

Q1. Create a database named 'Employee' and create a table 'Emp' with attributes 'ename', 'ecity', 'enumber', 'eaddress', 'deptname'.

A.

CREATE DATABASE EMPLOYEE;

USE DATABASE EMPLOYEE;

CREATE TABLE EMP(ENAME VARCHAR2(25) NOT NULL, ECITY VARCHAR2(25) NOT NULL, ENUMBER INTEGER(10) NOT NULL, EADDRESS VARCHAR2 (200) NOT NULL, DEPTNAME VARCHAR2(30) NOT NULL);

Q2. Create a table 'department' with attributes 'deptt_no', 'dcity', 'head', 'contact'.

A.

CREATE TABLE DPEARTMENT(DEPTT_ID INTEGER(6) NOT NULL, DCITY VARCHAR2(25) NOT NULL, HEAD VARCHAR2(25) NOT NULL, CONTACT INTEGER(10) NOT NULL);

Q3. Insert Data into tables using 'Insert' Command.

A.

INSERT INTO EMP(ENAME, ECITY, ENUMBER, EADDRESS, DEPTNAME) VALUES ("RAM SINGH", "AMRITSAR", 9876543210, "352, 3RD FLOOR MAYFAIR APPARTMENTS, G.T. ROAD, CHHEHARTA, AMRITSAR", "COMPUTER SCIENCE");

INSERT INTO DEPARTMENT(DEPTT_ID, DCITY, HEAD, CONTACT) VALUES(38, "AMRITSAR", "DR. HARDEEP SINGH", 9876543210);

Q4. To study command to modify the structure of table using Alter.

A.

ALTER TABLE DEPARTMENT ALTER COLUMN CONTACT INTEGER(12);

ALTER TABLE EMP ADD ESALARY INTEGER(6);

ALTER TABLE EMP ADD COLUMN EMPID INTEGER(6);

ALTER TABLE EMP DROP COLUMN EMPID;

Q5. To demonstrate various DML commands over above example.

A.

DML Commands: **SELECT, INSERT UPDATE DELETE.** Usage is as follows:

SELECT * FROM EMP;

SELECT HEAD, DCITY FROM DEPARTMENT;

INSERT INTO DEPARTMENT VALUES (34, "AMRITSAR", "DR. SANDEEP SHARMA", 9876543210);

UPDATE TABLE EMP SET ECITY="JALANDHAR" WHERE ENAME="RAM SINGH";

DELETE FROM DEPARTMENT WHERE DEPTT_ID=34;

Q6. Write and execute various queries to study the select command.

A.

SELECT * FROM DEPARTMENT;

SELECT ENAME, ECITY FROM EMP;

SELECT * FROM EMP WHERE ECITY="AMRITSAR";

Q7. Execute queries that execute commnads that involve compound conditions (and, or, in between, not between, like, not like etc.)

A.

SELECT ENAME, EADDRESS FROM EMP WHERE ECITY="AMRITSAR" AND ENAME="RAM SINGH";

SELECT ENAME, EADDRESS FROM EMP WHERE ECITY="AMRITSAR" OR ENAME="RAM SINGH";

SELECT ENAME, EADDRESS FROM EMP WHERE NOT ECITY="AMRITSAR" ;

SELECT * FROM DEPARTMENT WHERE DEPTT_ID BETWEEN 25 AND 66;

SELECT * FROM DEPARTMENT WHERE DEPTT_ID NOT BETWEEN 25 AND 66;

SELECT HEAD FROM DEPARTMENT WHERE HEAD LIKE "D%";

SELECT HEAD FROM DEPARTMENT WHERE HEAD NOT LIKE "%D";

Q8. To study character functions.

A.

The character functions are primarily used to manipulate strings. Following are some queries which describe and explain the major string or character functions.

1. **BIN(N)** – returns the binary string value of N, returns null if N is null.
 - a. **SELECT BIN(12)** – returns '1100'.
2. **ASCII(C)** – returns the numeric value of leftmost character.
 - a. **SELECT ASCII('Me')** – returns 01001101.
3. **CHAR_LENGTH(STR)** – returns the length of string.
 - a. **SELECT CHAR_LENGTH('Hello World')** – returns 11.
4. **CONCAT(STR1,STR2.....)** – returns the single concatenated string.
 - a. **SELECT CONCAT('Hello',' World')** – returns 'Hello World' as a single string.
5. **LOWER(STR)** – converts the string in lowercase letters.
 - a. **SELECT LOWER('HELLO')** – returns 'hello'
6. **UPPER(STR)** – converts the string in uppercase letters.
 - a. **SELECT UPPER('hello')** – returns 'HELLO'.
7. **LTRIM(STR)** – returns the string str with leading spaces removed.
 - a. **SELECT LTRIM(' HELLO')** – returns 'HELLO'.
8. **RTRIM(STR)** – returns the string str with trailing spaces removed.
 - a. **SELECT RTRIM('HELLO ')** – returns 'HELLO'.
9. **LPAD(STR,LEN,PADSTR)** – returns the string str, left-padded with the string padstr to a length of len characters. If str is longer than len, the return value is shortened to len characters.
 - a. **SELECT LPAD('HI',5,'X')** – returns 'XXXHI'.
10. **RPAD(STR,LEN,PADSTR)** – returns the string str, right-padded with the string padstr to a length of len characters. If str is longer than len, the return value is shortened to len characters.
 - a. **SELECT RPAD('HI',5,'X')** – returns 'HIXXX'.

Q9. To study the numeric functions.

A.

SQL numeric functions are primarily used for numeric manipulation or mathematical calculations. Following are some queries which describe and explain the major numeric functions.

1. **ABS(X)** – returns the absolute value of number X.
 - a. **SELECT ABS(-2)** – returns 2.
2. **CEIL(X)** – returns the smallest integer value that is not smaller than X.
 - a. **SELECT CEIL(3.46)** – returns 4.
3. **FLOOR(X)** – returns the largest integer value that is not larger than X.
 - a. **SELECT FLOOR(7.99)** – returns 7.
4. **LEAST(N1,N2,N3.....)** – returns the lowest number in the series or column.
 - a. **SELECT LEAST(3,5,1,8,33)** – returns 1.
5. **GREATEST(N1,N2,N3....)** – returns the highest number in the series or column.
 - a. **SELECT GREATEST(3,5,1,8,33)** – returns 33.

Besides this, we have all trigonometric functions and logarithmic functions to perform mathematical calculations which provide ease in Database programming.

Q10. Execute queries to study the aggregate functions (sum, count, max, min, average).

A.

SELECT SUM(ESAL) 'TOTAL SALARY GIVEN' FROM EMP;

SELECT COUNT(ENAME) AS 'TOTAL EMPLOYEES' FROM EMP;

SELECT MAX(ESAL) AS 'MAX SALARY' FROM EMP;

SELECT MIN(ESAL) AS 'MIN SALARY' FROM EMP;

SELECT * FROM EMP WHERE ESAL<AVERAGE(ESAL);
