

# DBMS-1

AI20BTECH11006

## Assignment-1

Question-1)

$$\pi_{sname}(\pi_{sid}((\pi_{bid}\sigma_{color='red'}Boats) \bowtie Reserves) \bowtie Sailors)$$

Answer-1)

**The query finds names of sailors who are sailing red boat**

We shall begin deciphering from the innermost parenthesis.

$$(\pi_{bid}\sigma_{color='red'}Boats)$$

Select the bid attribute from the Boats table where the color of the boat is red.

In SQL: `select bid from boats where color='red';`

$$(\pi_{bid}\sigma_{color='red'}Boats) \bowtie Reserves$$

We have to take natural join of the set returned from previous query with the Reserves table, the common column here being bid

$$\pi_{sid}((\pi_{bid}\sigma_{color='red'}Boats) \bowtie Reserves)$$

Here, we are selecting only the sid attribute from the set that we got in the previous step. (i.e natural join of 1st query and reserves table)

In SQL: `select sid from ((select bid from (boats natural join reserves) where color='red'))j1 ;`

or: `select sid from (select bid from boats where color='red')j natural join reserves;`

$$\pi_{sname}(\pi_{sid}((\pi_{bid}\sigma_{color='red'}Boats) \bowtie Reserves) \bowtie Sailors)$$

Select sname attribute from the set retrieved from previous query(nested) and natural join it with Sailors.

In SQL: `select distinct * from ((select sid from ((select bid from boats where color='red')B natural join reserves))R natural join sailors) ;`

To get distinct sname

Question-2)

$$2. \rho(Tempboats, (\sigma_{color='red' \vee color='green'} Boats)) \\ \pi_{sname}(Tempboats \bowtie Reserves \bowtie Sailors)$$

Answer-2)

The query finds the names of sailors who are either sailing red or green boat

$$(\sigma_{color='red' \vee color='green'} Boats)$$

This selects the rows which have either red or green color from the table Boats.

$$\rho(Tempboats, (\sigma_{color='red' \vee color='green'} Boats))$$

It selects all entries from boats whose color is either red or green, and then renames the table to Tempboats because rho is used to rename the table.

In SQL: `create table Tempboats as ( select * from boats where color = 'green' or color = 'red');`

$$\pi_{sname}(Tempboats \bowtie Reserves \bowtie Sailors)$$

This selects sname column from natural join of Tempboats, Reserves, Sailors, because we are constructing natural join of Tempboats which we created in the previous query, Reserves and Sailors.

In SQL: `select distinct * from (Tempboats natural join reserves natural join sailors);`

Question-3)

$$\rho(Temp1, \pi_{sid}((\sigma_{color='red'} Boats) \bowtie Reserves)) \\ \rho(Temp2, \pi_{sid}((\sigma_{color='green'} Boats) \bowtie Reserves)) \\ \pi_{sname}((Temp1 \cap Temp2) \bowtie Sailors)$$

Answer-3)

The query finds the names of those sailors who are sailing both red and green boats.

$$\pi_{sid}((\sigma_{color='red'} Boats) \bowtie Reserves)$$

Here, we are first selecting only the rows where color is red and then constructing natural join with reserves and then only selecting sid column.

$$\rho(Temp1, \pi_{sid}((\sigma_{color='red'} Boats) \bowtie Reserves))$$

Now, we are just calling the output of the previous query as Temp1., since rho is the rename operator.

In SQL: `create table Temp1 as (select sid from (select * from boats where color='red')) natural join reserves);`

$$\rho(Temp2, \pi_{sid}((\sigma_{color='green'} Boats) \bowtie Reserves))$$

It's essentially the same as above in Temp1, except the color is green, everything else is same

In SQL: `create table Temp2 as (select sid from (select * from boats where color='green')) natural join reserves);`

$$(Temp1 \cap Temp2)$$

Here, we are taking the intersection of Temp1 and Temp2

$$((Temp1 \cap Temp2) \bowtie Sailors)$$

Here, we are taking the natural join of the intersection of Temp1 and Temp2 and Sailors.

$$\pi_{sname}((Temp1 \cap Temp2) \bowtie Sailors)$$

Here, we are finding the intersection of rows in Temp1, and Temp2 and then constructing natural join with sailors and then selecting sname.

In SQL: `select distinct sname from (select T1.sid from Temp1 as T1, Temp2 as T2 where T1.sid = T2.sid) T1 natural join sailors;`

#### Question-4)

$$4. \rho(Reservations, \pi_{sid, sname, bid}(Sailors \bowtie Reserves))$$

$$\rho(ReservationPairs(1 \rightarrow sid1, 2 \rightarrow sname1, 3 \rightarrow bid1, 4 \rightarrow sid2, 5 \rightarrow sname2, 6 \rightarrow bid2), Reservations \times Reservations)$$

$$\pi_{sname1} \sigma_{(sid1=sid2) \wedge (bid1 \neq bid2)} ReservationPairs$$

Hint: On step 2, the tuples output from step 1, are being renamed “attribute/field wise”.

Answer-4)

The query outputs names of all the sailors who have made reservations on more than one boat.

$$\rho(\text{Reservations}, \pi_{sid, sname, bid}(\text{Sailors} \bowtie \text{Reserves}))$$

Here, we are forming a set Reservations and the attributes in it are sid, sname, bid.

$$\rho(\text{Sailors} \bowtie \text{Reserves})$$

Here, we are forming a natural join of Sailors and Reserves.

$$\pi_{sid, sname, bid}(\text{Sailors} \bowtie \text{Reserves})$$

Here we select sid, sname, bid from the natural join of Sailors and Reserves.

The SQL query is below

```
create table Reservations as (select sid, sname, bid from sailors natural join reserves);
```

$$\rho(\text{ReservationPairs}(1 \rightarrow sid1, 2 \rightarrow sname1, 3 \rightarrow bid1, 4 \rightarrow sid2, 5 \rightarrow sname2, 6 \rightarrow bid2), \text{Reservations} \times \text{Reservations})$$

Here, we are taking cartesian product with self and then renaming the columns as sid1, sname1, bid1 where these columns belong to the Reservations set written on left and sid2, sname2, bid2 belong to the Reservations written on right.

Here is the SQL query

```
create table ReservationPairs as (select r1.sid as sid1, r1.sname as sname1, r1.bid as bid1, r2.sid as sid2, r2.sname as sname2, r2.bid as bid2 from Reservations as r1 cross join Reservations as r2);
```

$$\pi_{sname1} \sigma_{(sid1=sid2) \wedge (bid1 \neq bid2)} \text{ReservationPairs}$$

It returns the name of sailors who have reservations on more than one boats, because we have sid1 = sid2 which means that we are looking at the same sailor because sid is expected to be unique, now if bid1 and bid2 are different, that means that the boats are different, so we are selecting the rows in which there are distinct boats but same sailor, after that we are selecting only sname1, which is the name of sailor, it will be same as sname2 since only 1 name should be associated with one sid.

MySQL query: 

```
select distinct sname1 from ReservationPairs where sid1=sid2 and bid1 != bid2;
```

Question-5)

$$5. \pi_{sid}(\sigma_{age > 20} \text{Sailors}) - \pi_{sid}((\sigma_{color='red'} \text{Boats}) \bowtie \text{Reserves} \bowtie \text{Sailors})$$

Answer-5)

The query attempts to find the sid of those sailors who have age more than 20 and are not sailing red boat.

$$\pi_{sid}(\sigma_{age>20}Sailors)$$

This selects the sid of those sailors whose age is greater than 20

In SQL: `select sid from sailors where age>20`

$$\pi_{sid}((\sigma_{color='red'}Boats) \bowtie Reserves \bowtie Sailors)$$

Here we are selecting the sid of those sailors who are sailing red colored boat. The step by step explanation is as follows

$$(\sigma_{color='red'}Boats)$$

First we select the rows in Boats where color is red.

$$(\sigma_{color='red'}Boats) \bowtie Reserves$$

Now, we construct natural join of this with reserves, this already has column sid.

$$\pi_{sid}((\sigma_{color='red'}Boats) \bowtie Reserves \bowtie Sailors)$$

This step becomes unnecessary because constructing natural join with Sailors doesn't help us in any form as we are going to filter out only sid out of this set anyways, and reserves already has sid.

Here is the MySQL query

```
select sid from (select * from boats where color='red')r natural join reserves natural join sailors;
```

$$\pi_{sid}(\sigma_{age>20}Sailors) - \pi_{sid}((\sigma_{color='red'}Boats) \bowtie Reserves \bowtie Sailors)$$

Now, we will take set difference, this means that we are removing all those sailors who sail red boat from the set of sailors who are older than 20 years. So, finally we get the sailors who are over the age of 20 and have reserved a boat that is not red in color.

Here is the MySQL query

```
select sid from sailors where age>20 and sid not in (select sid from (select * from boats where color='red')r natural join reserves natural join sailors);
```