

**ORACLE®**



**ORACLE®**



## **Database Migration Case Study**

Frank Wang  
Software Development Manager,  
Oracle Partner Solution Center, Shenzhen

# Agenda

- Case Background
  - ISV
  - Application
  - Why Needs Migration
- Migration Challenges
- Planning
- Tools
- Migration Process
  - Schema Migration
  - Data Migration
  - Application Migration
- Issues & Solutions
- Summary

# Case Background

## ISV – Shine Technology

- Shine Technology, is a renowned professional solution provider engaging in the development of application software for the domestic financial and securities communities.
- Shine Technology maintained its leadership among the software suppliers of primary and secondary clearing solutions for the securities industries in China
- the clearing products of Shine Technology captured market share of 100% and 80% in terms of primary clearing and secondary clearing solutions respectively.

# Case Background

## Application – E-SIM

- Securities Management Platform, Shine's flagship product
- The E-SIM is currently applicable in securities clearing, fund clearing and management, accounts opening and depository operation and also supports for fund transfer and management between bank and securities sectors.
- The major target customer of E-SIM application include clearing, treasury, finance and sales division of the securities entity headquarters, their sale office network and other relevant working units.

# Case Background

## Why migrate ?





 Oracle Differentiator

### CHALLENGES

- Existing hardware and database can not handle increasing number of data to process
- Requires absolute accuracy of the computing result
- Transactions are correlated, can only have one giant transaction for all data process and it created a huge challenge to the underlying database
- The process must be done in a short period of time to ensure the opening of next day's market.



### ORACLE'S SOLUTION

- SQL\*Loader for fast data loading 
- Enhanced SQL & PL/SQL engine to provide effective and high performance data processing
- MVRC mechanism ensures less dead lock and no lock escalation under high pressure 
- RAC for great performance, extensibility and manageability 
- Proven ability to handle large volume of data with great performance 



### VALUE

- Faster system
- Extensible for future business growth
- Better manageability

# Migration Challenges

- No Oracle Experts in Shine Technology's development team
- Customer required the migration in short period of time
- Large database to migrate
  - Multiple source databases
  - Hundreds of tables
  - Dozens of stored procedures (most of them are complex)
  - T-SQL specific features used (syntax/functions/procedures)

# Migration Planning

- Source Database Analysis
  - Use of auto increment ID type
  - Massive use of temporary tables
  - Invoking of shell commands
  - No UDT
  - Most of procedures are complex (over 200 lines)
- Migration Strategy
  - Step 1 : Migrate Schema
  - Step 2 : Migrate Data
  - Step 3 : Convert Stored Procedures and Triggers
  - Step 4 : Test Migrated Application
  - Step 5 : Create packaging scripts
- Method
  - Use automation tools to migrate schema and data
  - Use automation tools to migrate procedures
  - Manual verification and fixing of non-migrated procedures

# Migration Planning (cont'd)

- Use help from Oracle Partner Solution Center, Shenzhen
  - Free services to enable partners with Oracle products
  - Database migration, tuning, AS migration, embedding
  - Provide migration best practices to Shine Technology
  - Oracle's migration experts will assist Shine's developer to perform the migration

# Migration Tools

- **Oracle Migration Workbench (OMWB)**
  - Free Tool – Downloadable from <http://otn.oracle.com>
  - Supports multiple 3rd party RMDBS, including SQLServer 7/2000.
  - Provides complete set of features - can migrate schema, procedures ,functions, triggers and data
- **SQLDeveloper Migration Workbench**
  - New product (also free), as a plug-in of new SQLDeveloper
  - Currently supports only SQLServer, Access and MySQL
  - Built with better T-SQL parsing/converting capability, and provides better code quality and accessibility
- **OMWB will be used to migrate schema and data**
- **SQLDeveloper will be used to help convert procedures**

# SQLDeveloper Migration Workbench

The screenshot displays the SQL Developer Migration Workbench interface. It features two side-by-side editors comparing SQL code. The left editor, titled 'Scratch Editor', shows a script with several lines of SQL, including a SELECT statement and a SET statement. The right editor, titled 'Oracle Diff', shows a similar script with a CREATE OR REPLACE PROCEDURE statement. The code in both editors is highlighted in green and blue, indicating differences or specific sections of interest. The interface includes a toolbar with navigation icons and a status bar at the bottom.

```
(View-Only)
, @ComparePrice money OUTPUT
, @ListPrice money OUT
AS
SELECT p.[Name] AS Product, p.ListPrice AS 'List Price
FROM Production.Product AS p
JOIN Production.ProductSubcategory AS s
ON p.ProductSubcategoryID = s.ProductSubcategoryID
WHERE s.[Name] LIKE @Product AND p.ListPrice < @MaxPri
-- Populate the output variable @ListPrice.
SET @ListPrice = (SELECT MAX(p.ListPrice)
FROM Production.Product AS p
JOIN Production.ProductSubcategory AS s
ON p.ProductSubcategoryID = s.ProductSubcategory
WHERE s.[Name] LIKE @Product AND p.ListPrice < @Me
-- Populate the output variable @compareprice.
SET @ComparePrice = @MaxPrice;
```

```
(View-Only)
CREATE OR REPLACE PROCEDURE uspGetList
(
v_Product IN VARCHAR2,
v_MaxPrice IN NUMBER,
v_ComparePrice OUT NUMBER,
v_ListPrice OUT NUMBER,
cv_1 IN OUT SYS_REFCURSOR
)
AS
BEGIN
OPEN cv_1 FOR
SELECT p.NAME Product,
p.ListPrice "List Price"
FROM Product p
JOIN ProductSubcategory s
ON p.ProductSubcategoryID = s.ProductSubcategoryID
WHERE s.NAME LIKE v_Product
AND p.ListPrice < v_MaxPrice;
-- Populate the output variable @ListPrice.
SELECT MAX(p.ListPrice)
INTO v_ListPrice
FROM Product p
JOIN ProductSubcategory s
ON p.ProductSubcategoryID = s.ProductSubcategoryID
WHERE s.NAME LIKE v_Product
AND p.ListPrice < v_MaxPrice;
-- Populate the output variable @compareprice
```

# Work Mode

- Shine Technology Engineer explains the application architecture to Oracle PSC staff
- Analysis of source DB together to define migration strategy and plan
  - Oracle has rich experience on such cases
  - Oracle will lead this action
- Schema/Data Migration
  - Use automation tools
  - Shine Technology engineer examines the results for correctness, as the application logic is familiar to them
  - Oracle staff helps in syntax equivalence and issues
- Procedure Migration
  - Tool automatically generate and produce results
  - Oracle staff helps resolve conversion issues
  - Manually convert complex & non-standard statements
- Test & Performance Tuning

# Schema Migration

- OMWB captures source database
- Converts source db to Oracle Model
  - Database => Schema
  - ID => Sequence + Trigger
  - Automatically convert definitions of Table, View, Index, Constraints (All objects exclude procedures, functions & triggers)
- Export DDL scripts for migrated Oracle database
- Apply DDL scripts to test environment
- Validate the result
  - OMWB provides reports
  - Examine the correctness

# Data Migration

- OMWB to generate data export scripts
  - Use BCP to unload data into text files
  - One table to each file
- OMWB generates data load scripts
  - Scripts to use Oracle SQL\*Loader
  - Automatically create SQL\*Loader control files
- Manually check the date/time format in ctl file
- Examine failed rows reported by SQL\*Loader
  - Invalid constraints
    - NOT NULL, SQL Server allows '' (Empty String). But Oracle treats it as NULL. Need human decision (Remove NOT NULL constraint or use ' ')
  - Redo the failed loads

# Procedure Migration

- Export DDL scripts from SQLServer
- Use SQLDeveloper Migration Workbench to migrate the scripts and produce the initial Oracle PL/SQL version
- Compile to find syntax issues
  - Manually fix
- Review codes to find semantic issues
  - Manually fix
- SQLDeveloper Migration Workbench produces good code quality
  - Easy to spot issues with inline comments
  - 80% + codes are correct semantically

# Test & Performance Tuning

- Shine reuses its existing functional test plan
- Oracle helps to test performance and improve
  - Adjust system settings
  - Find TOP SQL and tuning

# Issues & Solutions

## Temporary Table

- SQLServer's temporary table can preserve data across session, while Oracle's global temporary table can only preserve data within a single transaction or a session.
- Workbench can generate temporary table definition for us
- We need to manually decide how to migrate SQLServer temporary tables in stored procedures
- Local temporary table – translate to Oracle Global Temporary Table directly
- Global Temporary Table (##), some temporary table is designed to store results between sessions (## global)
  - Consider to use normal table in Oracle
  - Rewrite query to avoid use of middle tier of query results (also applies to local temporary table)

# Issues & Solutions

## Execute Shell Commands

- EXEC XP\_CMDSHHELL 'copy a.txt b.txt',NO\_OUTPUT
- Oracle has no direct matching feature
- Can write a external procedure to do the work
  - Needs extra configuration work, including compile
  - Maybe not convenient for a packaged application
- Alternative: use DBMS\_SCHEDULER

```
EXECUTE IMMEDIATE '
    dbms_scheduler.create_job(
        job_name => yourName -- better be random
        job_type => ''EXECUTABLE'',
        job_action => 'copy a.txt b.txt',
        enabled => TRUE); '
DBMS_SCHEDULER.run_job(yourName);
```

# Issues & Solutions

## Identify Column

- In Oracle, there's no such auto-increment data type
- Use Trigger + Sequence to simulate
- Sequence is to generate sequential series of numbers
  - Accessed through CURRENTVAL and NEXTVAL
  - `SELECT YOUR_SEQUENCE.NEXTVAL FROM DUAL`
- Trigger to get NEXTAL every time a INSERT issued.
- OMWB will automatically create sequence and triggers
- But one may want to consider remove triggers on some table and manually get sequence value within the SQL statement. For better performance under certain situation.
- When doing data migration, triggers should be disabled. And sequence should be defined well to avoid primary key value conflicts
- @@Identity

# Issues & Solutions

## Exception Handling

- Exception Handling
  - T-SQL uses @@error to detect error
  - Continues after exception
  - Oracle's exception mechanism is similar to JAVA

```
BEGIN
```

```
...
```

```
EXCEPTION
```

```
WHEN ZERO_DIVIDE THEN -- handles 'division by  
zero' error          ... ;
```

```
WHEN OTHERS THEN -- handles all other errors
```

```
...
```

```
END;
```

- One needs to consider the application logic to decide the migration approach.

# Issues & Solutions

## Others

- Functions
  - Workbench will help to convert most of them correctly
  - Workbench will create a simulation package to simulate T-SQL functions
- Convert / Cast => TO\_NUMBER, TO\_DATE, TO\_CHAR...
- IS\_DATE
  - No matching function, need to implement a user defined function to simulate it.

# Summary

## Why migrate with PSC

- 1 Proven methods
- 2 Proven tools
- 3 Oracle product experts
- 4 Migration experts
- 5 Free services
- 6 Reduce migration cost
- 7 Less project time
- 8 Faster to market
- 9 Better migration quality
- 10 Localized support, no language barrier



**Quote Attribution**  
Director, Shine Technology


“The migration is a complete success and we learned a lot from it. And it can not be done without PSC Shenzhen’s help.”

# Resources

- Oracle Migration Technology Center
  - <http://www.oracle.com/technology/tech/migration/index.html>
- Partner Solution Center – Shenzhen
  - <http://www.oracle.com/cdc/sspocc/index.html>
  - [sspocc\\_ww@oracle.com](mailto:sspocc_ww@oracle.com)

# For More Information

search.oracle.com



or

oracle.com

**ORACLE®**