Many DBA’s are being required to support multiple DBMS’s on multiple platforms. Many IT shops today are running a combination of Oracle and DB2 which is resulting in either having to cross train DBA’s or hire new DBA’s with expertise on the specific DBMS. The new Oracle compatibility features available in DB2 9.7 LUW brings these 2 worlds much closer together. This presentation will compares the architectural differences between Oracle and DB2 LUW, and how DB2 9.7 will simplify the porting of Oracle applications over to DB2 for LUW.
Objective 1: Understand the basics of Oracle objects and terminology compared to DB2 LUW
Objective 2: Overview of key architectural differences
Objective 3: Differences in the way space is managed across databases
Objective 4: Compare and contrast administration of Oracle vs. DB2 LUW
Objective 5: Understand how DB2 9.7 can simplify crossing from Oracle to DB2
## Components of DBMS

**Oracle**
- Instance
- File
- Database
- Tablespace
- Schema
- Table
- Index
- View
- Trigger
- Rollback Segment
- Stored Procedures
- SQL Plus
- Packages

**DB2 LUW**
- Instance
- Container
- Database
- Tablespace
- Schema
- Table
- Index
- View
- Trigger
- NA
- Stored Procedures
- DB2 CLP
- Modules

Roles – Grouping of authorizations
### Oracle
- **Express Edition**
  - 1 CPU/1GB RAM
  - 4 GB max DB size
  - Windows/Linux only
- **Standard Edition**
  - 4 sockets
  - Unlimited scalability
- **Enterprise Edition**
  - Unlimited scalability
  - Table Partitioning
  - RAC

### DB2
- **Express-C**
  - 2 CPU/2GB RAM
  - Windows/Linux/Solaris/Mac
- **Express Edition**
  - 4 CPU/4GB RAM max
- **Workgroup Edition**
  - 4 CPUs/16 GB RAM
- **Enterprise Server Edition**
  - Unlimited scalability
  - Data partitioning (DPF)
  - Purescale

**Oracle**
- Enterprise Edition typically used. Required for OEM add on packs.
- Requires you to purchase RAC in order to do Parallel server. “Real Application Clusters”

**DB2**
The data partition facility or DPF is the clustered version of DB2. This was formerly known as Enterprise Extended Edition (“triple E”)
**System Dictionary/Catalog**

**Oracle**
- **Base tables**
  - SYS.xxxxx$  
- **System views**
  - SYS.GV_$ or GV$  
  - SYS.V_$ or VS  
  - ALL_  
  - DBA_  
  - USER_

**DB2**
- SYSIBM.xxxx  
- SYSCAT
  - Read-only views defined for catalog base tables  
- SYSSTAT
  - Updateable set of views  
  - Primarily used for access path manipulation  
- Administrative Views

**Oracle**
- VS views used like snapshots in DB2
Oracle

- OEM - GUI tool set
  - DBA Mgmt Pack (free)
  - Change Mgmt Pack ($)
  - Diagnostics Pack ($)
  - Tuning Pack ($)
  - Recovery Manager ($)
- Sqlplus

DB2

- Control Center GUI tool set
  - Command center
  - Command line processor
  - Command window
  - Script center
  - Visual Explain
- Data Studio
  - Replacing Control Center

Oracle

DBA Pack – Basic C,A,D Export/Import, SQL LOADER. Very similar to CC
Enterprise Edition only

- Change management
- Diag Pack
- Tuning Pack
- Recovery Manager
### Common Terms Different Meanings

#### Oracle
- Stored Procedure
  - PL/SQL
- Package
  - grouping of PL/SQL blocks
  - Nothing to do with optimization
- Segment
  - Set of Oracle Extents

#### DB2
- Stored Procedure
  - External - C, Java, Cobol
  - Internal, SQL/PL
- Package
  - Pre-compiled static SQL
- Segment
  - Block of pages in TS

---

**Oracle**

PL/SQL requires a C compiler

**DB2**

Extent – An extent is a block of data pages which get allocated based on the EXTENTSIZE parameter of the tablespace definition. See the tablespace foil for more details.
### Different Terms Similar Meaning

<table>
<thead>
<tr>
<th><strong>Oracle</strong></th>
<th><strong>DB2</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Data Block</td>
<td>• Data Page</td>
</tr>
<tr>
<td>• Dictionary</td>
<td>• Catalog</td>
</tr>
<tr>
<td>• Alert Log</td>
<td>• Diag log</td>
</tr>
<tr>
<td>• Redo Log</td>
<td>• Log Files</td>
</tr>
<tr>
<td>• Archive log</td>
<td>• Archival Logging</td>
</tr>
<tr>
<td>• Segments</td>
<td>• Space Consuming Objects</td>
</tr>
<tr>
<td>• Statement Cache</td>
<td>• Package Cache</td>
</tr>
<tr>
<td>• System Global Area (SGA)</td>
<td>• Instance shared memory/DB shared memory</td>
</tr>
</tbody>
</table>

Segments = Tables/IX’s

**DB2**

2 configuration files

- Database Manager configuration file – configuration parameters for entire instance
- Database configuration file – Created for each database
Instance & Database

Oracle
- Instance
  - A collection of Processes
  - One data dictionary
  - One active database
  - Init.ora parameter file

DB2
- Instance – Logical database server environment
  - Also referred to as a Node
  - 1 to many databases
  - Database Manager Configuration File

DB2
2 configuration files
Database Manager configuration file – configuration parameters for entire instance
Database configuration file – Created for each database
**Oracle**

- Control files are like BSDS or Recovery History File

Every Oracle database has a control file. A control file contains entries that specify the physical structure of the database:

- Database name
- Names and locations of datafiles and redo log files
- Time stamp of database creation

**DB2**

- On a single machine, one or more instances can exist

**DB2 Administration Server (DAS)**

- Enables remote administration of DB2 servers
- Provides scheduler used to schedule local and remote jobs
- Provides a mechanism for DB2 discovery to return information to remote clients

- Can connect to one or more databases at the same time on different instances and/or machines
- Cannot have SQL statements joining tables from different databases.

**Database Manager Configuration file**

- created when an instance is created
- parameters affect system wide resources at the product or instance level
- parameters can be changed from system default values improve performance or increase capacity.

**Database Configuration file**

- one config file is created when the database is created
- specifies amount of resources to be allocated to the database
Bufferpools

Oracle
- 3 bufferpools available
  - one fixed block (8k)
  - multi block size 4, 18, 16, 32
- Defined in init.ora
- Dynamic create, alter, and delete
- Objects assigned via DDL

DB2
- IBMDEFAULTBP
  automatically created with database
  - Additional pools created with DDL
- Hidden Bufferpools
  - 4k, 8k, 16k, & 32K
  - Share only within individual databases

9i allows 4, 18, 16, 32 pages
The biggest difference at the database level is how the system catalog is defined.

* In Oracle there is one common system catalog for the databases
* On the Unix/NT platform a new system catalog gets generated for every database defined within an instance.

  - The catalog tables and memory structures are actually defined within the database itself.
  - 3 SMS tablespaces are created by default
    - SYSCATSPACE – Contains system catalog tables = SYSTEM tablespace in Oracle
    - TEMPSPACE1 – Holds temp tables used by UDB
    - USERSPACE1 – Contains user tables unless tablespace specified (DSNDB04)
Storage Management

Oracle
- Data Files
  - Cooked
  - Raw
  - Auto extend

DB2
- Container
  - Device
  - Directory
  - File

Cooked – Formatted by O/S
Raw – Native device
Resize – Dynamically changes size
Auto extend – similar to SMS
### Tablespace

<table>
<thead>
<tr>
<th>Oracle</th>
<th>DB2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System</strong></td>
<td><strong>Regular</strong></td>
</tr>
<tr>
<td>- Automatically created</td>
<td>- Temporary</td>
</tr>
<tr>
<td><strong>SYSAUX</strong></td>
<td>- Large</td>
</tr>
<tr>
<td>- Auxiliary system TS</td>
<td></td>
</tr>
<tr>
<td><strong>UNDO</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Non-System</strong></td>
<td></td>
</tr>
<tr>
<td>- Permanent</td>
<td></td>
</tr>
<tr>
<td>- Temporary</td>
<td></td>
</tr>
<tr>
<td>- BIGFILE/SMALLFILE</td>
<td></td>
</tr>
</tbody>
</table>

Permanent used for all objects much like USERSPACE in LUW
Temporary used for sorts, temp workspace, etc.

Bigfile tablespaces can significantly increase the storage capacity of an Oracle database. Smallfile tablespaces can contain up to 1024 files, but bigfile tablespaces contain only one file that can be 1024 times larger than a smallfile tablespace. The total tablespace capacity is the same for smallfile tablespaces and bigfile tablespaces. However, because there is a limit of 64K datafiles for each database, a database can contain 1024 times more bigfile tablespaces than smallfile tablespaces, so bigfile tablespaces increase the total database capacity by 3 orders of magnitude. In other words, 8 exabytes is the maximum size of the Oracle database when bigfile tablespaces are used with the maximum block size (32 k).
Oracle System vs. Non-System

- System Tablespace
  - Data dictionary
  - Rollback segment
  - PL/SQL Rollback Units

- Non-SYSTEM
  - Rollback segments
  - Temporary segments
  - Application data
  - Application indexes
<table>
<thead>
<tr>
<th>Oracle</th>
<th>DB2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segments</td>
<td>Storage Object</td>
</tr>
<tr>
<td>Extents</td>
<td>Extents</td>
</tr>
<tr>
<td>Data Blocks</td>
<td>Page</td>
</tr>
<tr>
<td>Data Files</td>
<td>Containers</td>
</tr>
</tbody>
</table>
Datafile

A tablespace in an Oracle database consists of one or more physical datafiles. A datafile can be associated with only one tablespace and only one database.
The type of tablespace chosen depends on the characteristics of the data stored within the tablespace. While DMS tablespaces clearly provide more flexibility for storage capacity, SMS tablespaces are generally recommended for temporary tablespaces and catalog tablespaces.

In addition to understanding the types of tablespaces, it is important to understand how data is managed within the tablespace. All data within DB2 is stored in pages. A page size is defined at tablespace creation and can be specified in 4k, 8k, 16k, and 32k sizes. Row size, random vs. sequential access, and several other factors must be evaluated to determine the optimal page size for the tablespace.

Pages are grouped into allocation units called extents. Each time the tablespace needs to allocate additional storage, the extent size is used to determine the size. During insert activity, DB2 UDB will write to a container until the extent size has reached capacity, at that point, DB2 UDB will allocate an extent on the next container and continue the write activity.
Partitioning

**Oracle**
- Table Definition
  - 1-64k partitions
  - Partitioning on range, hash, both or list
  - Partitions assigned to TS
- Enhances Parallel Query
- Easier Data Management

**DB2**
- **Table partitioning**
  - Partitioning part of table DDL
  - Partitions can be in one or multiple tablespaces
  - Partitions can be rolled in/out
  - Partitions can be administered independently
- **Database Partitioning**
  - Database Partitioning Feature (DPF)
    - ESE Edition only
  - Database is partitioned across multiple servers
  - Multiple tables

---

**Oracle**
Partitioning very much like DB2 z/OS in concept. Designed primarily for manageability
Local indexes- Can create an index on an individual partition.
**Tables**

- **Oracle**
  - One to many tables defined in a tablespace
  - Tables and Indexes are independent of each other
  - Index Organized Tables

- **DB2**
  - One to many tables defined in a tablespace
  - Indexes directly tied to table definition

**Index organized tables** – Whole data row stored in IX not just RID’s
**Indexes**

**Oracle**
- Unique
- Non-unique
- Partitioning
- Function based
- Bit Map

**DB2**
- Unique
- Non-unique
- Clustering
- Multi-Dimensional Clustering

Function based- Index which performs a function such as doing CAP’s on a name

Bit Map – Useful for low cardinality values such as color
When creating a table in UDB you must have a tablespace pre-defined for both the table and any indexes you might add to the table. The indexspace specification is part of the table definition. Therefore all indexes for the table use the same indexspace.
Administration
<table>
<thead>
<tr>
<th>Optimizer</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Oracle</strong></td>
<td><strong>DB2</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>• Cost based</td>
<td>• Cost based more sophisticated than Oracle</td>
</tr>
<tr>
<td>• Rules Based</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Adjusted based on query complexity</td>
</tr>
<tr>
<td></td>
<td>• No Rules</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Choose will check if stats exist and then determine which optimization to use.
Optimizer Class

- **DB2 Optimizer Class**
  - Values are between 0 and 9, default is 5
  - Determines the intensity used by the DB2 SQL Compiler when rewriting SQL
  - Dynamic SQL can spend time optimizing, use lower class
  - Static SQL optimizes once, use a higher class
  - "Vbr_queryopt" database setting
  - SET CURRENT QUERY OPTIMIZATION n

<table>
<thead>
<tr>
<th>Level</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Minimal amount of optimization. Only recommended for very simple SQL accessing well-indexed tables. Only nested loop joins and 1E scans enabled.</td>
</tr>
<tr>
<td>1</td>
<td>Similar to 0 except Merge Scan and 1S scan enabled.</td>
</tr>
<tr>
<td>2</td>
<td>Recommended for very complex queries which are infrequently executed in a decision support or OLAP environment.</td>
</tr>
<tr>
<td>3</td>
<td>Closest to 3DS optimizer. Recommended for queries with 4 or more joins.</td>
</tr>
<tr>
<td>5</td>
<td>DEFAULT - Most cost effective method for mix of simple and complex queries. Optimization will be automatically reduced for complex dynamic SQL if optimizer determines that the resources are not necessary.</td>
</tr>
<tr>
<td>7</td>
<td>Same as 5 except optimization not reduced for complex dynamic SQL.</td>
</tr>
<tr>
<td>9</td>
<td>Used to determine whether more comprehensive optimization can generate better access plans for very complex long running queries using large tables.</td>
</tr>
</tbody>
</table>

Optimizer Class

**Static SQL**
- Controlled on PREP and BIND statements
- Stored in QUERYOPT in SYSCAT.PACKAGES

**Dynamic SQL**
- CURRENT QUERY OPTIMIZATION register using SET command

If not set, DFT_QUERYOPT database parameter
**Oracle**
Very similar to z/OS

**DB2**

**LUW**

EXPLAIN_ARGUMENT: Represents the unique characteristics for each individual operator.

EXPLAIN_INSTANCE: Main control table for all explain information. Each row of data in the explain tables is explicitly linked to one row in this table. Basic information about the source of the SQL statements being explained and environment information is kept in this table.

EXPLAIN_OBJECT: Contains data objects required by the access plan to satisfy the SQL statement.

EXPLAIN_OPERATOR: Contains all the operators needed to satisfy the SQL statement.

EXPLAIN_PREDICATE: Identifies which predicates are applied by a specific operator.

EXPLAIN_STATEMENT: Contains the text of the SQL statement in two forms. The original version entered by the user, and a rewritten version generated by the compilation process.

EXPLAIN_STREAM: This table represents the input and output data streams between individual operators and data objects.
## Oracle

- **Performance Snapshots - V$**
  - Instance/Database
  - Memory
  - Disk
  - User/Session
  - Connection
- **AWR (Statspack)**
  - Collects Stats over time
  - Similar to DB2 Snapshot
- **TKPROF**
  - Formats SQL trace data from applications

## DB2

- **Snapshot Monitor**
  - Show status of database counters at instant in time
  - Buff, Lock, Sort, Stmt, Tbl, UOW
- **Event Monitor**
  - Status after the event
    - Databases
    - Tablespace
    - Connections
    - Tables
    - Statements
    - Transactions
    - Deadlocks
- **Administrative Views and Table Procedures**

---

**Oracle**

UTLBSTAT/UTLESTAT – Stats pac – Automatically capture stats over time

TKPROF – Similar to z/OS SQL traces

**STATSPACK** is a performance diagnosis tool, available since Oracle8i. STATSPACK can be considered BSTAT/ESTAT's successor, incorporating many new features. STATSPACK is a diagnosis tool for instance-wide performance problems; it also supports application tuning activities by providing data which identifies high-load SQL statements. STATSPACK can be used both proactively to monitor the changing load on a system, and also reactively to investigate a performance problem.
### Utilities

<table>
<thead>
<tr>
<th>Oracle</th>
<th>DB2</th>
</tr>
</thead>
<tbody>
<tr>
<td>• RMAN</td>
<td>• BACKUP</td>
</tr>
<tr>
<td>• IMPORT</td>
<td>• IMPORT</td>
</tr>
<tr>
<td>• EXPORT</td>
<td>• EXPORT</td>
</tr>
<tr>
<td>• SQL*LOADER</td>
<td>• LOAD</td>
</tr>
<tr>
<td>• ANALYZE</td>
<td>• RUNSTATS</td>
</tr>
<tr>
<td>• DB_VERIFY</td>
<td>• RESTORE</td>
</tr>
<tr>
<td></td>
<td>• REORG</td>
</tr>
<tr>
<td></td>
<td>• REORGCHK</td>
</tr>
</tbody>
</table>

- Analyze = Runstats
- DB_Verify =
- RMAN = Backup/Restore
- Export/Import=Reorg
Backup/Recovery
Backups

Oracle

- Database
  - Cold
  - Hot
- Tablespace
- Components
  - Database Backup
  - REDO Log
  - ARCHIVED REDO logs
  - UNDO Records
  - Control File

DB2

- Database Online/Offline
- Tablespace
- Components
  - Backup Image
  - Incremental Copy (7.2)
  - Backup History File
  - Active Logs
  - Archive Logs

Oracle Backups
Cold- Can use any type of copy method
Hot – Run alter commands and then copy datafiles
Export database- Can use the export command to backup database. Very slow.
Standby database = Fail over database

REDO Log
The redo log is a set of files that protect altered database data in memory that has not been written to the datafiles. The redo log can consist of two parts: the online redo log and the archived redo log.

UNDO records
Undo records can be stored in either undo tablespaces or rollback segments. Oracle uses the undo data for a variety of purposes, including accessing before-images of blocks changed in uncommitted transactions. During database recovery, Oracle applies all changes recorded in the redo log and then uses undo information to roll back any uncommitted transactions

Control Files
The control files of a database keep, among other things, information about the file structure of the database and the current log sequence number being written by LGWR. During normal recovery procedures, the information in a control file is used to guide the automated progression of the recovery operation. Oracle can multiplex the control file, that is, simultaneously maintain a number of identical control files.
**Oracle**

Control file is a small binary file that describes the structure of the database.

Properties

• All necessary database files and log files are identified in the control file.

RMAN – Keeps track of backups

---

### Oracle

<table>
<thead>
<tr>
<th>Control File</th>
<th>DB2</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Database name</td>
<td>• Updated:</td>
</tr>
<tr>
<td>• Data file location</td>
<td>• Backup of DB/TS</td>
</tr>
<tr>
<td>• Redo log file location</td>
<td>• Restore/Roll Forward of DB/TS</td>
</tr>
<tr>
<td>• Tablespace names</td>
<td>• Drop/Load/Reorg/Stats of a table</td>
</tr>
<tr>
<td>• Current log sequence number</td>
<td>• Quiesce/Alter T3</td>
</tr>
<tr>
<td>• Checkpoint information</td>
<td>• Contains</td>
</tr>
<tr>
<td>• Log history</td>
<td>• Part of DB which was copied</td>
</tr>
<tr>
<td>• Backup information</td>
<td>• When DB was copied</td>
</tr>
</tbody>
</table>

**Oracle**

Control file is a small binary file that describes the structure of the database.

Properties

• All necessary database files and log files are identified in the control file.

RMAN – Keeps track of backups
Oracle

The redo log is a set of files that protect altered database data in memory that has not been written to the datafiles. The redo log can consist of two parts: the online redo log and the archived redo log.

Redo log files record all changes made to the database, and are used for data recovery. If the redo log files are multilexed, the same redo log information is written to multiple online redo log files.

• Redo files are written to in a circular fashion
• Must be at least 2 redo log groups

Rollback Segments exist only until data is committed.
Supports both crash and version type recoveries
Primary log files are allocated when the database is created.
Secondary log files are allocated as needed.
- Automatically de-allocated when no longer needed
- Good for periodic large units of work
Active (15,16) – Contains information related to units of work that have not yet been committed or rolled back. They also contain information for transactions that have committed, but whose changes have not been written to disk.

Online archive (14) – Contains information related to completed transactions that no longer require crash recovery protection. These are called online because they reside in the same subdirectory as the active logs.

Offline archive (12,13) – log files which have been removed from the active log subdirectory. The files must be moved manually. There is no auto-archiving in UDB.
<table>
<thead>
<tr>
<th>Oracle</th>
<th>DB2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Instance Recovery</strong></td>
<td><strong>Crash</strong></td>
</tr>
<tr>
<td>• RAC only</td>
<td>• Uses logs to recover from power interrupts or application ABENDS</td>
</tr>
<tr>
<td>• Uses Redo logs to recover from System/Instance failures</td>
<td></td>
</tr>
<tr>
<td><strong>Crash</strong></td>
<td><strong>Roll-Forward</strong></td>
</tr>
<tr>
<td>• Uses online REDO logs</td>
<td>• Image copy plus log apply</td>
</tr>
<tr>
<td>• Recovers data files</td>
<td>• LOCAL TIME (V8)</td>
</tr>
<tr>
<td><strong>Disaster</strong></td>
<td><strong>Version</strong></td>
</tr>
<tr>
<td><strong>Media Recovery</strong></td>
<td>• Image copy (TOCOPY)</td>
</tr>
<tr>
<td></td>
<td>• Import/Load (logical)</td>
</tr>
</tbody>
</table>
The Reorg Wizard can eliminate space problems by reorganizing database space usage. It also allows you to change objects storage settings and location. The Reorg Wizard gives you three reorganization options:

- Reorganize specific schema objects, such as tables or indexes
- Reorganize an entire tablespace
- Repair migrated rows

The Reorg Wizard performs reorganizations completely within the Oracle database using SQL. Reorganizations can be performed within the target tablespace or by temporarily using space in another tablespace. The Reorg Wizard allows you to indicate whether the generated script should favor availability or speed. If availability is chosen, the generated script will take full advantage of online capabilities supported by the server. On the other hand, if speed is chosen, the generated script will always use an off-line reorganization approach. The Reorg Wizard evaluates the reorganization operations and generates an Impact Report and reorganization script as shown below.

Online Redefinition Package – Designed to redefine a table from one type to another, but can be used like an online reorg.
Loading Data

Oracle

- SQL*Loader Utility
  - Insert/Append/Replace/Truncate
  - Direct=True|False
  - By-pass SGA build data block
  - No Triggers
  - No RI
  - Parallel=True|False
  - Multiple load files
  - Drop all Indexes
  - Recoverable/Unrecoverable

DB2

- Load
  - Locks single table in TR
  - Insert/Replace
  - RUNSTATS
  - Good for large amounts of data
  - READ ACCESS for rows not being loaded
  - Load directly from a SQL query

- ONLINE
  - Allow Read Access

- Import
  - Can dynamically create table
  - Insert process
    - Update
    - Replace
  - Good for small amounts of data
ORACLE COMPATIBILITY FEATURES
Enabling Oracle Compatibility Features

- **DB2 Compatibility Vector**
  - New registry value
    - Enables Oracle compatibility
  - Controls which compatibility features to turn on
  - `db2set DB2_COMPATIBILITY VECTOR=ORA`
    - Activates all features.

Compatibility Vector Options
- ROWNUM
- DUAL
- Outer join Operator
- Hierarchical queries
- NUMBER data type
- VARCHAR2 data type
- DATE data type
- TRUNCATE TABLE
- Character literals
- Collection methods
- Data disctionary-compatible views
- PL/SQL compilation
Compatibility Features

- Oracle data dictionary compatible views
  - ALL_*
  - DBA *
  - USER_*
- CLPPlus
  - SQL*Plus compatible command line processor
MODULES

- New DB2 Object
  - Groups object definitions together for an application
    - Functions
    - Procedures
    - TYPE
    - VARIABLE
- Same as Oracle PL/SQL Packages
Application Enablement

• Native PL/SQL support
  • No SQL translation required
  • PL/SQL compiler
  • PL/SQL debugger
  • PL/SQL Object support
    • After each row triggers
    • Anonymous Blocks
    • Before each row triggers
    • PL/SQL packages
    • Procedures
Common Oracle packages are commonly used in Oracle applications. These packages are now provided by DB2.

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBMS_ALERT</td>
<td>Allows sessions to semaphore</td>
</tr>
<tr>
<td>DBMS_JOB</td>
<td>DB2 task scheduler API</td>
</tr>
<tr>
<td>DBMSLOB</td>
<td>Oracle LOB API</td>
</tr>
<tr>
<td>DBMS_OUTPUT</td>
<td>Basic reporting</td>
</tr>
<tr>
<td>DBMS_PIPE</td>
<td>Allows sessions to send data back and forth</td>
</tr>
<tr>
<td>DBMS_SQL</td>
<td>Dynamic SQL API</td>
</tr>
<tr>
<td>DBMSUTILITY</td>
<td>Misc. application procedures</td>
</tr>
<tr>
<td>UTL_DIR</td>
<td>Maintains directory aliases used w/UTL_FILE</td>
</tr>
<tr>
<td>UTL_FILE</td>
<td>Allows working with files on DB2 server</td>
</tr>
<tr>
<td>UTL_MAIL</td>
<td>Send Email from SQL</td>
</tr>
</tbody>
</table>
New DB2 Data Types

**Oracle**
- BFILE
- BLOB
- CHAR(n)
- CLOB
- DATE
- LONG
- LONG RAW
- NCHAR(n)
- NCLOB
- NUMBER(p,s)
- NVARCHAR2(n)
- RAW(n)
- ROWID
- VARCHAR2(n)

**DB2**
- BOOLEAN
- INDEX BY
- NUMBER
- Oracle DATE
- Ref Cursor type
- ROW TYPE
- TIMESTAMP(n)
- VARCHAR2
- VARRAY
Summary

To be a successful cross platform DBA:

- Have a sound foundation of relational principles
- Understand the nuances of the individual platforms
- The new Oracle compatibility features built into Db2 9.7 provide an easy mechanism to transition existing Oracle applications over to DB2 LUW.
Jim Wankowski has more than 26 years of development and database administration experience with DB2. Jim is currently the DB2 product manager at Quest Software. Prior to joining Quest, he worked for nine years as a DB2 product specialist at Platinum Technology, and prior to that he worked as a DB2 DBA for Northrop Aircraft in Los Angeles. Jim is a well-known international speaker who has written articles for DB2 Magazine, and Z/Journal, and regularly presents at IDUG, regional DB2 user groups, and vendor seminars worldwide.