Lab 6 – Aggregate Function

Demo Lab for E-Bike

1] Find the Average **Price of the Parts**.

SELECT AVG(Price) FROM Parts;

AVG(Price)
2189.621667

2] Determine the salesperson ID who has Amount greater than average Amount.

SELECT i.salesperson_id FROM Generate_Invoice AS i WHERE i.total_bill > (SELECT AVG(i. total_bill) FROM Generate_Invoice AS i);

SalesPersonID
103

3] Find sum, maximum, minimum, average and count of **Total_Bill** in the table Generate_Invoice.

SELECT MAX(Total_Bill), MIN(Total_Bill), SUM(Total_Bill), AVG(Total_Bill), COUNT(Total_Bill) FROM Generate_Invoice;

MAX(Total_Bill)	MIN(Total_Bill)	SUM(Total_Bill)	AVG(Total_Bill)	COUNT(Total_Bill)
908907.87	78900.90	1450330.61	241721.768333	6

4] Retrieve all the details of parts which is having maximum price.

*SELECT * FROM parts WHERE price = (SELECT MAX(price) FROM parts);*

P_ID	Description	Qty	Price	Service_ID
705	Leg Guard	101	6217.99	403

5] Retrieve all details of the parts whose price is greater than the average price of parts in the table

*SELECT * FROM Parts WHERE Parts.price > (SELECT AVG(price) FROM Parts);*

p_id	description	qty	price	service_ID
703	Handle Bar	4	2519.99	405
705	Leg Guard	101	6217.99	403

6] Retrieve the count of the city of dealers.

SELECT COUNT(*), city FROM dealer GROUP BY city;

Cout(*)	City
10	Bangalore
5	Chennai
5	Mumbai

7] Display customer id and average amount for that customer whose bill count is more than one.
SELECT i.cust_id, AVG(total_bill) FROM Generate_Invoice i, Customer c WHERE i.cust_id=c.cust_id GROUP BY (c.cust_id) HAVING COUNT(*)>1;

cust_id	AVG(total_bill)
201	120544.935000

8] Retrieve model number, bike id, price of the bike whose service is not the first service.

SELECT b.vin, model FROM service_ticket s, bike b WHERE s.vin=b.vin GROUP BY vin HAVING COUNT(*)>1;

vin	model
300	CB500X

Railway Reservation System

Tasks:

1. Find the average distance between subsequent stations for every train
2. Find the average distance between subsequent stations for every train and display them in descending order of distance
3. Display the list of train numbers and the total distance traveled by each in descending order of the distance traveled

4. List those trains that have maximum and minimum number compartments and also display number of compartments they have. (2 queries one to find max and other to find min)
5. Display the number of phone numbers corresponding to the user_id(s) ADM_001, USR_006, USR_10
6. Find the average fare per km for each train type specified and display the train type and corresponding average fare per km as ‘Avg_Fare’ in decreasing order of Avg_Fare
7. Retrieve all details of the oldest passenger.
8. Count the number of passengers whose name consists of ‘Ullal’. (Hint: Use the LIKE operator)

Deliverables::

1. Create screenshots for each task, with the query and the result together. Paste all of them in a word file with appropriate labels (task1, task2 etc) . Convert the docx into a pdf file and submit.
2. A .sql file named as SRN_Lab6.sql with all the sql queries. (Separate each query by a commented line)