Unit 1: INTRODUCTION

Data:

Data are raw facts achieved through some observation or experiments and doesn't provide information until we process it.

Information:

Information are the outcomes of processed data that provides some knowledge about something of real time that it relates to.

Database:

It is a collection of data related to particular subjects or purpose. A database organizes data in easily accessible manner. A user can retrieve information from database in effective and efficient manner.

Database Management System (DBMS):

DBMS is a collection of interrelated data and set of programs to access those data. Its primary goal is to provide a way to store and retrieve database information in both efficient and convenient way.

Data Administrator and Database Administrator:

Data Administrator	Database Administrator
DAs are concerned with the data needs and data flows throughout the entire organization.	DBAs are concerned with storage and management of the information in the database.
The DA is involved more in the requirements gathering, analysis, and design phases	DBA in the design, development, testing, and operational phases.
DA determines long-term goals and enforces standards, policies and procedures.	DBA executes plans to achieve goals. He enforces standards, policies and procedures developed by DA.
DA also determines data requirements and develops conceptual and logical design.	DBA develops logical and physical design.
Work of DA is DBMS independent.	Work of DBA is DBMS dependent.

DBMS Architecture

Three Level Database Architecture:
In database, data are stored as bits, or numbers and strings. But it is difficult to work with data at this level. So, it is necessary to view data at different levels of abstraction.
DBMS architecture can be classified as three level schema architecture as:

- Internal/Physical level
- Conceptual level
- External level

Physical level

This level is the closest to the physical storage and is concerned with the way data is stored inside the system. It also provides a low-level description of the physical database. This level defines the record types and methods of storage as well as how stored files are represented, what physical sequence the stored records are in, and what other physical structure exists.

Conceptual Level:

Conceptual level describes the structure of the whole database for a group of users. It is also called as the data model. Conceptual schema is a representation of the entire content of the database. These schemas contains all the information to build relevant external records. It hides the internal details of physical storage.

External Level:

This is the highest level which is closest to the user. This level includes a number of user views or external schemas. External level is related to the data which is viewed by individual end users. External view describes the segment of the database that is required for a particular user group and hides the rest of the database from that user group.

DATA INDEPENDENCE

The ability to modify schema definition in one level without affecting schema definition in next higher level is called **data independence.**

There are two types of data independence:

- Logical Data Independence
- Physical Data Independence

Logical Data Independence:

It refers to the immunity of external models to the changes in the logical model. Changes to conceptual schema such as addition or removal of new entities, attributes or relationships should be possible without any changes to existing external schema. It occurs at user interface level.

Physical Data Independence:

It refers to the immunity of conceptual schema to the changes in physical/internal schema. Changes to internal schema such as using different file organization or storage structure, using different storage devices, modifying indexes should be possible without having changes to conceptual or external schema. It occurs at logical interface level.

Modifications at physical level are occasionally necessary to improve performance and so we can change physical level without affecting conceptual or external view of data.

Modifications at logical level are necessary whenever logical structure of database is altered. Example of Logical Data Independence: if we add/remove some new columns from able then user view and program should not change.

Logical independence is more difficult to achieve than physical independence, since application program are highly dependent on the logical structure of data that they access.

DBA ROLES AND RESPONSIBILITIES

Database Administrator is an Information Technology professional who creates actual data and put in place technical controls needed to enforce the various policies, decisions made by the data administrator. Some of the roles and responsibilities of DBA are:

- Installing and upgrading the Oracle server and application tools
- Allocating system storage and planning future storage requirements for the database system
- Creating primary database storage structures (tablespaces) after application developers have designed an application
- Creating primary objects (tables, views, indexes) once application developers have designed an application
- Modifying the database structure, as necessary, from information given by application developers
- Enrolling users and maintaining system security
- Ensuring compliance with your Oracle license agreement
- Controlling and monitoring user access to the database
- Monitoring and optimizing the performance of the database
- Planning for backup and recovery of database information
- Maintaining archived data on tape
- Backing up and restoring the database
- Contacting Oracle Corporation for technical support

Tasks of a Database Administrator

The following tasks present a prioritized approach for designing, implementing, and maintaining an Oracle Database:

Task 1: Evaluate the Database Server Hardware
Task 2: Install the Oracle Software
Task 3: Plan the Database
Task 4: Create and Open the Database Task 5: Back Up the Database
Task 6: Enroll System Users
Task 7: Implement the Database Design
Task 8: Back Up the Fully Functional Database
Task 9: Tune Database Performance

SQL *PLUS Overview

SQL*Plus is an interactive and batch query tool that is installed with every Oracle Database Server or Client installation. It has a command-line user interface, a Windows Graphical User Interface (GUI) and the *i*SQL*Plus web-based user interface.

SQL*Plus has its own commands and environment, and it provides access to the Oracle Database. It enables you to enter and execute SQL, PL/SQL, SQL*Plus and operating system commands to perform the following:

- Format, perform calculations on, store, and print from query results
- Examine table and object definitions
- Develop and run batch scripts
- Perform database administration

You can use SQL*Plus to generate reports interactively, to generate reports as batch processes, and to output the results to text file, to screen, or to HTML file for browsing on the Internet. You can generate reports dynamically using the HTML output facility of SQL*Plus, or using the dynamic reporting capability of *i*SQL*Plus to run a script from a web page.

SQL*Plus Command-line and Windows GUI Architecture

SQL*Plus command-line and the Windows GUI use a two-tier model comprising:

- Client (command-line user interface).
- Database (Oracle Database).

The two tiers may or may not be on the same machine.

SQL*Plus Client

The command-line user interface is the character based terminal implementation. The Windows GUI is an alternate user interface available in Windows installations.

Oracle Database

Oracle Database Net components provide communication between the SQL*Plus Client and Oracle Database.

PRODUCING MORE READABLE OUTPUT

Often, results returned from SQL *Plus wrap to the next line or do not have the proper formatting. You can use simple SQL *Plus formatting commands to produce more readable output and better-looking reports.

Setting Page and Line sizes:

Use SHOW command to find the values of the PAGESIZE and LINESIZE environment variables.
SQL> SHOW PAGESIZE LINESIZE
Now, adjust these setting as follows: SQL> SET PAGESIZE 55 LINESIZE 54

Formatting Columns:

Use COLUMN command to format the column headings and display column data. To display a different heading for the EMP_NAME column, use syntax:
SQL>COLUMN oldname HEADING "newname" Use FORMAT to change the column display width.
SQL> COLUMN emp_name HEADING "Employee Name" FORMAT A20 To display salary in money format:
SQL> COLUMN sal FORMAT "$9,999.99"
To wrap text : SQL> COLUMN comments HEADING "Comments" WORD_WRAPPED
For justification: SQL> JUSTIFY RIGHT FORMAT A30

Suppressing Duplicate Values:

You can suppress display of duplicate column values using BREAK ON column name command. The BREAK command has options to skip lines, pages and so on along with NODUPLICATE option.
Eg: if you want to suppress the display of duplicate JOB_ID values use: SQL> BREAK ON job_id SKIP 2 NODUPLICATES

ACCEPTING VALUES AT RUNTIME

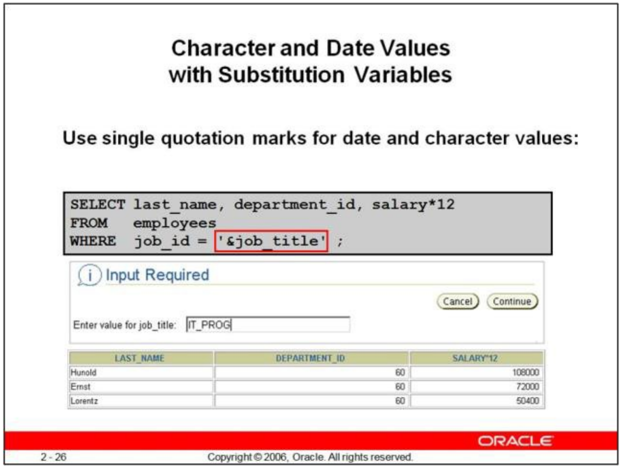
Single-Ampersand Substitution Variable

When running a report, users often want to restrict the data that is returned dynamically. iSQL*Plus provides this flexibility with user variables. Use an ampersand (&) to identify each variable in your SQL statement. You do not need to define the value of each variable.
The example in the slide creates an iSQL*Plus substitution variable for an employee number. When the statement is executed, iSQL*Plus prompts the user for an employee number and then displays the employee number, last name, salary, and department number for that employee.
With the single ampersand, the user is prompted every time the command is executed, if the variable does not exist.

Double-Ampersand Substitution Variable

You can use the double-ampersand (&&) substitution variable if you want to reuse the variable value without prompting the user each time. The user sees the prompt for the value only once. In the example in the slide, the user is asked to give the value for variable column name only once. The value that is supplied by the user (department_id) is used for both display and ordering of data.

iSQL*Plus stores the value that is supplied by using the DEFINE command; it uses it again whenever you reference the variable name



Character and Date Values with Substitution Variables

Use single quotation marks for date and character values:

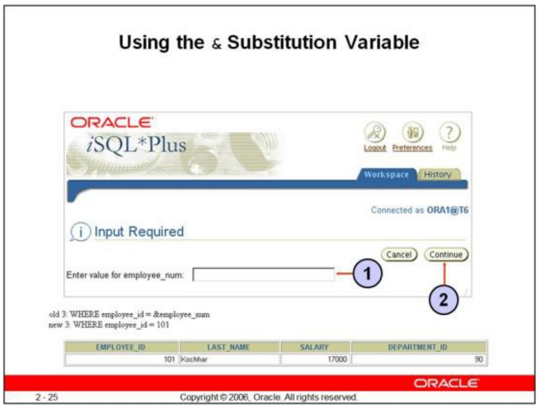
```
SELECT last_name, department_id, salary*12
FROM   employees
WHERE  job_id = '&job_title' ;
```

(i) Input Required

Enter value for job_title: IT_PROG

LAST_NAME	DEPARTMENT_ID	SALARY*12
Hunold	60	108000
Ernst	60	72000
Lorentz	60	50400

2 - 26



Using the & Substitution Variable

Connected as ORA1@T6

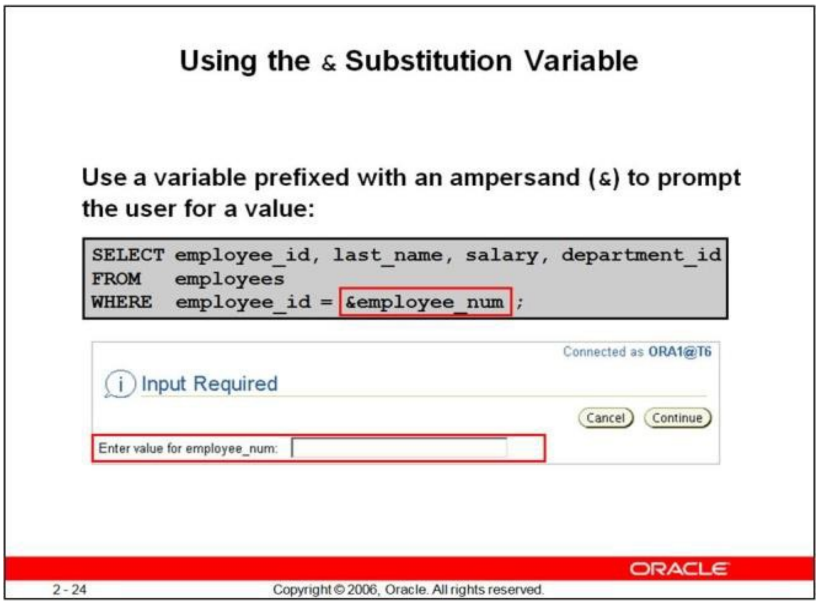
(i) Input Required

Enter value for employee_num:

```
old 3: WHERE employee_id = &employee_num
new 3: WHERE employee_id = 101
```

EMPLOYEE_ID	LAST_NAME	SALARY	DEPARTMENT_ID
101	Kochhar	17000	90

2 - 25



Using the & Substitution Variable

Use a variable prefixed with an ampersand (&) to prompt the user for a value:

```
SELECT employee_id, last_name, salary, department_id
FROM   employees
WHERE  employee_id = &employee_num ;
```

Connected as ORA1@T6

(i) Input Required

Enter value for employee_num:

2 - 24

USING iSQL *PLUS

*i*SQL*Plus

*i*SQL*Plus is an environment in which you can do the following:

- Execute SQL statements to retrieve, modify, add, and remove data from the database.
- Format, perform calculations on, store, and print query results in the form of reports.
- Create script files to store SQL statements for repeated use in the future

To log in from a browser environment:

- Start the browser.
- Enter the URL address of the iSQL*Plus environment.
- On the Login page, enter appropriate values in the Username and Password

iSQL*Plus Environment

In the browser, the iSQL*Plus Workspace page has several key areas:

- **Text box**: Area where you type the SQL statements and iSQL*Plus commands
- **Execute button**: Click to execute the statements and commands in the text box
- **Load Script button**: Brings up a form where you can identify a path and file name or a URL that contains SQL, PL/SQL, or SQL*Plus commands and load them into the text box
- **Save Script button:** Saves the contents of the text box to a file
- **Cancel button**: Stops the execution of the command in the text box
- **Clear Screen button**: Click to clear text from the text box
- **Logout icon**: Click to end the iSQL*Plus session and return to the iSQL*Plus Login page
- **Preferences icon**: Click to change your interface configuration, system configuration, or password
- **Help icon**: Provides access to iSQL*Plus help documentation