**ASSIGNMENT 2**

**AIM:-**

Design at least 10 SQL queries for suitable database application using SQL DML statements: all types of Join, Sub-Query and View.

**OBJECTIVE:-**

To implement Join and Sub query operations.

**THEORY-**

**SUB QUERY**

Subquery or Inner query or Nested query is a query in a query. A subquery is usually added in the WHERE clause of the sql statement.

Most of the time, a subquery is used when you know how to search for a value using a SELECT statement, but do not know the exact value.

Subqueries are an alternate way of returning data from multiple tables.

Subqueries can be used with the following sql statements along with the comparison operators like =, <, >, >=, <= etc.(these are considered as single-row operators) while IN,ANY,ALL these are considered as multiple row operators.

Enclose subqueries in parentheses.

Place subqueries on the right side of the comparison condition

Syntax:

SELECT select\_list

FROM table

WHERE expr operator(SELECT select\_list

FROM table);

**TYPES OF QUERIES**

***Single-row subquery***:- Queries that return only one row from the inner SELECT statement.

***Multiple-row*** subquery:-Queries that return more than one row from the inner SELECT statement.

***Multiple-column subquery***:-Queries that return more than one column from the inner SELECT statement.

**SINGLE ROW QUERY: -**

Usually, a subquery should return only one record, but sometimes it can also return multiple records when used with operators like IN, NOT IN in the where clause.

**MULTI ROW QUERY: -**

Return more than one row.

Use multiple-row comparison operators

|  |  |  |
| --- | --- | --- |
| **Operator** | **Meaning** |  |
| IN | Equal to any member in the list |  |
| ANY | Compare value to each value returned by the subquery |  |
| ALL | Compare value to every value returned by the subquery |  |

**JOIN:-**

The **JOIN keyword** is used in an SQL statement to ***query data*** from two or more tables, based on a relationship between certain columns in these tables.

Tables in a database are often related to each other with keys.

A ***primary key*** is a column (or a combination of columns) with a unique value for each row. Each primary key value must be unique within the table. The purpose is to bind data together, across tables, without repeating all of the data in every table.

**Different SQL JOIN**

JOIN/inner join: Return rows when there is at least one match in both tables.

***LEFT JOIN***: Return all rows from the left table, even if there are no matches in the right table.

***RIGHT JOIN***: Return all rows from the right table, even if there are no matches in the left table.

***FULL JOIN***: Return rows when there is a match in one of the tables.

**PROGRAM:-**

**Person table**

Create Table person (P\_ID int(10),Last\_Name Varchar(15),First\_Name varchar(15),Address varchar(20),City varchar(10));

Select \* From person ;

**Order\_item table**

Create Table order\_item(O\_id int(10),Order\_no int(10),P\_ID int(10));

Insert Into order\_item Values(4,2000,8);

Select \* From order\_item;

**INNER JOIN**

SELECT person.Last\_Name, person.First\_Name, order\_item.Order\_no

-> FROM person

-> INNER JOIN order\_item

-> on person.P\_ID=order\_item.P\_ID;

**RIGHT JOIN**

SELECT person.Last\_Name, person.First\_Name, order\_item.Order\_no

-> FROM person

-> RIGHT JOIN order\_item

-> on person.P\_ID=order\_item.P\_ID

-> ;

**LEFT JOIN**

SELECT person.Last\_Name, person.First\_Name, order\_item.Order\_no

-> FROM person

-> LEFT JOIN order\_item

-> on person.P\_ID=order\_item.P\_ID

-> ;

**EMPLOYEE DATABASE:**

SELECT\* FROM EMPLOYEE;

**IN CLAUSE:-**

SELECT name,DESIGNATION

FROM EMPLOYEE

WHERE DESIGNATION NOT IN ('WEB DEVELOPER','APP DEVELOPER');

**SUB QUERY:-**

SELECT id, name FROM employee

WHERE name IN (SELECT name FROM employee

WHERE DESIGNATION= 'WEB DEVELOPER');

**GREATER THAN:**\_

SELECT name

FROM employee

WHERE salary >

(SELECT salary

FROM employee

WHERE name='Raj');

**EQUAL TO:**-

SELECT name

FROM employee

WHERE DESIGNATION =

(SELECT DESIGNATION

FROM employee

WHERE name='Raj');

**IN SUB-QUERY:-**

SELECT id,name,salary

FROM employee

WHERE salary IN (SELECT ***MIN(salary)***

FROM employee

Group BY DESIGNATION);

Group By:- <https://www.w3schools.com/sql/sql_groupby.asp>

**ANY CLAUSE:-**

SELECT name,id,salary

FROM employee

WHERE salary < ANY

(SELECT salary

FROM employee

WHERE DESIGNATION='SOFTWARE DEVELOPER');

**ALL CLAUSE:-**

SELECT name,id,salary

FROM employee

WHERE salary < ALL

(SELECT salary

FROM employee

WHERE DESIGNATION='SOFTWARE DEVELOPER');

**Queries and Output Snapshots:**

**Add all types of sub queries, Join Operations along with select clauses in your assignments.**

**Order by:-** [**https://www.w3schools.com/sql/sql\_orderby.asp**](https://www.w3schools.com/sql/sql_orderby.asp)

**NOT:-** [**https://www.w3schools.com/sql/sql\_not.asp**](https://www.w3schools.com/sql/sql_not.asp)

**Enter password: \*\*\*\*\*\*\*\*\***

**Welcome to the MySQL monitor. Commands end with ; or \g.**

**Your MySQL connection id is 12**

**Server version: 8.0.36 MySQL Community Server - GPL**

**Copyright (c) 2000, 2024, Oracle and/or its affiliates.**

**Oracle is a registered trademark of Oracle Corporation and/or its**

**affiliates. Other names may be trademarks of their respective**

**owners.**

**Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.**

**mysql> create database ass**

**-> ;**

**Query OK, 1 row affected (0.02 sec)**

**mysql> show databases;**

**+--------------------+**

**| Database |**

**+--------------------+**

**| ass |**

**| information\_schema |**

**| mysql |**

**| performance\_schema |**

**| sys |**

**+--------------------+**

**5 rows in set (0.00 sec)**

**mysql> Use ass;**

**Database changed**

**mysql> create table employee;**

**ERROR 4028 (HY000): A table must have at least one visible column.**

**mysql> create table employee(emp\_id int(10) , emp\_name varchar(20) , salary int(10) , residence varchar(10));**

**Query OK, 0 rows affected, 2 warnings (0.04 sec)**

**mysql> insert into employee values((1 ,"Shreyas" , 120 , "pandharpur"),(2 ,"Akshay" , 500 , "pune"));**

**ERROR 1136 (21S01): Column count doesn't match value count at row 1**

**mysql> insert into employee values(1 ,"Shreyas" , 120 , "pandharpur");**

**Query OK, 1 row affected (0.00 sec)**

**mysql> ^C**

**mysql> insert into employee values(2 ,"Akshay" , 1820 , "pune");**

**Query OK, 1 row affected (0.01 sec)**

**mysql> insert into employee values(3 ,"Shreya" , 12 , "nasik");**

**Query OK, 1 row affected (0.04 sec)**

**mysql> select \* from employee;**

**+--------+----------+--------+------------+**

**| emp\_id | emp\_name | salary | residence |**

**+--------+----------+--------+------------+**

**| 1 | Shreyas | 120 | pandharpur |**

**| 2 | Akshay | 1820 | pune |**

**| 3 | Shreya | 12 | nasik |**

**+--------+----------+--------+------------+**

**3 rows in set (0.00 sec)**

**mysql> create table student;**

**ERROR 4028 (HY000): A table must have at least one visible column.**

**mysql> create table employee(std\_id int(10) , std\_name varchar(20) , mark int(10) , residence varchar(10));**

**ERROR 1050 (42S01): Table 'employee' already exists**

**mysql> create table student(std\_id int(10) , std\_name varchar(20) , mark int(10) , residence varchar(10));**

**Query OK, 0 rows affected, 2 warnings (0.03 sec)**

**mysql> insert into employee values(1 ,"Shre" , 12 , "pandharpur");**

**Query OK, 1 row affected (0.00 sec)**

**mysql> insert into employee values(2 ,"Aks" , 10 , "pune");**

**Query OK, 1 row affected (0.00 sec)**

**mysql> insert into employee values(3 ,"Sha" , 1 , "nasik");**

**Query OK, 1 row affected (0.01 sec)**

**mysql> select \* from student;**

**Empty set (0.00 sec)**

**mysql> select \* from employee**

**-> ;**

**+--------+----------+--------+------------+**

**| emp\_id | emp\_name | salary | residence |**

**+--------+----------+--------+------------+**

**| 1 | Shreyas | 120 | pandharpur |**

**| 2 | Akshay | 1820 | pune |**

**| 3 | Shreya | 12 | nasik |**

**| 1 | Shre | 12 | pandharpur |**

**| 2 | Aks | 10 | pune |**

**| 3 | Sha | 1 | nasik |**

**+--------+----------+--------+------------+**

**6 rows in set (0.00 sec)**

**mysql> insert into student values(1 ,"Shre" , 12 , "pandharpur");**

**Query OK, 1 row affected (0.01 sec)**

**mysql> insert into student values(2 ,"Aks" , 10 , "pune");**

**Query OK, 1 row affected (0.00 sec)**

**mysql> insert into student values(3 ,"Sha" , 1 , "nasik");**

**Query OK, 1 row affected (0.01 sec)**

**mysql> select \* from student**

**-> ;**

**+--------+----------+------+------------+**

**| std\_id | std\_name | mark | residence |**

**+--------+----------+------+------------+**

**| 1 | Shre | 12 | pandharpur |**

**| 2 | Aks | 10 | pune |**

**| 3 | Sha | 1 | nasik |**

**+--------+----------+------+------------+**

**3 rows in set (0.00 sec)**

**mysql> select employee.emp\_name,employee.salary,employee.residence,student.residences**

**-> ;**

**ERROR 1109 (42S02): Unknown table 'employee' in field list**

**mysql> select employee.emp\_name,employee.salary,employee.residence,student.residences**

**-> from employee**

**-> inner join student**

**-> on employee.residence=student.residence;**

**ERROR 1054 (42S22): Unknown column 'student.residences' in 'field list'**

**mysql> select employee.emp\_name,employee.salary,employee.residence,student.residence**

**-> from employee**

**-> inner join student**

**-> on employee.residence=student.residence;**

**+----------+--------+------------+------------+**

**| emp\_name | salary | residence | residence |**

**+----------+--------+------------+------------+**

**| Shreyas | 120 | pandharpur | pandharpur |**

**| Akshay | 1820 | pune | pune |**

**| Shreya | 12 | nasik | nasik |**

**| Shre | 12 | pandharpur | pandharpur |**

**| Aks | 10 | pune | pune |**

**| Sha | 1 | nasik | nasik |**

**+----------+--------+------------+------------+**

**6 rows in set (0.00 sec)**

**mysql> insert into student values(4 ,"Sa" , 8 , "nak");**

**Query OK, 1 row affected (0.00 sec)**

**mysql> select employee.emp\_name,employee.salary,employee.residence,student.residence**

**-> from employee**

**-> -> inner join student**

**-> from employee**

**-> ;**

**ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '-> inner join student**

**from employee' at line 3**

**mysql> select employee.emp\_name,employee.salary,employee.residence,student.residence**

**-> from employee**

**-> inner join student**

**-> on employee.residence=student.residence;**

**+----------+--------+------------+------------+**

**| emp\_name | salary | residence | residence |**

**+----------+--------+------------+------------+**

**| Shreyas | 120 | pandharpur | pandharpur |**

**| Akshay | 1820 | pune | pune |**

**| Shreya | 12 | nasik | nasik |**

**| Shre | 12 | pandharpur | pandharpur |**

**| Aks | 10 | pune | pune |**

**| Sha | 1 | nasik | nasik |**

**+----------+--------+------------+------------+**

**6 rows in set (0.00 sec)**

**mysql> select \* from student;**

**+--------+----------+------+------------+**

**| std\_id | std\_name | mark | residence |**

**+--------+----------+------+------------+**

**| 1 | Shre | 12 | pandharpur |**

**| 2 | Aks | 10 | pune |**

**| 3 | Sha | 1 | nasik |**

**| 4 | Sa | 8 | nak |**

**+--------+----------+------+------------+**

**4 rows in set (0.00 sec)**

**mysql> select employee.emp\_name,employee.salary,employee.residence,student.residence**

**-> from student**

**-> inner join employee**

**-> on employee.residence=student.residence;**

**+----------+--------+------------+------------+**

**| emp\_name | salary | residence | residence |**

**+----------+--------+------------+------------+**

**| Shreyas | 120 | pandharpur | pandharpur |**

**| Akshay | 1820 | pune | pune |**

**| Shreya | 12 | nasik | nasik |**

**| Shre | 12 | pandharpur | pandharpur |**

**| Aks | 10 | pune | pune |**

**| Sha | 1 | nasik | nasik |**

**+----------+--------+------------+------------+**

**6 rows in set (0.00 sec)**

**mysql> select employee.emp\_name,employee.salary,employee.residence,student.residence**

**-> from student**

**-> inner join employee**

**-> ;**

**+----------+--------+------------+------------+**

**| emp\_name | salary | residence | residence |**

**+----------+--------+------------+------------+**

**| Shreyas | 120 | pandharpur | nak |**

**| Shreyas | 120 | pandharpur | nasik |**

**| Shreyas | 120 | pandharpur | pune |**

**| Shreyas | 120 | pandharpur | pandharpur |**

**| Akshay | 1820 | pune | nak |**

**| Akshay | 1820 | pune | nasik |**

**| Akshay | 1820 | pune | pune |**

**| Akshay | 1820 | pune | pandharpur |**

**| Shreya | 12 | nasik | nak |**

**| Shreya | 12 | nasik | nasik |**

**| Shreya | 12 | nasik | pune |**

**| Shreya | 12 | nasik | pandharpur |**

**| Shre | 12 | pandharpur | nak |**

**| Shre | 12 | pandharpur | nasik |**

**| Shre | 12 | pandharpur | pune |**

**| Shre | 12 | pandharpur | pandharpur |**

**| Aks | 10 | pune | nak |**

**| Aks | 10 | pune | nasik |**

**| Aks | 10 | pune | pune |**

**| Aks | 10 | pune | pandharpur |**

**| Sha | 1 | nasik | nak |**

**| Sha | 1 | nasik | nasik |**

**| Sha | 1 | nasik | pune |**

**| Sha | 1 | nasik | pandharpur |**

**+----------+--------+------------+------------+**

**24 rows in set (0.00 sec)**

**mysql> select employee.emp\_name,employee.salary,employee.residence,student.residence**

**-> from student**

**-> left join employee**

**-> on employee.residence=student.residence;**

**+----------+--------+------------+------------+**

**| emp\_name | salary | residence | residence |**

**+----------+--------+------------+------------+**

**| Shre | 12 | pandharpur | pandharpur |**

**| Shreyas | 120 | pandharpur | pandharpur |**

**| Aks | 10 | pune | pune |**

**| Akshay | 1820 | pune | pune |**

**| Sha | 1 | nasik | nasik |**

**| Shreya | 12 | nasik | nasik |**

**| NULL | NULL | NULL | nak |**

**+----------+--------+------------+------------+**

**7 rows in set (0.00 sec)**

**mysql> select employee.emp\_name,employee.salary,employee.residence,student.residence**

**-> from employee**

**-> left join student**

**-> on employee.residence=student.residence;**

**+----------+--------+------------+------------+**

**| emp\_name | salary | residence | residence |**

**+----------+--------+------------+------------+**

**| Shreyas | 120 | pandharpur | pandharpur |**

**| Akshay | 1820 | pune | pune |**

**| Shreya | 12 | nasik | nasik |**

**| Shre | 12 | pandharpur | pandharpur |**

**| Aks | 10 | pune | pune |**

**| Sha | 1 | nasik | nasik |**

**+----------+--------+------------+------------+**

**6 rows in set (0.00 sec)**

**mysql> SELECT emp\_name from employee**

**-> where salary >( SELECT emp\_name from employee**

**-> where name = 'shreyas');**

**ERROR 1054 (42S22): Unknown column 'name' in 'where clause'**

**mysql> SELECT emp\_name from employee**

**-> where salary >( SELECT emp\_name from employee**

**-> where emp\_name = 'shreyas');**

**+----------+**

**| emp\_name |**

**+----------+**

**| Shreyas |**

**| Akshay |**

**| Shreya |**

**| Shre |**

**| Aks |**

**| Sha |**

**+----------+**

**6 rows in set (0.00 sec)**

**mysql> SELECT emp\_name from employee**

**-> where salary < ( SELECT emp\_name from employee**

**-> where emp\_name = 'shreyas');**

**Empty set (0.00 sec)**

**mysql> SELECT emp\_name from employee**

**-> where salary < ( SELECT salary from employee**

**-> where emp\_name = 'shreyas');**

**+----------+**

**| emp\_name |**

**+----------+**

**| Shreya |**

**| Shre |**

**| Aks |**

**| Sha |**

**+----------+**

**4 rows in set (0.00 sec)**

**mysql>**