Q #1) What are Inner JOINS used in SQL? Explain with Syntax.

Ans. Inner joins are the commands used to combine the values of two tables where matching common records are found, using a certain condition.

Syntax:

```
SELECT Column_list
FROM TABLE1
INNER JOIN TABLE2
ON Table1.ColName = Table2.ColName
```

Explanation:

```
create database infosys;

use infosys;


create table employee1(

empid VARCHAR(20),

empLN VARCHAR(20),

empFN VARCHAR(20),

empjob VARCHAR(20)

);


select * from employee1;

INSERT into employee1 VALUES('EC001','Verma','Akhil','Administration');

INSERT into employee1 VALUES('EC002','Samson','Nikita','Asst Manager');

INSERT into employee1 VALUES('EC003','Jordan','Nil','In Charge');

INSERT into employee1 VALUES('EC004','Smith','Joe','Technician');

create table employee2(

empid VARCHAR(20),

empLN VARCHAR(20),

empFN VARCHAR(20),

joindate VARCHAR(20)

);
```

select * from employee2;

INSERT into employee2 VALUES('EC012','Verma','Akhil','2016/04/18');

INSERT into employee2 VALUES('EC013','Samson','Nikita','2016/04/19');

INSERT into employee2 VALUES('EC014','Jordan','Nil','2016/05/01');

select employee1.empjob ,employee2.joindate

from employee1

inner join employee2 on employee1.empLN = employee2.empLN;

Q2.) What are Left JOINS used in SQL? Explain with Syntax
Ans. The LEFT JOIN keyword returns all records from the left table , and the matching records from the right table . The result is 0 records from the right side, if there is no match.

Syntax:

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON Table1.ColName = Table2.ColName
```

Explanation:

```
create database employeeinfo;

use employeeinfo;
create table employee1(
empid VARCHAR(20),
empLN VARCHAR(20),
empFN VARCHAR(20),
empjob VARCHAR(20)
);


select * from employee1;
INSERT into employee1
VALUES('EC011','Verma','Akhil','Administration');
INSERT into employee1 VALUES('EC012','Samson','Nikita','Asst
Manager');
INSERT into employee1 VALUES('EC013','Jordan','Nil','In Charge');
INSERT into employee1 VALUES('EC014','Smith','Joe','Technician');
```

```
create table employee2(
empid VARCHAR(20),
empLN VARCHAR(20),
empFN VARCHAR(20),
joindate VARCHAR(20)
);


select * from employee2;
INSERT into employee2 VALUES('EC012','Verma','Akhil','2016/04/18');
INSERT into employee2 VALUES('EC013','Samson','Nikita','2016/04/19');
INSERT into employee2 VALUES('EC014','Jordan','Nil','2016/05/01');
INSERT into employee2 VALUES('NULL','NULL','NULL','2016/03/01');


select employee1.empjob ,employee2.joindate
from employee1
left join employee2 on employee1.empLN = employee2.empLN;
```

Q #3) What are Right JOINS used in SQL? Explain with Syntax

Ans .   The RIGHT JOIN keyword returns all records from the right table , and the matching if

records from the left table (table1). The result is 0 records from the left side, there is no
match.

Syntax:

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```


 Explanation:

```
create database employeeinfo;
```


```
use employeeinfo;

create table employee1(

empid VARCHAR(20),

empLN VARCHAR(20),
```

```
empFN VARCHAR(20),
empjob VARCHAR(20)
);


select * from employee1;
INSERT into employee1 VALUES('EC011','Verma','Akhil','Administration');
INSERT into employee1 VALUES('EC012','Samson','Nikita','Asst Manager');
INSERT into employee1 VALUES('EC013','Jordan','Nil','In Charge');
INSERT into employee1 VALUES('EC014','Smith','Joe','Technician');
create table employee2(
empid VARCHAR(20),
empLN VARCHAR(20),
empFN VARCHAR(20),
joindate VARCHAR(20)
);


select * from employee2;
INSERT into employee2 VALUES('EC012','Verma','Akhil','2016/04/18');
INSERT into employee2 VALUES('EC013','Samson','Nikita','2016/04/19');
INSERT into employee2 VALUES('EC014','Jordan','Nil','2016/05/01');
INSERT into employee2 VALUES('NULL','NULL','NULL','2016/03/01');


select employee1.empjob ,employee2.joindate
from employee1
right join employee2 on employee1.empLN = employee2.empLN;
```

Q #4) What are Full JOINS used in SQL? Explain with Syntax.

Ans. The FULL OUTER JOIN keyword returns all records when there is a match in left or right table records.

Syntax: SELECT column_name(s)
           FROM table1
           FULL OUTER JOIN table2
           ON table1.column_name = table2.column_name;

Explanation:

create database employeeinfo;

use employeeinfo;

create table employee1(

empid VARCHAR(20),

empLN VARCHAR(20),

empFN VARCHAR(20),

empjob VARCHAR(20)

);

select * from employee1;

INSERT into employee1 VALUES('EC011','Verma','Akhil','Administration');

INSERT into employee1 VALUES('EC012','Samson','Nikita','Asst Manager');

INSERT into employee1 VALUES('EC013','Jordan','Nil','In Charge');

INSERT into employee1 VALUES('EC014','Smith','Joe','Technician');

create table employee2(

empid VARCHAR(20),

empLN VARCHAR(20),

empFN VARCHAR(20),

joindate VARCHAR(20)

);

select * from employee2;

INSERT into employee2 VALUES('EC012','Verma','Akhil','2016/04/18');

INSERT into employee2 VALUES('EC013','Samson','Nikita','2016/04/19');

INSERT into employee2 VALUES('EC014','Jordan','Nil','2016/05/01');

INSERT into employee2 VALUES('NULL','NULL','NULL','2016/03/01');

select employee1.empjob ,employee2.joindate

from employee1

full outer join employee2 on employee1.empLN = employee2.empLN;

Q1 Write an SQL query to fetch the current date-time from the system

Ans. Select CURRENT_TIMESTAMP AS "CURRENTTIMESTAMP";

**Q2. Find the Nth highest consultation fees from the PatientsCheckup table with
4 and without using the TOP/LIMIT keywords.**

Ans. Without LIMIT keywords

   select PatientId, consultationfees from patientscheckuptable p1

   where 0 = (select count(Distinct consultationfees) from patientscheckuptable p2
                  where p2.consultationfees > p1.consultationfees);

   With LIMIT Keywords

   select consultationfees from patientscheckuptable order by consultationfees desc
limit 1;

**Q3. Write a query to fetch top N records using the TOP/LIMIT, ordered by
   Consultation Fees**

Ans. select consultationfees from patientscheckuptable order by consultationfees asc limit n;

➔ Where n is the number of records.

Q4. Write a query to fetch even and odd rows from a table.

Ans. select * from patientscheckuptable where mod(PatientId,2) = 0;

select * from patientscheckuptable where mod(PatientId,2) != 0;