Shrawani shinde

2213180

**Experiment No. -5**

**Aim -** Write a Database Query for Joins, Nested queries, Sub-queries of Manufacturing industry / Hospital/ Company table**.**

**Software Required** - SQL Server 15.0 /16.0

**Theory :-**

* SQL stands for Structured Query Language. It is used for storing and managing data in relational database management system (RDMS).
* It is a standard language for Relational Database System. It enables a user to create, read, update and delete relational databases and tables.

**1)SQL Joins** :-

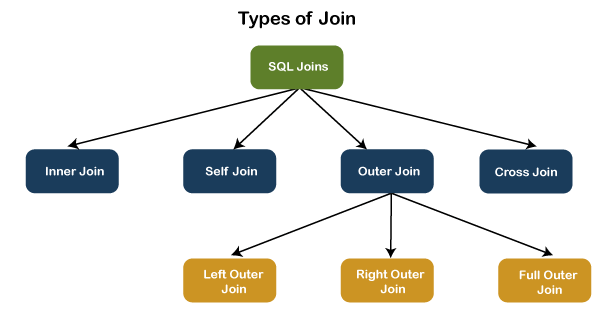
A SQL Join statement combines data or rows from two or more tables based on a common field between them. The join keyword merges two or more tables and creates a temporary image of the merged table. Then according to the conditions provided, it extracts the required data from the image table, and once data is fetched, the temporary image of the merged tables is dumped.

In a JOIN query, a condition indicates how two tables are related:

* Choose columns from each table that should be used in the join. A join condition indicates a foreign key from one table and its corresponding key in the other table.
* Specify the logical operator to compare values from the columns like =, <, or >.

**Types of JOINS in SQL Server**

[SQL Server](https://www.javatpoint.com/sql-server-tutorial) mainly supports **four types of JOINS**, and each join type defines how two tables are related in a query. The following are types of join supports in SQL Server:



a) **INNER JOIN**

This JOIN returns all records from multiple tables that satisfy the specified join condition. It is the simple and most popular form of join and assumes as a default join. If we omit the INNER keyword with the JOIN query, we will get the same output.

INNER JOIN Syntax

The following syntax illustrates the use of INNER JOIN in SQL Server:

SELECT columns

FROM table1

INNER JOIN table2 ON condition1

INNER JOIN table3 ON condition2

**b) SELF JOIN**

A table is joined to itself using the SELF JOIN. It means that each table row is combined with itself and with every other table row. The SELF JOIN can be thought of as a JOIN of two copies of the same tables. We can do this with the help of table name aliases to assign a specific name to each table's instance. The table aliases enable us to use the table's temporary name that we are going to use in the query.

**SELF JOIN Syntax**

SELECT T1.col\_name, T2.col\_name...

FROM table1 T1, table1 T2

WHERE join\_condition;

**c) CROSS JOIN**

CROSS JOIN in SQL Server combines all of the possibilities of two or more tables and returns a result that includes every row from all contributing tables. It's also known as CARTESIAN JOIN because it produces the Cartesian product of all linked tables.

**CROSS JOIN Syntax**

SELECT column\_lists

FROM table1

CROSS JOIN table2;

**d) OUTER JOIN**

OUTER JOIN in SQL Server returns all records from both tables that satisfy the join condition. In other words, this join will not return only the matching record but also return all unmatched rows from one or both tables.

* LEFT OUTER JOIN
* RIGHT OUTER JOIN
* FULL OUTER JOIN

**3)Nested Queries :-**

In nested queries, a query is written inside a query. The result of inner query is used in execution of outer query. We will use STUDENT, COURSE, STUDENT\_COURSE tables for understanding nested queries.

**STUDENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S\_ID** | **S\_NAME** | **S\_ADDRESS** | **S\_PHONE** | **S\_AGE** |
| S1 | RAM | DELHI | 9455123451 | 18 |
| S2 | RAMESH | GURGAON | 9652431543 | 18 |
| S3 | SUJIT | ROHTAK | 9156253131 | 20 |
| S4 | SURESH | DELHI | 9156768971 | 18 |

**COURSE**

|  |  |
| --- | --- |
| **C\_ID** | **C\_NAME** |
| C1 | DSA |
| C2 | Programming |
| C3 | DBMS |

**STUDENT\_COURSE**

|  |  |
| --- | --- |
| **S\_ID** | **C\_ID** |
| S1 | C1 |
| S1 | C3 |
| S2 | C1 |
| S3 | C2 |
| S4 | C2 |
| S4 | C3 |

In independent nested queries, query execution starts from innermost query to outermost queries. The execution of inner query is independent of outer query, but the result of inner query is used in execution of outer query. Various operators like IN, NOT IN, ANY, ALL etc are used in writing independent nested queries.

**IN:** If we want to find out **S\_ID** who are enrolled in **C\_NAME** ‘DSA’ or ‘DBMS’, we can write it with the help of independent nested query and IN operator. From **COURSE** table, we can find out **C\_ID**for **C\_NAME** ‘DSA’ or DBMS’ and we can use these **C\_ID**s for finding **S\_ID**s from **STUDENT\_COURSE** TABLE.

**STEP 1:** Finding **C\_ID** for **C\_NAME** =’DSA’ or ‘DBMS’

Select **C\_ID** from **COURSE** where **C\_NAME** = ‘DSA’ or **C\_NAME** = ‘DBMS’

**STEP 2:** Using **C\_ID** of step 1 for finding **S\_ID**

Select **S\_ID** from **STUDENT\_COURSE** where **C\_ID** IN

(SELECT **C\_ID** from **COURSE** where **C\_NAME** = ‘DSA’ or **C\_NAME**=’DBMS’);

Output :- The inner query will return a set with members C1 and C3 and outer query will return those S\_IDs for which C\_ID is equal to any member of set (C1 and C3 in this case). So, it will return S1, S2 and S4.

# 4)SQL Sub Query

A Subquery is a query within another SQL query and embedded within the WHERE clause.

* A subquery can be placed in a number of SQL clauses like WHERE clause, FROM clause, HAVING clause.
* You can use Subquery with SELECT, UPDATE, INSERT, DELETE statements along with the operators like =, <, >, >=, <=, IN, BETWEEN, etc.
* A subquery is a query within another query. The outer query is known as the main query, and the inner query is known as a subquery.
* Subqueries are on the right side of the comparison operator.
* A subquery is enclosed in parentheses.
* In the Subquery, ORDER BY command cannot be used. But GROUP BY command can be used to perform the same function as ORDER BY command.

**a). Subqueries with the Select Statement**

* SQL subqueries are most frequently used with the Select statement.
* Syntax:-

SELECT column\_name

FROM table\_name

WHERE column\_name expression operator

( SELECT column\_name  from table\_name WHERE ... );

**b) Subqueries with the INSERT Statement**

* SQL subquery can also be used with the Insert statement. In the insert statement, data returned from the subquery is used to insert into another table.
* In the subquery, the selected data can be modified with any of the character, date functions.
* Syntax:

INSERT INTO table\_name (column1, column2, column3....)

SELECT \*

FROM table\_name  WHERE VALUE OPERATOR

**c) Subqueries with the UPDATE Statement**

The subquery of SQL can be used in conjunction with the Update statement. When a subquery is used with the Update statement, then either single or multiple columns in a table can be updated.

Syntax

UPDATE table

SET column\_name = new\_value

WHERE VALUE OPERATOR

(SELECT COLUMN\_NAME

FROM TABLE\_NAME

WHERE condition);

**d) Subqueries with the DELETE Statement**

The subquery of SQL can be used in conjunction with the Delete statement just like any other statements mentioned above.

**Syntax**

DELETE FROM TABLE\_NAME

WHERE VALUE OPERATOR

   (SELECT COLUMN\_NAME

   FROM TABLE\_NAME

   WHERE condition);

**SQL QUERIES ASSIGNMENT FOR HOSPITAL MANAGEMENT**

**List of tables with attribute details in the hospital database:**

* **Physician**

employeeid – this is a unique ID of a physician

name – this is the name of a physician

position – this is the designation of a physician

ssn – this is a security number of a physician

* **Department**

departmentid – this is a unique ID for a department

name – this is the name of a department

head – this is the ID of the physician who is the head of a department, referencing to the column employeeid of the table physician

* **affiliated\_with**

physician – this is the ID of the physicians which is referencing to the column employeeid of the physician table

department – this is the ID the department which is referencing to the column departmentid of the department table

primary affiliation – this is a logical column which indicate that whether the physicians are yet to be affiliated or not

*Note: The combination of physician, department will come once in that table.*

* **patient**

ssn – this is a unique ID for each patient

name – this is the name of the patient

address – this is the address of the patient

phone – this is the phone number of the patient

insuranceid – this is the insurance id of the patient

pcp – this is the ID of the physician who primarily checked up the patient which is referencing to the column employeeid of the physician table

* **nurse:**

employeeid – this is the unique ID for a nurse

name – name of the nurses

position – the designation of the nurses

registered – this is a logical column which indicate that whether the nurses are registered for nursing or not

ssn – this is the security number of a nurse

* **Appointment**

appointmentid – this is the unique ID for an appointment

patient – this is the ID of each patient which is referencing to the ssn column of patient table

prepnurse – the ID of the nurse who may attend the patient with the physician, which is referencing to the column employeeid of the nurse table

physician – this is the ID the physicians which is referencing to the employeeid column of the physician table

start\_dt\_time – this is the schedule date and approximate time to meet the physician

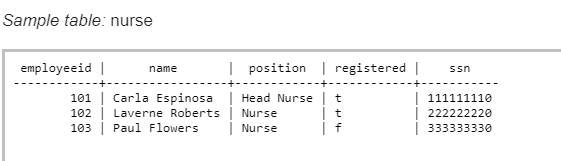
end\_dt\_time – this is the schedule date and approximate time to end the meeting

examinationroom – this the room where to meet a patient to the physician

* medication
* prescribes
* block
* room
* on\_call
* stay
* undergoes

**Note:- Create above tables and insert appropriate data into it using DDL and DML commands.**

**1)** Write a query in SQL to find all the information of the nurses who are yet to be registered



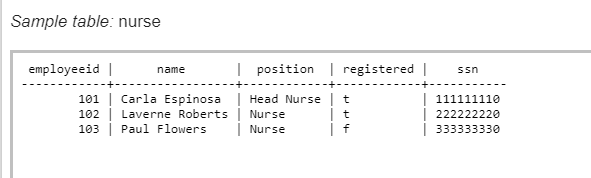
**Sample Solution:**

SELECT \*

FROM nurse

WHERE registered='false';

**2)** Write a query in SQL to find the name of the nurse who are the head of their department



**Sample Solution:**

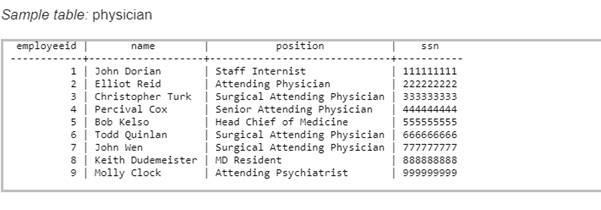
SELECT name AS "Name",

       POSITION AS "Position"

FROM nurse

WHERE POSITION='Head Nurse';

3) Write a query in SQL to obtain the name of the physicians who are the head of each department



**Sample Solution:**

SELECT d.name AS "Department",

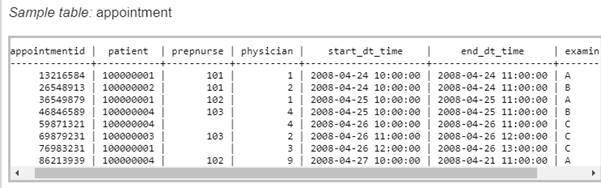
       p.name AS "Physician"

FROM department d,

     physician p

WHERE d.head=p.employeeid;

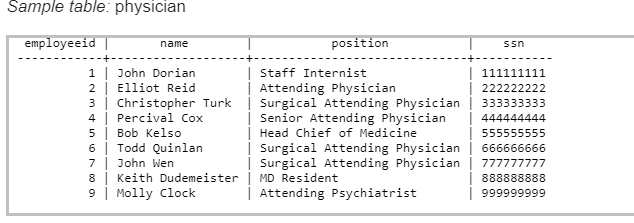
4) Write a query in SQL to count the number of patients who taken appointment with at least one physician.

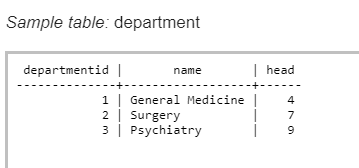


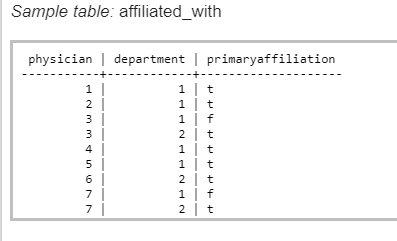
**Sample Solution:**

SELECT count(DISTINCT patient) AS "No. of patients taken at least one appointment" FROM appointment;

5) Write a query in SQL to obtain the name of the physician and the departments they are affiliated with.







**Sample Solution:**

SELECT p.name AS "Physician",

       d.name AS "Department"

FROM physician p,

     department d,

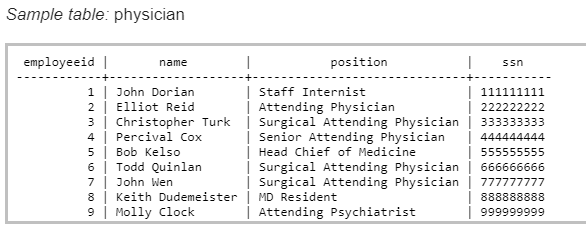
     affiliated\_with a

WHERE p.employeeid=a.physician

  AND a.department=d.departmentid;

**6)** Write a query in SQL to obtain the name of the patients with their physicians by whom they got their preliminary treatment.





**Sample Solution:**

SELECT t.name AS "Patient",

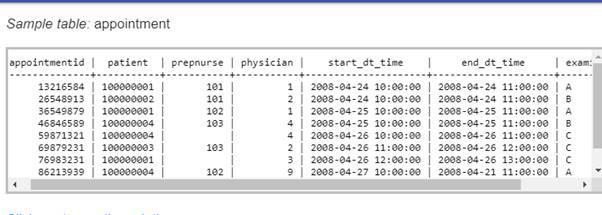
       t.address AS "Address",

       p.name AS "Physician"

FROM patient t

JOIN physician p ON t.pcp=p.employeeid;

7) Write a query in SQL to count number of unique patients who got an appointment for examination room C.



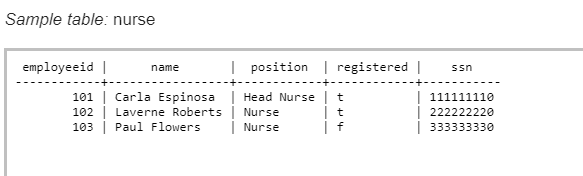
**Sample Solution:**

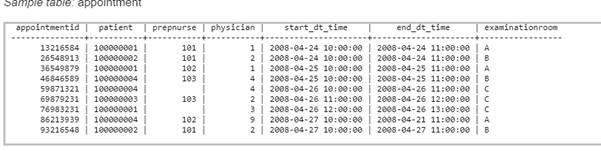
SELECT count(DISTINCT patient) AS "No. of patients got appointment for room C"

FROM appointment

WHERE examinationroom='C';

8) Write a query in SQL to find the name of the nurses and the room scheduled, where they will assist the physicians.





**Sample Solution:**

SELECT n.name AS "Name of the Nurse",

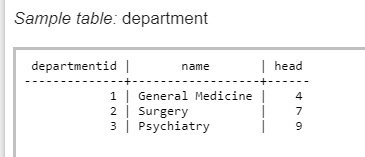
       a.examinationroom AS "Room No."

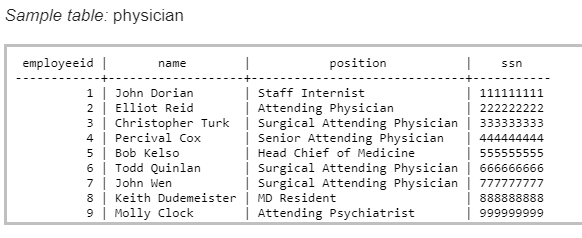
FROM nurse n

JOIN appointment a ON a.prepnurse=n.employeeid;

9) Write a query in SQL to Obtain the names of all patients whose primary care is taken by a physician who is not the head of any department and name of that physician along with their primary care physician.







**Sample Solution:**

SELECT pt.name AS "Patient",

       p.name AS "Primary care Physician"

FROM patient pt

JOIN physician p ON pt.pcp=p.employeeid

WHERE pt.pcp NOT IN

    (SELECT head     FROM department);

**Exercise**

1. **Write SQL nested, sub queries for Manufacturing Industry.**
2. **Write SQL nested, sub queries for Company database.**
3. **Write SQL nested, sub queries for Library management system.**
4. **Write SQL nested, sub queries for Online Railway reservation system**
5. **Write SQL nested, sub queries for any E-commerce website.**

