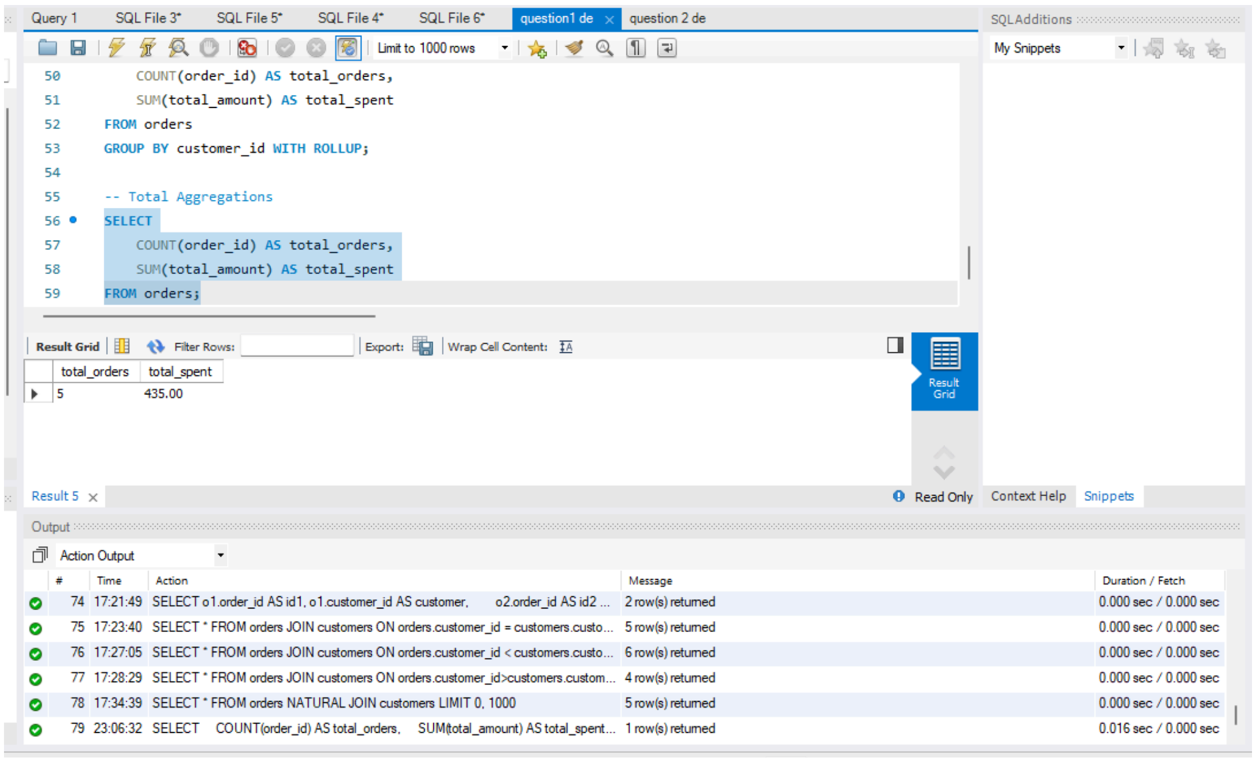
Asmita Porwal
Batch-1
Day-6
25/1/2023
Data engineering

Assignment-6

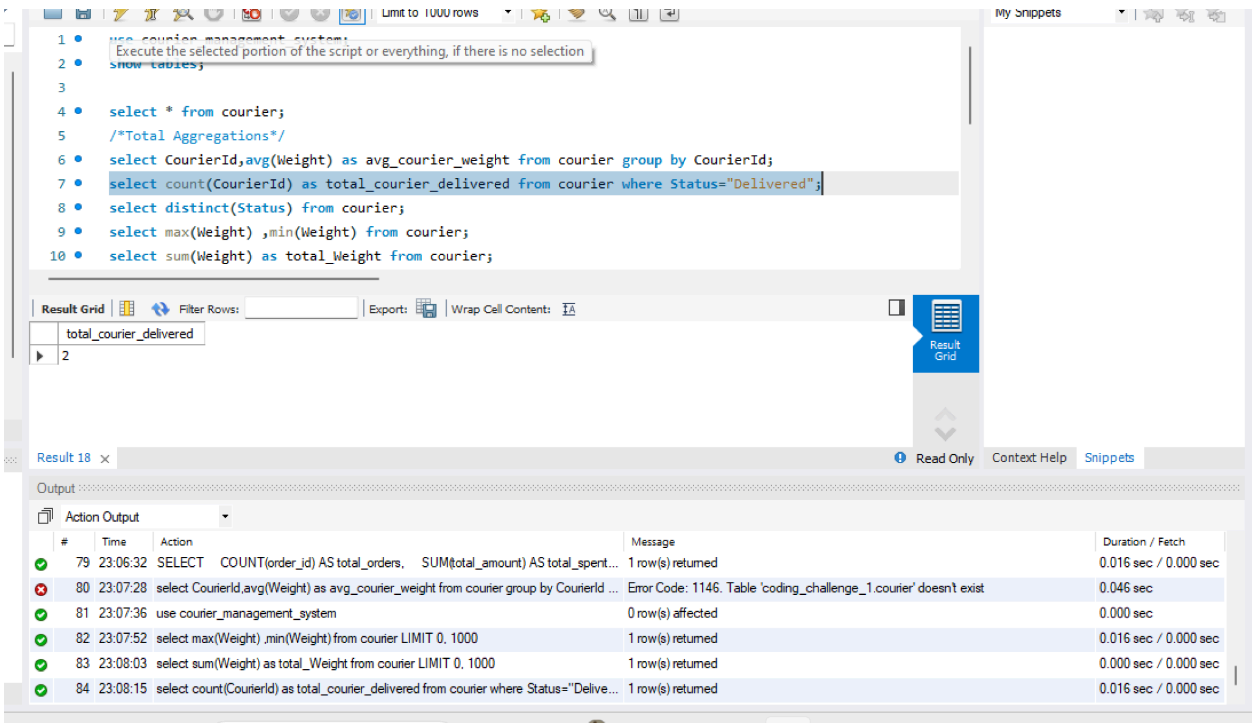
Total Aggregations using SQL Queries



```
    COUNT(order_id) AS total_orders,
    SUM(total_amount) AS total_spent
FROM orders
GROUP BY customer_id WITH ROLLUP;

-- Total Aggregations
SELECT
    COUNT(order_id) AS total_orders,
    SUM(total_amount) AS total_spent
FROM orders;
```

total_orders	total_spent
5	435.00

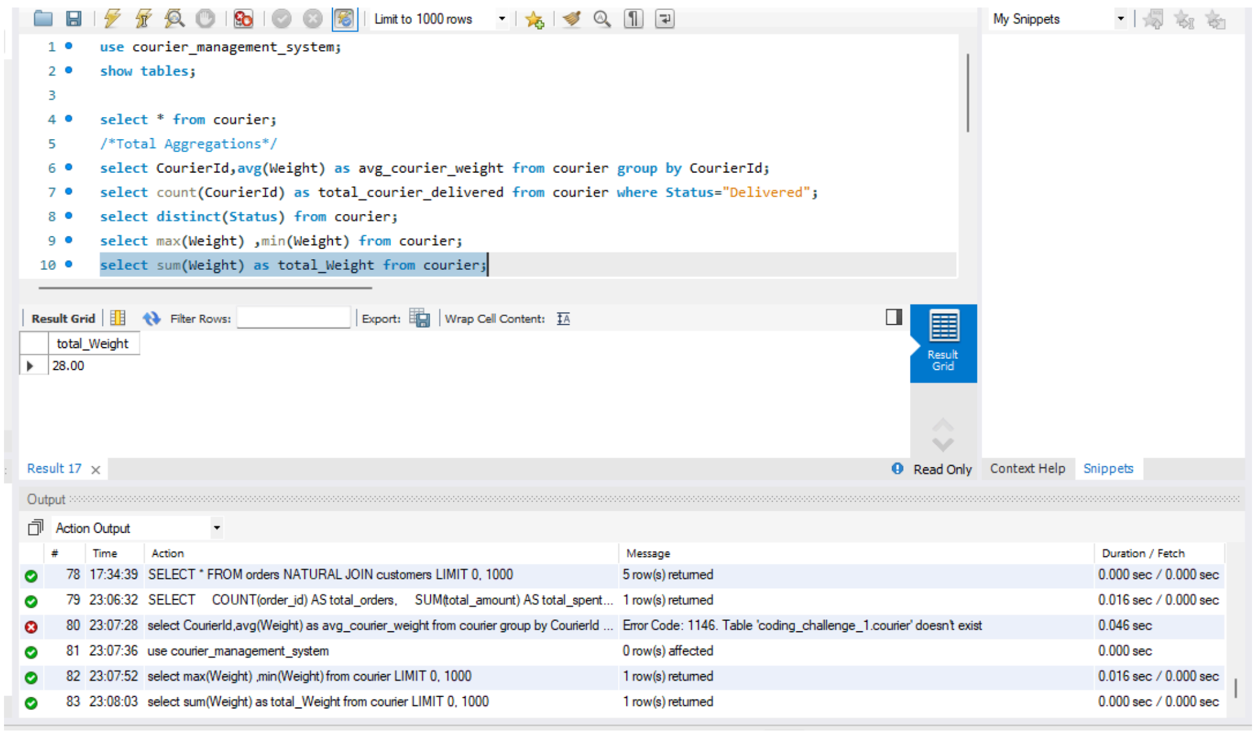
```
#   Time      Action                                                                     Message            Duration / Fetch
74  17:21:49  SELECT o1.order_id AS id1, o1.customer_id AS customer,    o2.order_id AS id2 ...   2 row(s) returned  0.000 sec / 0.000 sec
75  17:23:40  SELECT * FROM orders JOIN customers ON orders.customer_id = customers.custo...    5 row(s) returned  0.000 sec / 0.000 sec
76  17:27:05  SELECT * FROM orders JOIN customers ON orders.customer_id < customers.custo...    6 row(s) returned  0.000 sec / 0.000 sec
77  17:28:29  SELECT * FROM orders JOIN customers ON orders.customer_id>customers.custom...     4 row(s) returned  0.000 sec / 0.000 sec
78  17:34:39  SELECT * FROM orders NATURAL JOIN customers LIMIT 0, 1000                       5 row(s) returned  0.000 sec / 0.000 sec
79  23:06:32  SELECT    COUNT(order_id) AS total_orders,    SUM(total_amount) AS total_spent...  1 row(s) returned  0.016 sec / 0.000 sec
```



```sql
use courier_management_system;
show tables;

select * from courier;
/*Total Aggregations*/
select CourierId,avg(Weight) as avg_courier_weight from courier group by CourierId;
select count(CourierId) as total_courier_delivered from courier where Status="Delivered";
select distinct(Status) from courier;
select max(Weight) ,min(Weight) from courier;
select sum(Weight) as total_Weight from courier;
```

	total_courier_delivered
▶	2

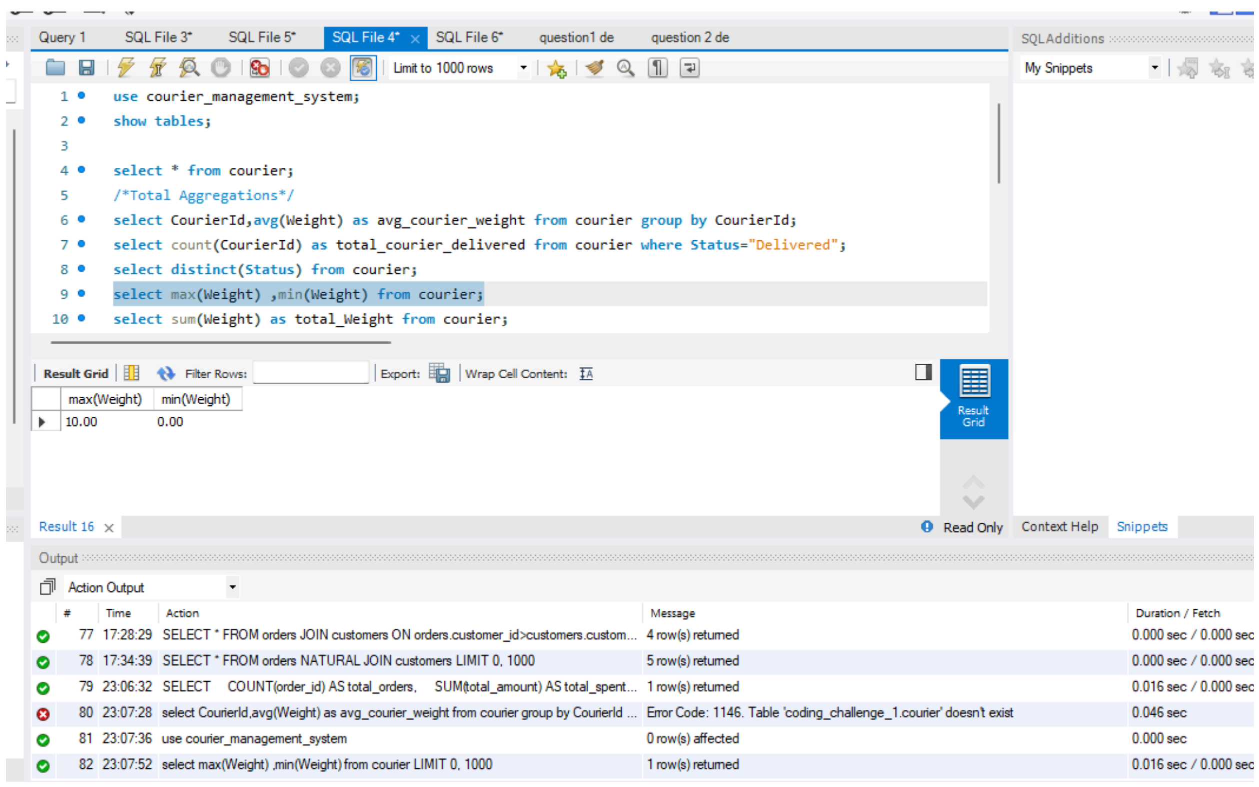
	#	Time	Action	Message	Duration / Fetch
✓	79	23:06:32	SELECT COUNT(order_id) AS total_orders, SUM(total_amount) AS total_spent...	1 row(s) returned	0.016 sec / 0.000 sec
✗	80	23:07:28	select CourierId,avg(Weight) as avg_courier_weight from courier group by CourierId ...	Error Code: 1146. Table 'coding_challenge_1.courier' doesn't exist	0.046 sec
✓	81	23:07:36	use courier_management_system	0 row(s) affected	0.000 sec
✓	82	23:07:52	select max(Weight) ,min(Weight) from courier LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec
✓	83	23:08:03	select sum(Weight) as total_Weight from courier LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
✓	84	23:08:15	select count(CourierId) as total_courier_delivered from courier where Status="Delive...	1 row(s) returned	0.016 sec / 0.000 sec



```sql
use courier_management_system;
show tables;

select * from courier;
/*Total Aggregations*/
select CourierId,avg(Weight) as avg_courier_weight from courier group by CourierId;
select count(CourierId) as total_courier_delivered from courier where Status="Delivered";
select distinct(Status) from courier;
select max(Weight) ,min(Weight) from courier;
select sum(Weight) as total_Weight from courier;
```

	total_Weight
▶	28.00

	#	Time	Action	Message	Duration / Fetch
✓	78	17:34:39	SELECT * FROM orders NATURAL JOIN customers LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
✓	79	23:06:32	SELECT COUNT(order_id) AS total_orders, SUM(total_amount) AS total_spent...	1 row(s) returned	0.016 sec / 0.000 sec
✗	80	23:07:28	select CourierId,avg(Weight) as avg_courier_weight from courier group by CourierId ...	Error Code: 1146. Table 'coding_challenge_1.courier' doesn't exist	0.046 sec
✓	81	23:07:36	use courier_management_system	0 row(s) affected	0.000 sec
✓	82	23:07:52	select max(Weight) ,min(Weight) from courier LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec
✓	83	23:08:03	select sum(Weight) as total_Weight from courier LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

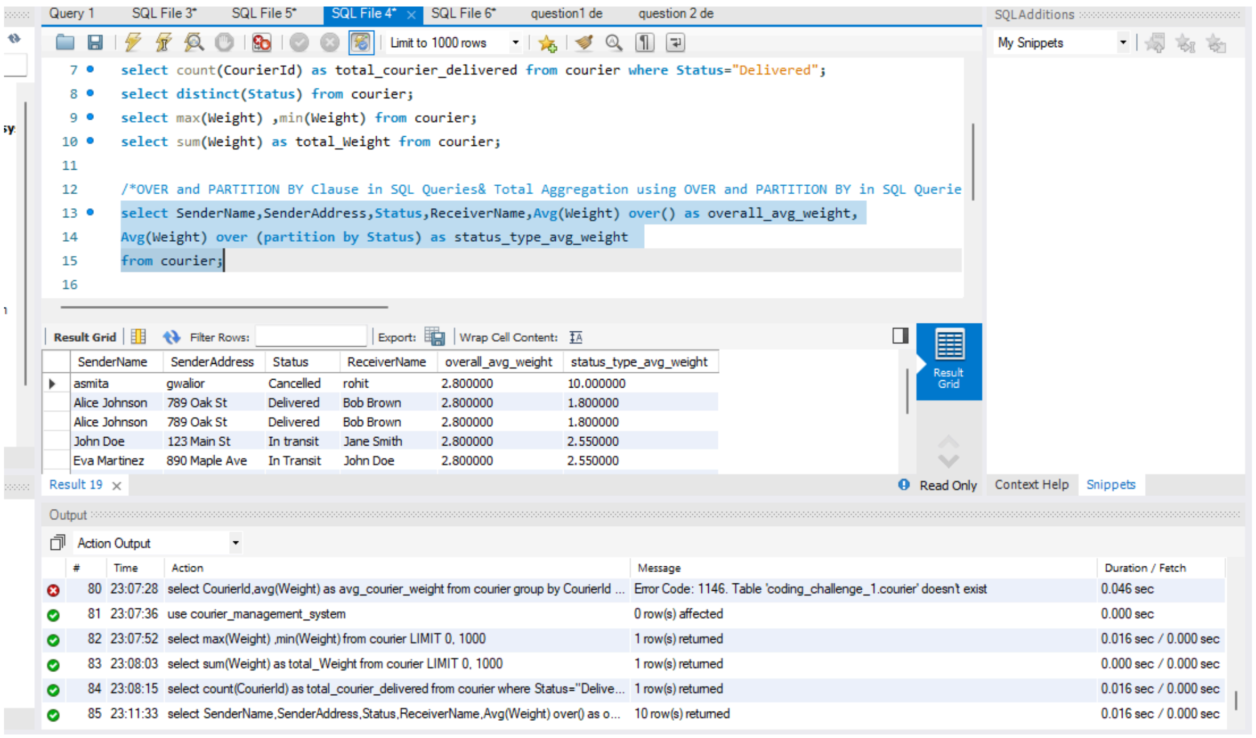


```
use courier_management_system;
show tables;

select * from courier;
/*Total Aggregations*/
select CourierId,avg(Weight) as avg_courier_weight from courier group by CourierId;
select count(CourierId) as total_courier_delivered from courier where Status="Delivered";
select distinct(Status) from courier;
select max(Weight) ,min(Weight) from courier;
select sum(Weight) as total_Weight from courier;
```

max(Weight)	min(Weight)
10.00	0.00

```
    #   Time      Action                                                                       Message                                                                  Duration / Fetch
✓   77  17:28:29  SELECT * FROM orders JOIN customers ON orders.customer_id>customers.custom...  4 row(s) returned                                                        0.000 sec / 0.000 sec
✓   78  17:34:39  SELECT * FROM orders NATURAL JOIN customers LIMIT 0, 1000                      5 row(s) returned                                                        0.000 sec / 0.000 sec
✓   79  23:06:32  SELECT    COUNT(order_id) AS total_orders,    SUM(total_amount) AS total_spent...  1 row(s) returned                                                    0.016 sec / 0.000 sec
✗   80  23:07:28  select CourierId,avg(Weight) as avg_courier_weight from courier group by CourierId ...  Error Code: 1146. Table 'coding_challenge_1.courier' doesn't exist  0.046 sec
✓   81  23:07:36  use courier_management_system                                                  0 row(s) affected                                                        0.000 sec
✓   82  23:07:52  select max(Weight) ,min(Weight) from courier LIMIT 0, 1000                     1 row(s) returned                                                        0.016 sec / 0.000 sec
```

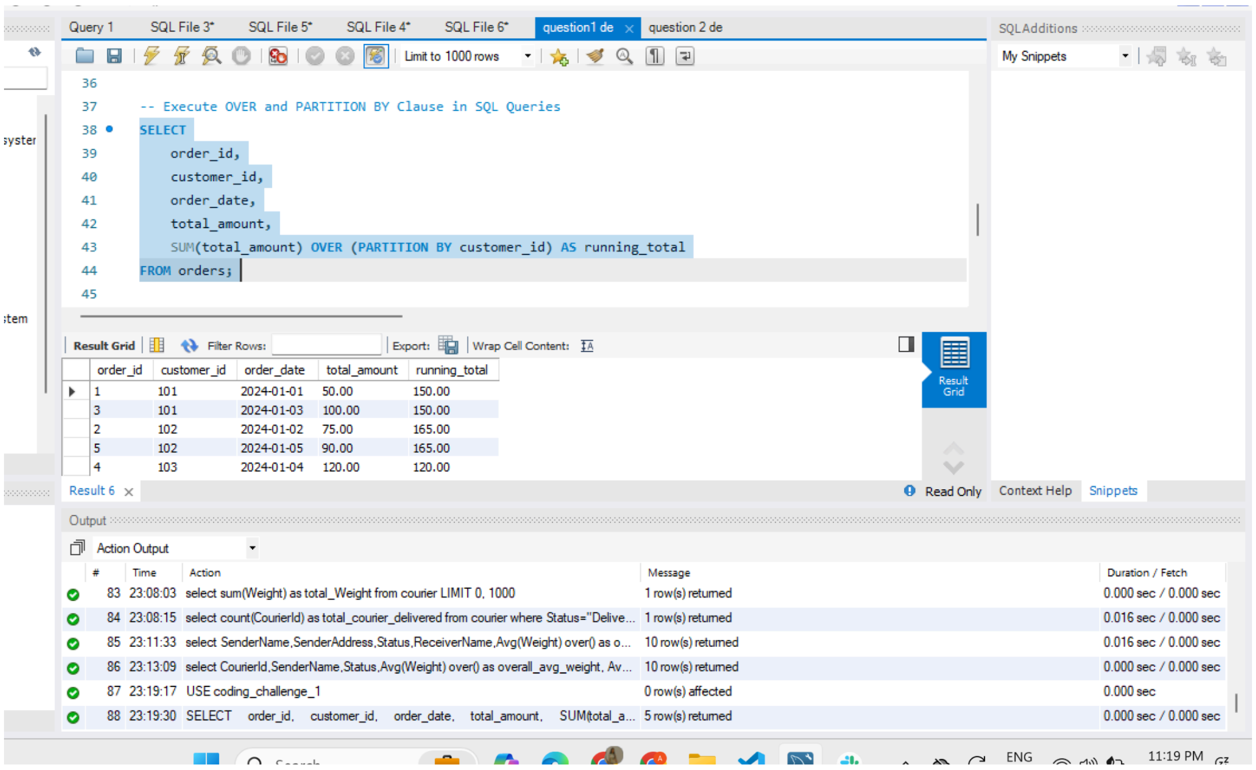
OVER and PARTITION BY Clause in SQL Queries & Total Aggregation using OVER and PARTITION BY in SQL Queries



```sql
select count(CourierId) as total_courier_delivered from courier where Status="Delivered";
select distinct(Status) from courier;
select max(Weight) ,min(Weight) from courier;
select sum(Weight) as total_Weight from courier;

/*OVER and PARTITION BY Clause in SQL Queries& Total Aggregation using OVER and PARTITION BY in SQL Querie
select SenderName,SenderAddress,Status,ReceiverName,Avg(Weight) over() as overall_avg_weight,
Avg(Weight) over (partition by Status) as status_type_avg_weight
from courier;
```

SenderName	SenderAddress	Status	ReceiverName	overall_avg_weight	status_type_avg_weight
asmita	gwalior	Cancelled	rohit	2.800000	10.000000
Alice Johnson	789 Oak St	Delivered	Bob Brown	2.800000	1.800000
Alice Johnson	789 Oak St	Delivered	Bob Brown	2.800000	1.800000
John Doe	123 Main St	In transit	Jane Smith	2.800000	2.550000
Eva Martinez	890 Maple Ave	In Transit	John Doe	2.800000	2.550000

```
#   Time      Action                                                                          Message                                                                 Duration / Fetch
80  23:07:28  select CourierId,avg(Weight) as avg_courier_weight from courier group by CourierId ...  Error Code: 1146. Table 'coding_challenge_1.courier' doesn't exist    0.046 sec
81  23:07:36  use courier_management_system                                                   0 row(s) affected                                                       0.000 sec
82  23:07:52  select max(Weight) ,min(Weight) from courier LIMIT 0, 1000                          1 row(s) returned                                                       0.016 sec / 0.000 sec
83  23:08:03  select sum(Weight) as total_Weight from courier LIMIT 0, 1000                       1 row(s) returned                                                       0.000 sec / 0.000 sec
84  23:08:15  select count(CourierId) as total_courier_delivered from courier where Status="Delive...  1 row(s) returned                                                  0.016 sec / 0.000 sec
85  23:11:33  select SenderName,SenderAddress,Status,ReceiverName,Avg(Weight) over() as o...      10 row(s) returned                                                      0.016 sec / 0.000 sec
```



```sql

-- Execute OVER and PARTITION BY Clause in SQL Queries
SELECT
    order_id,
    customer_id,
    order_date,
    total_amount,
    SUM(total_amount) OVER (PARTITION BY customer_id) AS running_total
FROM orders;
```

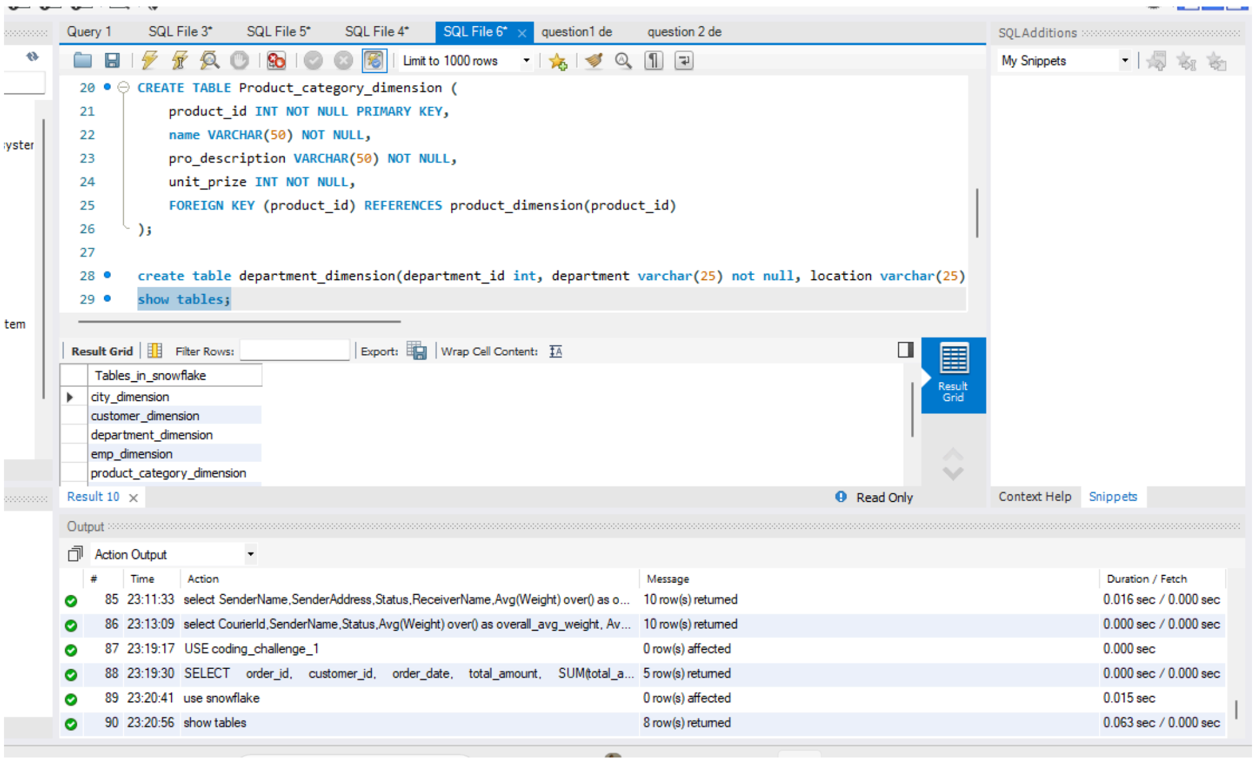
order_id	customer_id	order_date	total_amount	running_total
1	101	2024-01-01	50.00	150.00
3	101	2024-01-03	100.00	150.00
2	102	2024-01-02	75.00	165.00
5	102	2024-01-05	90.00	165.00
4	103	2024-01-04	120.00	120.00

```
#   Time      Action                                                                          Message              Duration / Fetch
83  23:08:03  select sum(Weight) as total_Weight from courier LIMIT 0, 1000                       1 row(s) returned    0.000 sec / 0.000 sec
84  23:08:15  select count(CourierId) as total_courier_delivered from courier where Status="Delive...  1 row(s) returned    0.016 sec / 0.000 sec
85  23:11:33  select SenderName,SenderAddress,Status,ReceiverName,Avg(Weight) over() as o...      10 row(s) returned   0.016 sec / 0.000 sec
86  23:13:09  select CourierId,SenderName,Status,Avg(Weight) over() as overall_avg_weight, Av...   10 row(s) returned   0.000 sec / 0.000 sec
87  23:19:17  USE coding_challenge_1                                                              0 row(s) affected    0.000 sec
88  23:19:30  SELECT   order_id,   customer_id,   order_date,   total_amount,   SUM(total_a...      5 row(s) returned    0.000 sec / 0.000 sec
```

snowflaking& Star schemas

```
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);
show tables;


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;



```
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)
show tables;
```

Tables_in_snowflake
city_dimension
customer_dimension
department_dimension
emp_dimension
product_category_dimension

#	Time	Action	Message	Duration / Fetch
85	23:11:33	select SenderName,SenderAddress,Status,ReceiverName,Avg(Weight) over() as o...	10 row(s) returned	0.016 sec / 0.000 sec
86	23:13:09	select CourierId,SenderName,Status,Avg(Weight) over() as overall_avg_weight, Av...	10 row(s) returned	0.000 sec / 0.000 sec
87	23:19:17	USE coding_challenge_1	0 row(s) affected	0.000 sec
88	23:19:30	SELECT order_id, customer_id, order_date, total_amount, SUM(total_a...	5 row(s) returned	0.000 sec / 0.000 sec
89	23:20:41	use snowflake	0 row(s) affected	0.015 sec
90	23:20:56	show tables	8 row(s) returned	0.063 sec / 0.000 sec

Rules and Restrictions to Group and Filter Data in SQL queries

GROUP BY Clause:

- Columns in the SELECT clause that are not part of an aggregate function must be included in the GROUP BY clause.
- You cannot use aliases in the GROUP BY clause; you must use the original column or expression.
- GROUP BY is usually used with aggregate functions like COUNT, SUM, AVG, MAX, or MIN.

HAVING Clause:

- The HAVING clause is used to filter the results of a GROUP BY clause based on specified conditions.
- It is similar to the WHERE clause but is used with aggregate functions.
- The HAVING clause must follow the GROUP BY clause.

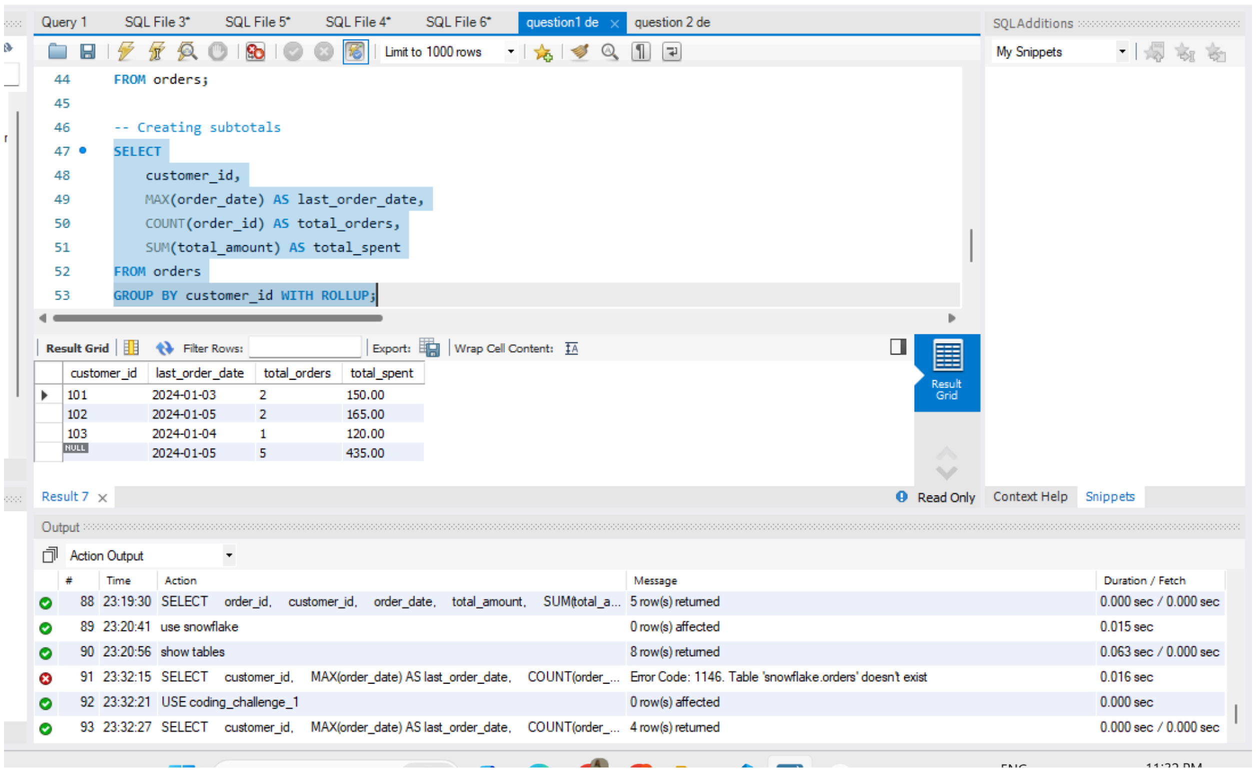
ORDER BY Clause:

- The ORDER BY clause is used to sort the result set.
- It can include column names, expressions, or positions of columns in the SELECT clause.
- Sorting can be done in ascending (ASC) or descending (DESC) order.

Order of Execution of SQL Queries:

FROM clause: Specifies the tables from which to retrieve the data.
WHERE clause: Filters the rows based on specified conditions.
GROUP BY clause: Groups rows that have the same values into summary rows.
HAVING clause: Filters groups based on specified conditions.
SELECT clause: Selects the columns to be included in the result set.
ORDER BY clause: Sorts the result set based on specified columns or expressions

How to calculate Subtotals in SQL Queries



```
FROM orders;

-- Creating subtotals
SELECT
    customer_id,
    MAX(order_date) AS last_order_date,
    COUNT(order_id) AS total_orders,
    SUM(total_amount) AS total_spent
FROM orders
GROUP BY customer_id WITH ROLLUP;
```

customer_id	last_order_date	total_orders	total_spent
101	2024-01-03	2	150.00
102	2024-01-05	2	165.00
103	2024-01-04	1	120.00
NULL	2024-01-05	5	435.00

```
#   Time      Action                                                              Message                                              Duration / Fetch
88  23:19:30  SELECT  order_id,  customer_id,  order_date,  total_amount,  SUM(total_a...  5 row(s) returned                                    0.000 sec / 0.000 sec
89  23:20:41  use snowflake                                                       0 row(s) affected                                    0.015 sec
90  23:20:56  show tables                                                         8 row(s) returned                                    0.063 sec / 0.000 sec
91  23:32:15  SELECT  customer_id,  MAX(order_date) AS last_order_date,  COUNT(order_...  Error Code: 1146. Table 'snowflake.orders' doesn't exist  0.016 sec
92  23:32:21  USE coding_challenge_1                                              0 row(s) affected                                    0.000 sec
93  23:32:27  SELECT  customer_id,  MAX(order_date) AS last_order_date,  COUNT(order_...  4 row(s) returned                                    0.000 sec / 0.000 sec
```

Differences Between UNION EXCEPT and INTERSECT Operators in SQL Server

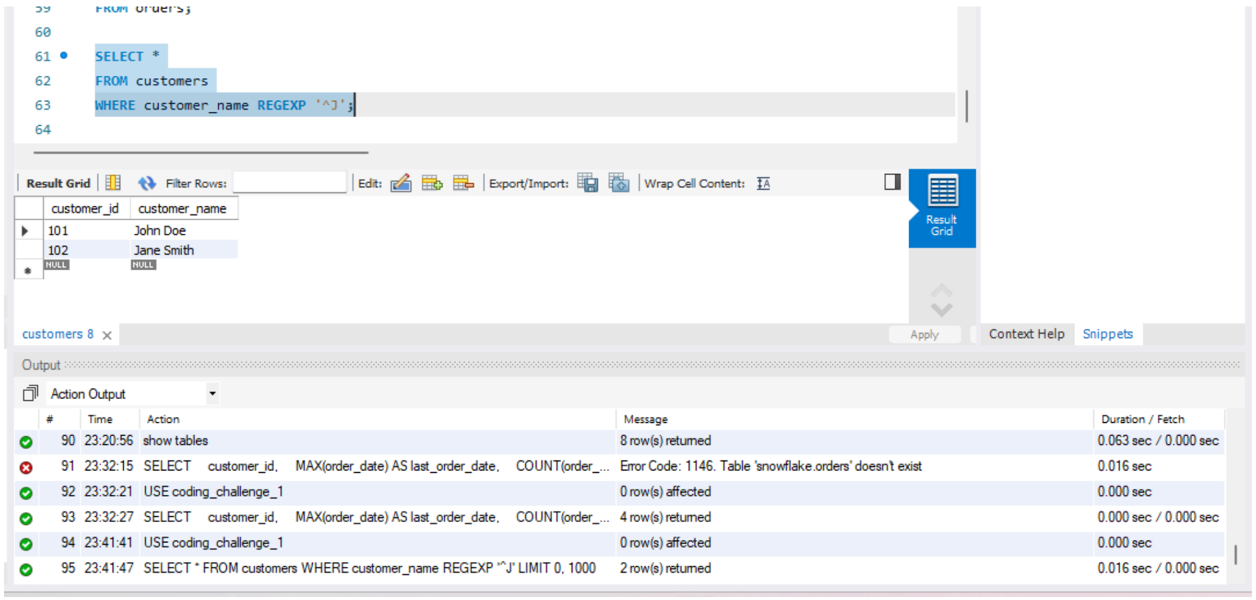
UNION:

- Combines the result sets of two or more SELECT statements.
- Removes duplicate rows from the combined result set.
- The number and order of columns must be the same in all SELECT statements.

EXCEPT:
- Returns the distinct rows from the left SELECT statement that are not present in the right SELECT statement.
- It is similar to the set difference operation in mathematics.

INTERSECT:
- Returns the distinct rows that are common to both SELECT statements.
- It is similar to the set intersection operation in mathematics.

REGEX

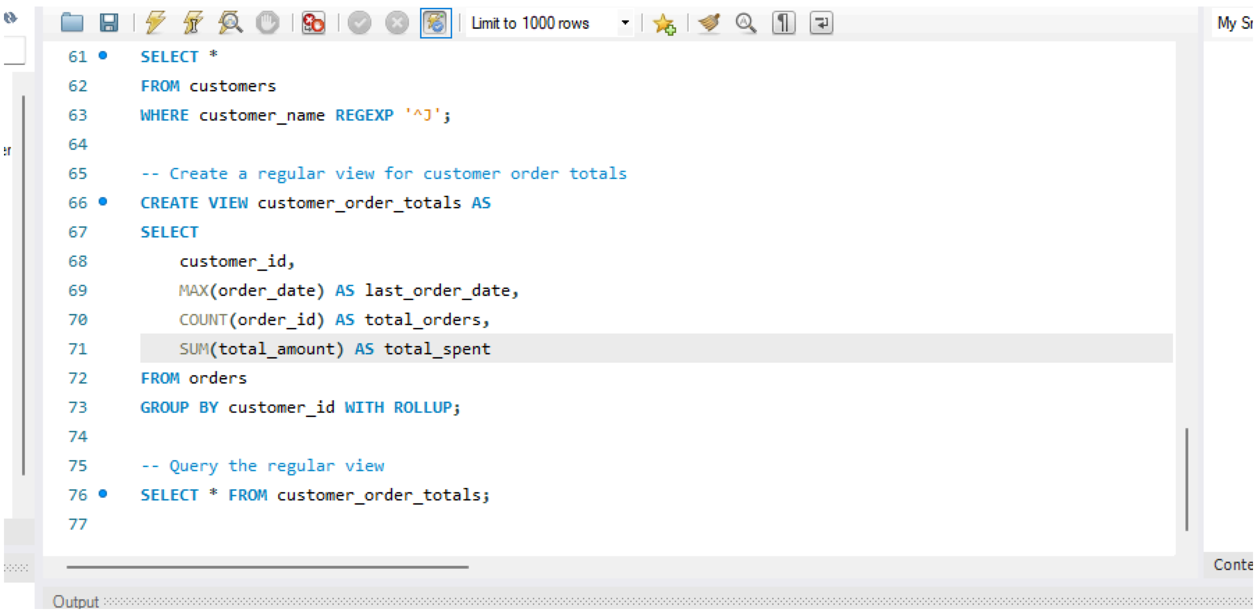


```
FROM orders;

SELECT *
FROM customers
WHERE customer_name REGEXP '^J';
```

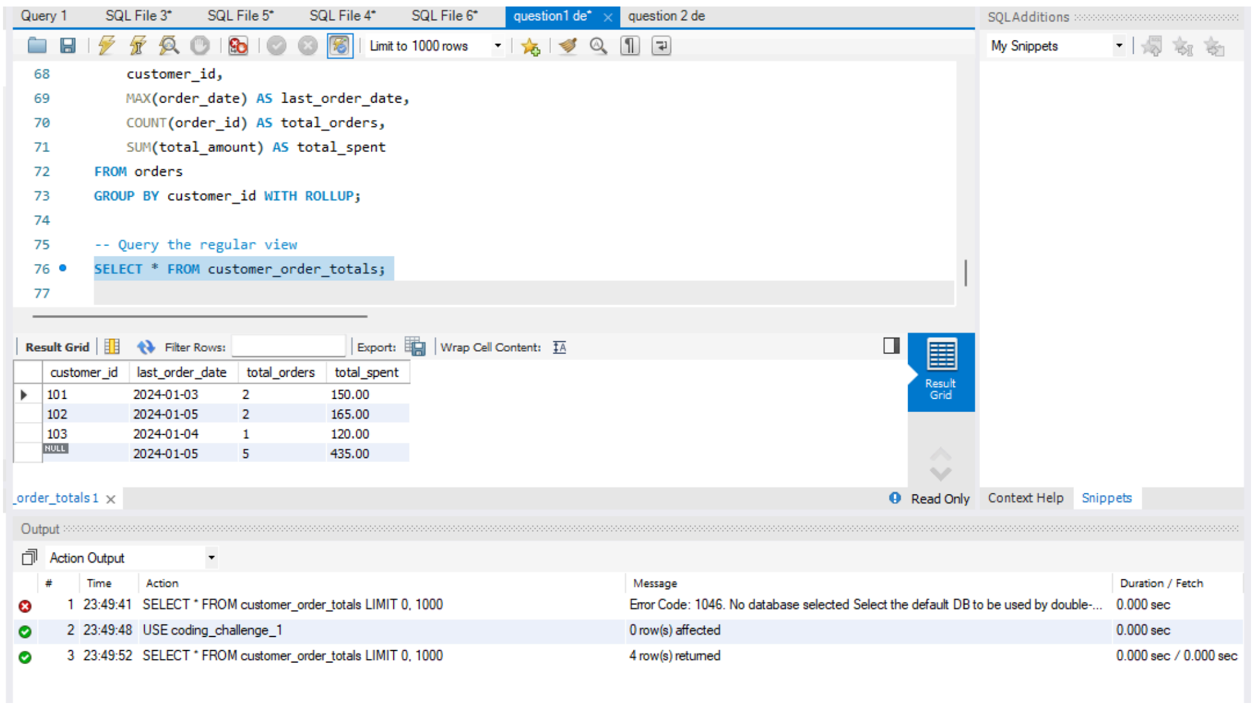
customer_id	customer_name
101	John Doe
102	Jane Smith
NULL	NULL

#	Time	Action	Message	Duration / Fetch
90	23:20:56	show tables	8 row(s) returned	0.063 sec / 0.000 sec
91	23:32:15	SELECT customer_id, MAX(order_date) AS last_order_date, COUNT(order_...	Error Code: 1146. Table 'snowflake.orders' doesn't exist	0.016 sec
92	23:32:21	USE coding_challenge_1	0 row(s) affected	0.000 sec
93	23:32:27	SELECT customer_id, MAX(order_date) AS last_order_date, COUNT(order_...	4 row(s) returned	0.000 sec / 0.000 sec
94	23:41:41	USE coding_challenge_1	0 row(s) affected	0.000 sec
95	23:41:47	SELECT * FROM customers WHERE customer_name REGEXP '^J' LIMIT 0, 1000	2 row(s) returned	0.016 sec / 0.000 sec

Materialized view



```sql
SELECT *
FROM customers
WHERE customer_name REGEXP '^J';

-- Create a regular view for customer order totals
CREATE VIEW customer_order_totals AS
SELECT
    customer_id,
    MAX(order_date) AS last_order_date,
    COUNT(order_id) AS total_orders,
    SUM(total_amount) AS total_spent
FROM orders
GROUP BY customer_id WITH ROLLUP;

-- Query the regular view
SELECT * FROM customer_order_totals;
```



```sql
    customer_id,
    MAX(order_date) AS last_order_date,
    COUNT(order_id) AS total_orders,
    SUM(total_amount) AS total_spent
FROM orders
GROUP BY customer_id WITH ROLLUP;

-- Query the regular view
SELECT * FROM customer_order_totals;
```

customer_id	last_order_date	total_orders	total_spent
101	2024-01-03	2	150.00
102	2024-01-05	2	165.00
103	2024-01-04	1	120.00
NULL	2024-01-05	5	435.00

```
#  Time      Action                                        Message                                                                         Duration / Fetch
1  23:49:41  SELECT * FROM customer_order_totals LIMIT 0, 1000  Error Code: 1046. No database selected Select the default DB to be used by double-...  0.000 sec
2  23:49:48  USE coding_challenge_1                        0 row(s) affected                                                               0.000 sec
3  23:49:52  SELECT * FROM customer_order_totals LIMIT 0, 1000  4 row(s) returned                                                               0.000 sec / 0.000 sec
```