Assignment 2:

1. Select col1, col2,... from TableName;
2. Select col1, col2,...
   from TableName
   where condition1;
3. Select col1, col2,...
   from TableName
   where condition1 and condition2 or ...;
4. Select distinct col1 from TableName;
5. Alter table TableName
   add column colNew datatype_colNew; // add a new column
6. Alter table TableName
   rename column col_oldname to col_newname;
7. Alter table TableName drop column colName;
8. Alter table TableName
   rename to NewTableName;
9. Alter table TableName
   modify colName new_datatype;
   //change datatype of an existing column
10. Delete from TableName;
    // delete all rows from table,
    or some rows based on some condition(s), like using WHERE clause
11. Truncate table TableName;
    //delete all rows from table; no conditions applicable, no WHERE clause.

Assignment 3:

1. SELECT column_name(s)
   FROM table_name
   WHERE column_name IN (value1, value2, ...);
2. NOT IN
3. SELECT column1, column2, ...
   FROM table_name
   WHERE column LIKE pattern;
   e.g. *a)* SELECT * FROM Customers
   WHERE CustomerName LIKE 'a%';
   b) SELECT * FROM Customers
   WHERE CustomerName LIKE '%a';
4. The _ wildcard represents a single character. It can be any character or number, but each _ represents one, and only one, character.
   SELECT * FROM Customers
   WHERE city LIKE 'R_hi_' and/or condition2;
   '%R%hi%' - Rohit, Arohi, Aarohi, Arushi, Rishi, Rakshit, Rakshita, Rashi
   % denotes zero or more characters.
5. select name, oldmarks,
   oldmarks + gracemarks + attendance * 0.20 as Newmarks
   from sec25_IDmarks;
6. select name from sec25_IDmarks
   where quizmarks between 15 and 20;
7. update sec25_IDmarks
   set IDmarks = IDmarks + 5;
8. update sec25_IDmarks
   set IDmarks = IDmarks + 5
   where quizmarks between 18 and 20;
9. update sec25_IDmarks
   set final_labgrade = 'A'

where lab_testmarks > 25

and quizmarks between 15 and 20;

10. select assignment_marks * 12

    from sec25_marks;

11. select assignment_marks * 12

    AS Annual Assignment Marks

    from sec25_marks;

12. Delete from TableName where condition1 and/or condition2;

Q2 - Aggregate Operators:

1. Select avg(colName) from tableName;
2. Select department,

   avg(salary)

   From dept_table

   Group by department;
3. Select ID, name, department

   From instructor_table where salary=

   (select max(salary) from instructor_table);

4. Select sum(credits) from course

   Where department='CS';
5. Select count(*), sum(salary)

   From instructor_table

Where department='cs' or department='physics';
//use IN

6. Select building, sum(budget)
   From dept_table
   Group by building;
7. Select department, count(*)
   From instructr_table
   Group by department;
8. Select semester, count(course_id)
   From teaches_table
   Group by semester;
9. Select department, count(*)
   From instructr_table
   Group by department
   Having count(*) <2;
10. Select department, count(*)
   From instructr_table
   Where department!='Finance'
   Group by department

Having count(*) >= 2

Order by count(*) desc;

11. Select department, sum(salary)

From instructr_table

Group by department

Having sum(salary) > 80k;

12. Select sum(budget) as total_budget

From dept_table

Where builder='Watson';

13. Select max(salary)

from instructr_table

Where department='CS';

Aggregate Functions:

Sum - select sum(salary) from instructor;

Count

Average - select avg(credits) from course where dept='CS';

Min

Max

Count(*) - #rows including nulls

Count(colName) - #not-null values of colName

Count(distinct colName) - #not-null+unique values of colName

Group By:

select deptname, avg(budget) from dept group by deptname;

SCALAR FUNCTIONS

1. The DUAL table is a special one-row, one-column table present by default in Oracle and other database installations.
2. In Oracle, the table has a single VARCHAR2(1) column called DUMMY that has a value of 'X'.
3. It is suitable for use in selecting a pseudo column such as SYSDATE or USER.
4. Oracle Scalar Functions allow you to perform different calculations on data values. These functions operate on single rows only and produce one result per row. There are different types of Scalar Functions,
    a. String functions – functions that perform operations on character values.
    b. Numeric functions – functions that perform operations on numeric values.
    c. Date functions – functions that perform operations on date values.

d. Conversion functions – functions that convert column data types.
e. NULL-related Functions – functions for handling null values.

EXAMPLE COMMANDS:

Follow https://ramkedem.com/en/oracle-scalar-functions/ for explanations of functions:

- SELECT CONCAT('Hello' , 'World') FROM dual
  Result: 'HelloWorld'
- SELECT INSTR('hello' , 'e') FROM dual
  Result: 2
- SELECT LENGTH('hello') FROM dual
  Result: 5
- SELECT RTRIM(' hello ') FROM dual
  Result: ' hello'
- SELECT LTRIM(' hello ') FROM dual
  Result: 'hello '
- SELECT REPLACE('hello' , 'e' , '$') FROM dual
  Result: 'h$llo'
- SELECT REVERSE('hello') FROM dual
  Result: 'olleh'
- SELECT SUBSTR('hello' , 2,3) FROM dual
  Result: 'ell'

- SELECT LOWER('HELLO') FROM dual
  Result: 'hello'
- SELECT UPPER('hello') FROM dual
  Result: 'HELLO'
- SELECT INITCAP('hello') FROM dual
  Result: 'Hello'
- SELECT ADD_MONTHS('05-JAN-2001' , 4) FROM dual
  Result : '05-MAY-2001'
- select length('string value') from dual;
- select upper(substr(string_variable, 1, 3)) from table;
- select to_char(date_of_join, 'dd/mm/yyyy') from faculty_info;
- select name, months_between(sysdate, date_of_join) as months_diff from faculty_info;
- SELECT EXTRACT (DAY FROM SYSDATE) FROM dual
  Result : 16
- SELECT LAST_DAY('15-AUG-2014') FROM DUAL
  Result: '31-AUG-2014'
- SELECT MONTHS_BETWEEN('01-MAY-2010', '01-JAN-2010') FROM dual
  Result : 4
- SELECT NEXT_DAY('30-AUG-2014' , 'Sunday') FROM dual
  Result: '31-AUG-2014'

- SELECT SYSDATE FROM dual
  Result: (current date)
- select name,
  - months_between(sysdate, date_of_join) as months_diff, floor(months_between(sysdate, date_of_join)/12) as years_diff from faculty_info;
- select name, to_char(date_of_join, 'day') as joining_day from faculty_info;
- SELECT ROUND(59.9) FROM dual
  Result: 60
- SELECT ROUND(59.1) FROM dual
  Result: 59
- SELECT TRUNC(59.9) FROM dual
  Result: 59
- SELECT TO_CHAR(1506) FROM dual
  Result : The string value '98'
- SELECT TO_CHAR(1507, '$9,999') FROM dual
  Result : The string value '$1,507'
- SELECT TO_CHAR(sysdate, 'dd/mm/yyyy') FROM dual
  Result : The string value '01/01/2015'
- SELECT TO_DATE('01-MAY-2015') FROM dual
  Result : The date value '01-MAY-2015'

- SELECT TO_DATE('01/05/2015' , 'dd/mm/yyyy') FROM dual

  Result : The date value : '01-MAY-2015'
- SELECT TO_NUMBER('9432') FROM dual

  Result : The numeric value : 9432
- SELECT TO_NUMBER('$9,324' , '$9,999')

  Result : The numeric value 9324
- select sysdate + 15 from dual;
- select round(23.5879, 2) from dual;
- select 41 + power(3, 8) from dual;
- select sqrt(65536) from dual;
- select upper('will you fail the test??') as lowercase from dual;