National University of Computer & Emerging Sciences, Karachi
Computer Science Department
Fall 2024, Lab Manual - 01

Course Code: CL-2005	**Course: Database Systems Lab**
Instructor(s):	**Mubashir**

Contents:

1. Database
2. SQL
3. Basic SQL Concepts

Database

A database is a systematic collection of data. They support electronic storage and manipulation of data. Databases make data management easy.

Example #1

An online telephone directory uses a database to store data of people, phone numbers, and other contact details. Your electricity service provider uses a database to manage billing, client-related issues, handle fault data, etc.

Example #2

Facebook needs to store, manipulate, and present data related to members, their friends, member activities, messages, advertisements, and a lot more. We can provide a countless number of examples for the usage of databases.

SQL

SQL is the standard language for dealing with Relational Databases. SQL can be used to insert, search, update, and delete database records. SQL can do lots of other operations, including optimizing and maintenance of databases. SQL stands for Structured Query language, pronounced as "S-Q-L" or sometimes as "See-Quel"... Relational databases like MySQL Database, Oracle, MS SQL Server, Sybase, etc. use ANSI SQL.

Basic SQL Concepts

I. Data Types

bigint	decimal	real	char	nvarchar
int	numeric	datetime	varchar	nvarchar(max)
smallint	money	smalldatetime	varchar(max)	ntext
tinyint	smallmoney	date	text	binary
bit	float	time	nchar	varbinary
varbinary(max)	image			

II. Arithmetic operators

Addition	Subtraction	Multiplication	Division	Modulus
+	**-**	*****	**/**	**%**

III. SQL Comparison Operators

=	Checks if the values of two operands are equal or not, if yes then condition becomes true.
!=	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.
<>	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.
!<	Checks if the value of left operand is not less than the value of right operand, if yes then condition becomes true.
!>	Checks if the value of left operand is not greater than the value of right operand, if yes then condition becomes true.

IV. SQL Logical Operators

ALL	The ALL operator is used to compare a value to all values in another value set.
AND	The AND operator allows the existence of multiple conditions in an SQL statement's WHERE clause
ANY	The ANY operator is used to compare a value to any applicable value in the list as per the condition.
BETWEEN	The BETWEEN operator is used to search for values that are within a set of values, given the minimum value and the maximum value.
EXISTS	The EXISTS operator is used to search for the presence of a row in a specified table that meets a certain criterion.
IN	The IN operator is used to compare a value to a list of literal values that have been specified.
LIKE	The LIKE operator is used to compare a value to similar values using wildcard operators.
NOT	The NOT operator reverses the meaning of the logical operator with which it is used. Eg: NOT EXISTS, NOT BETWEEN, NOT IN, etc. **This is a negate operator.**
OR	The OR operator is used to combine multiple conditions in an SQL statement's WHERE clause.
NULL	The NULL operator is used to compare a value with a NULL value.
UNIQUE	The UNIQUE operator searches every row of a specified table for uniqueness (no duplicates).

V. Basic SQL Queries

Note: Connect the HR Database in SqlDeveloper

Select * from EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
100	Steven	King	SKING	515.123.4567	17-Jun-03	AD_PRES
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-Sep-05	AD_VP
102	Lex	De Haan	LDEHAAN	515.123.4569	13-Jan-01	AD_VP
--------	--------	--------	--------	--------	--------	--------
1023	Lex3	De Haanas	LDEsdaHAAN	515.123.4569	13-Jan-01	AD_VPP

Select EMPLOYEE_ID, FIRST_NAME, SALARY from EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	SALARY
100	Steven	24000
101	Neena	17000
102	Lex	17000
----------	----------	----------
1023	Lex3	12000

Select EMPLOYEE_ID, FIRST_NAME, SALARY from EMPLOYEES where salary>2300

EMPLOYEE_ID	FIRST_NAME	SALARY
100	Steven	24000

Select EMPLOYEE_ID, FIRST_NAME, SALARY from EMPLOYEES where salary greater than or equal to 10000 and less than or equal to 12000

EMPLOYEE_ID	FIRST_NAME	SALARY
114	Den	11000
147	Alberto	12000
148	Gerald	11000
149	Eleni	10500
114	Den	11000

Let's Practice More!!

Comparison operator:

- SELECT * FROM EMPLOYEES WHERE MANAGER_ID = 205;
- SELECT * FROM EMPLOYEES WHERE EMPLOYEE_ID < 150;
- SELECT * FROM EMPLOYEES WHERE MANAGER_ID > 100;
- SELECT * FROM EMPLOYEES WHERE EMPLOYEE_ID >= 108;
- SELECT * FROM EMPLOYEES WHERE MANAGER_ID <= 150;
- ELECT * FROM EMPLOYEES WHERE EMPLOYEE_ID <> 115;

Logical Operators:

- SELECT FIRST_NAME,SALARY,JOB_ID,DEPARTMENT_ID FROM EMPLOYEES WHERE JOB_ID = 'IT_PROG' AND DEPARTMENT_ID = 60;
- SELECT FIRST_NAME,SALARY,JOB_ID,DEPARTMENT_ID FROM EMPLOYEES WHERE JOB_ID = 'AD_VP' OR DEPARTMENT_ID = 90;
- SELECT FIRST_NAME,SALARY,JOB_ID,DEPARTMENT_ID FROM EMPLOYEES WHERE Not JOB_ID = 'ST_MAN';

LAB TASKS:

1. Write a query to find the list of jobs whose max salary is less than 15000 and greater than 6000 from **jobs** table.
2. Write a query to find list of first name with job_id 'IT_PROG' but not with DEPARTMENT_ID '104' from **Employees** table.
3. Write a query to find the list of employee's first_name and last_name who hired before '07-FEB-07' from **Employees** table.
4. Write a query to find the list of employees who are hired with salary '1700' and with DEPARTMENT_ID=90 from **Employees** table.
5. Write a query to find the list of employees who are hired after '21-SEP-05' but not in Department with DEPARTMENT_ID=102 from employee table.
6. Write a query to find the list of employees whose COMMISSION_PCT is '0.1' and they do not belong to DEPARTMENT_ID 90 or 100 from **Employees** table
7. Write a query to find the employees having salaries between 3000 and 30000 from **Employees** table.
8. Write a query to find the Job_title whose min salary is greater than 9000 and less than 20,000 from **jobs** table.
9. Write a query to find employee whose ID are greater than 110 and less than 180 and their department_id is greater than 85 and less than 110 along with their F_name, Salary, phone number & Job ID from **Employees** table.
10. Write a query to find the list of street_address and postal code whose location IDs are greater than 2300 but not the country ID 'UK'.