SAINT MARY'S UNIVERSITY SINCE 1802

One University. One World. Yours.

MCDA 5540 – MANAGING & PROGRAMMING DATA

PROJECT TITLE: SEAGULL CRUISE LINE COMPANY

PROJECT DOCUMENTATION REPORT

12/10/2022

TEAM MEMBERS

NAMES	A#
Francis Alex Kuzhippallil	A00463084
Aravind Gopi	A00465022
Ajay Jain	A00455849

TABLE OF CONTENTS

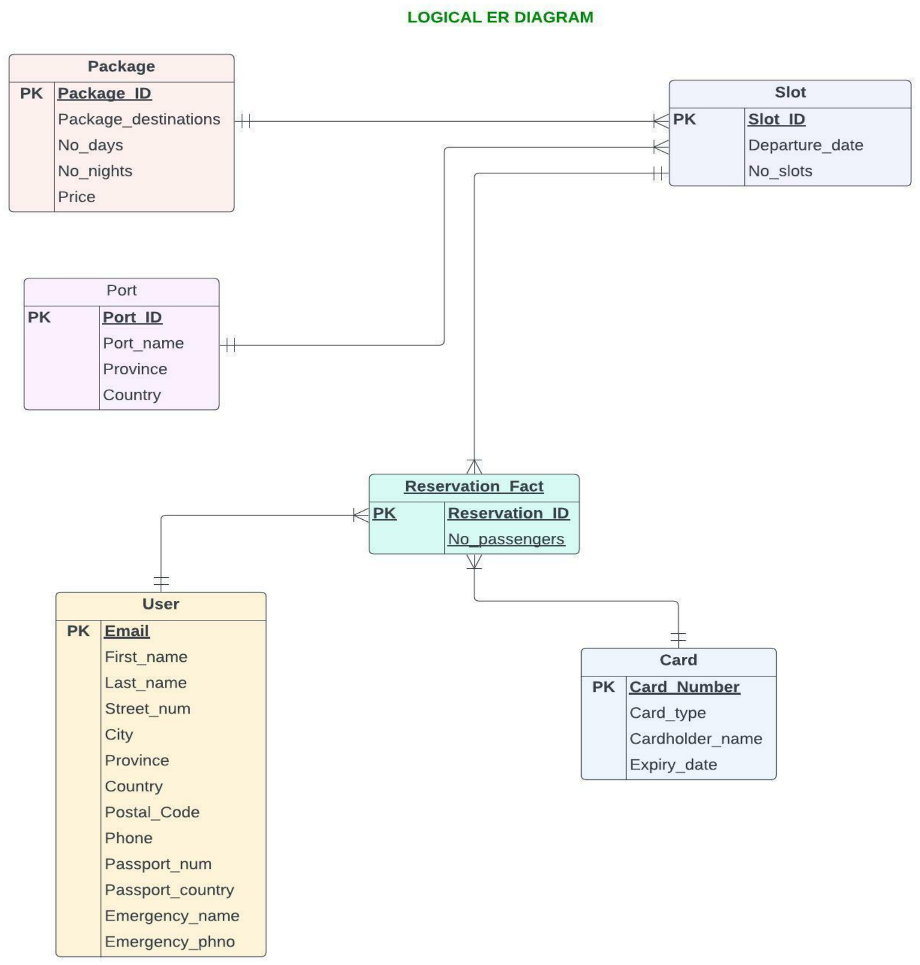
PROJECT BACKGROUND

To Design and implement a database for a luxury cruise line company *Seagull*. The database will store all data associated with online reservations of the *Seagull*. The implementation should facilitate the employees of the cruise line company to retrieve data from the database through SQL queries to process online reservations and analyze business data. The database users will be the cruise line company employees who process online reservations and/or analyze business data, not the company's customers who make online reservations.

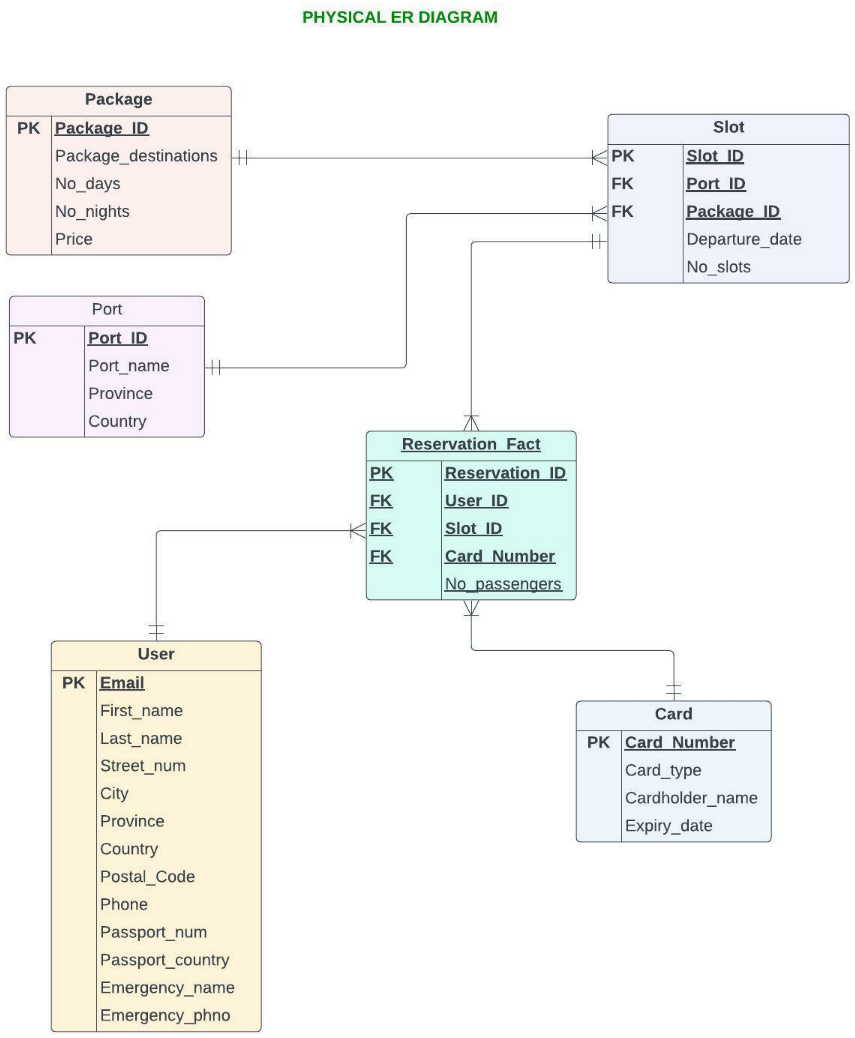
To retrieve data efficiently from the database, the database should be normalized. And to help the employee of the *Seagull* regarding the business outcomes, a set of analytical queries were discussed to increase the reservations, analyse the trends, find popular packages etc.

DATA MODEL

LOGICAL ER DIAGRAM



PHYSICAL ER DIAGRAM



FUNCTIONAL DEPENDENCIES

We have created six tables. Functional dependencies of each table are provided below.

Port Table

- Port_ID → Port_name
- Port_ID → Province
- Port_ID → Country

Package Table

- Package_ID → Package_Destinations
- Package_ID → No_days
- Package_ID → No_Nights

- Package_ID → Price

Slot Table

- Slot_ID → Port_ID
- Slot_ID → Package_ID
- Slot_ID → Departure_Date
- Slot_ID → No_Slots

User Table

- Email → First_name
- Email → Last_name
- Email → Street_num
- Email → City
- Email → Province
- Email → Country
- Email → Postal_code
- Email → Phone_Number
- Email → Passport
- Email → Passport_Issue_Country
- Email → Emergency_Name
- Email → Emergency_Phone

Card Table

- Card_Number → Card_type
- Card_number → Cardholder_name
- Card_number → Expiry_date

Reservation Table

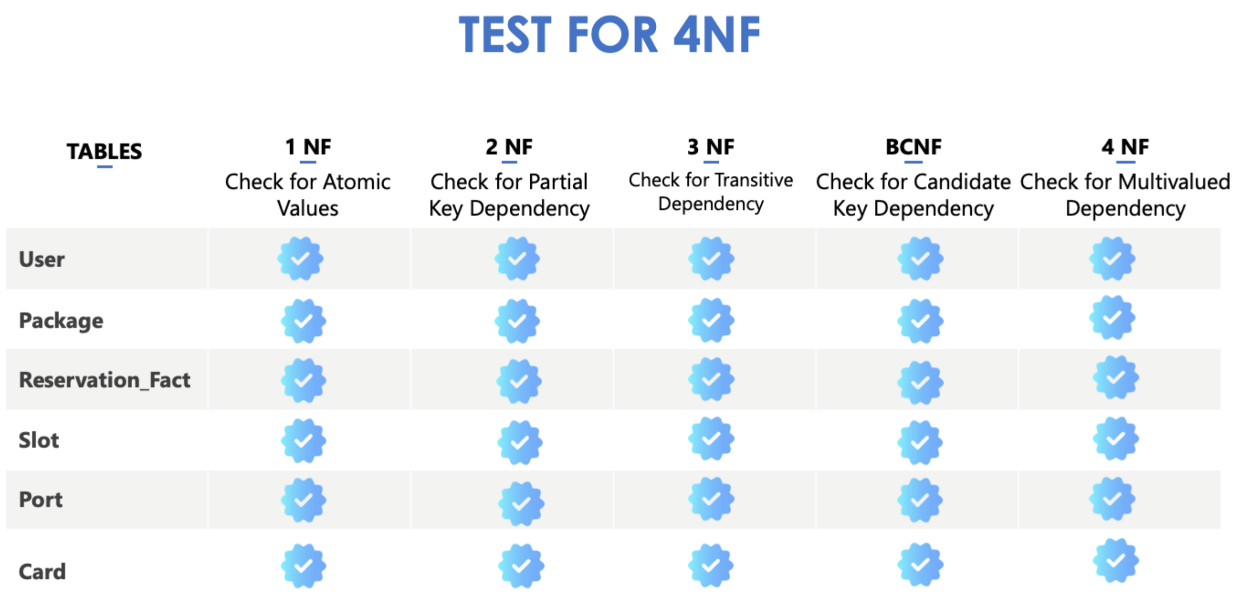
- Reservation_ID → User_ID
- Reservation_ID → Slot_ID
- Reservation_ID → Card_Number
- Reservation_ID → No_Passengers

ASSUMPTIONS

1. Added an extra field in the reservation form named "No_of_Guest", i.e., The person who is booking the cruise can be able to book for his/her family members or friends.
2. Assuming one person can have one or more credit cards for purchases. That is, one person may tie up with one or more banks.
3. Assuming one card can be used by one or more people. For Example, Husband's credit card can be used by the wife for purchases.
4. The number of slots for each package would default to 500.

5. Assuming the e-mail Id is unique for each user in the user table. So other attributes like phone number, and passport number will act as a candidate key.
6. When travellers make purchases on crise, the number of slots should reduce based on the number of guests and should include the person who is booking.

TEST FOR 4NF



TEST FOR 4NF

TABLES	1 NF Check for Atomic Values	2 NF Check for Partial Key Dependency	3 NF Check for Transitive Dependency	BCNF Check for Candidate Key Dependency	4 NF Check for Multivalued Dependency
User	✓	✓	✓	✓	✓
Package	✓	✓	✓	✓	✓
Reservation_Fact	✓	✓	✓	✓	✓
Slot	✓	✓	✓	✓	✓
Port	✓	✓	✓	✓	✓
Card	✓	✓	✓	✓	✓

The above figure depicts the testing of all six tables till 4NF. Since all tables do have a single primary key that determines all the attributes, therefore all six tables automatically follow till BCNF. Since there are no multivalued dependencies found, hence all six tables follow 4NF.

SYSTEM DESCRIPTION AND FEATURES

To enable *Seagull company* employees to easily retrieve business data from the database, we have come up with 19 SQL queries, ones that can provide useful business insights and help in making appropriate business decisions.

We have used Microsoft SQL Server Management Studio to create the tables following the Physical ER diagram schema. To generate data, Mockaroo website has been used. Table data, followed by SQL Scripts to create tables are provided in excel sheet (link below).

https://docs.google.com/spreadsheets/d/1jVC-5iSi6TEwO6h4FPuR6OceTdWWG6L2/edit?usp=share_link&ouid=109269890071558222722&rtpof=true&sd=true

The queries, insights and screenshots of output are provided below.

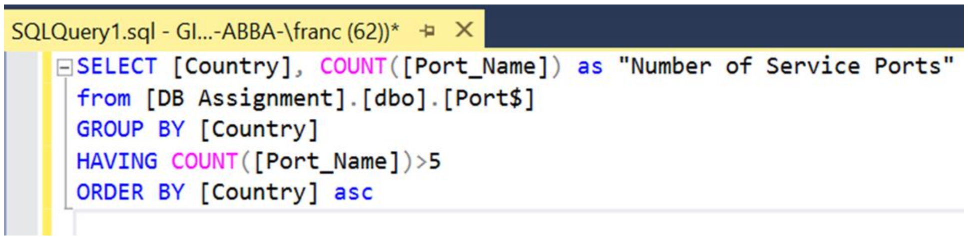
Query 1*: Provide a list of those countries which have more than 5 service ports in alphabetical order.*

Business Insights: This query helps to understand which country provides maximum ports, therefore increasing the number of packages in these ports will help to increase the profit margins.

SQL Query:

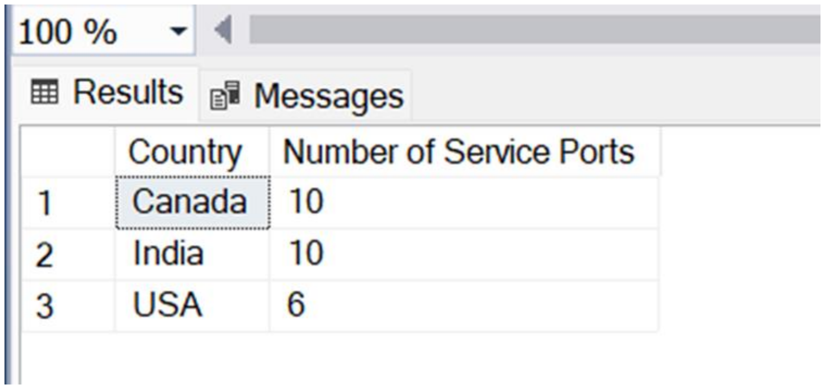
```
Select [Country], COUNT([Port_Name]) as "Number of Service Ports"
from [DB Assignment].[dbo].[Port$]
GROUP BY [Country]
HAVING COUNT([Port_Name])>5
ORDER BY [Country] asc
```

Screenshots of Query & Output



SQLQuery1.sql - GI...-ABBA-\franc (62))*

```
SELECT [Country], COUNT([Port_Name]) as "Number of Service Ports"
from [DB Assignment].[dbo].[Port$]
GROUP BY [Country]
HAVING COUNT([Port_Name])>5
ORDER BY [Country] asc
```



	Country	Number of Service Ports
1	Canada	10
2	India	10
3	USA	6

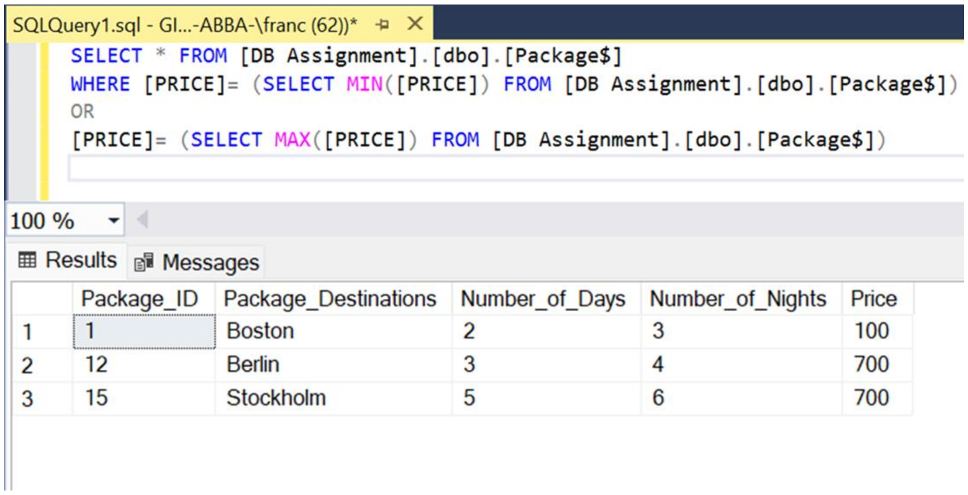
Query 2*: Providing details about the most expensive and least expensive packages available.*

Business Insights: This query helps to know the price range of the packages available. Based on the packages demand from different ports, the price for each package can be adjusted.

SQL Query:

```
SELECT * FROM [DB Assignment].[dbo].[Package$]
WHERE [PRICE]= (SELECT MIN([PRICE]) FROM [DB Assignment].[dbo].[Package$])
OR
[PRICE]= (SELECT MAX([PRICE]) FROM [DB Assignment].[dbo].[Package$])
```

Screenshots of Query & Output



```
SELECT * FROM [DB Assignment].[dbo].[Package$]
WHERE [PRICE]= (SELECT MIN([PRICE]) FROM [DB Assignment].[dbo].[Package$])
OR
[PRICE]= (SELECT MAX([PRICE]) FROM [DB Assignment].[dbo].[Package$])
```

	Package_ID	Package_Destinations	Number_of_Days	Number_of_Nights	Price
1	1	Boston	2	3	100
2	12	Berlin	3	4	700
3	15	Stockholm	5	6	700

Query 3*: Provide insights about the average number of slots booked over the past three months across all the ports.*

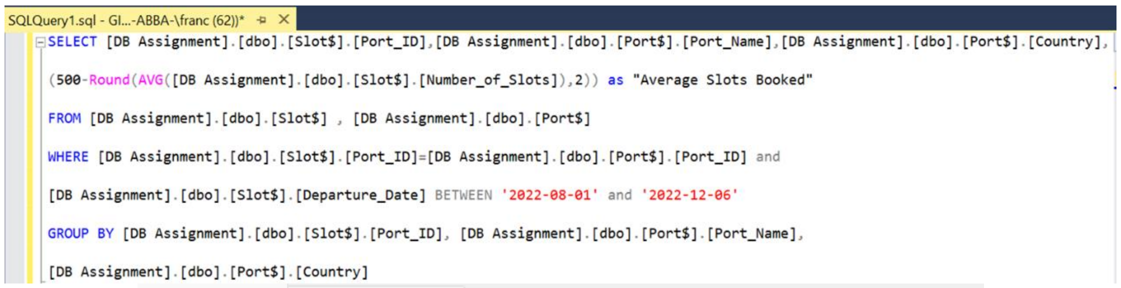
Business Insights: This query sheds light on the company's performance over the past three months across the ports, using the results, the company leadership team can make the appropriate decisions to ensure progress in upcoming months.

SQL Query:

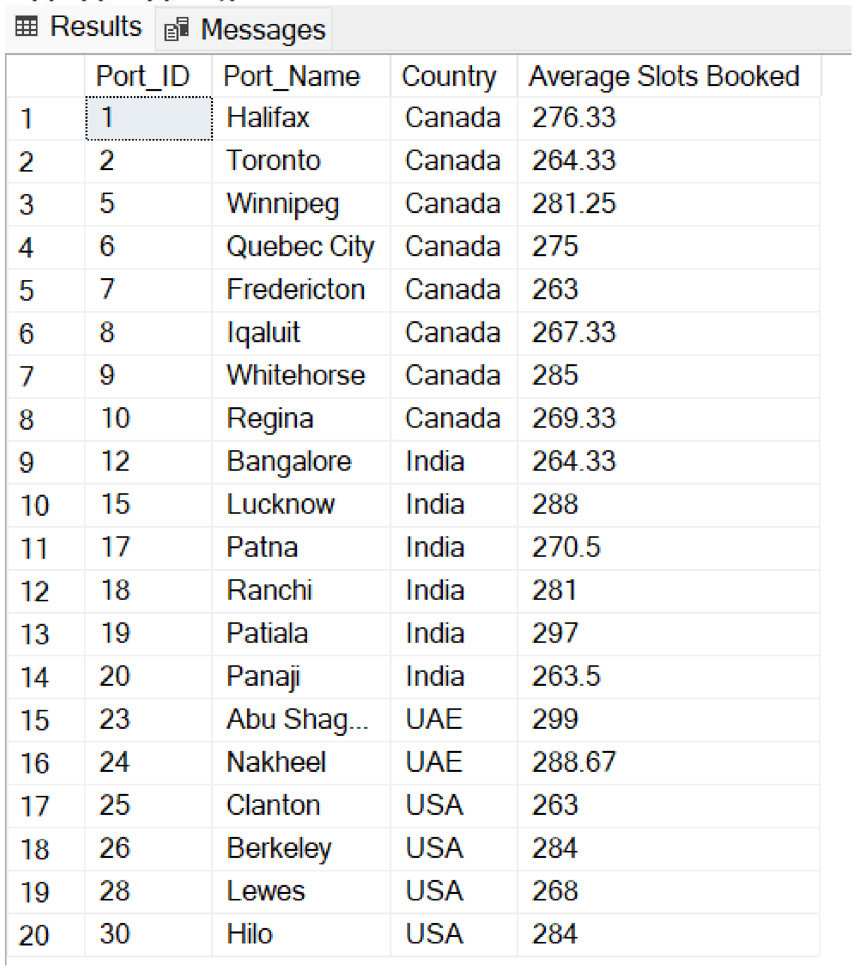
```
SELECT [DB Assignment].[dbo].[Slot$].[Port_ID],
   [DB Assignment].[dbo].[Port$].[Port_Name],
   [DB Assignment].[dbo].[Port$].[Country],
   (500-Round(AVG([DB Assignment].[dbo].[Slot$].[Number_of_Slots]),2))
              as "Average Slots Booked"

FROM [DB Assignment].[dbo].[Slot$] , [DB Assignment].[dbo].[Port$]

WHERE [DB Assignment].[dbo].[Slot$].[Port_ID]=[DB Assignment].[dbo].[Port$].[Port_ID]
  and
  [DB Assignment].[dbo].[Slot$].[Departure_Date]
             BETWEEN '2022-08-01' and '2022-12-06'
```

```
GROUP BY [DB Assignment].[dbo].[Slot$].[Port_ID],
         [DB Assignment].[dbo].[Port$].[Port_Name],
         [DB Assignment].[dbo].[Port$].[Country]
```

Screenshots of Query & Output



```
SELECT [DB Assignment].[dbo].[Slot$].[Port_ID],[DB Assignment].[dbo].[Port$].[Port_Name],[DB Assignment].[dbo].[Port$].[Country],

(500-Round(AVG([DB Assignment].[dbo].[Slot$].[Number_of_Slots]),2)) as "Average Slots Booked"

FROM [DB Assignment].[dbo].[Slot$] , [DB Assignment].[dbo].[Port$]

WHERE [DB Assignment].[dbo].[Slot$].[Port_ID]=[DB Assignment].[dbo].[Port$].[Port_ID] and

[DB Assignment].[dbo].[Slot$].[Departure_Date] BETWEEN '2022-08-01' and '2022-12-06'

GROUP BY [DB Assignment].[dbo].[Slot$].[Port_ID], [DB Assignment].[dbo].[Port$].[Port_Name],

[DB Assignment].[dbo].[Port$].[Country]
```



	Port_ID	Port_Name	Country	Average Slots Booked
1	1	Halifax	Canada	276.33
2	2	Toronto	Canada	264.33
3	5	Winnipeg	Canada	281.25
4	6	Quebec City	Canada	275
5	7	Fredericton	Canada	263
6	8	Iqaluit	Canada	267.33
7	9	Whitehorse	Canada	285
8	10	Regina	Canada	269.33
9	12	Bangalore	India	264.33
10	15	Lucknow	India	288
11	17	Patna	India	270.5
12	18	Ranchi	India	281
13	19	Patiala	India	297
14	20	Panaji	India	263.5
15	23	Abu Shag...	UAE	299
16	24	Nakheel	UAE	288.67
17	25	Clanton	USA	263
18	26	Berkeley	USA	284
19	28	Lewes	USA	268
20	30	Hilo	USA	284

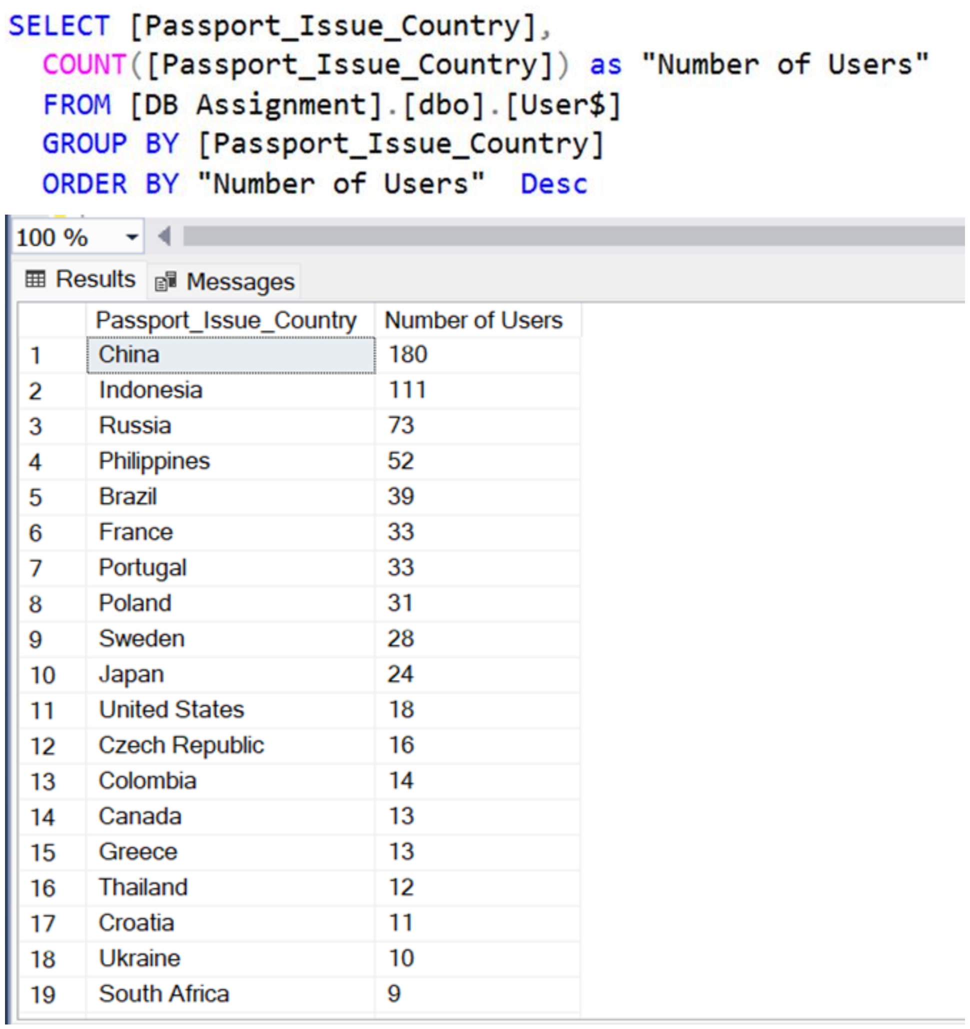
Query 4*: To find proportion of users along with their nationalities*

Business Insights: This query results can help to visualize the users nationalities distribution. Company can try to incorporate discounts, attractive deals based on seasons and festivals across the globe to attract more users form various nationalities.

SQL Query:

```
SELECT [Passport_Issue_Country],
COUNT([Passport_Issue_Country]) as "Number of Users"
FROM [DB Assignment].[dbo].[User$]
GROUP BY [Passport_Issue_Country]
ORDER BY "Number of Users"  Desc
```

Screenshots of Query & Output



```
SELECT [Passport_Issue_Country],
  COUNT([Passport_Issue_Country]) as "Number of Users"
  FROM [DB Assignment].[dbo].[User$]
  GROUP BY [Passport_Issue_Country]
  ORDER BY "Number of Users"  Desc
```

	Passport_Issue_Country	Number of Users
1	China	180
2	Indonesia	111
3	Russia	73
4	Philippines	52
5	Brazil	39
6	France	33
7	Portugal	33
8	Poland	31
9	Sweden	28
10	Japan	24
11	United States	18
12	Czech Republic	16
13	Colombia	14
14	Canada	13
15	Greece	13
16	Thailand	12
17	Croatia	11
18	Ukraine	10
19	South Africa	9

***Query 5**: To find top 5 users who have used maximum variety of cards for booking*

Business Insights: This query is intended to identify any fraudulent activity which can occur. If any anomalies are observed, employees can inform the user about the same, to bring the issue to users attention.

SQL Query:

SELECT TOP(5) [DB Assignment].[dbo].[User$].[First_Name],
  [DB Assignment].[dbo].[User$].[Last_Name],
  [DB Assignment].[dbo].[Reservation_Fact$].[Email],
  COUNT(DISTINCT [DB Assignment].[dbo].[Reservation_Fact$].[Card_Number]) as "Number of Variety Cards Used"

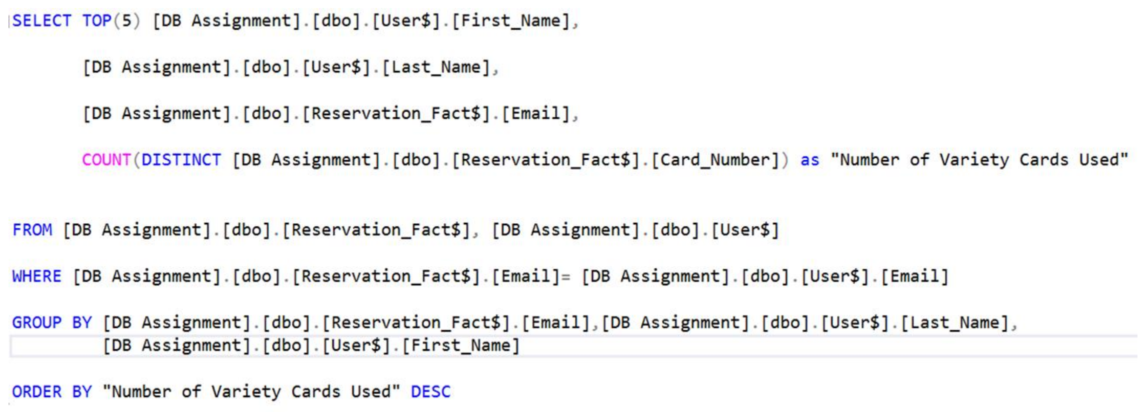
FROM [DB Assignment].[dbo].[Reservation_Fact$], [DB Assignment].[dbo].[User$]

WHERE [DB Assignment].[dbo].[Reservation_Fact$].[Email]= [DB Assignment].[dbo].[User$].[Email]

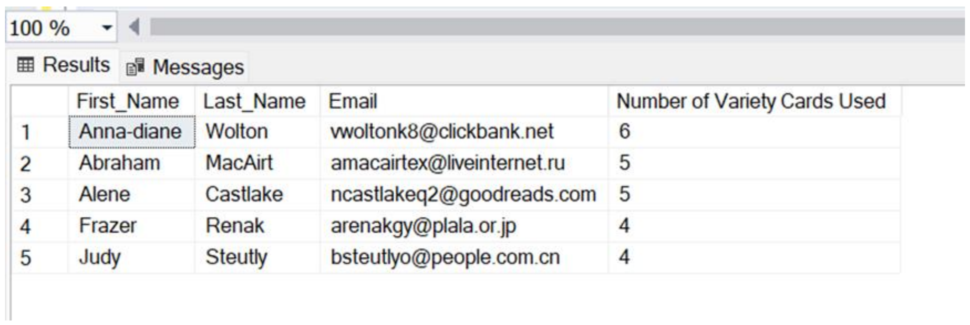
GROUP BY [DB Assignment].[dbo].[Reservation_Fact$].[Email],[DB Assignment].[dbo].[User$].[Last_Name],[DB Assignment].[dbo].[User$].[First_Name]

ORDER BY "Number of Variety Cards Used" DESC

Screenshots of Query & Output



```
SELECT TOP(5) [DB Assignment].[dbo].[User$].[First_Name],

       [DB Assignment].[dbo].[User$].[Last_Name],

       [DB Assignment].[dbo].[Reservation_Fact$].[Email],

       COUNT(DISTINCT [DB Assignment].[dbo].[Reservation_Fact$].[Card_Number]) as "Number of Variety Cards Used"

FROM [DB Assignment].[dbo].[Reservation_Fact$], [DB Assignment].[dbo].[User$]

WHERE [DB Assignment].[dbo].[Reservation_Fact$].[Email]= [DB Assignment].[dbo].[User$].[Email]

GROUP BY [DB Assignment].[dbo].[Reservation_Fact$].[Email],[DB Assignment].[dbo].[User$].[Last_Name],
         [DB Assignment].[dbo].[User$].[First_Name]

ORDER BY "Number of Variety Cards Used" DESC
```


100 %

Results | Messages

	First_Name	Last_Name	Email	Number of Variety Cards Used
1	Anna-diane	Wolton	vwoltonk8@clickbank.net	6
2	Abraham	MacAirt	amacairtex@liveinternet.ru	5
3	Alene	Castlake	ncastlakeq2@goodreads.com	5
4	Frazer	Renak	arenakgy@plala.or.jp	4
5	Judy	Steutly	bsteutlyo@people.com.cn	4

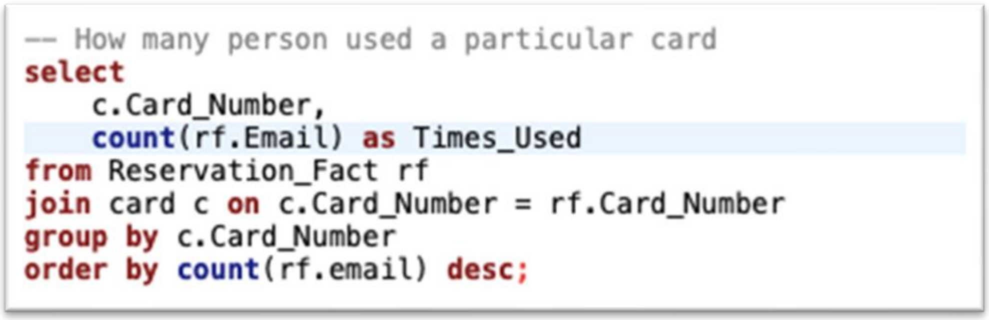
Query 6*: To find how many users have used a particular card*

Business Insights: This query is like the previous query. This query is also intended to identify the anomalies and fraudulent activities and ensure users are updated about the same.

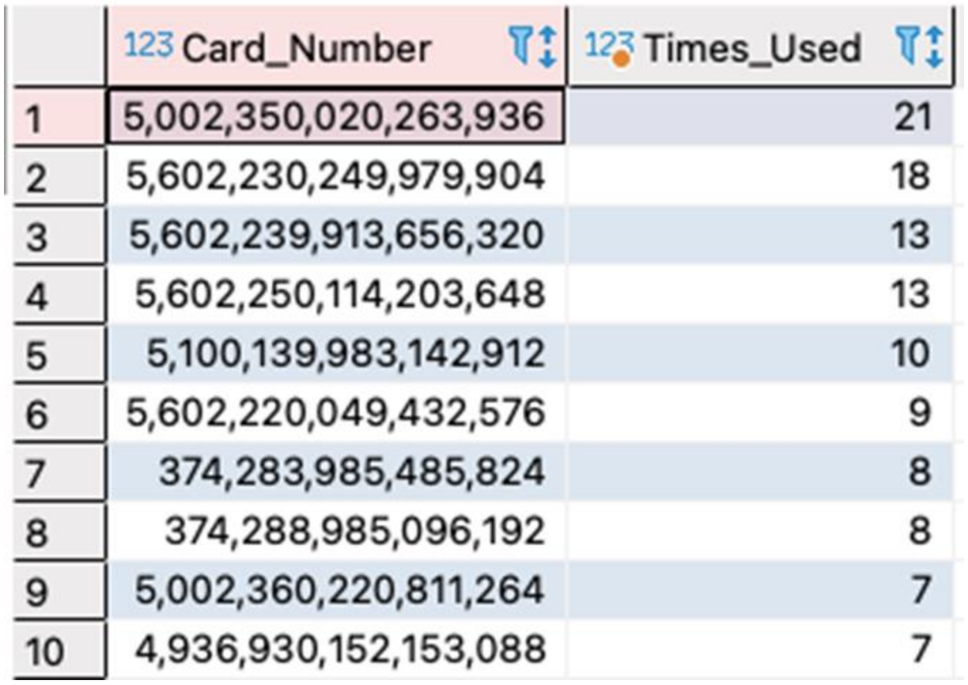
SQL Query:

```
Select  c.Card_Number, count(rf.Email) as Times_Used
from Reservation_Fact rf
join card c on c.Card_Number = rf.Card_Number
group by c.Card_Number
order by count(rf.email) desc;
```

Screenshots of Query & Output



```
-- How many person used a particular card
select
    c.Card_Number,
    count(rf.Email) as Times_Used
from Reservation_Fact rf
join card c on c.Card_Number = rf.Card_Number
group by c.Card_Number
order by count(rf.email) desc;
```



	Card_Number	Times_Used
1	5,002,350,020,263,936	21
2	5,602,230,249,979,904	18
3	5,602,239,913,656,320	13
4	5,602,250,114,203,648	13
5	5,100,139,983,142,912	10
6	5,602,220,049,432,576	9
7	374,283,985,485,824	8
8	374,288,985,096,192	8
9	5,002,360,220,811,264	7
10	4,936,930,152,153,088	7

Query 7: *To find the maximum number of passengers in a single reservation*

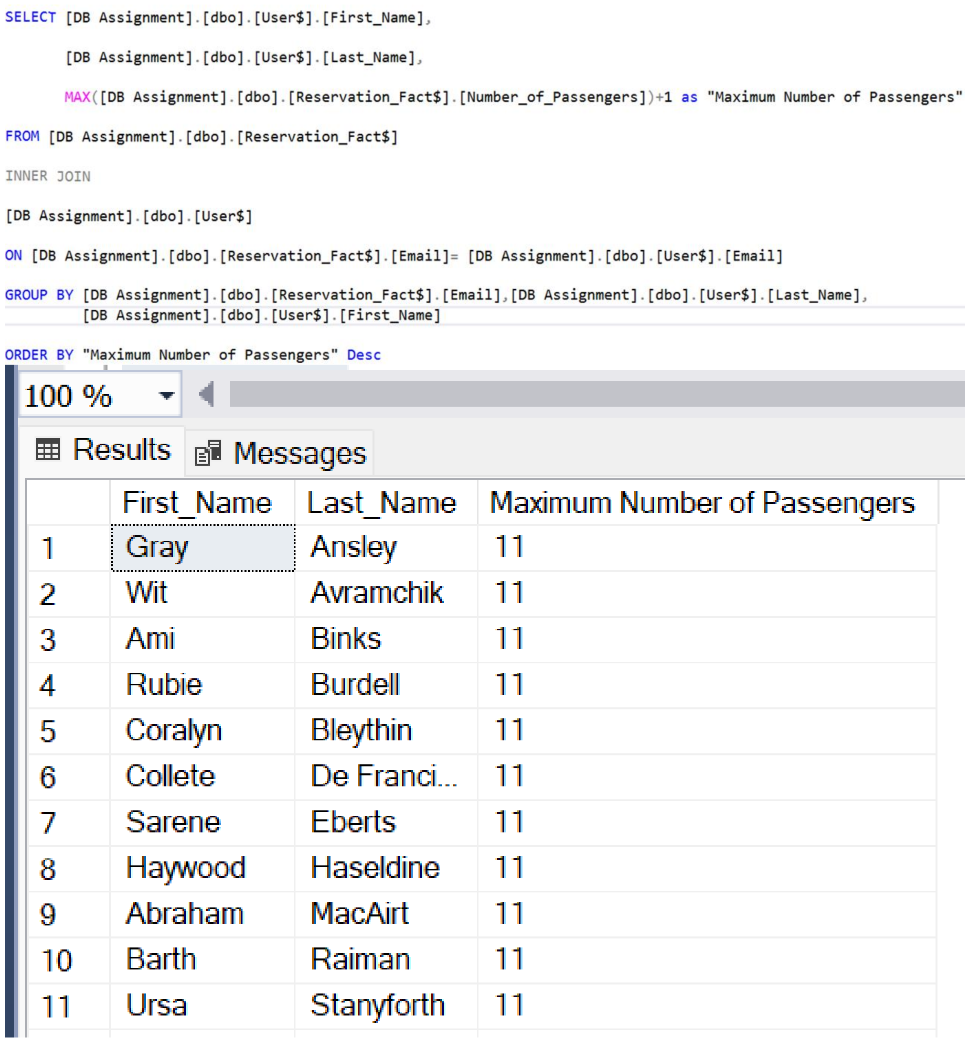
Business Insights: This query gives insights about the maximum number of passengers in a single reservation. To attract users, those who have booked for group of 5 and above can be provided with few discounts in payment.

SQL Query:

```
SELECT [DB Assignment].[dbo].[User$].[First_Name],
  [DB Assignment].[dbo].[User$].[Last_Name],
  MAX([DB Assignment].[dbo].[Reservation_Fact$].[Number_of_Passengers])+1 as "Maximum
Number of Passengers"

FROM [DB Assignment].[dbo].[Reservation_Fact$]

INNER JOIN

[DB Assignment].[dbo].[User$]

ON [DB Assignment].[dbo].[Reservation_Fact$].[Email]= [DB Assignment].[dbo].[User$].[Email]

GROUP BY [DB Assignment].[dbo].[Reservation_Fact$].[Email],
        [DB Assignment].[dbo].[User$].[Last_Name],
 [DB Assignment].[dbo].[User$].[First_Name]
```

ORDER BY "Maximum Number of Passengers" Desc

Screenshots of Query & Output



```
SELECT [DB Assignment].[dbo].[User$].[First_Name],

       [DB Assignment].[dbo].[User$].[Last_Name],

       MAX([DB Assignment].[dbo].[Reservation_Fact$].[Number_of_Passengers])+1 as "Maximum Number of Passengers"

FROM [DB Assignment].[dbo].[Reservation_Fact$]

INNER JOIN

[DB Assignment].[dbo].[User$]

ON [DB Assignment].[dbo].[Reservation_Fact$].[Email]= [DB Assignment].[dbo].[User$].[Email]

GROUP BY [DB Assignment].[dbo].[Reservation_Fact$].[Email],[DB Assignment].[dbo].[User$].[Last_Name],
         [DB Assignment].[dbo].[User$].[First_Name]

ORDER BY "Maximum Number of Passengers" Desc
```

	First_Name	Last_Name	Maximum Number of Passengers
1	Gray	Ansley	11
2	Wit	Avramchik	11
3	Ami	Binks	11
4	Rubie	Burdell	11
5	Coralyn	Bleythin	11
6	Collete	De Franci...	11
7	Sarene	Eberts	11
8	Haywood	Haseldine	11
9	Abraham	MacAirt	11
10	Barth	Raiman	11
11	Ursa	Stanyforth	11

Query 8*: To find from which country, the maximum and minimum number of reservations have been made*

Business Insights: This query gives insights about the country-wise reservations. This result table shows from which country maximum profit is generated and from which country minimum profit is generated. This result can help in make decisions to ensure the minimum profit generated countries also progress.

SQL Query:

SELECT [DB Assignment].[dbo].[User$].Country, COUNT([DB Assignment].[dbo].[User$].Country) "Country-wise Reservations"

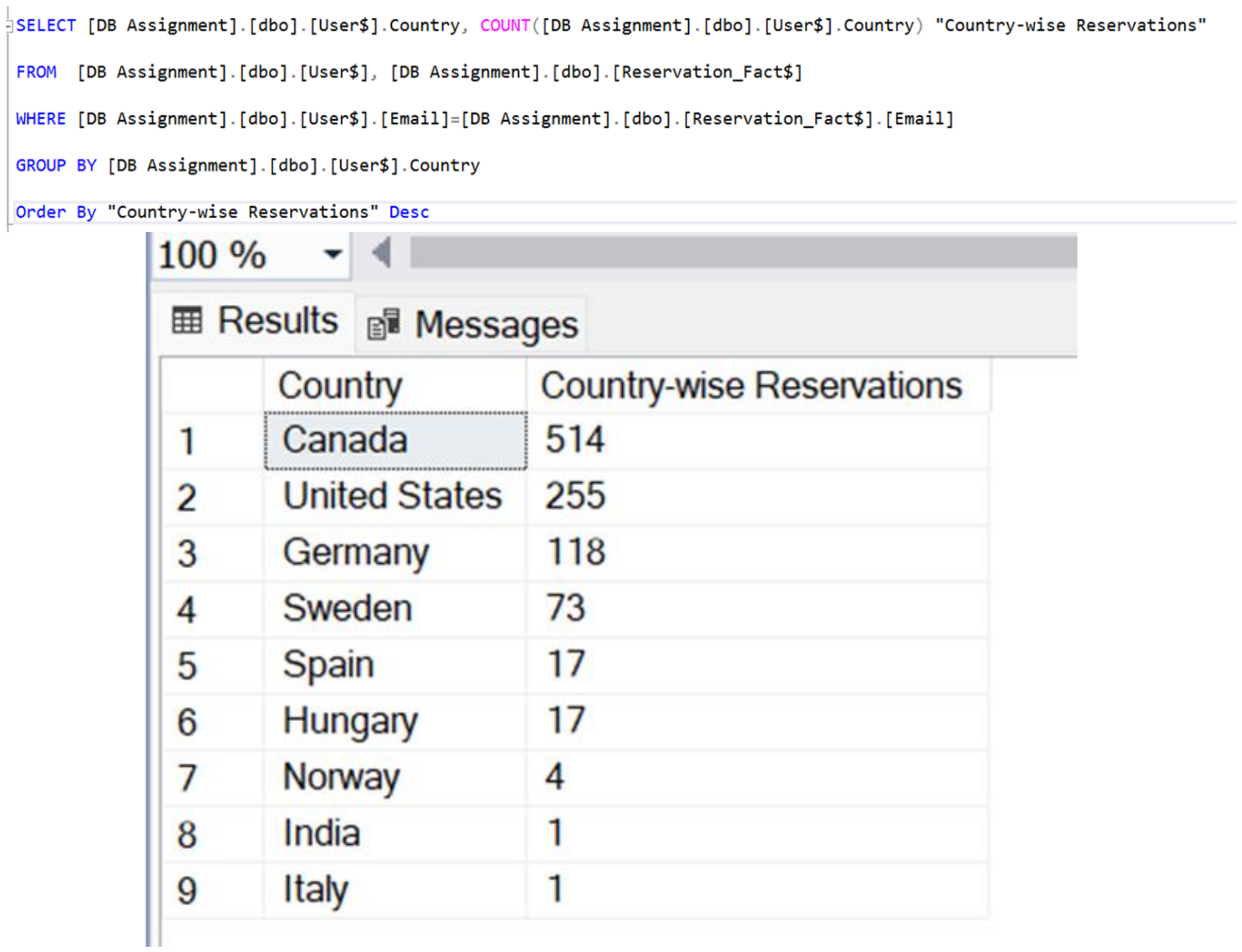
FROM [DB Assignment].[dbo].[User$], [DB Assignment].[dbo].[Reservation_Fact$]

WHERE [DB Assignment].[dbo].[User$].[Email]=[DB Assignment].[dbo].[Reservation_Fact$].[Email]

GROUP BY [DB Assignment].[dbo].[User$].Country

Order By "Country-wise Reservations" Desc

Screenshots of Query & Output



```
SELECT [DB Assignment].[dbo].[User$].Country, COUNT([DB Assignment].[dbo].[User$].Country) "Country-wise Reservations"

FROM  [DB Assignment].[dbo].[User$], [DB Assignment].[dbo].[Reservation_Fact$]

WHERE [DB Assignment].[dbo].[User$].[Email]=[DB Assignment].[dbo].[Reservation_Fact$].[Email]

GROUP BY [DB Assignment].[dbo].[User$].Country

Order By "Country-wise Reservations" Desc
```

	Country	Country-wise Reservations
1	Canada	514
2	United States	255
3	Germany	118
4	Sweden	73
5	Spain	17
6	Hungary	17
7	Norway	4
8	India	1
9	Italy	1

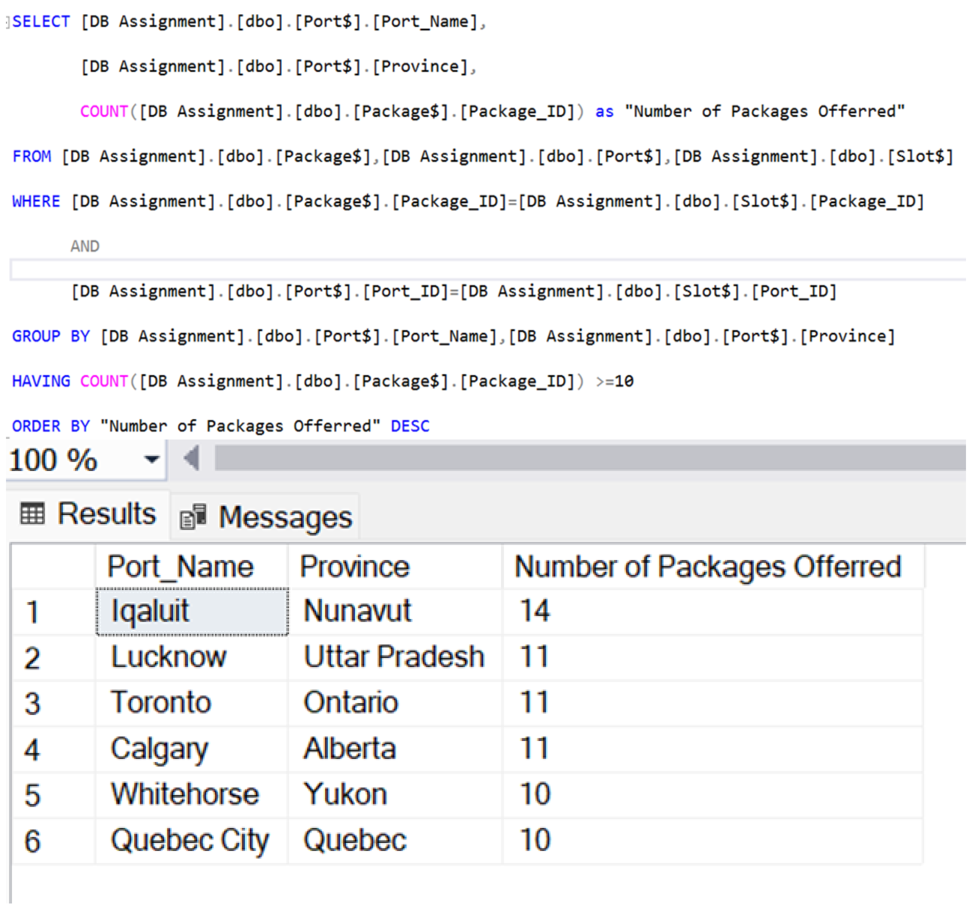
Query 9*: To display those ports which offer at least 10 packages*

Business Insights: This query gives insights about those ports which offer maximum packages. Company can try to increase the packages offered from those ports which have less than 10 packages, to increase their profit margins.

SQL Query:

```
SELECT [DB Assignment].[dbo].[Port$].[Port_Name],

      [DB Assignment].[dbo].[Port$].[Province],

   COUNT([DB Assignment].[dbo].[Package$].[Package_ID]) as "Number of Packages Offerred"

FROM [DB Assignment].[dbo].[Package$],[DB Assignment].[dbo].[Port$],[DB
Assignment].[dbo].[Slot$]

WHERE [DB Assignment].[dbo].[Package$].[Package_ID]=[DB
Assignment].[dbo].[Slot$].[Package_ID]

  AND

     [DB Assignment].[dbo].[Port$].[Port_ID]=[DB Assignment].[dbo].[Slot$].[Port_ID]

GROUP BY [DB Assignment].[dbo].[Port$].[Port_Name],[DB
Assignment].[dbo].[Port$].[Province]

HAVING COUNT([DB Assignment].[dbo].[Package$].[Package_ID]) >5

ORDER BY "Number of Packages Offerred" DESC
```

Screenshots of Query & Output



```
SELECT [DB Assignment].[dbo].[Port$].[Port_Name],
       [DB Assignment].[dbo].[Port$].[Province],
       COUNT([DB Assignment].[dbo].[Package$].[Package_ID]) as "Number of Packages Offerred"
FROM [DB Assignment].[dbo].[Package$],[DB Assignment].[dbo].[Port$],[DB Assignment].[dbo].[Slot$]
WHERE [DB Assignment].[dbo].[Package$].[Package_ID]=[DB Assignment].[dbo].[Slot$].[Package_ID]
      AND
      [DB Assignment].[dbo].[Port$].[Port_ID]=[DB Assignment].[dbo].[Slot$].[Port_ID]
GROUP BY [DB Assignment].[dbo].[Port$].[Port_Name],[DB Assignment].[dbo].[Port$].[Province]
HAVING COUNT([DB Assignment].[dbo].[Package$].[Package_ID]) >=10
ORDER BY "Number of Packages Offerred" DESC
```

	Port_Name	Province	Number of Packages Offerred
1	Iqaluit	Nunavut	14
2	Lucknow	Uttar Pradesh	11
3	Toronto	Ontario	11
4	Calgary	Alberta	11
5	Whitehorse	Yukon	10
6	Quebec City	Quebec	10

Query 10*: Based on the reservations made until now, the total amount gained*

Business Insights: This query gives insights about the total revenue generated. Based on the performance, company can decide on how to proceed to ensure constant increase in the revenue.

SQL Query:

SELECT SUM([DB Assignment].[dbo].[Package$].[Price]) as "Total Amount Gained"

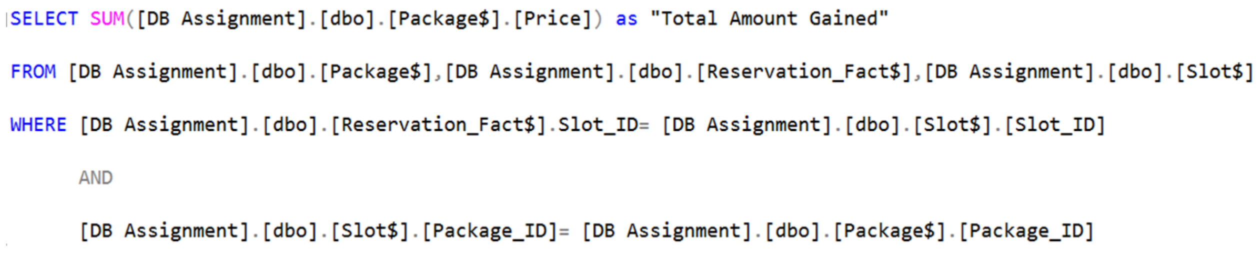
FROM [DB Assignment].[dbo].[Package$],[DB Assignment].[dbo].[Reservation_Fact$],[DB Assignment].[dbo].[Slot$]

WHERE [DB Assignment].[dbo].[Reservation_Fact$].Slot_ID= [DB Assignment].[dbo].[Slot$].[Slot_ID]

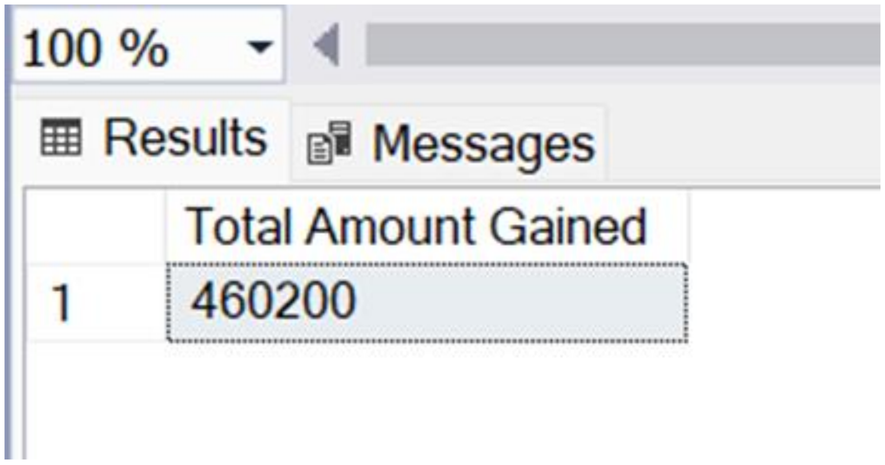
AND

[DB Assignment].[dbo].[Slot$].[Package_ID]= [DB Assignment].[dbo].[Package$].[Package_ID]

Screenshots of Query & Output



```
SELECT SUM([DB Assignment].[dbo].[Package$].[Price]) as "Total Amount Gained"

FROM [DB Assignment].[dbo].[Package$],[DB Assignment].[dbo].[Reservation_Fact$],[DB Assignment].[dbo].[Slot$]

WHERE [DB Assignment].[dbo].[Reservation_Fact$].Slot_ID= [DB Assignment].[dbo].[Slot$].[Slot_ID]

      AND

      [DB Assignment].[dbo].[Slot$].[Package_ID]= [DB Assignment].[dbo].[Package$].[Package_ID]
```



	Total Amount Gained
1	460200

Query 11*: Filter search those package destinations with the letter ‘o’ with their seats available for next one month from three different ports*

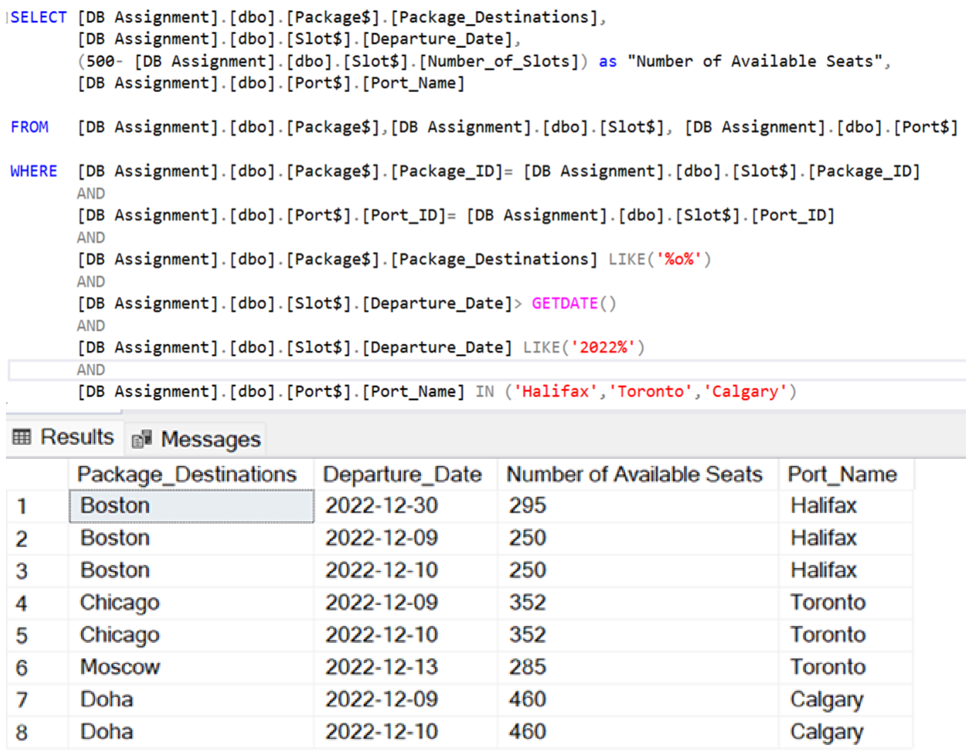
Business Insights: This query is to demonstrate, privilege provided to employees to retrieve data easily. Employees can search for certain places with their letters alone, rather than entering the entire details. This query also intends to show the status of the upcoming trips from three different ports namely Halifax, Toronto and Calgary.

SQL Query:

SELECT [DB Assignment].[dbo].[Package$].[Package_Destinations],
  [DB Assignment].[dbo].[Slot$].[Departure_Date],
  (500- [DB Assignment].[dbo].[Slot$].[Number_of_Slots]) as "Number of Available Seats",
  [DB Assignment].[dbo].[Port$].[Port_Name]

```
FROM   [DB Assignment].[dbo].[Package$],[DB Assignment].[dbo].[Slot$], [DB
Assignment].[dbo].[Port$]

WHERE  [DB Assignment].[dbo].[Package$].[Package_ID]= [DB
Assignment].[dbo].[Slot$].[Package_ID]
     AND
  [DB Assignment].[dbo].[Port$].[Port_ID]= [DB Assignment].[dbo].[Slot$].[Port_ID]
  AND
  [DB Assignment].[dbo].[Package$].[Package_Destinations] LIKE('%o%')
  AND
  [DB Assignment].[dbo].[Slot$].[Departure_Date]> GETDATE()
  AND
  [DB Assignment].[dbo].[Slot$].[Departure_Date] LIKE('2022%')
  AND
  [DB Assignment].[dbo].[Port$].[Port_Name] IN ('Halifax','Toronto','Calgary')
```

Screenshots of Query & Output



```
SELECT [DB Assignment].[dbo].[Package$].[Package_Destinations],
       [DB Assignment].[dbo].[Slot$].[Departure_Date],
       (500- [DB Assignment].[dbo].[Slot$].[Number_of_Slots]) as "Number of Available Seats",
       [DB Assignment].[dbo].[Port$].[Port_Name]

FROM   [DB Assignment].[dbo].[Package$],[DB Assignment].[dbo].[Slot$], [DB Assignment].[dbo].[Port$]

WHERE  [DB Assignment].[dbo].[Package$].[Package_ID]= [DB Assignment].[dbo].[Slot$].[Package_ID]
       AND
       [DB Assignment].[dbo].[Port$].[Port_ID]= [DB Assignment].[dbo].[Slot$].[Port_ID]
       AND
       [DB Assignment].[dbo].[Package$].[Package_Destinations] LIKE('%o%')
       AND
       [DB Assignment].[dbo].[Slot$].[Departure_Date]> GETDATE()
       AND
       [DB Assignment].[dbo].[Slot$].[Departure_Date] LIKE('2022%')
       AND
       [DB Assignment].[dbo].[Port$].[Port_Name] IN ('Halifax','Toronto','Calgary')
```

	Package_Destinations	Departure_Date	Number of Available Seats	Port_Name
1	Boston	2022-12-30	295	Halifax
2	Boston	2022-12-09	250	Halifax
3	Boston	2022-12-10	250	Halifax
4	Chicago	2022-12-09	352	Toronto
5	Chicago	2022-12-10	352	Toronto
6	Moscow	2022-12-13	285	Toronto
7	Doha	2022-12-09	460	Calgary
8	Doha	2022-12-10	460	Calgary

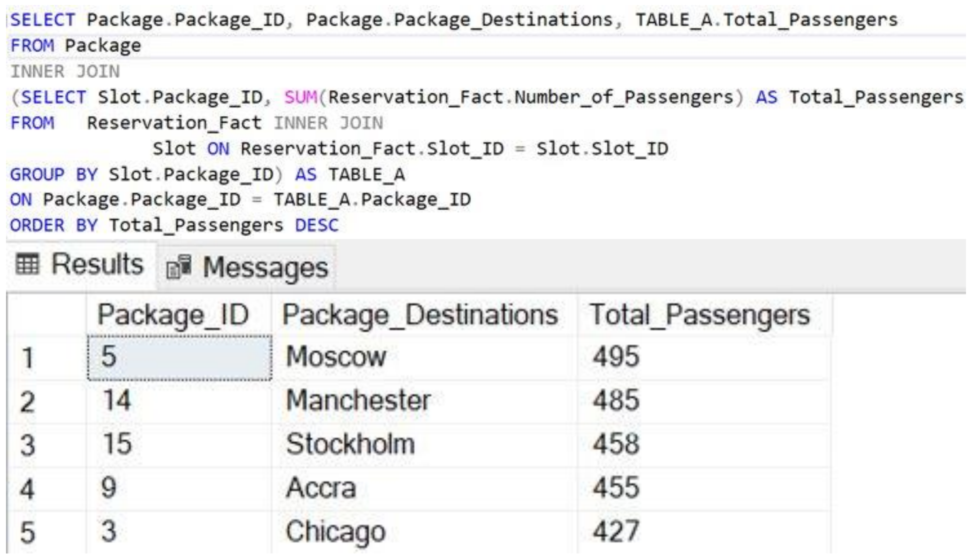
Query 12*: Listing of Packages in terms of popularity defined by total passengers.*

Business Insights: This query finds out which packages are more popular amongst customers and maybe invest more towards them.

SQL Query:

```
SELECT Package.Package_ID, Package.Package_Destinations, TABLE_A.Total_Passengers
FROM Package
INNER JOIN
(SELECT Slot.Package_ID, SUM(Reservation_Fact.Number_of_Passengers) AS Total_Passengers
FROM   Reservation_Fact INNER JOIN
          Slot ON Reservation_Fact.Slot_ID = Slot.Slot_ID
GROUP BY Slot.Package_ID) AS TABLE_A
ON Package.Package_ID = TABLE_A.Package_ID
ORDER BY Total_Passengers DESC
```

Screenshots of Query & Output



```
SELECT Package.Package_ID, Package.Package_Destinations, TABLE_A.Total_Passengers
FROM Package
INNER JOIN
(SELECT Slot.Package_ID, SUM(Reservation_Fact.Number_of_Passengers) AS Total_Passengers
FROM   Reservation_Fact INNER JOIN
          Slot ON Reservation_Fact.Slot_ID = Slot.Slot_ID
GROUP BY Slot.Package_ID) AS TABLE_A
ON Package.Package_ID = TABLE_A.Package_ID
ORDER BY Total_Passengers DESC
```

	Package_ID	Package_Destinations	Total_Passengers
1	5	Moscow	495
2	14	Manchester	485
3	15	Stockholm	458
4	9	Accra	455
5	3	Chicago	427

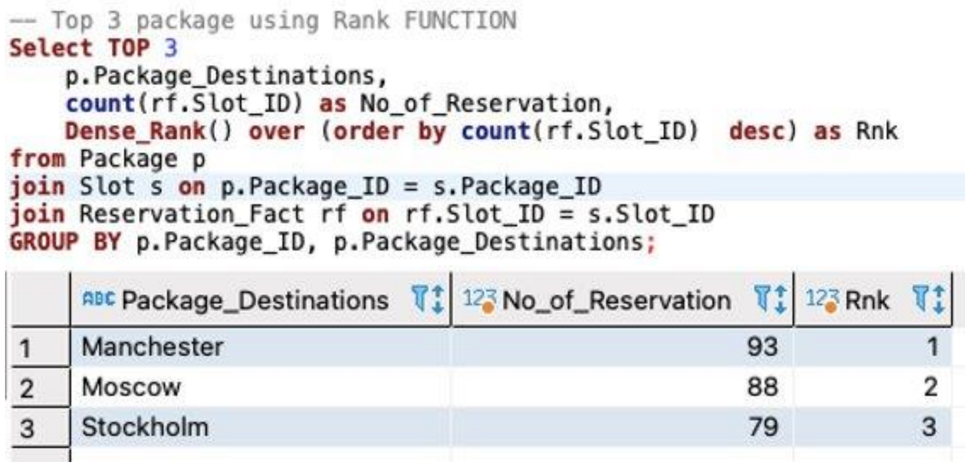
Query 13*: Filter Top 3 popular package using Rank Function.*

Business Insights: This query finds out top 3 packages in terms of total reservations. It can be used to determine lucrative packages for the company.

SQL Query:

```
Select TOP 3
 p.Package_Destinations,
 count(rf.Slot_ID) as No_of_Reservation,
 Dense_Rank() over (order by count(rf.Slot_ID) desc) as Rnk
from
 Package p
join Slot s on p.Package_ID = s.Package_ID
join Reservation_Fact rf on rf.Slot_ID = s.Slot_ID
GROUP BY p.Package_ID, p.Package_Destinations;
```

Screenshots of Query & Output



```
-- Top 3 package using Rank FUNCTION
Select TOP 3
    p.Package_Destinations,
    count(rf.Slot_ID) as No_of_Reservation,
    Dense_Rank() over (order by count(rf.Slot_ID)  desc) as Rnk
from Package p
join Slot s on p.Package_ID = s.Package_ID
join Reservation_Fact rf on rf.Slot_ID = s.Slot_ID
GROUP BY p.Package_ID, p.Package_Destinations;
```

	Package_Destinations	No_of_Reservation	Rnk
1	Manchester	93	1
2	Moscow	88	2
3	Stockholm	79	3

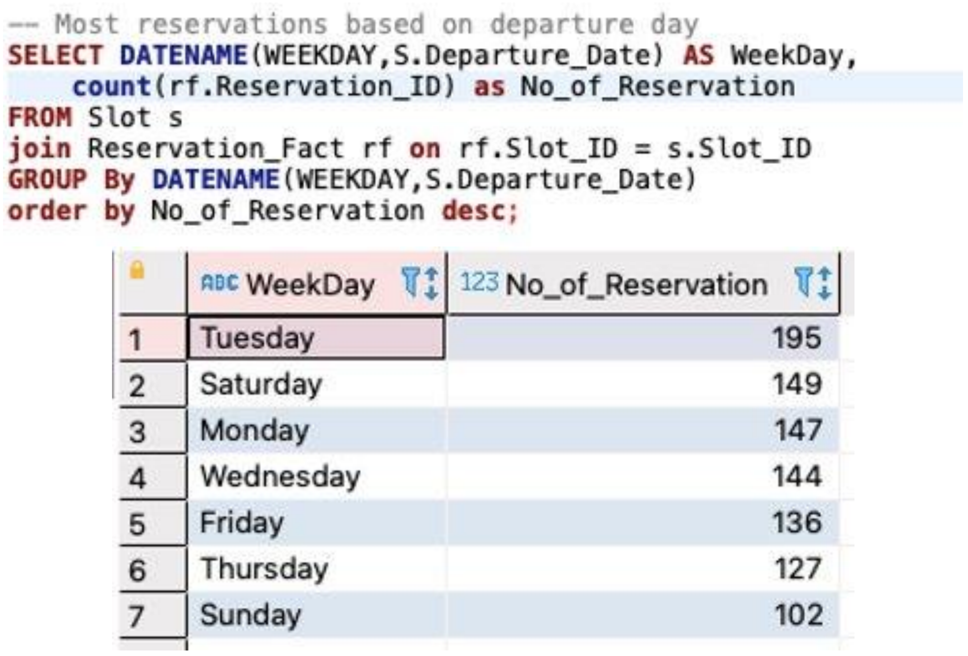
Query 14*: Filter Most reservations based on day of the week.*

Business Insights: This query finds out number of reservations for each day of the week. This can determine which days are more popular amongst customers.

SQL Query:

```
SELECT DATENAME(WEEKDAY,S.Departure_Date) AS WeekDay, count(rf.Reservation_ID) as
No_of_Reservation
FROM Slot s
join Reservation_Fact rf on rf.Slot_ID = s.Slot_ID
GROUP By DATENAME(WEEKDAY,S.Departure_Date)
order by No_of_Reservation desc;
```

Screenshots of Query & Output



```
-- Most reservations based on departure day
SELECT DATENAME(WEEKDAY,S.Departure_Date) AS WeekDay,
    count(rf.Reservation_ID) as No_of_Reservation
FROM Slot s
join Reservation_Fact rf on rf.Slot_ID = s.Slot_ID
GROUP By DATENAME(WEEKDAY,S.Departure_Date)
order by No_of_Reservation desc;
```

	WeekDay	No_of_Reservation
1	Tuesday	195
2	Saturday	149
3	Monday	147
4	Wednesday	144
5	Friday	136
6	Thursday	127
7	Sunday	102

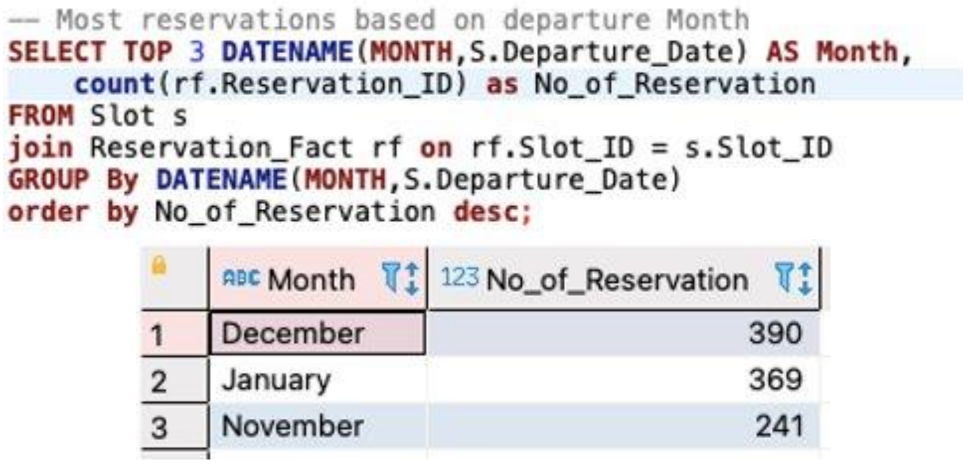
Query 15*: Filter Most reservations based on departure Month.*

Business Insights: This query finds out number of reservations for each month. This can determine which months are more popular amongst customers.

SQL Query:

SELECT TOP 3 DATENAME(MONTH,S.Departure_Date) AS Month, count(rf.Reservation_ID) as
No_of_Reservation
FROM Slot s
join Reservation_Fact rf on rf.Slot_ID = s.Slot_ID
GROUP By DATENAME(MONTH,S.Departure_Date)
order by No_of_Reservation desc;

Screenshots of Query & Output



```
-- Most reservations based on departure Month
SELECT TOP 3 DATENAME(MONTH,S.Departure_Date) AS Month,
    count(rf.Reservation_ID) as No_of_Reservation
FROM Slot s
join Reservation_Fact rf on rf.Slot_ID = s.Slot_ID
GROUP By DATENAME(MONTH,S.Departure_Date)
order by No_of_Reservation desc;
```

	Month	No_of_Reservation
1	December	390
2	January	369
3	November	241

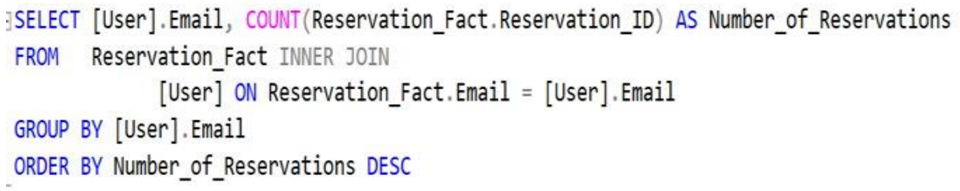
***Query 16**: Listing of Users in sorted order by number of bookings.*

Business Insights: This query finds out number of reservations for each user in sorted order. This can determine which users have booked most reservations with company and give them discounts/coupons.

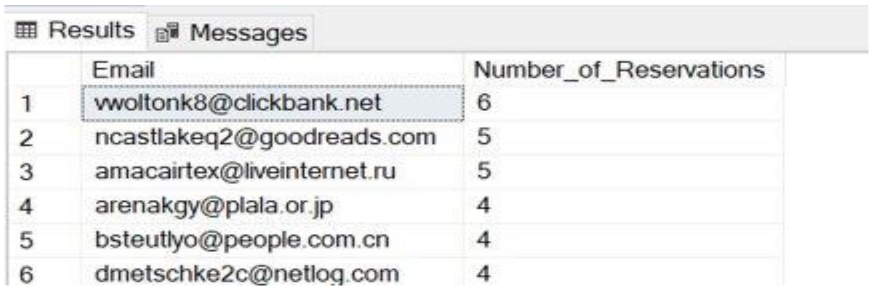
SQL Query:

```
SELECT [User].Email, COUNT(Reservation_Fact.Reservation_ID) AS Number_of_Reservations
FROM   Reservation_Fact INNER JOIN
          [User] ON Reservation_Fact.Email = [User].Email
GROUP BY [User].Email
ORDER BY Number_of_Reservations DESC
```

Screenshots of Query & Output



```
SELECT [User].Email, COUNT(Reservation_Fact.Reservation_ID) AS Number_of_Reservations
FROM   Reservation_Fact INNER JOIN
          [User] ON Reservation_Fact.Email = [User].Email
GROUP BY [User].Email
ORDER BY Number_of_Reservations DESC
```



Results Messages

	Email	Number_of_Reservations
1	vwoltonk8@clickbank.net	6
2	ncastlakeq2@goodreads.com	5
3	amacairtex@liveinternet.ru	5
4	arenakgy@plala.or.jp	4
5	bsteutlyo@people.com.cn	4
6	dmetschke2c@netlog.com	4

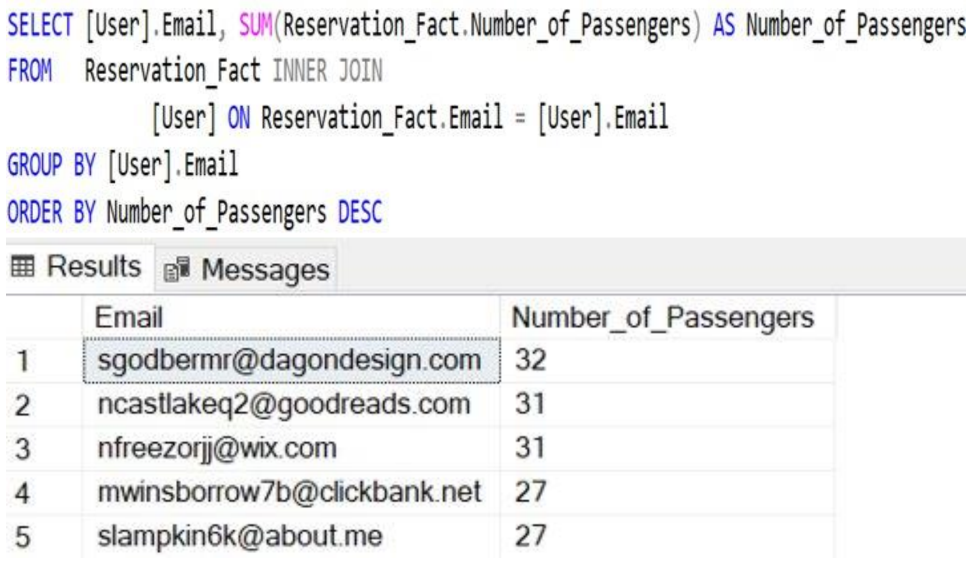
***Query 17**: Listing of Users in sorted order with most passenger bookings.*

Business Insights: This query finds out total seats booked by each user in sorted order. This can determine which users have booked most number of seats with company and give them discounts/coupons.

SQL Query:

```
SELECT [User].Email,SUM(Reservation_Fact.Number_of_Passengers) AS
Number_of_Passengers
FROM   Reservation_Fact INNER JOIN
           [User] ON Reservation_Fact.Email = [User].Email
GROUP BY [User].Email
ORDER BY Number_of_Passengers DESC
```

Screenshots of Query & Output



```
SELECT [User].Email, SUM(Reservation_Fact.Number_of_Passengers) AS Number_of_Passengers
FROM   Reservation_Fact INNER JOIN
           [User] ON Reservation_Fact.Email = [User].Email
GROUP BY [User].Email
ORDER BY Number_of_Passengers DESC
```

	Email	Number_of_Passengers
1	sgodbermr@dagondesign.com	32
2	ncastlakeq2@goodreads.com	31
3	nfreezorjj@wix.com	31
4	mwinsborrow7b@clickbank.net	27
5	slampkin6k@about.me	27

***Query 18**: List of departures with most empty slots. To aid in decision making for discounts.*

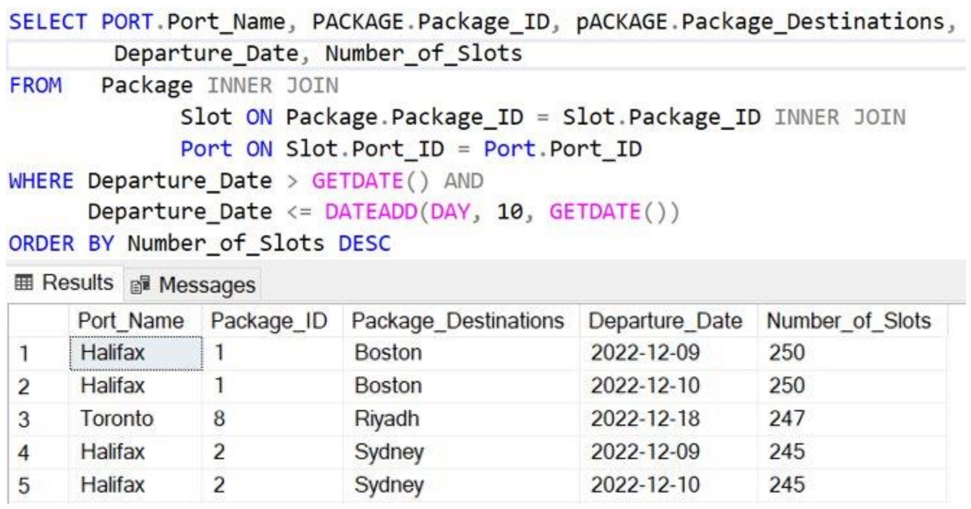
Business Insights: This query lists departures in sorted order based on empty seats. This can be used to determine if discounts or aggressive sales promotion is required for slots with many empty seats.

SQL Query:

```
SELECT PORT.Port_Name, PACKAGE.Package_ID, pACKAGE.Package_Destinations,
```

Departure_Date, Number_of_Slots
FROM   Package INNER JOIN
            Slot ON Package.Package_ID = Slot.Package_ID INNER JOIN
            Port ON Slot.Port_ID = Port.Port_ID
WHERE Departure_Date > GETDATE() AND
  Departure_Date <= DATEADD(DAY, 10, GETDATE())
ORDER BY Number_of_Slots DESC

Screenshots of Query & Output



```
SELECT PORT.Port_Name, PACKAGE.Package_ID, pACKAGE.Package_Destinations,
        Departure_Date, Number_of_Slots
FROM   Package INNER JOIN
            Slot ON Package.Package_ID = Slot.Package_ID INNER JOIN
            Port ON Slot.Port_ID = Port.Port_ID
WHERE Departure_Date > GETDATE() AND
      Departure_Date <= DATEADD(DAY, 10, GETDATE())
ORDER BY Number_of_Slots DESC
```

	Port_Name	Package_ID	Package_Destinations	Departure_Date	Number_of_Slots
1	Halifax	1	Boston	2022-12-09	250
2	Halifax	1	Boston	2022-12-10	250
3	Toronto	8	Riyadh	2022-12-18	247
4	Halifax	2	Sydney	2022-12-09	245
5	Halifax	2	Sydney	2022-12-10	245

***Query 19**: Top 10 users and there booking details for personalization and offers.*

Business Insights: This query lists top 10 customers who have booked the most with the company and their booking details like destination, package etc. This can be used to give them personalized recommendations.

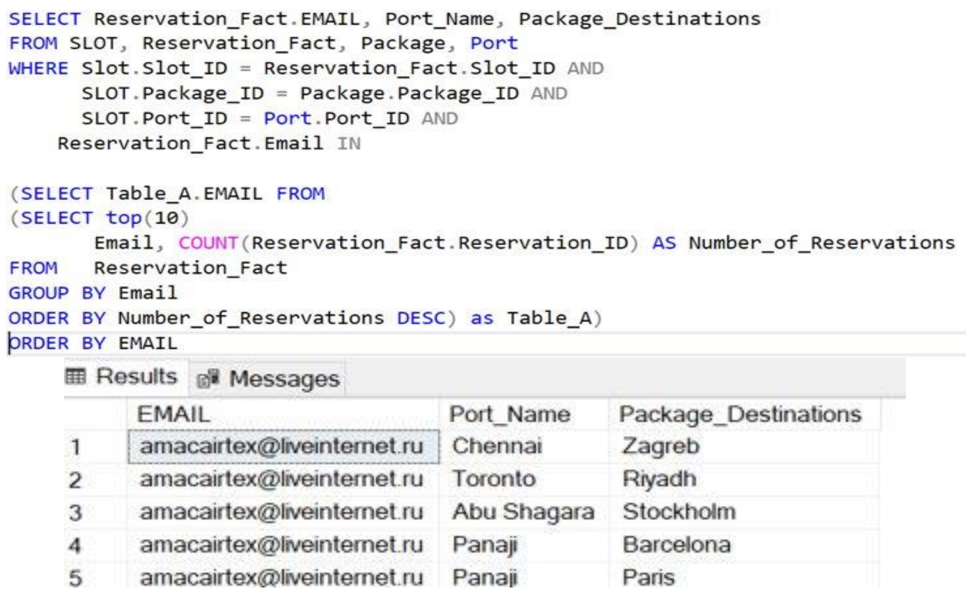
SQL Query:

SELECT Reservation_Fact.EMAIL, Port_Name, Package_Destinations
FROM SLOT, Reservation_Fact, Package, Port
WHERE Slot.Slot_ID = Reservation_Fact.Slot_ID AND
  SLOT.Package_ID = Package.Package_ID AND
  SLOT.Port_ID = Port.Port_ID AND
Reservation_Fact.Email IN

(SELECT Table_A.EMAIL FROM

```
(SELECT top(10)
   Email, COUNT(Reservation_Fact.Reservation_ID) AS Number_of_Reservations
FROM   Reservation_Fact
GROUP BY Email
ORDER BY Number_of_Reservations DESC) as Table_A)
ORDER BY EMAIL
```

Screenshots of Query & Output

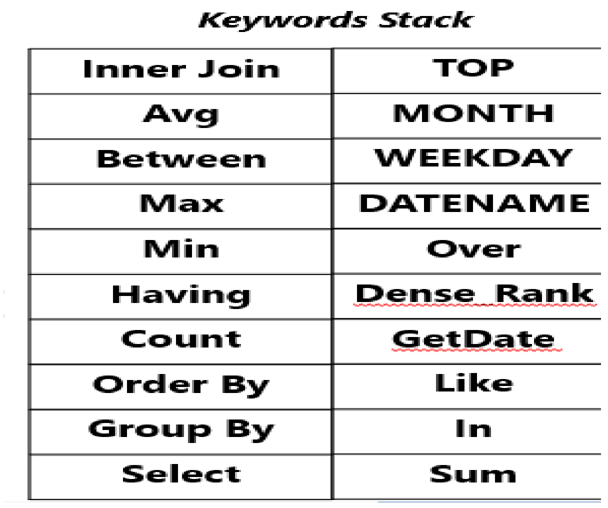


```
SELECT Reservation_Fact.EMAIL, Port_Name, Package_Destinations
FROM SLOT, Reservation_Fact, Package, Port
WHERE Slot.Slot_ID = Reservation_Fact.Slot_ID AND
      SLOT.Package_ID = Package.Package_ID AND
      SLOT.Port_ID = Port.Port_ID AND
    Reservation_Fact.Email IN

(SELECT Table_A.EMAIL FROM
(SELECT top(10)
       Email, COUNT(Reservation_Fact.Reservation_ID) AS Number_of_Reservations
FROM   Reservation_Fact
GROUP BY Email
ORDER BY Number_of_Reservations DESC) as Table_A)
ORDER BY EMAIL
```

	EMAIL	Port_Name	Package_Destinations
1	amacairtex@liveinternet.ru	Chennai	Zagreb
2	amacairtex@liveinternet.ru	Toronto	Riyadh
3	amacairtex@liveinternet.ru	Abu Shagara	Stockholm
4	amacairtex@liveinternet.ru	Panaji	Barcelona
5	amacairtex@liveinternet.ru	Panaji	Paris

LIST OF IMPORTANT KEYWORDS USED

We have used several keywords throughout the 19 queries. The keyword stack is provided below.



Keywords Stack

Inner Join	TOP
Avg	MONTH
Between	WEEKDAY
Max	DATENAME
Min	Over
Having	Dense_Rank
Count	GetDate
Order By	Like
Group By	In
Select	Sum

TEAMWORK

The Favorite part of the group project is Teamwork, hereby discussed the roles and responsibilities each member has taken and worked on

Task	Description	Aravind Gopi	Francis Kuzhippallil	Ajay Jain
Design Schema	Designing the schema and relationship between the tables, make sure the database is in 4th Normal Form.	✓	✓	✓
Logical ER Diagram	Creation of Logical Entity Relationship Diagram.	✓		
Physical ER Diagram	Creation of Physical Entity Relationship Diagram		✓	
Schema Creation	Creating the schema in MSSQL Server			✓
Mock Data Generation	Generation of mock data to perform analytical queries	✓	✓	✓
Analytical Queries	Creation of analytical queries to help employees of *Seagull* to analyse the business. Each person came up with 5 queries.	✓	✓	✓
Making Presentation	Creation of the ppt for presenting our work	✓	✓	✓
Making Report	Creation of the documentation report	✓	✓	✓