# Basic Databases – Report04

Wroclaw University of Technology, Date: November 23, 2018

| Student: | Email: 242363@student.pwr.edu.pl | Grade |
|---|---|---|
| Identifier | 242363 | |
| First name | Patryk | |
| Last name | Szwed | |

This package consists of 6 tasks. If you cannot solve a complete problem, try to give a partial answer. Remember to write your name, and identifier.

The tasks contain exercises to practice two subjects:

1. Retrieving Information from Many Tables using **common table expressions** **CTE**

*Note: CTE facilitates solving of the following tasks*

Source data: **Database: AdventureWorks2017**

## Task 1

Create reports according to the following definition:

- **City, Number of orders**

Table: `Person.Address, …?`

- Create a list of cities with the number of completed orders
- Create a list of cities to which were supplied more than 500 orders

**Solution**

Queries for these tasks are:

- Create a list of cities with the number of completed orders

```
WITH CityOrders AS (
	SELECT City, COUNT(*) AS "Number of completed orders"
	FROM Person.Address
		INNER JOIN Sales.SalesOrderHeader ON AddressID = BillToAddressID
	GROUP BY City
)
SELECT * FROM CityOrders;
```

Partial result is:

| City | Number of completed orders |
|---|---|
| Cheltenham | 79 |
| Braunschweig | 48 |
| Reading | 51 |
| Suresnes | 43 |
| Fontana | 4 |

| | |
|---|---|
| Sooke | 190 |
| Newark | 8 |
| Modesto | 16 |
| Cheyenne | 17 |
| Oxnard | 8 |

There are 535 results.

- Create a list of cities to which were supplied more than 500 orders

```
WITH CityOrders AS (
	SELECT City, COUNT(*) AS "Number of completed orders"
	FROM Person.Address
		INNER JOIN Sales.SalesOrderHeader ON AddressID = BillToAddressID
	GROUP BY City
	HAVING COUNT(*)>500
)
SELECT * FROM CityOrders;
```

| City | Number of completed orders |
|---|---|
| Paris | 575 |
| London | 756 |

There are 2 results.

**Task 2**

Table: **?**

Prepare a report which presents a list of customers that have at least four orders in the following months: 5, 6, 7, 8, 9

The structures of reports:

| CustId | Year | Month | NoOfOrders |
|---|---|---|---|
| 11176 | 2013 | 7 | 6 |
| 11330 | 2014 | 5 | 4 |
| 11331 | 2013 | 7 | 6 |
| 11212 | 2013 | 7 | 4 |
| 11276 | 2014 | 5 | 4 |

Rec: 5/?

**Solution**

Query for this task:

```
WITH BasicInfo AS
(
	SELECT OH.CustomerID AS "CustId", YEAR(OH.OrderDate) AS "Year", MONTH(OH.OrderDate)
AS "Month", COUNT(*) AS "NoOfOrders"
```

```
        FROM Sales.SalesOrderHeader OH
        GROUP BY CustomerID, YEAR(OH.OrderDate), MONTH(OH.OrderDate)
        HAVING COUNT(*) >= 4
)
SELECT *
FROM BasicInfo
WHERE BasicInfo.Month IN(5,6,7,8,9);
```

And the records returned:

| CustId | Year | Month | NoOfOrders |
|---|---|---|---|
| 11176 | 2013 | 7 | 6 |
| 11330 | 2014 | 5 | 4 |
| 11331 | 2013 | 7 | 6 |
| 11212 | 2013 | 7 | 4 |
| 11276 | 2014 | 5 | 4 |
| 11142 | 2014 | 5 | 4 |
| 11091 | 2013 | 9 | 4 |
| 11176 | 2014 | 6 | 4 |
| 11519 | 2013 | 9 | 4 |
| 11276 | 2014 | 6 | 4 |

There are 10 results.

## Task 3

Calculate the monthly number of orders based on order date for each customer separately from 2012 to 2014

The structures of reports:

- **custID, Name (Last Name, First name), Year, Month, Number of orders**

### Solution

```
WITH MonthlyOrders AS (
SELECT sc.CustomerID AS "custId",
       p.FirstName + ',' + p.LastName AS "Last Name, First Name",
       YEAR(sh.OrderDate) AS "Year",
       MONTH(sh.OrderDate) AS "Month",
       COUNT(sh.salesOrderID) AS "NoOfOrders"
FROM Sales.Customer sc
INNER JOIN Sales.SalesOrderHeader sh
ON sc.CustomerID = sh.CustomerID
INNER JOIN Person.Person p ON p.BusinessEntityID = sc.PersonID
GROUP BY sc.CustomerID,
         p.FirstName,
         p.LastName,
         YEAR(sh.OrderDate),
         MONTH(sh.OrderDate)
HAVING YEAR(sh.OrderDate) BETWEEN 2012 AND 2014
)
```

```
SELECT *
FROM MonthlyOrders
ORDER BY MonthlyOrders.custId
```

Partial result is:

| custId | Last Name, First Name | Year | Month | NoOfOrders |
|---|---|---|---|---|
| 11000 | Jon,Yang | 2013 | 6 | 1 |
| 11000 | Jon,Yang | 2013 | 10 | 1 |
| 11001 | Eugene,Huang | 2013 | 6 | 1 |
| 11001 | Eugene,Huang | 2014 | 5 | 1 |
| 11002 | Ruben,Torres | 2013 | 6 | 1 |
| 11002 | Ruben,Torres | 2013 | 7 | 1 |
| 11003 | Christy,Zhu | 2013 | 6 | 1 |
| 11003 | Christy,Zhu | 2013 | 10 | 1 |
| 11004 | Elizabeth,Johnson | 2013 | 6 | 1 |
| 11004 | Elizabeth,Johnson | 2013 | 10 | 1 |
| 11005 | Julio,Ruiz | 2013 | 6 | 1 |
| 11005 | Julio,Ruiz | 2013 | 10 | 1 |
| 11006 | Janet,Alvarez | 2013 | 5 | 1 |
| 11006 | Janet,Alvarez | 2013 | 10 | 1 |
| 11007 | Marco,Mehta | 2013 | 6 | 1 |

There are 29129 rows.

**Task 4**

Table: ?

- Create a list of customers and for each specifies a value Y or N depending on whether the number of his orders is ten times greater than the average number of customer orders

| CustID | Valuable | NumberOfOrders |
|---|---|---|
| 11019 | Y | 17 |
| 11078 | Y | 17 |
| 11091 | Y | 28 |
| 11142 | Y | 17 |
| … | … | … |

### Solution

```
WITH Orders AS(
	SELECT CustomerID AS "CustID", COUNT(*) AS "NumberOfOrders"
	FROM Sales.SalesOrderHeader
	GROUP BY CustomerID
),
AverageOfOrders AS(
	SELECT CAST(AVG(NumberOfOrders) AS FLOAT) AS "Average"
	FROM Orders
)
SELECT CustID,
CASE
	WHEN NumberOfOrders > 10 * Average THEN 'Y'
	ELSE 'N'
END "Valuable",
NumberOfOrders
FROM Orders, AverageOfOrders
```

| CustID | Valuable | NumberOfOrders |
|---|---|---|
| 11000 | N | 3 |
| 11001 | N | 3 |
| 11002 | N | 3 |
| 11003 | N | 3 |
| 11004 | N | 3 |
| 11005 | N | 3 |
| 11006 | N | 3 |
| 11007 | N | 3 |
| 11008 | N | 3 |
| 11009 | N | 3 |
| 11010 | N | 3 |

There are 19119 results.

### Task 5

Create report according to the following definition:
Table: ?

| EmpID | Order year | CustID | Last name, first name | Total sum |
|---|---|---|---|---|
| 279 | 2011 | 29624 | Cantoni, Joseph | 279655.51 |
| … | | | … | |
| 282 | 2011 | 29722 | D'sa, Reuben | 183329.02 |
| … | | | … | |
| 276 | 2011 | 29710 | Diaz, Brenda | 170660.07 |
| … | | | … | |

**Solution**

```
WITH CustomerInfo AS(
       SELECT sc.CustomerID AS "custID",
                 p.FirstName + ', ' + p.LastName AS "Last name, first name"
       FROM Person.Person p
       INNER JOIN Sales.Customer sc ON p.BusinessEntityID = sc.PersonID
)
SELECT sh.SalesPersonID AS "EmpID",
          YEAR(sh.OrderDate) AS "Order year",
          sh.CustomerID AS "CustID",
          c.[Last name, first name],
          ROUND(SUM(sh.TotalDue), 2) AS "Total sum"
FROM Sales.SalesOrderHeader sh
INNER JOIN CustomerInfo c ON c.custID = sh.CustomerID
WHERE sh.SalesPersonID IS NOT NULL
GROUP BY sh.SalesPersonID,
               YEAR(sh.OrderDate),
               sh.CustomerID,
               c.[Last name, first name]
```

The partial result is:

| EmpID | Order year | CustID | Last name, first name | Total sum |
|---|---|---|---|---|
| 274 | 2011 | 29493 | Ronald, Adina | 2417.48 |
| 274 | 2011 | 29764 | John, Ford | 4723.11 |
| 274 | 2011 | 29803 | Hattie, Haemon | 2297.03 |
| 274 | 2011 | 29818 | Roger, Harui | 23130.30 |
| 274 | 2012 | 29491 | Carla, Adams | 37625.43 |
| 274 | 2012 | 29576 | Eli, Bowen | 60.14 |
| 274 | 2012 | 29604 | Edward, Buensalido | 729.64 |
| 274 | 2012 | 29605 | Megan, Burke | 33206.02 |
| 274 | 2012 | 29616 | Barbara, Calone | 83549.58 |
| 274 | 2012 | 29669 | Jeanette, Cole | 4460.07 |
| 274 | 2012 | 29675 | Aaron, Con | 4792.25 |
| 274 | 2012 | 29680 | Dorothy, Contreras | 3048.00 |
| 274 | 2012 | 29691 | Jose, Curry | 236.35 |

| | | | | |
|---|---|---|---|---|
| 274 | 2012 | 29707 | Helen, Dennis | 90167.33 |
| 274 | 2012 | 29716 | Blaine, Dockter | 68918.24 |
| 274 | 2012 | 29722 | Reuben, D'sa | 51204.06 |
| 274 | 2012 | 29755 | Kelly, Focht | 13744.23 |
| 274 | 2012 | 29771 | Dominic, Gash | 3302.52 |
| 274 | 2012 | 29792 | Abigail, Gonzalez | 26027.71 |

There are 1802 records.

### Task 6

Count for each customer the difference of his/her orders between June and July 2014
The structures of reports:

- **custID, Name (Last Name, First name), Gender, Difference of orders June/July 2014**

#### Solution

```
WITH OrdersJune AS(
	SELECT COUNT(*) AS "OrdersAmountJune", CustomerID
	FROM Sales.SalesOrderHeader sh
	WHERE MONTH(sh.ShipDate) = 6 AND YEAR(sh.ShipDate) = 2014
	GROUP BY CustomerID
),
OrdersJuly AS(
	SELECT COUNT(*) AS "OrdersAmountJuly", CustomerID
	FROM Sales.SalesOrderHeader sh
	WHERE MONTH(sh.ShipDate) = 7 AND YEAR(sh.ShipDate) = 2014
	GROUP BY CustomerID
),
DifferenceJuneJuly AS(
	SELECT june.CustomerID, OrdersAmountJuly - OrdersAmountJune AS "Difference of
orders June/July 2014"
	FROM OrdersJuly july
	INNER JOIN OrdersJune june
	ON june.CustomerID = july.CustomerID
)
SELECT diff.CustomerID AS "custID", LastName + ', ' + FirstName AS "Last name, First
name", vp.Gender, [Difference of orders June/July 2014]
FROM DifferenceJuneJuly diff
INNER JOIN Sales.Customer sc ON sc.CustomerID = diff.CustomerID
INNER JOIN Person.Person p ON p.BusinessEntityID = sc.PersonID
INNER JOIN Sales.vPersonDemographics vp ON vp.BusinessEntityID = p.BusinessEntityID
```

Result is:

| custID | Last name, First name | Gender | Difference of orders June/July 2014 |
|---|---|---|---|
| 15758 | Andersen, Arturo | M | 0 |
| 11501 | Chandra, Brandy | F | -2 |
| 11276 | Chapman, Nancy | F | -3 |
| 11711 | Davis, Daniel | M | -2 |
| 16443 | Gao, Dawn | F | 0 |

| | | | |
|---|---|---|---|
| 11287 | Garcia, Henry | M | -1 |
| 16370 | Gonzalez, Molly | F | 0 |
| 11185 | Henderson, Ashley | F | 0 |
| 11642 | Hughes, Morgan | F | 0 |
| 18535 | Kapoor, Donald | M | 0 |
| 15868 | Lal, Caleb | M | 0 |
| 11078 | Martin, Gina | F | 0 |
| 16035 | Morgan, Sebastian | M | 0 |
| 11786 | Murphy, Ian | M | 0 |
| 11927 | Murphy, Nicole | F | 0 |
| 11502 | Peterson, Jared | M | 1 |
| 11176 | Roberts, Mason | M | -2 |
| 16721 | Sanders, Xavier | M | 0 |
| 11262 | Simmons, Jennifer | F | -1 |
| 18749 | Stewart, Olivia | F | 0 |
| 12088 | Thompson, Lauren | F | 0 |
| 11823 | Turner, Morgan | F | 0 |
| 12606 | Van, Arthur | M | 0 |

There are 23 records.

**Please do not forget about the conclusions being a summary of considered problems and proposed solutions!**

**Conclusions:**

CTE expressions are very useful as we can use them in order to create more complex queries. In addition, they are more efficient than traditional subqueries and help us to save lines of code. I used joining here in order to obtain appropriate data. ROUND() function rounds the obtained result with to a given decimal place. We should also be aware of the fact that using WHERE with aggregation functions will not work, we have to use HAVING in such case – like in task 2. I used CAST() function to calculate an accurate average using float numbers.