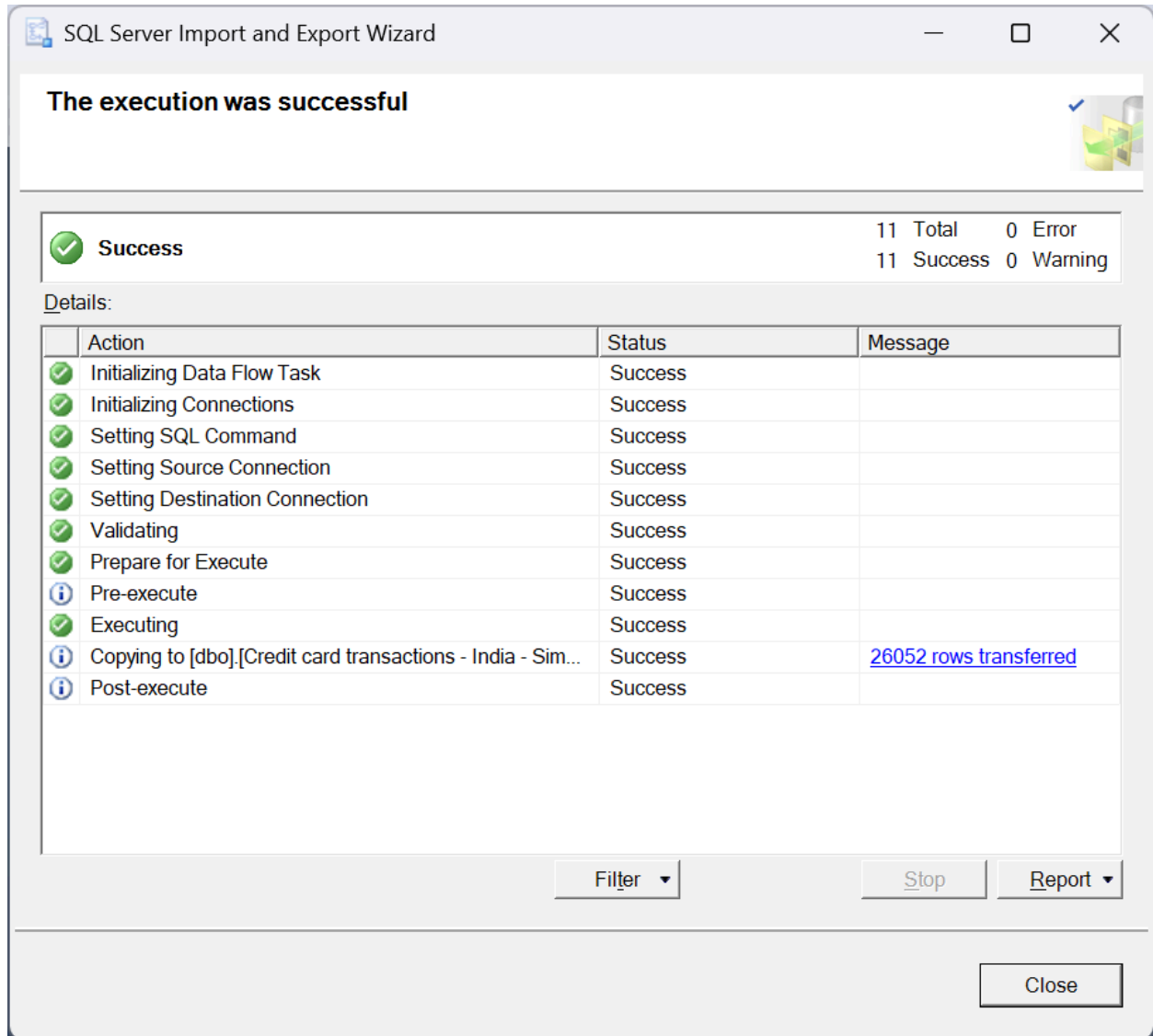


Import the CSV file into SQL server.
Before importing,
change the column names to lower case
replace space within column names with underscore
change the data types of columns



Below is the data preview

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the file is 'SQLQuery2.sql' in the 'SRIDLAPPY-2798\SQLEXPRESS.credit_card_transactions' database. The Object Explorer on the left shows the database structure, including 'credit_card_transactions' and its tables. The query editor in the center contains the following SQL query:

```
SELECT TOP (1000) [index]
,[city]
,[date]
,[card_type]
,[exp_type]
,[gender]
,[amount]
FROM [credit_card_transactions].[dbo].[Credit card transactions - India - Simple]
```

The Results pane at the bottom displays the first 13 rows of the query results. The status bar at the bottom indicates 'Query executed successfully' and '1,000 rows'.

	index	city	date	card_type	exp_type	gender	amount
1	14153	Peddapuram, India	2014-08-15	Silver	Grocery	F	45476.00
2	14154	Rewa, India	2014-11-25	Platinum	Grocery	M	139310.00
3	14155	Barnala, India	2014-06-22	Signature	Bills	F	90718.00
4	14156	Barpeta, India	2014-07-21	Gold	Bills	M	64188.00
5	14157	Sironj, India	2013-11-11	Platinum	Entertainment	M	297574.00
6	14158	Fatehabad, India	2014-04-18	Silver	Entertainment	M	257574.00
7	14159	Pachora, India	2013-11-01	Signature	Fuel	M	110996.00
8	14160	Vikramasingapuram, India	2013-10-24	Platinum	Food	M	93207.00
9	14161	Mangaluru, India	2014-02-13	Platinum	Food	F	178429.00
10	14162	Palwal, India	2014-09-13	Platinum	Entertainment	M	55110.00
11	14163	Nabarangapur, India	2014-01-14	Signature	Food	M	68016.00
12	14164	Balkan, India	2014-06-14	Silver	Bills	F	125105.00
13	14165	Vedaranvam, India	2014-05-05	Gold	Grocery	M	150725.00

I explored the dataset initially, by using few SQL queries

To see the date ranges in my dataset, I used the below query

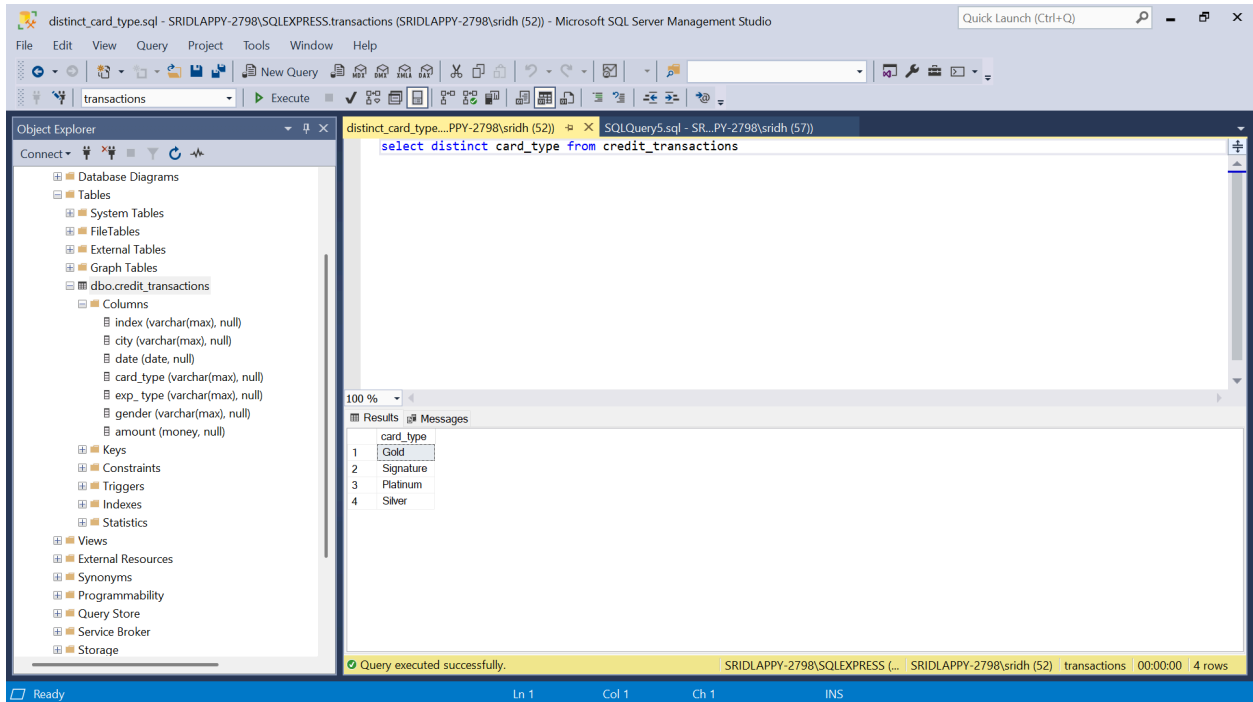
The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the file is 'date_range.sql' in the 'SRIDLAPPY-2798\SQLEXPRESS.transactions' database. The Object Explorer on the left shows the database structure, including 'credit_transactions' and its columns. The query editor in the center contains the following SQL query:

```
select min(date) as start_dt, max(date) as end_dt from credit_transactions;
```

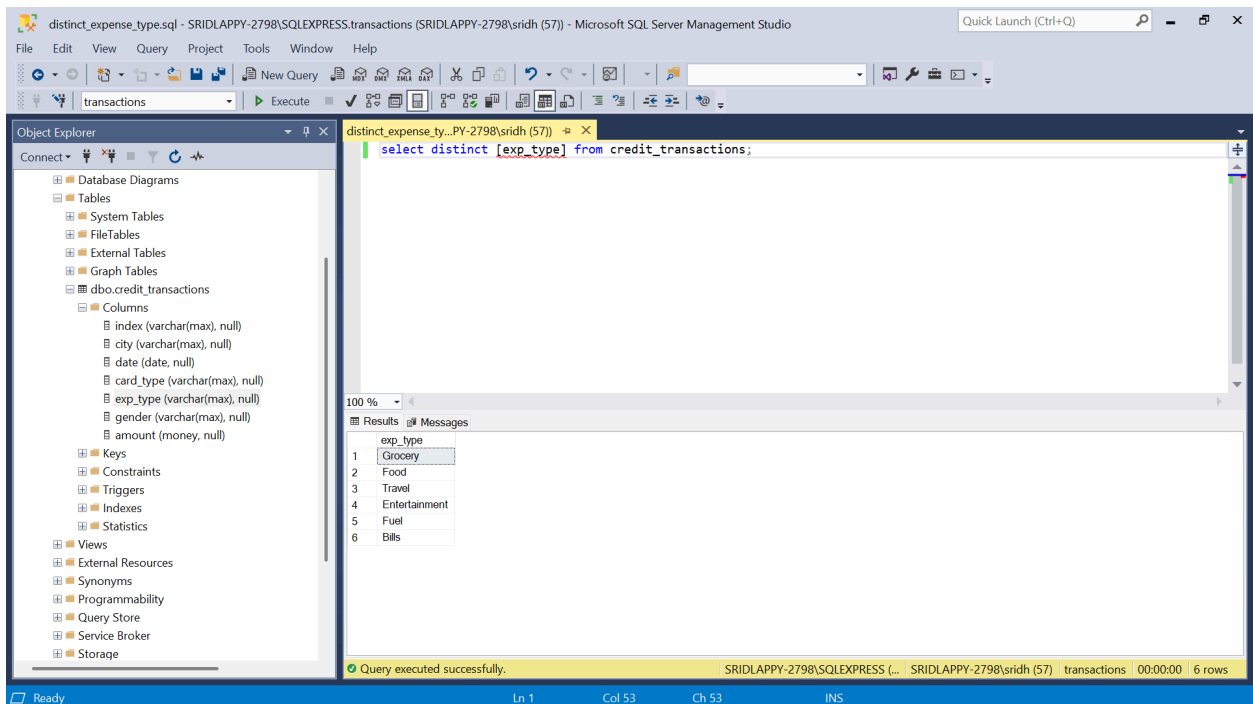
The Results pane at the bottom displays the results of the query. The status bar at the bottom indicates 'Query executed successfully' and '1 rows'.

	start_dt	end_dt
1	2013-10-04	2015-05-26

To see the different types of cards, I can see that the four different types of cards in the dataset are Gold, Signature, Platinum and Silver.



I see that the different types of expenses on which credit cards were used caters to Entertainment, Food, Bills, Fuel, Travel and Grocery



Also, the dataset analyzes the credit card spending behavior across 986 cities in India

distinct_city.sql - SRIDLAPPY-2798\SQLEXPRESS.transactions (SRIDLAPPY-2798\sridh (65)) - Microsoft SQL Server Management Studio

Query: `select distinct city from credit_transactions`

city
975 Yanam, India
976 Yavatmal, India
977 Yawal, India
978 Yellandu, India
979 Yemmiganur, India
980 Yerraguntla, India
981 Yevla, India
982 Zaidpur, India
983 Zamaia, India
984 Zira, India
985 Zirakpur, India
986 Zunheboto, India

Query executed successfully. SRIDLAPPY-2798\SQLEXPRESS (SRIDLAPPY-2798\sridh (65)) transactions 00:00:00 986 rows

Analysis:

I tried to see the top 5 cities which are having the highest spends and how they are contributing towards total credit card spends. From the below query I found that, “Greater Mumbai” city tops the list in percentage contribution towards total credit card spends

SQLQuery7.sql - SRIDLAPPY-2798\SQLEXPRESS.transactions (SRIDLAPPY-2798\sridh (52)) - Microsoft SQL Server Management Studio

Query: `use transactions; with t1 as (select city, sum(amount) as spend, rank() over (order by sum(amount) desc) as x from credit_transactions group by city), t2 as (select sum(amount) as y from credit_transactions) select top 5 *, (t1.spend/t2.y)*100 as percent_contrib from t1,t2;`

city	spend	x	y	percent_contrib
1 Greater Mumbai, India	578751476.00	1	4074833373.00	14.15
2 Bengaluru, India	572326739.00	2	4074833373.00	14.04
3 Ahmedabad, India	567794310.00	3	4074833373.00	13.93
4 Delhi, India	556929212.00	4	4074833373.00	13.66
5 Kolkata, India	115466943.00	5	4074833373.00	2.83

Query executed successfully. SRIDLAPPY-2798\SQLEXPRESS (SRIDLAPPY-2798\sridh (52)) transactions 00:00:00 5 rows

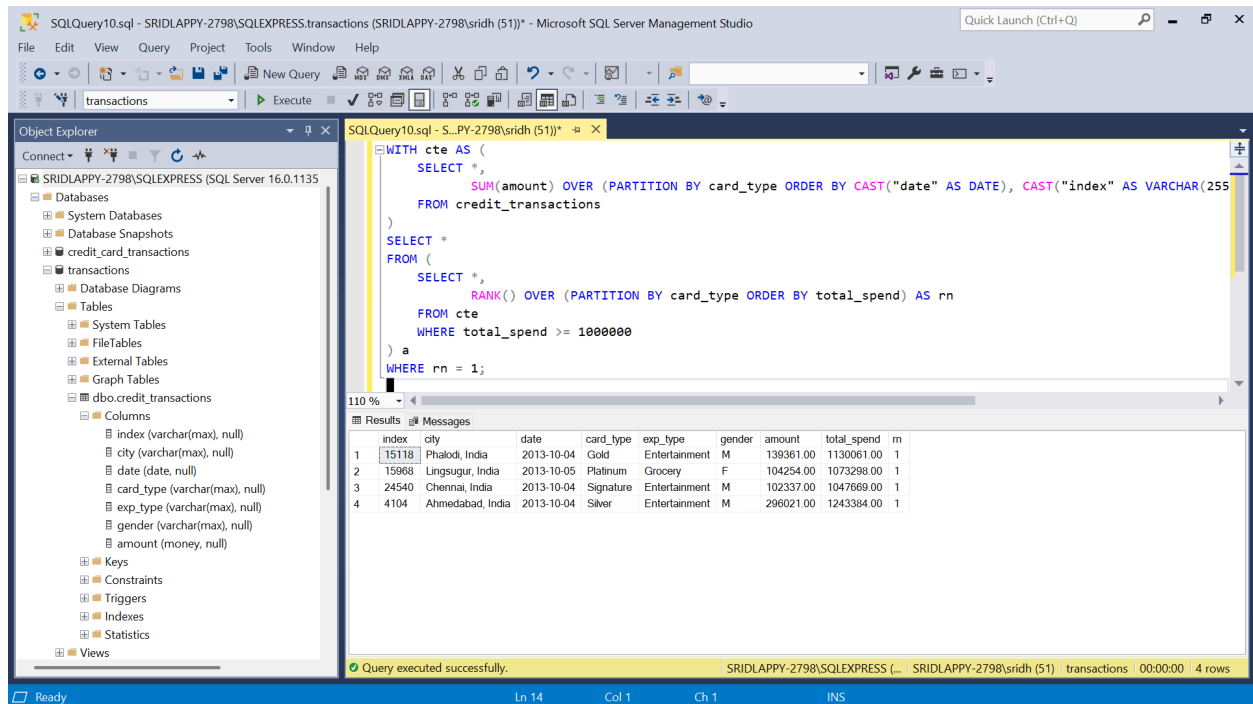
From the below query, I can see that ,
“Gold” type cards are used most during the month of “January” .

“Platinum” type cards is being most used during the month of August

“Signature” type cards is being most used during the month of December

“Silver” type cards is being most used during the month of March

To identify key transactions contributing to milestones or thresholds in spending for reporting and detecting significant patterns, I used the below query



The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL code:

```
WITH cte AS (
    SELECT *,
        SUM(amount) OVER (PARTITION BY card_type ORDER BY CAST("date" AS DATE), CAST("index" AS VARCHAR(255))
        FROM credit_transactions
    )
    SELECT *
    FROM (
        SELECT *,
            RANK() OVER (PARTITION BY card_type ORDER BY total_spend) AS rn
        FROM cte
        WHERE total_spend >= 1000000
    ) a
    WHERE rn = 1;
```

The results pane shows the following data:

	index	city	date	card_type	exp_type	gender	amount	total_spend	rn
1	15118	Phalodi, India	2013-10-04	Gold	Entertainment	M	139361.00	1130061.00	1
2	15968	Lingsugur, India	2013-10-05	Platinum	Grocery	F	104254.00	1073298.00	1
3	24540	Chennai, India	2013-10-04	Signature	Entertainment	M	102337.00	1047669.00	1
4	4104	Ahmedabad, India	2013-10-04	Silver	Entertainment	M	296021.00	1243384.00	1

“Achalpur, India” is the city with lowest credit card spend for gold card type as below

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'SRIDLAPPY-2798\SQLEXPRESS'. The main query window contains the following SQL code:

```

with cte as (
    select top 1 city, card_type, sum(amount) as amount
    , sum(case when card_type='Gold' then amount end) as gold_amount
    from credit_transactions
    group by city, card_type)
select
    city, sum(gold_amount)*1.0/sum(amount) as gold_ratio
from cte
group by city
having count(gold_amount) > 0 and sum(gold_amount)>0
order by gold_ratio;

```

The Results pane shows the output of the query:

city	gold_ratio
Achalpur, India	1.0000000000000000

The status bar at the bottom indicates 'Query executed successfully.' and '1 rows'.

To see, the highest spend and lowest spend on different expenses across each city, I used the below query

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'SRIDLAPPY-2798\SQLEXPRESS'. The main query window contains the following SQL code:

```

with cte as (
    select city, exp_type, sum(amount) as total_amount from credit_transactions
    group by city, exp_type)
select
    city , max(case when rn_asc=1 then exp_type end) as lowest_exp_type
    , min(case when rn_desc=1 then exp_type end) as highest_exp_type
from
    (select *
    , rank() over(partition by city order by total_amount desc) rn_desc
    , rank() over(partition by city order by total_amount asc) rn_asc
    from cte) A
group by city;

```

The Results pane shows the output of the query:

city	lowest_exp_type	highest_exp_type
1 Achalpur, India	Entertainment	Grocery
2 Adilabad, India	Food	Bills
3 Adityapur, India	Grocery	Food
4 Adoni, India	Entertainment	Bills
5 Adoor, India	Bills	Fuel
6 Atzalpur, India	Food	Fuel
7 Agartala, India	Food	Grocery
8 Agra, India	Grocery	Bills
9 Ahmedabad, India	Grocery	Fuel
10 Ahmednagar, India	Grocery	Food
11 Aizawl, India	Grocery	Food
12 Ajmer, India	Fuel	Entertainment
13 Akola, India	Fuel	Bills
14 Akot, India	Entertainment	Fuel
15 Alambur, India	Entertainment	Food

The status bar at the bottom indicates 'Query executed successfully.' and '986 rows'.

To see what percentage of transactions are from female consumers, I used the below query and I found that highest percentage of transactions were in “Bills” category from female consumers.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'SRIDLAPPY-2798\SQLEXPRESS'. The main query window contains the following SQL code:

```
select exp_type,
sum(case when gender='F' then amount else 0 end)/sum(amount) as percentage_female_contribution
from credit_transactions
group by exp_type
order by percentage_female_contribution desc;
```

The Results pane shows the following data:

exp_type	percentage_female_contribution
Bills	0.63945879327770097
Food	0.54905310632226301
Travel	0.51132870420723746
Grocery	0.50911022600902162
Fuel	0.49710380717846034
Entertainment	0.49372907542046395

The status bar at the bottom indicates 'Query executed successfully.' and '6 rows'.

To see the combination of “card” and “expense_type” which saw the highest month over month growth rate in jan 2014, I used below query and I found that the combination of “Platinum” card and “Grocery” expense type has the highest month over month growth

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'SRIDLAPPY-2798\SQLEXPRESS'. The main query window contains the following SQL code:

```
with cte as (
select card_type,exp_type,datepart(year,"date") yt
,datepart(month,"date") mt,sum(amount) as total_spend
from credit_transactions
group by card_type,exp_type,datepart(year,"date"),datepart(month,"date")
)
select top 1 *, (total_spend-prev_mont_spend) as mom_growth
from (
select *
,lag(total_spend,1) over(partition by card_type,exp_type order by yt,mt) as prev_mont_spend
from cte) A
where prev_mont_spend is not null and yt=2014 and mt=1
order by mom_growth desc;
```

The Results pane shows the following data:

card_type	exp_type	yt	mt	total_spend	prev_mont_spend	mom_growth
Platinum	Grocery	2014	1	12256343.00	7757562.00	4498781.00

The status bar at the bottom indicates 'Query executed successfully.' and '1 rows'.

To find which city has the highest “total spend to total number of transactions” ratio during weekends, I used the below query and I found that “Sonepur, India” is the city with highest spend.

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the file is 'city_with_highest_spend_during_weekends.sql' in the 'SRIDLAPPY-2798\SQLEXPRESS.transactions' database. The Object Explorer on the left shows the database structure, including tables like 'credit_card_transactions' and 'transactions'. The query editor in the center contains the following SQL query:

```
select top 1 city , sum(amount)*1.0/count(1) as ratio
from credit_transactions
where datepart(weekday,"date") in (1,7)
group by city
order by ratio desc;
```

The Results pane at the bottom shows the output of the query:

city	ratio
Sonepur, India	299905.000000000000000000

The status bar at the bottom indicates the query was executed successfully, returning 1 row.

To find the city which took least number of days to reach its 500th transaction after the first transaction, I used below query and I found that it's “Bangalore, India”

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the file is 'city_with_earliest_500th_transaction.sql' in the 'SRIDLAPPY-2798\SQLEXPRESS.transactions' database. The Object Explorer on the left shows the database structure. The query editor in the center contains the following SQL query:

```
with cte as (
select *
, row_number() over(partition by city order by "date","index") as rn
from credit_transactions)
select top 1 city,datediff(day,min("date"),max("date")) as datediff1
from cte
where rn=1 or rn=500
group by city
having count(1)=2
order by datediff1
```

The Results pane at the bottom shows the output of the query:

city	datediff1
Bangalore, India	81

The status bar at the bottom indicates the query was executed successfully, returning 1 row.

