



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



<b>Course Code: CL-2005</b>	<b>Course: Database Systems Lab</b>
<b>Instructor(s):</b>	<b>Mehak Mazhar</b>

**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**

<b>ALL</b>	The ALL operator is used to compare a value to all values in another value set.
<b>AND</b>	The AND operator allows the existence of multiple conditions in an SQL statement's WHERE clause
<b>ANY</b>	The ANY operator is used to compare a value to any applicable value in the list as per the condition.
<b>BETWEEN</b>	The BETWEEN operator is used to search for values that are within a set of values, given the minimum value and the maximum value.
<b>EXISTS</b>	The EXISTS operator is used to search for the presence of a row in a specified table that meets a certain criterion.
<b>IN</b>	The IN operator is used to compare a value to a list of literal values that have been specified.
<b>LIKE</b>	The LIKE operator is used to compare a value to similar values using wildcard operators.
<b>NOT</b>	The NOT operator reverses the meaning of the logical operator with which it is used. Eg: NOT EXISTS, NOT BETWEEN, NOT IN, etc. <b>This is a negate operator.</b>
<b>OR</b>	The OR operator is used to combine multiple conditions in an SQL statement's WHERE clause.
<b>NULL</b>	The NULL operator is used to compare a value with a NULL value.
<b>UNIQUE</b>	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	LDEsdaHAA N	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 DEPARTMENT\_ID = 60;
- SELECT \* FROM EMPLOYEES WHERE MANAGER\_ID < 110;
- SELECT \* FROM EMPLOYEES WHERE MANAGER\_ID > 200;
- SELECT \* FROM EMPLOYEES WHERE MANAGER\_ID >= 200;
- SELECT \* FROM EMPLOYEES WHERE MANAGER\_ID <= 150;
- ELECT \* FROM EMPLOYEES WHERE MANAGER\_ID <> 114;

#### **Logical Operators:**

- SELECT FIRST\_NAME,SALARY,JOB\_ID,DEPARTMENT\_ID FROM EMPLOYEES WHERE JOB\_ID = 'AD\_VP' AND DEPARTMENT\_ID = 90;
- 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 = 'AD\_VP';