|  |  |
| --- | --- |
| **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;     1. **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;     1. **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;     1. **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 ,10000);  insert into employee values("Ram Kumar","Permanent",2,1234,"Kumar","Ram","Kalina","Mumbai",12346,20000);  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",12348,40000);  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",12348,40000);  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**    **2)Table Employee**    **3)Table Room**    **4)Table Reservation**    **5)Table Customer**    **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;    **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;    **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;    **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    **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;    **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    **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    **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;    **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;    **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    **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    **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    **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;    **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;    **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    **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;    **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;    **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;    **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;    **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;    **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;    **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;    **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;    **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;    **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; |
| **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.** | |