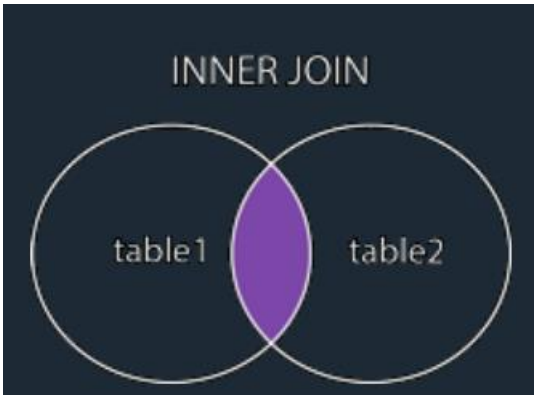
Name	Adwait S Purao
UID no.	2021300101
Experiment No.	4

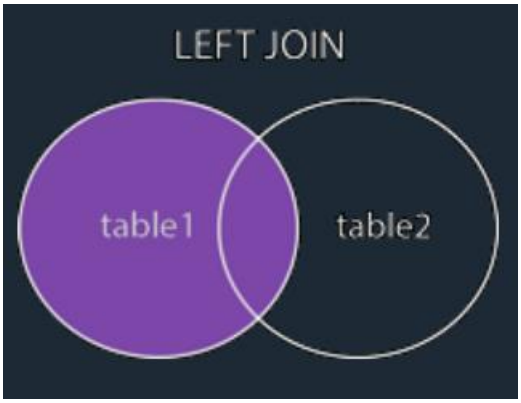
AIM:	To learn and apply the various Joins to the database
Program 1	
PROBLEM STATEMENT :	Peform Join operations on dtabase. -Inner joins -Natural joins -Outer joins -left -right -cross Joins
Theory :	SQL Join statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are as follows: INNER JOIN LEFT JOIN RIGHT JOIN CROSS JOIN NATURAL JOIN SQL Joins: □ A JOIN clause is used to combine rows from two or more tables, based on a related column between them □ It is used to retrieve data from multiple tables Different types of joins are: **1. Inner Join:** □ It returns the values which have matching values in both tables. □ Syntax: SELECT column_name(s) FROM table1 INNER JOIN table2 ON table1.column_name = table2.column_name;



2. Left Outer Join:

 It returns all the matched records of both the tables and unmatched values of the left table.
 Syntax:

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

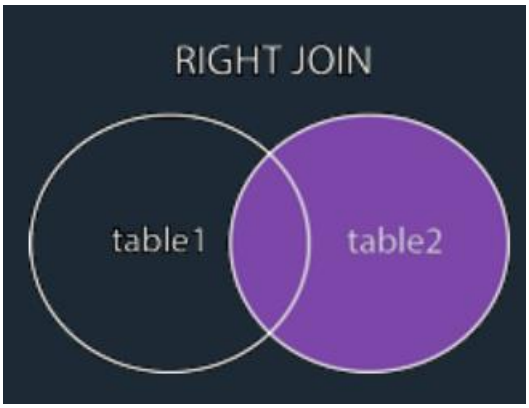


3. Right Outer Join:

 It returns all the matched records of both the tables and unmatched values of the right table.
 Syntax:

```
SELECT column_name(s)
```

FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;



4. Full Join:

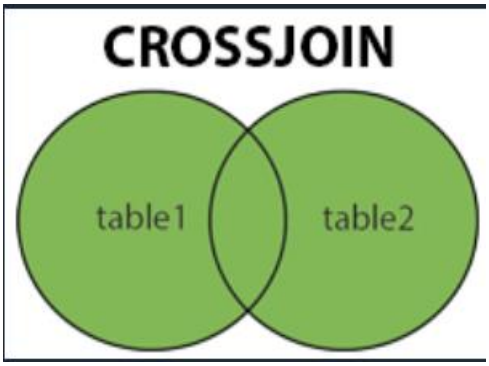
 It returns all records when there is a match in left (table1) or right (table2) table records.
 Syntax:

SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;

5. Self Join:

 It is a regular join but the table is joined with itself.
 Syntax:

SELECT column_name(s)
FROM table1 T1, table1 T2

	WHERE condition; **6. Cross join:** □ It returns all the records from both the tables. □ Syntax: SELECT column_name(s) FROM table1 CROSS JOIN table2; 
Queries	Code: CREATE DATABASE Hotel; use Hotel CREATE TABLE Hotel (H_Name Varchar(120) Not Null, H_ID int Primary key, H_Address Varchar(200) Not Null, H_Num_Emp int, H_vacancies int) ; CREATE TABLE Employee (E_Name Varchar(70), E_Type Varchar(50), E_ID int primary key, H_ID int,

```
foreign key(H_ID) references Hotel(H_ID),
LastName varchar(255),
FirstName varchar(255),
Address varchar(255),
City varchar(255),
E_Contact int,
E_Salary int not null check(E_salary>0)

);
create table Room(
R_no int primary key,
R_vacany boolean default true,
R_price int not null,
R_type varchar(30),
H_ID int references Hotel(H_ID)

);

create table Reservation(
Reservation_no int primary key,
R_intime datetime not null,
R_outtime datetime,
Amount int not null check(Amount>0),
R_no int references Room(R_no),
C_ID int references Customer(C_ID)

);
CREATE TABLE Customer(
C_Id int primary key,
C_Name Varchar(50) Not Null,
Reservation_no int,
C_Age int ,
C_Address Varchar(70) Not Null,
C_contact int,
C_cin_time int,
C_cout_t int,
foreign key(Reservation_no) references Reservation(Reservation_no)
);

alter table hotel rename hotel_info;
```

```
insert into hotel_info values("marriot",1234,"Pune",3456,5);
insert into hotel_info values("The Plaza",2345,"New York ",4567,7);
insert into hotel_info values("Claridge's",3456,"London",5678,7);
insert into hotel_info values("Raffles",5678,"Singapore",6789,8);
insert into hotel_info values("Taj Mahal Palace",6789,"Mumbai ",7890 ,9);
insert into hotel_info values("Beverly Hills Hotel",8970,"Los
Angeles",8907,2);

insert into employee values("Adwait
Purao","Permanent",1,1234,"Purao","Adwait","Kurla","Mumbai",12345 ,10
000);

insert into employee values("Ram
Kumar","Permanent",2,1234,"Kumar","Ram","Kalina","Mumbai",12346,20
000);

insert into employee values("Akshay
Kumar","Temporary",3,3456,"Kumar","Akshay","Ram
chowk","Ramgad",12347,30000);

insert into employee values("Ranbir
Kapoor","Permanent",4,2345,"Kapoor","Ranbir","Roopnagar","Agra",1234
8,40000);

insert into employee values("Angelina
Jolie","Permanent",5,8970,"Jolie","Angelina","Beverly Hills","Los
Angeles",12349,50000);

alter table customer modify C_cin_time time ;
alter table customer modify C_cout_t time ;
alter table reservation modify R_intime time ;
alter table reservation modify R_outtime time ;

insert into reservation values(1,"12:56:23","16:56:23",1000,12,1234);
insert into reservation values(2,"13:54:43","19:26:13",2000,13,1235);
insert into reservation values(3,"11:24:41","20:55:53",1500,14,1236);
insert into reservation values(4,"22:21:45","16:25:33",2500,15,1237);

insert into customer values(1234,"Sam
```

```
Vaz",1,34,"Ghatkopar",123456,"12:56:23","16:56:23");

insert into customer values(1235,"Ram
Sharma",2,44,"Ghansoli",123457,"13:54:43","19:26:13");

insert into customer values(1236,"Sachin
Tendulkar",3,50,"Colaba",123458,"11:24:41","20:55:53");

insert into customer values(1237,"Virat
Kohli",4,30,"Dadar",123459,"22:21:45","16:25:33");

insert into room values(12,1,1000,"Basic",1234);
insert into room values(13,0,2000,"Deluxe",2345);
insert into room values(14,1,1500,"Suite",5678);
insert into room values(15,0,2500," Luxury Suite",6789);

select * from hotel_info;
select * from employee;
select * from room;
select * from reservation;
select * from customer;

use hotel;
alter table info_hotel
rename to hotel_info;

insert into employee values("Ranbir
Kapoor","Permanent",4,2345,"Kapoor","Ranbir","Roopnagar","Agra",1234
8,40000);

alter table customer
add R_no int;

use hotel;
alter table customer add constraint foreign key(r_no) references
room(r_no);

use hotel;
alter table reservation add constraint foreign key(c_id) references
customer(c_id);
```

```
alter table room
add Reservation_no int;

use hotel;
alter table room add constraint foreign key(reservation_no) references
reservation(reservation_no);

update room set Reservation_no=1 where r_no=12;
update room set Reservation_no=2 where r_no=13;
update room set Reservation_no=3 where r_no=14;
update room set Reservation_no=4 where r_no=15;

update customer set r_no=12 where reservation_no=1;
update customer set r_no=13 where reservation_no=2;
update customer set r_no=14 where reservation_no=3;
update customer set r_no=15 where reservation_no=4;

update customer
set C_Name="Ramesh Verma",C_Address="Dharavi"
where C_ID=1234;
```

Original tables

1)Table hotel_info

Result Grid | Filter Rows: | Edit:

H_Name	H_ID	H_Address	H_Num_Emp	H_vacancies
marriot	1234	Pune	3456	5
The Plaza	2345	New York	4567	7
Claridge's	3456	London	5678	7
Raffles	5678	Singapore	6789	8
Taj Mahal Palace	6789	Mumbai	7890	9
Beverly Hills Hotel	8970	Los Angeles	8907	2
NULL	NULL	NULL	NULL	NULL

2)Table Employee

E_Name	E_Type	E_ID	H_ID	LastName	FirstName	Address	City	E_Contact	E_Salary
Adwait Purao	Permanent	1	1234	Purao	Adwait	Kurla	Mumbai	12345	10000
Akshay Kumar	Temporary	3	3456	Kumar	Akshay	Ram chowk	Ramgad	12347	30000
Ranbir Kapoor	Permanent	4	2345	Kapoor	Ranbir	Roopnagar	Agra	12348	40000
Angelina Jolie	Permanent	5	8970	Jolie	Angelina	Beverly Hills	Los Angeles	12349	50000

3)Table Room

R_no	R_vacany	R_price	R_type	H_ID	Reservation_no
12	1	1000	Basic	1234	1
13	0	2000	Deluxe	2345	2
14	1	1500	Suite	5678	3
15	0	2500	Luxury Suite	6789	4

4)Table Reservation

Reservation_no	R_intime	R_outtime	Amount	R_no	C_ID
1	12:56:23	16:56:23	1000	12	1234
2	13:54:43	19:26:13	2000	13	1235
3	11:24:41	20:55:53	1500	14	1236
4	22:21:45	16:25:33	2500	15	1237

5)Table Customer

C_Id	C_Name	Reservation_no	C_Age	C_Address	C_contact	C_cin_time	C_cout_t	R_no
1234	Ramesh Verma	1	34	Dharavi	123456	12:56:23	16:56:23	12
1235	Ram Sharma	2	44	Ghansoli	123457	13:54:43	19:26:13	13
1236	Sachin Tendulkar	3	50	Colaba	123458	11:24:41	20:55:53	14
1237	Virat Kohli	4	30	Dadar	123459	22:21:45	16:25:33	15

Queries:

1)Relation Customer and Reservation
1)Left Join

Statement:

We will be using left join to show a resultant table which contains all the

matched value of both tables and the unmatched values from the left table that is customer table

Code:

```
SELECT customer.c_name, reservation.r_no
FROM customer
LEFT JOIN reservation
ON customer.c_id=reservation.c_id
ORDER BY customer.c_name;
```

Result Grid | Filter R

c_name	r_no
Ram Sharma	13
Ramesh Verma	12
Sachin Tendulkar	14
Virat Kohli	15

2)Right Join

Statement:

We will be using right join to show a resultant table which contains all the matched value of both tables and the unmatched values from the right table that is reservation table

Code:

```
SELECT customer.c_name, reservation.r_no,reservation.amount
FROM customer
Right JOIN reservation
ON customer.c_id=reservation.c_id
ORDER BY customer.c_name;
```

Result Grid | Filter Rows:

c_name	r_no	amount
Ram Sharma	13	2000
Ramesh Verma	12	1000
Sachin Tendulkar	14	1500
Virat Kohli	15	2500

3)Natural Join

Statement:

A NATURAL JOIN compares all columns of two tables which have the same column-name and the resulting joined table contains those columns once which are same in name in both the tables.

Code:

```
SELECT *
FROM customer
NATURAL JOIN reservation;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

C_Id	Reservation_no	C_Name	C_Age	C_Address	C_contact	C_cin_time	C_cout_t	R_intime	R_outtime	Amount	R_no
1234	1	Ramesh Verma	34	Dharavi	123456	12:56:23	16:56:23	12:56:23	16:56:23	1000	12
1235	2	Ram Sharma	44	Ghansoli	123457	13:54:43	19:26:13	13:54:43	19:26:13	2000	13
1236	3	Sachin Tendulkar	50	Colaba	123458	11:24:41	20:55:53	11:24:41	20:55:53	1500	14
1237	4	Virat Kohli	30	Dadar	123459	22:21:45	16:25:33	22:21:45	16:25:33	2500	15

4)Cross Join

Statement:

We will be using cross join to show a resultant table which contains all the matched value of both table and where c_id is same from customer table and reservation table.

Code:

```
SELECT customer.c_name, reservation.r_no,C_CONTACT
FROM customer
CROSS JOIN reservation
ON customer.c_id=reservation.c_id
```

Result Grid | Filter Rows:

c_name	r_no	C_CONTACT
Ramesh Verma	12	123456
Ram Sharma	13	123457
Sachin Tendulkar	14	123458
Virat Kohli	15	123459

5)Inner Join

Statement:

We will be using inner join to show a resultant table which contains all the matched value .

Code:

```
SELECT *
FROM customer
Inner JOIN reservation;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

R_no	R_vacany	R_price	R_type	H_ID	Reservation_no	R_intime	R_outtime	Amount	R_no	C_ID
15	0	2500	Luxury Suite	6789	1	12:56:23	16:56:23	1000	12	1234
14	1	1500	Suite	5678	1	12:56:23	16:56:23	1000	12	1234
13	0	2000	Deluxe	2345	1	12:56:23	16:56:23	1000	12	1234
12	1	1000	Basic	1234	1	12:56:23	16:56:23	1000	12	1234
15	0	2500	Luxury Suite	6789	2	13:54:43	19:26:13	2000	13	1235
14	1	1500	Suite	5678	2	13:54:43	19:26:13	2000	13	1235
13	0	2000	Deluxe	2345	2	13:54:43	19:26:13	2000	13	1235
12	1	1000	Basic	1234	2	13:54:43	19:26:13	2000	13	1235
15	0	2500	Luxury Suite	6789	3	11:24:41	20:55:53	1500	14	1236
14	1	1500	Suite	5678	3	11:24:41	20:55:53	1500	14	1236
13	0	2000	Deluxe	2345	3	11:24:41	20:55:53	1500	14	1236
12	1	1000	Basic	1234	3	11:24:41	20:55:53	1500	14	1236
15	0	2500	Luxury Suite	6789	4	22:21:45	16:25:33	2500	15	1237
14	1	1500	Suite	5678	4	22:21:45	16:25:33	2500	15	1237
13	0	2000	Deluxe	2345	4	22:21:45	16:25:33	2500	15	1237
12	1	1000	Basic	1234	4	22:21:45	16:25:33	2500	15	1237

2) Relation Employee and Hotel

1)Left Join

Statement:

We will be using left join to show a resultant table which contains all the matched value of both tables and the unmatched values from the left table that is employee table

Code:

```
use hotel;
SELECT employee.e_name, hotel_info.h_name
FROM employee
LEFT JOIN Hotel_info
ON employee.h_id=Hotel_info.h_id
```

Result Grid | Filter Rows:

e_name	h_name
Adwait Purao	marriot
Akshay Kumar	Claridge's
Ranbir Kapoor	The Plaza
Angelina Jolie	Beverly Hills Hotel

2)Right Join

Statement:

We will be using right join to show a resultant table which contains all the matched value of both tables and the unmatched values from the right table that is hotel_info table

Code:

```
use hotel;
SELECT employee.e_name, hotel_info.h_name,Hotel_info.h_address
FROM employee
RIGHT JOIN Hotel_info
ON employee.h_id=Hotel_info.h_id
```

Result Grid | Filter Rows:

e_name	h_name	h_address
Adwait Purao	marriot	Pune
Ranbir Kapoor	The Plaza	New York
Akshay Kumar	Claridge's	London
NULL	Raffles	Singapore
NULL	Taj Mahal Palace	Mumbai
Angelina Jolie	Beverly Hills Hotel	Los Angeles

3)Natural Join

Statement:

A NATURAL JOIN compares all columns of two tables which have the same column-name and the resulting joined table contains those columns once which are same in name in both the tables.

Code:

```
SELECT *
FROM employee
NATURAL JOIN hotel_info;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

H_ID	E_Name	E_Type	E_ID	LastName	FirstName	Address	City	E_Contact	E_Salary	H_Name	H_Address	H_Num_Emp	H_vacancies
1234	Adwait Purao	Permanent	1	Purao	Adwait	Kurla	Mumbai	12345	10000	marriot	Pune	3456	5
3456	Akshay Kumar	Temporary	3	Kumar	Akshay	Ramgad	Bihar	12347	30000	Claridge's	London	5678	7
2345	Ranbir Kapoor	Permanent	4	Kapoor	Ranbir	Roopnagar	Agra	12348	40000	The Plaza	New York	4567	7
8970	Angelina Jolie	Permanent	5	Jolie	Angelina	Beverly Hills	Los Angeles	12349	50000	Beverly Hills Hotel	Los Angeles	8907	2

4)Inner Join

Statement:

We will be using inner join to show a resultant table which contains all the matched value where the condition is specified which is those rows having same h_id in both employee table and hotel_info.

Code:

```
SELECT *
FROM employee
Inner JOIN hotel_info
on employee.h_id=hotel_info.h_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

E_Name	E_Type	E_ID	H_ID	LastName	FirstName	Address	City	E_Contact	E_Salary	H_Name	H_ID	H_Address	H_Num_Emp	H_vacancies
Adwait Purao	Permanent	1	1234	Purao	Adwait	Kurla	Mumbai	12345	10000	marriot	1234	Pune	3456	5
Akshay Kumar	Temporary	3	3456	Kumar	Akshay	Ramgad	Bihar	12347	30000	Claridge's	3456	London	5678	7
Ranbir Kapoor	Permanent	4	2345	Kapoor	Ranbir	Roopnagar	Agra	12348	40000	The Plaza	2345	New York	4567	7
Angelina Jolie	Permanent	5	8970	Jolie	Angelina	Beverly Hills	Los Angeles	12349	50000	Beverly Hills Hotel	8970	Los Angeles	8907	2

5)Cross Join

Statement:

We will be using cross join to show a resultant table which contains all the matched value of both table and where h_id is same from employee table and hotel_info table.

Code:

```
use hotel;
SELECT hotel_info.h_name, employee.e_name,employee.e_id
FROM hotel_info
CROSS JOIN employee
ON employee.h_id=hotel_info.h_id
```

Result Grid | Filter Rows:

h_name	e_name	e_id
marriot	Adwait Purao	1
Claridge's	Akshay Kumar	3
The Plaza	Ranbir Kapoor	4
Beverly Hills Hotel	Angelina Jolie	5

3)Relation Room and Hotel

1)Left join

Statement:

We will be using left join to show a resultant table which contains all the matched value of both tables and the unmatched values from the left table that is room table

Code:

```
use hotel;
SELECT room.r_no, hotel_info.h_name,room.r_vacany
FROM room
LEFT JOIN Hotel_info
ON room.h_id=Hotel_info.h_id
```

Result Grid | Filter Rows:

r_no	h_name	r_vacany
12	marriot	1
13	The Plaza	0
14	Raffles	1
15	Taj Mahal Palace	0

2)Right join

Statement:

We will be using right join to show a resultant table which contains all the matched value of both tables and the unmatched values from the right table that is Hotel_info table

Code:

```
use hotel;
SELECT room.r_no, hotel_info.h_name,room.r_vacany
FROM room
RIGHT JOIN Hotel_info
ON room.h_id=Hotel_info.h_id
```

Result Grid | Filter Rows:

r_no	h_name	r_vacany
12	marriot	1
13	The Plaza	0
NULL	Claridge's	NULL
14	Raffles	1
15	Taj Mahal Palace	0
NULL	Beverly Hills Hotel	NULL

3)Natural join

Statement:

A NATURAL JOIN compares all columns of two tables which have the same column-name and the resulting joined table contains those columns once which are same in name in both the tables.

Code:

```
use hotel;
SELECT *
FROM room
NATURAL JOIN hotel_info;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

H_ID	R_no	R_vacany	R_price	R_type	H_Name	H_Address	H_Num_Emp	H_vacancies
1234	12	1	1000	Basic	marriot	Pune	3456	5
2345	13	0	2000	Deluxe	The Plaza	New York	4567	7
5678	14	1	1500	Suite	Raffles	Singapore	6789	8
6789	15	0	2500	Luxury Suite	Taj Mahal Palace	Mumbai	7890	9

4)Inner join

Statement:

We will be using inner join to show a resultant table which contains all the matched value where the condition is specified which is those rows having

same h_id in both room table and hotel_info table.

Code:

```
use hotel;
SELECT *
FROM room
Inner JOIN hotel_info
on hotel_info.h_id=room.h_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

R_no	R_vacany	R_price	R_type	H_ID	H_Name	H_ID	H_Address	H_Num_Emp	H_vacancies
12	1	1000	Basic	1234	marriot	1234	Pune	3456	5
13	0	2000	Deluxe	2345	The Plaza	2345	New York	4567	7
14	1	1500	Suite	5678	Raffles	5678	Singapore	6789	8
15	0	2500	Luxury Suite	6789	Taj Mahal Palace	6789	Mumbai	7890	9

5)Cross join

Statement:

We will be using cross join to show a resultant table which contains all the matched value of both table and where h_id is same from room table and hotel_info table.

Code:

```
use hotel;
SELECT room.r_no, hotel_info.h_name,room.r_vacany
FROM room
CROSS JOIN Hotel_info
ON room.h_id=Hotel_info.h_id
```

Result Grid | Filter Rows:

r_no	h_name	r_vacany
12	marriot	1
13	The Plaza	0
14	Raffles	1
15	Taj Mahal Palace	0

4)Relation Room and Customer

1)Left Join

Statement:

We will be using left join to show a resultant table which contains all the matched value of both tables and the unmatched values from the left table that is customer table

Code:

```
use hotel;
SELECT customer.c_name, room.r_no,room.r_type
FROM customer
LEFT JOIN room
ON customer.r_no=room.r_no;
```

Result Grid | Filter Rows:

c_name	r_no	r_type
Ramesh Verma	12	Basic
Ram Sharma	13	Deluxe
Sachin Tendulkar	14	Suite
Virat Kohli	15	Luxury Suite

2)Right Join

Statement:

We will be using right join to show a resultant table which contains all the matched value of both tables and the unmatched values from the right table that is room table

Code:

```
use hotel;
SELECT customer.c_name, room.r_no,room.r_type
FROM customer
Right JOIN room
ON customer.r_no=room.r_no;
```

Result Grid | Filter Rows:

c_name	r_no	r_type	r_price
Ramesh Verma	12	Basic	1000
Ram Sharma	13	Deluxe	2000
Sachin Tendulkar	14	Suite	1500
Virat Kohli	15	Luxury Suite	2500

3)Inner join

Statement:

We will be using inner join to show a resultant table which contains all the matched value where the condition is specified which is those rows having same r_no in both room table and customer table.

Code:

```
use hotel;
SELECT *
FROM room
Inner JOIN customer
ON customer.r_no=room.r_no;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

R_no	R_vacany	R_price	R_type	H_ID	C_Id	C_Name	Reservation_no	C_Age	C_Address	C_contact	C_cin_time	C_cout_t	r_no
12	1	1000	Basic	1234	1234	Ramesh Verma	1	34	Dharavi	123456	12:56:23	16:56:23	12
13	0	2000	Deluxe	2345	1235	Ram Sharma	2	44	Ghansoli	123457	13:54:43	19:26:13	13
14	1	1500	Suite	5678	1236	Sachin Tendulkar	3	50	Colaba	123458	11:24:41	20:55:53	14
15	0	2500	Luxury Suite	6789	1237	Virat Kohli	4	30	Dadar	123459	22:21:45	16:25:33	15

4)Natural join

Statement:

A NATURAL JOIN compares all columns of two tables which have the same column-name and the resulting joined table contains those columns once which are same in name in both the tables.

Code:

```
use hotel;
SELECT *
FROM room
NATURAL JOIN customer;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

R_no	R_vacany	R_price	R_type	H_ID	C_Id	C_Name	Reservation_no	C_Age	C_Address	C_contact	C_cin_time	C_cout_t
12	1	1000	Basic	1234	1234	Ramesh Verma	1	34	Dharavi	123456	12:56:23	16:56:23
13	0	2000	Deluxe	2345	1235	Ram Sharma	2	44	Ghansoli	123457	13:54:43	19:26:13
14	1	1500	Suite	5678	1236	Sachin Tendulkar	3	50	Colaba	123458	11:24:41	20:55:53
15	0	2500	Luxury Suite	6789	1237	Virat Kohli	4	30	Dadar	123459	22:21:45	16:25:33

5)Cross join

Statement:

We will be using cross join to show a resultant table which contains all the matched value of both table and where r_no is same from customer table and room table.

Code:

```
use hotel;
SELECT customer.c_name, room.r_no,room.r_type
FROM customer
Cross JOIN room
ON customer.r_no=room.r_no;
```

Result Grid | Filter Rows:

c_name	r_no	r_type
Ramesh Verma	12	Basic
Ram Sharma	13	Deluxe
Sachin Tendulkar	14	Suite
Virat Kohli	15	Luxury Suite

5)Relation Room and Reservation

1)Inner join

Statement:

We will be using inner join to show a resultant table which contains all the matched value where the condition is specified which is those rows having same reservation_no in both room table and reservation table.

Code:

```
use hotel;
select room.reservation_no,room.r_type,reservation.c_id
from room
inner join reservation
on room.reservation_no=reservation.reservation_no;
```

Result Grid | Filter Rows:

reservation_no	r_type	c_id
1	Basic	1234
2	Deluxe	1235
3	Suite	1236
4	Luxury Suite	1237

2)Left join

Statement:

We will be using left join to show a resultant table which contains all the matched value of both tables and the unmatched values from the left table that is room table

Code:

```
use hotel;
select room.reservation_no,room.r_type,reservation.c_id,room.h_id
from room
left join reservation
on room.reservation_no=reservation.reservation_no;
```

Result Grid | Filter Rows:

reservation_no	r_type	c_id	h_id
1	Basic	1234	1234
2	Deluxe	1235	2345
3	Suite	1236	5678
4	Luxury Suite	1237	6789

3)Right join

Statement:

We will be using right join to show a resultant table which contains all the matched value of both tables and the unmatched values from the right table that is room table

Code:

```
use hotel;
```

select
room.reservation_no,room.r_type,reservation.c_id,room.h_id,reservation.amount
from room
right join reservation
on room.reservation_no=reservation.reservation_no;

Result Grid | Filter Rows: | Export:

reservation_no	r_type	c_id	h_id	amount
1	Basic	1234	1234	1000
2	Deluxe	1235	2345	2000
3	Suite	1236	5678	1500
4	Luxury Suite	1237	6789	2500

4)Cross join

Statement:

We will be using cross join to show a resultant table which contains all the matched value of both table and where reservation_no is same from room table and reservation table.

Code:

use hotel;
select room.reservation_no,room.r_type,room.h_id,reservation.amount
from room
cross join reservation
on room.reservation_no=reservation.reservation_no;

Result Grid | Filter Rows:

reservation_no	r_type	h_id	amount
1	Basic	1234	1000
2	Deluxe	2345	2000
3	Suite	5678	1500
4	Luxury Suite	6789	2500

5)Natural join

Statement:

A NATURAL JOIN compares all columns of two tables which have the same column-name and the resulting joined table contains those columns

once which are same in name in both the tables.

Code:

```
use hotel;
select *
from room
natural join reservation;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

R_no	Reservation_no	R_vacany	R_price	R_type	H_ID	R_intime	R_outtime	Amount	C_ID
12	1	1	1000	Basic	1234	12:56:23	16:56:23	1000	1234
13	2	0	2000	Deluxe	2345	13:54:43	19:26:13	2000	1235
14	3	1	1500	Suite	5678	11:24:41	20:55:53	1500	1236
15	4	0	2500	Luxury Suite	6789	22:21:45	16:25:33	2500	1237

Conclusion

We learned about various types of joins in SQL . We learned about inner join, left join, right join and cross join in this experiment. This experiment helped us to understand the use of joins in SQL and how we can achieve our desired output. We understood how the joins show us the relation between two tables which are linked by a foreign key.