Lab 5 - SQL – Joins: Inner, Outer; Sub queries: Correlated and Uncorrelated

What is a 'Join' Clause?

Join Clause is used to combine two or more tables based on a related column between them

Joins are of two types; Inner Join and Outer Join. By default, the word 'Join' refers to Inner Join. Outer Join is specifically referred to as 'Outer Join'

An INNER JOIN will keep only the information from both tables that's related to each other An Outer Join, on the other hand, will also keep information that is not related to the other table in the resulting table.

There are two types of 'Outer Join': Left Outer Join and Right Outer Join

Join Clause Syntax

```
SELECT *
FROM <table1> INNER JOIN <table2>
ON <condition>
```

Some of the SQL formats also allow the following syntax (Use JOIN instead of INNER JOIN)

```
SELECT *
FROM <table1> JOIN <table2>
ON <condition>
```

Interestingly, Join operation can also be done without using a Join Clause as below. JOIN is replaced by a comma (,)

```
SELECT *
FROM <table1>, <table2>
WHERE <condition>
```

It's possible to use more than two tables in a JOIN operation as below

```
SELECT *
FROM ((<Table 1>
INNER JOIN <Table 2> ON <Condition1>)
INNER JOIN <Table 3> ON <Condition2>);
```

1) INNER JOIN

Before we start this exercise let's insert a few more rows into Customer table

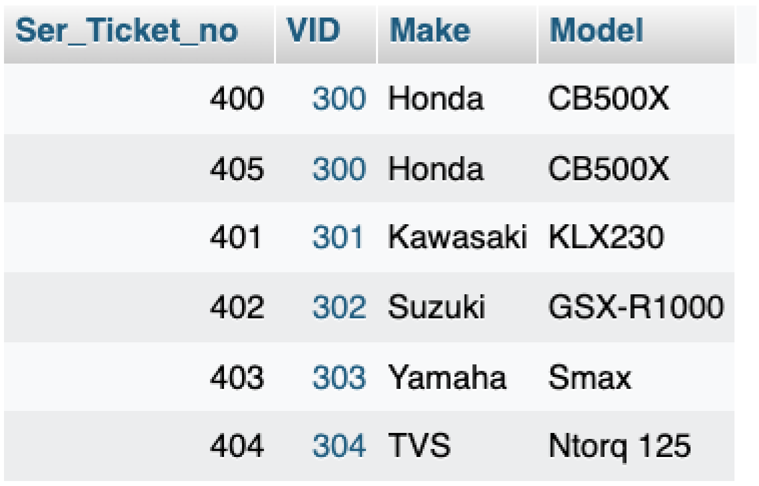
Insert into Customer values
(206,'Savi','Shenoy','Goa','savishenoy@gmail.com',7788993344),
(207,'Appu','Kali','Banaras','appuk@gmail.com',8877995566);

Let's delete one row from Generate_Invoice Table

```
DELETE FROM `Generate_invoice` WHERE Cust_ID = 202;
```

1.a) Retrieve make and model of all bikes that got serviced.

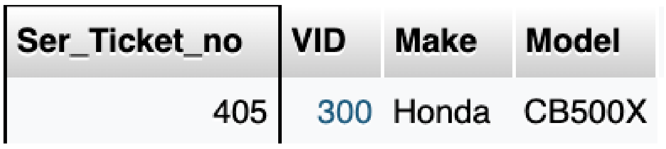
```
SELECT Ser_Ticket_no,Service_Ticket.VID, Make,Model FROM Bike JOIN Service_Ticket ON
Bike.Vid =Service_Ticket.VID;
```



Ser_Ticket_no	VID	Make	Model
400	300	Honda	CB500X
405	300	Honda	CB500X
401	301	Kawasaki	KLX230
402	302	Suzuki	GSX-R1000
403	303	Yamaha	Smax
404	304	TVS	Ntorq 125

1.b) To get information about the bike that got serviced through a particular service ticket (405), add another condition as below

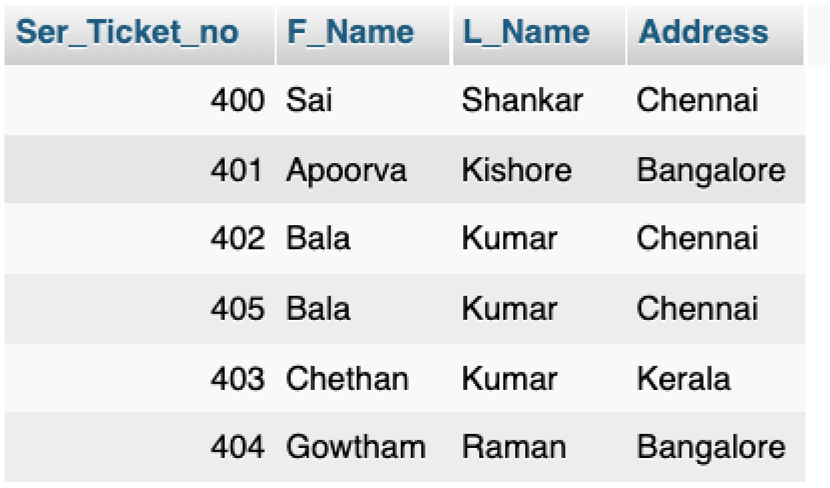
```
SELECT Ser_Ticket_no,Service_Ticket.VID, Make,Model FROM Bike JOIN Service_Ticket ON
Bike.Vid =Service_Ticket.VID WHERE Service_Ticket.Ser_Ticket_no = 405;
```



Ser_Ticket_no	VID	Make	Model
405	300	Honda	CB500X

2. a)Retrieve the customer's details of the bikes serviced (Equijoin)

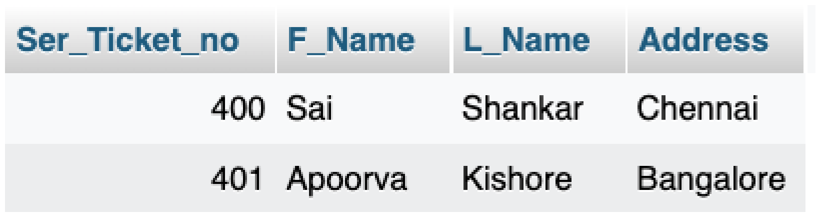
```
SELECT Ser_Ticket_no,F_Name,L_Name, Address
FROM Customer JOIN Service_Ticket ON Customer.Cust_id = Service_Ticket.CID
```



Ser_Ticket_no	F_Name	L_Name	Address
400	Sai	Shankar	Chennai
401	Apoorva	Kishore	Bangalore
402	Bala	Kumar	Chennai
405	Bala	Kumar	Chennai
403	Chethan	Kumar	Kerala
404	Gowtham	Raman	Bangalore

2,b)
Retrieve the customer details of the bikes serviced on a particular date (2022-01-24)

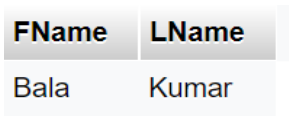
```
SELECT  Ser_Ticket_no,F_Name,L_Name,  Address  FROM  Customer  JOIN  Service_Ticket  ON
Customer.Cust_id = Service_Ticket.CID WHERE Service_Ticket.Date_del = '2022-01-24';
```



Ser_Ticket_no	F_Name	L_Name	Address
400	Sai	Shankar	Chennai
401	Apoorva	Kishore	Bangalore

3. Retrieve the customer names whose total bill is greater than 1,50,000. (Theta Join)

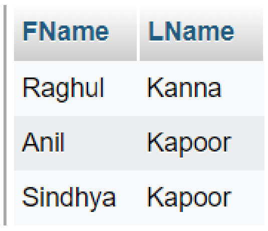
```
SELECT Customer.F_Name, Customer.L_Name
FROM Customer JOIN Generate_Invoice ON Customer.Cust_Id = Generate_Invoice.Cust_id
WHERE Generate_Invoice.Amount > 150000;
```



FName	LName
Bala	Kumar

4. Retrieve the names of the salespersons who are under dealers 1, 15, 19
A select statement of the syntax shown below can also be used for typical inner join

```
SELECT F_Name, L_Name FROM Salesperson INNER JOIN Dealer ON Salesperson.Did =
Dealer.Did WHERE Dealer.Did IN (1, 15, 19);
```



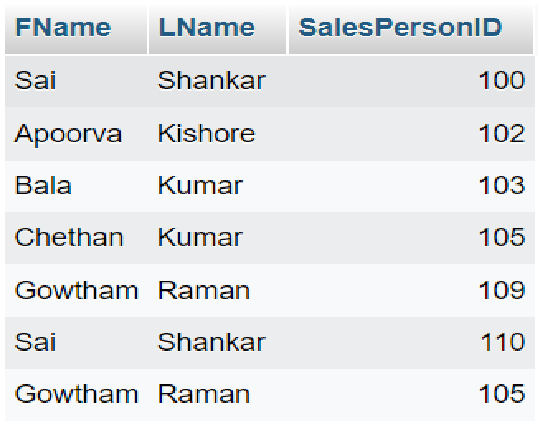
FName	LName
Raghul	Kanna
Anil	Kapoor
Sindhya	Kapoor

Note:
Alternately inner join can also be done without using the join keyword by using a condition in the WHERE clause.
For the above example the syntax can also be:

```
SELECT F_Name, L_Name
FROM Salesperson, Dealer
WHERE Salesperson.Did = Dealer.Did and Dealer.Did IN (1, 15, 19);
```

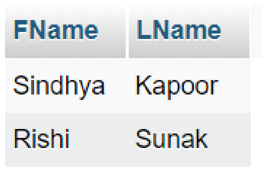
NATURAL JOIN

5. Retrieve the first name, last name of both customer and salesperson who attended them.
SELECT c.F_Name,c.L_Name,i.S_emp_id
FROM Customer AS c NATURAL JOIN Generate_Invoice AS i



FName	LName	SalesPersonID
Sai	Shankar	100
Apoorva	Kishore	102
Bala	Kumar	103
Chethan	Kumar	105
Gowtham	Raman	109
Sai	Shankar	110
Gowtham	Raman	105

6. Retrieve the First name and the last name of the salesperson who served the customer with id 205.
SELECT s.FName, s.LName
FROM Generate_Invoice AS i NATURAL JOIN Salesperson AS s
WHERE i.Cust_id=205;



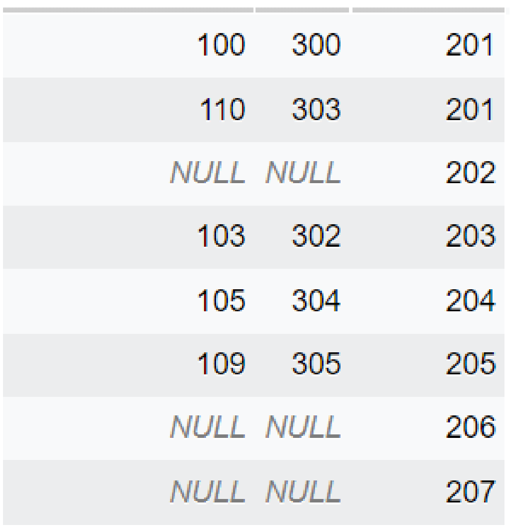
FName	LName
Sindhya	Kapoor
Rishi	Sunak

RIGHT OUTER JOIN

7. a)

Retrieve the Vehicle ID and the Customer ID of the vehicles that were not bought from any of the dealers (Invoice was not generated)

```
SELECT Generate_invoice.S_emp_id,VID,Customer.Cust_id
FROM Generate_invoice RIGHT OUTER JOIN Customer ON Customer.Cust_id =
Generate_invoice.Cust_Id;
```



100	300	201
110	303	201
NULL	NULL	202
103	302	203
105	304	204
109	305	205
NULL	NULL	206
NULL	NULL	207

Note

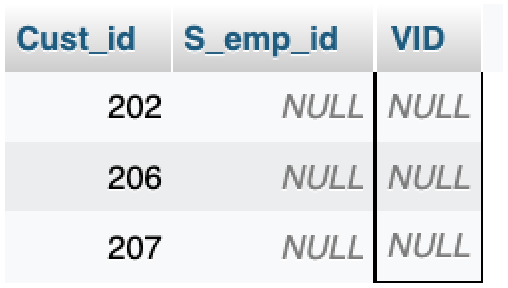
NULL values for S_emp_id and VID implies the corresponding customer never bought any vehicle.

The order of the columns can be changed through the select statement.

7.b)

The following command generates only those tuples corresponding to such customer

```
SELECT Customer.Cust_id, Generate_invoice.S_emp_id,VID FROM Generate_invoice RIGHT
OUTER JOIN Customer ON Customer.Cust_id = Generate_invoice.Cust_Id Where VID is NULL;
```

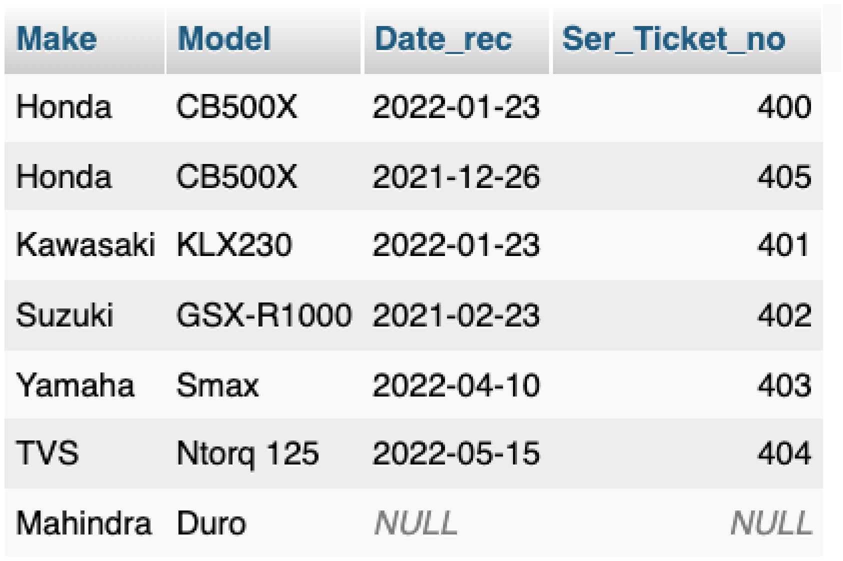


Cust_id	S_emp_id	VID
202	NULL	NULL
206	NULL	NULL
207	NULL	NULL

LEFT OUTER JOIN

8. For all Bikes, list the make, model, and also Date_rec , Ser_ticket_no (if given for service). Notice the different syntax used.

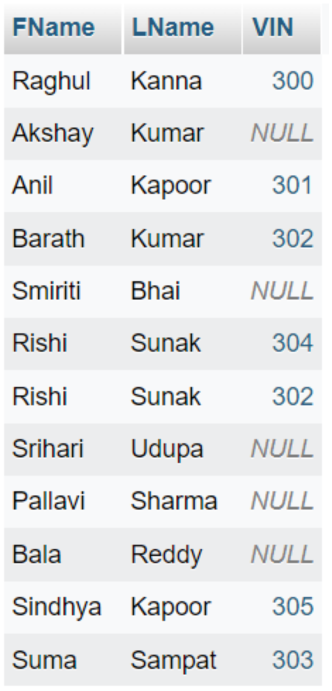
SELECT b.Make,b.Model,s.Date_rec,s.Ser_Ticket_no FROM Bike AS b LEFT OUTER JOIN Service_Ticket AS s ON b.Vid=s.VID;



Make	Model	Date_rec	Ser_Ticket_no
Honda	CB500X	2022-01-23	400
Honda	CB500X	2021-12-26	405
Kawasaki	KLX230	2022-01-23	401
Suzuki	GSX-R1000	2021-02-23	402
Yamaha	Smax	2022-04-10	403
TVS	Ntorq 125	2022-05-15	404
Mahindra	Duro	NULL	NULL

9. For all sales persons list the First Name, Last Name and also the VID if he has made any sales

SELECT s.F_Name,s.L_Name,i.VID
FROM Salesperson AS s LEFT OUTER JOIN Generate_Invoice AS i ON s.S_emp_id=i.S_emp_id;



FName	LName	VIN
Raghul	Kanna	300
Akshay	Kumar	NULL
Anil	Kapoor	301
Barath	Kumar	302
Smiriti	Bhai	NULL
Rishi	Sunak	304
Rishi	Sunak	302
Srihari	Udupa	NULL
Pallavi	Sharma	NULL
Bala	Reddy	NULL
Sindhya	Kapoor	305
Suma	Sampat	303

Note:
Alternately, left outer join can also be done without using the join keyword as below

```
select s.F_Name, s.L_Name,i.VID from Salesperson as s,Generate_invoice as i where
s.S_emp_id=i.S_emp_id
union all
select s.F_Name, s.L_Name,null from Salesperson as s where not exists ( select * from
Generate_invoice WHERE Generate_invoice.S_emp_id=s.S_emp_id );
```

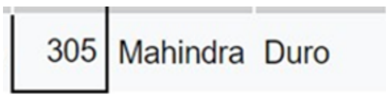
Note that the second part of the Union fetches only those rows where the condition is not true.

FULL OUTER JOIN
MySQL does not support FULL OUTER JOIN. Full Outer Joins are implemented by the Union of Left Outer Join and Right Outer Join

NESTED QUERIES

10. Retrieve the details of the bikes which were not given for service. (Corelated nested query)

```
SELECT b.Vid, Make, Model
FROM Bike as b
WHERE b.Vid NOT IN (SELECT Bike.Vid
                    FROM Bike JOIN Service_Ticket ON Bike.Vid = Service_Ticket.VID);
```



```
305 Mahindra Duro
```

RAILWAY RESERVATION

1.Update price of the ticket.

Hint:

This requires creation of two views.

One for calculating the price of one ticket for a given PNR. This involves computation of distance traveled and fare per kilometer.

view1(PNR,Train_no, Departure,Arrival, Distance, FareperKM)

Other one calculating the number of passengers traveling in the ticket corresponding to that PNR

view2(PNR, Passenger_number)

Using the above views, ticket price is updated as (Distance x FareperKM x Passenger_number)

NATURAL JOIN

2. Retrieve the all stations along route of the Trains along with the distance between the stations

INNER JOIN (equijoin)

3.Retrieve the Train no of train which is leaving Bengaluru and arriving at Chennai with compartments availability greater than 10

4.Retrieve first and last name of users who have booked a ticket with price greater than 500

LEFT OUTER JOIN

5. Retrieve the first name, last name, DOB and ticket PNR if they've bought it for all users.

6. Retrieve the first name, last name,of the Users who have not bought a ticket.

RIGHT OUTER JOIN

7. Retrieve the ticket PNR, Train number, travel date and along with all users first name and last name.

8. Retrieve the user id if they've traveled in a train along with train id and name of all trains.

NESTED QUERIES

9. Retrieve the train no and name of trains whose destination is not Mangaluru and distance is not less than 100km and departure time is not 8:30:00 PM.

10. Retrieve the User ID who has spent more ticket price than the average ticket price.

Submission:
Please submit a zip file containing the following files

.pdf file containing the screenshots, named SRN_lab05.pdf

.sql containing the sql script named SRN_lab05.sql