CA2: Introduction to databases

National College of Ireland

Michal Borkowski

Student no 17164532

2019

National College of Ireland

Table of Contents

JULY 6

National College of Ireland
Authored by: Michal Borkowski

Project Background and Description

Using the Database that was created in CA1, in this work are presented scripts to extract data and create report of the company revenue and business trend.

1 . Show all transactions for a given week in your business.

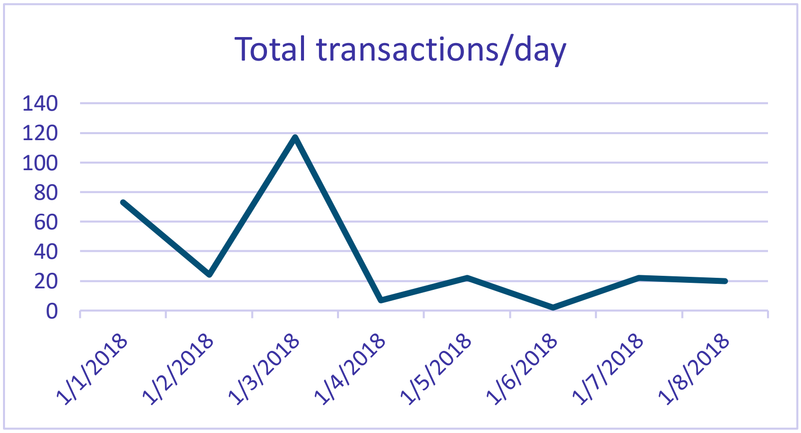
The following script creates a table in Data Mart to store all the transaction completed in the first week of 2018, showing information: dates of transactions, day in the week and transaction sum:

SQL code:

```
use twoCA;

select
            OrderDate as Date_,
            sum(Quantity) as "Total transactions/given week",
  DAYNAME(Orders.OrderDate) as Day_in_week
 from Orders
 where OrderDate
  between date '2018-01-01' and '2018-01-08'
  group by OrderDate;
```

Chart 1: shows all transactions in given week (week 1-8.03. in peak month of transactions).



2. Create a trigger that stores stock levels once a sale takes place.

The following script creates a trigger that update the table “Previous_stock” every time the level of a stock is modified:

SQL code:

```
use twoCA;

Drop table if exists previous_stock;

create table Previous_stock (
id int auto_increment primary key,
ProductNo varchar(10) not null default 'P',
LastLevel int not null,
ChangeDate datetime default null,
Actions varchar(50) default null
);

Drop Trigger if exists level_stock;

delimiter $$
create trigger level_stock
before update on Stock #trigger is set on stock table
for each row
      begin
      insert into previous_stock #saved on previous_stock table
      set actions = 'update',
         ProductNo = OLD.ProductNo,
         LastLevel = OLD.CurrentStock,
         ChangeDate = now();
      end$$
delimiter;

drop trigger level_stock;

update Stock set CurrentStock='999' where ProductNo='P11036';
```

3. Create a view of stock (by supplier) purchased by you.

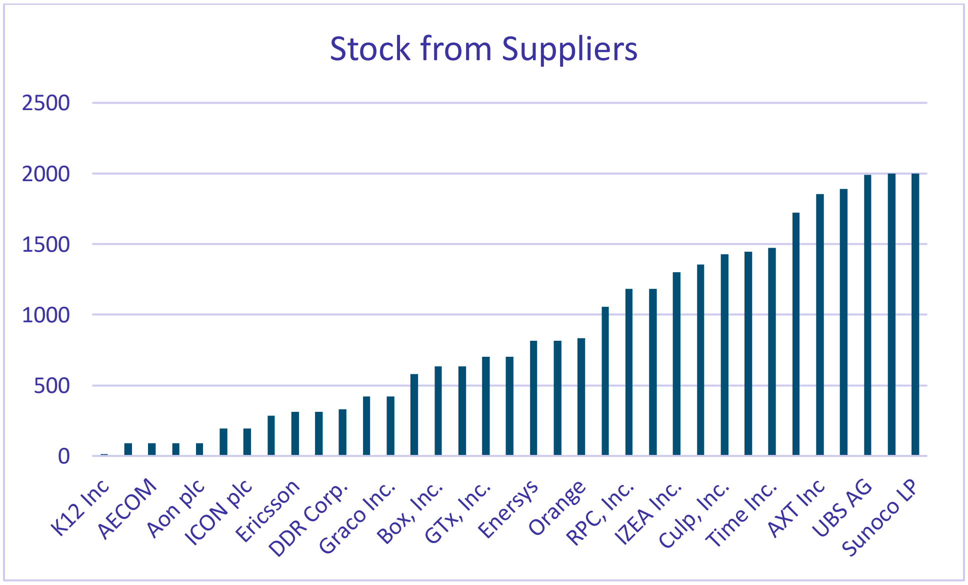
The following script creates a view of stock purchased in Operational Data by company from particular suppliers showing detailed information about suppliers and supplier`s products which were sold to company:

SQL code:

```
use twoCA;

create view my_stock_bySupplier
as
select
                Supply.Supplier,
   Supply.TimeZone,
                stock.ProductNo,
   stock.CurrentStock as my_Purchase;
from Stock
inner join Supply on Stock.ProductNo=Supply.ProductNo
order by CurrentStock asc;
```

Chart 2: below chart shows the total items bought from particular suppliers by company:



4. Total stock sold to general public (by supplier) (A group by with roll-up).

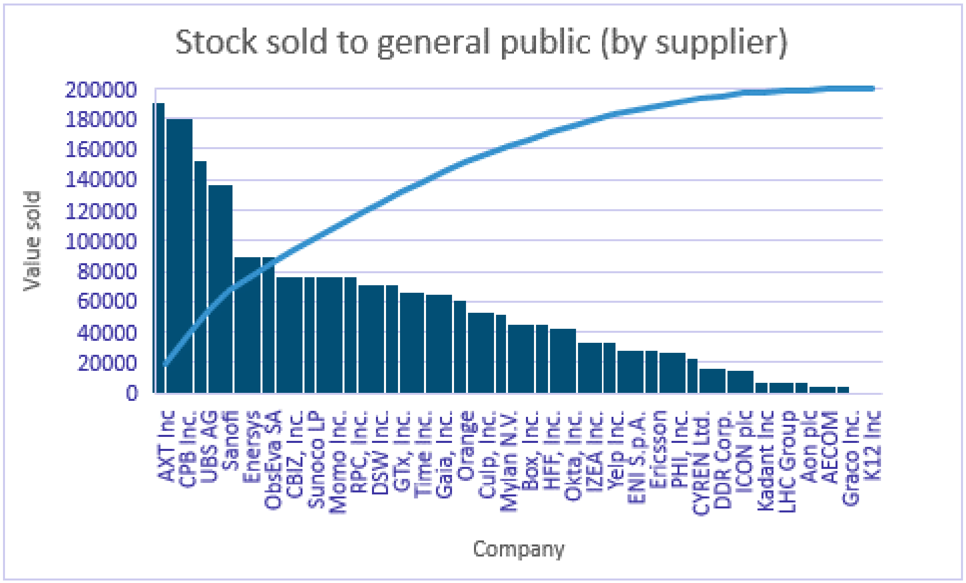
The following script generates a total stock sold by supplier:

SQL code:

```
use twoCA;

SELECT
    Supply.Supplier,
    SUM(Price * CurrentStock) AS "Total revenue"
  FROM Stock
  inner join Supply on Stock.ProductNo=Supply.ProductNo
  group by Supplier with rollup;
```

Chart 3: The following graphs demonstrate distribution of stock (by supplier) in descending order with a secondary axis as percentage of the total (Pareto style). Values are presented in units.



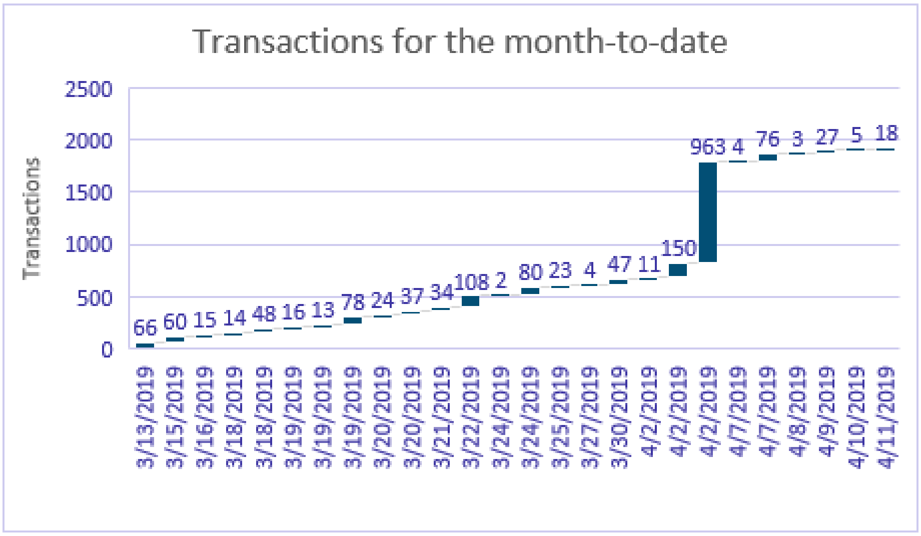
5. Detail and total transactions for the month-to-date. (A Group By with Roll-up).

The following script stroes all the transaction for the month-to-date grouped by rollup sql function:

SQL code:

```
use twoCA;
select
            OrderDate as Date_,
            sum(Quantity) as Total_MTD_Transactions,
             DAYNAME(ORDERS.OrderDate) as Day_in_month
            from Orders
            where OrderDate between '2019-03-11' and now()
group by Quantity with rollup;
```

Chart 4: show all the transaction completed (total: 1926) month-to-date every day of transaction:



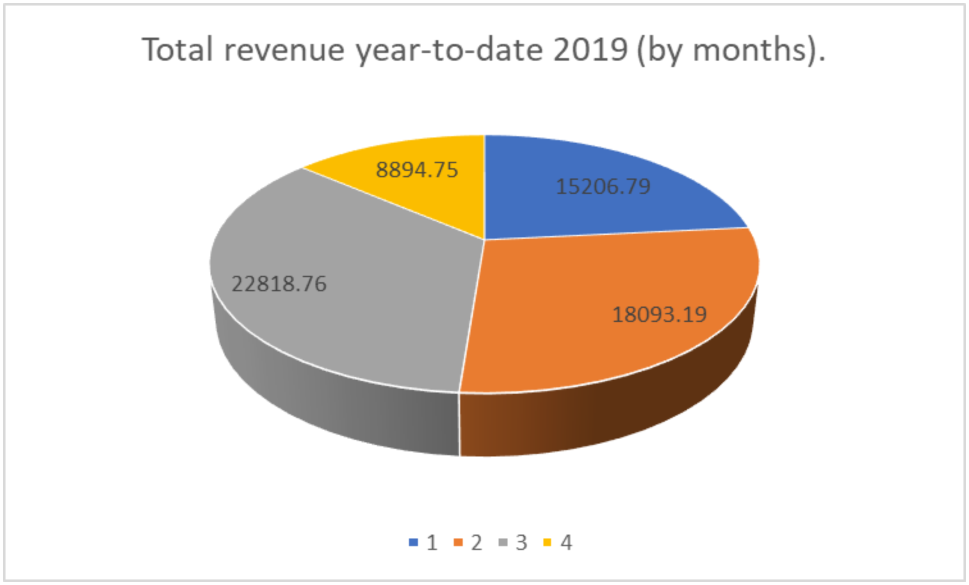
6. Detail and total revenue for the year-to-date. (A group By with Roll-up).

The following script stores the total revenue for the year-to-date grouped with sql rollup function:

SQL code:

```
use twoCA;
select
        sum(TotalPrice) as income_statement,
        monthname(ORDERS.OrderDate) as Month_in_2019
        from Orders
        where OrderDate between '2019-01-01' and now()
group by Month(Orders.OrderDate) with rollup;
```

Chart 5: below shows distribution of revenue divided by months (1-4 in 2019), classified in pie-chart type, in the last 4 months:



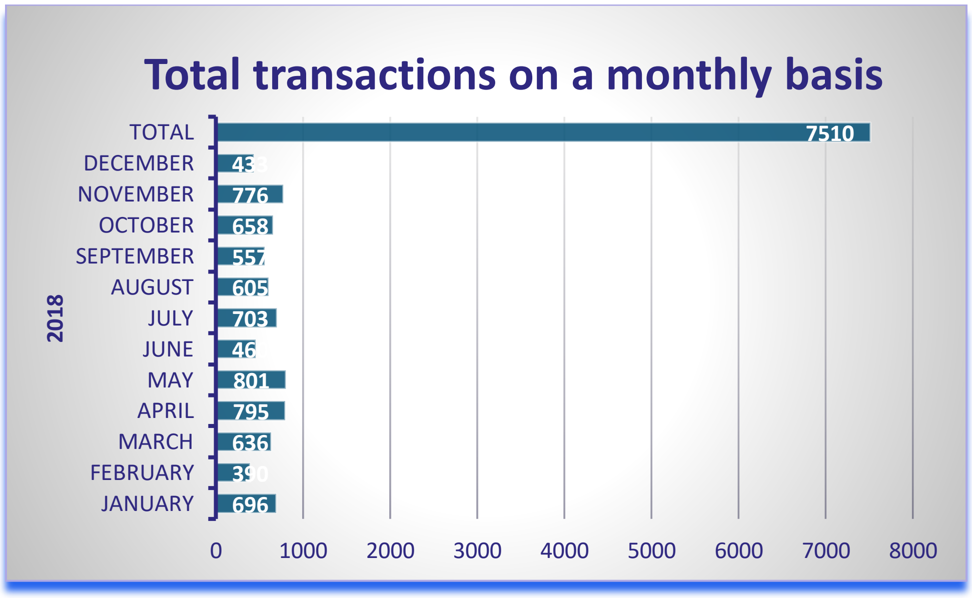
7. Detail and total transactions broken down on a monthly basis for 1 year. (A group By with Roll-up).

The following script creates stores all the transaction completed in 2018, broken down on monthly basis and final summary.

SQL code:

```
use twoCA;
select MONTHNAME(ORDERS.OrderDate) as OrderDate,
            sum(ORDERS.Quantity) as SumQuantity,
            month(ORDERS.OrderDate) as MonthNum
            from ORDERS
            where OrderDate between '2018-01-01' and '2018-12-31'
group by
            month(ORDERS.OrderDate) with rollup;
```

Chart 6: shows the 2018 transactions divided by month, in monthly order with a cumulative top line.



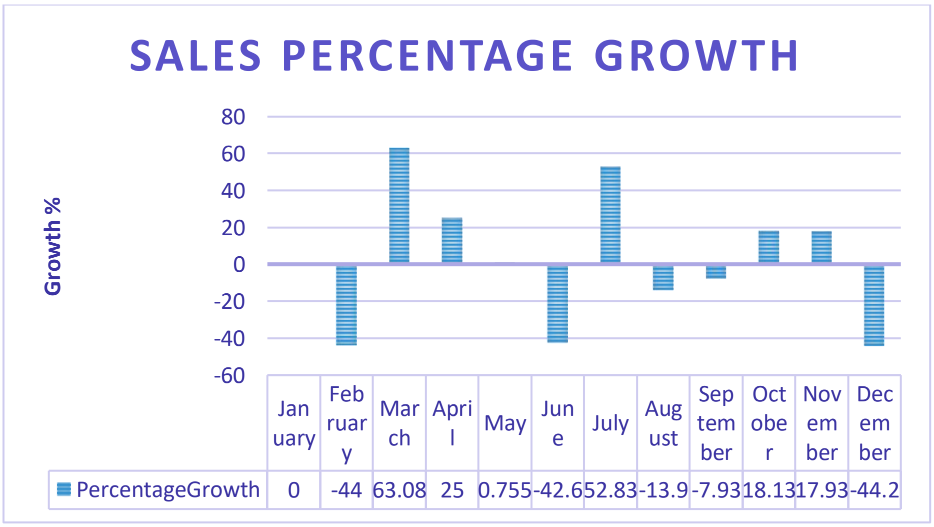
8. Display the growth in sales/services (as a percentage) for your business, from the 1st month of opening until now.

The following script provides information regarding the percentage growth in sales/services of company as income/lost for the business from the 1st transaction to the last one:

SQL code:

```
use twoCA;

set @start_growth := 0;
select OrderDate, TotalService, PercentageGrowth
from (
            select OrderDate, TotalService,
   case when @start_growth = 0
            then null
   else (TotalService - @start_growth) * 100.00 / @start_growth
            end as PercentageGrowth,
            @start_growth := TotalService,
            MonthNum
from (
   select MONTHNAME(Orders.OrderDate) as OrderDate,
   sum(Orders.Quantity) as TotalService,
            month(ORDERS.OrderDate) as MonthNum
   from Orders
            where OrderDate between '2018-01-01' and '2018-12-31'
            group by MONTHNAME(Orders.OrderDate),
            month(Orders.OrderDate)
            ) as T
            order by MonthNum) as SQ
order by MonthNum;
```



	January	February	March	April	May	June	July	August	September	October	November	December
▤ PercentageGrowth	0	-44	63.08	25	0.755	-42.6	52.83	-13.9	-7.93	18.13	17.93	-44.2