

National University of Computer & Emerging Sciences, Karachi



Computer Science Department

Fall 2023, Lab Manual - 01



Course Code: CL-2005	Course : Database Systems Lab
Instructor(s) :	Abeer Gauher

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
+	-	*	/	%

SQL Numeric Data Types

Datatype	From	To
bit	0	1
tinyint	0	255
smallint	-32,768	32,767
int	-2,147,483,648	2,147,483,647
bigint	-9,223,372,036, 854,775,808	9,223,372,036, 854,775,807
decimal	-10 ³⁸ +1	10 ³⁸ -1
numeric	-10 ³⁸ +1	10 ³⁸ -1
float	-1.79E + 308	1.79E + 308
real	-3.40E + 38	3.40E + 38

SQL Date and Time Data Types

Datatype	Description
DATE	Stores date in the format YYYY-MM-DD
TIME	Stores time in the format HH:MI:SS
DATETIME	Stores date and time information in the format YYYY-MM-DD HH:MI:SS
TIMESTAMP	Stores number of seconds passed since the Unix epoch ('1970-01-01 00:00:00' UTC)
YEAR	Stores year in 2 digits or 4 digit format. Range 1901 to 2155 in 4-digit format. Range 70 to 69, representing 1970 to 2069.

SQL Character and String Data Types

Datatype	Description
CHAR	Fixed length with a maximum length of 8,000 characters
VARCHAR	Variable-length storage with a maximum length of 8,000 characters
VARCHAR(max)	Variable-length storage with provided max characters, not supported in MySQL
TEXT	Variable-length storage with maximum size of 2GB data

Note that all the above data types are for character stream, they should not be used with Unicode data.

SQL Unicode Character and String Data Types

Datatype	Description
NCHAR	Fixed length with maximum length of 4,000 characters
NVARCHAR	Variable-length storage with a maximum length of 4,000 characters
NVARCHAR(max)	Variable-length storage with provided max characters
NTEXT	Variable-length storage with a maximum size of 1GB data

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.

IV. SQL Logical Operators

AND	The AND operator allows the existence of multiple conditions in an SQL statement's WHERE clause
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).

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

More Examples

Comparison operator:

- SELECT * FROM EMPLOYEES WHERE MANAGER_ID = 101;
- 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;
- SELECT * 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';

LAB#01 TASKS:

1. Write a SQL statement to display all the information of table Jobs.
2. Write a SQL query to find those employees whose Salary is equal to 24000 from Employees table.
3. Write a SQL query to find the Jobs where salaries are higher than or equal to \$15000 from Employees table.
4. Write a SQL query to find the details of employees whose last name is 'Taylor'. Return emp_id, fname, lname, and emp_dept id.
5. Write a SQL query to find the details of the employees who work in the department 50. Return emp_idno, emp_fname, emp_lname and emp_dept id.
6. Write a query to find the phone_number of the DEPARTMENT_ID=80 and MANAGER_ID=100 of Employees table. Return emp_idno, emp_fname, emp_lname emp_dept id, manager_id.
7. Write a SQL query to find the Employees with the First name 'John' 'Neena' and 'Peter'.
8. Write a query to find the list of cities with country ID 'IT' from locations table.
9. Write a query to find the list of city except country ID 'IN' and 'CH' from locations table.
10. Write a query to find the list of phone numbers with DEPARTMENT_ID '90' but not with job_id 'IT_PROG' from Employees table. Return phone_number, department_id, job_id.
11. Write a query to find the list of employees who are hired after '12-Dec-07' from employee table.
12. Write a query to find the list of employees who are hired after '12-Dec-07' in Department with DEPARTMENT_ID=80 from employee table.