



**British
Geological Survey**

NATURAL ENVIRONMENT RESEARCH COUNCIL

Rationalisation of the reports databases held on OLIB and IDA

National Geoscience Information Service

Internal Report IR/03/171

BRITISH GEOLOGICAL SURVEY

INTERNAL REPORT IR/03/171

Rationalisation of the reports databases held on OLIB and IDA

M. G. Alam

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

Report; key; words.

Front cover

Cover picture details, delete if no
cover picture.

Bibliographical reference

ALAM, M. G. 2003. Rationalisation
of the reports databases held in
OLIB and IDA. *British Geological
Survey Internal Report*, IR/03/171.
47pp.

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: bgs london@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

Contents

Contents.....	i
Summary	ii
1 Introduction.....	1
2 IDA application study	2
3 OLIB application study	2
3.1 OLIB Introduction	2
3.2 OLIB data structure.....	2
3.3 OLIB-IDA database comparison	3
3.4 Current database location.....	3
4 Design and development of the system.....	4
4.1 Design	4
4.1.1 Physical design.....	4
4.1.2 Logical design	5
4.1.3 System process.....	8
5 System pre-requisite and drawback	10
5.1 Pre-requisite	10
5.2 Drawback	11
Appendix 1 OLIB tables and views.....	12
Appendix 2 Tables having IDA data	23
Appendix 3 Package to extract OLIB data	25
Appendix 4 Temporary table	33
Appendix 5 Current IDA table.....	35
Appendix 6 Final materialised view	37
Appendix 7 Indexes for performance	40
Appendix 8 Error log table	41

Summary

This report describes the rationale behind the merger of the BGS reports databases that were held on the BGS OLIB and IDA systems. Detailed coding of the application development is included in the Appendices.

1 Introduction

The BGS Intranet Data Architecture (IDA) has evolved over several years and now provides BGS staff with a mature software environment for routine searching (browse) and data management functionalities that link to a wide range of BGS data. The IDA has a well-established application to enable staff to query the BGS reports database.

BGS Library has purchased commercial software called Oracle Library (OLIB) from Fretwell-Downing Informatics Limited (FDI) for their library system. One of the modules comprising OLIB stores BGS report publication data.

Both applications are using their own data repository. Two different repositories containing essentially the same data will increase the staff overhead to maintain the systems, an increased likelihood of human error and data unavailability. Some of the Reports data available via the IDA is unique; it is not available via OLIB. Similarly some of the data available via OLIB is not replicated in the IDA. Clearly this is a waste of resources, both capital and staff effort to maintain the two systems, as well as being inefficient means of access information.

BGS has long recognised the importance of data management and is committed to high standards of data availability. BGS has recognised that there currently exists duplicate publication reports data within two different repositories and accessed by two different entry points (IDA and OLIB).

A sub-project task to merge the two data repositories (IDA & OLIB) was set up as part of the BGS Electronic Dissemination of Information project. The aim was to establish an integrated data repository for digital BGS publication reports data and provide a single entry point to maintain and enhance this data. The single entry point would be the OLIB database. The IDA would then access the reports data held in OLIB. Thus providing BGS an opportunity to maintain the quality and availability at minimum cost.

2 IDA application study

The IDA (Intranet Data Access) system provides routine searching (browse) and data management functionality for a wide range of BGS data. The whole IDA application is based on a table called BGS_UNPUB_REPORTS (See Appendix 5). The IDA is only using 50 % of BGS_UNPUB_REPORTS columns. The rest of the columns do not contain any data and are primarily designed for future use.

3 OLIB application study

3.1 OLIB INTRODUCTION

OLIB is a Fretwell-Downing Informatics Limited (FDI) product based on Oracle, OLIB7 provides an effective solution for managing physical libraries with open extensions into networked digital libraries. Designed specifically for corporate, research academic and legal environments. BGS is using OLIB7.3 to manage their library data and information.

3.2 OLIB DATA STRUCTURE

OLIB7.3 data structure is quite complex. There is no documentation available to understand the structure and flow of the data. It has around 450 tables and views (see Appendix1) with no primary key and foreign key applied on the tables. OLIB is maintaining all the constraint (primary key and foreign key) with stored procedures (See fig.1).

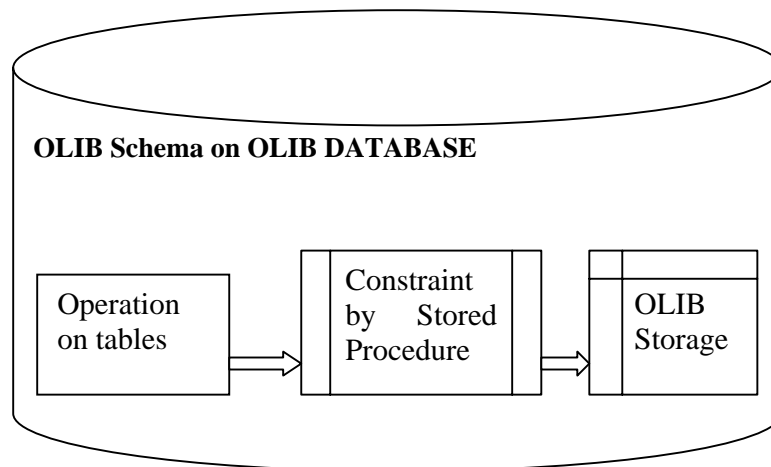


Figure 1

The result of OLIB design restricts BGS in its ability to reverse engineer the OLIB database and determine the flow of the system. Reverse engineering will give only a standalone table diagram, which is of little benefit. Since BGS cannot reverse engineer OLIB database and technical documentation is not available, the only realistic option left was to study each individual table and identify the relevant data from each of the tables. The prerequisite of the process design is to study both databases (OLIB and IDA) and compare them.

3.3 OLIB-IDA DATABASE COMPARISON

After studying OLIB and IDA database it was clear that only thirteen tables into OLIB database contained data relevant to the IDA. OLID data is not organised in such a way as to be of immediate applicability to the IDA and *vice versa*. Once extracted from the storage, it may be necessary to process them into a specific format. In some instances, this could involve considerable staff effort, e.g. OLIB.Titles.PRClassmark column has multiple IDA column information. Similarly, the IDA table column SUBJ_INDX has more than one subject index column in OLIB.

After identifying the relevant tables in both databases, it was necessary to determine the optimum method to connect the two databases and extract the relevant information from them.

3.4 CURRENT DATABASE LOCATION

The OLIB Server is currently in the de-militarised zone (DMZ) and the BGS corporate data server (KWA) is sited on the BGS LAN. Hence, the OLIB-IDA software application has to be able to access both servers simultaneously. BGS research data is commercially sensitive and BGS has a policy to protect the database. For security reasons, it was decided not to access the DMZ from BGS LAN (see fig 2).

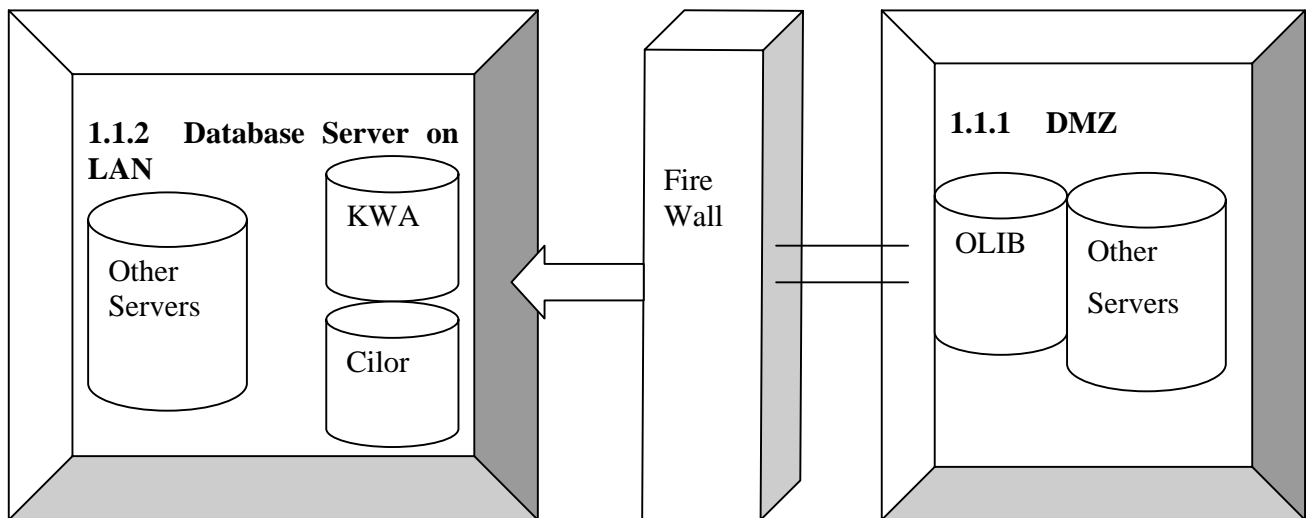


Figure 2

4 Design and development of the system

4.1 DESIGN

The way in which the system is designed to provide maximum flexibility for IDA applications. BGS_UNPUB_VOL, BGS_UNPUB_REPORTS & BGS_UNPUB_REPORTS_MV1 have the same field and data types ensuring that the application does not have to modify any code in the application with the exception of the table name.

The system is designed such that if all the data is available in the OLIB database, the IDA can drop its materialised view and directly access BGS_UNPUB_VOL. Again the column's name and data type is the same, therefore, the application is not required to change anything except the table name.

4.1.1 Physical design

4.1.1.1 LIST OF DATABASES

- OLIB
- KWA

4.1.1.2 LIST OF SCHEMA

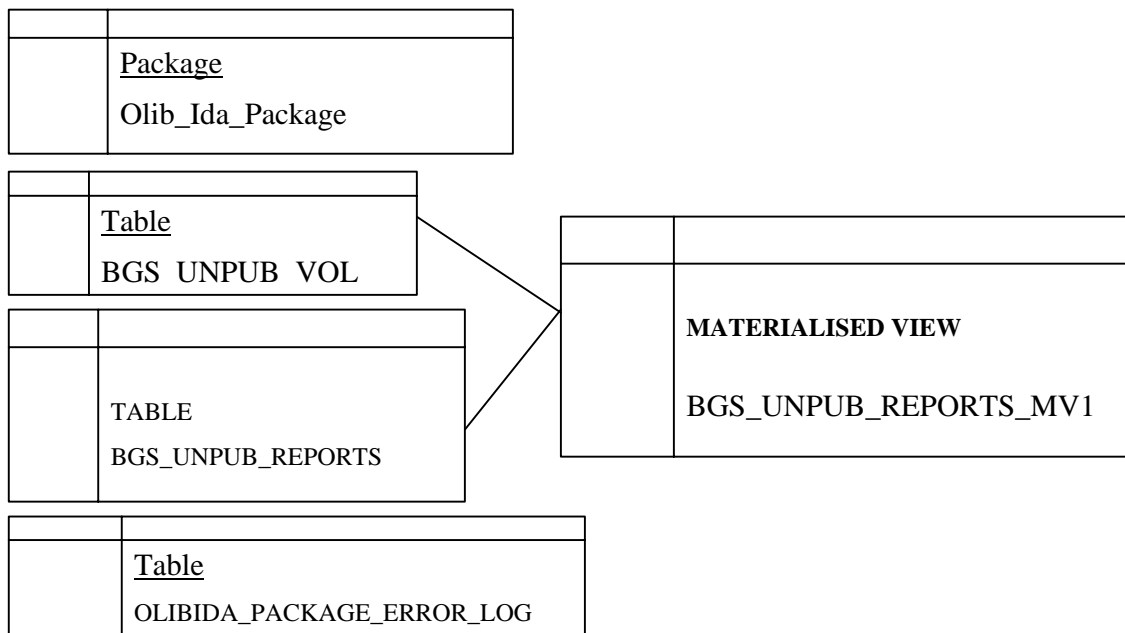
- OLIB.OLIB
- KWA.OLIB_VIEW
- KWA.BGS

4.1.1.3 LIST OF ENTITIES

- OLIB.OLIB – Refer to Appendix 1
- KWA.OLIB_VIEW - Refer to Appendix 2
- KWA.BGS see table below

Name	Description	Remarks
BGS_UNPUB_VOL	Table to store update OLIB data	This is a temporary table used to enhance the application performance
BGS_UNPUB_REPORTS	Old IDA table	
OLIBIDA_PACKAGE_ERROR_LOG	Table to store any error occurred during data transfer	
BGS_UNPUB_REPORTS_MV1	Materialised view that hold all data from BGS_UNPUB_VOL and non duplicate data from BGS_UNPUB_REPORTS	Oracle scheduler will update materialised view every day at 8:00 am

4.1.1.4 ENTITY RELATIONSHIP



4.1.1.5 LIST OF ENTITIES AND ASSOCIATED ATTRIBUTES

Entity Name	Attribute Details
BGS_UNPUB_VOL	See Appendix 4
BGS_UNPUB_REPORTS	See Appendix 5
OLIBIDA_PACKAGE_ERROR_LOG	See Appendix 3
BGS_UNPUB_REPORTS_MV1	See Appendix 6

4.1.2 Logical design

4.1.2.1 LIST OF PACKAGES

Name	Description
OLIB_IDA_PACKAGE	This package will filter all the required IDA data from OLIB, format it for IDA and then Insert it into the temporary table

4.1.2.2 SCHEDULE

Task	Time
OLIB data transfer	8:00 pm everyday
Data in BGS_UNPUB_VOL	00:00 everyday
FINAL DATA	8:00 am everyday

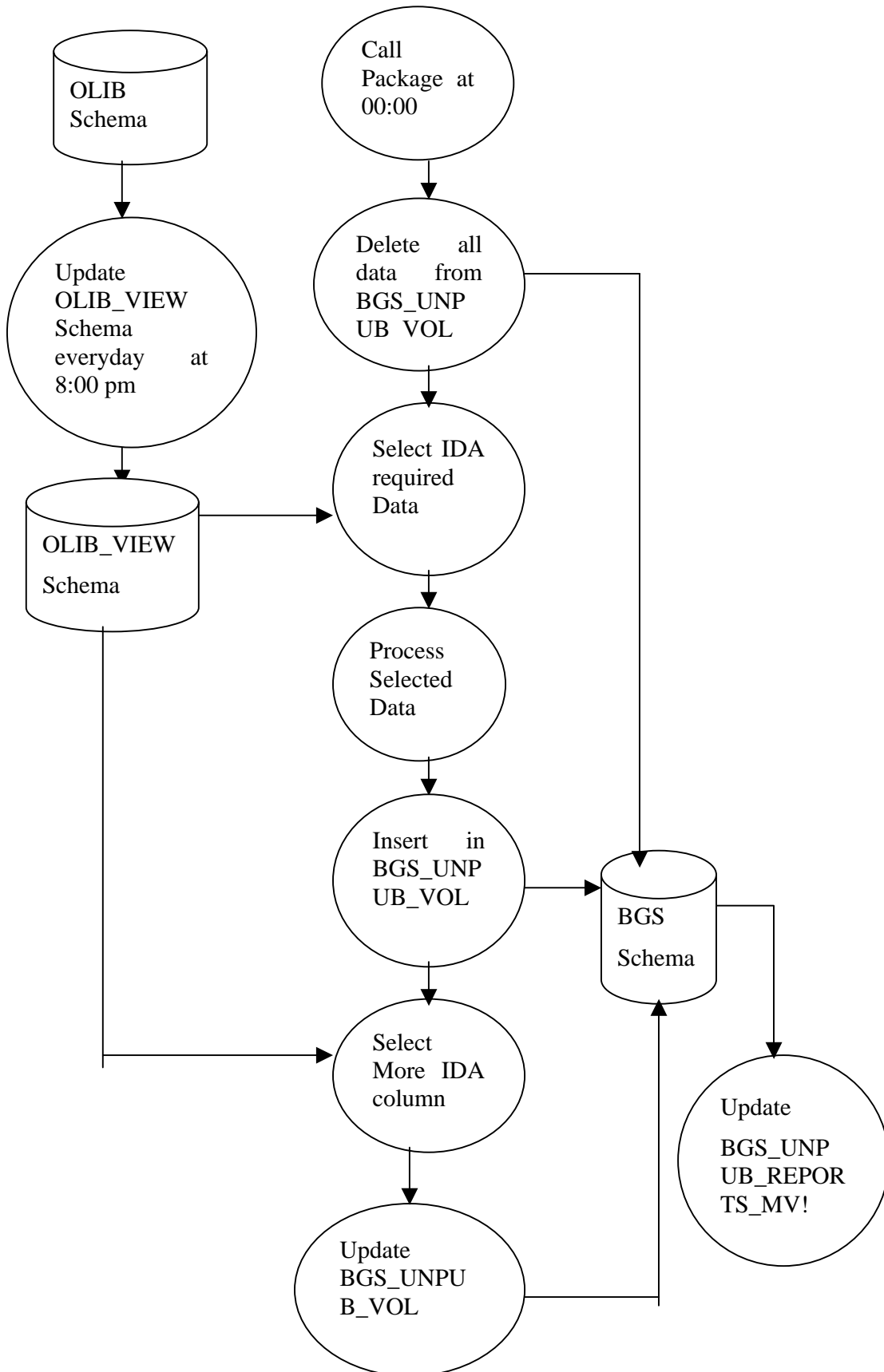
OLIB DATA TRANSFER: -- System transfers the required IDA tables from OLIB into BGS.OLIB_VIEW every day at 8:00 pm thus ensuring that the most up-to-date information residing on OLIB is available on OLIB_VIEW Schema.

DATA IN BGS_UNPUB_VOL: -The package OLIB_IDA_PACKAGE selects the required data from OLIB_VIEW Schema, processes it and then inserts it into the temporary table BGS_UNPUB_VOL everyday at 00:00.

FINAL DATA IN MATERIALIZED VIEW:-The system updates BGS_UNPUB_REPORTS _MV1 everyday at 8:00am with all data from temporary table BGS_UNPUB_VOL and any non-duplicate data from IDA table BGS_UNPUB_REPORTS.

Daily scheduling of the update process was decided due to OLIB not updating, inserting or deleting its reports publication data very frequently. More frequent updates can be scheduled should the need arise – e.g. should OLIB be updated more frequently.

4.1.2.3 SYSTEM FLOW DIAGRAM



4.1.3 System process

The system designed has no designated front-end, the application will run on the server automatically thus reducing the operational costs involved in maintaining the system and eliminating the potential for human error.

4.1.3.1 OLIB DATA ON KWA

OLIB database in the DMZ is not the part of LAN. BGS data are historic and commercially sensitive. Due to security reasons, the DMZ is not accessed from the LAN. Instead the 13 tables required are loaded into OLIB_VIEW Schema. OLIB_VIEW Schema updates the 13 tables daily at 8:00 pm. This is to ensure that all new up-to-date information is available in the BGS LAN server KWA (see figure3).

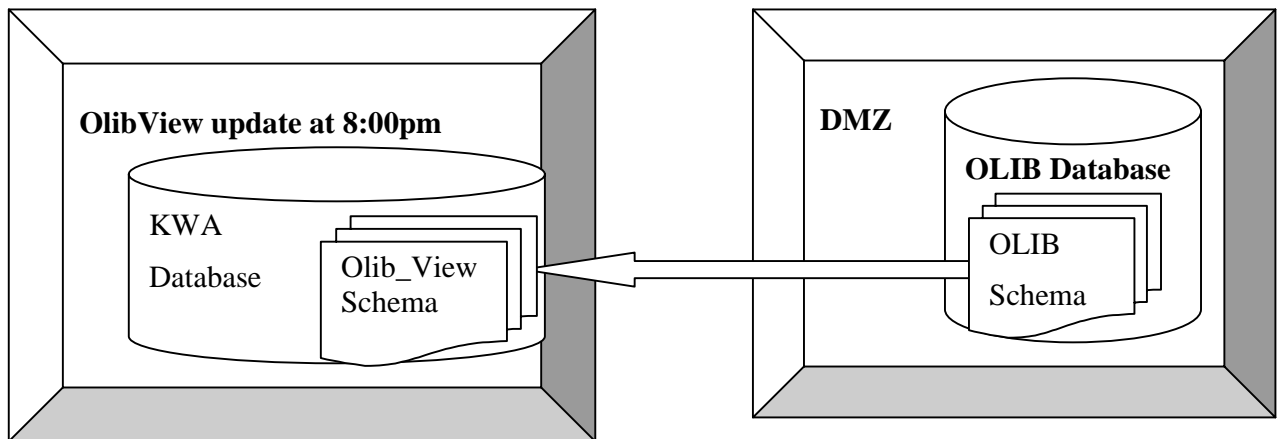


Figure 3

As OLIB_VIEW receives the latest OLIB data daily (at 8:00 pm), the IDA application will only reflect any changes to the data the following day, however, any changes to the reports database can be viewed immediately via the OLIB application. The next step is to find and process the OLIB data and insert it into the temporary table.

4.1.3.2 OLIB IDA PACKAGE

The system developed filters, processes and then inserts the required data into a temporary table. This event is scheduled to execute daily at 00:00. Since the required data is scattered into 13 different OLIB_VIEW tables with variable data formats. It is not a simple task to ensure the IDA-OLIB application receives the data in the same format that the system requires. Therefore, a package named OLIB_IDA_PACKAGE comprising two procedures OLIBDATA and OLIBDATA_UPDATE is called (see Appendix 3).

4.1.3.3 PROCEDURES

olibdata

OLIBDATA procedure selects all the data from several OLIB tables, processes the data retrieved to maintain the IDA format and then inserts it into the temporary table BGS_UNPUB_VOL.

Olibdata_update

OLIBDATA_UPDATE selects the rest of the columns from the other OLIB tables, processes the retrieved data and then updates the temporary table BGS_UNPUB_VOL. This is because OLIB has many columns in different record sets whereas the IDA application only has a single column for the equivalent data

4.1.3.4 IDA - OLIB DATABASE COLUMN MATCHING

The most difficult task was to determine the IDA required data into OLIB tables. Thanks go to Steve Prince, BGS library, who assisted in identifying the data locations in the OLIB tables and difference in formats between the two databases entries. OLIB data is not organised in such a way as to be of immediate benefit to the IDA and *vice versa*. Once extracted from the database, it may be necessary to process them into specific formats. Below is the one-to-one column mapping and the information required to process the columns between the IDA and the OLIB databases.

IDA TABLE (BGS_UNPUB_Reports) Column	OLIB Column	Comment
SERIES	TITLES.PRCLASSMARK	First character of TITLES.PRCLASSMARK
ACTIVITY	TITLES.PRCLASSMARK	Second Character of TITLES.PRCLASSMARK
TITLE	IF TITLES.COPYCAT='LTD1' THEN TITLE:='Title Confidential'; IF TITLES.COPYCAT='LTD2' THEN TITLE:='Title Confidential'; OTHERWISE TITLEENTRY:=TITLES.TITLE;	
AUTHORS	TITLES.STMNT_RESBTY	
MONTH	TITLES.CATDATE	
YEAR	TITLES.CATDATE	
ACCNO	TITLES.PRCLASSMARK	Number Between last '/' and last char of TITLES.PRCLASSMARK For example IR/03/160R ACCNO is 160
CONF	IF TITLES.COPYCAT='LTD1' THEN CONF='R' IF TITLES.COPYCAT='LTD2' THEN CONF='C'; OTHERWISE CONF=NULL;	Confidentiality of this record
INDCODE	TITLES.PRCLASSMARK	
COLLN	TITLEPHYSICAL.LENGTH	
SUBJ_INDXX	SUBJECTS.SUBJECT_DISP	OLIB has more than one record for subject index where as IDA has single column
COSTTXT	TITLES.MUSFORM,	
FREENGDC	TITLES.Notes,	

OLDCODE	TITLES.PRCLASSMARK	Middle information of classmark
USER_ENTERED		
DATE_ENTERED		
CONF_TITLE	IF TITLES.COPYCAT='LTD1' THEN TITLES.TITLE; IF TITLES.COPYCAT='LTD2' THEN TITLES.TITLE; OTHERWISE NULL	
SECOND_ENTRY	0	No second_entry data available in OLIB. Since this is concatenated IDA primary key column. Therefore passing 0 to make it working
SWN	TITLES.LL_NORTHING	
NEN	TITLES.UR_NORTHING	
NEE	TITLES.UR_EASTING	
SWE	TITLES.LL_EASTING	
SWC	TITLES.LL_SHEET	
NEC	TITLES.UR_SHEET	
YEAR_RELEASE	TITLES.PUBDATE	
NOTES	TITLES.NOTES	
FILE_URL	OBJECTS.LOCATOR	

4.1.3.5 ERROR LOG

The system stores any error generated during the application process into a table called OLIBIDA_PACKAGE_ERROR_LOG. The server administrator can view these errors.

4.1.3.6 CREATING MATERIALISED VIEW

After the up-to-date data is available in the temporary table a batch routine executes daily at 8:00 am. The system creates a materialized view called BGS_UNPUB_REPORTS_MV1 to hold all the data from the temporary table (BGS_UNPUB_VOL) and also non duplicate data from the original IDA table (BGS_UNPUB_REPORTS.IDA). This view can be deleted once all the IDA data is available in OLIB.

5 System pre-requisite and drawback

5.1 PRE-REQUISITE

- When Fretwell-Downing Informatics Limited (FDI) updates the OLIB database with the records that exclusively appear in the IDA, a new review will be required.
- Maintain the same table column for data entry as of present

- Maintain same class mark column structure will as of present

5.2 DRAWBACK

Because of the system design, any OLIB data update will only be reflected in IDA applications the following day.

Appendix 1 OLIB tables and views

ACCINSTS_BORROWERS_COM
ACCINSTS_BORROWERS_INV
ACCOUNTS
ACCOUNT_EVENTS
ACCOUNT_INSTANCES
ACQ_FILM_STATUS
ACTIVITY_AUDIT
AC_SERVICES
ADDRCATS
ADDRESSES
ADDRLINKS
ALL_LOCATION_MEMBERS (view)
APP_FLDS
ARCHIVE_ILL_PENDING
ARCH_COPYRES
ARCH_INVOICES
ARCH_INVOICE_ITEMS
ARCH_ORDERS
ARCH_ORDER_ITEMS
AREAS
ATPS
AUTHENTICATION_DRIVERS
AUTHENTICATION_SERVICES
AUTHENTICATION_SERVICE_PARAMS
AUTHORS
AUTHORTPS
AUTH_SERVICE_PARAM_LINKS
BINDTPS
BOOKING_SLOTS
BOOKING_SLOTS_SEQ
BORAUDIT
BORAUDIT_TRAIL
BORCATLOAN
BORCATS

BORKYWORDS
BOROBS
BORROWERS
BORROWER_ADMIN_BUREAUS
BORROWER_COURSES
BORROWER_DEPTS
BORROWER_FOLDERS
BORROWER_PRESIGNED_COPYRIGHT
BORROWER_RAPID
BORTPS
BTRANTPS
CARTCODETPS
CATALOG_STATS
CIRCLISTS
CIRCLISTTITLES
CIRCLIST_MEMBERS
CIRCLIST_ORDERITEMS
CIRC_STATS
CIRC_STAT_DEFS
CIRC_STAT_DOMAINS
CITATION_PATTERNS
CLAIM_EVENTS
CLAIM_EVENT_TPS
CLAIM_HEADERS
CLAIM_SEQS
CLAIM_STEPS
CLASSES
CLASSSUBJS
CLASSTPS
COLLNS
CONTEXT_PENDING_UPDATE
CONTROLNOTPS
COPIES
COPYCATS
COPYDEL
COPYRES
COPYSTS
COPYTRANS

COUNTRIES
COURSES
CREDIT_DETAILS (view)
CTRANTPS
CURRENCIES
DATABASE_SERVER_ACCOUNT
DEBIT_DETAILS (view)
DELIVERY_METHODS
DEPTS
DISTRICTS
DOC_DISPOSAL_CONDITIONS
DUBLIN_CORE (view)
EAD_LINES (view)
ENQOBS
ENQPRIORITYTPS
ENQRECEIPTMETHODS
ENQUIRIES
ENQUIRYACTIONS
ENQUIRYSTS
ENQUIRYSUBJECTS
ENQUIRYTITLES
ENQUIRYTPS
ENQUIRYUSERTPS
EVENT_TYPES
EXPLAIN_VIEW (view)
EXPORT_FORMATS
EXTRACTSTS
FILMS
FILM_ITEMS
FINANCIAL_PERIODS
FINESEQS
FINESTEPS
FINETPS
FIXED_ADDR_DOMS
FORMATS
FREQUENCIES
GEO_CODES
GEO_COLLECTIONS

HOLIDAYS
IC_CIRCLISTMEMBERS
ILLS
ILL_ACTIONS
ILL_ACTION_AUTO_AUTHSTATUS
ILL_ACTION_AUTO_FORWARD
ILL_ADDRESS_MAP_LIST
ILL_ADDRESS_MAP_LIST_ITEM
ILL_ADDRESS_TYPE_LIST
ILL_ADDRESS_TYPE_LIST_ITEM
ILL_ADD_DEFAULTS_LIST
ILL_ADD_DEFAULTS_LIST_ITEM
ILL_ALERTS
ILL_ALERT_ACTIONS
ILL_ALERT_COPYRIGHTS
ILL_ALERT_LINKS
ILL_ALERT_NOTIFICATIONS
ILL_ANSWER_REASONS
ILL_ART_REPLY_MESSAGES
ILL_ART_REQUEST_TEXT
ILL_ART_RESPONSE
ILL_ART_RESPONSE_OLD
ILL_ATTR_LIST
ILL_ATTR_LIST_ATTRIBUTES
ILL_AUTH_STATUS
ILL_AUTORESP_DATABASE_LINKS
ILL_AUTORESP_SEARCH_LINKS
ILL_AUTO_BROKER_CONFIG
ILL_AUTO_BROKER_MESSAGES
ILL_AUTO_RESPONDER_CONFIG
ILL_AUTO_RESPONDER_LINKS
ILL_BL_FORM_NUMBERS
ILL_CONTROL_NO_DEF_LIST
ILL_CONTROL_NO_DEF_LIST_ITEM
ILL_CONTROL_NUMBERS
ILL_COPYRIGHT_ALERTS
ILL_COPYRIGHT_CLIENT_TEXT
ILL_COPYRIGHT_POLICY

ILL_COPYRIGHT_POLICY_HEADER
ILL_COPYRIGHT_TYPES
ILL_COPYR_HEAD_POLICY_LINKS
ILL_DEFAULT_ACCOUNTS
ILL_DEFAULT_BILLING_METHODS
ILL_DEFAULT_COPYR_NOTIFY_TEXT
ILL_DEFAULT_DATABASES
ILL_DEFAULT_LINKS
ILL_DEFAULT_MATERIAL_AUTH
ILL_DEFAULT_SEARCHES
ILL_DEFAULT_SERVICE_LEVELS
ILL_DEFAULT_WORK
ILL_DEF_COPYRIGHT_PERMISSIONS
ILL_DEF_COPYRIGHT_SETTINGS
ILL_DEF_COPYRIGHT_SUB_TYPES
ILL_DELIVERYMETHODS
ILL_DISALLOWED_VAL
ILL_ERRORS
ILL_HELD_LOCALLY_LIST
ILL_HELD_LOCALLY_LIST_ITEM
ILL_ILLSTATUS_FOR_ALERTS
ILL_ISO_PDU_FILES
ILL_LOCATION_LIST
ILL_LOCATION_LIST_LOC
ILL_LOCS
ILL_LOC_COPYRIGHTS
ILL_LOG_MESSAGES
ILL_MAIL_FORMAT
ILL_MATERIAL_TYPE
ILL_MEDIUMS
ILL_MESSAGES
ILL_NAMING_AUTHORITIES
ILL_NOTES
ILL_NOTETPS
ILL_OBJECTS
ILL_PDUS
ILL_PROCESS_STATUS
ILL_PROFILE

ILL_PROFILE_ATTRIBUTE
ILL_PROFILE_ATTRIBUTE_MAP
ILL_PROFILE_MAP_VALUE
ILL_PROTOCOLS
ILL_PROTOCOL_CHOICE
ILL_REJECTION_NOTES
ILL_REQUESTED_ITEMS
ILL_ROTA
ILL_ROTA_ITEM
ILL_ROTA_SORT_LIST
ILL_ROTA_SORT_LIST_ITEM
ILL_SEARCH (view)
ILL_SEARCHES
ILL_SEARCHES_ATTRIBUTES
ILL_SERVICETPS
ILL_SERVICE_LEVEL_DEFAULTS
ILL_SIP_PROFILE
ILL_SIP_PROFILE_FIELDS
ILL_STATE_TABLE
ILL_STATE_TABLE_LINKS
ILL_STATE_TRANSITIONS
ILL_STATUS
ILL_TRANSITIONS
ILL_TRANSSTORE_DATA
ILL_USER_ACCOUNTS
ILL_USER_ALERTS
ILL_USER_LOCATION_DEFAULTS
ILL_USER_PAYMENTTPS
ILL_VALID_SERVICE_TYPES
ILL_WORK_SORT
ILL_Z_REQUEST_CONV
ILL_Z_REQUEST_CONV_COPY
ILL_Z_REQUEST_CONV_VALUES
INSTS
INVOICEITEM_EVENTS
INVOICES
INVOICE_ITEMS
IRR_FREQUENCIES

ISNTPS
ISSCOPIES
ISSUES
ISS_LOC_COPIES
ITEM_MVTYPES
ITEM_TYPES
LANGS
LCSH_TAGS (view)
LINKTPS
LOANSTS
LOANTERMS
LOCATION_GROUP_MEMBERS
LOCATION_GROUP_RELATIONSHIP
LOCS
LOCTPS
LOC_ARIEL_WORKSTATIONS
LOC_AUTH_SERVICE_PARAM_LINKS
LOC_ILL_ACCOUNTS
LOC_ILL_BILLING_METHODS
LOC_ILL_DELIVERY_METHODS
LOC_ILL_MAIL_ADDRESSES
LOC_ILL_MANDATORY_FIELDS
LOC_ILL_PAYMENT_METHODS
LOC_ILL_PROTOCOLS
LOC_ILL_PROTO_MAT_MAIL_FORMAT
LOC_ILL_REQ_RESP_DETAILS
LOC_ILL_SERVICES
LOC_ILL_SERVICE_ATTRIBUTE
LOC_ILL_SERVICE_ATTRIB_DETAILS
LOC_ILL_SERVICE_HEADER
LOC_ILL_SERVICE_LEVELS
LOC_ILL_STATISTICS
LOC_ILL_SYMBOLS
MARC_EXPORT_REC
MARC_INDICATORS
MARC_INDICATOR_VALUE
MARC_SUBFIELDS
MARC_TAGS

MEDIA
MEDIATPS
MODULES
MODULES_TABLES
MODULES_WORK
MODULE_DEPENDENCIES
MODULE_VERSION
MUSFORMS
NORMARC_EXPORT_TAGS (view)
NORMARC_TAGS (view)
NORMRULES
NOTETPS
NOTICES
OBJECTS
OBJECTSTS
OBJECTTPS
OBJECT_AUDIT
OBJECT_ROWS
OCCUPS
ODUE_AUDIT
ODUE_SEQS
ODUE_STEPS
OI_EDI_MESSAGES
OLIB_REGISTRY
OLSTF_EXPORT_TAGS (view)
OPAC_DISPLAYLIST
OPAC_HITLIST
OPAC_STOPLIST
ORACLE_SERVER_DETAILS
ORDERITEM_EVENTS
ORDERS
ORDER_ITEMS
ORDER_ITEMS_ORDEREDFOR
ORDER_STATUS
ORGTPS
OTHERCOSTS_EVENTS
PICFORMS
POSTAL_LOCS

PRLOANS
PROBLEM_LOG
PUBPATS
RECSRCS
RECSTS
RENEWTRANS
REQU
REQUTPS
REQ_AUTHORS_ALL (view)
REQ_TITLES_ALL (view)
RES
RESBTYTPS
RESPONDER_CALENDAR
RESPONDER_HOLIDAYS
RETURNDATES (view)
ROTA_SELECTION_CRITERIA
SALESITEM_EVENTS
SEQNOS
SERCOPIES
SERCOPY_ORDERITEMS
SERCOPY_TRIGGERTPS
SERIES
SERIESTPS
SERIES_RAPID
SERSTS
SERVICES
SGML_LINES (view)
SHELVES
SIPV1_AUTHENTICATION
SIP_SERVER_DETAILS
SIP_SERVER_DETAILS_FIELDS
SPCCITPATS
SPCPUBPATS
SUBJECTS
SUBJECTSUBJECTS
SUBJECTS_ALL (view)
SUBJECTTPS
SUBJECT_SUBDIVS

SUBSTPS
SUPMEDIATPS
SUPPLIERS
SUPPLIER_RAPID
SUPSERVS
SUPSTS
SUPTPS
SYSGLOBS
TEMP_RESPONDER_HOLIDAYS
TIME_PERIODS
TITLEAUTHS
TITLECARTCODES
TITLECARTDATA
TITLECLASSES
TITLECONTROLNOS
TITLEISNS
TITLELANGS
TITLENOTES
TITLEOBS
TITLEPHYSICAL
TITLEPREGEN
TITLEPRICES
TITLEPUB
TITLES
TITLESERIES
TITLESUBJS
TITLES_ALL (view)
TITLETITLES
TITLETPS
TITLEXREFS
TITLE_RAPID
TRANMETHODS
TRAPLIST
TRAPS
UKMARC_EXPORT_TAGS (view)
UKMARC_TAGS (view)
USER_AVAILABLE_SERVER
USMARC_EXPORT_TAGS (view)

USMARC_TAGS (view)
VATRATES
VDX_CONVERSION_NOTE
VDX_CONVERSION_NOTE_HEAD
VDX_PROGRAMS
VDX_PROGRAMS_PROCESS
VDX_PROGRAMS_PROCESS_LOCKS
VDX_SETTING_ENTRIES
VDX_SETTING_ENTRY_VALUES
VDX_SYSGLOBS
WAIVTPS
WEBVIEW_CIRC_ERRORS
WORD_INDEX
XMIT_BACKOFF_INTERVAL
Z_MAPPINGS
Z_MAPPINGS_TAG
Z_MAPPINGS_TAG_INDICATORS
Z_MAPPINGS_TAG_SUBFIELDS
Z_ACCESS_RESTTP
Z_ATTRIBUTES
Z_ATTRIBUTE_SETS
Z_ATTRIBUTE_TYPES
Z_ATTR_COMBINATIONS
Z_ATTR_COMB_ATTR
Z_DATABASE_CLASS
Z_DEDUPLICATION
Z_DEDUPLICATION_ELEMENT
Z_DIAGNOSTIC_CONDITIONS
Z_DIAGNOSTIC_SETS
Z_ELEMENTS
Z_ELEMENT_DATATYPES
Z_ISO_AVAILABILITY_MAPPING
Z_OPERATORS
Z_QUERY_TYPES
Z_RECORDS
Z_RECORD_SYNTAXES
Z_RECORD_SYNTAX_SCHEMA
Z_RECORD_TAGS

Z_SCHEMAS
Z_TAGSETS
Z_TAGSET_ELEMENTS
Z_TARGET_NETWORK_ADDRS
Z_TDBS
Z_TDB_ACCESS_REST
Z_TDB_ATTRIBUTES
Z_TDB_ATTRIBUTE_ADAPTION
Z_TDB_ATTRIBUTE_SETS
Z_TDB_ATTRIBUTE_TYPES (view)
Z_TDB_CHARGES
Z_TDB_DIAGNOSTIC_SETS
Z_TDB_NICKNAMES
Z_TDB_QUERY_TP_PRIVATE
Z_TDB_QUERY_TP_PRIVATE_OPS
Z_TDB_QUERY_TP_PRIV_DESC
Z_TDB_QUERY_TP_RPN
Z_TDB_QUERY_TP_RPN_OPS
Z_TDB_QUERY_TP_RPN_PROX_UNITS
Z_TDB_QUERY_TP_SEARCH_KEYS
Z_TDB_QUERY_TP_STRING
Z_TDB_QUERY_TYPES (view)
Z_TDB_RECORD_SYNTAXES
Z_TDB_SCHEMAS
Z_TDB_TERM_LISTS
Z_UNITS

Appendix 2 Tables having IDA data

1. CLASSES
2. OBJECTS
3. SERIES
4. SUBJECTS
5. TITLECLASSES
6. TITLENOTES
7. TITLEOBS
8. TITLEPHYSICAL
9. TITLEPRICES

10. TITLEPUB
11. TITLES
12. TITLESERIES
13. TITLESUBJS

Appendix 3 Package to extract OLIB data

CREATE OR REPLACE PACKAGE BODY Olib_Ida_Package AS

-- File : olibdata_packgae_body.sql

-- Developed By :Ghazanfar Alam

-- Date :19/09/03

-- Purpose :This package body bundles all the related procedure to perform the data
transfer from OLIB to IDA temporary table

--

--Package object: Procedure purpose

-- -----

-- OLIBDATA This procedure will take all the necessary
data from olib tables and insert it into
the temporary ida table bgs_unpub_vol

--

-- OLIBDATA_UPDATE This procedure will concatenate all the
subject and find out the physical location
for a title and then update the data
into the table bgs_unpub_vol

--

--Version hist : Date change purpose

-- 6/11/03 Added second_entry field Bgs.bgs_unpub_reports has primary key
based on indcode and secondentry

-- value '0'into BGS_unpub_vol

--

-- *****

ERRMSG VARCHAR2(200);

PROCEDURE OLIBDATA IS

CURSOR C1 IS

```
SELECT  A.TITLENO,
        SUBSTR(A.TITLE,1,240),
        SUBSTR(A.STMNT_RESBTY,1,200),
        A.CATDATE,
        SUBSTR(A.PRCLASSMARK,1,20),
        A.COPYCAT,
        SUBSTR(A.MUSFORM,1,1),
```

SUBSTR(A.NOTES,1,240),
 A.PUBDATE,
 A.LL_OS_NORTHING,
 A.UR_OS_NORTHING,
 A.LL_OS_EASTING,
 A.UR_OS_EASTING,
 A.LL_OS_SHEET,
 A.UR_OS_SHEET,
 B.LENGTH

FROM OLIB_VIEW.TITLES A,OLIB_VIEW.TITLEPHYSICAL B
 WHERE A.PRCLASSTP='BGS' AND A.PRCLASSMARK LIKE '___/___/%' AND
 A.TITLENO=B.TITLENO(+);

--VARIABLE FOR CURSOR

TITLENO NUMBER(10);
 TITLE VARCHAR2(240);
 AUTHORS VARCHAR2(200);
 CATDATE DATE;
 CLASSMARK VARCHAR2(20);
 COPYCAT VARCHAR2(5);
 MUSFORM VARCHAR2(1);
 NOTES VARCHAR2(240);
 PUBDATE DATE;
 LL_NORTHING VARCHAR2(4);
 UR_NORTHING VARCHAR2(4);
 LL_EASTING VARCHAR2(4);
 UR_EASTING VARCHAR2(4);
 LL_SHEET VARCHAR2(2);
 UR_SHEET VARCHAR2(2);
 COLLN VARCHAR2(240);

--VARIABLE FOR MANUPULATION

TITLEENTRY VARCHAR2(240);
 VAR_ACCNO VARCHAR2(20);
 ACCNO NUMBER(8);
 CONF VARCHAR2(1);
 LENSTR NUMBER(3);
 N_COUNT NUMBER(3);


```

V_STR_LENGTH    NUMBER(3);
V_FULL_LENGTH   NUMBER(3);
V_START_LENGTH  NUMBER(3);
V_END_LENGTH    NUMBER(3);
V_NOW_LENGTH    NUMBER(3);
V_OLD_CODE      VARCHAR2(25);
TITLE_CONF      VARCHAR2(240);

```

```
BEGIN
```

```
    Delete from BGS_UNPUB_VOL;
```

```
    commit;
```

```
    OPEN c1;
```

```
        LOOP
```

```
            FETCH c1 INTO TITLENO,
```

```
                TITLE,
```

```
                AUTHORS,
```

```
                CATDATE,
```

```
                CLASSMARK,
```

```
                COPYCAT,
```

```
                MUSFORM,
```

```
                NOTES,
```

```
                PUBDATE,
```

```
                LL_NORTHING,
```

```
                UR_NORTHING,
```

```
                LL_EASTING,
```

```
                UR_EASTING,
```

```
                LL_SHEET,
```

```
                UR_SHEET,
```

```
                COLLN;
```

```
        EXIT WHEN c1%NOTFOUND;
```

```
            V_STR_LENGTH:=instr(CLASSMARK,',',-1);
```

```
            VAR_ACCNO:=substr(CLASSMARK,V_STR_LENGTH+1,8);
```

```
            LENSTR:=LENGTH(VAR_ACCNO);
```

```
            N_COUNT:=1;
```

```
            --find out if there is any character into var_accno
```

```
            LOOP
```

```
                IF UPPER(SUBSTR(VAR_ACCNO,N_COUNT,1)) IN
                ('A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z','(',')') THEN
```

```
                    EXIT;
```

```

        END IF;

        N_COUNT:=N_COUNT+1;
        EXIT WHEN N_COUNT > LENSTR;
    END LOOP;
-- if the first character is not number then ind out number in between
last two salsh
    IF N_COUNT=1 THEN
        V_FULL_LENGTH:=LENGTH(CLASSMARK);
        V_START_LENGTH:=(V_STR_LENGTH-
V_FULL_LENGTH-2);
        V_END_LENGTH:=(V_FULL_LENGTH-V_STR_LENGTH);

        V_NOW_LENGTH:=INSTR(CLASSMARK, '/', V_START_LENGTH)+1;

        ACCNO:=TO_NUMBER(SUBSTR(CLASSMARK, V_NOW_LENGTH, V_END_LENGTH));

        ELSE

        ACCNO:=TO_NUMBER(SUBSTR(VAR_ACCNO, 1, N_COUNT-1));
        END IF;
        -- finding old code
        IF          UPPER(SUBSTR(CLASSMARK, 4, 1))          IN
('A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z') THEN

        V_OLD_CODE:=SUBSTR(CLASSMARK, 4, LENGTH(CLASSMARK));
        ELSE
            V_OLD_CODE:=NULL;
        END IF;

        -- finding confidentiality
        IF COPYCAT='LTD1' THEN
            TITLEENTRY:='Title Confidential';
            CONF:='R';
            TITLE_CONF:=TITLE;
        ELSIF COPYCAT='LTD2' THEN
            TITLEENTRY:='Title Confidential';
            CONF:='C';
            TITLE_CONF:=TITLE;
        ELSE
            TITLEENTRY:=TITLE;

```

```
CONF:=NULL;
TITLE_CONF:=NULL;
END IF;
-- inserting data
INSERT INTO BGS_UNPUB_VOL
    (TITLENO,
     SERIES,
     ACTIVITY,
     TITLE,
     AUTHORS,
     MONTH,
     YEAR,
     ACCNO,
     CONF,
     INDCODE,
     COLLN,
     COSTTXT,
     FREENGDC,
     OLDCODE,
     USER_ENTERED,
     DATE_ENTERED,
     CONF_TITLE,
     SECOND_ENTRY,
     SWN,
     NEN,
     NEE,
     SWE,
     SWC,
     NEC,
     YEAR_RELEASE,
     NOTES)
VALUES
    (TITLENO,
     SUBSTR(CLASSMARK,1,1),
     SUBSTR(CLASSMARK,2,1),
     TITLEENTRY,
     AUTHORS,
     TO_CHAR(CATDATE,'MM'),
```

```

TO_NUMBER(TO_CHAR(CATDATE,'YYYY')),

ACCNO,
CONF,
CLASSMARK,
COLLN,
MUSFORM,
NOTES,
V_OLD_CODE,
'BY PACKAGE',
SYSDATE,
TITLE_CONF,
'0',
LL_NORTHING,
UR_NORTHING,
UR_EASTING,
LL_EASTING,
LL_SHEET,
UR_SHEET,

TO_NUMBER(TO_CHAR(PUBDATE,'YYYY')),

NOTES );

END LOOP;
CLOSE c1;
COMMIT;

EXCEPTION
/*Possible error in
columns:--classmark,var_accno,accno,v_old_code
to check:-- Actual Data,Data type and data length
*/
WHEN OTHERS THEN
ERRMSG:=SUBSTR(SQLERRM,1,240);
INSERT INTO OLIBIDA_PACKAGE_ERROR_LOG
values(TITLENO,sysdate,ERRMSG,'OLIBDATA');
DBMS_OUTPUT.PUT_LINE(CLASSMARK);
END OLIBDATA;
PROCEDURE OLIBDATA_UPDATE IS

```

CURSOR C2 IS

```

SELECT  A.SUBJECT_DISP,B.TITLENO
FROM    OLIB_VIEW.SUBJECTS A,
        OLIB_VIEW.TITLESUBJS B,
        OLIB_VIEW.TITLES c
WHERE   A.SUBJECT=B.SUBJECT AND
        B.TITLENO=c.titleno AND
        C.PRCLASSTP='BGS' AND C.PRCLASSMARK LIKE '___/___/%%'
ORDER BY B.TITLENO;

```

--VARIABLE FOR CURSOR

```

SUBJ_TITLENO    NUMBER(10);
SUBJECT VARCHAR2(240);

```

--VARIABLE FOR MANUPULATION

```

V_SUBJECT    VARCHAR2(240);
N_OLD_TITLENO NUMBER(10);

```

BEGIN

--update location

```

UPDATE BGS_UNPUB_VOL D SET D.FILE_URL=
        (SELECT A.LOCATOR
         FROM    OLIB_VIEW.OBJECTS A,
                OLIB_VIEW.TITLEOBS B,
                OLIB_VIEW.TITLES C
         WHERE   D.TITLENO=B.TITLENO AND
                A.TITLE='Report in pdf (BGS internal use ONLY)' AND
                A.OBJECTNO=B.OBJECTNO AND
                B.TITLENO=C.TITLENO AND
                C.PRCLASSTP='BGS' AND C.PRCLASSMARK LIKE
'___/___/%%');
V_SUBJECT:=NULL;
N_OLD_TITLENO:=NULL;
OPEN C2;

```

LOOP

```

FETCH C2 INTO SUBJECT, SUBJ_TITLENO;
EXIT WHEN C2%NOTFOUND;

```

```

-- concatenate and update the subject

IF N_OLD_TITLNO=SUBJ_TITLNO THEN

    IF LENGTH(V_SUBJECT||','||SUBJECT)<=240 THEN
        V_SUBJECT:=V_SUBJECT||','||SUBJECT;
    END IF;

    ELSIF N_OLD_TITLNO IS NULL THEN
        V_SUBJECT:=SUBJECT;
    ELSIF N_OLD_TITLNO<>SUBJ_TITLNO THEN
        UPDATE BGS_UNPUB_VOL SET SUBJ_INDX=V_SUBJECT WHERE
TITLNO=N_OLD_TITLNO;
        V_SUBJECT:=SUBJECT;
    END IF;
    N_OLD_TITLNO:=SUBJ_TITLNO;
END LOOP;
CLOSE C2;
COMMIT;

EXCEPTION
/*Possible error in
    columns:--v_subject
    to check:-- lenght exceeds 240
*/
WHEN OTHERS THEN
ERRMSG:=SUBSTR(SQLERRM,1,240);
INSERT INTO OLIBIDA_PACKAGE_ERROR_LOG
values(N_OLD_TITLNO,sysdate,ERRMSG,'OLIBDATA_UPDATE');
    DBMS_OUTPUT.PUT_LINE (ERRMSG);
END OLIBDATA_UPDATE;
END Olib_Ida_Package;

```

Appendix 4 Temporary table

Table: --BGS_UNPUB_VOL

```

*****
-- File           : olib_temp_table.sql
-- Developed By   : Ghazanfar Alam
-- Date          : 19/09/03
-- Purpose       : To create temporary table
--
--Version hist   : Date           change           purpose
--
*****

```

```

CREATE TABLE BGS_UNPUB_VOL
(
  TITLENO        NUMBER(10),
  SERIES          VARCHAR2(1),
  ACTIVITY       VARCHAR2(1),
  TITLE          VARCHAR2(240),
  AUTHORS        VARCHAR2(200),
  MONTH          VARCHAR2(2),
  YEAR           NUMBER(4),
  ACCNO          NUMBER(8,4),
  CONF           VARCHAR2(1),
  FUND_CODE      VARCHAR2(14),
  FREEDIV        VARCHAR2(240),
  INDCODE        VARCHAR2(20),
  COLLN          VARCHAR2(240),
  SUBJ_INDX      VARCHAR2(240),
  GEOG_INDX      VARCHAR2(240),
  COSTTXT        VARCHAR2(1),
  COSTMAP        VARCHAR2(1),
  LOCMST         VARCHAR2(20),
  LOCDYE         VARCHAR2(40),
  NOSOLD         NUMBER(3),
  FREENGDC       VARCHAR2(240),

```

```

OLDCODE      VARCHAR2(25),
OFFICE        VARCHAR2(2),
USER_ENTERED  VARCHAR2(10),
DATE_ENTERED  DATE,
USER_UPDATED  VARCHAR2(10),
DATE_UPDATED  DATE,
REGFILE       VARCHAR2(20),
SUBJCHEK      VARCHAR2(1),
GEOGCHEK      VARCHAR2(1),
CONF_TITLE    VARCHAR2(240),
PRICE         NUMBER(7,2),
SECOND_ENTRY  VARCHAR2(1),
SWN           VARCHAR2(7),
NEN           VARCHAR2(7),
NEE           VARCHAR2(6),
SWE           VARCHAR2(6),
SWC           VARCHAR2(2),
NEC           VARCHAR2(2),
QS            VARCHAR2(6),
COSTPLT       VARCHAR2(1),
SUBACTIVITY   VARCHAR2(2),
NEW_INDCODE   VARCHAR2(20),
NEW_ACCNO     NUMBER(8,4),
YEAR_RELEASE  NUMBER(4),
NOTES         VARCHAR2(240),
ACCNO_SUFFIX  VARCHAR2(5),
DIAGRAM_ENC   CHAR(1)          DEFAULT 'N',
FILE_URL      VARCHAR2(255)
)
/
CREATE INDEX BGS_UNPUB_VOL_indx1 ON BGS_UNPUB_VOL
(TITLENO)
/
CREATE INDEX BGS_UNPUB_VOL_indx2 ON BGS_UNPUB_VOL
(indcode)
/

```


Appendix 5 Current IDA table

Table: -BGS_UNPUB_REPORTS

Field	Data Type	Constraint
SERIES	VARCHAR2(1),	
ACTIVITY	VARCHAR2(1),	
TITLE	VARCHAR2(0),	
AUTHORS	VARCHAR2(0),	
MONTH	VARCHAR2(2),	
YEAR	NUMBER(4),	
ACCNO	NUMBER(8,4),	
CONF	VARCHAR2(1),	
FUND_CODE	VARCHAR2(14),	
FREEDIV	VARCHAR2(0),	
INDCODE	VARCHAR2()	NOT NULL,
COLLN	VARCHAR2(0),	
SUBJ_INDX	VARCHAR2(0),	
GEOG_INDX	VARCHAR2(0),	
COSTTXT	VARCHAR2(1),	
COSTMAP	VARCHAR2(1),	
LOCMST	VARCHAR2(),	
LOCDYE	VARCHAR2(),	
NOSOLD	NUMBER(3),	
FREENGDC	VARCHAR2(0),	
OLDCODE	VARCHAR2(),	
OFFICE	VARCHAR2(2),	
USER_ENTERED	VARCHAR2(10),	
DATE_ENTERED	DATE,	
USER_UPDATED	VARCHAR2(10),	
DATE_UPDATED	DATE,	
REGFILE	VARCHAR2(),	
SUBJCHEK	VARCHAR2(1),	
GEOGCHEK	VARCHAR2(1),	
CONF_TITLE	VARCHAR2(0),	
PRICE	NUMBER(7,2),	
SECOND_ENTRY	VARCHAR2(1)	NOT NULL,
SWN	VARCHAR2(7),	
NEN	VARCHAR2(7),	
NEE	VARCHAR2(6),	

SWE	VARCHAR2(6),	
SWC	VARCHAR2(2),	
NEC	VARCHAR2(2),	
QS	VARCHAR2(6),	
COSTPLT	VARCHAR2(1),	
SUBACTIVITY	VARCHAR2(2),	
NEW_INDCODE	VARCHAR2(),	
NEW_ACCNO	NUMBER(8,4),	
YEAR_RELEASE	NUMBER(4),	
NOTES	VARCHAR2(0),	
ACCNO_SUFFIX	VARCHAR2(5),	
DIAGRAM_ENC	CHAR(1)	DEFAULT 'N',
FILE_URL	VARCHAR2(5)	

Appendix 6 Final materialised view

```
CREATE MATERIALIZED VIEW BGS_UNPUB_REPORTS_MV1
```

```
*****
```

```
-- File      : Bgs_Unpub_Reports_Matview.sql
-- Developed By :Ghazanfar Alam
-- Date      :19/09/03
-- Purpose   :Create a view from existing IDA table and the temp table which holds
--           the olid data.
--
--Version hist : Date      change      purpose
--
```

```
*****
```

```
REFRESH START WITH ROUND(SYSDATE) + 8/24
```

```
NEXT TRUNC(SYSDATE+1) + (08/24)
```

```
AS
```

```
select SERIES,
ACTIVITY,
TITLE      ,
AUTHORS    ,
MONTH      ,
YEAR       ,
ACCNO      ,
CONF       ,
FUND_CODE  ,
FREEDIV    ,
INDCODE    ,
COLLN      ,
SUBJ_INDX  ,
GEOG_INDX  ,
COSTTXT    ,
COSTMAP    ,
LOCMST     ,
LOCDYE     ,
NOSOLD     ,
FREENGDC   ,
OLDCODE    ,
OFFICE     ,
USER_ENTERED ,
DATE_ENTERED ,
```

USER_UPDATED ,
 DATE_UPDATED ,
 REGFILE ,
 SUBJCHEK ,
 GEOGCHEK ,
 CONF_TITLE ,
 PRICE ,
 SECOND_ENTRY ,
 SWN ,
 NEN ,
 NEE ,
 SWE ,
 SWC ,
 NEC ,
 QS ,
 COSTPLT ,
 SUBACTIVITY ,
 NEW_INDCODE ,
 NEW_ACCNO ,
 YEAR_RELEASE ,
 NOTES ,
 ACCNO_SUFFIX ,
 DIAGRAM_ENC ,
 FILE_URL

from bgs.bgs_unpub_reports a where not exists (select B.INDCODE from bgs_unpub_vol b where a.indcode=b.indcode)

union ALL

select SERIES,
 ACTIVITY,
 TITLE ,
 AUTHORS ,
 MONTH ,
 YEAR ,
 ACCNO ,
 CONF ,
 FUND_CODE ,
 FREEDIV ,
 INDCODE ,
 COLLN ,
 SUBJ_INDX ,
 GEOG_INDX ,

```
COSTTXT      ,
COSTMAP      ,
LOCMST      ,
LOCDYE      ,
NOSOLD      ,
FREENGDC    ,
OLDCODE     ,
OFFICE      ,
USER_ENTERED ,
SYSDATE     ,
USER_UPDATED ,
DATE_UPDATED ,
REGFILE     ,
SUBJCHEK    ,
GEOGCHEK   ,
CONF_TITLE  ,
PRICE       ,
SECOND_ENTRY ,
SWN         ,
NEN         ,
NEE         ,
SWE         ,
SWC         ,
NEC         ,
QS          ,
COSTPLT     ,
SUBACTIVITY ,
NEW_INDCODE ,
NEW_ACCNO   ,
YEAR_RELEASE ,
NOTES       ,
ACCNO_SUFFIX ,
DIAGRAM_ENC ,
FILE_URL
from bgs_unpub_vol;
```

Appendix 7 Indexes for performance

```
*****
-- File      : Bgs_Unpub_Reports_Matview_INDEXES.sql
-- Developed By :Ghazanfar Alam
-- Date      :19/09/03
-- Purpose   :Creating the index on materialized view
--
--Version hist : Date      change      purpose
--
*****

Create index bgs_unpub_report_mv1_nu1 on BGS_UNPUB_REPORTS_MV1(year)
/
Create index bgs_unpub_report_mv1_nu2 on BGS_UNPUB_REPORTS_MV1(Series)
/
Create index bgs_unpub_report_mv1_nu3 on BGS_UNPUB_REPORTS_MV1(Title)
/
Create index bgs_unpub_report_mv1_nu4 on BGS_UNPUB_REPORTS_MV1(Activity)
/
Create index bgs_unpub_report_mv1_nu5 on BGS_UNPUB_REPORTS_MV1(Authors)
/
Create index bgs_unpub_report_mv1_nu6 on BGS_UNPUB_REPORTS_MV1(indcode)
/
Create index bgs_unpub_report_mv1_nu7 on BGS_UNPUB_REPORTS_MV1(oldcode)
/
```

Appendix 8 Error log table

```
*****
-- File      : olib_temp_table.sql
-- Developed By :Ghazanfar Alam
-- Date      :19/09/03
-- Purpose   :To create error log table
--
--Version hist : Date      change      purpose
--
*****

CREATE TABLE OLIBIDA_PACKAGE_ERROR_LOG
      (TITLENO  NUMBER(10),
      ERRDATE  DATE,
      ERRSTRING  VARCHAR2(240),
      ERR_PROC VARCHAR2(50))

/
```