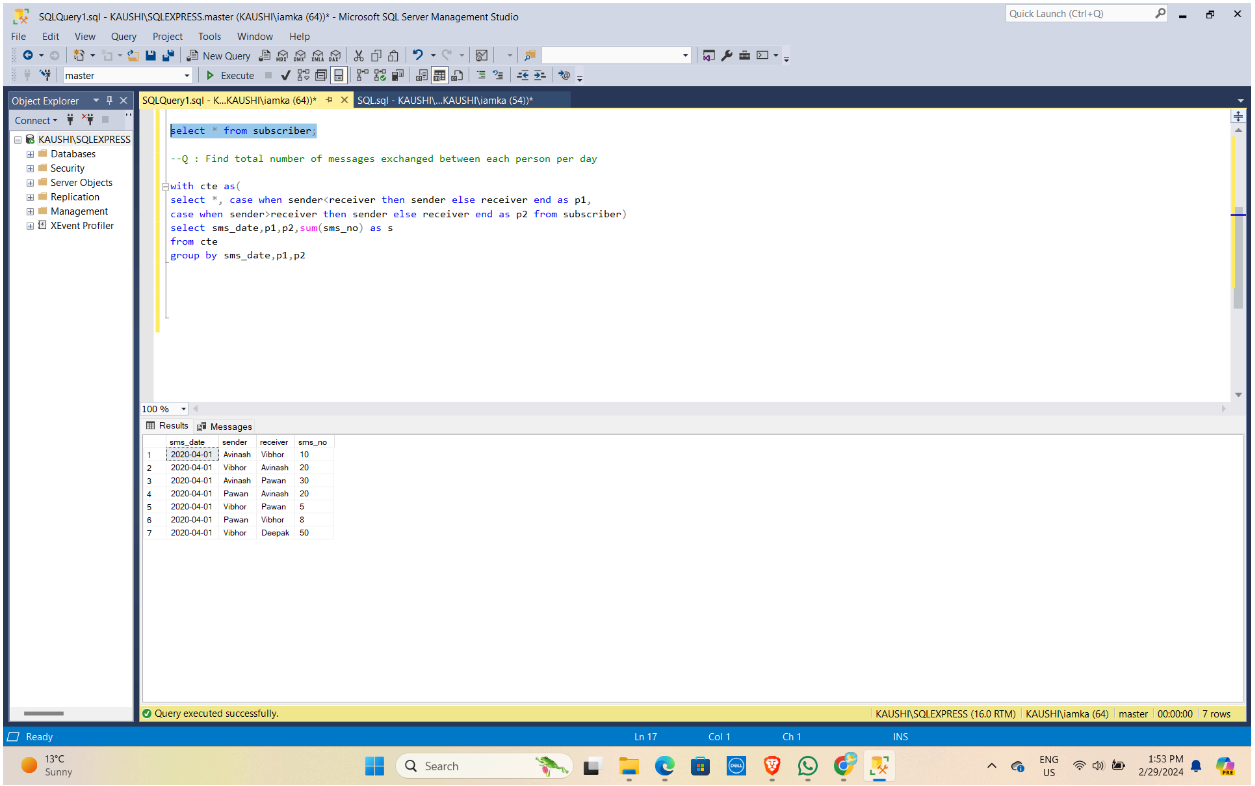
Day24 - Feb29th 2024

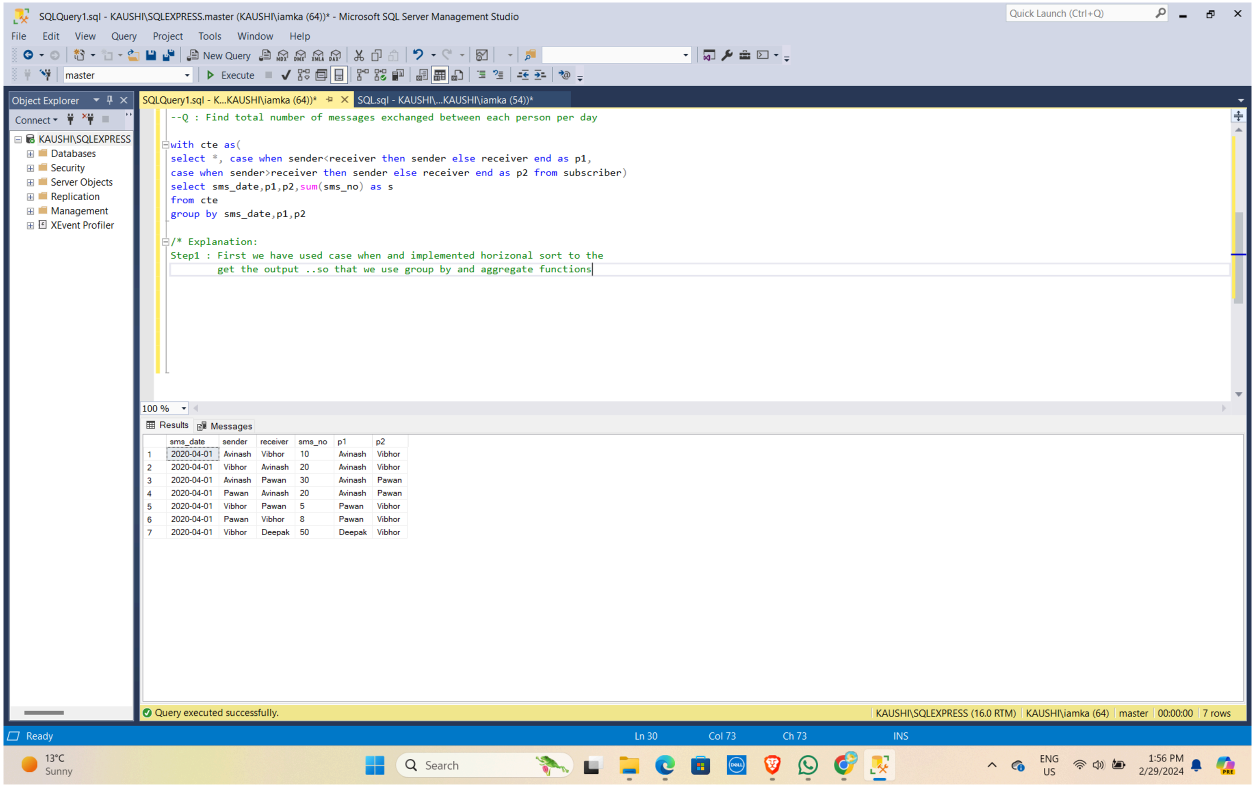
1. Started my day as usual
2. Marketed my profile for data engineering
3. Cooked food for friends and for myself
4. Solved one leetcode medium problem
5. Pyspark project is in the process
6. Ended my day by solving two complex SQL questions from Ankit’s YT


SQLQuery1.sql - KAUSHI\SQLEXPRESS.master (KAUSHI\iamka (64))* - Microsoft SQL Server Management Studio
SQLQuery1.sql - K...KAUSHI\iamka (64))*
SQL.sql - KAUSHI\...KAUSHI\iamka (54))*
Object Explorer
KAUSHI\SQLEXPRESS
Databases
Security
Server Objects
Replication
Management
XEvent Profiler

```sql
select * from subscriber;

--Q : Find total number of messages exchanged between each person per day

with cte as(
select *, case when sender<receiver then sender else receiver end as p1,
case when sender>receiver then sender else receiver end as p2 from subscriber)
select sms_date,p1,p2,sum(sms_no) as s
from cte
group by sms_date,p1,p2
```

	sms_date	sender	receiver	sms_no
1	2020-04-01	Avinash	Vibhor	10
2	2020-04-01	Vibhor	Avinash	20
3	2020-04-01	Avinash	Pawan	30
4	2020-04-01	Pawan	Avinash	20
5	2020-04-01	Vibhor	Pawan	5
6	2020-04-01	Pawan	Vibhor	8
7	2020-04-01	Vibhor	Deepak	50

Query executed successfully. KAUSHI\SQLEXPRESS (16.0 RTM) KAUSHI\iamka (64) master 00:00:00 7 rows

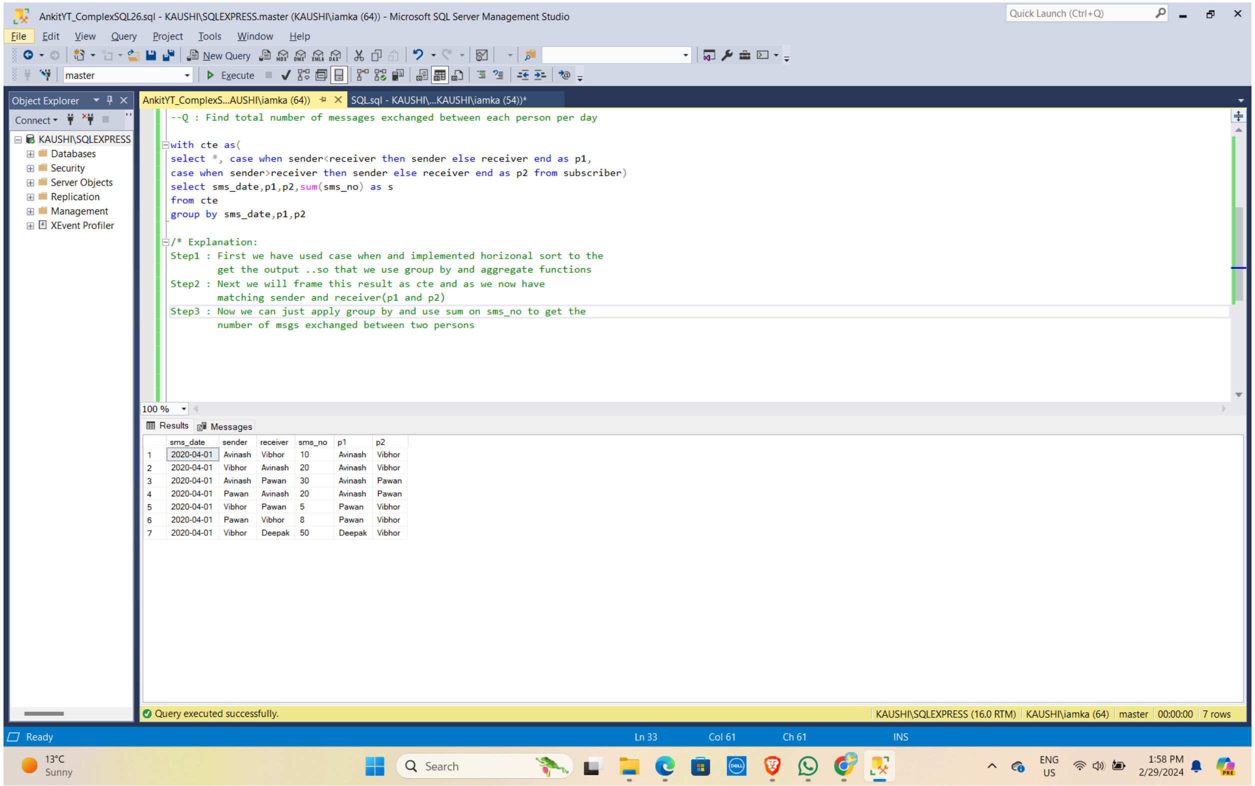


```sql
--Q : Find total number of messages exchanged between each person per day

with cte as(
select *, case when sender<receiver then sender else receiver end as p1,
case when sender>receiver then sender else receiver end as p2 from subscriber)
select sms_date,p1,p2,sum(sms_no) as s
from cte
group by sms_date,p1,p2

/* Explanation:
Step1 : First we have used case when and implemented horizonal sort to the
        get the output ..so that we use group by and aggregate functions
```

	sms_date	sender	receiver	sms_no	p1	p2
1	2020-04-01	Avinash	Vibhor	10	Avinash	Vibhor
2	2020-04-01	Vibhor	Avinash	20	Avinash	Vibhor
3	2020-04-01	Avinash	Pawan	30	Avinash	Pawan
4	2020-04-01	Pawan	Avinash	20	Avinash	Pawan
5	2020-04-01	Vibhor	Pawan	5	Pawan	Vibhor
6	2020-04-01	Pawan	Vibhor	8	Pawan	Vibhor
7	2020-04-01	Vibhor	Deepak	50	Deepak	Vibhor

Query executed successfully.

KAUSHI\SQLEXPRESS (16.0 RTM) | KAUSHI\iamka (64) | master | 00:00:00 | 7 rows

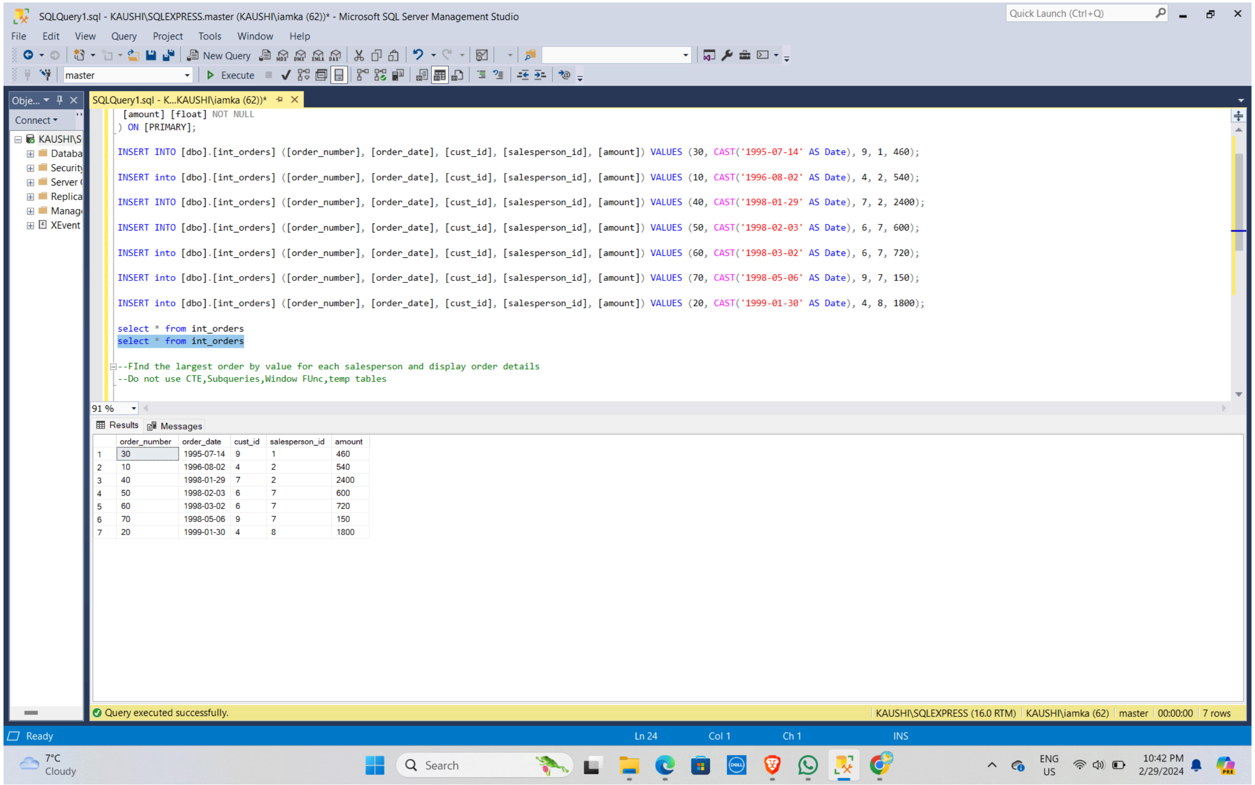

AnkitYT_ComplexSQL26.sql - KAUSHI\SQLEXPRESS.master (KAUSHI\iamka (64)) - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
master
Execute
Object Explorer
Connect
KAUSHI\SQLEXPRESS
Databases
Security
Server Objects
Replication
Management
XEvent Profiler
AnkitYT_ComplexS...AUSHI\iamka (64))
SQL.sql - KAUSHI\...KAUSHI\iamka (54))*
100 %
Results
Messages
Ready
Ln 33
Col 61
Ch 61
INS
13°C
Sunny
Search
ENG
US
1:58 PM
2/29/2024

```sql
--Q : Find total number of messages exchanged between each person per day

with cte as(
select *, case when sender<receiver then sender else receiver end as p1,
case when sender>receiver then sender else receiver end as p2 from subscriber)
select sms_date,p1,p2,sum(sms_no) as s
from cte
group by sms_date,p1,p2

/* Explanation:
Step1 : First we have used case when and implemented horizonal sort to the
        get the output ..so that we use group by and aggregate functions
Step2 : Next we will frame this result as cte and as we now have
        matching sender and receiver(p1 and p2)
Step3 : Now we can just apply group by and use sum on sms_no to get the
        number of msgs exchanged between two persons
```

	sms_date	sender	receiver	sms_no	p1	p2
1	2020-04-01	Avinash	Vibhor	10	Avinash	Vibhor
2	2020-04-01	Vibhor	Avinash	20	Avinash	Vibhor
3	2020-04-01	Avinash	Pawan	30	Avinash	Pawan
4	2020-04-01	Pawan	Avinash	20	Avinash	Pawan
5	2020-04-01	Vibhor	Pawan	5	Pawan	Vibhor
6	2020-04-01	Pawan	Vibhor	8	Pawan	Vibhor
7	2020-04-01	Vibhor	Deepak	50	Deepak	Vibhor

Query executed successfully.

KAUSHI\SQLEXPRESS (16.0 RTM) | KAUSHI\iamka (64) | master | 00:00:00 | 7 rows



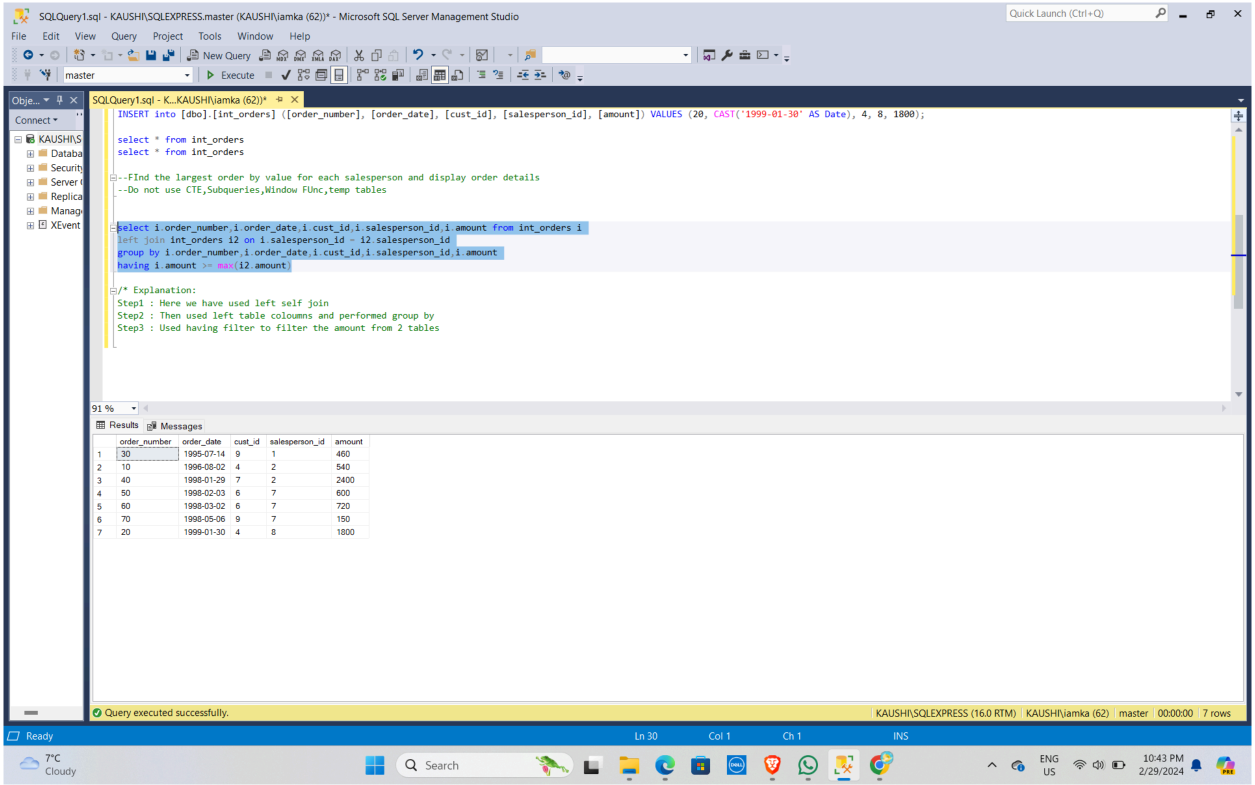
```
  [amount] [float] NOT NULL
) ON [PRIMARY];

INSERT INTO [dbo].[int_orders] ([order_number], [order_date], [cust_id], [salesperson_id], [amount]) VALUES (30, CAST('1995-07-14' AS Date), 9, 1, 460);

INSERT into [dbo].[int_orders] ([order_number], [order_date], [cust_id], [salesperson_id], [amount]) VALUES (10, CAST('1996-08-02' AS Date), 4, 2, 540);

INSERT INTO [dbo].[int_orders] ([order_number], [order_date], [cust_id], [salesperson_id], [amount]) VALUES (40, CAST('1998-01-29' AS Date), 7, 2, 2400);

INSERT INTO [dbo].[int_orders] ([order_number], [order_date], [cust_id], [salesperson_id], [amount]) VALUES (50, CAST('1998-02-03' AS Date), 6, 7, 600);

INSERT into [dbo].[int_orders] ([order_number], [order_date], [cust_id], [salesperson_id], [amount]) VALUES (60, CAST('1998-03-02' AS Date), 6, 7, 720);

INSERT into [dbo].[int_orders] ([order_number], [order_date], [cust_id], [salesperson_id], [amount]) VALUES (70, CAST('1998-05-06' AS Date), 9, 7, 150);

INSERT into [dbo].[int_orders] ([order_number], [order_date], [cust_id], [salesperson_id], [amount]) VALUES (20, CAST('1999-01-30' AS Date), 4, 8, 1800);

select * from int_orders
select * from int_orders

--FInd the largest order by value for each salesperson and display order details
--Do not use CTE,Subqueries,Window FUnc,temp tables
```

	order_number	order_date	cust_id	salesperson_id	amount
1	30	1995-07-14	9	1	460
2	10	1996-08-02	4	2	540
3	40	1998-01-29	7	2	2400
4	50	1998-02-03	6	7	600
5	60	1998-03-02	6	7	720
6	70	1998-05-06	9	7	150
7	20	1999-01-30	4	8	1800

Query executed successfully.

KAUSHI\SQLEXPRESS (16.0 RTM) | KAUSHI\iamka (62) | master | 00:00:00 | 7 rows

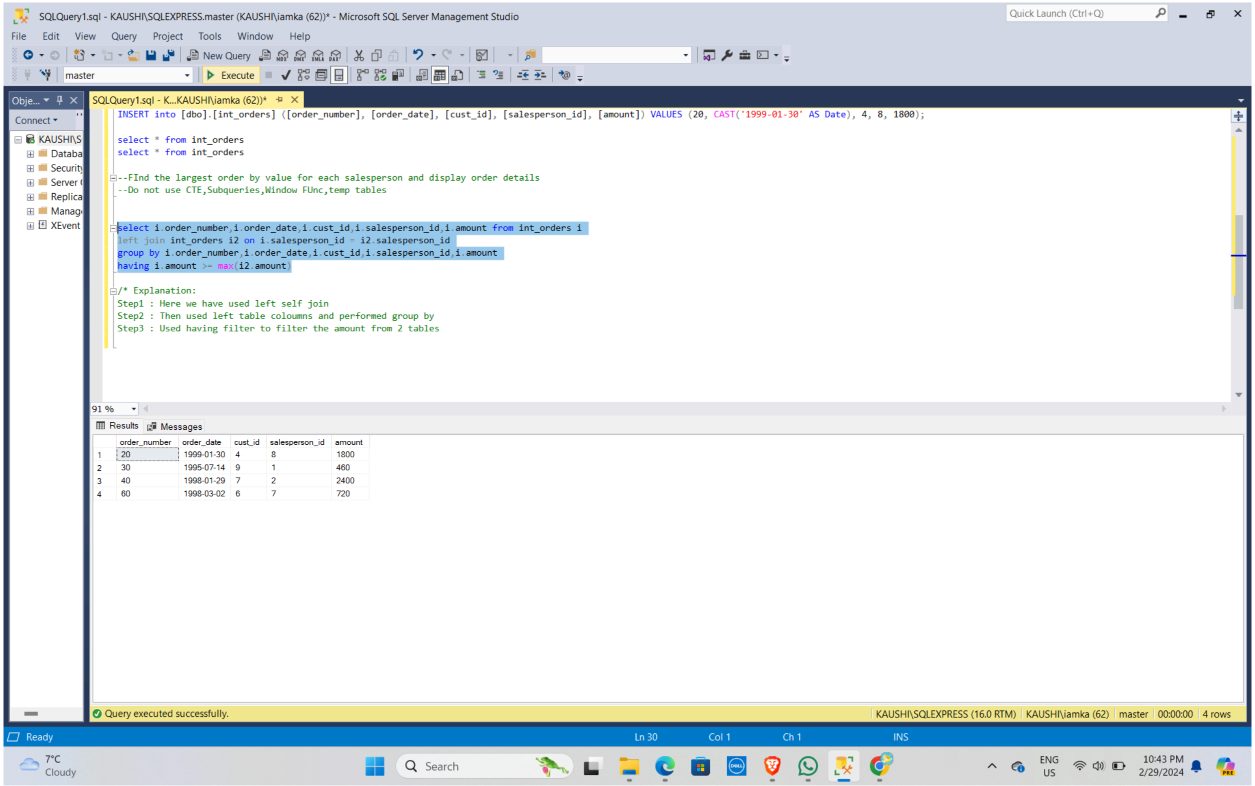
Ready Ln 24 Col 1 Ch 1 INS



```
INSERT into [dbo].[int_orders] ([order_number], [order_date], [cust_id], [salesperson_id], [amount]) VALUES (20, CAST('1999-01-30' AS Date), 4, 8, 1800);

select * from int_orders
select * from int_orders

--FInd the largest order by value for each salesperson and display order details
--Do not use CTE,Subqueries,Window FUnc,temp tables


select i.order_number,i.order_date,i.cust_id,i.salesperson_id,i.amount from int_orders i
left join int_orders i2 on i.salesperson_id = i2.salesperson_id
group by i.order_number,i.order_date,i.cust_id,i.salesperson_id,i.amount
having i.amount >= max(i2.amount)

/* Explanation:
Step1 : Here we have used left self join
Step2 : Then used left table coloumns and performed group by
Step3 : Used having filter to filter the amount from 2 tables
```

	order_number	order_date	cust_id	salesperson_id	amount
1	30	1995-07-14	9	1	460
2	10	1996-08-02	4	2	540
3	40	1998-01-29	7	2	2400
4	50	1998-02-03	6	7	600
5	60	1998-03-02	6	7	720
6	70	1998-05-06	9	7	150
7	20	1999-01-30	4	8	1800

Query executed successfully. KAUSHI\SQLEXPRESS (16.0 RTM) | KAUSHI\iamka (62) | master | 00:00:00 | 7 rows



```
INSERT into [dbo].[int_orders] ([order_number], [order_date], [cust_id], [salesperson_id], [amount]) VALUES (20, CAST('1999-01-30' AS Date), 4, 8, 1800);

select * from int_orders
select * from int_orders

--FInd the largest order by value for each salesperson and display order details
--Do not use CTE,Subqueries,Window FUnc,temp tables


select i.order_number,i.order_date,i.cust_id,i.salesperson_id,i.amount from int_orders i
left join int_orders i2 on i.salesperson_id = i2.salesperson_id
group by i.order_number,i.order_date,i.cust_id,i.salesperson_id,i.amount
having i.amount >= max(i2.amount)

/* Explanation:
Step1 : Here we have used left self join
Step2 : Then used left table coloumns and performed group by
Step3 : Used having filter to filter the amount from 2 tables
```

	order_number	order_date	cust_id	salesperson_id	amount
1	20	1999-01-30	4	8	1800
2	30	1995-07-14	9	1	460
3	40	1998-01-29	7	2	2400
4	60	1998-03-02	6	7	720

Query executed successfully. KAUSHI\SQLEXPRESS (16.0 RTM) | KAUSHI\iamka (62) | master | 00:00:00 | 4 rows