

**Data Modeling**

**2018/2019**

**Adventure Works Cycles  
OLTP to OLAP model**https://docs.google.com/drawings/d/sc2cNfYB47Oppg6M5fTXd8w/image?w=557&h=1&rev=1&ac=1&parent=1SWU-hxAxzsNmfPMsh6xgnfdnNGvATs9nRW_aLsqCa2Q

Project report

**Group members: Group ID: 12**

54842, Ondřej Zemánek

54888, Hubert Kamil Olkiewicz

50932, João Gonçalo Vaz Oliveira

December 15, 2018

# Business model Adventure Works Cycles is a retail company that sells variety of bikes, accessories and clothes at online store as well as it has plenty of retail shops. Sales is divided into distinct territories where sellers, employed at store, are responsible for selling and sending bikes to customers on contrary to online store where product selling process is automatic, without any employee supervising. Sales territories are not bound to provinces hence there can be few on one. Company is prepared to sell products in different countries or states where tax rates and currencies differs. Special offers might regard only one product, from start to end date and described minimum and maximum quantity of discounted product. Each product has its own documentation including its price history, production cost history, review, transaction history, vendors, photos, and current location. Each department has its own history of employees who were working there at each shift. Each customer has to provide personal data before buying and company keeps this data. Company also uses database for human resources management such as recruiting people.

# Analytical Tasks

* Relation between month and most sold subcategory products.
* How does the season influence sales of each product\_category.
* Most and worst profitable, for Online and Retail sales, day of the week for each territory.
* What is most profitable reason to place order in online shop for each product category?
* Which salesman sold less then his sales quota in exact month?

# Building multidimensional model

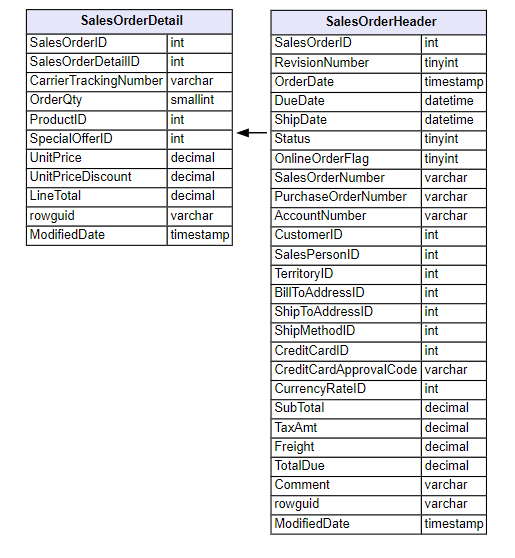
## Classification

Our relational model contains 6 transaction entities, 14 component entities and 47 classification entities. You can see result of our classification in modified relational model diagram **oltp-table-classification.png**.

### Transaction entities

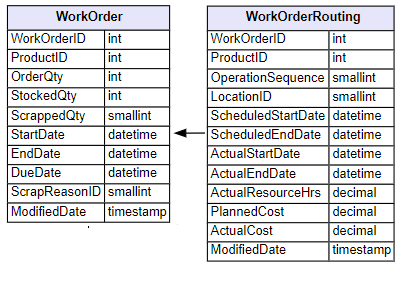
1. **SalesOrderDetail** and **SalesOrderHeader**

SalesOrder is separated into parent-child tables - detail and header. Both tables contains measurements or quantities. Header table contains informations about entire order and detail table contains attributes that refers to individual ordered products. SalesOrderHeader can be considered as component entity of SalesOrderDetail, because relationship is 1:N, but SalesOrderHeader itself is transaction entity. Considering the rules for resolving ambiguities, SalesOrderHeader is transaction entity. So both tables are considered as transaction entities.



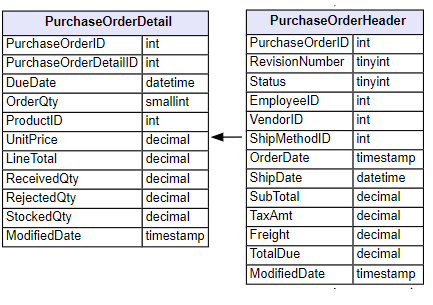
1. **WorkOrder** and **WorkOrderRouting**

WorkOrder is separated into two tables - WorkOrder and WorkOrderRouting. RelationShip between these tables is 1:1 and both tables are refers to the same specific product. Both tables contain attributes with measurements, dates and quantities, so they are transaction tables. Their purpose is to collect data about repairing products that are scrapped. But for its not relevant for our business process.

****

1. **PurchaseOrderDetail** and **PurchaseOrderHeader**

PurchaseOrder is separated into parent-child tables - detail and header. Both tables contains measurements or quantities. Header table contains informations about entire order and detail table contains attributes that refers to individual ordered products. PurchaseOrderHeader can be considered as component entity of PurchaseOrderDetail, because relationship is 1:N, but PurchaseOrderHeader itself is transaction entity. Considering the rules for resolving ambiguities, PurchaseOrderHeader is transaction entity. So both tables are considered as transaction entities.

****

### Component entities

**SpecialOffer**Specialoffer table is component entity for SalesOrderDetail transaction entity. It has relationship with SalesOrderDetail through SpecialOfferProduct table, which is not important for our analytical model.

**Address**Address table has two one-to-many relationships with SalesOrderHeader, defining BillToAddress and ShipToAddress.

**CurrencyRate**CurrencyRate is component entity for SalesOrderHeadertransaction entity, defining currencyRate for each salesOrder.

**Customer**Customer is important component for SalesOrderHeader. It contains mandatory information about customer.

**SalesPerson**SalesPerson is component for SalesOrderHeader, but It used only for orderes from retail sales (defined in SalesOrderHeader by OnlineOrderFlag attribute).

**SalesTerritory**SalesTerritory defines all world regions in which are company selling their products. It is component for SalesOrderHeader.

**SalesReason**SalesReason defines all reasons for sale. Relationship with SalesOrderHeader is many-to-many.

### Classification entities

The classification entities are connected to component entities by one-to-many relationships. Because there are a lot of classification entities, we decided to not include them inside report.

## Collapsing hierarchies

**SalesReason**The SalesReason component entity has no classification entity dependencies - nothing to collapse.

**SpecialOffer**SpecialOffer has no classification entity dependencies - nothing to collapse.

**Customer**Collapsed Customer component entity including attributes from Person, Store and BusinessEntity tables.

**Address**Into Address component entity were collapsed StateProvince and CountryRegion. SalesTerritory was not concluded because of two reasons. Its component entity and as well its data is irrelevant for our analysis is address context.

**CurrencyRate**Into CurrencyRate component entity was collapsed only Currency classification entity.

**SalesTerritory**Collapsed SalesTerritory component entity including additional attributes from CountryRegion.

**ShipMethod**This entity has no classification dependent entities. Nothing to collapse.

**Product**Into Product component entity were collapsed UnitMeasure, ProductModel (both entities without any further dependencies) and ProductSubcategory which was dependent on ProductCategory classification entity.  
  
**Vendor**It’s one-to-one with BusinessEntity which is holding only primary key.

**SalesPerson**Collapsed SalesPerson contains additional attributes from Employee, Person and BusinessEntity.

**Employee**Collapsed Employee contains additional attributes from Person and BusinessEntity.

## Dimensional Design process

### Multidimensional model

Based on our analytical tasks defined before, we decided to use star schema for our multidimensional model. We have to define our fact tables and dimension tables. Because we are transforming OTLP model to OLAP model, we can apply rules to transform classified tables from source relational model to fact and dimension tables.

### Dimensions tables

From every collapsed entity we cutted off *rowguid* and *ModifiedDate* attributes, as well as all *ID\_PK* of collapsed entities hence they yield no analytical data.

**DimSalesReason**SalesReason dimension contains every attribute from component table SalesReason. It’s important dimension for our analytical tasks, because we can measure benefit of each reason for our business.

**DimSpecialOffer**SpecialOffer dimension bears information about promotions for products. For example we give discount to user volume discounts or discount for seasonal products.

**DimCustomer**Custumer dimension contains attributes necessary for analytical tasks, because it helps to recognize the customer profile. From this dimension were deleted three columns - *AdditionalContactInfo*, *PersonDemographics* and *StoreDemographics*, because they are contain xml structure with detailed informations so they are useless for our business process.

**DimEmployee**Employee dimension is generalized from Employee and SalesPerson tables. These two tables are merged together, because collapsed SalesPerson component entity contains attributes from Employee table plus few additional attributes which can be added to Employee table with new *SalesPersonFlag* attribute, which indicates if Employee is SalesPerson or not.

**DimVendor**Vendor dimension bears information about our vendors/supplier. It can supports decision in inventory supply strategy. *PurchasingWebServiceURL*, *AccountNumber* attributes deleted, not subject of our analysis.

**DimShipMethod**ShipMethod dimension contains information relating to logistic companies and their prices and rates.

**DimSalesTerritory**SalesTerritory is just dimension containing information about Year-To-Date sales and costs of distinct territory.

**DimProduct**Product dimension contains all informations regarding product, its description, characteristics and costs.  
*CatalogDescription* and *Instructions* attributes (regarding *.xml* files) were deleted because they are not analysis subject.

**DimCurrency**Currency dimension yields information regarding sale currencies and their rates.

**DimAddress**Address dimension describes sales billing and shipping address.

**DimDate**After analyzing attributes for time and date in available dataset, we decided to create only Date dimension, because granularity of time in source database is day. It was created based on fact tables attributes not component tables. This new dimension contains attributes for *date*, *month*, *year dayOfWeek* and *quarter*. It’s one of the most important dimension. Dimension is populated with every day from ‘2011-01-01’ until now, because its first date used in source database.

### Fact tables

Fact tables are based on transaction tables from source OLTP model. Regarding our analytical requirements, we used from source model only **SalesOrder** (header and detail tables) and **PurchaseOrder** (header and detail tables) transaction entities. Also because these transaction tables are divided into parent-child tables, we decide to merge these tables into single fact tables for lower the number of join operations and improve performance for simple queries.

**FactOnlineSales**

FactOnlineSales fact table is used for storing sale transactions from **online shop**. Each transaction is single line of online sale order, that contains measurements for product cost, product selling price, product discount percentage, order line price, order line tax amount, freight price and also ordered quantity of ordered product. Each row has also aggregation attributes based on measurements for order line total price, total discount amount and gross margin in percentage and also total price of order including shipping and taxes. Granularity is related to ShipMethod, Product, Date, Territory, Customer, SpecialOffer, Address and Currency dimension tables. The fact table is based on SalesOrderDetail and SalesOrderHeader transaction tables. Source transaction entity SalesOrderHeader contains *OnlineOrderFlag* attribute, that defines if transaction is from online store or retail store. In the fact table are only transactions from online shop.

**FactRetailSales**

FactRetailSales fact table is used for storing sale transactions from **retail shop**. Each transaction is single line of online sale order, that contains measurements for product cost, product selling price, product discount percentage, order line price, order line tax amount, freight price and also ordered quantity of ordered product. Each row has also aggregation attributes based on measurements for order line total price, total discount amount and gross margin in percentage and also total price of order including shipping and taxes. Granularity is related to Employee, ShipMethod, Product, Date, Territory, Customer, SpecialOffer, Address and Currency dimension tables. The fact table is based on SalesOrderDetail and SalesOrderHeader transaction tables. Source transaction entity SalesOrderHeader contains *OnlineOrderFlag* attribute, that defines if transaction is from online store or retail store. In this fact table are only transactions from retail shops.

**FactPurchaseOrder**

FactPurchaseOrder fact table storing transactions refers to orders from vendors. One row is single line of purchase order, that contains measurements for product cost, quantity, received and rejected quantity, subtotal price, tax amount and Freight. From these measurements are also aggregated attributes for order line total price and stocked quantity (received - rejected). The fact tables is based on PurchaseOrderDetail and PurchaseOrderHeader tables.

### Factless fact tables

**FactSalesReason**

FactSalesReason is factless defining sales reasons for each sale. If sale has no sale reason, FactSalesReason has row for this sale with “No reason filled” value.

## Populating database

Database was populated using *dim\_pop\_script.sql.* It was made mostly by joining all collapsed entities and then inserting chosen fields into final dimension. Sometimes it was also required to create function to handle more custom attributes like *SalesPersonFlag.*

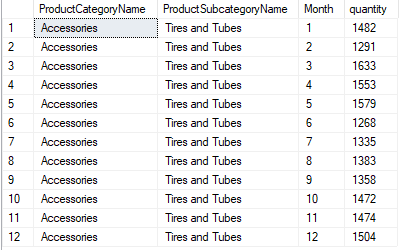
# Analytical tasks implementation and results

## Relation between month and most sold subcategory products

Query:

|  |
| --- |
| SELECT p.ProductCategoryName, p.ProductSubcategoryName, d.Month, SUM(o.OrderQuantity) AS quantity FROM [starschema].[dbo].[FactOnlineSales] o JOIN DimProduct p ON p.ProductKey = o.ProductKey JOIN DimDate d ON d.DateKey = o.OrderDateKey JOIN DimSalesTerritory st ON st.SalesTerritoryKey = o.SalesTerritoryKey GROUP BY p.ProductCategoryName, p.ProductSubcategoryName, d.Month ORDER BY p.ProductCategoryName, p.ProductSubcategoryName, quantity DESC |

Example query result:

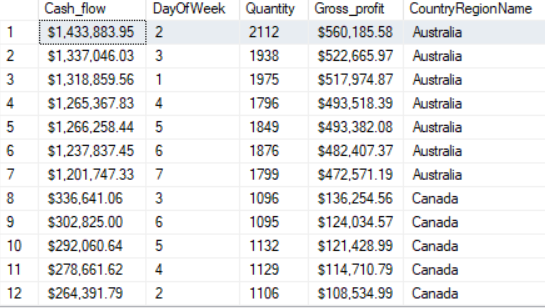


## Most and worst profitable, for Online and Retail sales, day of the week for each territory.

Query:

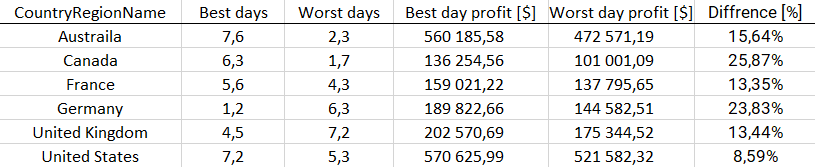
|  |
| --- |
| SELECT FORMAT(sum(o.UnitSellingPrice), 'C') as Cash\_flow, d.DayOfWeek, sum(OrderQuantity) as Quantity, FORMAT(sum(o.UnitSellingPrice \* o.LineTotalGrossMarginPct), 'C') as Gross\_profit, st.CountryRegionName  FROM [starschema].[dbo].[FactOnlineSales] o  JOIN DimProduct p ON p.ProductKey = o.ProductKey  JOIN DimDate d ON d.DateKey = o.OrderDateKey  JOIN DimSalesTerritory st ON st.SalesTerritoryKey = o.SalesTerritoryKey  GROUP BY d.DayOfWeek, st.CountryRegionName  ORDER BY st.CountryRegionName, Gross\_profit DESC |

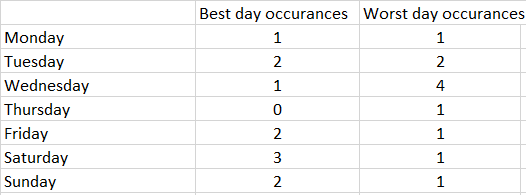
Example query result:



Conclusions:

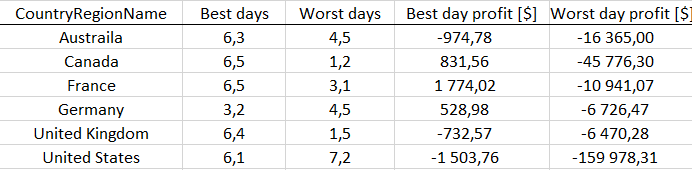
OnlineSales:

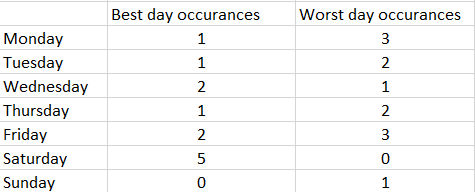
  
Best and worst profitable days, for Online sales, for each country region

  
Best and worst profitable days, for Online sales,, occurrences

It is hard to define exactly which day is worst and best in terms of profit in online sales. What we can observe is that Wednesday tends to be worst one in most countries, and days around weekend are best ones (Friday, Saturday, Sunday), but it happens that best day in one country might be worst in another. The biggest profit difference between worst and best selling day was in Canada (~25%) and Germany (~23%), and smallest in United States (~8,6%). Canada and Germany was most noticeable, but on average difference was around 17% in profit.

RetailSales:

  
Best and worst profitable days, for retail sales, for each country region

  
Best and worst profitable days, for retail sales, occurrences

The biggest notable thing is that retail shops are yielding negative profit. Yet it might be overall profitable because or retails shop marketing function (brand staying subconsciously in customer mind). In terms of best selling days there is no surprise that best one is Saturday, but Sunday is not even once noted in two best selling days. Sunday is even once worst selling day. It is because products sold at retail stores are giving negative profit hence the more products are sold, more money is lost. So this table should be read in reverted way, because of this fact, because less amount of sold products means less loss for company. So even though Saturday and Sunday is worst day for company profit, in terms of quantity sold, these are best days.

## What is most profitable reason to place order in online shop for each product category?

Query:

|  |
| --- |
| SELECT \* FROM (SELECT p.ProductCategoryName, r.Name, SUM(s.UnitSellingPrice-s.UnitCost) AS selling\_sum FROM [starschema].[dbo].FactOnlineSales s  LEFT JOIN [starschema].[dbo].[FactSalesReason] sr ON sr.SalesOrderNumber = s.SalesOrderNumber LEFT JOIN [starschema].[dbo].[DimProduct] p ON p.ProductKey = s.ProductKey LEFT JOIN [starschema].[dbo].[DimSalesReason] r ON r.SalesReasonKey = sr.SalesReasonKey WHERE r.Name is not null GROUP BY r.Name, p.ProductCategoryName) x GROUP BY x.selling\_sum, x.Name, x.ProductCategoryName ORDER BY x.selling\_sum DESC, x.Name |

Result:



Conclusions:

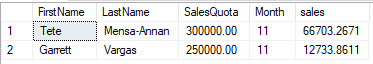
From result we can see, that the most frequent reason to buy product is price. Also we can see that customers second most common reason to buy are reviews. For accessories customers are also bid on TV addvertisment. Suprisingly customers didn’t selected quality as reason for buying accessories or clothing.

## Which salesman sold less then his sales quota in exact month?

Query:

|  |
| --- |
| SELECT e.FirstName, e.LastName, e.SalesQuota, d.Month, SUM(r.TotalPrice) AS sales FROM [starschema].[dbo].[DimEmployee] e  LEFT JOIN [starschema].[dbo].[FactRetailSales] r ON r.EmployeeKey = e.EmployeeKey LEFT JOIN [starschema].[dbo].[DimDate] d ON d.DateKey = r.OrderDateKey WHERE SalesQuota is not null AND d.Year = 2012 AND d.Month = 11 GROUP BY e.SalesQuota, e.FirstName, e.LastName, d.Month HAVING e.SalesQuota > SUM(r.TotalPrice) ORDER BY d.Month, SalesQuota DESC |

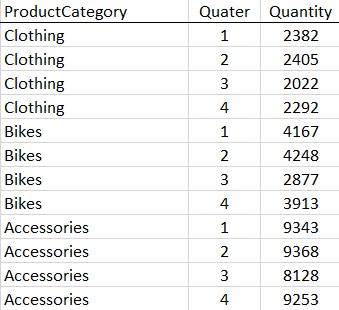
Result:

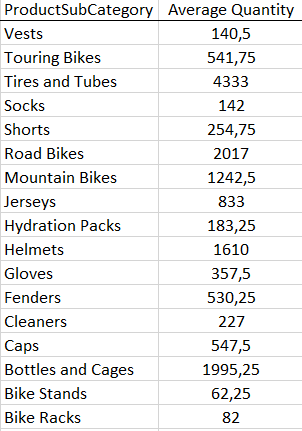


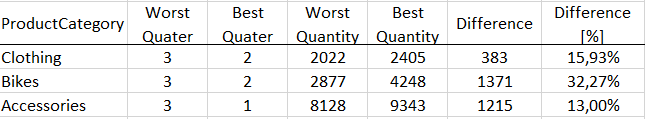
## How does the season influence sales of each product\_category

Query:

|  |
| --- |
| SELECT p.ProductSubcategoryName, d.Quarter, sum(OrderQuantity) as Quantity  FROM [starschema].[dbo].[FactOnlineSales] o  JOIN DimProduct p ON p.ProductKey = o.ProductKey  JOIN DimDate d ON d.DateKey = o.OrderDateKey  GROUP BY d.Quarter, p.ProductSubcategoryName   ORDER BY p.ProductSubcategoryName DESC  Exemplary query result:    Quantity of products sold by every quarter divided by SubcategoryName  Quantity of products sold by every quarter divided by CategoryName |

Conclusions:  
  
Quantity of products sold by every quarter divided by ProductCategory

  
Average Quantity of products sold yearly divided by ProductSubCategory

  
Best and worst quantity of products sold divided by quarter

As we can see, worst quarter for sales is Autumn one. Biggest sale decline is visible in bikes (roughly 32%). All sale declines are reasonable because of upcoming Winter season. Also best quarter for accessories is Spring one because everyone who has bike already is preparing his bike for usage. Overall is clearly visible that Winter season is much worse in sales than Spring/Summer time. But still clothing and accessories overall quantity difference is not big between worst and best quarter (~15%). Bikes are noting most decline.