

ORACLE®

“This presentation is for informational purposes only and may not be incorporated into a contract or agreement.”

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decision. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

# Oracle JDeveloper 10g for Database Developers and DBAs

---

**Sue Harper**

Product Management

Application Development Tools

Oracle Corporation

# Get Information



[otn.oracle.com](http://otn.oracle.com)

<http://www.oracle.com/technology/products/jdev>

**Free Software Download**

**JDeveloper Discussion Forum**

**Tutorials, Demos, How-To's and More**

# Agenda

- JDeveloper: Database Support
- Database and SQL Development
- Schema Modeling and Offline Editing
- PL/SQL Development and Debugging
- Questions and Answers

# Why Database Support?

- The database is an important part of web application development.
- Users can develop the database schema objects together with the Java and XML portions of their application.
- Developers have more flexibility and control with a consistent interface.
- Data support is a valuable addition for a true IDE for web application developers.

# The Features in Brief

- Browse and edit online schema objects
- Use the SQL Worksheet to execute SQL queries
- Visualize existing database structures (Oracle & others)
- Visually design new databases, then generate & execute DDL
- Synchronize design and implementation by reverse engineering & reconciliation
- Create, edit and debug PL/SQL
- Use code completion to facilitate PL/SQL editing



# Agenda

- JDeveloper: Database Support
- ➔ Database and SQL Development
- Schema Modeling and Offline Editing
- PL/SQL Development and Debugging
- Questions and Answers

# Connection Manager

- Create and test connections with a wizard
- Store often-used connections
- Import and export connections
- Reuse connections throughout lifecycle
  - Database, PL/SQL Development
  - Application Development
  - Deployment
- Additional connection types
  - Application Server, WebDAV Server, SOAP Server, Oracle SCM Server, UDDI Registry, Oracle Designer

# Browsing the Online Database

- View schema objects from the Navigator
  - Tables (Indexes, Columns, Constraints, Data)
  - Views
  - Synonyms
  - Sequences
  - PL/SQL subprograms
    - Triggers, packages, procedures
  - Object Types
  - Materialized Views (Snapshots)
  - Java Resources
- User-defined filters at all levels



# Creating and Editing Database Objects (Online)

- Create and edit schema objects
  - PL/SQL subprograms
  - Tables
  - Views
  - Triggers
  - Sequences
  - Synonyms
  - Object Types

Create Table - Step 5 of 8: Foreign Keys

Foreign Keys

DEMOTAB\_FK1

Add

Remove

Constraint Name: DEMOTAB\_FK1  Enabled

Referenced Schema: TRAVEL

Referenced Table: COUNTRIES

Referenced Constraint: COUNTRIES\_PK

Associations:

| Column | Referenced Column |
|--------|-------------------|
| ID     | COUNTRY_ID        |

On Delete: RESTRICT

Help < Back Next > Finish Cancel

# SQL Worksheet

- Execute SQL statements
- Browse query results
- Explain Plan
- SQL History
- Incremental Fetch

The screenshot displays the Oracle SQL Worksheet interface. The top section, titled "Enter SQL Statement:", contains the following SQL query:

```
SELECT e.LAST_NAME,
       e.SALARY,
       d.DEPARTMENT_NAME,
       j.JOB_TITLE,
       l.CITY,
       c.COUNTRY_NAME
FROM EMPLOYEES e,
     DEPARTMENTS d,
     JOBS j,
     LOCATIONS l,
     COUNTRIES c,
     REGIONS r
WHERE e.DEPARTMENT_ID = d.DEPARTMENT_ID AND d.LOCATION_ID = l.LOCATION_ID AND l.COUNTRY_ID =
```

The bottom section, titled "Explain Plan Results", shows a tree view of the execution plan. The plan starts with a "SELECT STATEMENT" operation, which is a "HASH JOIN". This join involves "TABLE ACCESS(FULL) HR.JOBS" and another "HASH JOIN". The second "HASH JOIN" is a "NESTED LOOPS" join involving "TABLE ACCESS(FULL) HR.DEPARTMENTS" and another "NESTED LOOPS". This nested loop involves a "MERGE JOIN" between "TABLE ACCESS(FULL) HR.DEPARTMENTS" and "INDEX(UNIQUE SCAN) HR.COUNTRY\_C\_ID\_PK". The "MERGE JOIN" also involves "TABLE ACCESS(BY INDEX ROWID) HR.LOCATIONS", which is accessed via "INDEX(FULL SCAN) HR.LOC\_ID\_PK".

| Operation                                 | Optimizer | Cost | Cardinality | Bytes | Partitions |
|---|-----------|------|-------------|-------|------------|
| SELECT STATEMENT                          | ALL_ROWS  | 13   | 106         | 10812 |            |
| HASH JOIN                                 |           | 13   | 106         | 10812 |            |
| TABLE ACCESS(FULL) HR.JOBS                | ANALYZED  | 3    | 19          | 513   |            |
| HASH JOIN                                 |           | 10   | 106         | 7950  |            |
| NESTED LOOPS                              |           | 6    | 27          | 1377  |            |
| NESTED LOOPS                              |           | 6    | 27          | 1296  |            |
| MERGE JOIN                                |           | 6    | 27          | 918   |            |
| TABLE ACCESS(BY INDEX ROWID) HR.LOCATIONS | ANALYZED  | 2    | 23          | 345   |            |
| INDEX(FULL SCAN) HR.LOC_ID_PK             | ANALYZED  | 1    | 23          |       |            |
| SORT(JOIN)                                |           | 4    | 27          | 513   |            |
| TABLE ACCESS(FULL) HR.DEPARTMENTS         | ANALYZED  | 3    | 27          | 513   |            |
| INDEX(UNIQUE SCAN) HR.COUNTRY_C_ID_PK     | ANALYZED  | 0    | 1           | 14    |            |

ORACLE®

D E M O N S T R A T I O N

# Database Browser

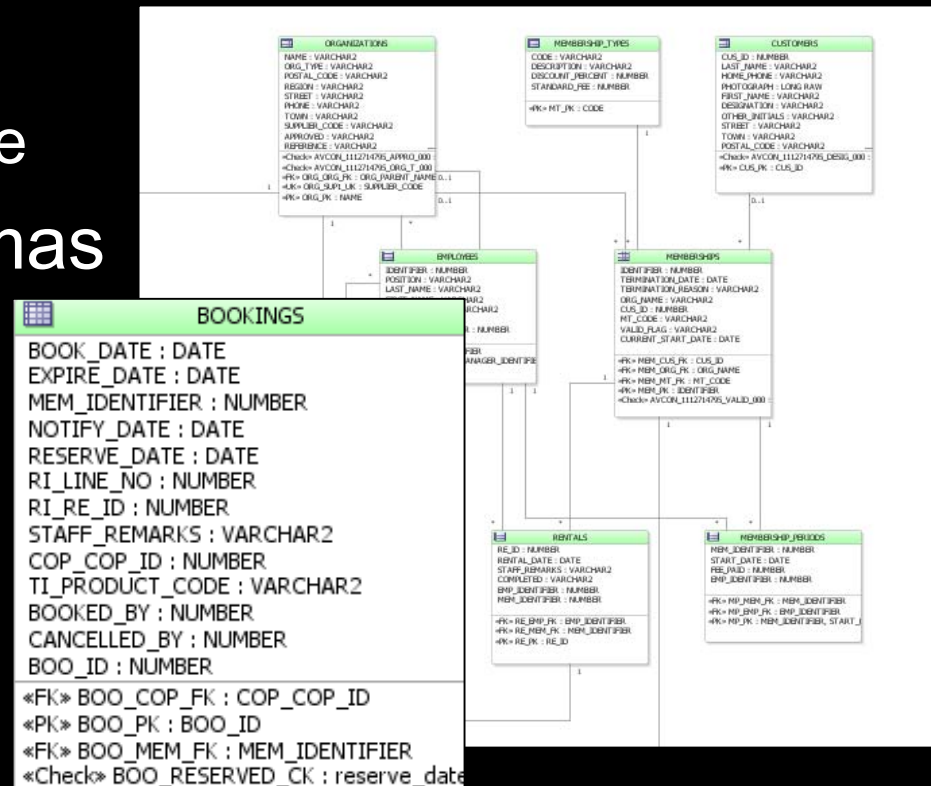
ORACLE®

# Agenda

- JDeveloper: Database Support
- Database and SQL Development
- ➔ Schema Modeling and Offline Editing
- PL/SQL Development and Debugging
- Questions and Answers

# Schema Modeling

- Design-capture existing schema objects
  - Drag & Drop
  - Import from database
- Visually design schemas
  - Tables
  - Views
  - Synonyms
  - Sequences



# Defining Views

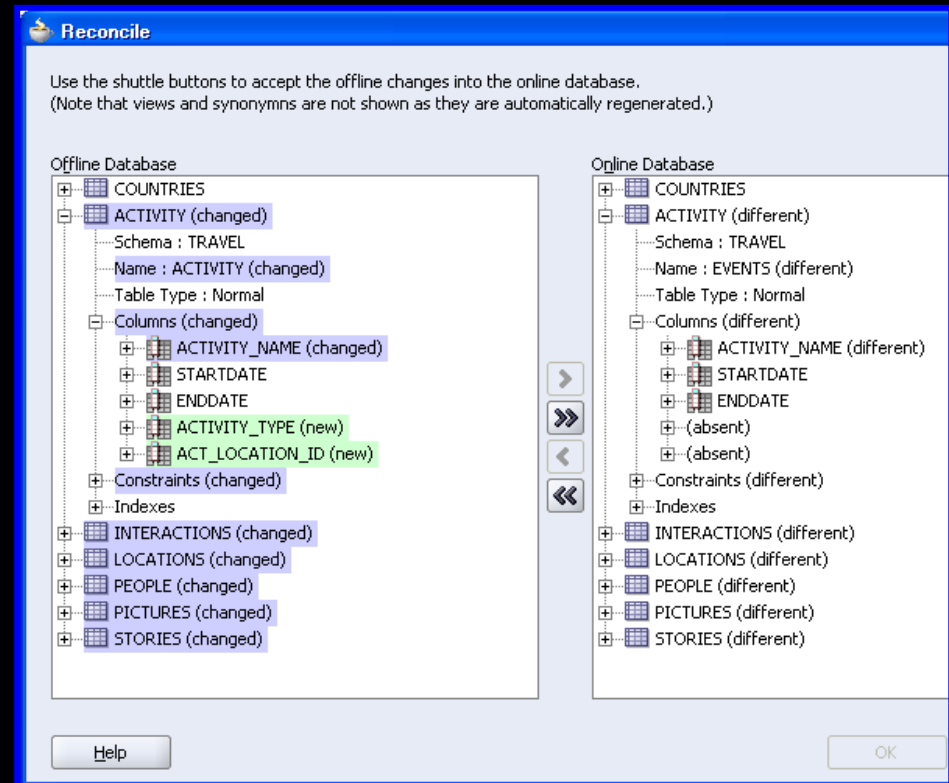
- Visually
- Declaratively
- Online
- Offline

The image displays two screenshots from Oracle SQL Developer. The top screenshot is the 'Edit Offline View' dialog box, which is used for defining and editing views. It features a 'View Information' tree on the left, a 'SELECT clause' editor on the right, and an 'Expression Palette' at the bottom right. The 'SELECT clause' editor shows a list of columns: e.EMPLOYEE\_ID, e.JOB\_ID, e.MANAGER\_ID, e.DEPARTMENT\_ID, and d.LOCATION\_ID. The 'Expression Palette' lists database objects: e (EMPLOYEES), d (DEPARTMENTS), j (JOBS), l (LOCATIONS), c (COUNTRIES), and r (REGIONS). A 'Validate' button is present in the editor area.

The bottom screenshot is a visual query diagram showing the relationships between tables. The tables are represented as boxes: EMP\_DETAILS\_VIEW, EMPLOYEES, DEPARTMENTS, JOBS, LOCATIONS, COUNTRIES, and REGIONS. Lines connect the tables to show their relationships. The EMP\_DETAILS\_VIEW table is highlighted with a blue border, and its columns (e.EMPLOYEE\_ID, e.JOB\_ID, e.MANAGER\_ID, e.DEPARTMENT\_ID, e.COMMISSION\_PCT) are listed. The EMPLOYEES table is also highlighted, and its columns (e.COMMISSION\_PCT, e.SALARY, e.LAST\_NAME, e.FIRST\_NAME, e.MANAGER\_ID, e.DEPARTMENT\_ID, e.JOB\_ID, e.LOCATION\_ID) are listed. The DEPARTMENTS table is highlighted, and its columns (d.DEPARTMENT\_NAME, d.LOCATION\_ID) are listed. The JOBS table is highlighted, and its column (j.JOB\_TITLE) is listed. The LOCATIONS table is highlighted, and its columns (l.STATE\_PROVINCE, l.CITY, l.COUNTRY\_ID) are listed. The COUNTRIES table is highlighted, and its column (c.COUNTRY\_NAME) is listed. The REGIONS table is highlighted, and its column (r.REGION\_NAME) is listed.

# Generate DDL

- Generate from the diagram to
  - Create or alter objects
- Create SQL files
  - Prefix with schema name
  - Include spooling commands
- Generate directly to database
- Reconcile with database
  - Create or alter objects
- Review modifications
- Perform manual reconciliation



ORACLE®

D E M O N S T R A T I O N

# Schema Modeling

ORACLE®

# Agenda

- JDeveloper: Database Support
- Database and SQL Development
- Schema Modeling and Offline Editing
- ➔ PL/SQL Development and Debugging
- Questions and Answers

# Editing and Compiling PL/SQL

- Comprehensive editor features
  - Bookmarks
  - Macros
  - Code Templates
  - Search and Replace
  - Syntax Highlighting
  - ...
- PL/SQL Code Insight
- Synchronized Structure window
- Errors reported in Log window



# Running PL/SQL



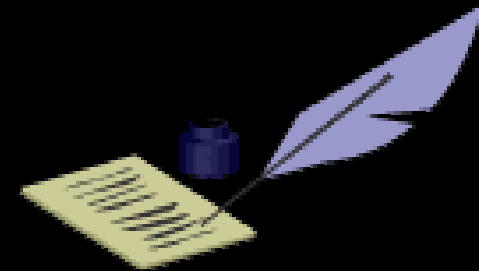
- Run Procedures, Functions, and Packages
  - DBMS\_OUTPUT
  - Function return values
  - OUT parameters
- Run PL/SQL dialog
  - Specify run target
  - Shows parameter information
  - Generates editable PL/SQL block for populating parameters

# Debugging PL/SQL

- Supported with Oracle8*i* and later
  - Uses JDWP implementation starting with Oracle9*i* Release 2
- Control program execution
  - Step Into, Step Over, Run to Cursor, etc
- Inspect and modify variables
  - Tooltips in Code Editor
- Configure breakpoint conditions



# PL/SQL Debugging Requirements



- Compiled with debug information
  - JDeveloper “Database Connections” preferences panel
  - ALTER SESSION SET PLSQL\_DEBUG = TRUE
  - ALTER <PROG\_UNIT> COMPILE DEBUG
- Oracle8i and Oracle9i specifics
  - CREATE ANY PROCEDURE (debugging other schemas)
- Oracle9i Release 2 and Oracle 10g specifics
  - DEBUG ANY PROCEDURE
  - DEBUG CONNECT SESSION
  - Compiled in Interpreted (not Native) mode

# Additional PL/SQL Debugger Features

- Debug Java stored procedures
- Seamless debugging between server-side Java and PL/SQL
- Inspect elements of PL/SQL Collections (Tables, Records, etc)
- Remotely debug PL/SQL calls from any client



ORACLE®

D E M O N S T R A T I O N

# PL/SQL Development

ORACLE®

# What's New in JDeveloper 10.1.3?

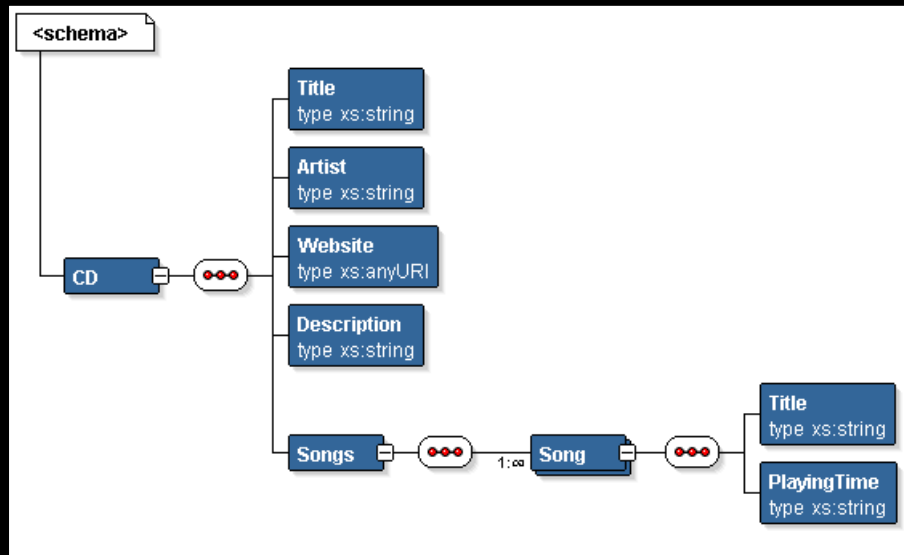
- Support for additional schema objects (create, update, visualize and generate...)
  - Support for views (visualize with relationships to tables)
  - Object types, PL/SQL packages, procedures, functions
  - Sequences, synonyms, expression based indexes
  - System Defined Types: SDO\_Geometry, XMLTypes...
- Foreign database support for DB2, Informix, MySQL, SQL Server and Sybase
- Visual reconcile tool for the generation of scripts

# More Database-Related Features

- Application Development Framework (Oracle ADF)
- Database Web Services
- SQLJ and JPublisher
- Oracle TopLink
- Java Stored Procedures
- OLAP (BI Beans)

# XML Schema

- Register schemas in XDB
- Review XDB resources using WebDav
- Generate an XML instance document from a registered XSD schema



# Why JDeveloper?

- Pure Java
  - Supported on Windows, Linux, Mac OSX, Solaris, HP/UX
- Multiple Database Support
  - Oracle / Oracle Lite, SQL Server, DB2, Sybase, MySQL
- No Oracle client install is required
- Integrated database modeling
- PL/SQL debugging is powerful
  - Remote PL/SQL debugging
  - Seamless Server-Side Java and PL/SQL debugging
  - Automatic detection of variables
  - Drill-down into composite variable types
- Complete end-to-end development

# Heard about Project Raptor?

- Oracle's graphical alternative to SQL\*Plus
- Allows the Database Developer a convenient way to perform basic tasks.
  - Browse database objects
  - Run SQL statements and SQL scripts
  - Edit and debug PL/SQL statements



Oracle Raptor : VIEW OE.PRODUCTS@OE

File Edit View Navigate Run Debug Source Tools Help

Connections Reports OE PRODUCTS

Columns Data Grants Dependencies Details SQL Grants

Connections

- OE
  - Tables
    - CATEGORIES\_TAB
    - CUSTOMERS
    - INVENTORIES
    - ORDER\_ITEMS
    - ORDERS
    - PRODUCT\_DESCRIPTIONS
    - PRODUCT\_INFORMATION
    - PRODUCT\_REF\_LIST\_NESTEDTA
    - PROMOTIONS
    - PS\_TXN
    - STYLESHEET\_TAB
    - SUBCATEGORY\_REF\_LIST\_NES
    - WAREHOUSES
  - Views
    - BOMBAY\_INVENTORY
    - DEPTVIEW
    - OC\_CORPORATE\_CUSTOMERS
    - OC\_CUSTOMERS
    - OC\_INVENTORIES
    - OC\_ORDERS
    - OC\_PRODUCT\_INFORMATION
    - PRODUCT\_PRICES
    - PRODUCTS
    - SYDNEY\_INVENTORY
    - TORONTO\_INVENTORY

```

CREATE OR REPLACE FORCE VIEW "OE"."PRODUCTS" ("PRODUCT_ID", "LANGUAGE_ID", "P
SELECT i.product_id
,      d.language_id
,      CASE WHEN d.language_id IS NOT NULL
          THEN d.translated_name
          ELSE TRANSLATE(i.product_name USING NCHAR_CS)
        END      AS product_name
,      i.category_id
,      CASE WHEN d.language_id IS NOT NULL
          THEN d.translated_description
          ELSE TRANSLATE(i.product_description USING NCHAR_CS)
        END      AS product_description
,      i.weight_class
,      i.warranty_period
,      i.supplier_id
,      i.product_status
,      i.list_price
,      i.min_price

```

Messages - Log

```

Opening:select c.column_name, case when data_type = 'CHAR' then      data_type||'('

```

All Rows Fetched

Editing 8M of 12M

# Summary



- Industry leading query development, also available to develop ADF View Objects.
- Industry leading integration of development, editing, running and debugging for Java, SQL & PL/SQL in one product.
- Developing web applications is an end to end process, from database to class and back.
- Offering support for both online and offline object creation and manipulation with full support for reconciling an offline schema definition against a live database.

A large, stylized logo in the background consisting of a grey 'Q', a red ampersand '&', and a grey 'A'. The text 'QUESTIONS' and 'ANSWERS' is overlaid on this logo.

QUESTIONS  
ANSWERS

ORACLE®