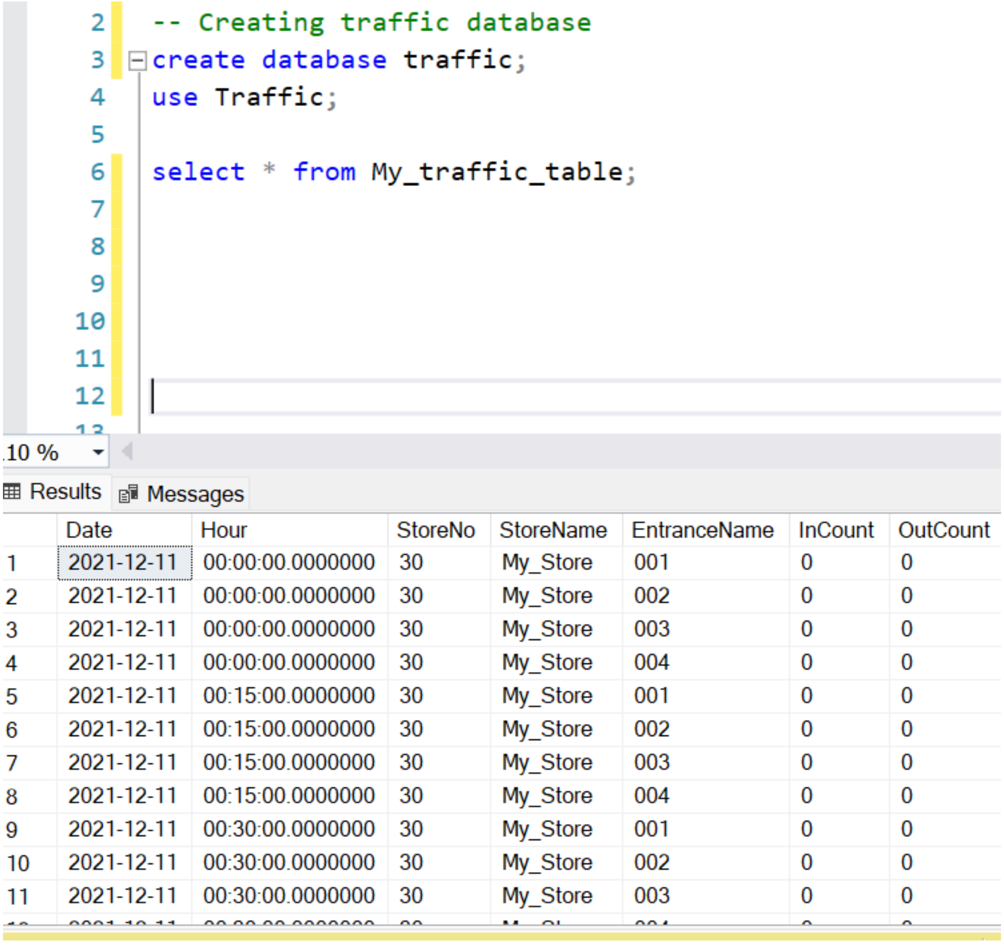
Task 6

1. **Traffic database**

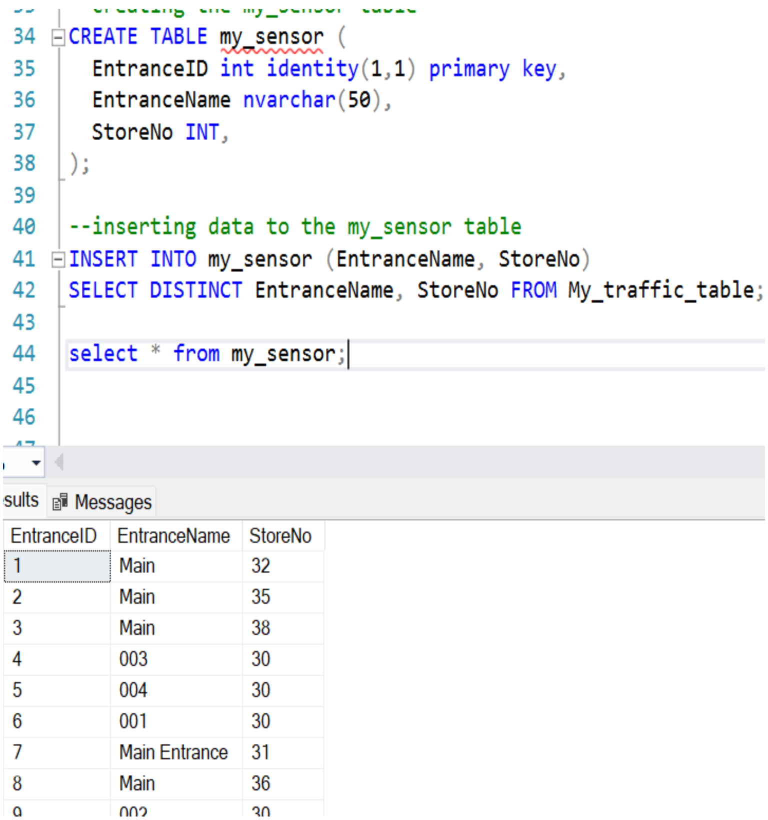
```
create database traffic;
use Traffic;
```



```
-- Creating traffic database
create database traffic;
use Traffic;

select * from My_traffic_table;
```

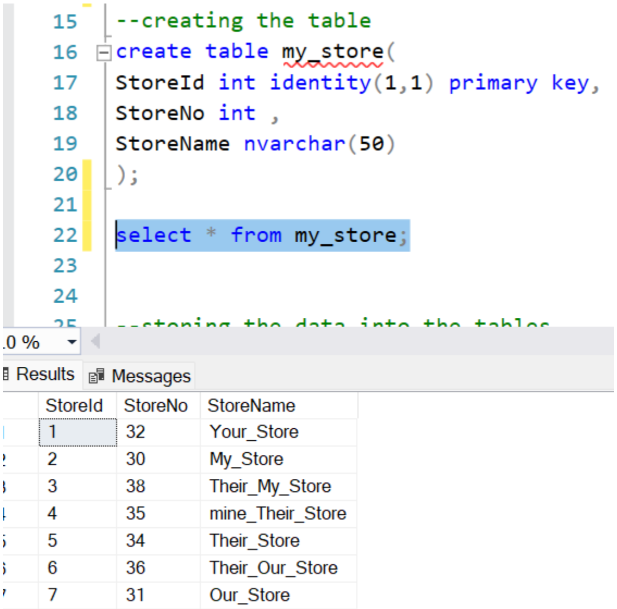
	Date	Hour	StoreNo	StoreName	EntranceName	InCount	OutCount
1	2021-12-11	00:00:00.0000000	30	My_Store	001	0	0
2	2021-12-11	00:00:00.0000000	30	My_Store	002	0	0
3	2021-12-11	00:00:00.0000000	30	My_Store	003	0	0
4	2021-12-11	00:00:00.0000000	30	My_Store	004	0	0
5	2021-12-11	00:15:00.0000000	30	My_Store	001	0	0
6	2021-12-11	00:15:00.0000000	30	My_Store	002	0	0
7	2021-12-11	00:15:00.0000000	30	My_Store	003	0	0
8	2021-12-11	00:15:00.0000000	30	My_Store	004	0	0
9	2021-12-11	00:30:00.0000000	30	My_Store	001	0	0
10	2021-12-11	00:30:00.0000000	30	My_Store	002	0	0
11	2021-12-11	00:30:00.0000000	30	My_Store	003	0	0

2. Creating my_store and my_sensor table to insert store and entrance data respectively



```
--creating the my_sensor table
CREATE TABLE my_sensor (
  EntranceID int identity(1,1) primary key,
  EntranceName nvarchar(50),
  StoreNo INT,
);

--inserting data to the my_sensor table
INSERT INTO my_sensor (EntranceName, StoreNo)
SELECT DISTINCT EntranceName, StoreNo FROM My_traffic_table;

select * from my_sensor;
```

EntranceID	EntranceName	StoreNo
1	Main	32
2	Main	35
3	Main	38
4	003	30
5	004	30
6	001	30
7	Main Entrance	31
8	Main	36
9	002	30



```
--creating the table
create table my_store(
StoreId int identity(1,1) primary key,
StoreNo int ,
StoreName nvarchar(50)
);

select * from my_store;


--storing the data into the tables
```

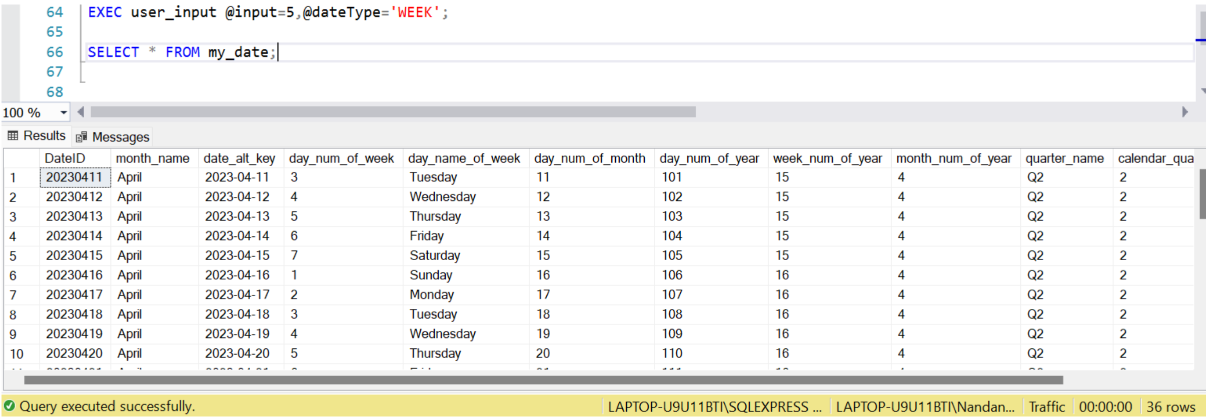
StoreId	StoreNo	StoreName
1	32	Your_Store
2	30	My_Store
3	38	Their_My_Store
4	35	mine_Their_Store
5	34	Their_Store
6	36	Their_Our_Store
7	31	Our_Store

3. Create my_time_table and my_date_table

For date table:

```
use Traffic;

- creating my_date table

--creating procedure
go

CREATE PROCEDURE user_input
       @input INT,
       @dateType nvarchar(30)
AS
BEGIN
       DROP TABLE IF EXISTS my_date;
       create table my_date(
                DateID INT PRIMARY KEY ,
                month_name nvarchar(50),
                date_alt_key DATE,
                day_num_of_week INT,
                day_name_of_week NVARCHAR(20),
                day_num_of_month INT,
                day_num_of_year INT,
                week_num_of_year INT,
                month_num_of_year INT,
                quarter_name NVARCHAR(20),
                calendar_quarter INT,
                calendar_year INT
       );

-- User input and get start and end dates

       DECLARE @current_date DATE=GETDATE();
       DECLARE @start DATE=(SELECT CASE @dateType
                                                WHEN 'DAY' THEN DATEADD(DAY,-
@input,@current_date)
                                                WHEN 'WEEK' THEN DATEADD(WEEK,-
@input,@current_date)
                                                WHEN 'MONTH' THEN
DATEADD(MONTH,-@input,@current_date)
                                                WHEN 'YEAR' THEN DATEADD(YEAR,-
@input,@current_date)
                                                ELSE Null
                                         END)
       DECLARE @end DATE=@current_date;

       WHILE @start<=@end
       BEGIN
              INSERT INTO
              my_date(DateID,month_name, date_alt_key, day_num_of_week,
day_name_of_week, day_num_of_month, day_num_of_year, week_num_of_year,
month_num_of_year, quarter_name, calendar_quarter, calendar_year)
              VALUES(
              CONVERT(INT,FORMAT(@start,'yyyyMMdd')),
              FORMAT(@start,'MMMM'),
              @start,
              DATEPART(WEEKDAY,@start),
              FORMAT(@start,'dddd'),
              DATEPART(DAY,@start),
              DATEPART(DAYOFYEAR,@start),
              DATEPART(WEEK,@start),
              DATEPART(MONTH,@start),
              CONCAT('Q',DATEPART(QUARTER,@start)),
              DATEPART(QUARTER,@start),
              DATEPART(YEAR,@start)
);

       SET @start=DATEADD(DAY,1,@start);
       END;
END;

EXEC user_input @input=5,@dateType='WEEK';

SELECT * FROM my_date;
```

Output:



```
EXEC user_input @input=5,@dateType='WEEK';

SELECT * FROM my_date;
```

	DateID	month_name	date_alt_key	day_num_of_week	day_name_of_week	day_num_of_month	day_num_of_year	week_num_of_year	month_num_of_year	quarter_name	calendar_qua
1	20230411	April	2023-04-11	3	Tuesday	11	101	15	4	Q2	2
2	20230412	April	2023-04-12	4	Wednesday	12	102	15	4	Q2	2
3	20230413	April	2023-04-13	5	Thursday	13	103	15	4	Q2	2
4	20230414	April	2023-04-14	6	Friday	14	104	15	4	Q2	2
5	20230415	April	2023-04-15	7	Saturday	15	105	15	4	Q2	2
6	20230416	April	2023-04-16	1	Sunday	16	106	16	4	Q2	2
7	20230417	April	2023-04-17	2	Monday	17	107	16	4	Q2	2
8	20230418	April	2023-04-18	3	Tuesday	18	108	16	4	Q2	2
9	20230419	April	2023-04-19	4	Wednesday	19	109	16	4	Q2	2
10	20230420	April	2023-04-20	5	Thursday	20	110	16	4	Q2	2

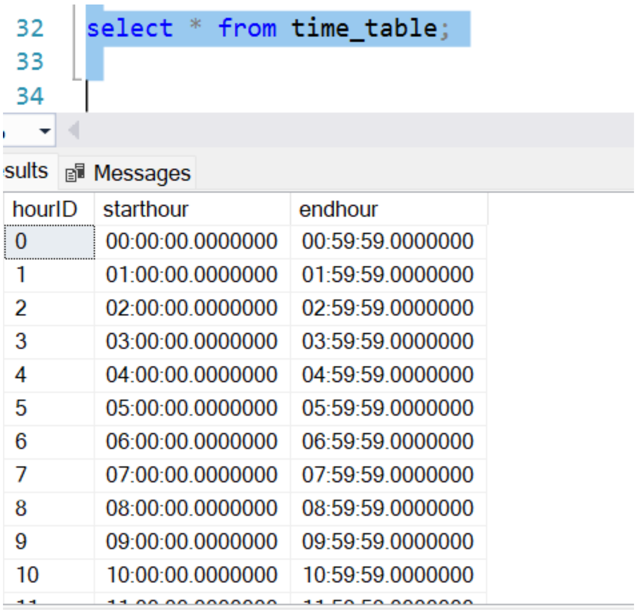
Query executed successfully. | LAPTOP-U9U11BTI\SQLEXPRESS ... | LAPTOP-U9U11BTI\Nandan... | Traffic | 00:00:00 | 36 rows

For time_table:

```
--creating time table

create table time_table(
hourID int primary key,
starthour time,
endhour time
);

--declaring the variables
declare @starttime time = '00:00:00';
declare @endtime time = '23:59:59';

declare @iter time = @starttime
declare @stop time = @starttime;

--conditioning the end
while (@stop!= @endtime)
begin
       set @stop = dateadd(second, 59, dateadd(minute,59,@iter))

       insert into time_table(hourID,starthour,endhour)
       values
       (
       convert(int, FORMAT(@iter, 'hh')),
       @iter,
       @stop
       )

SET @iter = DATEADD(HOUR, 1, @iter)
```

```
END;

select * from time_table;
```

Output:



```
select * from time_table;
```

hourID	starthour	endhour
0	00:00:00.0000000	00:59:59.0000000
1	01:00:00.0000000	01:59:59.0000000
2	02:00:00.0000000	02:59:59.0000000
3	03:00:00.0000000	03:59:59.0000000
4	04:00:00.0000000	04:59:59.0000000
5	05:00:00.0000000	05:59:59.0000000
6	06:00:00.0000000	06:59:59.0000000
7	07:00:00.0000000	07:59:59.0000000
8	08:00:00.0000000	08:59:59.0000000
9	09:00:00.0000000	09:59:59.0000000
10	10:00:00.0000000	10:59:59.0000000

4. Creating traffic_hourly_data for storing hourly data

```
CREATE TABLE traffic_hourly_data
(
DateID int,
hourID int,
StoreID int,
EntranceId int,
InCount int,
OutCount int,
PRIMARY KEY (DateID, HourID, StoreID, EntranceId),
FOREIGN key (DateID) REFERENCES my_date(DateID),
FOREIGN key (hourID) REFERENCES time_table(hourID),
FOREIGN key (StoreID) REFERENCES my_store(StoreID),
FOREIGN key (EntranceID) REFERENCES my_sensor(EntranceId)
);

;WITH CTE AS

(
SELECT
*,
SUM(InCount) OVER
(PARTITION BY

FORMAT(HOUR,'hh'),Date,StoreNo,EntranceName

ORDER BY Date
) AS Total_IN,
SUM(OutCount) OVER
(PARTITION BY

FORMAT(HOUR,'hh'),Date,StoreNo,EntranceName

ORDER BY Date
) AS Total_OUT
FROM my_traffic_table
)

-- SELECT * FROM CTE;

INSERT INTO traffic_hourly_data
(

DateID,
hourID,
StoreID,
EntranceId,
InCount,
OutCount
)
SELECT DISTINCT
dt.DateID,
tt.hourID,
mst.StoreID,
msn.EntranceID,
CTE.Total_IN,
CTE.Total_OUT
FROM CTE
join my_date as dt
on dt.date_alt_key=CTE.[Date]
join time_table as tt
on tt.hourID = FORMAT(CTE.[Hour], 'hh')
join my_store as mst
on mst.StoreNo=CTE.StoreNo
join my_sensor as msn
on msn.EntranceName=CTE.EntranceName
and msn.StoreID=mst.StoreID
ORDER BY 1, 2, 3, 4


SELECT * FROM traffic_hourly_data
```