**Name: Kanak Agrawal**

**Day-6 Assignment**

**Question:** Over and Partition by clause in sql

**Answer:**

**Query:** SELECT

car\_make,

car\_model,

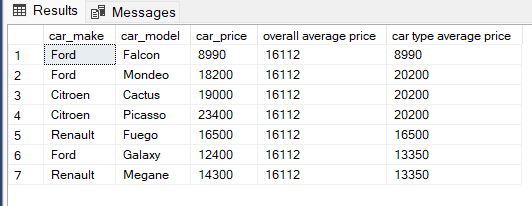
car\_price,

AVG(car\_price) OVER() AS "overall average price",

AVG(car\_price) OVER (PARTITION BY car\_type) AS "car type average price"

FROM car\_list\_price;

**Output:**

****

**Question:** Total Aggregation using over and partition by in sql Queries

**Answer:**

* **SUM**

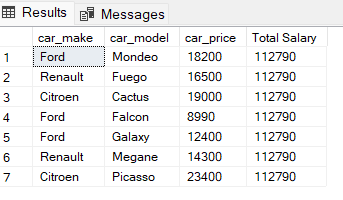
**Query:** select

car\_make, car\_model ,car\_price,

Sum(car\_price) over() As "Total Salary"

from car\_list\_price;

**Output:**

****

* **GROUP BY**

**Query:** select

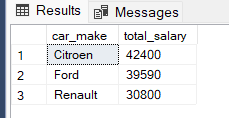
car\_make,

sum(car\_price) AS total\_salary

from car\_list\_price

group by car\_make;

**Output:**

****

* **AVERAGE**

**Query:** SELECT

car\_make,

car\_model,

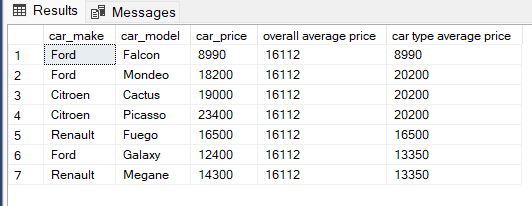
car\_price,

AVG(car\_price) OVER() AS "overall average price",

AVG(car\_price) OVER (PARTITION BY car\_type) AS "car type average price"

FROM car\_list\_price;

**Output:**

****

**Question:** snowflaking& Star schemas

**Answer:**

* **SNOWFLAKE**

**Query:** create database snowflake;

use snowflake;

create table salestable

(

product\_id int not null primary key,

order\_id int not null,

customer\_id int not null,

employeer\_id int not null,

total int not null ,

Quantity int not null,

discount int );

create table time\_dimension

(

order\_id int not null primary key,

order\_date date not null);

create table customer\_dimension

(customer\_id int not null primary key,

city\_id int not null,

customer\_name char(30) not null,

address varchar(50) not null,

city char(25) not null,

zip int not null);

create table product\_dimension

(

product\_id int not null primary key,

Product\_name varchar(50) not null ,

product\_prize decimal not null);

create table emp\_dimension

(employeer\_id int not null primary key,

emp\_name varchar(30) not null,

department varchar(25) not null,

department\_id int not null);

create table city\_dimension

(city\_id int not null primary key,

city\_name char(30) not null,

state char(25), country char(20)

);

create table Product\_category\_dimension

(product\_id int not null primary key,

name varchar(50) not null,

pro\_description varchar(50) not null,

unit\_prize int not null

FOREIGN KEY (product\_id) REFERENCES product\_dimension(product\_id));

create table department\_dimension

(department\_id int,

department varchar(25) not null,

location varchar(25) not null);

select \* from salestable;

select \* from time\_dimension;

select \* from customer\_dimension;

select \* from product\_dimension;

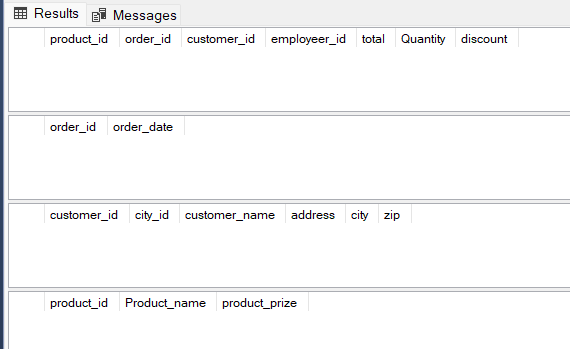
select \* from emp\_dimension;

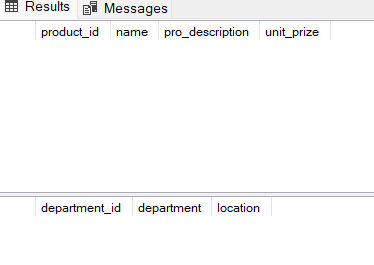
select \* from city\_dimension;

select \* from Product\_category\_dimension;

select \* from department\_dimension;

**Output:**

****

****

**Question:** How to calculate Subtotals in SQL Queries.

**Answer:**

**Query:** SELECT CASE

WHEN GROUPING(SalesQuartes)=1 AND GROUPING(SalesYear)=0

THEN 'SubTotal'

WHEN GROUPING(SalesQuartes)=1 AND GROUPING(SalesYear)=1

THEN 'Grand Total'

ELSE

CAST(SalesYear AS varchar(10))

END AS SalesYear,

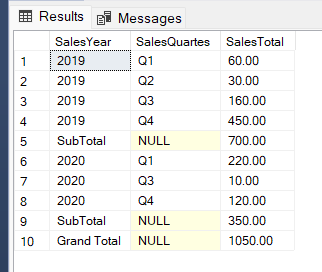
SalesQuartes,

SUM(SalesTotal) AS SalesTotal

FROM SalesList

GROUP BY ROLLUP(SalesYear,SalesQuartes);

**Output:**

****

**Question:** Differences Between UNION EXCEPT and INTERSECT Operators in SQL Server.

**Answer:**

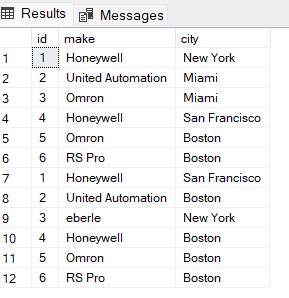
* **UNION**

**Query:** select \* from sensor\_1

UNION All

select \* from sensor\_2;

**Output:**

****

* **EXCEPT**

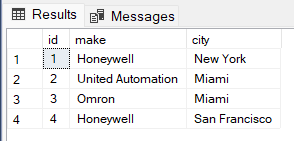
**Query:**

select \* from sensor\_1

Except

select \* from sensor\_2;

**Output:**

****

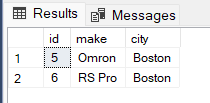
* **INTERSECT**

**Query:** select \* from sensor\_1

Intersect

select \* from sensor\_2;

**Output:**

****