IBM DB2 日常维护

尤祖喜
Information Management Partner Ecosystem
youzuxi@cn.ibm.com

议程

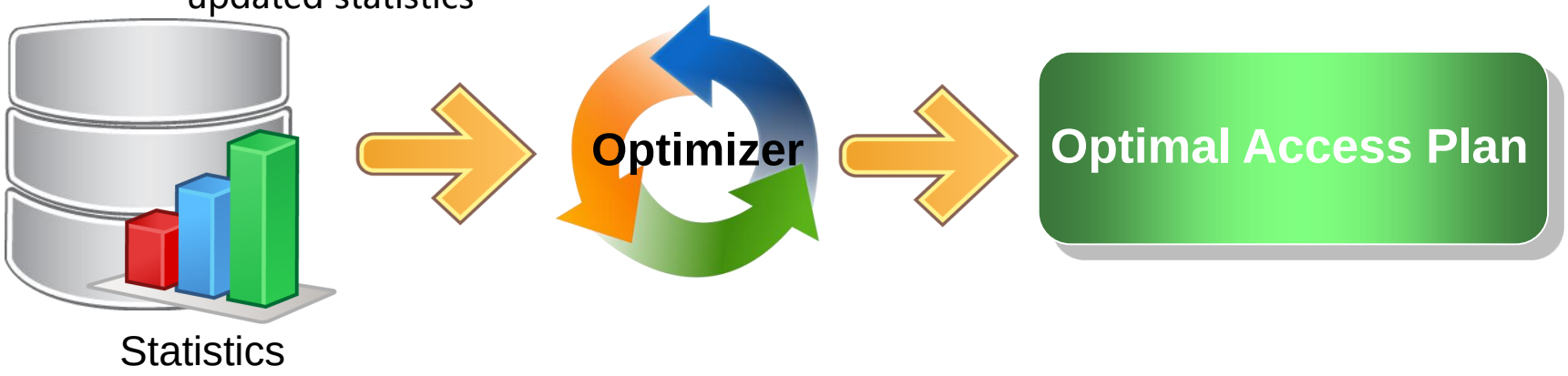
- DB2基本维护和自理功能
- DB2数据库备份和恢复功能
  - 数据库备份
  - 数据库恢复
- DB2数据导入导出
- 轻松将Oralce迁移到DB2数据库

Agenda

- **Basic Maintenance & Autonomic Features**
  - Statistics Collection
  - Data Reorganization
  - Table Space Maintenance
  - Automatic Maintenance
  - Self-Tuning Memory Manager (STMM)

Statistics Collection

- DB2 Optimizer needs to intelligently determine the most efficient way of servicing a SQL query. Its decisions are heavily influenced by statistical information from DB objects.
- DB2 **RUNSTATS** command updates statistics that profile the physical characteristics of a database table, along with its associated indexes.
  - Use after many DML changes or a reorganization of a table
  - Statistics are stored in the catalog tables (E.g.: `SYSCAT.TABLES, SYSSTAT.INDEXES`)
- After running **RUNSTATS:**
  - **Static SQL**
    - Requires explicit REBIND after statistics are updated using RUNSTATS
  - **Dynamic SQL**
    - Statements are prepared and executed at run time. They will automatically use updated statistics



RUNSTATS Command

```
db2 RUNSTATS ON TABLE FOR DB2INST1.TABLE1 AND INDEXES ALL
```

```
db2 RUNSTATS ON TABLE DB2INST1.TABLE1 AND INDEXES ALL TABLESAMPLE SYSTEM (10)
```

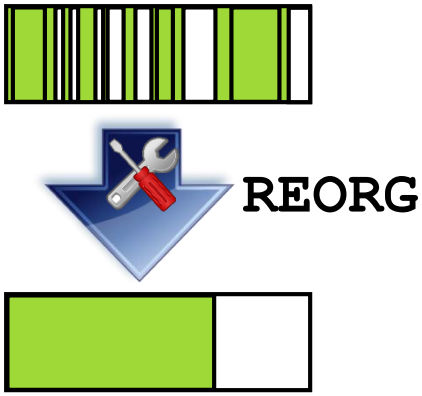
- Some situations when updating statistics would be beneficial
  - After data has been loaded into a table and appropriate indexes have been created
  - After creating a new index on a table
  - After a table has been reorganized with the REORG utility
  - After a table and its indexes have been significantly modified through upd insert, or delete operations

Guidelines for Collecting Statistics

- It’s important to keep statistics updated in order to provide the optimizer with accurate information for access plan selection.
- Collecting statistics of very large tables is challenging and can affect workload performance of the system.
- Improving the performance of RUNSTATS
  - Collect basic statistics only for relevant columns. E.g.: columns used to join tables
  - Consider specifying only those columns for which data distribution statistics should be collected.
  - Use row-level or page-level sampling when running RUNSTATS
  - Use throttling option to limit the performance impact of RUNSTATS execution

Data Reorganization

- Over time, data can become fragmented resulting in
  - Increased size of tables/indexes
  - Degraded performance as more pages need to be read
- **Table reorganization**
  - Eliminates fragmentation of table data
  - Reduces number of read operations to access data
  - Reorganizes table data to match index
  - Reclaim wasted space
- **Index reorganization**
  - Rebuilds the index data into un-fragmented, physically contiguous pages
  - Reduces I/O costs because of fragmented leaf pages, badly clustered index (which affects sequential prefetching) and indexes with too many leaves



Data Reorganization

- **Types of Reorg**
  - **Classic** (**offline**) → Sort, build, replace, recreate all indexes
    (+) fastest, perfectly clustered tables and indexes
    (-) limited table access, less control (can't be paused), more space required
  - **In-place** (**online**) → Select *n* pages, vacate the range, fill the range, truncate the table
    (+) full table access, recoverable, less space required
    (-) slower, potentially high logging requirements, subsequent index reorg might be needed
- **How to determine if tables, indexes, or both, need to be reorganized?**
  - Run the REORGCHK command
  - It calculates statistics on the database to determine if REORG is required.

```
db2 REORGCHK CURRENT STATISTICS ON TABLE db2inst1.orders
```

- **Options for REORGCHK Command:**

```
CURRENT STATISTICS or UPDATE STATISTICS
ON SCHEMA schemaname
or ON TABLE ALL or SYSTEM or USER tablename
```

- In addition to the REORGCHK command, you can use these stored procedures:
  - `SYSPROC.REORGCHK_TB_STATS` for tables
  - `SYSPROC.REORGCHK_IX_STATS` for indexes

Data Reorganization

- Perform reorganization of objects
  - Re-org a **Table**

indicates an online reorg

```
db2 REORG TABLE TABLE1 INDEX INDEX_TS1 INPLACE
```

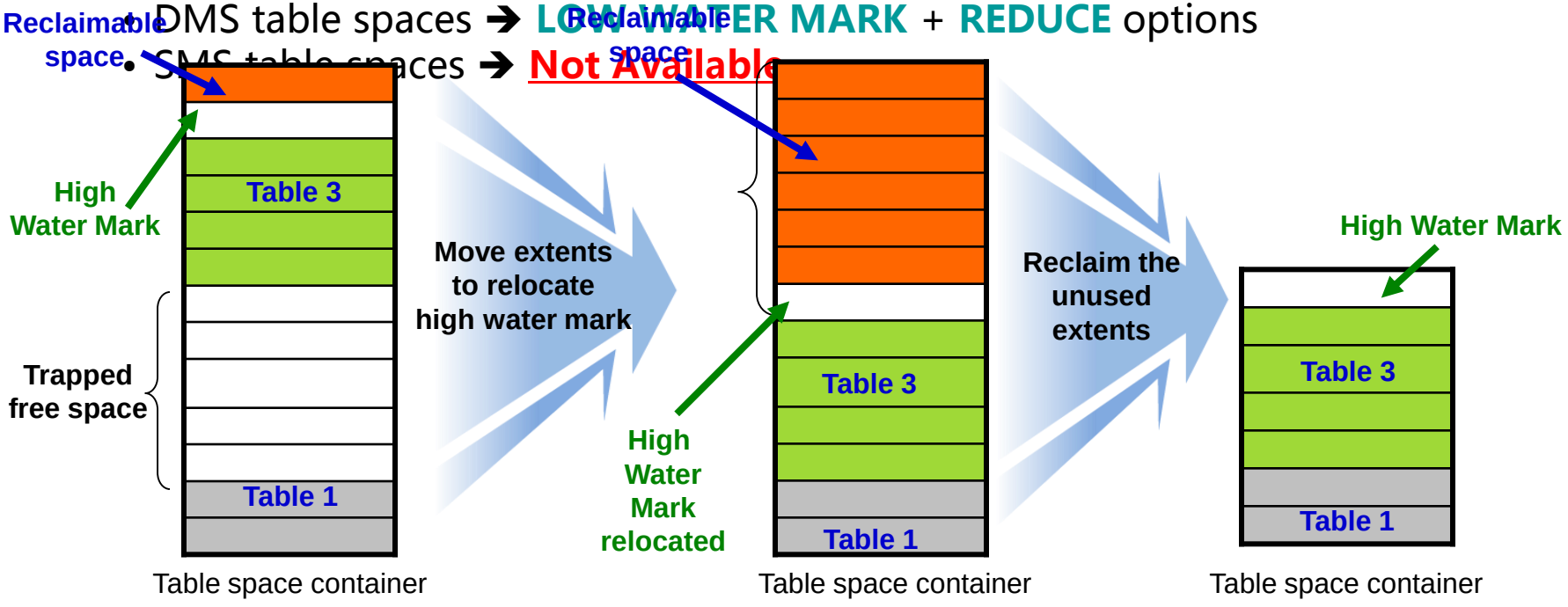
  - Re-org an **Index**

```
db2 REORG INDEXES ALL FOR TABLE TABLE1
```

- **Optional Step:** monitor progress in case of an online reorg
- **Post-tasks**:
  - Update statistics on reorganized objects
  - Rebind applications that access reorganized objects

Optimizing Table Space Usage

- Re-arrange the extents to lower the high water mark, then reclaim the unused extents
  - Use **ALTER TABLESPACE** command
    - Automatic storage table spaces ➔ **REDUCE** option
    - DMS table spaces ➔ **LOW WATER MARK** + **REDUCE** options
    - SMS table spaces ➔ **Not Available**



Reduce Size: Automatic Storage Table Space

- The DB manager attempts to lower the *high water mark and* reduce size of table space containers
  - Empty containers are dropped or resized, extents are moved to free space
  - Size can be reduced by a specific amount of pages, bytes (K,M,G),% or max. value

Syntax

```
ALTER TABLESPACE <tsname> REDUCE <size>
```

Example 1

```
ALTER TABLESPACE tsname REDUCE MAX
```

Example 2

```
ALTER TABLESPACE tsname REDUCE 25 PERCENT
```

- Example 1, the keyword MAX suggests to move the maximum number of extents possible to the beginning of the table space.
- Example 2, attempts to reduce the size of the table space TS1 to 75% of it's original size, if possible.

Reduce Size: DMS Table Space

- To reclaim unused storage, explicitly lower High Water Mark:
  1. Re-arrange extents in the table to make use of the free extents lower in the table space

```
ALTER TABLESPACE <tsname> LOWER HIGH WATER MARK
```

...unt.

Example

Alternatively, specify a container name

```
ALTER TABLESPACE ts LOWER HIGH WATER MARK
ALTER TABLESPACE ts REDUCE (ALL CONTAINERS 5 M)
```

- To reclaim unused index space on tables that reside in DMS table spaces

New in DB2 v10.1

```
REORG INDEX ALL FOR TABLE tbs RECLAIM EXTENTS
```

- Moves index pages around to create empty extents
- Frees pages from exclusive use by the index object

Determining Free Space to Reclaim

- **Check Table Space for Free Space**

```
db2 "select varchar(tbsp_name, 15) as tbsp_name, tbsp_type,
   reclaimable_space_enabled, tbsp_free_pages from
   table(mon_get_tablespace('EXTENTREMAP',-2)) as t"
```

TBSP_NAME	TBSP_TYPE	RECLAIMABLE_SPACE_ENABLED	TBSP_FREE_PAGES
EXTENTREMAP	DMS	1	6526

- **Determine Free Pages below/ above High Water Mark**

```
db2 "select varchar(tbsp_name, 15) as tbsp_name, tbsp_free_pages,
tbsp_total_pages, tbsp_page_top from table
(mon_get_tablespace('EXTENTREMAP',-2)) as t"
```

TBSP_NAME	TBSP_FREE_PAGES	TBSP_TOTAL_PAGES	TBSP_PAGE_TOP
EXTENTREMAP	6526	3538280	3536394

Free pages above HWM = TBSP_TOTAL_PAGES – TBSP_PAGE_TOP

Free pages below HWM = TBSP_FREE_PAGES – (free pages above HWM)

Autonomic Computing in DB2

- By sensing and responding to changes in the environment, DB2' s autonomic computing features **automatically adjust the system to optimize its operation**
- Included in ALL DB2 editions

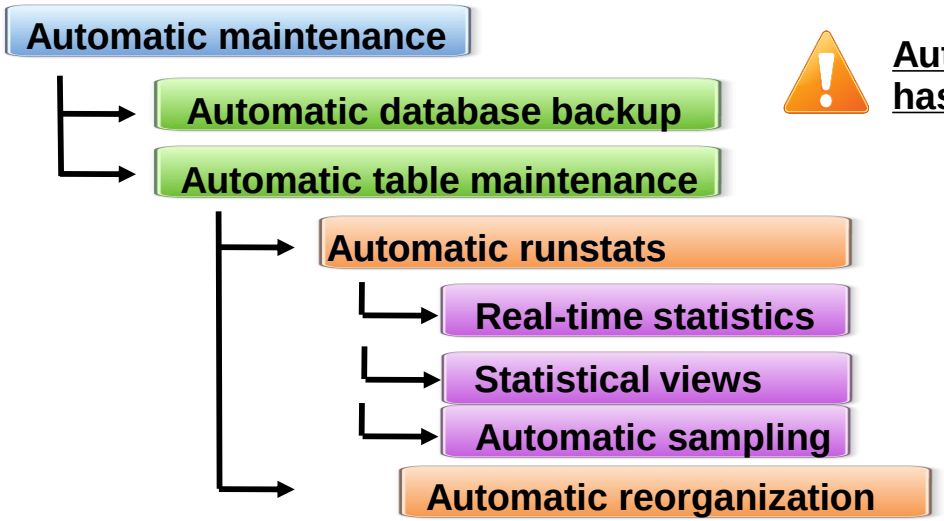
Self-Healing	Self-Configuring	Self-Optimizing	Self-Managing
• Data Studio web console	• Configuration Advisor • Design Advisor	• Utility Throttling • Self-tuning Memory Manager	• Automatic Storage • Automatic Maintenance • Automatic Dictionary Creation

- **Time and cost savings**
  - It shifts the burden of managing a computing environment from DBAs to technology
  - Over 70% of IT budgets are consumed by labor costs

Automatic Object Maintenance

- Automatic Maintenance for statistics, reorganizations and other tasks
  - **AUTO_MAINT** parameter is the **master on/off switch**
    - Individual Child parameters can be set to ON/OFF and the settings are persisted in the database configuration file
- Set maintenance policy via
  - Database configuration
  - Data Studio
  - Stored Procedures
    - SYSPROC.AUTOMAINT_SET_POLICY
    - SYSPROC.AUTOMAINT_SET_POLICYFILE

} Sample XML configuration samples located in: *$YOURINSTANCEHOME/sqllib/samples/automaintcfg*



Automatic statistics profiling (auto_stats_prof) has been deprecated

Automatic Object Maintenance Parameters

- **Automatic Statistics**
  - **Real-time statistics**
    - Enables/disables collection of real-time stats
  - **Statistical views**
    - Automatically maintain statistical views
  - **Automatic sampling**
    - Control the use of sampling when collecting stats on a large table, sampling rate is automatically determined
- **Automatic Reorg**
  - Automate the reorganization of tables and indexes
  - A reorganization policy may be used to specify the behavior
- **Automatic Backup**
  - A backup policy may be used to specify the automated behavior

Automatic Statistics Collection

- **Enabled by default at database creation**
  - Automatic background statistics collection **auto_runstats**
  - Automatic real-time statistics collection **auto_stmt_stats**
- The **query optimizer** determines how statistics should be collected
  - Based on the query and amount of table update activity
- **Asynchronous statistics collection**
  - Collect statistics that are available to run in the background using runstats utility
- **Real-time statistics collection (synchronous statistics collection)**
  - Provide timely and more accurate statistics at statement compilation
  - Statistics can be *fabricated* using certain meta-data.
  - Statistics can be maintained by the index and data manager and stored directly in the catalog.

Self-Tuning Memory Manager (STMM)

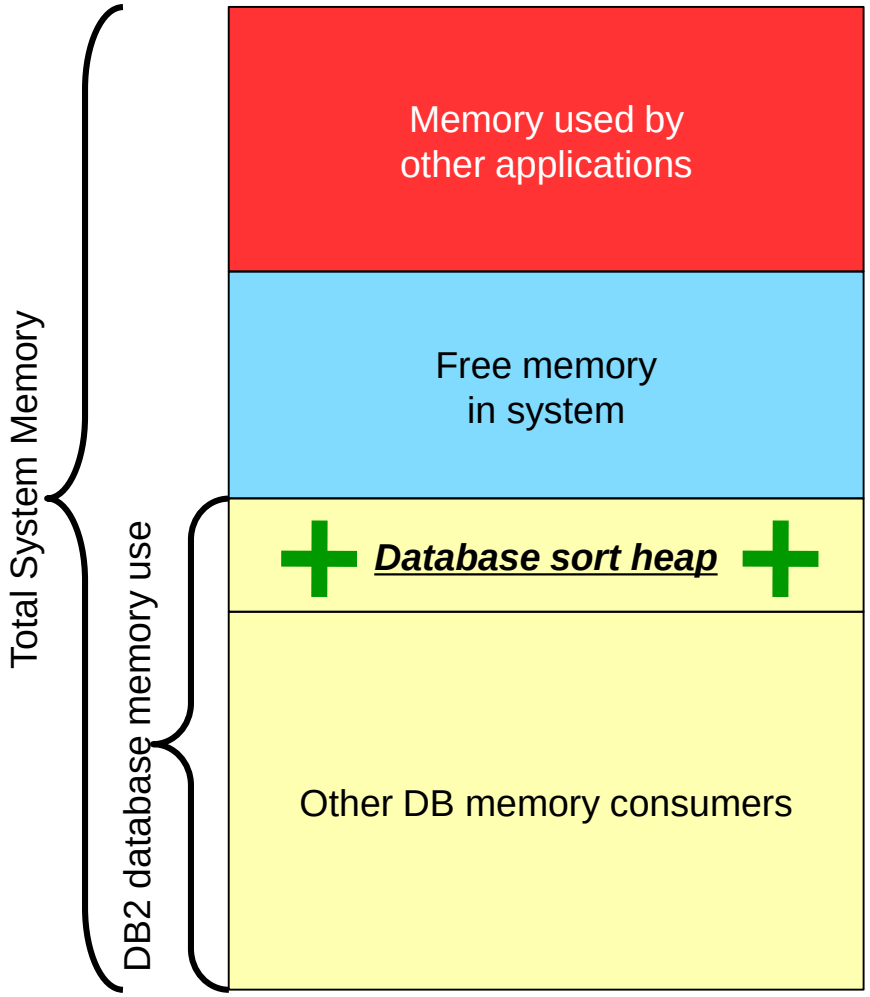
- Optimizes the performance of your database by automatically adjusting the values of:
  - **Total instance memory**
  - **Sort heap, lock list, package cache, SQL statement heap, application heap, application memory size, and total DB memory**
  - **Size of buffer pools**
- STMM is ON by default for all new databases since DB2 9
  - Works with the Database Partitioning Feature (DPF)
  - All affected memory parameters are fully dynamic (does not require instance restart)
- The database configuration parameter **SELF_TUNING_MEM** is the master switch for STMM
  - Configuration parameters and buffer pool sizes should be set to **AUTOMATIC** to enable

```
jasoncha@ncis:~
File  Edit  View  Terminal  Tabs  Help
db2inst1@ncis:~> db2 get db cfg for magnum |grep AUTOMATIC
 Size of database shared memory (4KB)  (DATABASE_MEMORY) = AUTOMATIC(87040)
 Max storage for lock list (4KB)              (LOCKLIST) = AUTOMATIC(6200)
 Percent. of lock lists per application       (MAXLOCKS) = AUTOMATIC(60)
 Package cache size (4KB)                   (PCKCACHESZ) = AUTOMATIC(1533)
 Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = AUTOMATIC(8892)
 Sort list heap (4KB)                         (SORTHEAP) = AUTOMATIC(1778)
 Database heap (4KB)                            (DBHEAP) = AUTOMATIC(2339)
 SQL statement heap (4KB)                     (STMTHEAP) = AUTOMATIC(4096)
 Default application heap (4KB)             (APPLHEAPSZ) = AUTOMATIC(256)
 Application Memory Size (4KB)              (APPL_MEMORY) = AUTOMATIC(40000)
 Statistics heap size (4KB)                  (STAT_HEAP_SZ) = AUTOMATIC(4384)
```

STMM Operating Modes

- DATABASE_MEMORY = AUTOMATIC

1)A change in workload occurs that requires more memory for sorts



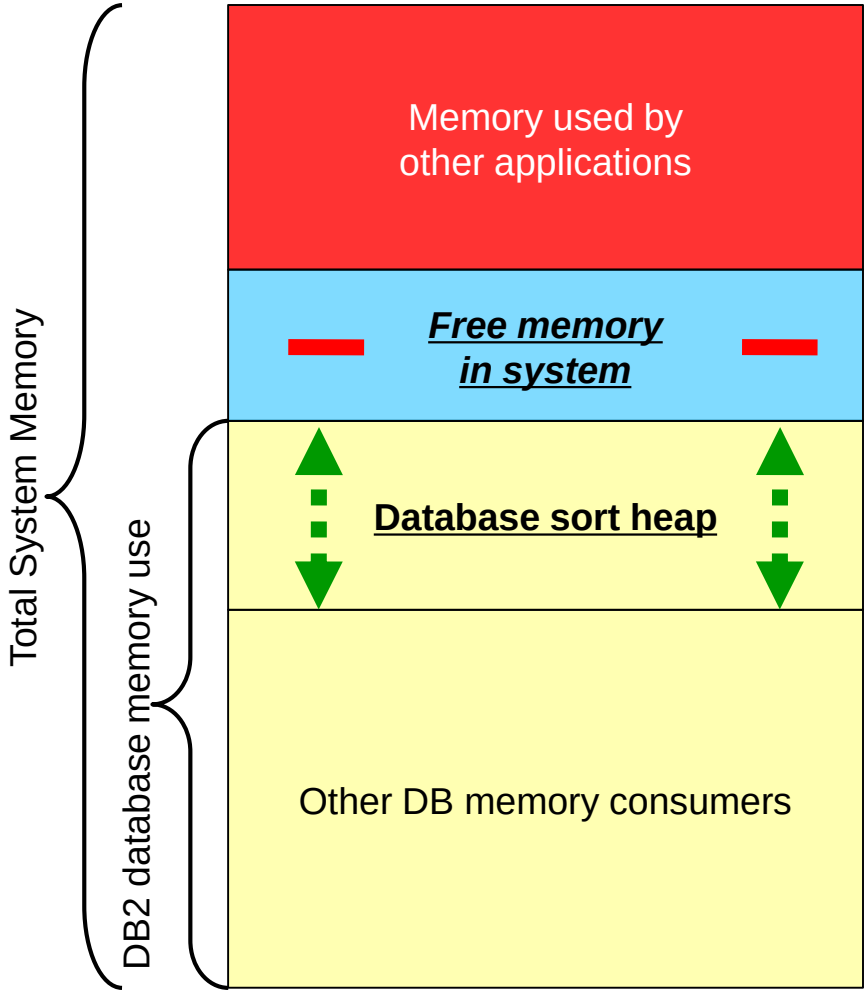
STMM Operating Modes

- DATABASE_MEMORY = AUTOMATIC

1)A change in workload that requires more memory for sorts

2)DB2 requests and gets more memory from the OS

- **Free memory in the system shrinks**
- **DB2 uses newly acquired memory in sort heap**

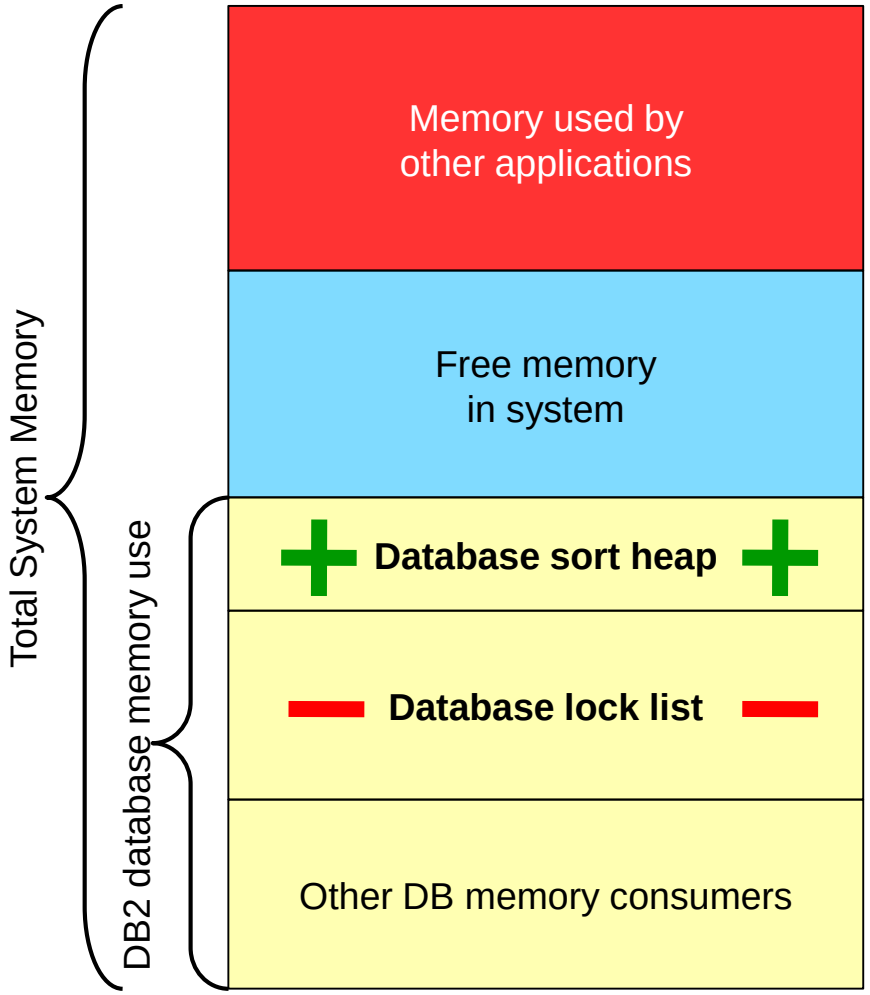


STMM Operating Modes

- DATABASE_MEMORY = COMPUTED or <number>

1)Scenario involves a change in workload that now requires more memory for sorts

2)DB2 database is set at fixed memory usage, and thus cannot take memory from OS. Therefore identifies another memory consumer in the database that does not need its memory anymore (e.g. lock list)



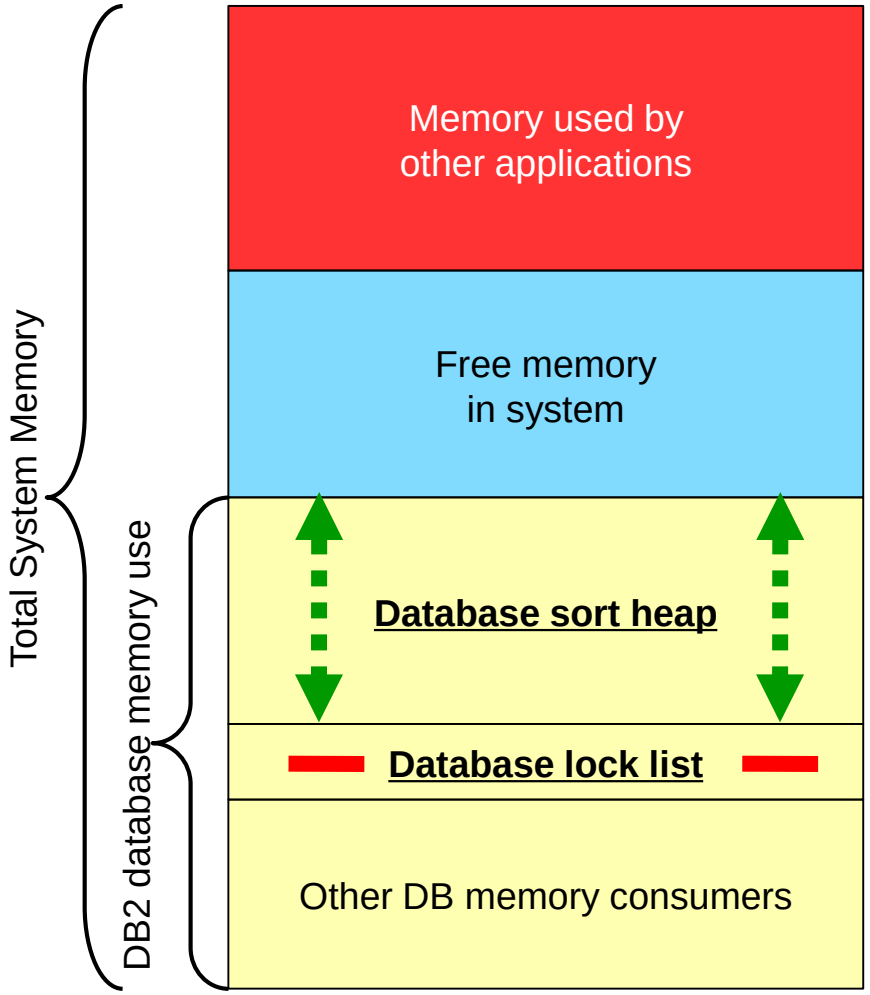
STMM Operating Modes

- DATABASE_MEMORY = COMPUTED or <number>

1)Scenario involves a change in workload that now requires more memory for sorts

2)DB2 database is set at fixed memory usage, and thus cannot take memory from OS. Therefore identifies another memory consumer in the database that does not need its memory anymore (e.g. lock list)

3)The memory is transferred between the memory consumers. The overall memory usage for this DB2 database stays the same.

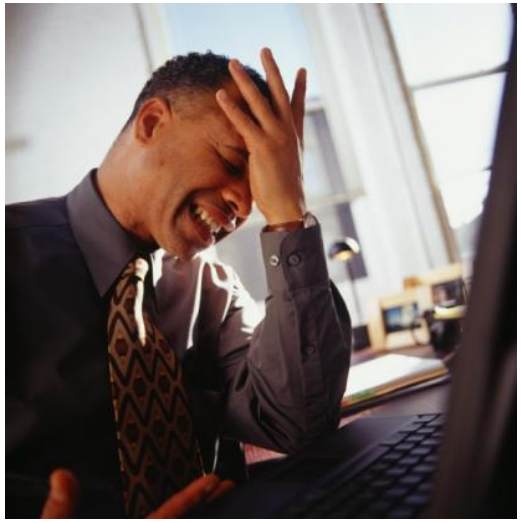


议程

- DB2基本维护和自理功能
- DB2数据库备份和恢复功能
  - 数据库备份
  - 数据库恢复
- DB2数据导入导出
- 轻松将Oralce迁移到DB2数据库

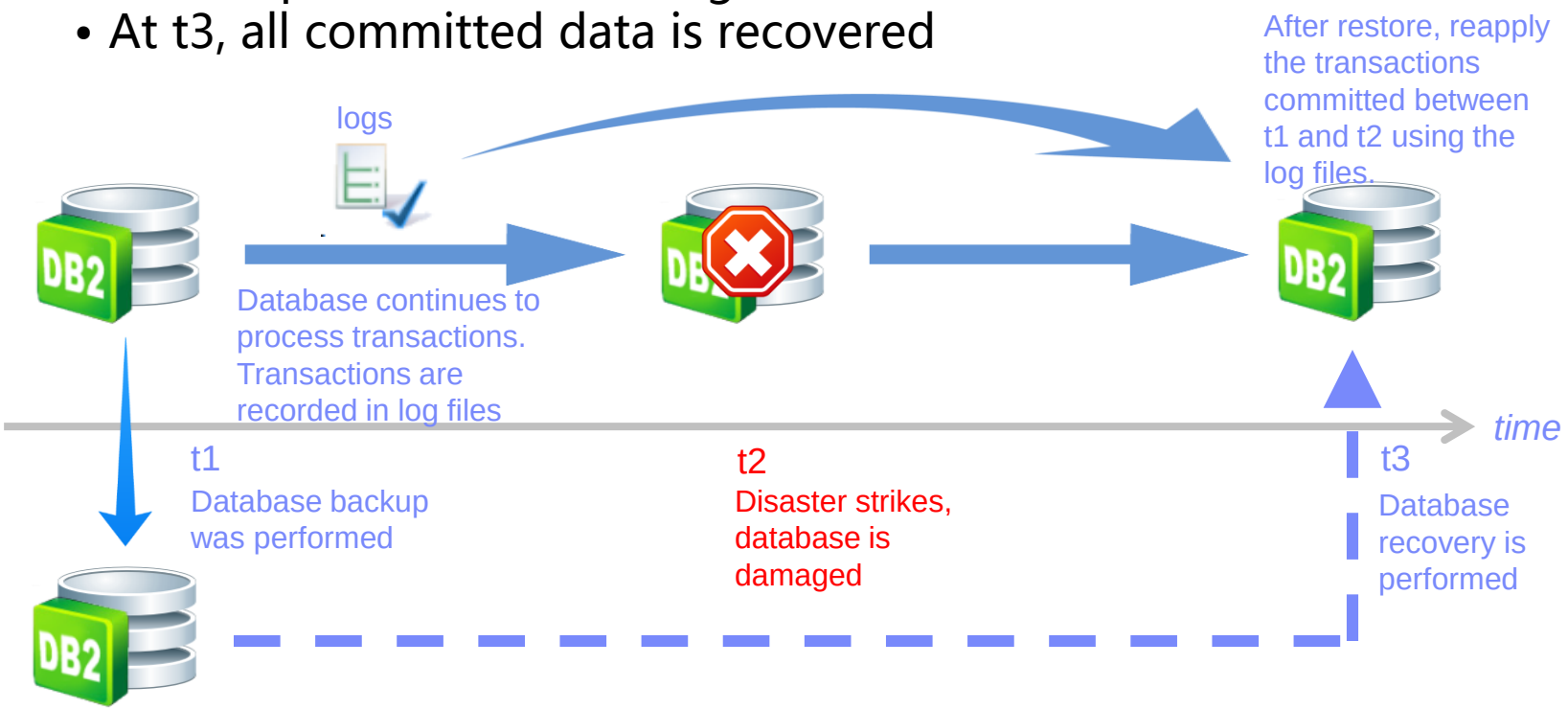
Why Backup Data

- Backing up data is vital for businesses
  - Lost information can cause a major crisis or worse, lead to business failure.
- Common problems:
  - System outage
    - Power failure
    - Hardware failure
  - Transaction failure
    - Users may inadvertently corrupt the database
  - Media failure
    - Disk drive becomes unusable
  - Disaster
    - Database facility damaged by fire, flooding or other catastrophe
- DB2 backup and recovery methods are designed to help you keeping your information safe!



Basic Concept of Backup and Recovery

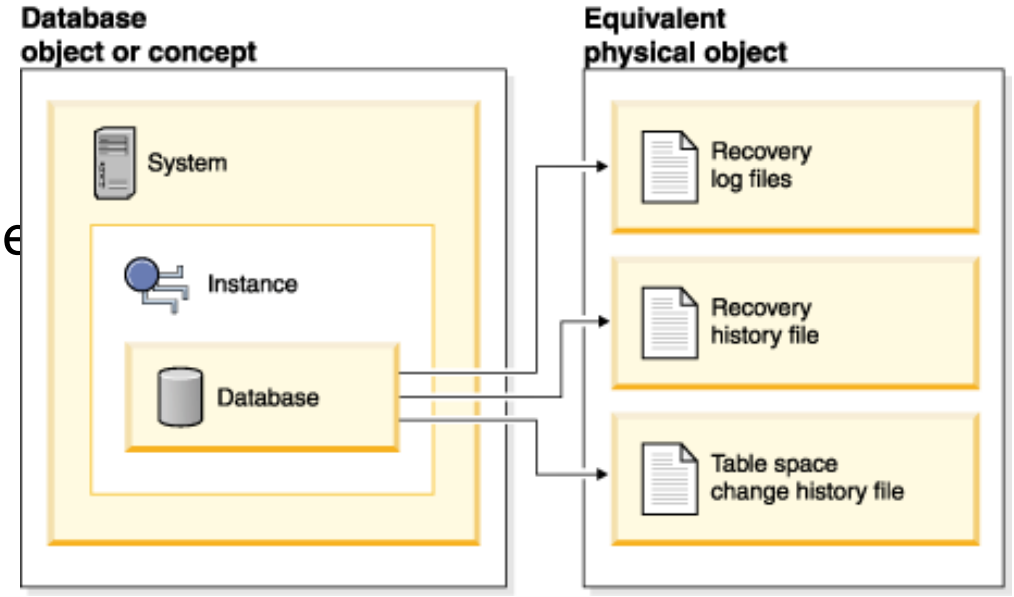
- Process of making copies of data and which may be used to restore the original in case of failure or loss of data
  - E.g.
    - At t1, a database backup operation is performed
    - At t2, a problem that damages the database occurs
    - At t3, all committed data is recovered



Database Logging

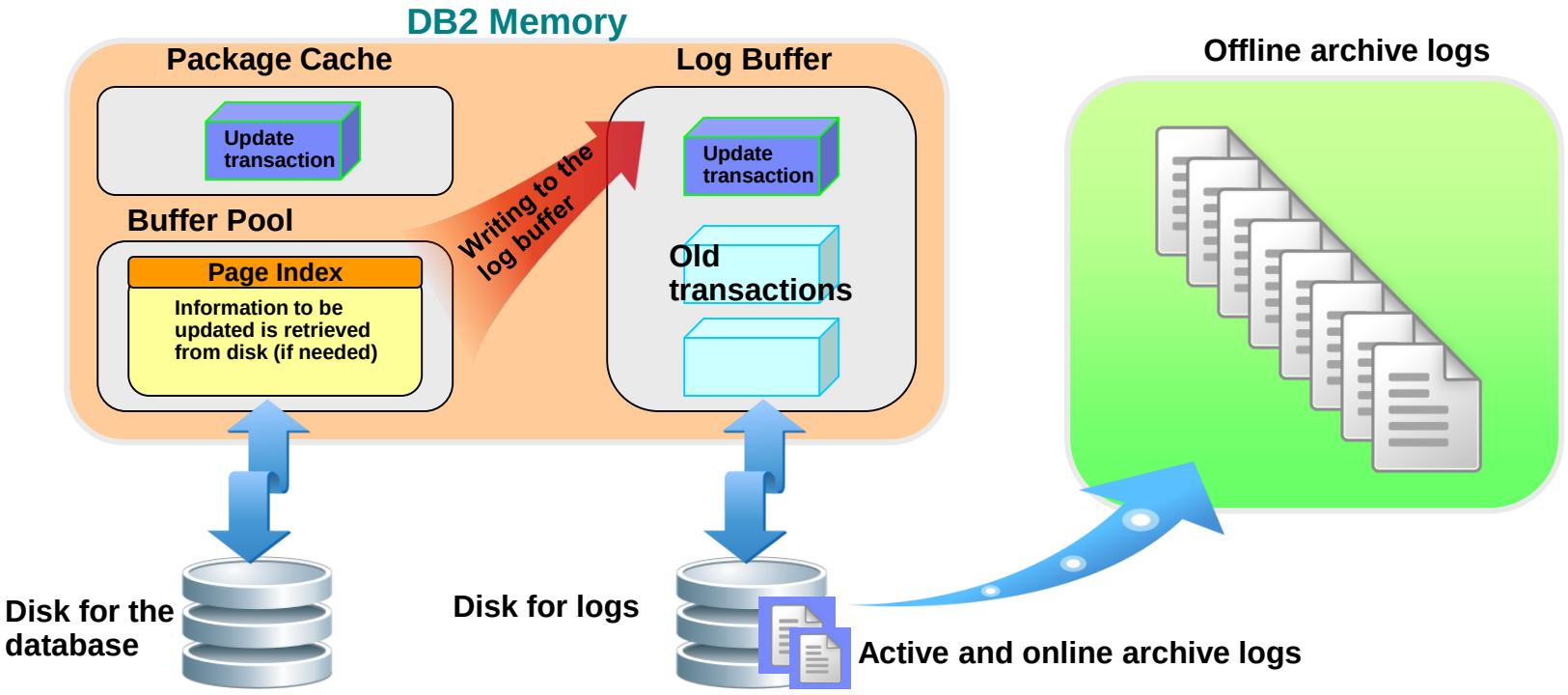
- Keep track of changes made to database objects and data
- During the recovery process, DB2 examines these logs and decides which changes to redo or undo
- Key element of any high availability strategy
- Can be stored in files or on devices
- Log records are written to disk when one of the
- following occurs:
  - Log buffer is full;
  - A transaction commits;
  - A group of transactions
  - commit, as defined by the
  - `mincommit` value

Deprecated in DB2 10



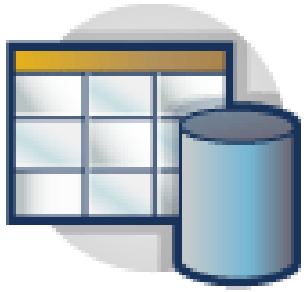
Log File States

- Active logs
  - Contain at least 1 transaction that has not been committed or rolled back
- Online archive logs
  - Contain committed and externalized transactions in the active log directory
- Offline archive logs
  - Contain committed and externalized transactions in a separate repository



Database Backup

- Copy of a database or table space
  - User data
  - DB2 catalogs
  - All control files, e.g. buffer pool files, table space file, database configuration file

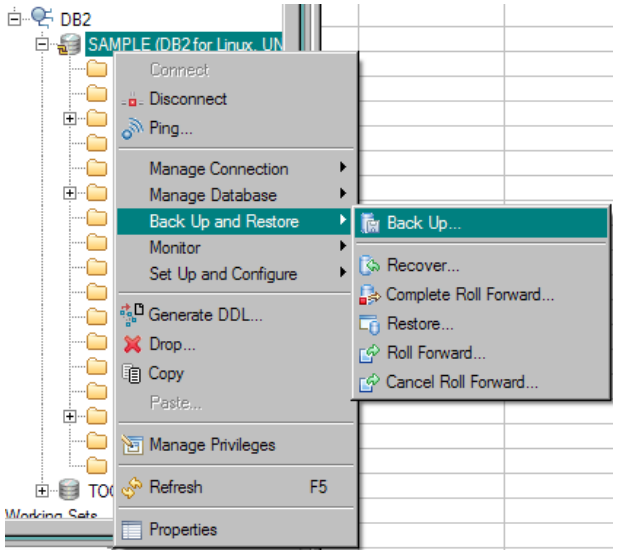


- Backup modes:
  - **Offline Backup**
    - Does not allow other applications or processes to access the database
    - Only option when using circular logging
  - **Online Backup**
    - Allows other applications or processes to access the database
    - Available to users during backup
    - Can backup to disk, tape, TSM and other storage vendors

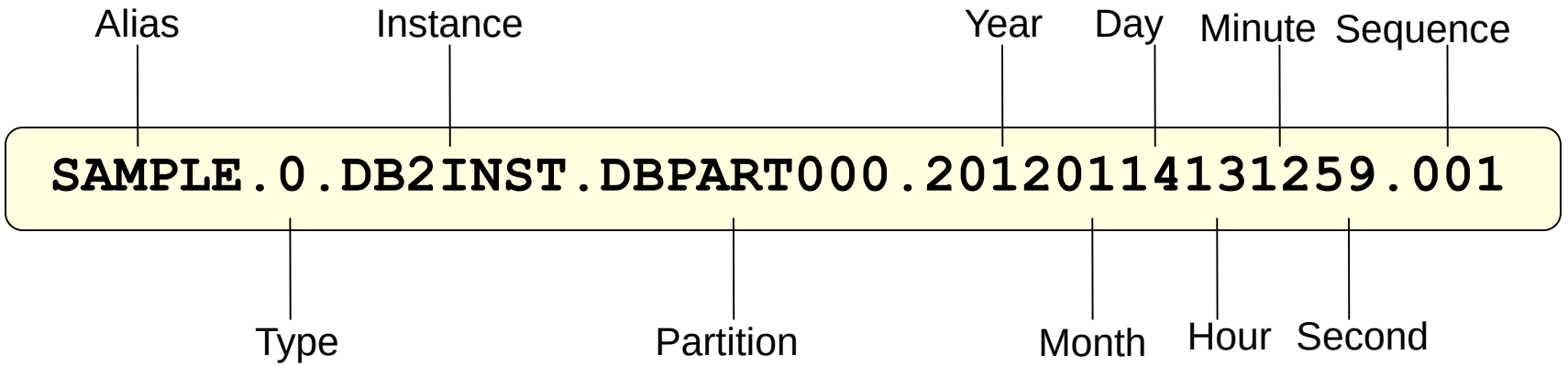
Database Backup

- Command-line:
- `db2 backup database <db_name> <online> to <dest_path>`
  - Online backup example
- `db2 backup database mydb online to /home/db2inst1/backups`
  - Offline backup example
- `db2 backup database mydb to /home/db2inst1/backups`
- IBM Data Studio:

1. Connect to target database
2. Right click on database
3. Select “Back Up and Restore” then “Back Up”



Database Backup – File Naming Convention



Backup Type:

0 = Full Backup

3 = Table space Backup

4 = backup image generated by the LOAD COPY TO command

Backing Up Partitioned Databases

- There are four possible ways:
  - Back up each database partition one at a time
  - Use the db2_all command
  - Run a single system view – SSV (Recommended)

```
ON DBPARTITIONNUMS
```

  - Use backup task assistant in IBM Data Studio

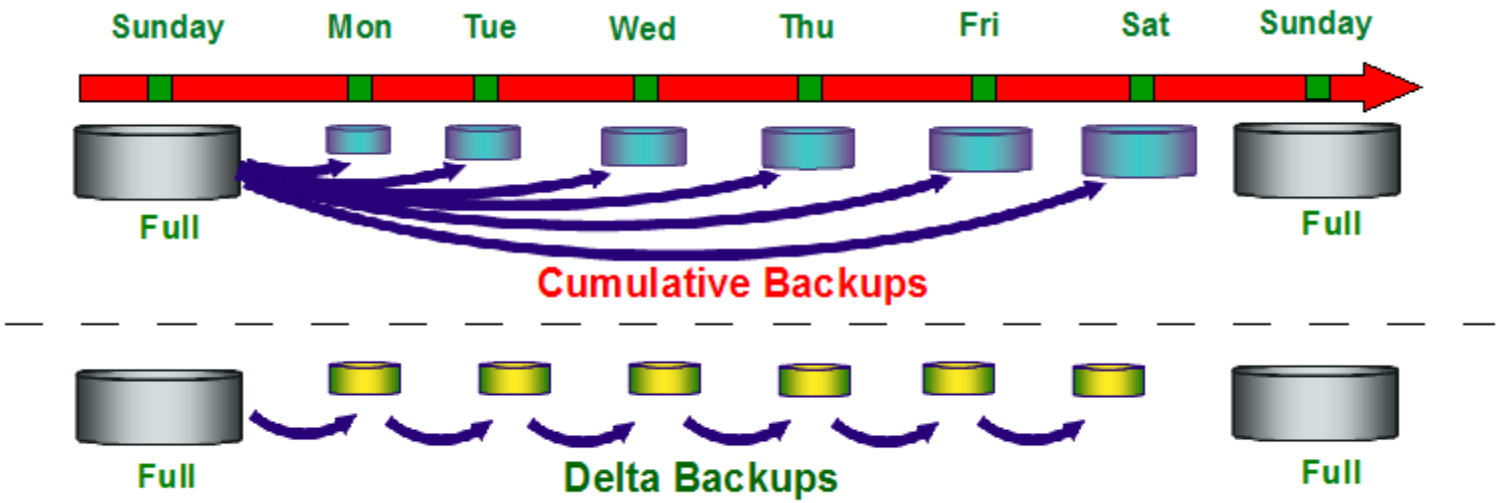
```
db2 backup database mydb1 ON ALL DBPARTITIONNUMS
db2 backup database mydb1 ON DBPARTITIONNUMS (2,4)
```

Table Space Backup

- Enables user to backup a subset of database
- Multiple table spaces can be specified
- Database must be using archive logging
- Table space backup can run in both online and offline backup
- Table space can be restored from either a database backup or table space backup of the given table space
- Use the keyword TABLESPACE to specify table spaces
- Supported in DB2 pureScale environment

```
db2 backup database mydb1 TABLESPACE (TBSP1) ONLINE to
/home/db2inst1/backup
```

Incremental Backups



- Incremental (a.k.a. cumulative) - Backup of all database data that has changed since the most recent, successful, full backup operation
- Incremental Delta - Backup of all database data that has changed since the last successful backup (full, incremental, or delta) operation.
- Requires `TRACKMOD` database configuration parameter set to ON
  - Supports both database and table space backups.
- Suitable for large databases, considerable savings by only backing up incremental changes.

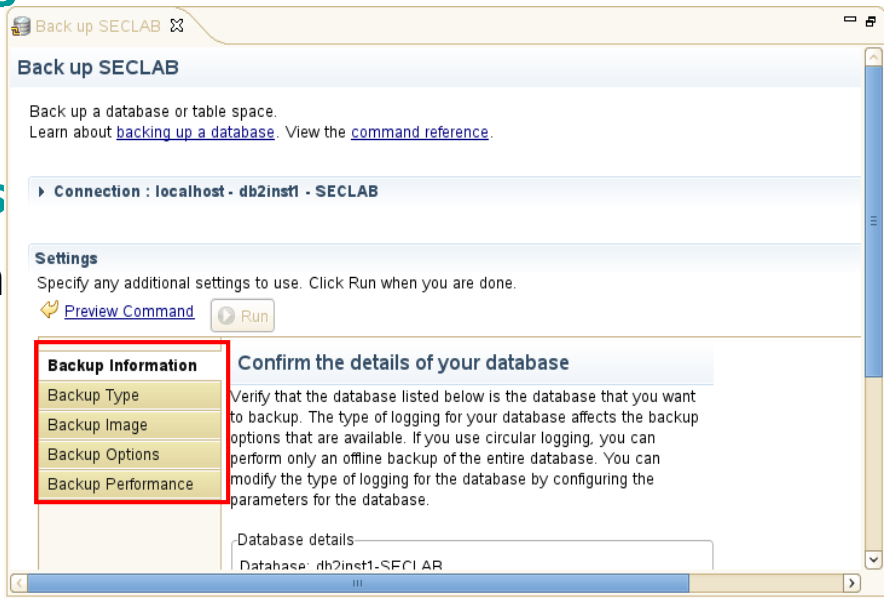
Database Backup – Compression

- DB2 backups can be automatically compressed
  - Significantly reduce backup storage costs
- Performance characteristics
  - CPU costs typically increased (due to compression computation)
  - Media I/O time typically decreased (due to decreased image size)
  - Overall backup/restore performance can increase or decrease; depending on whether CPU or media I/O is a bottleneck
- E.g.

```
–db2 backup database DS2 to /home/db2inst1/backups compress
```

Automatic Database Backup

- Simplifies database backup management tasks for the DBA
- Ensures that the database is backed up both properly and regularly
- To configure automatic backup
  - **DB configuration parameters**
    - AUTO_DB_BACKUP
    - AUTO_MAINT
  - **Graphical user interface tools**
    - Data Studio's "Backup Assistant"
  - **System stored procedure**
    - AUTOMAINT_SET_POLICY



Optimizing Backup Performance

- During a backup operation, DB2 automatically chooses an optimal value for:
  - **PARALLELISM** *n*
    - Number of table spaces backed up in parallel
  - **WITH** *num_buffers* **BUFFERS**
    - Number of buffers used
    - Use at least **twice as many buffers** as backup targets (or sessions) to ensure that the backup target devices do not have to wait for data.
  - **Buffer** *buffer-size*
    - Backup buffer size
- Allocate more memory to backup utility by increasing utility heap size (`UTIL_HEAP_SZ`) configuration parameter.
- Backup subset of data where possible:
  - Table space backups
  - Incremental backups
- Use multiple target devices

DB2CKBKP – Check Backup

- This utility can be used to test the integrity of a backup image
  - determine whether the image can be restored.
  - display the meta-data stored in the backup header.

```
$ db2ckbkp -h SAMPLE.0.moba.NODE0000.CATN0000.20041008013428.001

=====================
MEDIA HEADER REACHED:
=====================
        Server Database Name          -- SAMPLE
        Server Database Alias         -- SAMPLE
        Client Database Alias         -- SAMPLE
        Timestamp                     -- 20041008013428
        Database Partition Number     -- 0
        Instance                      -- moba
        Sequence Number               -- 1
        Release ID                    -- A00
        Database Seed                 -- 92DBF20F
        DB Comment's Codepage (Volume)-- 0
        DB Comment (Volume)           --
        DB Comment's Codepage (System)-- 0
        DB Comment (System)           --
        Authentication Value          -- 255
        Backup Mode                   -- 1
        Includes Logs                 -- 1
        Compression                   -- 0
 ・・・（略）・・・
```

This backup is an online backup with INCLUDE LOGS option
0: Not included in the log file
1: contains log file

Backup is not compressed
0: not compressed
1: compressed

Database Recovery

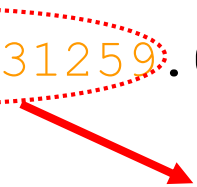
- Recovery is the rebuilding of a database or table space after a problem such as media or storage failure, power interruption, or application failure.
- Types of Recovery
  - **Crash or restart recovery**
    - Protects the database from being left inconsistent
  - **Version recovery**
    - Restores a snapshot of the database
  - **Roll forward recovery**
    - Extends version recovery by using full database and table space backup in conjunction with the database log files
- Crash recovery and Version recovery are enabled in DB2 by default

DB2 Restore Utility

- Restores database or table space from a previously taken backup
- Invoked using:
  - The **Restore Database** command
  - The **db2Restore** API
  - IBM Data Studio **Restore** task assistant
- `TAKEN AT` - Specify the time stamp of the database backup image
- Without prompting – Overrides any warnings

- Example:

- SAMPLE.0.DB2INST.DBPART000.20120114131259.001



```
RESTORE DATABASE dbalias FROM <db_path> TAKEN AT 20120114131259
```

Table Space Restore Operation

- Table space restore occurs in offline and online mode
  - Other table spaces can be used concurrently while restore in progress
- Restored table space is in Roll Forward Pending state
  - can be either rolled forward to **End of Logs** or a **Point In Time**.
- Minimum recovery time can be checked using
  - `db2 list tablespaces show detail`
- User table space must be in line with catalog table space
  - E.g.: if catalog indicates table T1 exists in table space TSP1, table T1 must exist in the TSP1 table space, otherwise database becomes inconsistent
- Recommended to take a table space backup after restore to a Point In Time
  - Transactions that come after the point in time are lost
  - backup required as new point of reference for future recoveries

Incremental Restore

- Restore a database with incremental backup images
- AUTOMATIC (recommended) - All required backup images will be applied automatically by restore utility
- MANUAL – User applies the required backups manually
  - **db2ckrst** can provide the sequence for applying backups
- ABORT - aborts an in-progress manual cumulative restore

```
RESTORE DATABASE sample INCREMENTAL AUTOMATIC FROM /db2backup/dir1;

ROLLFORWARD DATABASE sample TO END OF LOGS AND COMPLETE;
```

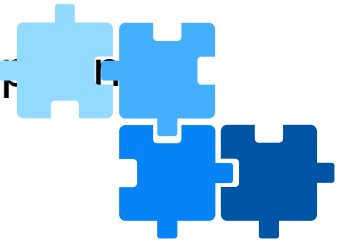
Redirected Restore

- A redirected restore operation is performed when:
  - Restore a backup image to a machine that is different than the source machine
  - Restore table space containers into a different physical location
  - One or more containers is inaccessible
  - Redefine the paths of a defined storage group
- Restrictions
  - Cannot use a redirected restore to move data from one operating system to another
  - Cannot create or drop a storage group during the restore process
  - Cannot modify storage group paths during a table space restore process
- Two-step database restore process with an intervening table space container definition step

  1) Issue **RESTORE DATABASE** command with **REDIRECT** option

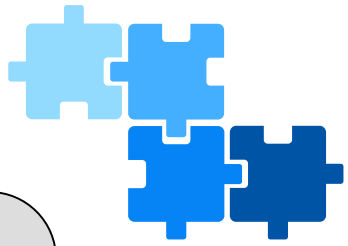
  2) Take one of the following steps:
  - **SET TABLESPACES CONTAINERS**
  - **SET STOGROUP PATHS**

  3) Issue **RESTORE DATABASE** command with **CONTINUE** option



Restore with Transport

- The `RESTORE DATABASE` command can transport a set of table spaces or schemas
  - The **`TRANSPORT`** option of the **`RESTORE`** command copies objects from a backup image to a database
  - Database objects are re-created to reference the new database and the data is restored
- Transporting a database schema involves:
  - Backup valid transportable sets from the source database
  - Restore transportable sets on the target db using the TRANSPORT option
  - Optionally, you can redirect the container paths for the table spaces being transported
- Restrictions
  - Database Schema must be transported in its entirety.
  - Source and Target should not share the same schema names



```
db2 restore db originaldb tablespace (mydata1)
schema(schema1,schema2) from <Media_Target_clause> taken
at <date-time> transport into targetdb redirect

db2 list tablespaces db2 set tablespace containers for
<tablespace ID for mydata1> using (path '/db2DB/data1')

db2 restore db originaldb continue
```

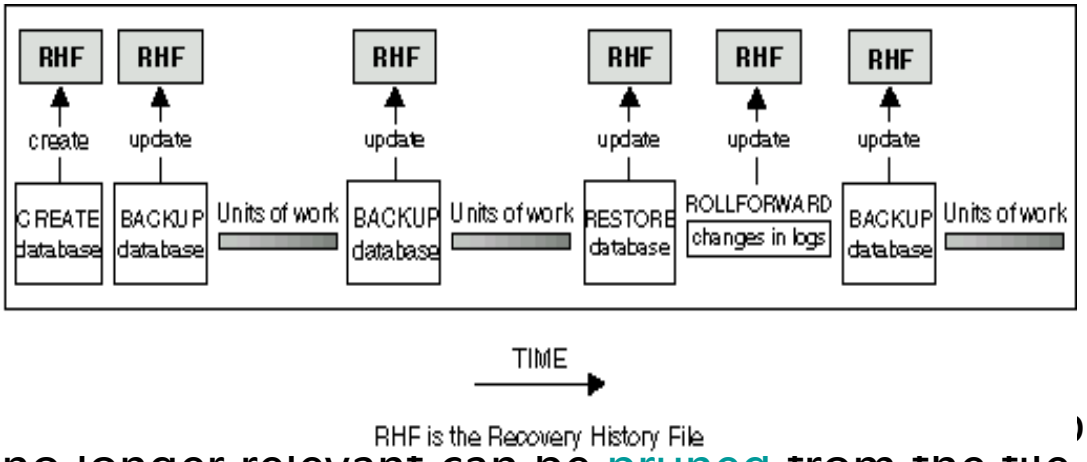
Recovery History File

- Purpose
  - In addition to recovery log files, the recovery history file is also created automatically when a database is created.
  - Used to recover all or part of a database to a point in time using the summarized backup information in this file
- View
  - To see the entries in the recovery history file:

```
LIST HISTORY
```

OR

```
SELECT * FROM SYSIBMADM.DB_HISTORY
```



- Pruning
  - Rec ut entries that are no longer relevant can be pruned from the file using command `PRUNE HISTORY`

DB2 Recover Utility

- DB2 Recover Utility:
  - Performs restore and rollforward operations to recover a database to a specified time, based on information found in the recovery history file or to the end of logs
  - Can be issued for an incomplete recover operation that ended during the rollforward phase, the recover utility attempts to continue the previous recover operation without redoing the restore phase
  - Automatically establishes connection to the database
- In IBM® Data Studio Version 3.1 or later, you can use the task assistant for recovering databases
- To invoke the recover utility use
  - RECOVER DATABASE command
  - db2Recover application programming interface (API)
- Restrictions
  - Do not support Tablespace & Incremental restore operations
  - If it is interrupted during the restore phase, it cannot be continued. RECOVER command has to be re-issued

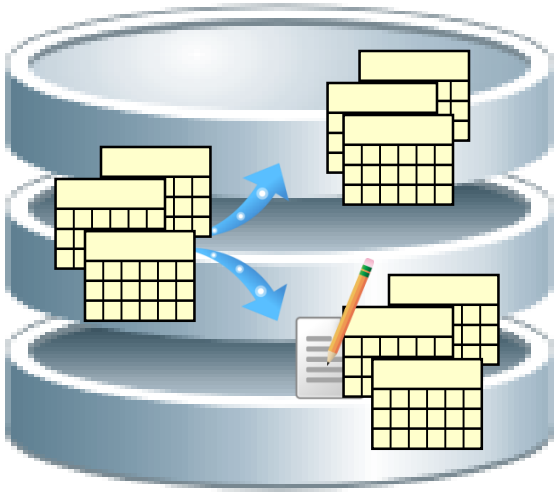
```
db2 RECOVER DATABASE SAMPLE TO 2001-12-31-04.00.00
db2 RECOVER DATABASE SAMPLE to END OF LOGS
```

议程

- DB2基本维护和自理功能
- DB2数据库备份和恢复功能
  - 数据库备份
  - 数据库恢复
- DB2数据导入导出
- 轻松将Oralce迁移到DB2数据库

Table Maintenance Procedure – ADMIN_COPY_SCHEMA

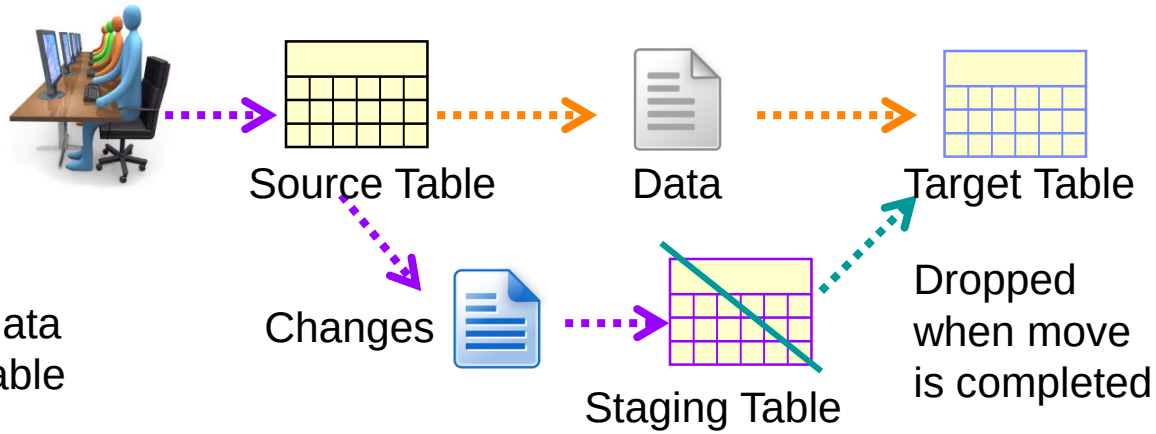
- The **ADMIN_COPY_SCHEMA** procedure is used to copy a specific schema and its objects.



- Procedure can be run in 3 different copy modes:
  - 'DDL': create empty copies of all supported objects from the source.
  - 'COPY': Create copies and load data from the source tables to target. LOAD is done in NONRECOVERABLE mode.
  - 'COPYNO': LOAD is done in 'COPY NO' mode.

```
CALL SYSPROC.ADMIN_COPY_SCHEMA
('SOURCE_SCHEMA', 'TARGET_SCHEMA', 'COPY', NULL,
 'SOURCETBSP1 , TARGETTBSP1, 'ERRORTABSCHEMA', 'ERRORTABNAME')
```

Table Maintenance Procedure – ADMIN_MOVE_TABLES



- The **ADMIN_MOVE_TABLE** stored procedure moves the data in an active table into a new table
- Allows data to remain online and available for access.
- Two ways to use the procedure:
  - Move table while modifying certain parts of the table definition for the target table

```
CALL SYSPROC.ADMIN_MOVE_TABLE (… , 'MOVE')
```

  - Move table with greater control on table creation by allowing you to create the target table beforehand.

```
CREATE TABLE  (COL1 VARCHAR(5))

CALL SYSPROC.ADMIN_MOVE_TABLE (…, 'MOVE')
```

DB2LOOK, Export, Import and LOAD Utilities

- **DB2LOOK**
  - Extract DDL statements Object by Object

```
db2look -d department -wlm -e -l
```

- **DB2 Export**
  - Extract data from table or view to files
  - Invoke from Data Studio, CLP, or the db2Export API via a user application
- **DB2 Import**
  - Import data from a file using SQL inserts
  - Invoke from Data Studio, CLP, or the db2Import API via a user application
- **DB2 LOAD**
  - Move large quantities of data efficiently into tables
  - 4 distinct phases: LOAD, BUILD, DELETE, INDEX COPY
  - With CURSOR file type
    - Load results of an SQL query directly into a target table
    - No intermediate storage of data needed
    - LOAD can be used across multiple databases in a CURSOR

Writes formatted pages directly into the database

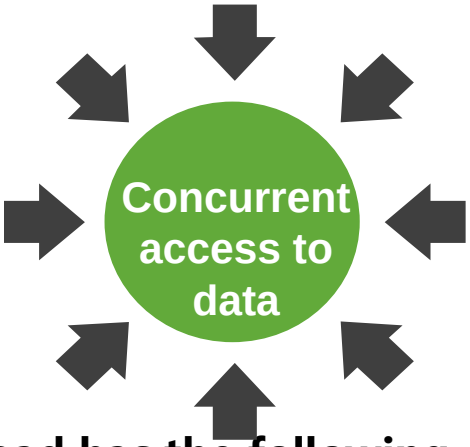
```
DECLARE C1 CURSOR DATABASE SRCDB USER user1 USING password FOR
SELECT * FROM SOURCE_TABLE;
LOAD FROM C1 OF CURSOR REPLACE INTO TARGET_TABLE;
```

- **DB2MOVE**
  - Utilizes the EXPORT/IMPORT/LOAD/COPY APIs using an internal staging file db2move.lst

```
db2move SRCDB COPY –sn “SANTA” -co target_db TESTDB –u db2admin –p password
```

Continuous Data Ingest

- **Data Warehouse environments now require *continuous data input* to allow:**



- **However, today's conventional batch load has the following drawbacks:**

Batch loads require a window where the database operations are locked out.

Batch loaders do not usually support loading from a continuous data stream

Nightly or weekly batch loads do not provide current enough data

Current Data Movement Utilities (up to DB2 v9.7)

IMPORT Utility

Best suitable for small amount of data

Uses a simple SQL INSERT statement

All rows are logged

LOAD Utility

Best suitable for large amount of data

Writes formatted DB2 pages directly into the database table

Minimal logging

- Both utilities have the following shortcoming:
  - Can’ t perform data transformation
  - Use of PIPES is limited to LOAD only
  - Processing is done at the DB2 server machine
  - Only the LOAD utility has an option to recover from failure

DB2 10 - INGEST utility

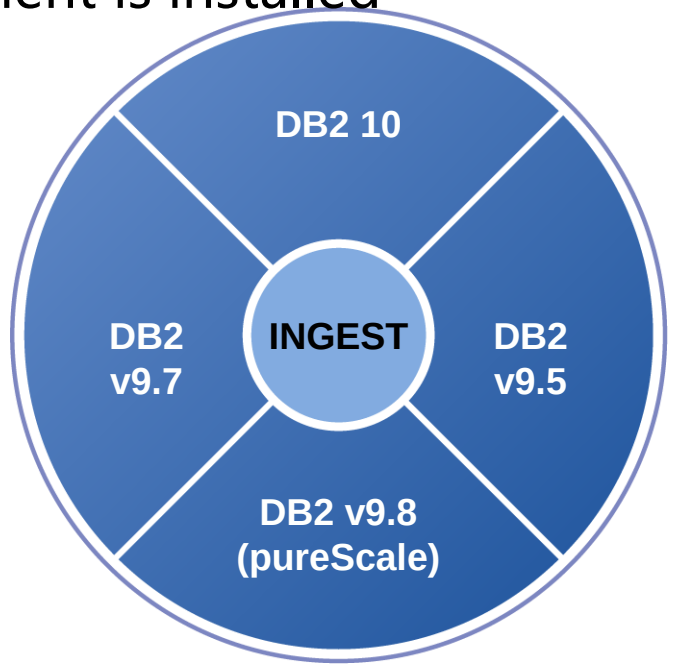
- **High-speed client-side DB2 utility that ingests data from files and pipes into DB2 LUW tables, using SQL-like commands.**

- **Advantages:**
  - Move and process large amounts of **real-time data** **without affecting availability**
    - Allows decisions to be based on the latest set of data
    - Increases data analysis capabilities
    - DPF-aware: routes rows to correct partition

No need to choose between data concurrency and availability!

Setup/Installation/Compatibility

- **Part of the DB2 Client**
  - It can be used anywhere where the DB2 client is installed
- **Possible installation location**
  - Existing DB2 server
  - New dedicated (separate) INGEST server
  - Existing shared server for ETL
  - DPF: coordinator partition
- **No additional license required!**



- **It can be used against following down releases of DB2:**
  - v9.5
  - v9.7
  - v9.8 (pureScale)

Supported Data Inputs

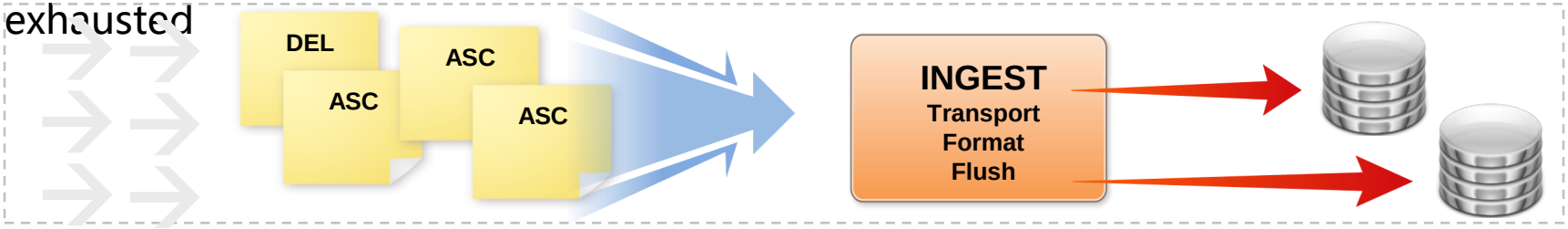
- **Input data formats:**
  - **Delimited ASCII (DEL) Format**
    - Stream of characters separated by row and column delimiters

```
1, "Mark Kate", "DB2 Specialist", "IBM Canada"
2, "John Doe", "DB2 Specialist", "IBM Canada"
3, "Steven Johns", "Manager", "IBM London"
```

  - **Fixed Format ASCII (ASC)**
    - Each column length in the file has the same length of the column definition

```
1   Mark Kate          DB2 Specialist    IBM Canada
2   John Doe           DB2 Specialist    IBM Canada
3   Steven Johns       Manager           IBM London
```

- Continuously pumps data into DB2 tables using SQL arrays until the source is exhausted



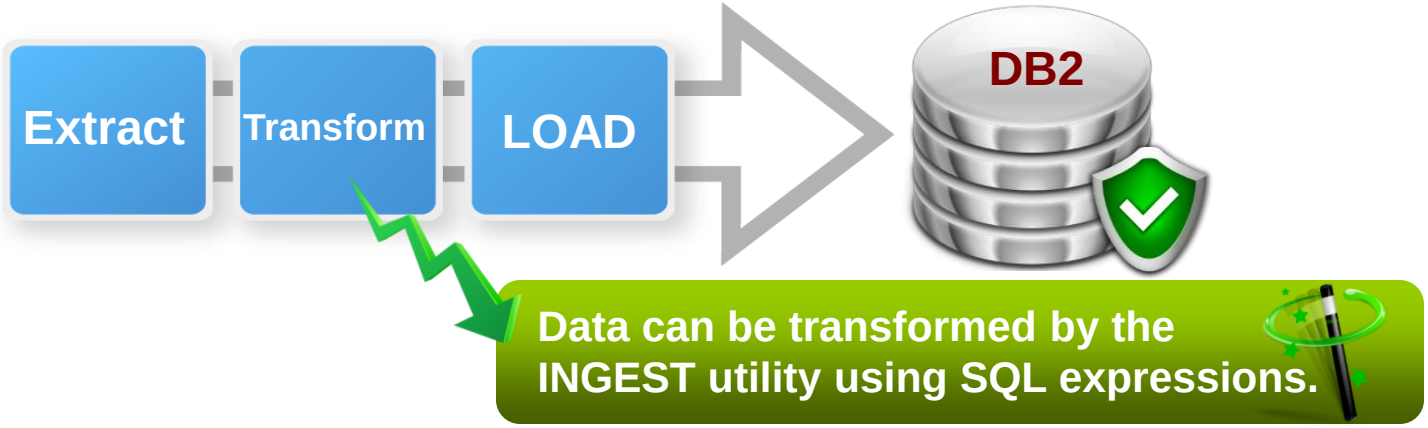
Supported Operations

- **DML Operations**
  - INSERT, UPDATE, DELETE, MERGE (REPLACE option available)

```
INGEST FROM FILE my_file.del FORMAT DELIMITED
   ( $key_fld1 INTEGER EXTERNAL,
     $key_fld2 INTEGER EXTERNAL,
     $data_fld1 CHAR(8),
     $data_fld2 CHAR(8),
     $data_fld3 CHAR(8) )
   UPDATE my_table SET (data1_col, data2_col, data3_col) = ($data_fld1,
$data_fld2, $data_fld3)
   WHERE (key_col1 = $key_fld1) AND (key_col2 = $key_fld2);
```

IMPORT and LOAD DO NOT support data transformation

- **Lightweight ETL (Extract, Transform, Load)**
  - SQL expressions including basic predicates and casting ✅



INGEST Command Examples

-
```
INGEST FROM FILE my_file.txt FORMAT DELIMITED INSERT INTO my_table;
```

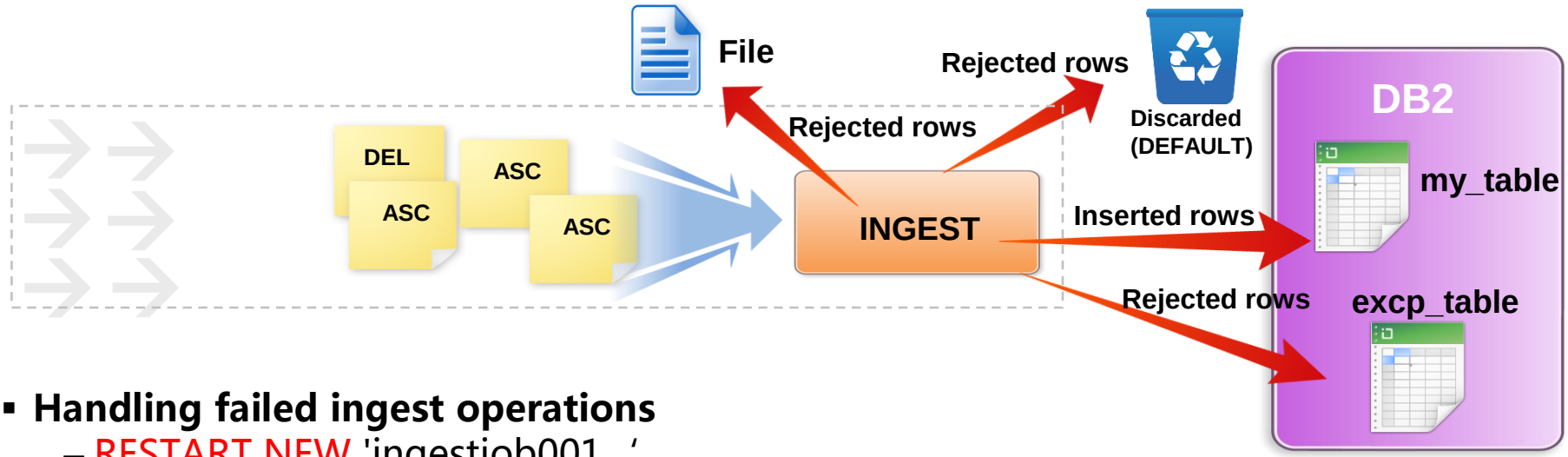
-
```
INGEST FROM FILE my_file.txt FORMAT POSITIONAL(
        $field1 POSITION(1:8) INTEGER EXTERNAL,
        $field2 POSITION(10:19) DATE 'yyyy-mm-dd',
        $field3 POSITION(25:34) CHAR(10))
INSERT INTO my_tableVALUES($field1, $field2, $field3);
```

-
```
INGEST FROM PIPE mypipe FORMAT DELIMITED BY '/' (
        $prod_ID CHAR(8),
        $description CHAR(32),
        $price DECIMAL(5,2) EXTERNAL,
        $sales_tax DECIMAL(4,2) EXTERNAL,
        $shipping DECIMAL(3,2) EXTERNAL )
INSERT INTO my_table(prod_ID, description, total_price)
        VALUES($prod_id, $description, $price + $sales_tax + $shipping);
```

Recoverability

- **Rejected rows can be discarded or placed into a file or table.**

```
INGEST FROM FILE my_file.txt FORMAT DELIMITED (
          $field1 INTEGER EXTERNAL,
          $field2 DATE 'mm/dd/yyyy',
          $field3 CHAR(32) )
EXCEPTION TABLE excp_table MESSAGES messages.txt
INSERT INTO my_table VALUES($field1, $field2, $field3);
```



- **Handling failed ingest operations**
  - RESTART NEW 'ingestjob001 '
  - RESTART CONTINUE 'ingestjob001
  - RESTART TERMINATE 'ingestjob001'

Comparison Between INGEST, LOAD and IMPORT

- **Supported Table Types**

Object	INGEST	LOAD	IMPORT
Detached table	☐	☐	☐
Global Temporary Table	☐	☐	☐
Multidimensional clustering table	☑	☑	☑
Materialized query table (MQT) that is maintained by user	☑	☑	☑
Nickname	☑	☐	☑
Range-clustered table	☑	☐	☑
Range-partitioned table	☑	☑	☑
Summary table	☑	☑	☑
Typed table	☐	☐	☑
Untyped (regular) table	☑	☑	☑
Updatable view (except typed view)	☑	☐	☑

When To Use INGEST

- Use **INGEST** when any of the following is true:
  - You need other applications to update the table while it is being loaded.
  - The input file contains fields you want to skip over.
  - You need to specify an SQL statement other than INSERT.
  - You need to specify an SQL expression (to construct a column value from field values).
  - You need to recover and continue on when the utility gets a recoverable error.
- **Performance vs. LOAD**:
  - **50% faster** than to load into a staging table *followed by* multiple *sequential* INSERT/SELECTs from the staging table to the target table**

****Tests were run inserting into a table with**:
- 100 byte rows
- 4 indexes
- an 8 partition D or E class BCU with a 4 core admin node and 2 data nodes with 4 cores each
- From 30 million to 500 million rows

Continuous Data Ingest Summarized

- **INGEST utility meets modern DW requirements by processing data from a continuous data stream**
  - Minimizes impact to concurrent use with no table locking
  - Performs data transformation which is not possible with IMPORT or LOAD
  - Unwanted rows can be placed into an exception file or table, or just discarded
- **Better Recoverability**
  - Failed INGEST operations can be restarted from the last commit
- **Part of DB2 Client, No additional license required**
  - It can be installed and executed in a separate machine from the DB2 server

议程

- DB2基本维护和自理功能
- DB2数据库备份和恢复功能
  - 数据库备份
  - 数据库恢复
- DB2数据导入导出
- 松将Oralce迁移到DB2数据库

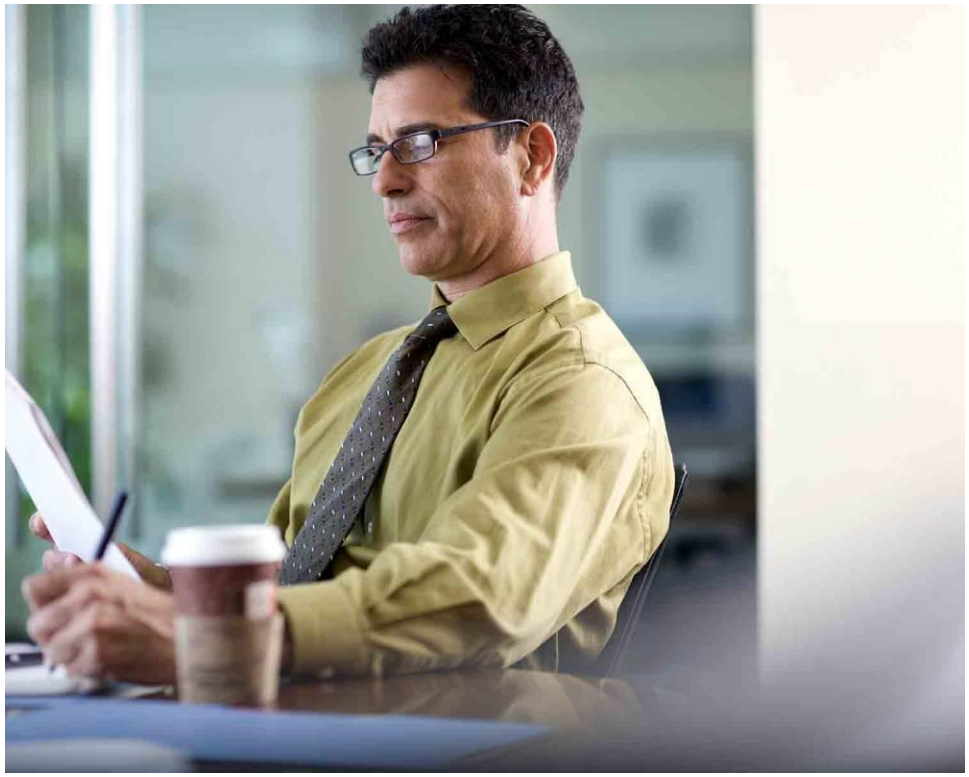
移植到IBM，您的新选择！

IBM在技术、方法学和工具方面的组合投资，加上大量的项目实践，能让你现在就开始摆脱束缚！

- 如何摆脱Sybase软件发展后续乏力，应用最新软硬技术，提升整体系统运营水平。
- 战略和专业知识：迁移到IBM比您想的容易的多！
- 开始行动: IBM Special Offers

大量客户移植至IBM软件是因为他们想…

- 降低基础架构费用
- 避免厂商锁定
- 最大化投资回报率
- 转向一个坚实的路线图
- … …



WHY - 为什么要移植到DB2?

- 我作为 客户
  - DB2是性价比最高的产品
  - 当前的数据库性能不能满足需求
  - 在同数据库供应商谈判时，没办法得到更优的价格
- 我作为 服务商
  - 我想扩展客户群
  - 我想得到IBM全方位的支持
  - 我的数据库供应商同时也是我的竞争对手
  - DB2是性价比最高的产品

DB2 for Linux, UNIX and Windows的商业价值

- ✓ **更少的管理需要** – 相对于其他DBMSs
  - 意味着需要更少的 DBA 资源从事日常维护工作
- ✓ **更好的性能** – 相对于其他DBMSs
  - 意味着在同等负荷需求下只需购买较少的CPU内核
- ✓ **更好的压缩能力** – 比其他DBMSs拥有
  - 意味更少的存储消耗以及提高性能
- ✓ **更佳的可用性** – 比其他DBMSs拥有
  - 意味着更少的停机时间，避免造成您商业机会/信誉的损失
- ✓ **更优秀的工作负荷管理** – 比其他DBMSs拥有
  - 意味着对应用程序的精确控制可以确保服务级别的准确调整和性能目标的达成
- ✓ **更好的开发能力** – 在极少甚至毫无变更下支持 PL/SQL 和 T-SQL
  - 如果您想合并 Oracle and/or Sybase 应用程序，你可以在DB2中完成 – 只需极少变更甚至无需变更！

降低TCO
无可比拟的自动化
深度压缩
闪电般快捷

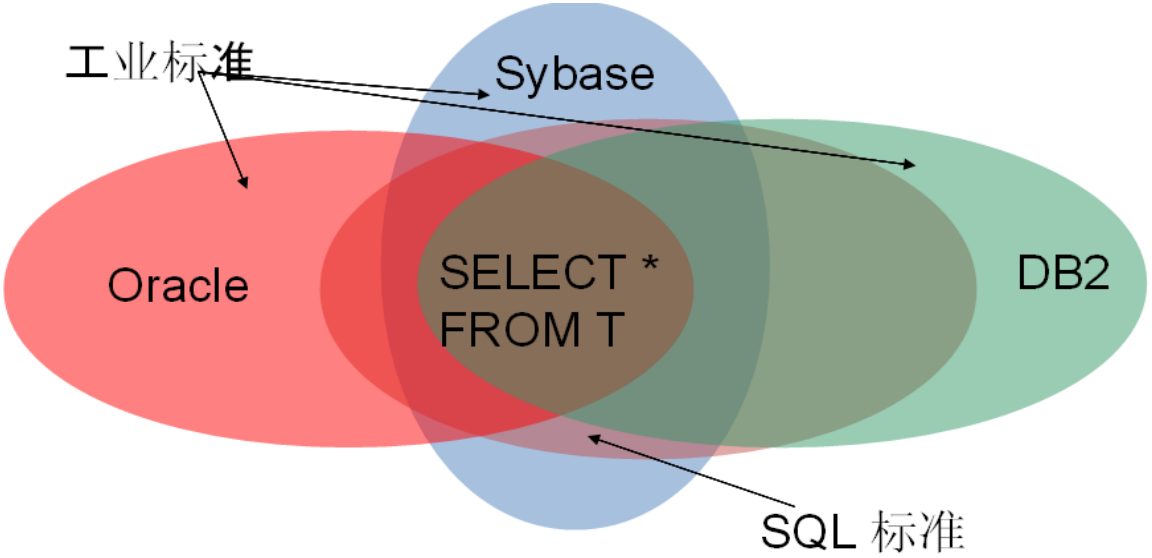
易用性
灵活开发
领先业界的 XML 支持
自我管理

最佳可靠性
世界级的审计和安全
易用的高可用性
工作负荷管理

为什么还没有移植到DB2?

- 作为 客户 我认为
  - 移植关键应用还是有很大的风险
  - 投入回报周期可能很长
- 作为 服务商 我认为
  - 移植并维护另一个数据库平台可能代价很大
  - 投入回报周期可能很长

数据库移植的难点



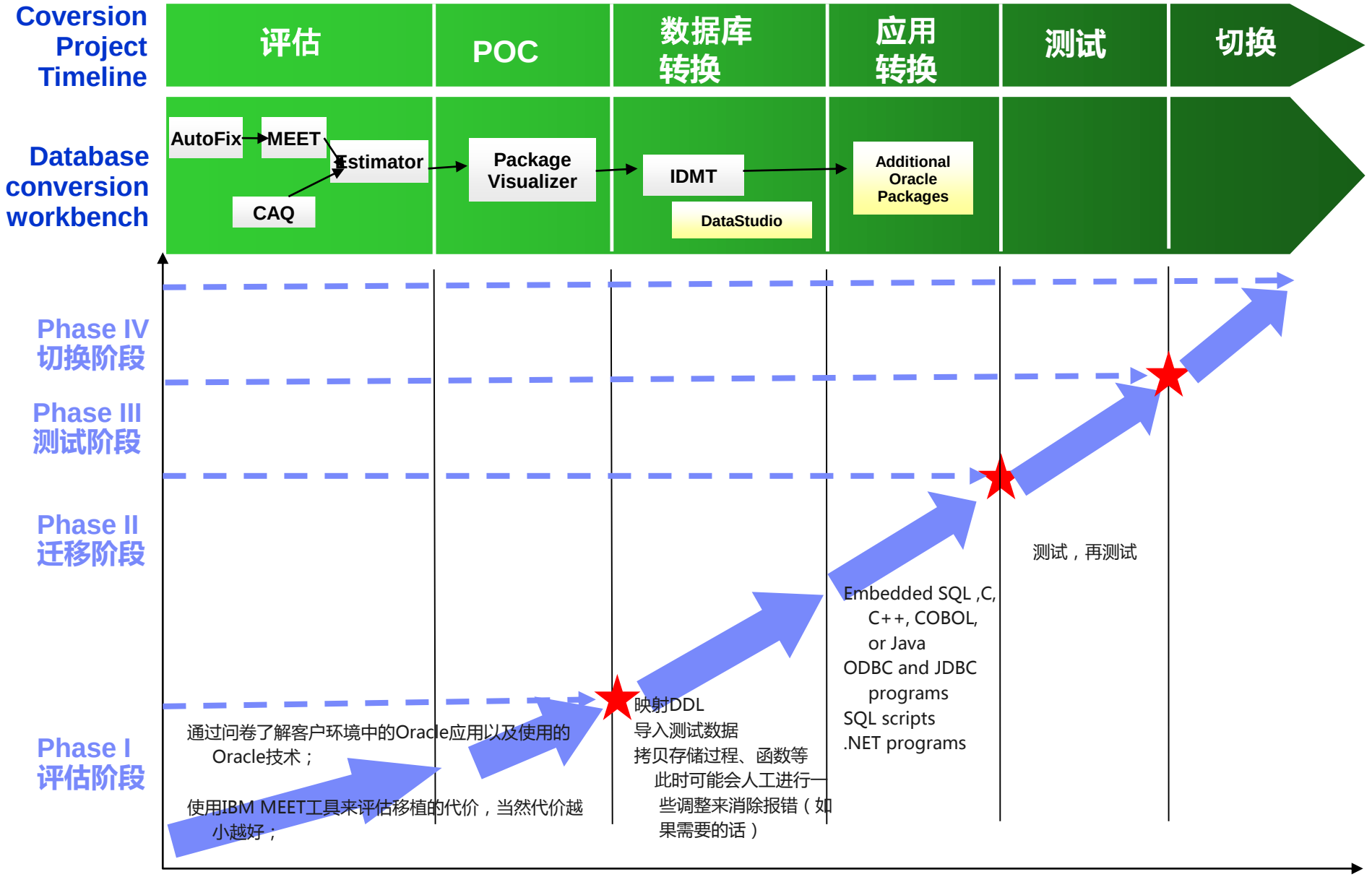
- 用户的选择：
  - 选择其中一种数据库并与之绑定
  - 支持两种或多种数据库并维护针对每种数据库的实现
  - 只使用各种数据库都支持的功能
- 上面的任何一种选择都不理想

HOW - DB2如何移植？

- 传统方法是
  - 首先实现SQL标准中的功能
  - 之后对SQL标准进行扩展加入一些业界常用功能
  - 对于待移植数据库中的私有SQL通过移植工具进行映射

- 传统方法存在的问题
  - 移植工具往往只能在DDL层次上帮助移植
    - 例如 procedures, triggers, functions, tables
  - 普通SQL通常散布在应用程序代码中，很难识别并转换
  - 转换完之后很容易引入性能问题

从Oracle向DB2转换的步骤



Migration Roadmap



tools accurate rapid

1- Assessment
- Understand the environment to be migrated
- Estimate effort


tools automated

2- Database Migration
- Move DB objects (tables, functions, SPs, etc) and data

3- Application Code Migration
- Point application to DB2 ser

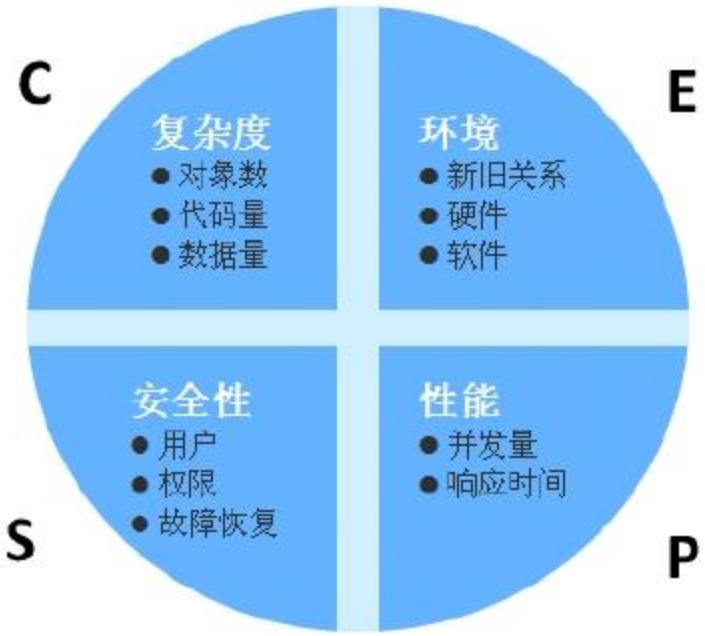
minimal changes for standard interfaces

4- Perform Functional Verification
- Execute regular set of tests for t

pplication(s)

阶段一：评估阶段

- 我方分析需要移植的系统, 得出工作量和合理的移植计划和总体移植方案
- 需要分析的内容包括
  - 需要移植的数据库的对象
  - 需要移植的代码量和复杂度
  - 需要移植的数据量
  - 目标系统和源系统的关系
  - 需要达到的性能
  - 需要达到的安全性



阶段一：评估结论

- 需要得出的结论是
  - 可以移植的内容
  - 可以完成项目的时间
  - 需要的人员
  - 需要的软硬件环境
  - 移植所采用的方法
  - 需要达到的安全性
- 输出：Sizing, 移植计划和总体方案，移植问卷

2 - Application Information (approximate counts/estimates are acceptable)

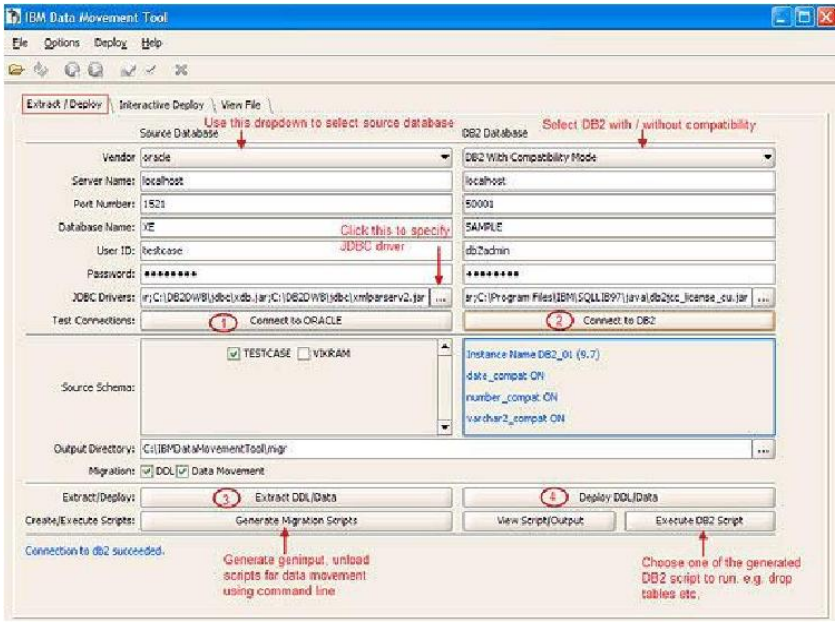
Name of application and version under consideration for enabling to DB2	
What application servers are currently supported (if any)	
What third-party integrated components are used (if any)	
What type of workload is expected	◯ OLTP ◯ OLAP/BI ◯ DSS ◯ Mixed If **OLAP/BI** or **Mixed**, is any special OLAP/BI functionality required

3 - Database Information (approximate counts/estimates are acceptable)

General	
What is the approximate database size	Average Size: GB Largest Known Size: GB
If more than one database is used by the application, please describe how	
If the database is partitioned across multiple servers, please describe architecture	
Describe any data loading requirements (data movement scripts, special loading utility, automatic data generation programs, etc.)	
Does the database store Unicode data	◯ Yes ◯ No

阶段二：迁移阶段

- 对数据库对象，应用程序进行移植
  - 数据库对象的移植
    - ✓ 表结构
    - ✓ 视图
    - ✓ 索引
    - ✓ 序列
    - ✓ 存储过程
    - ✓ 触发器
    - ✓ 用户自定义函数
    - ✓ 权限
    - ✓ 表空间
  - 数据的移植
    - ✓ 导入测试数据



阶段二：迁移阶段

- 应用程序的移植
  - ✓ Embedded SQL programs such as those written in C, C++, COBOL, or Java
  - ✓ ODBC and JDBC programs
  - ✓ SQL scripts
  - ✓ .NET programs
- 单元测试
  - ✓ 保证移植后的基本功能可以正常工作

- 输出：移植状态追踪文档, 问题追踪文档，移植后的脚本

阶段三：测试阶段

- 导入一定量的生产数据，我方协助甲方的测试人员对移植后的系统进行测试，同时对性能和出问题的地方进行调整。
- 功能测试
  - 测试移植后各个模块的功能是否正常
- 集成测试
  - 测试移植后的各个功能模块之间的交互是否正常
- 性能测试
  - 关键任务或报表的执行时间
  - 多并发下的系统状态
- 输出：测试报告，问题追踪文档

阶段四：切换阶段

- 我方协助甲方的维护人员完成两个数据库之间的切换，同时对维护人员进行新数据库的培训。
- 完全切换
  - 业务系统的原数据库完全替换
- 部分切换。
  - 两种数据库并存，需要考虑两个数据库的数据同步问题
- 输出: 培训资料

TOOLs - DB2提供哪些帮助？

- MEET DB2
  - 分析可以兼容的Oracle/Sybase的语法
- IDMT (IBM Data Movement Tool)
  - 移植数据库对象和数据的工具
- Optim Data Studio（ODS）
  - 开发、测试数据库对象、Queries和数据库逻辑运用
  - 支持Oracle PL/SQL 语法

MEET DB2-迁移工作量评估的工具

Migration Enablement Evaluation Tool for DB2

Identifies code that uses features not supported in DB2 9.7

Windows only. Requires Java.

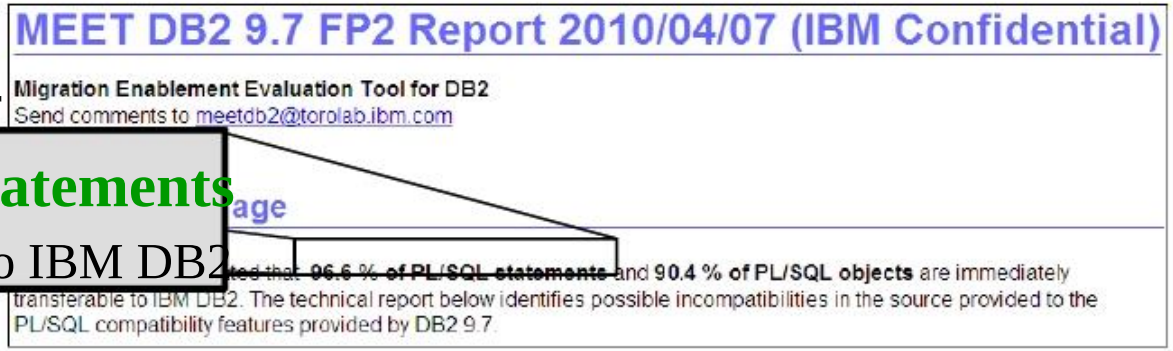
Available f or IBMers and Business Partners at :

http://tinyurl.com/meetdb2

- Clients should cont act their IBM reps.

HTML report

- Provides summary st at istics
- List s details and source code line number



96.6%of PL/ SQL statements immediately t ransf erable to IBM DB2

一个中等规模的迁移项目的评估示例

MEET DB2 9.7 FP2 Report 2010/09/22 (IBM Confidential)

Migration Enablement Evaluation Tool for DB2
Send comments to meetdb2@torolab.ibm.com

99.4% of statements
immediately transferable to IBM DB2

PL/SQL Coverage

MEET DB2 has estimated that **99.4 % of PL/SQL statements** and **92.2 % of PL/SQL objects** are immediately transferable to IBM DB2. The technical report below identifies possible incompatibilities in the source provided to the PL/SQL compatibility features provided by DB2 9.7.

Benefits

- **Rapid assessment** of application
- **Early confirmation** of compatibility
- **Lists details** and source code line number for exceptions

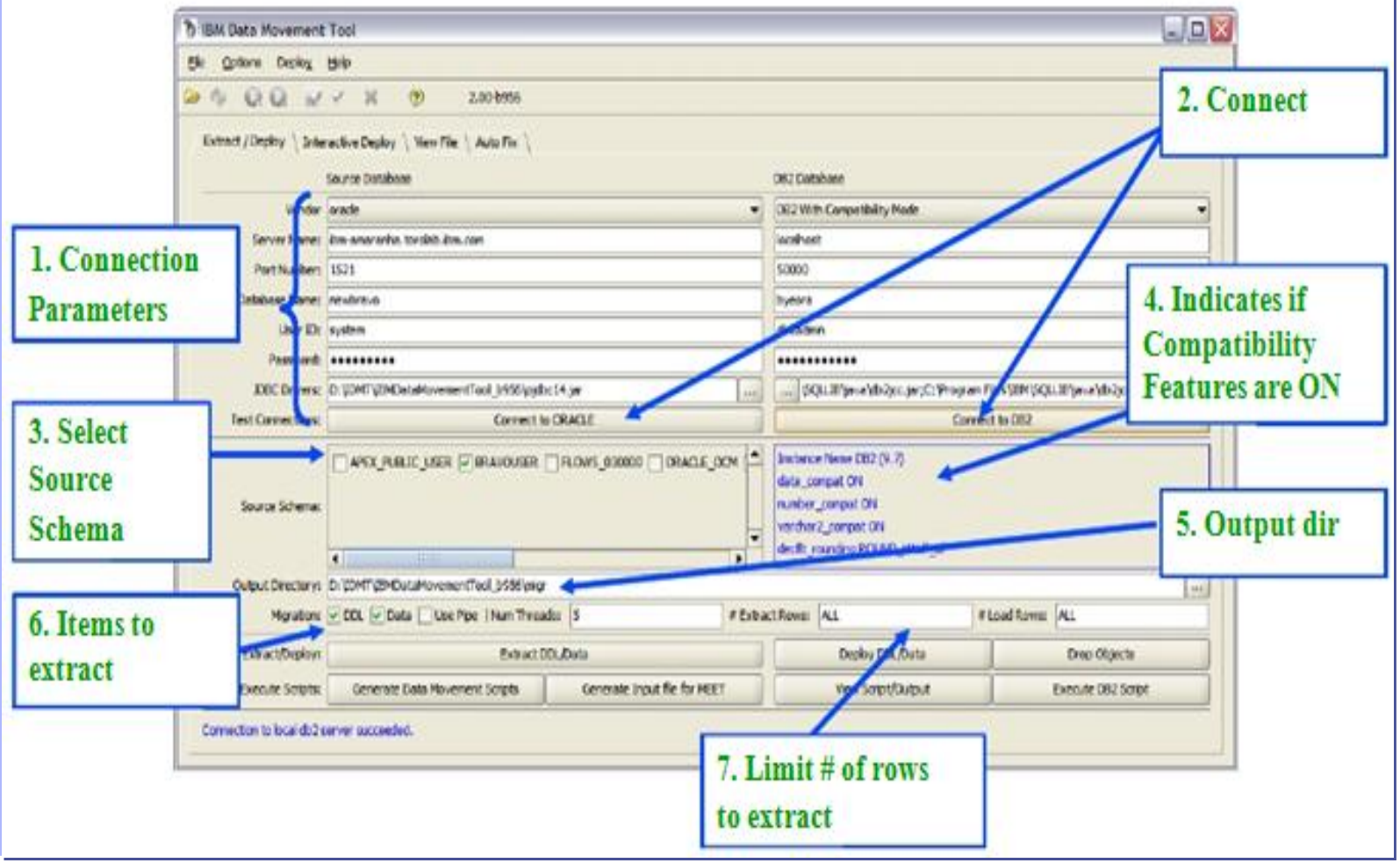
Object Type	Total Number	Number That Require Attention	Percent That Require Attention
Anonymous blocks	3	0	0 %
Procedures	4000	164	4 %
Functions	501	32	6 %
Packages	10	4	40 %
Triggers	319	176	55 %
Total Objects	4833	376	7.8 %
Statements	187451	1117	0.6 %

Feature	Description	#
NLS LIBRARY	Natural Language Library is not supported.	1
NUMBER PRECISION	The precision of the data type is out of range.	368

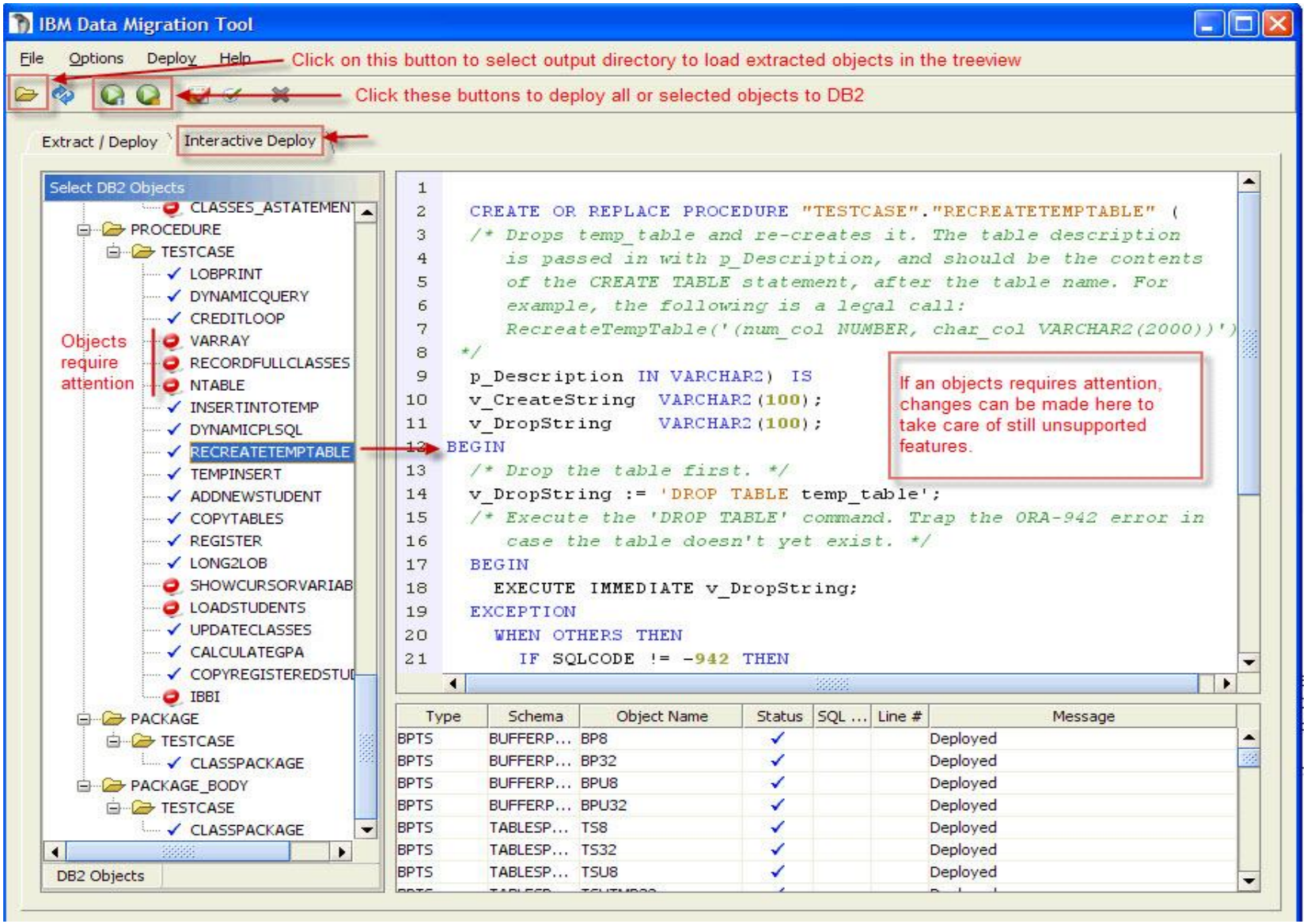
Line 247393	[package] "ADAM"."SP_DECLARA_CONCEPTO"	i_pconcepto NUMBER(38),
Line 247394	[package] "ADAM"."SP_DECLARA_CONCEPTO"	i_cond_turno NUMBER(38),

IDMT-数据库对象和数据迁移的工具

- Run IBMDataMovementTool.sh (Linux/UNIX) or IBMDataMovementTool.cmd (Windows)

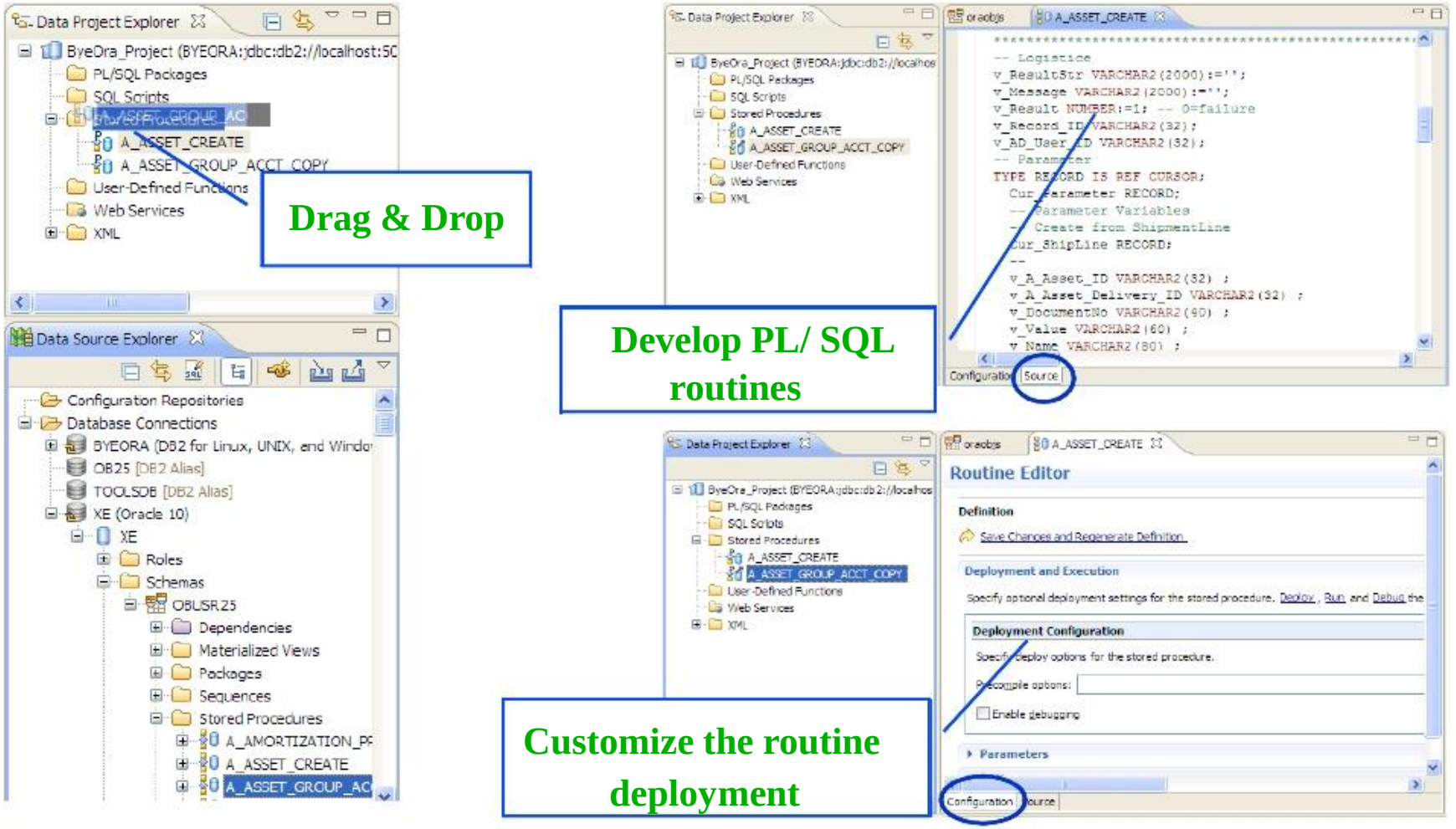


通过拖/拉/拽方便检查和修改SQL语句



ODS-开发和测试数据库对象生成、查询和应用逻辑的工具

ODS 3.1 – 支持拖拽和删除的方式编译PL/SQL 存储过程



© 20 10 IBM Co rp o ratio n

Sybase中 数据类型	DB2中 数据库类型
ID	CHARACTER(11)
CHAR(20)	CHARACTER(20)
NCHAR(20)	CHARACTER(20)
UNICAHR(20)	CHARACTER(20)
VARCHAR(20)	VARCHAR(20)
NVARCHAR(20)	VARCHAR(20)
UNIVARCHAR(20)	VARCHAR(20)
SYSNAME(30)	VARCHAR(30)
LONGSYSNAME	VARCHAR(255)
BINARY	VARCHAR(2) FOR BIT DATA
TIMESTAMP	VARCHAR(16) FOR BIT DATA
VARBINARY(20)	VARCHAR(40) FOR BIT DATA

Sybase中 数据类型	DB2中 数据库类型
BIT	SMALLINT(2)
SMALLINT	SMALLINT(2)
TINYINT	SMALLINT(2)
UNSIGNED SMALLINT	SMALLINT(2)
INT	INTEGER(4)
UNSIGNED INTEGER	INTEGER(4)
BIGINT(10)	BIGINT(8)
SMALLMONEY	DECIMAL(10,4)
DECIMAL(10,5)	DECIMAL(10,5)
MONEY	DECIMAL (19,4)
NUMERIC(10,5)	DECIMAL(10,5)
FLOAT(8)	DOUBLE(8)
REAL	DOUBLE(8)
UNSIGNED	BIGINT BIGINT(8)

Sybase中 数据类型	DB2中 数据库类型
DATE	DATE
TIME	TIMESTAMP(10,6)
DATETIME	TIMESTAMP(10)
SMALLDATETIME	TIMESTAMP(10,6)
BIGTIME	TIMESTAMP(10)
BIGDATETIME	TIMESTAMP(10)

Sybase中 数据类型	DB2中 数据库类型
TEXT	CLOB(2147483647)
UNITEXT	CLOB(1073741823)
IMAGE	BLOB(1073741824)

RISK - IBM如何控制？

- DB2兼容性
- 专业化的团队
- 迁移流程细致规范
  - 迁移前细致的评估
  - 迁移中工具自动化

风险控制一：DB2 兼容性

平均 SQL 兼容性提升到了 98% 以上
轻松摆脱更昂贵的 Oracle 数据库；在 DB2 中利用 Oracle 技能

Reliance Life Insurance

"在 IBM 系统上运行的 DB2 的总体拥有成本几乎只有 Sun 系统上的 Oracle Database 的一半。"

Banco de Crédito del Peru

*"我们从 Oracle Database 迁移到了 IBM DB2，**将我们的成本削减了一半**，同时改善了业务应用程序的性能和可靠性。"*
–Sandro Reátegui，Banco

JSC Rietumu Banka

- 从 Oracle Database 迁移到了 IBM DB2
- 使用了 "兼容性功能"
- 查询性能加快了 **3 至 30 倍**
- 数据库可用性改进了 **200%**

9.7.1	SUB STRB	提高兼容性
9.7.1	UDF 参数：INOUT	提高兼容性
9.7.1	FORALL/BULK COLLECT	提高兼容性
9.7.1	改进布尔值	提高兼容性
9.7.1	条件编译	增强
9.7.1	基本 DPF 支持	扩大覆盖范围
9.7.1	OCI 支持	扩大覆盖范围
9.7.2	UDF 参数：DEFAULT	提高兼容性
9.7.2	模糊化	增强
9.7.2	NCHAR、NVARCHAR、NCLOB	提高兼容性
9.7.3	数字性能	性能
9.7.3	运行时 "纯度级别" 执行	提高兼容性
9.7.3	RATIO_TO_REPORT 函数	提高兼容性
9.7.3	RAISE_APPLICATION_ERROR	提高兼容性
9.7.3	小 LOB 对比	提高兼容性
9.7.4	多操作触发器和更新前触发器	提高兼容性
9.7.4	自主 Tx 改进	提高兼容性
9.7.4	LIKE 改进、LISTAGG	提高兼容性
9.7.4	行和行数组 JDBC 支持	提高兼容性
9.7.5	Pro*C 支持	提高兼容性
9.7.5	嵌套复杂对象	提高兼容性
10	本地过程定义	提高兼容性
10	本地类型定义	提高兼容性
10	PL/SQL 性能	性能

Based on internal tests and reported client experience from 28 Sep 2011 to 07 Mar 2012.

DB2支持下面的Oracle功能

Oracle	→	DB2
并发控制	→	无需任何更改
Oracle SQL方言	→	无需任何更改
PL/SQL	→	无需任何更改
PL/SQL包	→	无需任何更改
系统函数库	→	无需任何更改
JDBC 扩展	→	无需任何更改
OCI	→	无需任何更改
SQL*Plus 脚本	→	无需任何更改

DB2内置Oracle的程序包

Library	说明
DBMS_ALERT	跨会话的信号灯
DBMS_DDL	Obfuscate DDL objects
DBMS_JOB	任务调度
DBMS_LOB	等同于DB2自带的LOB函数
DBMS_OUTPUT	用于打印输出信息或者调试信息
DBMS_PIPE	跨会话的数据通道
DBMS_SQL	等同于 PREPARE/EXECUTE
DBMS_UTILITY	辅助的函数和过程工具
MONREPORT	Monitoring data and generating text reports
UTL_MAIL	服务器端的邮件接口
UTL_SMTP	服务器端的SMTP接口
UTL_FILE	服务器端的IO接口
UTL_DIR	Routines for maintaining directory aliases

DB2 SQL兼容性一览

Currently Committed (log based)	Scalar functions TO_CHAR, TO_DATE, TO_NUMBER, TO_TIMESTAMP	Arithmetic functions, String Functions	CLPPLUS	MEET Assessment
Weak Typing	Dictionary Views	JDBC Extensions	OPTIM Development Studio	Data Movement
Number Varchar2 Timestamp (n) Date Boolean VARRAY Index By Table ROWTYPE Cursor Type	CONNECT BY, Outer Join(+), DUAL, ROWNUM, SELECT INTO FOR UPDATE, ROWID, AUTONOMOUS TX, TRUNCATE table, Public synonym, CGTT	PL/SQL logic, EXCEPTION, Constant variables, FOR LOOP (over range over SELECT over cursor), User Defined Exceptions, %TYPE, %ROWTYPE, PRAGMA Autonomous	Anonymous block Procedure Function Trigger PACKAGE – VARIABLE, CURSOR TYPE, EXCEPTION, FUNCTION, PROCEDURE PACKAGE SYNONYM	DBMS_OUTPUT UTL_FILE DBMS_ALERT DBMS_PIPE DBMS_JOB DBMS_LOB DBMS_SQL DBMS_UTILITY UTL_MAIL UTL_SMTP

风险控制二：专业化的团队

WHO - 大中华区信息管理技术生态系统团队（IMTE）
Information Management Technology Ecosystem

- 一只技术势力雄厚，值得信赖的全球化专业DB2迁移团队
- 编制在IBM开发试验室中
- 雄厚的技术积累
  - ✓ ORACLE
  - ✓ DB2
  - ✓ SAP Migration
  - ✓ DATA WareHouse
  - ✓ IDS embed
  - ✓ Optim

Information Management Technology Ecosystem Worldwide

350+ Experts

IM Product Areas:

- DB2 for LUW
- DB2 for z/OS
- DB2 pureScale
- SAP and DB2
- Informix
- InfoSphere Warehouse
- InfoSphere Information Server
- Infosphere DataStage
- InfoSphere QualityStage
- InfoSphere Change Data Capture
- InfoSphere MDM Server/MDM for PIM
- Optim Governance
- Guardium
- Netezza

IMTE具备专业知识的深度和广度

- 多个成功案例
- 经过验证的实施方法论
- 世界级别/范围的执行能力

风险控制三：迁移流程标准化

- 迁移前：通过Meet工具细致评估
  - 迁移工程师能提前对Oracle数据库进行自动化的统计和分析
  - 迁移工程师能根据分析报表情况在正式迁移前能有充分准备去解决可能带来风险相关问题
  - 为提供调动和计划各类资源提供依据和时间
- 迁移中：通过IDMT工具自动迁移
  - 自动化的DB2迁移工具确保数据和数据库对象在迁移过程中正确性和可靠性

成功案例一

- 国内大型ERP企业产品迁移计划

任务名称	工期	开始时间	完成时间	前置任务
1、数据库测试计划编制	1 个工作日	2012年10月16日	2012年10月16日	
2、DB2测试用数据准备	10 个工作日	2012年10月16日	2012年10月29日	1SS
3、测试方案编写	3 个工作日	2012年10月16日	2012年10月18日	1SS
4、准备测试用例	5 个工作日	2012年10月19日	2012年10月25日	3
5、DB2数据库适配	16 个工作日	2012年10月19日	2012年11月9日	1,3
6、在, 集团搭建DB2硬件环境	2 个工作日	2012年11月12日	2012年11月13日	5
7、在 集团进行测试及修改(DB2)	5 个工作日	2012年11月14日	2012年11月20日	2,4,6
8、新华服务器硬件到位并安装调试完成	25 个工作日	2012年10月17日	2012年11月20日	1,7FF
9、新华现场硬件环境准备	2 个工作日	2012年11月21日	2012年11月22日	8,7
10、客户环境验证(DB2)	5 个工作日	2012年11月23日	2012年11月29日	9

- 时间紧
- 功能多
- 产品相对封闭

- 实际迁移时间
  - 1天评估
  - 2两天迁移
  - 2周测试
  - 2天上线安装

- 大型银行核心系统的迁移状态跟踪

模块		总个数	已经移植的个数	还没有移植的个数	描述
数据库对象	Table	1654	1653	1	QTAB_LOG包含UDT，本次不需要修改
	Sequence	2746	2746		
	Default value	167	167		
	Check Constrainst	2245	2245		
	Primary Key	1160	1159	1	不需要修改,PK_QTAB_LOG(表QTAB_LOG不存在)
	Unique Index	102	102		
	Index	696	694		不需要修改,(表QTAB_LOG不存在)
	Foreign keys	289	270	19	
	Type	44	0	44	
	Function	10	5	5	总数355个 本次需要修改的10个
	View	171	163	8	
	Trigger	481	481		
	Procedure	117	71	46	
	Package	135	99	36	总数401个， 本次需要修改的136个
	package_body	136	99	36	总数399个， 本次需要修改的136个
	Directory	2			
	表空间	10			
应用程序的移植	功能点	414	250	164	
测试数据的移植					

- 产品时间跨度大
- 功能点多
- 应用数据库特性多

- 实际迁移时间
  - 5天评估
  - 3周迁移
  - 3周测试

成功案例二

- 国内大型医院行业性能测试结果

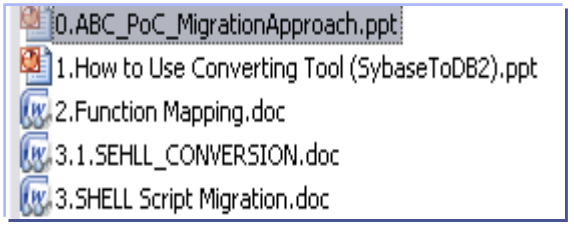
分类	项目	数据量	oracle开始	oracle结束	oracle耗时(秒)	DB2开始	DB2结束	DB2耗时(秒)	比例
IMP	IMP	146182	2012-04-28 11:41:04	2012-04-28 11:47:05	361	2012-04-28 10:57:33	2012-04-28 10:59:34	121	33.52%
DCT	抽取	146182	2012-04-28 11:52:23	2012-04-28 11:55:48	205	2012-04-28 11:02:51	2012-04-28 11:03:53	62	30.24%
	存储	146182	2012-04-28 11:55:49	2012-04-28 11:57:59	130	2012-04-28 11:03:53	2012-04-28 11:08:22	269	206.92%
	清洗	146182	2012-04-28 11:57:59	2012-04-28 12:00:00	179	2012-04-28 11:08:23	2012-04-28 11:10:46	143	79.89%
	输出	146182	2012-04-28 12:06:51	2012-04-28 12:35:56	1145	2012-04-28 11:14:43	2012-04-28 11:25:44	661	57.73%

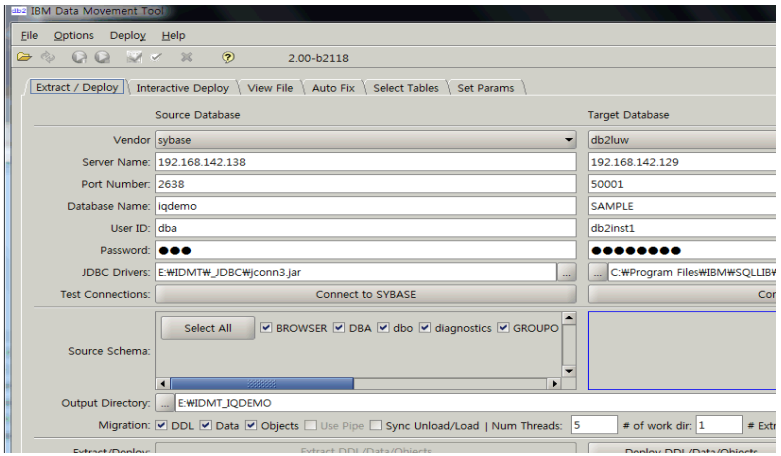
- 大型银行Sysbase迁移到DB2培训资料

Migration Approach for each phase

- **IMPLEMENTATION PHASE**

Tasks	Approach	Checkpoints
SCHEMA GENERATION	Build the schema in target database under the table migration design. Check any conflict issues against target database.	Are there any unsupported schema objects ? Are there any naming conflict issues ?
SQL CONVERSION	Once the schema generated, start to compile user stored objects (function, procedure, packages). Find alternative and comparable feature on meeting new syntax from user's SQL	Check if SQL queries run well against sample data.
DATA LOAD	Load user data into newly built database object using IDMT if usable.	Are there any data not loadable ? Check numeric data representation. Check if varchar data is null trimmed
SHELL SCRIPT CONVERSION	Convert user shell scripts written in and ISQL to DB2 CLP. Find alternative and comparable feature on meeting new syntax from user's stored procedure	Check the scripts run well against sample case. Check the shell makes necessary output (ex. standard output, file).






Thank
You