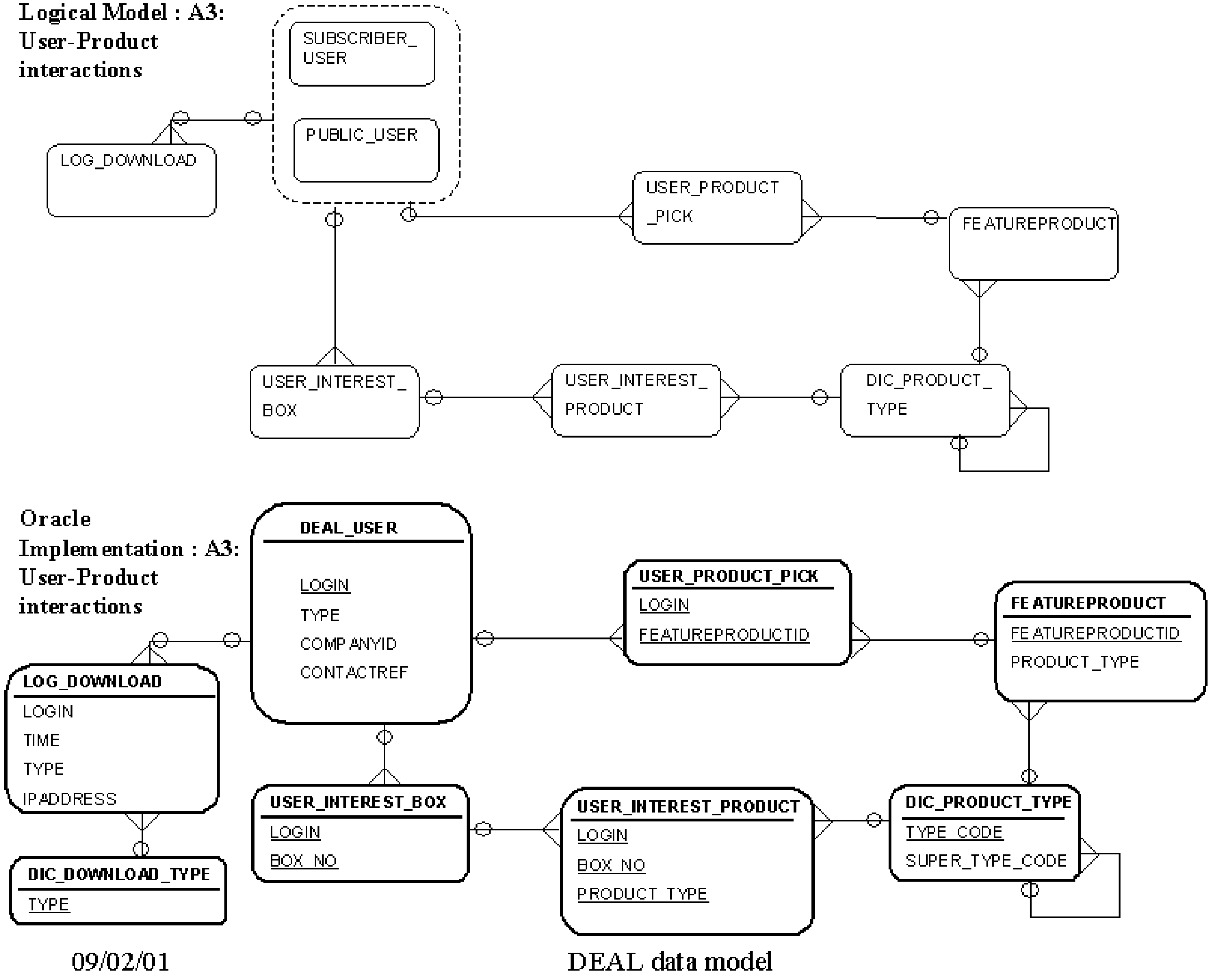


Oracle Implementation : A2: Products

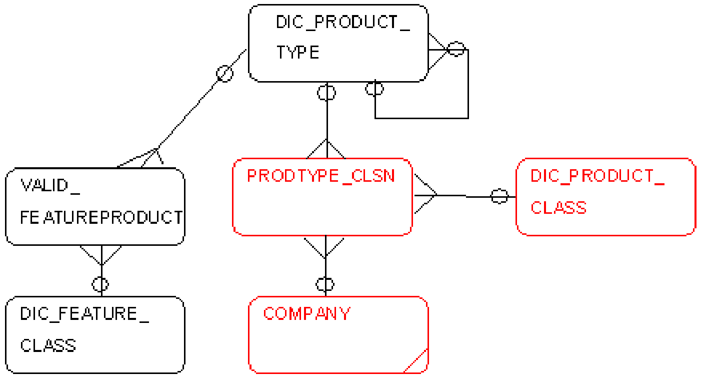


1/31/01

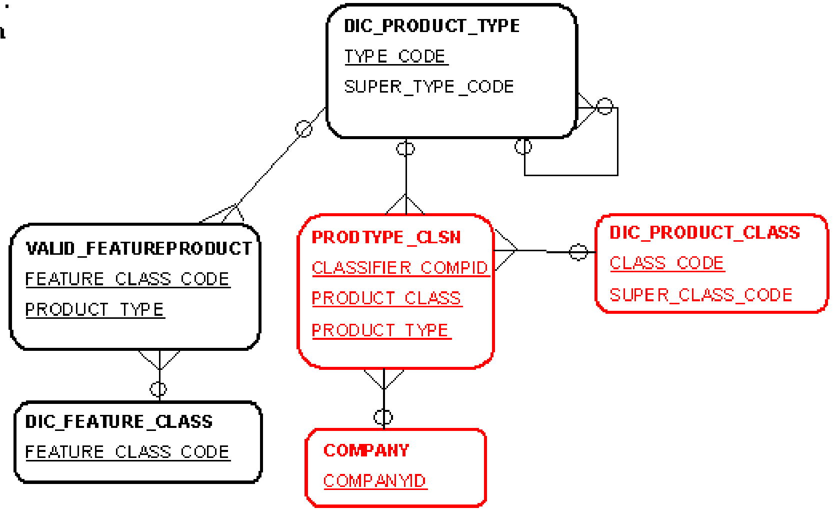
DEAL data model


Logical Model : A3:
User-Product
interactions
SUBSCRIBER_ USER
PUBLIC_USER
LOG_DOWNLOAD
USER_PRODUCT _PICK
FEATUREPRODUCT
USER_INTEREST_ BOX
USER_INTEREST_ PRODUCT
DIC_PRODUCT_ TYPE
Oracle
Implementation : A3:
User-Product
interactions
DEAL_USER
LOGIN
TYPE
COMPANYID
CONTACTREF
USER_PRODUCT_PICK
LOGIN
FEATUREPRODUCTID
FEATUREPRODUCT
FEATUREPRODUCTID
PRODUCT_TYPE
LOG_DOWNLOAD
LOGIN
TIME
TYPE
IPADDRESS
DIC_DOWNLOAD_TYPE
TYPE
USER_INTEREST_BOX
LOGIN
BOX_NO
USER_INTEREST_PRODUCT
LOGIN
BOX_NO
PRODUCT_TYPE
DIC_PRODUCT_TYPE
TYPE_CODE
SUPER_TYPE_CODE
09/02/01
DEAL data model

Logical Model : A4 :
Product classification

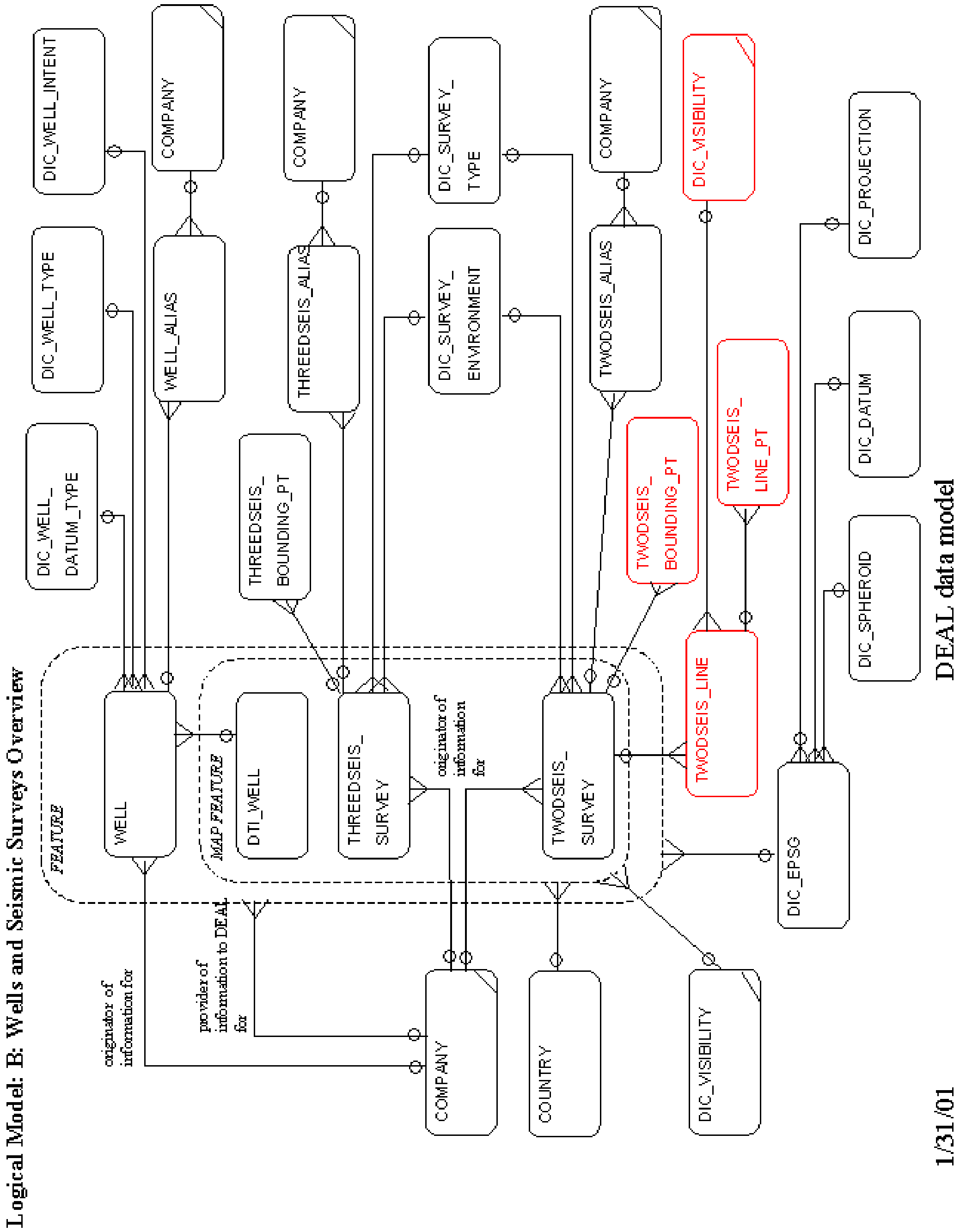


Oracle
Implementation : A4 :
Product classification



1/31/01

DEAL data model

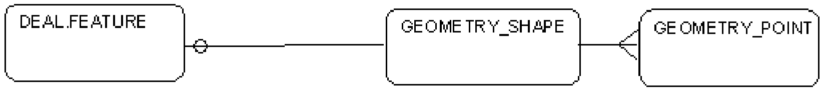


Logical Model: B: Wells and Seismic Surveys Overview

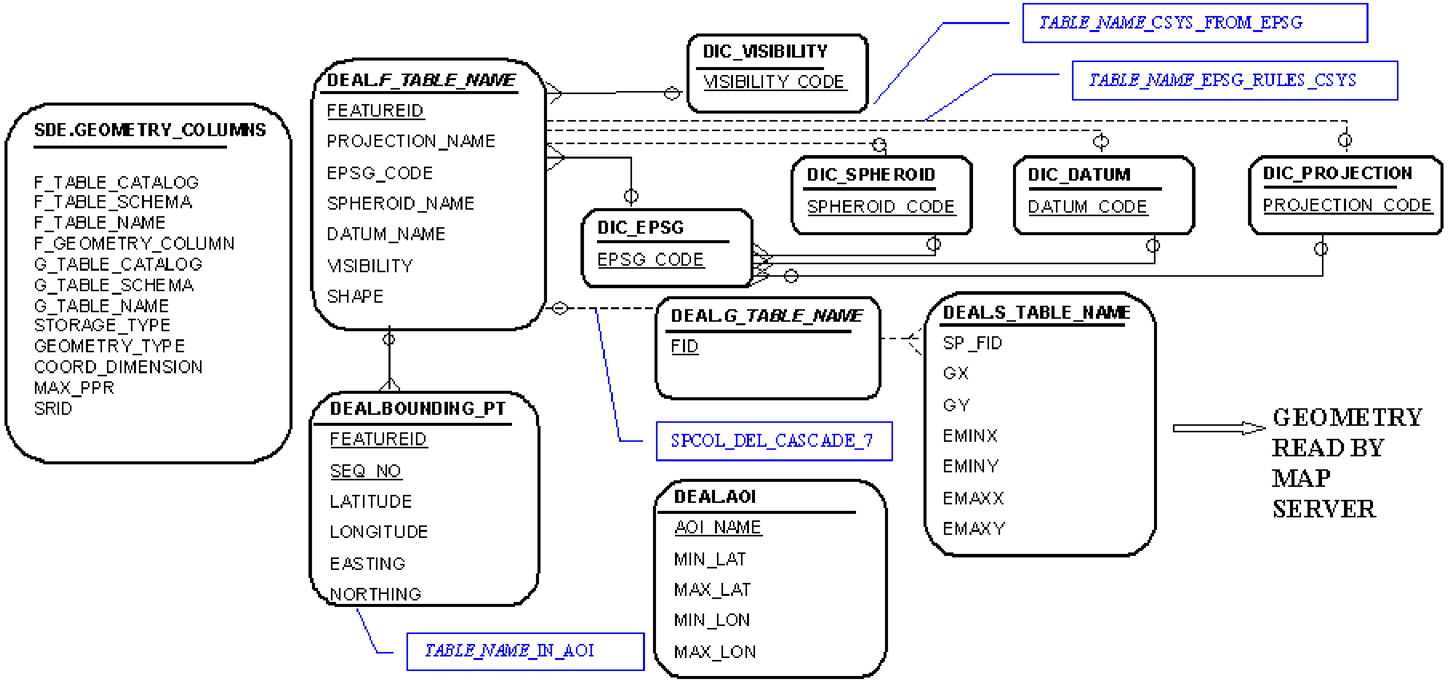
DEAL data model

1/31/01

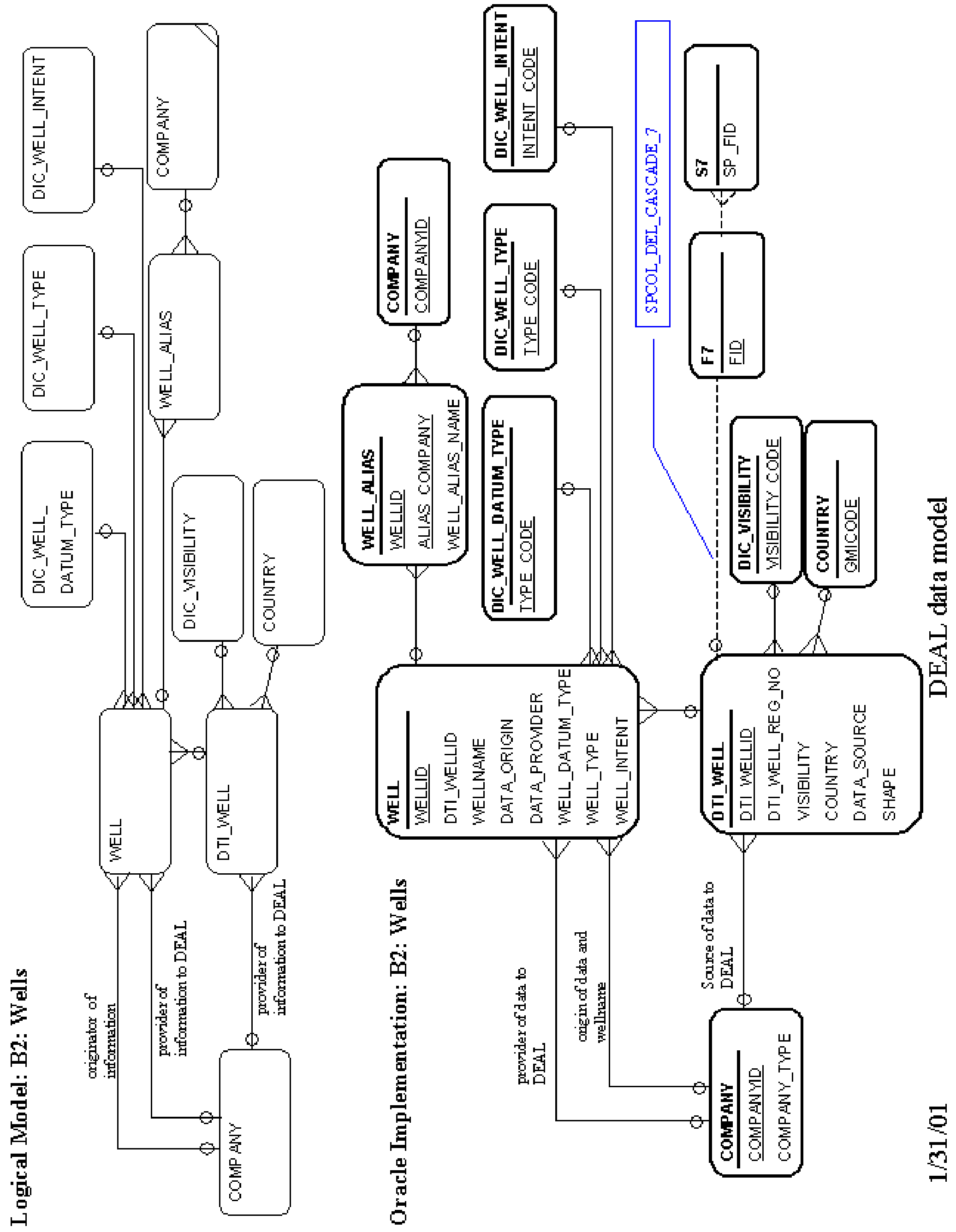
Logical Model: B1: Features with SDE geometry



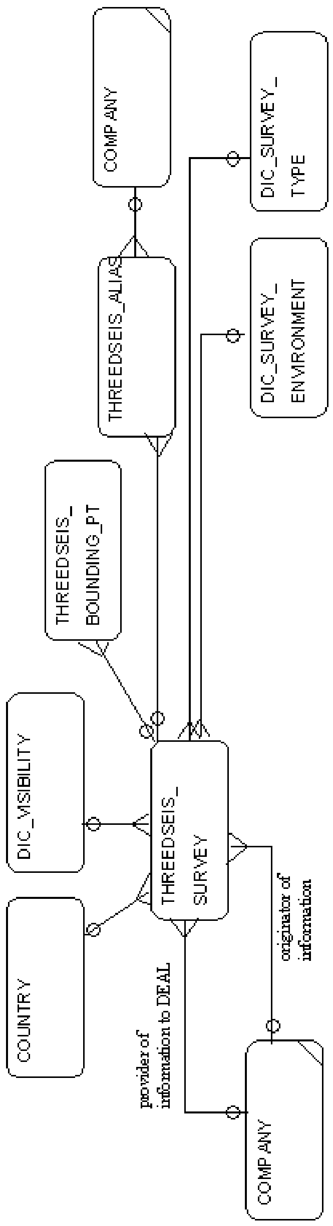
Oracle Implementation: B1: Features with SDE geometry



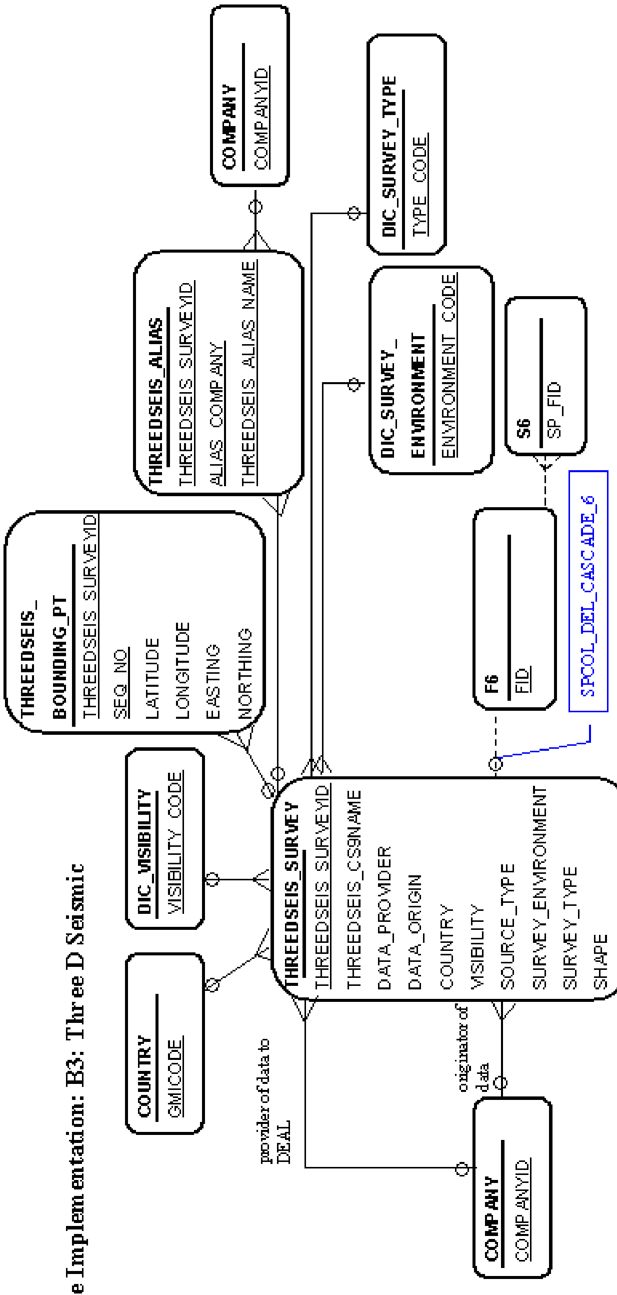
1/31/01 DEAL data model


Logical Model: B2: Wells
DIC_WELL_DATUM_TYPE
DIC_WELL_TYPE
DIC_WELL_INTENT
COMPANY
WELL_ALIAS
DIC_VISIBILITY
COUNTRY
WELL
DTI_WELL
COMPANY
originator of information
provider of information to DEAL
provider of information to DEAL
Oracle Implementation: B2: Wells
WELL_ALIAS
WELLID
ALIAS_COMPANY
WELL_ALIAS_NAME
COMPANY
COMPANYID
DIC_WELL_DATUM_TYPE
TYPE_CODE
DIC_WELL_TYPE
TYPE_CODE
DIC_WELL_INTENT
INTENT_CODE
SPCOL_DEL_CASCADE_7
S7
SP_FID
F7
FID
WELL
WELLID
DTI_WELLID
WELLNAME
DATA_ORIGIN
DATA_PROVIDER
WELL_DATUM_TYPE
WELL_TYPE
WELL_INTENT
DTI_WELL
DTI_WELLID
DTI_WELL_REG_NO
VISIBILITY
COUNTRY
DATA_SOURCE
SHAPE
DIC_VISIBILITY
VISIBILITY_CODE
COUNTRY
GMICODE
COMPANY
COMPANYID
COMPANY_TYPE
provider of data to DEAL
origin of data and wellname
Source of data to DEAL
DEAL data model
1/31/01

Logical Model: B3: Three D Seismic



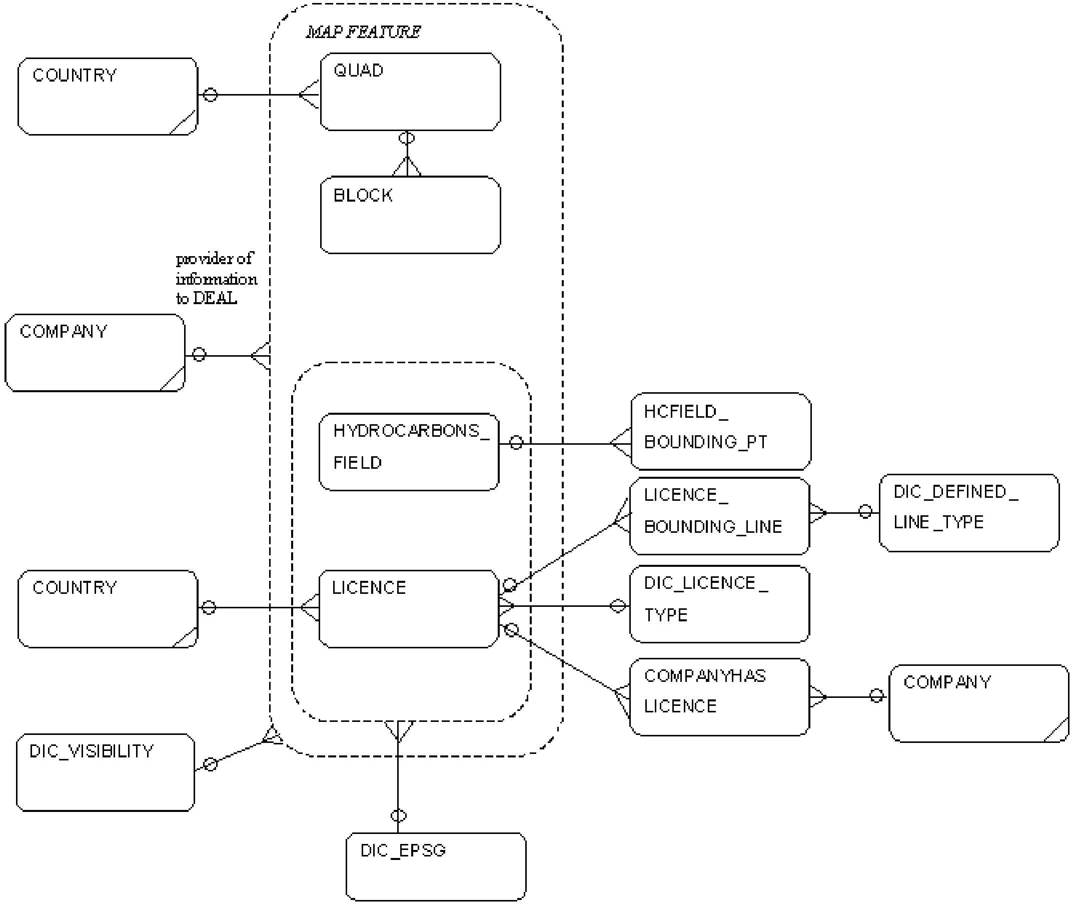
Oracle Implementation: B3: Three D Seismic



DEAL data model

1/31/01

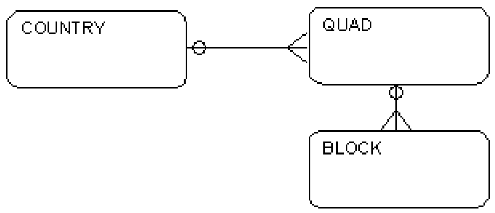
Logical Model: C: Quads, Licences, Fields Overview



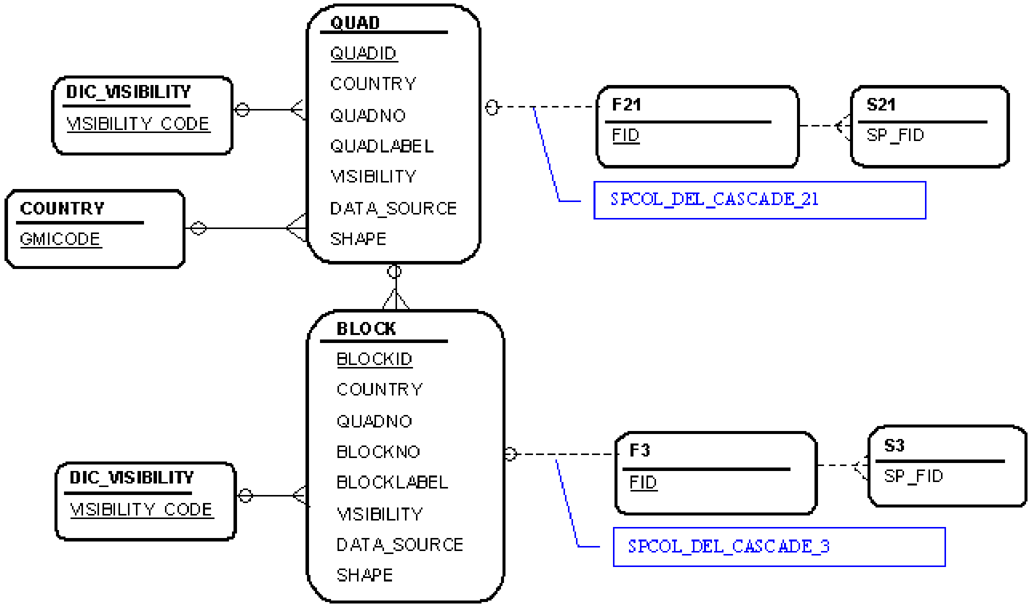
1/31/01

DEAL data model

Logical Model: C1: Quads and Blocks

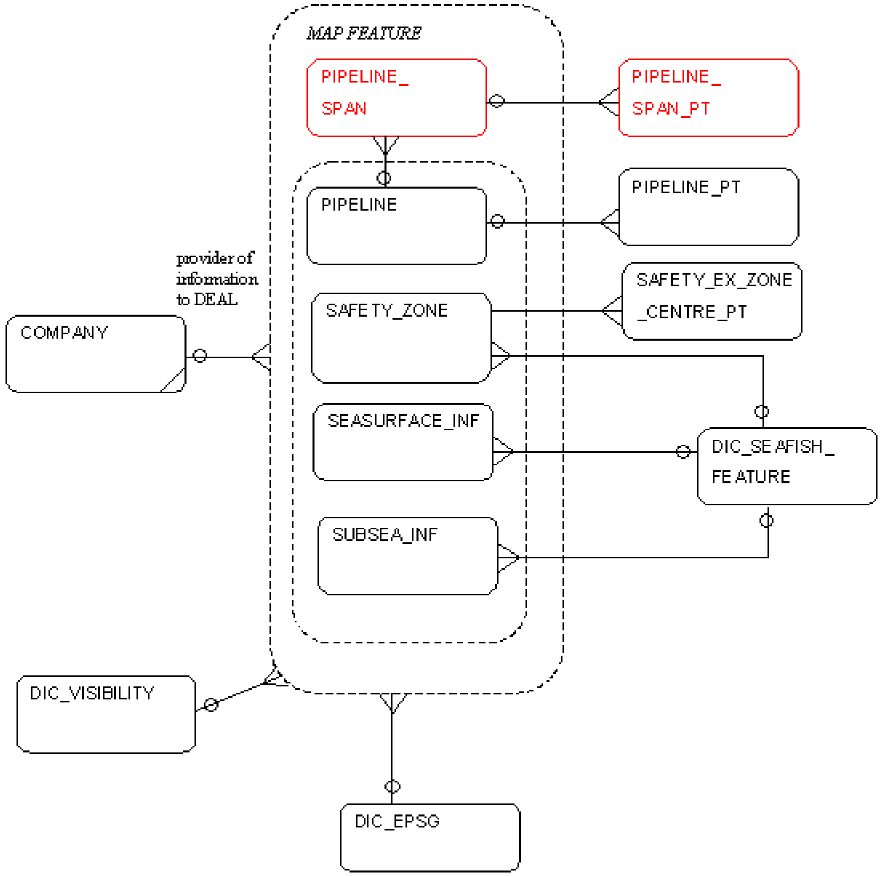


Oracle Implementation: C1: Quads and Blocks



1/31/01 DEAL data model

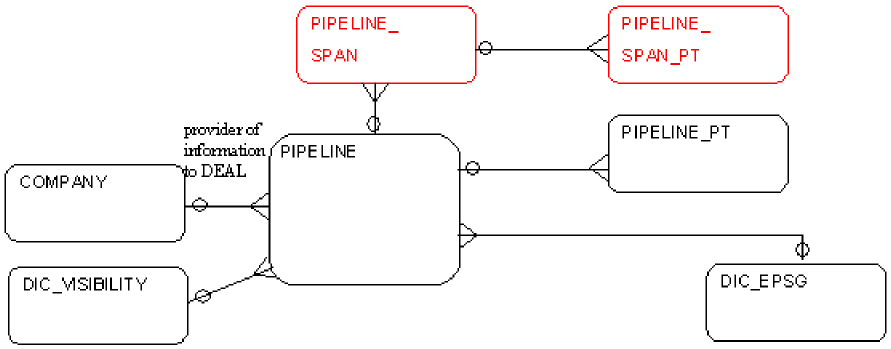
Logical Model: D: Infrastructure Overview



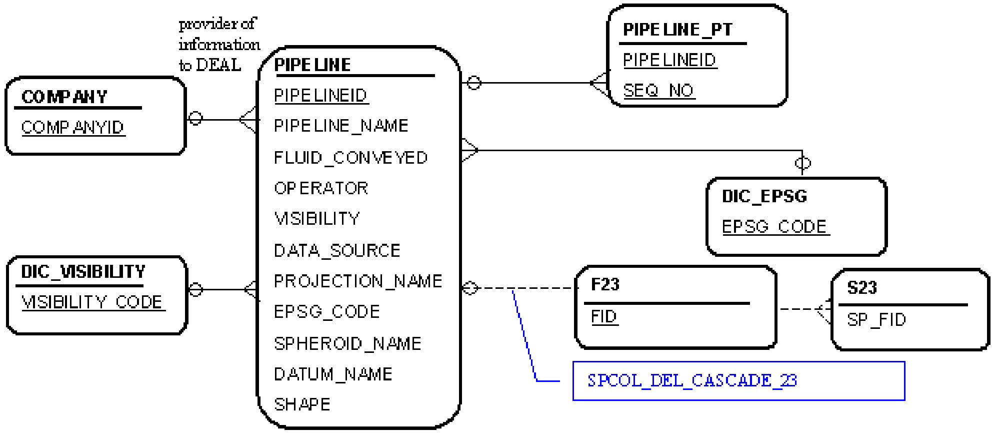
1/31/01

DEAL data model

Logical Model: D1: Pipelines



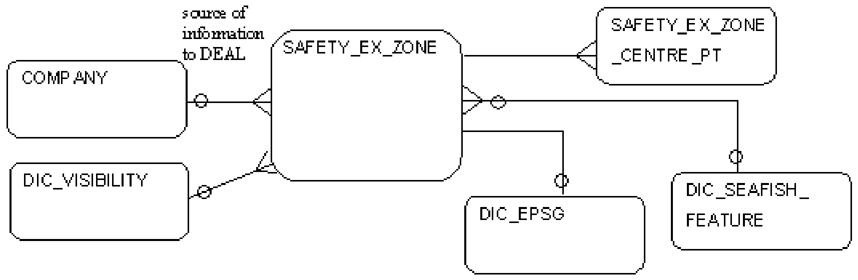
Oracle Implementation: D1: Pipelines



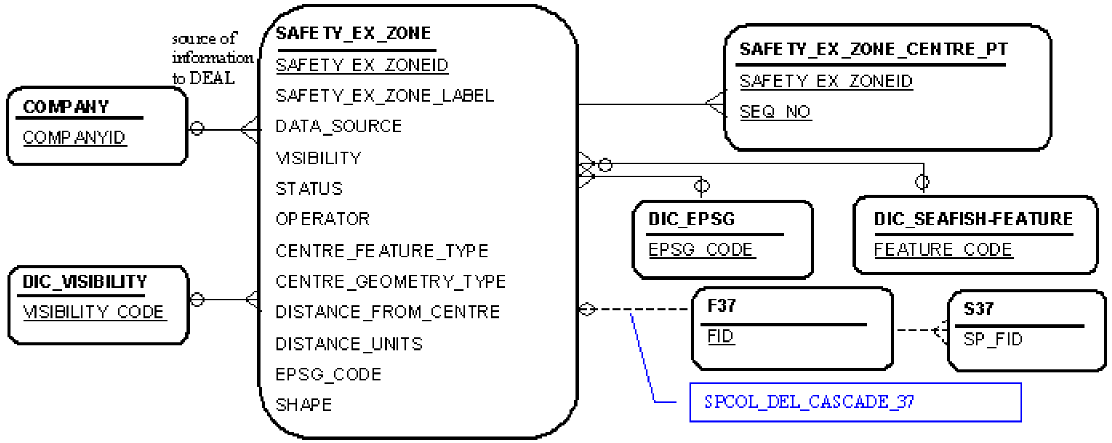
1/31/01

DEAL data model

Logical Model: D2: Safety/Exclusion Zones

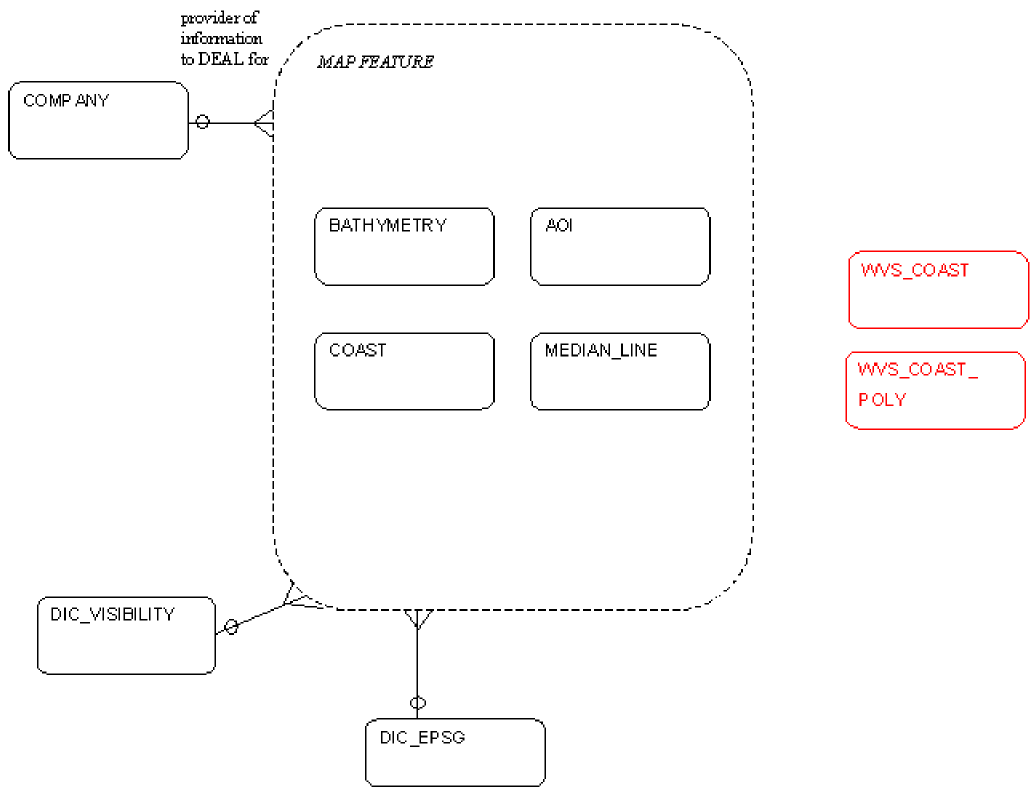


Oracle Implementation: D2: Safety/Exclusion Zones



1/31/01 DEAL data model

Logical Model: E: Other Cultural Features Overview



1/31/01

DEAL data model