

# Oracle Database Management Tools

## A Comprehensive Guide for DBAs

---

## **Learning Objectives**

By the end of this session, you will be able to:

- List common Oracle database management tools
  - Be familiar with using SQL\*Plus
  - Describe SQL Developer
  - Describe EM Express
  - Describe Oracle Enterprise Manager Cloud Control
  - Describe the Database Management Cloud Service
-

# Oracle Database Management Tools Overview

To manage Oracle databases, we have a list of management tools. Each tool has its own pros and cons.

## Available Tools:

- **SQL\*Plus** - Command-line utility
  - **SQL Developer** - Java-based GUI interface
  - **SQLcl** - Modern command-line tool
  - **EM Express** - Web-based GUI for local databases
  - **Oracle Enterprise Manager Cloud Control (OEM)** - Enterprise management system
  - **Database Management Cloud Service** - Cloud-based DBA solution
  - **Third-party tools** - Commercial database management solutions
-

# **SQL\*Plus**

## **Overview**

SQL\*Plus is probably the **most common tool** used by DBAs to manage Oracle databases.

### **What is SQL\*Plus?**

- Command-line utility that allows users to type commands in the command prompt
- Lightweight utility that can be easily installed and used
- Can run scripts and be called from script files
- Has been around since very early Oracle versions

### **Key Features**

✓ Submit SQL statements to the database ✓ Run scripts against Oracle databases ✓  
Formatting commands for displayed data ✓ Can be used interactively or in scripts ✓ Callable  
from shell scripts

---

# **SQL\*Plus - Advantages**

## **Why SQL\*Plus is Popular**

### **Advantages:**

- Lightweight and easy to install
- Widely used among DBAs
- Can run scripts from command line
- Can be called from other scripts
- Available on all platforms
- Included with Oracle Database software

### **Best For:**

- Running automated scripts
  - Quick database queries
  - Administrative tasks
  - Shell script integration
-

# **SQL\*Plus - Limitations**

## **Challenges with SQL\*Plus**

### **Disadvantages:**

- Tedious to display data with many columns
- Formatting retrieved data is time-consuming
- Not easy to see command history
- Re-executing previous commands requires using history command
- No built-in visual interface

**Note:** Third-party utilities exist to fix some issues, but they are not supported by Oracle

---

# SQL Developer

## Overview

SQL Developer is a **Java-based GUI interface** that allows us to submit SQL statements to the database.

## Key Information

- Oracle has been improving SQL Developer extensively in recent years
- Rich with advanced options and features
- Free to download and use

## Distribution Changes

- **Previously:** Shipped with Oracle Database software
  - **Currently (from Oracle 19c):** Downloaded separately from Oracle website
  - Easy to download and install
-

# **SQL Developer - Features**

## **Main Features**

### **Data Display:**

- Retrieved data displayed in formatted tabular style
- Easy export of data to external files
- Professional presentation of query results

### **User Interface:**

- GUI-based interface
- Shortcuts for frequent operations
- Visual database object browser
- Query builder

### **Productivity:**

- Advanced code completion
  - Debugging capabilities
  - Data modeling tools
-

# SQLcl (SQL Command Line)

## Overview

SQLcl is a **modern command-line utility** that comes with SQL Developer.

## Key Information

- Developed to **replace SQL\*Plus**
  - To invoke from command line: type `sql` (not `sqlcl`)
  - Still shipped with Oracle Database software
  - Latest version can be downloaded separately
-

# **SQLcl - Advanced Features**

## **Features That SQL\*Plus Lacks**

### **Modern Capabilities:**

- **Inline editor** - Edit commands directly
- **Change management** - Track modifications
- **Command history** - Easy access to previous commands
- **Auto-complete** - Tab key completion for object names and keywords
- **SQL\*Plus compatibility** - Supports SQL\*Plus commands

### **Why It's Better**

SQLcl is a beautiful and feature-rich utility that covers all SQL\*Plus limitations while maintaining backward compatibility.

**Note:** Despite being better, SQL\*Plus is still more widely known and used among DBAs

---

# Oracle EM Express

## Overview

**Oracle Enterprise Manager Database Express (EM Express)** is a web-based GUI interface for monitoring and partially managing local databases.

## Key Characteristics

- Web-based interface
  - Configured at database creation time
  - **Local only** - Cannot connect to remote databases
  - Focused on monitoring, not full management
-

# **EM Express - Main Pages**

## **Three Major Pages**

- 1. Performance Hub**
  - Monitoring database performance
  - Real-time performance metrics
- 2. Tablespace Monitoring**
  - Storage management
  - Space utilization
- 3. SQL Performance Analyzer**
  - Query performance analysis
  - SQL tuning recommendations

**Note:** Despite looking simple, do not underestimate its value!

---

# **EM Express - Performance Monitoring**

## **Capabilities**

### **What EM Express Provides:**

- Visualize database performance indicators
- List sessions that stress the database
- Identify resource-consuming SQL statements
- Detect high wait events
- Performance troubleshooting

## **Performance Monitoring Excellence**

EM Express provides something difficult to achieve with other tools: comprehensive performance monitoring and troubleshooting.

---

# **EM Express - Licensing Requirements**

## **Important Licensing Information**

**To use Performance Hub features, you must own:**

- Diagnostics Pack license
- Tuning Pack license

**⚠ Warning:** These packs are **NOT** included in the Oracle Database Enterprise Edition license

**Cost Consideration:** Using the Performance Hub comes with additional licensing costs beyond the base database license.

---

# **Oracle Enterprise Manager Cloud Control (OEM)**

## **Overview**

OEM is an **advanced system management software** that provides comprehensive IT infrastructure management.

### **Full Name**

Oracle Enterprise Manager Cloud Control

### **Scope**

- Oracle technologies
  - Non-Oracle technologies
  - Complete IT infrastructure
-

# **OEM - Capabilities**

## **What OEM Offers**

### **Centralized Management:**

- Monitor multiple databases from single console
- Administration of complete Oracle IT infrastructure
- Lifecycle management functionality
- Web-based GUI interface

### **Full Management:**

- Unlike EM Express, OEM provides **full management** capabilities
  - Not limited to local databases
  - Enterprise-scale operations
-

# **OEM - Requirements**

## **Implementation Requirements**

### **Agent Installation:**

- Must install an agent on each system
- Agent registration required in OEM system
- Enables monitoring and management

### **Licensing:**

- Requires separate license
- Additional cost beyond database license

### **Learning Curve:**

- Some learning involved
  - More complex than simpler tools
-

# **Database Management Cloud Service**

## **Overview**

A **cloud service provided by Oracle** for managing Oracle databases.

## **Service Model**

- No installation required
- Subscribe to the service
- Use it
- Pay for usage

**Simple Deployment:** Subscribe → Use → Pay

---

# **Database Management Cloud Service - Features**

## **Key Capabilities**

### **Unified Console:**

- Manage on-premises databases
- Manage cloud databases
- Single interface for all databases

### **Specialized Focus:**

- Dedicated to Oracle databases only
- Not general IT infrastructure like OEM

### **Advanced Features:**

- Database fleet diagnostics
  - Performance tuning
  - Issue troubleshooting
  - Performance optimization
-

# **Database Management Cloud Service - Use Cases**

## **When to Consider This Service**

### **Recommended For:**

- Managing high number of Oracle databases
- Organizations with both on-premises and cloud databases
- Teams needing unified management console
- Businesses wanting to avoid infrastructure setup

### **Benefits:**

- No local installation required
  - Cloud-based scalability
  - Pay-as-you-go model
  - Always up-to-date features
-

# **Third-Party Tools**

## **Commercial Database Management Solutions**

**Overview:** The market provides third-party tools to manage Oracle databases.

### **Advantages:**

- Make monitoring and managing Oracle databases much easier
- Professional interfaces
- Advanced features
- Better user experience

### **Cost:**

---

- Require license purchase
- Additional expense beyond Oracle licensing

# **SQL\*Plus - Getting Started**

## **Why Focus on SQL\*Plus?**

### **Popularity:**

- Most common tool used by DBAs
- Been around since very early Oracle versions
- Still commonly used despite newer alternatives

### **Course Approach:**

- We will stick with SQL\*Plus in course practices
  - Once you know SQL\*Plus, switching to SQLcl is easy
  - Skills transfer directly
-

# Starting SQL\*Plus

**Command:** `sqlplus`

**Location:**

- Saved in `bin` directory under Oracle Home
- Added to PATH variable
- Visible to OS command line

## Methods of Invocation

### 1. Without Database Connection:

```
sqlplus /nolog
```

- Opens SQL\*Plus command prompt
  - No database connection
  - Use `connect` command to connect later
-

# **Connecting to Database - Normal User**

## **Connection Method**

### **Command Format:**

```
sqlplus username/password
```

### **Key Points:**

- Password can be omitted from command line
- If omitted, SQL\*Plus will prompt for password
- Password is hidden when typing
- Username is NOT case sensitive
- Password IS case sensitive

### **Multiple Instances:**

- Connects to database pointed by `ORACLE_SID` variable
  - If `ORACLE_SID` not set, connection fails
-

# **Connecting as SYS User**

## **Super User Connection**

### **SYS User:**

- Super user in Oracle databases
- Can do almost anything in the database
- **Not the same as SYSTEM user**

### **Required Phrase:**

- Must use `as sysdba` with SYS connections
  - This phrase is **mandatory** for all SYS connection attempts
-

# **OS Authentication - SYS Connection**

## **Connecting Without Password**

### **Command:**

```
sqlplus / as sysdba
```

### **Requirements:**

- OS user must belong to `dba` group (Linux)
- OS user must belong to `oradba` group (Windows)
- Typically only the software owner (Oracle user) belongs to this group

**Method Name:** This is called **OS Authentication** (Operating System Authentication)

---

# SYS Connection with Password

## For Non-DBA Group Users

### Command Format:

```
sqlplus sys/password as sysdba
```

### Requirements:

- User must know SYS password
- Must include `as sysdba` phrase
- Can be used by any OS user who knows the password

**Important:** The `as sysdba` phrase is **required** for all SYS connections, whether using OS authentication or password authentication.

---

# **Remote SQL\*Plus Connections**

## **Client Machine Access**

### **Local vs Remote:**

- Previous examples assume local database server
- SQL\*Plus can also be invoked from client machines
- Can connect via network connection

### **Network Connections:**

- Connect to Oracle databases remotely
- Requires network configuration
- Uses TNS (Transparent Network Substrate)

**Note:** Network connections will be covered in a separate lecture

---

# **Running Scripts with SQL\*Plus**

## **Command-Line Script Execution**

### **Script Content:**

- SQL statements
- PL/SQL code execution
- SQL\*Plus commands

**Important Note:** SQL\*Plus commands are **NOT** SQL statements - they are different!

---

# Script Execution Syntax

## Running Scripts from Command Line

### Command Format:

```
sqlplus username/password @scriptname
```

### Key Points:

- Use @ symbol before script filename
- File extension is **optional** (.sql assumed)
- Script executes and SQL\*Plus exits

**Use Case:** This method allows calling SQL\*Plus from shell scripts

---

# **Interactive SQL\*Plus Usage**

## **Working at SQL\*Plus Prompt**

### **After Connecting:**

1. SQL\*Plus opens command-line prompt
2. Waits for user to enter SQL statement
3. User types statement directly

### **Statement Rules:**

- Can be entered as multiple lines
  - **Must end with semicolon (;**
  - Press Enter after semicolon to execute
-

# **Running Scripts from SQL\*Plus Prompt**

## **Two Methods Available**

### **Method 1: Using @ Symbol**

```
@scriptname
```

### **Method 2: Using START Command**

```
start scriptname
```

### **Both methods:**

- File extension is optional
  - Execute the specified script
  - Work identically
-

# **SQL Developer - Installation**

## **Getting SQL Developer**

### **Download Information:**

- Not shipped with Oracle Database 19c and later
- Must download separately from Oracle website
- Free download

### **Installation:**

- Super easy to download
- Super easy to install
- Even easier to use

**Hands-On:** Installation will be covered in a practice lecture

---

## **SQLcl - Additional Information**

### **SQL\*Plus Alternative**

#### **Key Points:**

- Comes with SQL Developer
- Can be downloaded separately for latest version
- Modern alternative to SQL\*Plus

**Learning More:** Refer to Oracle documentation for detailed SQLcl information and advanced features

---

# **Summary - Management Tools**

## **What We Covered**

### **Tools Reviewed:**

1. SQL\*Plus - Command-line utility (most common)
2. SQL Developer - GUI interface
3. SQLcl - Modern command-line tool
4. EM Express - Web-based local monitoring
5. OEM Cloud Control - Enterprise management
6. Database Management Cloud Service - Cloud-based DBA

### **Each tool has:**

- Specific use cases
  - Advantages and disadvantages
  - Different licensing requirements
-

## **Summary - Learning Objectives Achieved**

### **You Should Now Be Able To:**

- ✓ List common Oracle database management tools
- ✓ Be familiar with using SQL\*Plus
- ✓ Describe SQL Developer
- ✓ Describe EM Express
- ✓ Describe Oracle Enterprise Manager Cloud Control
- ✓ Describe Database Management Cloud Service

### **Next Steps**

In the next lecture: Hands-on practice with SQL\*Plus utility

---

# **SQL\*Plus Practice Session**

## **Hands-On with SQL\*Plus**

---

### **Practice Session Overview**

#### **What You'll Learn**

##### **Focus Areas:**

- Different ways of invoking SQL\*Plus
- Logging into the database
- Operating system authentication
- Basic SQL\*Plus capabilities
- Common SQL\*Plus commands

##### **Platform:**

- Windows platform VM (demonstrated)
  - All steps applicable to Linux environments as well
-

# **Starting SQL\*Plus - Method 1**

## **Without Database Connection**

**Step 1:** Open Command Prompt

**Step 2:** Issue command:

```
sqlplus /nolog
```

**Result:**

---

- SQL\*Plus opens without connecting to any database
- SQL\*Plus command prompt appears
- Ready for connect command

# **Connecting to Database**

## **Using CONNECT Command**

### **Command:**

```
conn / as sysdba
```

### **Abbreviations:**

- `conn` is short for `connect`
- Nearly all SQL\*Plus commands have abbreviation forms

### **Result:**

- Connected to database as SYS
  - No password required (OS authentication)
-

# OS Authentication Explained

## Why No Password Required?

### Requirements Met:

- Logged-on user is Oracle
- Oracle user is member of `oradba` group (Windows)
- Oracle user is member of `dba` group (Linux)

**Method:** This is called **Operating System Authentication**

**Multiple Instances:** If multiple databases running, connects to instance pointed by `ORACLE_SID` variable

**Note:** OS Authentication will be covered in detail in a later lecture

---

## **Exiting SQL\*Plus**

### **Two Commands Available**

#### **Option 1:**

`exit`

#### **Option 2:**

`quit`

#### **Both commands:**

- Work the same way
  - Close SQL\*Plus session
  - Return to OS command prompt
-

## **Starting SQL\*Plus - Method 2**

### **With Immediate Connection**

#### **Command:**

```
sqlplus / as sysdba
```

#### **What Happens:**

- SQL\*Plus starts
  - Connects to database as sysdba
  - One command does both actions
-

# **Checking Current User**

## **SHOW USER Command**

**Command:**

```
show user
```

**Result When Connected as SYSDBA:**

- Current user displays as `SYS`
  - `SYS` is the super user for Oracle Database
  - Has full privileges
  - Can run almost any command
-

# **Connecting as Different Users**

## **SYSTEM User Connection**

### **Command:**

```
sqlplus system/password
```

### **Important:**

- Username is **NOT** case sensitive
- Password **IS** case sensitive

### **Result:**

- Connected as SYSTEM user
  - Different privileges than SYS
-

# **Connecting as HR User**

## **Application Schema User**

### **Command:**

```
conn hr/password
```

### **Result:**

- Connection successful
  - Now working as HR user
  - Access to HR schema objects
-

# **DESCRIBE Command**

## **Viewing Table Structure**

### **Command:**

```
desc employees
```

or

```
describe employees
```

### **Key Points:**

- `describe` or `desc` is a **SQL\*Plus command**, NOT a SQL statement
- `desc` is the short form
- SQL\*Plus commands are case insensitive

**Result:** Displays structure of the employees table (columns, data types, etc.)

---

# Running SQL Queries

## Sample Query

### Command:

```
SELECT employee_id, first_name, salary  
FROM employees  
WHERE department_id = 90;
```

### Important:

- Statement ends with **semicolon** (;
  - After typing semicolon and pressing Enter, statement executes
-

# **SQL\*Plus Buffer**

## **LIST Command**

### **Command:**

list

or short form:

l

### **Function:**

- Displays contents of SQL\*Plus buffer
  - Shows most recent executed SQL statement
  - Useful for reviewing and re-executing commands
-

# **EDIT Command**

## **Opening Buffer in Editor**

### **Command:**

`edit`

or short form:

`ed`

### **Function:**

- Opens SQL\*Plus buffer in standard editor (Notepad on Windows)
- Allows visual editing of SQL statement

**Potential Issue:** May require Administrator privileges depending on current directory permissions

---

# Working with Editor

## Notepad Behavior

### When Editor Opens:

- Semicolon replaced with forward slash (/)
- This is expected SQL\*Plus behavior
- Forward slash and semicolon have same effect

**Important Rule:** When editing in Notepad, you **must** use forward slash, not semicolon

---

# **Setting Default Editor**

## **DEFINE \_EDITOR Command**

### **Windows:**

```
define _editor=notepad
```

### **Linux:**

```
define _editor=vi
```

### **Function:**

- Controls which editor opens with EDIT command
  - Works in both Windows and Linux
  - Can use any available editor
-

# **Modifying Buffer Content**

## **Making Changes in Editor**

### **Process:**

1. Modify the SQL statement in editor
2. Change formatting
3. Add/modify conditions
4. Save the changes
5. Close editor

**Verification:** Use `list` command to verify changes saved to buffer

---

# **Running Buffer Contents**

## **Multiple Methods**

### **Method 1: Forward Slash**

/

### **Method 2: RUN Command**

run

or short form:

r

**Result:** Executes whatever statement is currently in the buffer

---

# **COLUMN Command**

## **Formatting Output**

**Purpose:** Control the format of displayed column values

### **Example:**

```
column first_name format a10
column salary format 999,999
```

### **Effects:**

- Sets FIRST\_NAME maximum width to 10 characters
  - Adds thousand separator to SALARY column
-

# **SAVE Command**

## **Saving Buffer to File**

### **Command:**

```
save filename
```

### **Example:**

```
save list-emps
```

### **Behavior:**

- Saves buffer contents to external file
  - Automatically adds .sql extension
  - Creates list-emps.sql file
-

# **HOST Command**

## **Running OS Commands from SQL\*Plus**

### **Command:**

```
host dir
```

or on Linux:

```
host ls
```

### **Function:**

- Execute operating system commands from within SQL\*Plus
- No need to exit SQL\*Plus
- Returns to SQL\*Plus after command completes

### **Example:**

```
host type filename.sql
```

Views contents of saved file

---

# **Running External Scripts**

## **@ Command**

### **Syntax:**

`@filename`

### **Example:**

`@list-emps`

### **Behavior:**

- Executes SQL script file
  - File extension optional
  - Reads and executes all statements in file sequentially
  - If one statement fails, continues to next statement
-

# **START Command**

## **Alternative to @ Symbol**

### **Syntax:**

```
start filename
```

### **Function:**

- Equivalent to @ symbol
- Runs SQL script files
- Same behavior as @ command

**Usage:** Personal preference whether to use @ or start

---

# SPOOL Command

## Saving Query Output

**Purpose:** Save the **output** of SQL statements to external files (not the statements themselves)

### Workflow:

1. Start spooling: `spool filename`
2. Execute SQL statements
3. Stop spooling: `spool off`

**Result:** All output between `spool` and `spool off` saved to file

---

# SPOOL Example

## Complete Process

### Commands:

```
spool employee_output
SELECT * FROM employees WHERE department_id = 90;
spool off
```

### Result:

- Creates file with query output
- File contains results, not SQL code
- Can edit file to remove unwanted characters

**Note:** SQL\*Plus commands don't need to end with semicolon

---

# **Running SQL\*Plus from Shell**

## **One-Line Execution**

### **Command:**

```
sqlplus hr/password @list-emps
```

### **What Happens:**

1. SQL\*Plus starts
2. Logs in as HR user
3. Executes the script
4. If script contains `quit`, SQL\*Plus exits automatically

**Use Case:** Perfect for automation and shell script integration

---

# **SET SQLPROMPT Command**

## **Customizing the Prompt**

### **Command:**

```
set sqlprompt "_user'@'_connect_identifier>"
```

### **Variables:**

- `_user` - Displays current username
- `_connect_identifier` - Displays Oracle SID

**Result:** Prompt shows current user and database, e.g.:

```
SYS@ORCL>  
HR@ORCL>
```

---

# **Essential SQL\*Plus Commands Summary**

## **Commands Covered**

<b>Command</b>	<b>Abbreviation</b>	<b>Purpose</b>
CONNECT	CONN	Connect to database
DESCRIBE	DESC	Show table structure
LIST	L	Display buffer contents
EDIT	ED	Open buffer in editor
RUN	R	Execute buffer
SAVE	-	Save buffer to file
HOST	!	Run OS commands
SPOOL	-	Save output to file
EXIT/QUIT	-	Exit SQL*Plus

---

# **Buffer Management Commands**

## **Working with SQL Buffer**

### **View Buffer:**

- `list` or `l`

### **Edit Buffer:**

- `edit` or `ed`

### **Run Buffer:**

- `/` (forward slash)
- `run` or `r`

### **Save Buffer:**

- `save filename`
-

# **Script Execution Commands**

## **Running Scripts**

### **From Command Line:**

```
sqlplus user/pass @script
```

### **From SQL\*Plus Prompt:**

```
@scriptname
```

or

```
start scriptname
```

**Note:** File extension (.sql) is always optional

---

## **Key Takeaways - SQL\*Plus**

### **Important Points to Remember**

1. **Most Common DBA Tool** - SQL\*Plus is still the most widely used
  2. **Command Abbreviations** - Nearly all commands have short forms
  3. **Semicolon Required** - SQL statements must end with ;
  4. **Case Sensitivity** - Commands not case sensitive, passwords are
  5. **Buffer Usage** - One statement stored at a time
  6. **OS Authentication** - Connect without password (with proper group membership)
  7. **Script Friendly** - Perfect for automation
-

# Practice Session Complete

## What You Learned

**Completed Tasks:** ✓ Invoked SQLPlus *multiple ways* ✓ *Connected to database with different methods* ✓ *Used OS authentication* ✓ *Explored SQLPlus buffer* ✓ Edited and executed SQL statements ✓ Saved and ran scripts ✓ Spooled query output ✓ Customized SQL\*Plus environment

## Next Steps

Next lecture: Using SQL Developer

---

# **Thank You**

**Continue Your Learning Journey**

**Remember:**

- Practice these commands regularly
- SQL\*Plus skills are fundamental for DBAs
- Master these basics before moving to advanced features

**Coming Up:** Hands-on with SQL Developer GUI tool

---