# Basic Databases – Report02

Wroclaw University of Science and Technology, Date: March 16, 2022

| Student: | ---------- | Grade |
|---|---|---|
| Identifier | 245784 | |
| First name | Rahul | |
| Last name | Vijayvargiya | |

*This laboratory assignment consists of 1 task. If you cannot solve the task, try to give at least a partial solution or justification for the reason for the lack of a solution.*

Source of data: **AdventureWorks**

**Task 1**

Data Source: SalesOrderHeader

Define the following queries:

1.1 Specify the years in which the orders were registered in the database.

1.2 Create a list of orders placed in the first year of order registration (ID, Year, Order Amount).

1.3 Create a list of orders placed in May in individual years (year, month, ID, Order amount)

1.1. SQL query + fragment of the result (4 records from ?)

**Table 1.1 Fragment of the query results for task 1.1**

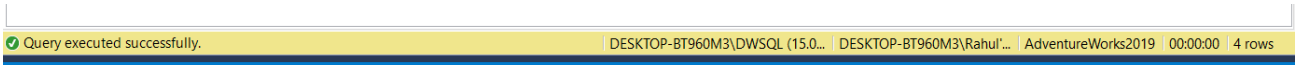
| Lata |
|---|
| 2012 |
| 2013 |
| … |

Rec: 2/?

**Solution:**

```
SELECT DISTINCT Year(OrderDate) AS "Year"
FROM [Sales].[SalesOrderHeader]
ORDER BY "Year" ASC;
```

| Year |
|---|
| 2011 |
| 2012 |
| 2013 |
| 2014 |




---

1.2. SQL query + fragment of the result (4 records from ?)

**Table 1.2 Fragment of the query results for task 1.2**

| Identifier | Year | Amount |
|---|---|---|
| 45266 | 2012 | 27605.63 |

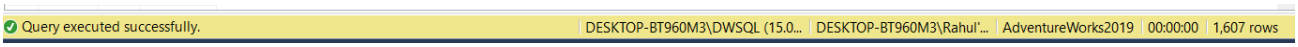
| | | |
|---|---|---|
| 45267 | 2012 | 3899.68 |
| 45268 | 2012 | 944.62 |
| 45269 | 2012 | 2280.14 |

Rec.: 4/?

**Solution:**

```
SELECT SalesOrderID as Identifier, YEAR(OrderDate) as "Year", TotalDue as Amount
FROM Sales.SalesOrderHeader
WHERE YEAR(OrderDate) = (
SELECT MIN(YEAR(OrderDate)) FROM Sales.SalesOrderHeader
   );
```

| Identifier | Year | Amount |
|---|---|---|
| 43659 | 2011 | 23153.2339 |
| 43660 | 2011 | 1457.3288 |
| 43661 | 2011 | 36865.8012 |
| 43662 | 2011 | 32474.9324 |




---

1.3. SQL query + fragment of the result (4 records from ?)

**Table 1.3 Fragment of the query results for task 1.3**

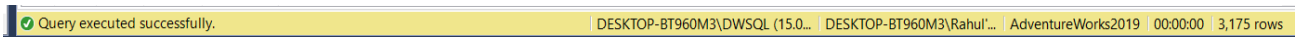
| Year | Month | Identifier | Amount |
|---|---|---|---|
| 2012 | 5 | 46685 | 2410.63 |
| 2012 | 5 | 46686 | 865.20 |
| 2013 | 5 | 50775 | 2264.25 |
| 2013 | 5 | 50776 | 1105.48 |

Rec: 4/?

**Solution:**

```
SELECT YEAR(OrderDate) AS "Year",
MONTH(OrderDate) AS "Month",
SalesOrderID AS "Identifier",
TotalDue AS "Amount"
FROM sales.SalesOrderHeader
WHERE MONTH(OrderDate)=5;
```

| Year | Month | Identifier | Amount |
|---|---|---|---|
| 2011 | 5 | 43659 | 23153.2339 |
| 2011 | 5 | 43660 | 1457.3288 |
| 2011 | 5 | 43661 | 36865.8012 |
| 2011 | 5 | 43662 | 32474.9324 |



---

**Task 2**

2.1. Create a list of customers with more than 25 orders (use CTE). An example of the query result is presented in Table 2.1 below:

**Table 2.1. Fragment of the query results for task 2.1**

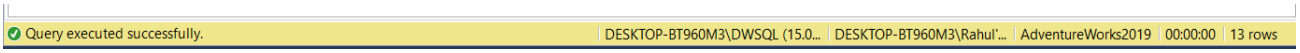
| CustomerId | Last name, First name | Number of orders |
|---|---|---|
| 11091 | Perez, Dalton | 28 |
| 11176 | Roberts, Mason | 28 |
| 11185 | Henderson, Ashley | 27 |
| 11200 | Griffin, Jason | 27 |
| … | … | … |

Rec.: 4/?

**Solution:**

```
WITH clientsCTE ( CustomerId, "Last name, First name", "Number of orders")
AS (
SELECT S.CustomerID as "CustomerId", P.LastName + ', ' + P.FirstName as "last
name,first name",
COUNT(S.SalesOrderID) as "Number Of Orders"
FROM Sales.SalesOrderHeader as S
JOIN Person.Person P ON S.CustomerID = P.BusinessEntityID
GROUP By S.CustomerID, P.LastName, P.FirstName
HAVING COUNT(S.SalesOrderID) > 25
)
SELECT CustomerId, "Last name, First name", "Number Of Orders" FROM clientsCTE ORDER BY 3
DESC;
```

| CustomerId | Last name, First name | Number Of Orders |
|---|---|---|
| 11176 | Miller, Morgan | 28 |
| 11091 | Taylor, Jennifer | 28 |
| 11277 | Vazquez, Ruben | 27 |
| 11262 | Wilson, Natalie | 27 |




---

2.2. Determine what factors affect the number of orders. An example of the query result is presented in Table 2.2 below.

Data Source: SalesOrderHeaderSalesReason, ?

**Table 2.2 Fragment of the query results for task 2.2**

| Factor | Orders |
|---|---|
| Price | 17473 |
| On Promotion | 3515 |
| Manufacturer | 1746 |
| … | … |

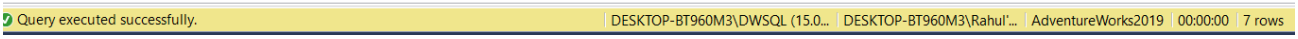
Rec.: 3/?

2.1 SQL query + fragment of the result (4 records from ?)

**Solution:**

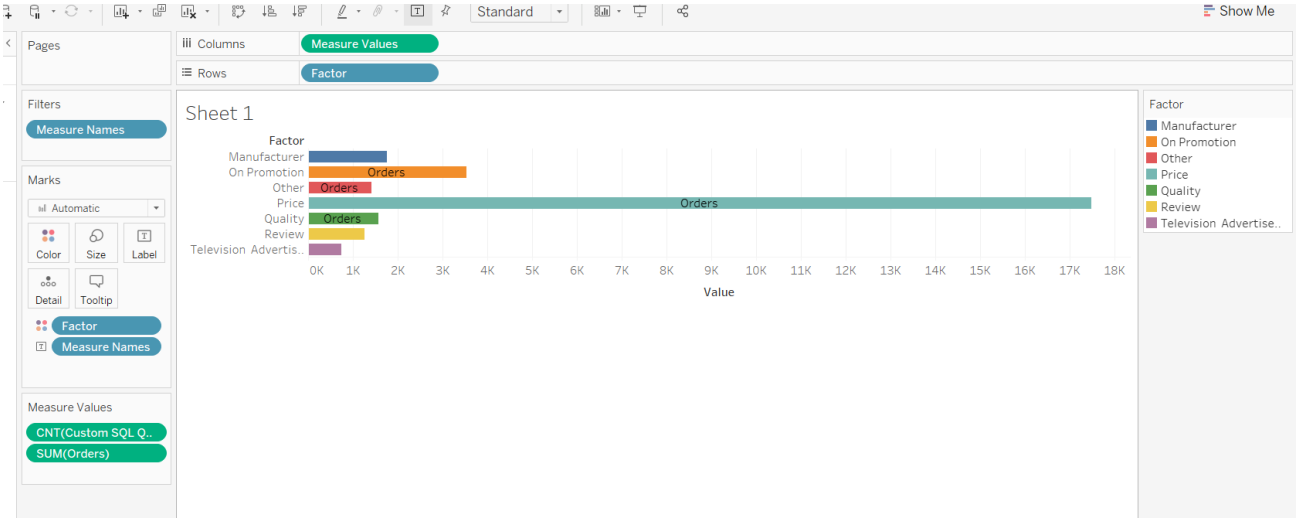
```
SELECT SR.[Name] as "Factor", COUNT(SR.[Name]) as "Orders"
FROM Sales.SalesOrderHeaderSalesReason as OSR
JOIN Sales.SalesReason as SR
ON OSR.SalesReasonID = SR.SalesReasonID
GROUP BY SR.[Name]
ORDER BY COUNT(SR.[Name]) DESC;
```

| Factor | Orders |
|---|---|
| Price | 17473 |
| On Promotion | 3515 |
| Manufacturer | 1746 |
| Quality | 1551 |




2.2 Tableau - the same result in graphical form

**Solution:**



---

**Task 3**

Define a query that determines the sales made by employees to individual customers in the years recorded in the database. An example of the query result is presented in Table 3 below:

**Table 3 Fragment of the query results for task 3**

| SalesPersonID | CustomerID | 2011 | 2012 | 2013 | 2014 |
|---|---|---|---|---|---|
| 274 | 30075 | Lack | Lack | Lack | 29524.05 |
| 274 | 30096 | Lack | 26305.46 | Lack | Lack |
| 275 | 29486 | Lack | Lack | 151107.24 | 53531.92 |
| 275 | 29487 | 37621.78 | 22003.89 | Lack | Lack |
| … | … | … | … | … | … |

Rec.: 4/?

3.1 SQL query + fragment of the result (4 records from ?)

**Solution:**

```
SELECT * FROM (
SELECT SalesPersonId, CustomerID, TotalDue, YEAR(OrderDate) AS "Year"
FROM Sales.SalesOrderHeader
WHERE SalesPersonId IS NOT NULL
GROUP BY CustomerID, SalesPersonId, TotalDue, YEAR(OrderDate)
) As SalesResult
PIVOT (
SUM([TotalDue])
FOR [Year] in ([2011], [2012], [2013], [2014])
) AS P
ORDER BY P.SalesPersonID;
```

| SalesPersonId | CustomerID | 2011 | 2012 | 2013 | 2014 |
|---|---|---|---|---|---|
| 274 | 29491 | NULL | 37625.4303 | NULL | NULL |
| 274 | 29493 | 2417.4793 | NULL | NULL | NULL |
| 274 | 29514 | NULL | NULL | 3931.9148 | NULL |
| 274 | 29523 | NULL | NULL | NULL | 38762.014 |


3.2 Tableau - the same result in graphical form

**Solution:**


---

**Task 4**

Create a pivot table that shows:

4.1 The average annual amount of purchases made by customers in 2013-2014 using the PIVOT operator.

4.2 The average annual amount of purchases made by customers in 2013-2014 without the PIVOT operator.

**Table 4 Fragment of the query results for task 4**

| Name | CustomerID | 2013 | 2014 |
|---|---|---|---|
| Achong, Gustavo | 29484 | 30937.91 | NULL |
| Abel, Catherine | 29485 | 28773.45 | 27670.88 |
| Abercrombie, Kim | 29486 | 37776.81 | 26765.96 |
| Acevedo, Humberto | 29487 | 2461.74 | 465.15 |

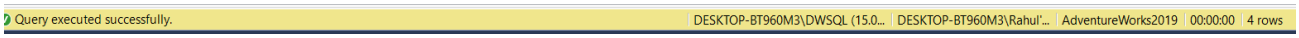
Rec: 4/?

4.3 SQL query + fragment of the result (4 records from ?)

**Solution:**

```
SELECT TOP 4 * FROM (
SELECT so.CustomerID, so.TotalDue,P.LastName+','+P.FirstName as "Lastname, Firstname",
YEAR(OrderDate) AS "Year"
FROM Sales.SalesOrderHeader as so
JOIN Person.Person P on so.CustomerID = P.BusinessEntityID
GROUP BY so.CustomerID, P.LastName,P.FirstName,so.TotalDue, YEAR(OrderDate)
) SalesResult
PIVOT (
AVG([TotalDue])
FOR [Year] in ( [2013], [2014])
) AS P
```

| | | | |
|---|---|---|---|
| 16867 | Adams,Aaron | NULL | 2632.0437 |
| 16901 | Adams,Adam | NULL | 44.1779 |
| 16724 | Adams,Alex | NULL | 44.1779 |
| 16699 | Adams,Angel | 36.023 | 38.6529 |

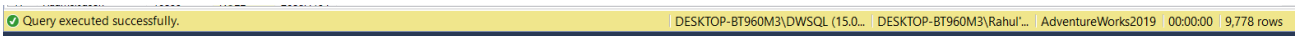



4.4 SQL query + fragment of the result (4 records from ?)

**Solution:**

```
SELECT P.LastName+','+P.FirstName as "Lastname, Firstname",So.CustomerID,
AVG(CASE WHEN YEAR(OrderDate) = 2013 THEN [TotalDue] END) "2013",
AVG(CASE WHEN YEAR(OrderDate) = 2014 THEN [TotalDue] END) "2014"
FROM Sales.SalesOrderHeader as So
JOIN Person.Person P on So.CustomerID = P.BusinessEntityID
GROUP BY So.CustomerID, P.LastName,P.FirstName
ORDER BY P.LastName ASC;
```

| Lastname, Firstname | CustomerID | 2013 | 2014 |
|---|---|---|---|
| Adams,Aaron | 16867 | NULL | 2632.0437 |
| Adams,Adam | 16901 | NULL | 44.1779 |
| Adams,Alex | 16724 | NULL | 44.1779 |
| Adams,Angel | 16699 | 36.023 | 38.6529 |




## **CONCLUSIONS**:

*Use this section to provide your conclusions:*

Queries were Complicated, A lot of expression were used during the lab, Got Familiar with Pivot Operator, CTE

Tried to make data from queries look better but dataset was too small to use proper Visualization techniques, I tried at my end to make it look more appealing

Tried another as well, I tried to Connect Database with python, Did some analysis with python as well

---

## **REMARKS:**

- *A report without final conclusions will not be checked and results in a negative score!*