

DATABASE SYSTEMS LAB

Course Code	: 30102422
--------------------	-------------------

Credit Hours	: 1
---------------------	------------

Prerequisite	: 30102421
---------------------	-------------------



Instructor Information

Name	:Dr. Jihad Nader				
Office No.	B17 F4 R8				
Tel (Ext)	N/A				
E-mail	jihadnader@bau.edu.jo				
Office Hours	Thur ,Tues (10-11) mon, wed (1-2)				
Class Times	Building	Day	Start Time	End Time	Room No.
	-	Monday	8	11	مختبر المعالجات



Course Description:

This Lab. practices the concepts introduced in the Database systems course using Oracle Database. The students are expected to implement a database project for some problem.

Course Title: Database Systems Lab

Credit Hour(1)

[Pre-req. Course Code(30102421)]

Textbook: *Oracle Database 10g: SQL Fundamentals I, Volume I • Student Guide*

**Oracle Database 10g: SQL
Fundamentals I**

Volume I • Student Guide

D17108GC11
Edition 1.1
August 2004
D39786

ORACLE®



COURSE OBJECTIVES:

Upon completion of this course, students will have gained knowledge of the DBMS (Oracle) concepts and the ability to:

- Understand the concepts of relational databases and the Oracle Database 10g database technology.
- Use the powerful SQL programming language and its features.
- Identify features of Relational Database Management System (RDBMS).
- Categorize the main database objects
- Understand how constraints are created at the time of table creation
- Describe each data manipulation language (DML) statement
- List the capabilities of SQL SELECT statements
- Write SELECT statements to access data from more than one table using equijoins and non-equijoins
- Employ SQL functions to generate and retrieve customized data
- Identify when a subquery can help solve a question
- Write subqueries when a query is based on unknown values
- Use a set operator to combine multiple queries into a single query

COURSE SYLLABUS

Week	Course Topic	Notes
Week 1	Creating and Managing Tables: <ul style="list-style-type: none">- Database Objects- Naming Conventions- The Create Table Statement- Creating a Table by Using a Subquery- Querying the Data Dictionary- The Alter Table Statement- Truncating a Table- Adding Comments to a Table	
Week 2	Including Constraints <ul style="list-style-type: none">- Defining Constraints<ul style="list-style-type: none">o The Not Null Constrainto The Unique Constrainto The Primary Key Constrainto The Foreign Key Constrainto The Check Constraint- Adding a Constraint- Dropping a Constraint- Enabling and Disabling Constraints- Viewing Constraints	
Week 3	Manipulating Data <ul style="list-style-type: none">- Data Manipulating Language.- The Insert Statement- Copying Rows from another Table- The Update Statement- The Delete Statement- Database Transactions- Commit and Rollback Statements	
Week 4	Writing Basic SQL Statements <ul style="list-style-type: none">- Selecting Specific Columns- Arithmetic Expressions- Concatenation Operator- Using Column Aliases- Eliminating Duplicate Rows	

COURSE SYLLABUS

Week	Course Topic	Notes
Week 5	Restricting and Sorting Data <ul style="list-style-type: none">- Where Clause- Comparison Operators- Special Operators- Logical Operator (And, Or, Not)- Order By Clause	
Week 6	Displaying Data from Multiple Tables <ul style="list-style-type: none">- Cartesian Product.- Types of Joins- Table Aliases.	
Week 7	Single-Row Functions <ul style="list-style-type: none">- Character Functions.- Number Functions- Date Functions	
Week 8	Midterm Exam	Midterm Exam
Week 9	Project Proposal	
Week 10	Single-Row Functions <ul style="list-style-type: none">- Conversion Functions- General Functions	

COURSE SYLLABUS

Week	Course Topic	Notes
Week 11	Aggregating Data using Group Functions <ul style="list-style-type: none">- Types of Group Functions (AVG, SUM, MAX, MIN, COUNT).- Creating Groups of data: Group By Clause.- Excluding Group Results: Having Clause.- Nested Group Functions	
Week 12	Subqueries <ul style="list-style-type: none">- Types of Subqueries<ul style="list-style-type: none">▪ Single-Row Subqueries▪ Multiple-Row Subqueries	
Week 13	Multiple-Column Subqueries <ul style="list-style-type: none">- Column Comparisons- Null Values in a subquery- Using a subquery in the From Clause	
Week 14	Using the Set Operators <ul style="list-style-type: none">- Union / Union All- Intersect- Minus	
Week 15	Project Discussion	
Week 16	Final Exam	Final Exam

Week 4

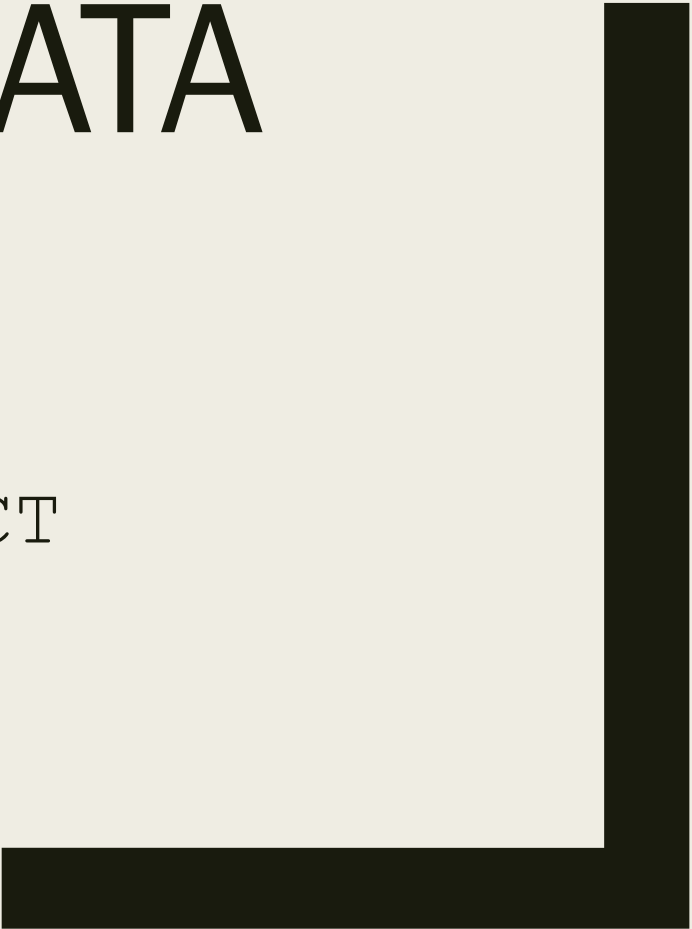


Chapter 3:

Basic SQL Statements



RETRIEVING DATA USING THE SQL SELECT STATEMENT



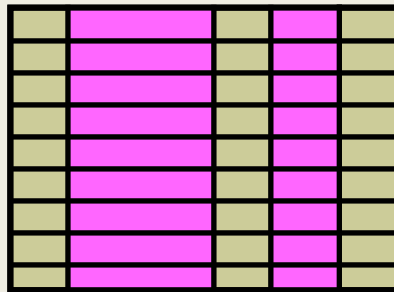
Objectives

After completing this lesson, you should be able to do the following:

- *List the capabilities of SQL `SELECT` statements*
- *Execute a basic `SELECT` statement*

Capabilities of SQL SELECT Statements

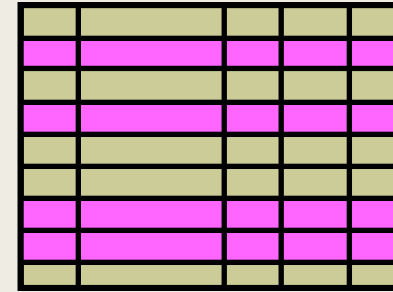
Projection



A 10x5 grid representing a table. The second, third, fourth, fifth, sixth, seventh, and eighth columns are highlighted in pink, illustrating the result of a projection operation that selects specific columns from the original table.

Table 1

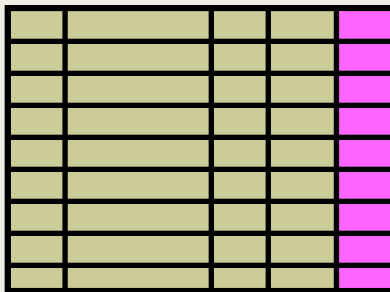
Selection



A 10x5 grid representing a table. The first, third, fourth, sixth, seventh, eighth, and ninth rows are highlighted in pink, illustrating the result of a selection operation that filters specific rows from the original table.

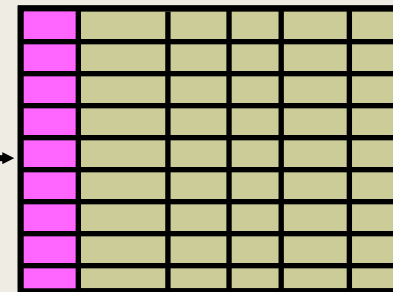
Table 1

Join



A 10x5 grid representing a table. The entire grid is highlighted in pink, representing the result of a join operation between two tables.

Table 1



A 10x5 grid representing a table. The first column is highlighted in pink, representing the result of a join operation between two tables.

Table 2

Basic SELECT Statement

```
SELECT * | { [DISTINCT] column | expression [alias], ... }  
FROM      table;
```

- *SELECT* identifies the columns to be displayed.
- *FROM* identifies the table containing those columns.

Selecting All Columns

```
SELECT *  
FROM departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

8 rows selected.

Selecting Specific Columns

```
SELECT department_id, location_id  
FROM departments;
```

DEPARTMENT_ID	LOCATION_ID
10	1700
20	1800
50	1500
60	1400
80	2500
90	1700
110	1700
190	1700

8 rows selected.

Writing SQL Statements

- *SQL statements are not case sensitive.*
- *SQL statements can be on one or more lines.*
- *Keywords cannot be abbreviated or split across lines.*
- *Clauses are usually placed on separate lines.*
- *Indents are used to enhance readability.*

Arithmetic Expressions

Create expressions with number and date data by using arithmetic operators.

Operator	Description
+	Add
-	Subtract
*	Multiply
/	Divide

Using Arithmetic Operators

```
SELECT last_name, salary, salary + 300  
FROM   employees;
```

LAST_NAME	SALARY	SALARY+300
King	24000	24300
Kochhar	17000	17300
De Haan	17000	17300
Hunold	9000	9300
Ernst	6000	6300

■ ■ ■

20 rows selected.

Operator Precedence

```
SELECT last_name, salary, 12*salary+100
FROM employees;
```

1

LAST_NAME	SALARY	12*SALARY+100
King	24000	288100
Kochhar	17000	204100
De Haan	17000	204100

■ ■ ■

20 rows selected.

```
SELECT last_name, salary, 12*(salary+100)
FROM employees;
```

2

LAST_NAME	SALARY	12*(SALARY+100)
King	24000	289200
Kochhar	17000	205200
De Haan	17000	205200

■ ■ ■

20 rows selected.

Defining a Null Value

- *A null is a value that is unavailable, unassigned, unknown, or inapplicable.*
- *A null is not the same as a zero or a blank space.*

```
SELECT last_name, job_id, salary, commission_pct  
FROM employees;
```

LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT
King	AD_PRES	24000	
Kochhar	AD_VP	17000	
...			
Zlotkey	SA_MAN	10500	.2
Abel	SA_REP	11000	.3
Taylor	SA_REP	8600	.2
...			
Gietz	AC_ACCOUNT	8300	

20 rows selected.

Null Values in Arithmetic Expressions

Arithmetic expressions containing a null value evaluate to null.

```
SELECT last_name, 12*salary*commission_pct  
FROM employees;
```

Kochhar	
King	
LAST_NAME	12*SALARY*COMMISSION_PCT
...	
Zlotkey	25200
Abel	39600
Taylor	20640
...	
Gietz	

20 rows selected.

Defining a Column Alias

A column alias:

- *Renames a column heading*
- *Is useful with calculations*
- *Immediately follows the column name (There can also be the optional `AS` keyword between the column name and alias.)*
- *Requires double quotation marks if it contains spaces or special characters or if it is case sensitive*

Using Column Aliases

```
SELECT last_name AS name, commission_pct comm
FROM employees;
```

NAME	COMM
King	
Kochhar	
De Haan	

...

20 rows selected.

```
SELECT last_name "Name" , salary*12 "Annual Salary"
FROM employees;
```

Name	Annual Salary
King	288000
Kochhar	204000
De Haan	204000

...

20 rows selected.

Concatenation Operator

A concatenation operator:

- *Links columns or character strings to other columns*
- *Is represented by two vertical bars (||)*
- *Creates a resultant column that is a character expression*

```
SELECT    last_name||job_id AS "Employees"  
FROM      employees;
```

Employees
KingAD_PRES
KochharAD_VP
De HaanAD_VP
...

20 rows selected.

Literal Character Strings

- *A literal is a character, a number, or a date that is included in the `SELECT` statement.*
- *Date and character literal values must be enclosed by single quotation marks.*
- *Each character string is output once for each row returned.*

Using Literal Character Strings

```
SELECT last_name || ' is a ' || job_id  
       AS "Employee Details"  
FROM   employees;
```

Employee Details
King is a AD_PRES
Kochhar is a AD_VP
De Haan is a AD_VP
Hunold is a IT_PROG
Ernst is a IT_PROG
Lorentz is a IT_PROG
Mourgos is a ST_MAN
Rajs is a ST_CLERK

...

20 rows selected.

Alternative Quote (q) Operator

- *Specify your own quotation mark delimiter*
- *Choose any delimiter*
- *Increase readability and usability*

```
SELECT department_name ||  
       q'[, it's assigned Manager Id: ]'  
       || manager_id  
       AS "Department and Manager"  
FROM departments;
```

Department and Manager

Administration, it's assigned manager ID: 200

Marketing, it's assigned manager ID: 201

Shipping, it's assigned manager ID: 124

...

8 rows selected.

Duplicate Rows

The default display of queries is all rows, including duplicate rows.

```
SELECT department_id  
FROM employees;
```

1

DEPARTMENT_ID	
	90
	90
	90

...

20 rows selected.

```
SELECT DISTINCT department_id  
FROM employees;
```

2

DEPARTMENT_ID	
	10
	20
	50

...

8 rows selected.

Displaying Table Structure

Use the `DESCRIBE` command to display the structure of a table:

```
DESC[RIBE] tablename
```

Displaying Table Structure

```
DESCRIBE employees
```

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

Summary

In this lesson, you should have learned how to:

- Write a *SELECT* statement that:
 - Returns all rows and columns from a table
 - Returns specified columns from a table
 - Uses column aliases to display more descriptive column headings

```
SELECT *|{[DISTINCT] column/expression [alias],...}  
FROM table;
```