



A PRACTITIONER'S GUIDE TO  
SUCCESSFULLY MIGRATE FROM  
ORACLE TO SYBASE ASE – PART 2

PETER DOBLER  
PRESIDENT DOBLER CONSULTING  
SEPTEMBER 14<sup>TH</sup> 2011



## ABOUT PETER DOBLER

- Peter Dobler is an accomplished IT database professional, with 25+ years of experience, who delivers results by building sophisticated data systems; by implementing advanced application software; and, by improving efficiencies and reducing costs. He founded Dobler Consulting, ([www.doblerconsulting.com](http://www.doblerconsulting.com)) a Tampa, Florida consulting firm that delivers implementation expertise for Sybase, Oracle and MS SQL Server in 2000. Dobler Consulting is a Sybase System Integration partner and a premier partner for Sybase Professional Services. Peter designed and deployed large scale, high performance, enterprise level DW systems exceeding 40 TB and managed large IT groups of DBAs, System Engineers, Storage Engineering, and Security experts. He is a nationally recognized premier Sybase expert who frequently gets invited to speak at Sybase events. He can be reached at: <mailto:pdobler@doblerconsulting.com> or 813-322-3240.
- Read Peter's Database Technology Trends Blog (Behind the Scenes of Database Evolution) at: <http://www.peterdobler.com>, and his Technology Tips Blog (Step-by- step Instructions on Today's Challenging Technology) at <http://www.techtipsntrick.com>.



# ABOUT THIS PRESENTATION

- This session is part 2 of a 2 part session
  - Oracle And Sybase ASE Compared
  - Portability Check With Workarounds
  - SQL Language Differences With Sample Code
  - Application Migration Examples
  - Data Migration Example
  - Special Considerations



# OVERVIEW



# MIGRATION PATH

Full Migration	3 <sup>rd</sup> Party Software / ERP Migration
Planning the migration	Planning the migration
Designing architecture migration	
Implementing schema migration	
Implementing data migration	Implementing data migration
Implementing application migration	
Implementing operation migration	Implementing operation migration
Validating the migration	Validating the migration



# MIGRATION OBJECTIVES

- The focus must be on migrating the entire application with the least effort possible. Otherwise a migration project can easily turn into a new application development project.
- Migrating the syntax from Oracle to Sybase ASE is only one step of the entire process. The same command in Oracle can have different implications in Sybase ASE.
- An end user should never be able to tell if the application is running on Oracle or Sybase ASE.



# ORACLE AND SYBASE ASE COMPARED

7 – October 18, 2011





# ORACLE AND SYBASE ASE COMPARED

## Database And Instances

Oracle	Sybase
<p><b>Database:</b> A collection of physical operating system files or disk. When using Oracle 10g or 11g and the Automatic Storage Management (ASM), this might not appear as individual separate files in the operating system, but the definition for a database stays the same.</p> <p><b>Instance:</b> A set of Oracle background processes/threads and a shared memory area, which use memory that is shared across those processes/threads running on a single computer. A database instance can exist without any disk storage. This understanding helps to draw the line between a database and an instance in Oracle.</p>	<p>Applying a one-to-one translation from Oracle to Sybase ASE, an instance would be the dataserver process in Sybase ASE, which controls all the operations of a Sybase ASE server. A database would be the entire collection of system and user databases in Sybase ASE.</p> <p>To complete the functions of an Oracle instance, the next closest Sybase process would be the Sybase Unified Agent.</p>



# ORACLE AND SYBASE ASE COMPARED

## Processes

Oracle	Sybase
<p><b>Server processes:</b> These perform work based on a client's request. The dedicated and shared servers are part of this process class. Each client access spawns a new process.</p> <p><b>Background processes:</b> These are the processes that start up with the database and perform various maintenance tasks, such as writing blocks to the disk, maintaining the online redo log, cleaning up aborted processes, and so on. Over 30 processes at a minimum.</p> <p><b>Slave processes:</b> These are similar to background processes, but they are processes that perform extra work on behalf of either the background or a server process.</p>	<ul style="list-style-type: none"><li>• <i>Sybase Dataserver</i></li><li>• <i>Sybase Unified Agent</i></li><li>• <i>Sybase Backup Server</i></li><li>• <i>Sybase XP Server</i></li><li>• <i>Sybase Job Scheduler</i></li><li>• <i>Sybase Monitor Server</i></li><li>• <i>Sybase Historical Server</i></li><li>• <i>Sybase Web Services</i></li></ul>



# ORACLE AND SYBASE ASE COMPARED

## Case Sensitivity

Oracle	Sybase
<p><b>Oracle is not case sensitive.</b></p> <p>Select * from TEST =&gt; is the same as Select * from test</p> <ul style="list-style-type: none"><li>• This can become a migration problem if developers didn't adhere to a strict naming convention.</li></ul>	<p><b>Sybase ASE is case sensitive by default.</b></p> <p>If the same database object is referenced in different ways. (e.g. , TEST vs. test, then one option is to change the default sort order to change ASE to case insensitive. This has other consequences. The better option is to find the case mismatch in the source code.</p>



# ORACLE AND SYBASE ASE COMPARED

## Storage Architecture Mapping

Oracle	Sybase
Data Files	Database Devices
Data Block	Page
Extent	Extent
Segments	N/A
Tablespaces Tablespace Extent Tablespace Segments	Databases Segments
Redo Log Files	Log Devices
Control Files	Master Database



# ORACLE AND SYBASE ASE COMPARED

## Logical Transaction Handling

Oracle	Sybase
<p>Statements are not automatically committed to the database. Oracle transactions are implicit. This means that the logical transaction starts as soon as data changes in the database.</p> <ul style="list-style-type: none"><li>• <b>COMMIT [WORK]</b> commits the pending changes to the database.</li><li>• <b>ROLLBACK</b> undoes all the transactions after the last <b>COMMIT [WORK]</b> statement. Savepoints can be set in transactions with the following command:<ul style="list-style-type: none"><li>• <b>SET SAVEPOINT</b> savepoint_name</li></ul></li><li>• The following command rolls back to the specified savepoint:<ul style="list-style-type: none"><li>• <b>ROLLBACK</b> &lt;savepoint_name&gt;</li></ul></li></ul>	<p>By default, statements are automatically committed to the database. As a result, any statement not within a transaction is automatically committed. Sybase ASE allows you to nest BEGIN TRAN/COMMIT TRAN statements. When nested, the outermost pair of transactions creates and commits the transaction. The inner pairs keep track of the nesting levels.</p> <ul style="list-style-type: none"><li>• <b>begin transaction</b> – marks the beginning of the transaction block.</li><li>• <b>save transaction</b> – marks a savepoint within a transaction</li><li>• <b>commit</b> – commits the entire transaction</li><li>• <b>rollback</b> – rolls a transaction back to a savepoint or to the beginning of a transaction</li></ul>



# ORACLE AND SYBASE ASE COMPARED

## Exception-Handling and Error-Handling Semantics

Oracle	Sybase
Oracle places an implicit SAVEPOINT at the beginning of a procedure. The built-in RAISE_APPLICATION_ERROR procedure rolls back to this SAVEPOINT, or the last committed transaction within the procedure. The control is returned to the calling routine.	Sybase ASE passes the control from one SQL statement to another without checking for errors. This means that you have to check for errors after every SQL statement within a procedure. Sybase ASE's equivalent to Oracle's RAISE_APPLICATION_ERROR is called RAISERROR. Unlike Oracle, RAISERROR does not return the controls to the calling routine.



# ORACLE AND SYBASE ASE COMPARED

## Parallel Execution

Oracle	Sybase
<p><b>Parallel Query:</b> By default the Oracle Database is enabled for parallel execution and when Automatic Degree of Parallelism (Auto DOP) is active.</p> <ul style="list-style-type: none"><li>• <code>alter table customers parallel;</code></li><li>• <code>SELECT /*+ parallel(c) parallel(s) */ c.state_province, sum(s.amount) revenue FROM customers c, sales s WHERE s.cust_id = c.cust_id AND s.purchase_date BETWEEN TO_DATE('01- JAN-2010','DD-MON-YYYY') AND TO_DATE('31-DEC-2010','DD-MON-YYYY');</code></li><li>• <code>alter session force parallel query ;</code></li></ul>	<p><b>Parallel Query:</b> Sybase ASE supports vertical, horizontal and pipelined parallel query execution. To enable parallelism on Sybase ASE the following configuration parameters have to be configured:</p> <ul style="list-style-type: none"><li>• <b>number of worker processes</b></li><li>• <b>max parallel degree</b></li><li>• <b>max resource granularity</b></li><li>• <b>max repartition degree</b></li><li>• <b>max scan parallel degree</b></li><li>• <b>prod-consumer overlap factor</b></li><li>• <b>min pages for parallel scan</b></li><li>• <b>max query parallel degree</b></li></ul>



# ORACLE AND SYBASE ASE COMPARED

## Parallel Execution

Oracle	Sybase
<b>Parallel DML:</b> Data Manipulation Language (DML) operations such as INSERT, UPDATE, and DELETE can be parallelized by Oracle.	<b>Parallel DML:</b> Sybase ASE does not support parallel execution on INSERT, UPDATE and DELETE statements.
<b>Parallel DDL:</b> Parallel DDL works for both tables and indexes, whether partitioned or non-partitioned.	<b>Parallel DDL:</b> Sybase ASE supports parallel DDL on index creation only.
<b>Parallel Data Loading:</b> Parallel loading with SQL*Loader with multiple session into the same table.	<b>Parallel Data Loading:</b> Loading data into a partitioned table using parallel bcp lets you direct the data to a particular partition in the table.
<b>Parallel Recovery:</b> Parallel recovery can speed up both instance recovery and media recovery.	<b>Parallel Recovery:</b> Sybase ASE supports parallel recovery during start-up and failover, allowing databases to be recovered in parallel by multiple recovery tasks.



# PORTABILITY CHECK WITH WORKAROUNDS

16 – October 18, 2011





## PORTABILITY CHECK WITH WORKAROUNDS

- The portability check is designed to identify migration incompatibilities and to develop workarounds.
- The main goal of an effective Oracle to Sybase ASE migration is to keep code re-writes to a minimum.
- It is important to create a complete inventory of **all** database and application components.
- Scanning of the stored procedures, user defined functions and triggers happens at the database data dictionary level.  
Application source code can be scanned in source control software.



# PORTABILITY CHECK WITH WORKAROUNDS

## Triggers

- Oracle AFTER STATEMENT triggers are compatible with Sybase ASE triggers. For the other triggers, apply these guidelines:

Oracle	Sybase
BEFORE STATEMENT Trigger	Create a rule or incorporate program logic into trigger. Additional transaction handling is necessary.
BEFORE ROW Trigger AFTER ROW Trigger	Migrate logic to use the pseudo tables <code>deleted</code> and <code>inserted</code> .



# PORTABILITY CHECK WITH WORKAROUNDS

## Synonyms

- Oracle synonyms are alternative names that provide data independence and location transparency.

Oracle	Sybase
Synonyms on table, view, sequence, operator, procedure, function, package, materialized view, Java class objects, user-defined object types or another synonym.	Table and view synonyms can be implemented via a view in a separate database to cross reference the physical tables and views in the application database. All other references to synonyms must be replaced with the fully qualified name to an object. <code>&lt;remote_server&gt;.&lt;database_name&gt;.&lt;owner&gt;.&lt;object_name&gt;</code>



# PORTABILITY CHECK WITH WORKAROUNDS

## Sequences

Oracle	Sybase
<p>A sequence creates a unique key to be used as a unique identifier in table's primary key.</p> <pre>CREATE SEQUENCE test_seq MINVALUE 1 STARTWITH 1 INCREMENTED by 1 CACHE 20;  INSERT INTO m_table VALUES (test.seq.nextval, ..);</pre>	<p>Can be implemented via an identity column or with this code:</p> <pre>CREATE TABLE my_seq (seq int) go //initialize the sequence INSERT INTO my_seq select 0 go CREATE PROCEDURE get_seq (@seq int OUTPUT) AS UPDATE my_seq SET seq = seq+1 SELECT @seq = seq FROM my_seq go // execute the sp to get the next sequence number DECLARE @seq int EXEC get_seq @seq OUTPUT INSERT INTO m_table VALUES (@seq, ..) go</pre>



# PORTABILITY CHECK WITH WORKAROUNDS

## Materialized Views

- Sybase ASE does not support the concept of materialized views.
- Oracle uses materialized views for one way replication via dblink connections and updateable materialized views for two-way replication.
- Sybase Replication Server will handle these replication needs outside Sybase ASE.
- **Info:** Sybase ASE supports materialized computed columns in tables, which are not the same as materialized views.



# PORTABILITY CHECK WITH WORKAROUNDS

## Table Types

Oracle	Sybase
<b>Heap organized tables:</b> This is the Oracle default, row locking table construct.	Create a table with the lock datarows clause.
<b>Index organized tables:</b> These tables are stored in an index structure, according to the primary key.	Create a table with the lock allpages clause and create a clustered index on the primary key.
<b>Index clustered tables:</b> These are clusters of one or more tables stored on the same database block. The data is “clustered” around the cluster key value.	Create a table with the lock allpages clause and use the partition clause to partition the table across segments. Create a clustered index on the primary key.



# PORTABILITY CHECK WITH WORKAROUNDS

## Table Types

Oracle	Sybase
<b>Hash clustered tables:</b> Similar to clustered tables but instead of using a B*Tree index for the key the hash cluster hashes the key to the cluster. The data is the index.	Create a table with the lock allpages clause and create a clustered index on the primary key.
<b>Sort hash clustered tables:</b> Similar to the hash clustered table, but used on timestamp-based records using the first in first out (FIFO) approach.	Create a table with the lock allpages clause and create a clustered index on a timestamp. No FIFO support.
<b>Nested tables:</b> These are system-generated and –maintained child tables in a parent/child relationship. These are not “stand-alone” tables.	N/A



# PORTABILITY CHECK WITH WORKAROUNDS

## Table Types

Oracle	Sybase
<b>Temporary tables:</b> These are tables that store data for the life a transaction or the life of a session. Data is only visible to the session that created the temporary table:	Temporary tables in the tempdb that are identified by starting with a # sign. These tables are created by the session with either create table #temp_table or select ... into #temp_table command.
<b>Object tables:</b> These tables contain special attributes not available with non-object tables such as system generated REF (object identifier) for each row. This is a special case of heap, index organized and temporary tables.	N/A.  With the Java extension it is possible to store object classes as object columns in a table.
<b>External tables:</b> These tables are not stored in the database itself. These are files residing outside the database, but are made available to the database query engine as if they are tables. Very powerful in loading data from external sources.	Sybase ASE provides proxy tables with the Component Integration Services (CIS). This allows access to external data in files on filesystems or tables in other, non-Sybase ASE database systems.



# PORTABILITY CHECK WITH WORKAROUNDS

## Index Types

Oracle	Sybase
<b>B*Tree Index:</b> This is the most commonly used index in Oracle. It is similar to a binary tree index, but Oracle refers to it as a balanced tree index.	<b>Non-Clustered Index:</b> This is the most commonly used index in Sybase ASE. This is a B*Tree index like Oracle. It includes root level, leaf level and intermediate level entries for fast access to the indexed data.
N/A	<b>Clustered Index:</b> Clustered indexes on all pages locked tables maintain the index key data in key order. The index is the data. You can create only one clustered index per table.
<b>Bitmap Index:</b> With B*Tree indexes there is a one-to-one relationship between a data row and an index entry pointing to this data row. A bitmap index has a single index entry pointing to many data rows simultaneously.	N/A



# PORTABILITY CHECK WITH WORKAROUNDS

## Index Types

Oracle	Sybase
<b>Bitmap Join Index:</b> This index allows joining other tables to an indexed column in the main table. This will eliminate the need to join tables at the query level.	N/A
<b>Function-based Index:</b> This is an index based on the result of a function. You can consider them an index on a virtual (or derived) column.	<b>Function-based Index:</b> This index can be used to provide an inexpensive option for enhancing the performance of certain legacy applications. Function-based indexes allow you to create indexes based directly on one or more Transact-SQL expressions.
<b>Application domain index:</b> These are indexes you build and store yourself, either in Oracle or perhaps even outside Oracle. The Oracle text index is an example of an application domain index. It uses a set of tables to implement its concept of an index. Typically used for a specialized domain, such as spatial or image processing.	N/A



# PORTABILITY CHECK WITH WORKAROUNDS

## Data Types

- There are a couple of data types in Oracle that do not translate directly into a Sybase ASE data type, or additional measures have to be implemented.

Oracle	Sybase
<ul style="list-style-type: none"><li>• BLOB</li><li>• CLOB</li></ul>	The BLOB data type can be migrated to an IMAGE data type and the CLOB to a TEXT data type. Check the max length of the BLOB and CLOB to ensure that they don't exceed 4GB.
<ul style="list-style-type: none"><li>• VARCHAR2</li></ul>	The max length of VARCHAR2 determines the database page size. If there is data that fills the 4000 characters available, the database page size has to be at least 8k.



# PORTABILITY CHECK WITH WORKAROUNDS

## HINTS

- Oracle uses HINTS to provide some additional guidance to the query optimizer. Sybase does not support this syntax.

Oracle	Sybase
<pre>select /*+ index (a) */ object_name from my_objects a where object_type = 'INDEXTYPE';</pre>	<p>The best way to find HINTS in SQL code is to search for /*+ , and remove them to make to code compatible.</p>



# PORTABILITY CHECK WITH WORKAROUNDS

## ROWID

- Every Oracle table contains a unique identifier that is always there, even if you don't have a primary key for the table defined.

Oracle	Sybase
<pre>select ROWID from emp;  ROWID ----- AAAQ+LAAEAAAAfAAA AAAQ+LAAEAAAAfAAB AAAQ+LAAEAAAAfAAC AAAQ+LAAEAAAAfAAD .....</pre>	<p>To provide the ROWID functionality to the application, create an identity column named ROWID for each table that uses ROWID in the Oracle SQL code.</p> <p>The value in ROWID is secondary, the uniqueness is what ROWID is used for.</p>



# PORTABILITY CHECK WITH WORKAROUNDS

## ROWNUM

Oracle	Sybase
<pre>SELECT * FROM emp WHERE state = 'CA' AND ROWNUM &lt; 21</pre>	<p>Sybase ASE does not implement pseudo columns with its table store. The Oracle pseudo column ROWNUM does not exist, but comparable functionality can be achieved, using rowcount.</p> <pre>Set rowcount 20 SELECT * FROM emp WHERE state = 'CA' Set rowcount</pre>



# PORTABILITY CHECK WITH WORKAROUNDS

## Flashback

- Sybase ASE does not provide the Flashback functionality or an equivalent.

Oracle	Sybase
<p>Oracle provides SQL syntax extension that allows you to query the Flashback area from within applications.</p> <p>Any user can access its own data from the Flashback area. E.g. This can be useful for content management applications as a way of managing a recycle bin with minimal code, using the Flashback area.</p>	<p>If the application requires access to a Flashback area as part of its functionality and design, the following re-write approach can be successful:</p> <ol style="list-style-type: none"><li>1. Create an archive database</li><li>2. Create triggers that insert all DML changes into the archive database</li><li>3. Re-write the SQL code and replace Flashback references with the archive database references.</li></ol>



# PORTABILITY CHECK WITH WORKAROUNDS

## Flashback

- Install and enable the auditing feature in Sybase ASE. This will allow for timestamp searches on actions that need to be reversed or data that needs to be extracted before a certain action.
- With the exact timestamp, you will be able to create an archive database from backup logs and transaction logs.
  1. Create an archive database based on the last full backup taken from this database.
  2. Apply all transaction logs to roll forward all transactions to the desired timestamp.
  3. Use the until time parameter in the load transaction command on the last transaction log file to pinpoint the restore timestamp just before the desired timestamp. This will give you the exact data constellation before a destructive or faulty action has taken place.



# PORTABILITY CHECK WITH WORKAROUNDS

## Resource Manager

- Sybase ASE provides an equivalent to Oracle's Resource Manager in the form of the Resource Governor to set limits on how resources are being used.

Oracle	Sybase
<p>The Oracle Resource Manager triggers actions based on CPU consumption and I/O.</p> <p>Oracle maintains rules based action diagrams that will reduce the available resources for the offender and moves sessions into a lower resource pool.</p>	<p>Sybase ASE can match this and adds numbers of rows selected and the amount of space used in the tempdb as triggers for actions.</p> <p>Sybase ASE does not throttle sessions that reaches the resource limits. Instead it can raise a warning, aborts the query / transaction, or ends the user session altogether.</p> <p>There is no workaround for this action.</p>



# PORTABILITY CHECK WITH WORKAROUNDS

## Oracle SQL Plan Management

Oracle	Sybase
<p>Oracle has the ability to store and maintain query plans to support the query optimizer to make better decisions.</p> <p>Every time a query gets executed, the query optimizer compares the current query plan with the stored query plan and chooses the better plan automatically.</p>	<p>Sybase ASE can create abstract query plans that can be manually attached to a query. This is mostly used to reproduce production query behavior in a test environment where the data volume would not allow the query optimizer to make the same exact decisions.</p> <p>Sybase ASE does not provide a similar functionality to Oracle SQL Plan Management.</p>



# PORTABILITY CHECK WITH WORKAROUNDS

## Automatic Workload Repository (AWR)

Oracle	Sybase
<p>When activated, Oracle stores every query with corresponding performance indicators and metrics in a repository called AWR (Automatic Workload Repository). By default, 7 days worth of queries will be captured and stored.</p> <p>Queries can be analyzed. It allows selecting the top poorly performing queries automatically to be tuned and new query plans managed by the SQL Plan Management will be used when the query are being executed again.</p>	<p>With Sybase ASE, a similar repository can be created using MDA tables and a custom developed data storage function that exports the MDA table information on a regular interval, 1 minute or less, and stores the information in a repository for future reference and analysis.</p> <p>These analytical reports on MDA table performance tracking data have to be custom created as well.</p> <p>Sybase Control Center (SCC) is starting to offer similar functionality. But Sybase does not offer automatic query plan tuning.</p>



# PORTABILITY CHECK WITH WORKAROUNDS

## Pseudo Columns

- Oracle supports and maintains a set of pseudo columns that are part of every table definition, but do not need to be configured.
- These pseudo columns have to be eliminated from the SQL code and alternative usage has to be developed, like creating identity columns in place of ROWID.
  - ORA\_ROWSCN
  - VERSION\_XID
  - VERSION\_STARTSCN
  - VERSION\_ENDSCN
  - VERSION\_STARTTIME,
  - VERSION\_ENDTIME,
  - VERSION\_OPERATION
  - ROWID



# PORTABILITY CHECK WITH WORKAROUNDS

## SQL Syntax

- The following SELECT syntax attributes are not supported by Sybase ASE. This may trigger a complete rewrite of the application logic.

- ONLY
- INTERSECT
- MINUS
- FOR
- UNPIVOT
- INCLUDE
- EXCLUDE
- NULLS
- SAMPLE
- SEED
- CROSS
- NATURAL
- CONNECT BY
- NOCYCLE
- START WITH
- ROLLUP
- CUBE
- GROUPING SETS
- MODEL
- IGNORE
- KEEP
- NAV
- UNIQUE
- DIMENSION
- SINGLE REFERENCE
- RETURN ALL ROWS
- RETURN UPDATED ROWS
- REFERENCE
- ON
- MAIN
- PARTITION BY
- DIMENSION BY
- MEASURES
- RULES
- ITERATE
- SIBLINGS
- NULLS FIRST
- NULLS LAST
- GROUPING SETS
- NOWAIT
- WAIT
- SKIP LOCKED
- AS OF TIMESTAMP
- SYSTIMESTAMP
- AS OF



# PORTABILITY CHECK WITH WORKAROUNDS

## Oracle Web Access from PL/SQL

- The function calls from PL/SQL against the Oracle Web Services are incompatible with Sybase ASE.
- Check the PL/SQL code for the following keywords:
  - OWA\_CUSTOM
  - OWA\_CX
  - OWA\_OPT\_LOCK
  - OWA\_SEC
  - OWA\_TEXT
  - OWA\_UTIL



# PORTABILITY CHECK WITH WORKAROUNDS

## Search for Keywords in Oracle Source

Before you can migrate an Oracle schema, Oracle stored procedure, functions or triggers, you need to check for keywords that we already identified as either problematic or non-migrateable.

The following code will scan any object within the Oracle database for certain keywords and return the name and owner of the object as well as the object type:

```
SELECT owner, name, type FROM dba_source  
WHERE instr(UPPER(text), UPPER('<keyword>')) > 0
```



# PORTABILITY CHECK WITH WORKAROUNDS

## Search for Keywords in Oracle Source

Sometimes, it is important to retrieve the exact code and line number of the occurrence within a stored procedure, function or trigger. Please be cautious while using this option as it can return a lot of data:

```
SELECT owner, name, type, line, text FROM dba_source  
WHERE instr(UPPER(text), UPPER('<keyword>')) > 0
```



# PORTABILITY CHECK WITH WORKAROUNDS

Search for Sybase Reserved Keywords in Oracle Source

Not only is it necessary to search for incompatibilities. You also need to make sure that the Oracle code does not contain any Sybase ASE reserved keywords.

Check the Sybase ASE manuals for reserved keywords and use the code listed on the previous slide to search those keywords.



# SQL LANGUAGE DIFFERENCES WITH SAMPLE CODE

# SQL LANGUAGE DIFFERENCES WITH SAMPLE CODE

## SELECT Statements

- There may be significant migration issues with the SQL Language since Oracle has implemented extensions to the SQL-92 SQL Languages standard and beyond.

Oracle	Sybase
<pre>SELECT   DECODE(T1.C1, 'CLNN',          T2.C2,          T3.C3) as P_ID,   T4.ID as ID FROM T4, T1,       T2, T3 WHERE T4.ID = T1.ID(+) and T1.ID(+) = 'L3' and T4.ID = T2.ID(+) and T4.ID = T3.ID(+) and T3.TYPE='C' and T3.STAT = '00'</pre>	<pre>SELECT   case T1.C1     when 'CLNN' then T2.C2     else T3.C3   end as P_ID,   T4.ID as ID into tempdb..query13 FROM T1   left outer join T2 on T1.ID = T2.ID and T1.ID = 'L3'   left outer join T3 on T1.ID = T3.ID   left outer join T4 on T1.ID = T4.ID where T3.TYPE='C' and T3.STAT = '00'</pre>



# SQL LANGUAGE DIFFERENCES WITH SAMPLE CODE

## SELECT Statements

- In an Oracle database, retrieving system information requires a SELECT statement that executes against a table with one row.

Oracle	Sybase
<pre>SELECT sysdate FROM dual;  scott@oral&gt; desc dual Name                               Null?    Type ----- DUMMY VARCHAR2(1)  scott@oral&gt; select * from dual;  D - X</pre>	<pre>SELECT getdate()  Create a table named dual with one column and insert one row into this table.  Then you only need to search and replace sysdate with getdate().  SELECT getdate() FROM dual</pre>



# SQL LANGUAGE DIFFERENCES WITH SAMPLE CODE

## INSERT and DELETE Statements

- The second FROM clause or sub-query is a Sybase ASE extension that allows the user to make deletions based on the data in other tables. Oracle does not have this option.

Oracle	Sybase
<pre>INSERT INTO state (state_abbrev) VALUES ('WA');  COMMIT;  SELECT * FROM state; /</pre>	<pre>INSERT INTO state (state_abbrev) VALUES ('WA') GO  SELECT * FROM state GO</pre>
<pre>DELETE FROM SALES_HIST WHERE SALES_DATE &lt;= '1999/12/31' /</pre>	<pre>DELETE FROM SALES_HIST WHERE SALES_DATE &lt;= '1999/12/31' GO</pre>



# SQL LANGUAGE DIFFERENCES WITH SAMPLE CODE

## UPDATE Statements

Oracle	Sybase
<pre>DECLARE   v_salary emp.salary%TYPE;   CURSOR c_emp IS     SELECT *     FROM emp     WHERE deptid IN ('01')     FOR UPDATE OF salary; BEGIN   FOR v_empInfo IN c_emp LOOP   SELECT sal_increase     INTO v_salary     FROM sal_increase_by_rating     WHERE rating = ABS(v_empinfo.rating)   UPDATE emp     SET salary = salary + v_salary     WHERE CURRENT OF c_emp;   END LOOP;   COMMIT; END;</pre>	<pre>UPDATE emp SET salary = salary + b.sal_increase FROM sal_increase_by_rating b WHERE deptid IN ('01') AND ABS(rating) = b.rating</pre>



# APPLICATION MIGRATION EXAMPLES



# APPLICATION MIGRATION EXAMPLES

## PL/SQL vs. T-SQL

Some migration tasks can be done by a simple search and replace approach, but the differences between PL/SQL and T-SQL run much deeper. The transaction handling, for example, or the error handling.

Reverse engineering stored procedures, user-defined functions and triggers is a good starting point. **PowerDesigner** is a good tool to offer these functionalities. For larger applications, it is advisable to use the help of a tool for the migration. **SwisSQL** is a good tool to migrate PL/SQL to T-SQL.



# APPLICATION MIGRATION EXAMPLES

## Transaction Handling: PL/SQL vs. T-SQL

Oracle	Sybase
<p>Oracle applies ANSI-standard implicit transaction methods.</p> <pre>UPDATE test_table SET coll = 'value1'; SAVEPOINT first_sp; UPDATE emp SET coll = 'value2'; ROLLBACK TO SAVEPOINT first_sp; COMMIT; /* coll is 'value1' */</pre>	<p>Sybase ASE offers two different transaction models; the ANSI-standard implicit transaction model and the T-SQL explicit transaction model.</p> <pre>insert into publishers       values ("9906", null, null, null) begin transaction delete from publishers where pub_id = "9906" rollback transaction</pre> <p>T-SQL mode will only rollback the delete statement. Using the <code>set chained on</code> directive will enforce the ANSI-standard and the entire example transaction will be rolled back.</p>



# APPLICATION MIGRATION EXAMPLES

## Transaction Handling: PL/SQL vs. T-SQL

Oracle	Sybase
	BEGIN TRAN
SAVEPOINT tran_1	BEGIN TRAN tran_1
COMMIT	COMMIT TRAN
COMMIT	COMMIT TRAN tran_1
ROLLBACK	ROLLBACK TRAN
ROLLBACK TO SAVEPOINT tran_1	ROLLBACK TRAN tran_1



# APPLICATION MIGRATION EXAMPLES

## Error Handling: PL/SQL vs. T-SQL

Oracle	Sybase
<b>SQLCODE</b>  The server error code indicating the execution status of the most recently executed PL/SQL statement.	<b>@@error</b>  The server error code indicating the execution status of the most recently executed T-SQL statement.
<b>SQL%ROWCOUNT</b>  The number of rows affected by the most recently executed PL/SQL statement.	<b>@@rowcount</b>  The number of rows affected by the most recently executed T-SQL statement.

# APPLICATION MIGRATION EXAMPLES

## Error Handling: PL/SQL vs. T-SQL

Oracle	Sybase
<pre>CREATE OR REPLACE PROCEDURE procl AS   v_sys_error NUMBER := 0;   v_x NUMBER(10,0); BEGIN   BEGIN     SELECT COUNT(*)     INTO v_x     FROM emp ;   EXCEPTION     WHEN OTHERS THEN       v_sys_error := SQLCODE;   END;   IF SQL%ROWCOUNT = 0 THEN     DBMS_OUTPUT.PUT_LINE('No rows found.');</pre>	<pre>DROP PROC procl GO  CREATE PROC procl AS BEGIN   DECLARE @x int   SELECT @x=count(*) FROM emp   IF @@rowcount = 0     print 'No rows found.'   IF @@error = 0     print 'No errors.' END GO</pre>



# APPLICATION MIGRATION EXAMPLES

## Data Types: PL/SQL vs. T-SQL

Oracle	Sybase
<b>RECORD</b> You can declare a variable to be of type RECORD. Records have uniquely named fields. Logically related data that is dissimilar in type can be held together in a record as a logical unit.	These special PL/SQL variables cannot be translated into T-SQL
<b>TABLE</b> PL/SQL tables can have one column and a primary key, neither of which can be named. The column can belong to any scalar data type. The primary key must belong to type BINARY_INTEGER.	Creating a temporary table to replace the TABLE PL/SQL data type is a good approach; special considerations have to be made when the TABLE data type is being shared amongst several PL/SQL stored procedures in a stored procedure package.

# APPLICATION MIGRATION EXAMPLES

## OCI Calls vs. CT-Lib

Oracle	Sybase
<p>Connecting to the database:</p> <pre>OCILogon( env_hp, // environment handle err_hp, // error handle &amp;svc_hp, // service context user_name, // username strlen (user_name), // length of username password, // password strlen (password), // length of password db_name // database name strlen (db_name)); // length of database name</pre>	<pre>/* First, allocate a connection structure. */ ret = ct_con_alloc(context, &amp;connection);  /* These two calls set the user credentials (username and ** password) for opening the connection. */ ret = ct_con_props(connection, CS_SET, CS_USERNAME, Ex_username, CS_NULLTERM, NULL); ret = ct_con_props(connection, CS_SET, CS_PASSWORD, Ex_password, CS_NULLTERM, NULL);  /* Create the connection. */ ret = ct_connect(connection, (CS_CHAR *) EX_SERVER, strlen(EX_SERVER));</pre>



# APPLICATION MIGRATION EXAMPLES

## OCI Calls vs. CT-Lib

- Some of functions of OCI have no equivalents in Ct-Library. The functionality must be implemented either in SQL or in the C application directly. For example, following functions of OCI:

`OCIserverVersion()`

### **Transaction functions:**

`OCItransCommit()`  
`OCItransRollback()`  
`OCItransPrepare()`

### **Navigational Functions:**

`OCIcacheFlush()`  
`OCIcacheRefresh()`

### **Date Functions:**

`OCIdateCheck()`  
`OCIdateLastDate()`  
`OCIdateNextDate()`  
`OCIintervalAdd()`

### **Table Functions:**

`OCITableSize()`  
`OCITableExists()`



# DATA MIGRATION EXAMPLE



# DATA MIGRATION EXAMPLE

## Oracle ASCII Flat File Export

The Oracle tools import/export are very fast data extracts, but store the data in binary form only. To create an ASCII flat file export from Oracle, the spool command has to be used.

```
select to_char(sysdate, 'Dy DD-Mon-YYYY HH24:MI:SS') as "Start Time"
from dual;
clear breaks
clear columns
set verify off;
set heading off;
set trimspool on ;
set flush off;
set pagesize 0;
set linesize 475;
set arraysize 5000;
set feedback off;
set showmode off;
set colsep '~' ;
set recsep '@';
set term off;
spool '/oracle_table_output/emp.dat';
SELECT EMPNO,ENAME, JOB,MGR,to_char(HIREDATE, 'mm/dd/yy
hh:mi:ssAM'),
       SAL, COMM, DEPTNO
FROM EMP ;
SPOOL OFF ;
set term on;
select to_char(sysdate, 'Dy DD-Mon-YYYY HH24:MI:SS') as "End
Time" from dual;
```



# DATA MIGRATION EXAMPLE

## Express Connect Data Access Option for Oracle

Using Sybase ASE, you can join Oracle tables with tables in Sybase ASE. Access to these objects through Sybase ASE is transparent to the application and allows you to copy data from Oracle tables to Sybase ASE tables. The main advantage is that ECDA takes care of the data type conversions.

```
sp_addserver ORACLEDC, direct_connect, ORACLEDC
```

In the Oracle database there is the following table:

```
example_table  
(id_num int,  
name varchar(30),  
phone varchar(20) null,  
birthdate date null)
```

In the Sybase ASE database the following proxy tables will be created:

```
create existing table example_table  
(id_num int,  
name varchar(30),  
phone varchar(20) null,  
birthdate smalldatetime null)  
external table  
at 'ORACLEDC.oradb..example_table'
```



# DATA MIGRATION EXAMPLE

## Sybase Replication Server

Depending of the availability requirements of your systems, the data move might exceed the threshold to be completed entirely during the final switch over. It is not recommended that you write incremental load scripts yourself.

### For the initial load:

- Install Sybase Replication Server with Express Connect for Oracle
- Establish a one –way replication for the selected tables
- In order to minimize the impact on the servers the initial load of the data should be done outside the Replication Server.
- ECDA method
  - Copy the data from Oracle to Sybase ASE via ECDA with an INSERT statement.
- Flat File Method
  - Unload the data with a custom Oracle PL/SQL spooling procedure.
  - Load the data with the bcp tool.

### For incremental loads:

- No further action necessary. The data in these tables are automatically in sync.



# SESSION SUMMARY



## SESSION SUMMARY

- Check for incompatibilities before you start the migration project.
- Know your options. Just because there is not a direct migration path, alternatives are available.
- Sometimes the best migration is to not migrate it in the first place. Sybase ASE has a completely different index strategy. Migrating Oracle's index strategy is pointless. A new index strategy has to be developed.



## Q&A

**Thank You.**

**Peter Dobler, Dobler Consulting**

**[pdobler@doblerconsulting.com](mailto:pdobler@doblerconsulting.com)**