ADVENTURE WORKS

Contents

[1. BACKGROUND FOR THE DEVELOPMENT OF CHOSEN SYSTEM 2](#_Toc13597238)

[2. SCOPE 2](#_Toc13597239)

[3. ERD/RELATIONAL SCHEMA – SHOWING THE SCOPE 3](#_Toc13597240)

[4. IMPLEMENTATION IN SQL SERVER – STAGES OF IMPLEMENTATION 4](#_Toc13597241)

[4.1. FACT AND DIMENSION TABLES WITH KEYS 4](#_Toc13597242)

[5. ANALYSIS & REPORTS 6](#_Toc13597243)

[5.1 . VISUALISATIONS IN R 6](#_Toc13597244)

[5.1.1 ORDER SUMMARY BY RESELLER 6](#_Toc13597245)

[5.1.2 MOST POPULAR RESELLER IN TORONTO 7](#_Toc13597246)

[5.1.3 DISCOUNT RANGE IN EACH PRODUCT LINE 8](#_Toc13597247)

[5.1.4 QUARTERLY SALES REPORT 9](#_Toc13597248)

[5.2 VISUALISATIONS IN TABLEAU 10](#_Toc13597249)

[5.2.1 Annual sales by business type 10](#_Toc13597250)

[5.2.2 Employees by business type 11](#_Toc13597251)

[5.2.3 Cities with 100 million revenue 12](#_Toc13597252)

[5.2.4 Annual sales by product type 13](#_Toc13597253)

[6. IMPLEMENTING A WAREHOUSE USING XML SCHEMA 14](#_Toc13597254)

[7. IMPLEMENTATING A WAREHOUSE USING NEO4J 15](#_Toc13597255)

[8. CONCLUSIONS 15](#_Toc13597256)

[9. BIBLIOGRAPHY 15](#_Toc13597257)

[Appendix – visualization and r code 16](#_Toc13597258)

# 1. BACKGROUND FOR THE DEVELOPMENT OF CHOSEN SYSTEM

AdventureWorks is a fictitious database containing Online Transaction Processing (OLTP) data for a bicycle manufacturing and sales company. The warehouse has mainly been designed to track reseller sales. It holds key information like territory sales information, Shipment tracking information, Sales Promotions, Discounts etc. in addition to basic product sales information such as quantity, date, product info etc.

The information stored in the warehouse can then be used to analyse various selling patterns, offer promotions and discounts etc.

# 2. SCOPE

* Primarily to store various reseller information related to the product
* To analyse geographical trends in products sold
* Revenue generated over a period of time(Quarterly/Monthly)
* Performance of Employees
* Individual product demand
* Shipment Tracking
* Keep track of discounts and promotions

# 3. ERD/RELATIONAL SCHEMA – SHOWING THE SCOPE



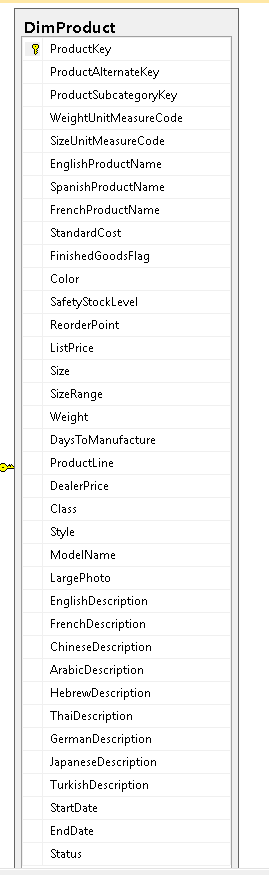
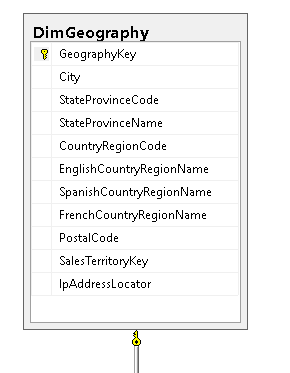
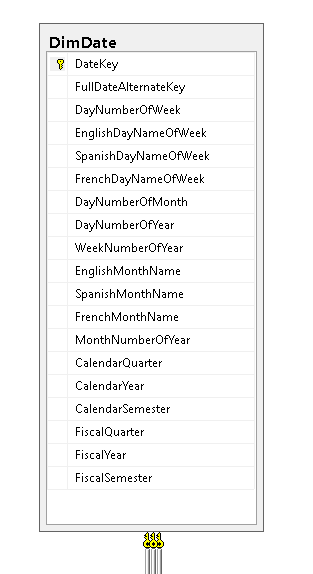
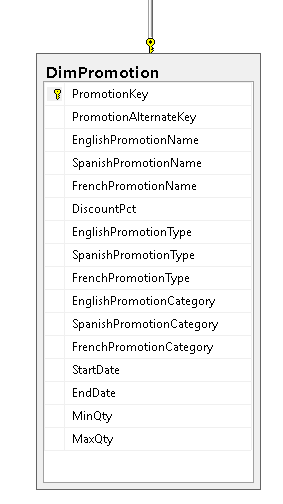
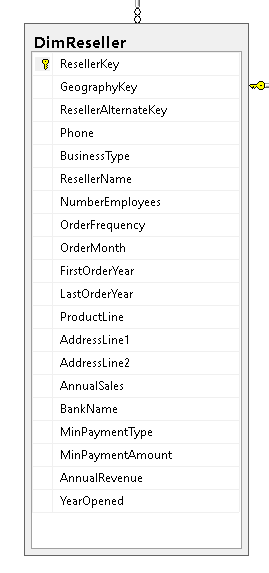
Upon observing the available input data and the objectives facilitating the need for a data warehouse it was evident that the above schema is best suited in this particular case.

# 4. IMPLEMENTATION IN SQL SERVER – STAGES OF IMPLEMENTATION

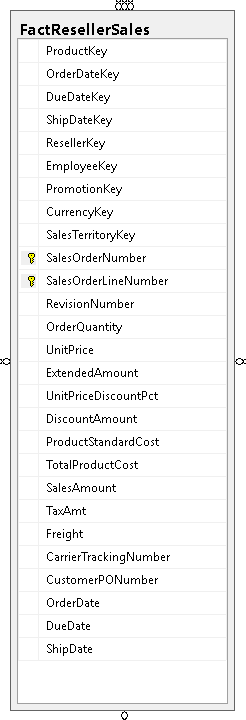
* The flat file data source was imported into the data warehouse using the Import files feature of MS SQL Server Management Studio
* The fact tables were created and populate using SQL Queries(Appendix A) in accordance with the requirements of the intended analysis.
* The Primary Key of each dimension was identified and was set using the design feature of the SQL Server Management Studio.
* The Database diagram was populated.
* The Foreign Key relationship was set using the Database Diagram in Management Studio and mapping the corresponding keys directly from the Database Diagram and assigning the relationship.

## 4.1. FACT AND DIMENSION TABLES WITH KEYS

Five Dimensions have been modelled namely DimPromotion, DimReseller, DimGeography, DimProduct, DimDate.



Fact Table:



# 5. ANALYSIS & REPORTS

## 5.1 . VISUALISATIONS IN R

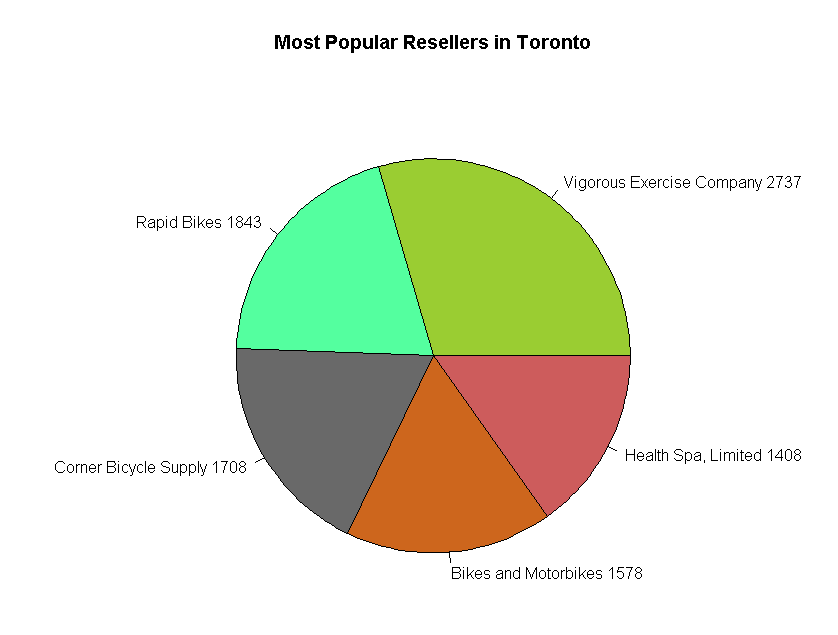
### 5.1.1 ORDER SUMMARY BY RESELLER

A screenshot of a cell phone

Description automatically generated

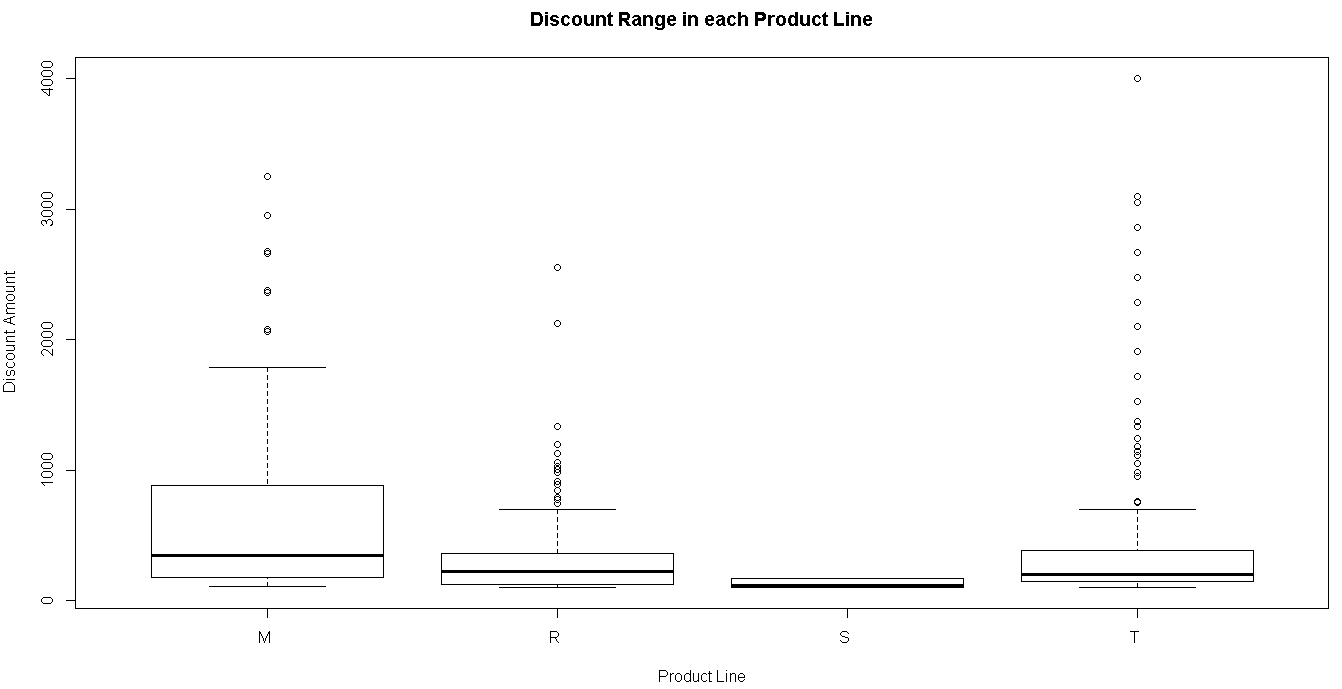
The above plot shows the Orders placed by each of the reseller in total. As can be seen from the plot Vigorous Exercise Company has the highest number of orders in place and Metropolitan Equipment has the lowest number of orders.

### 5.1.2 MOST POPULAR RESELLER IN TORONTO



The above plot shows the most popular resellers in Toronto region. Vigorous Exercise Company once again tops the list followed by Rapid Bikes, Corner Bicycle Supply, Bikes and Motorbikes and Health Spa, Limited. The popularity between the most popular and the next best shows a huge difference indicating a higher performance from their side.

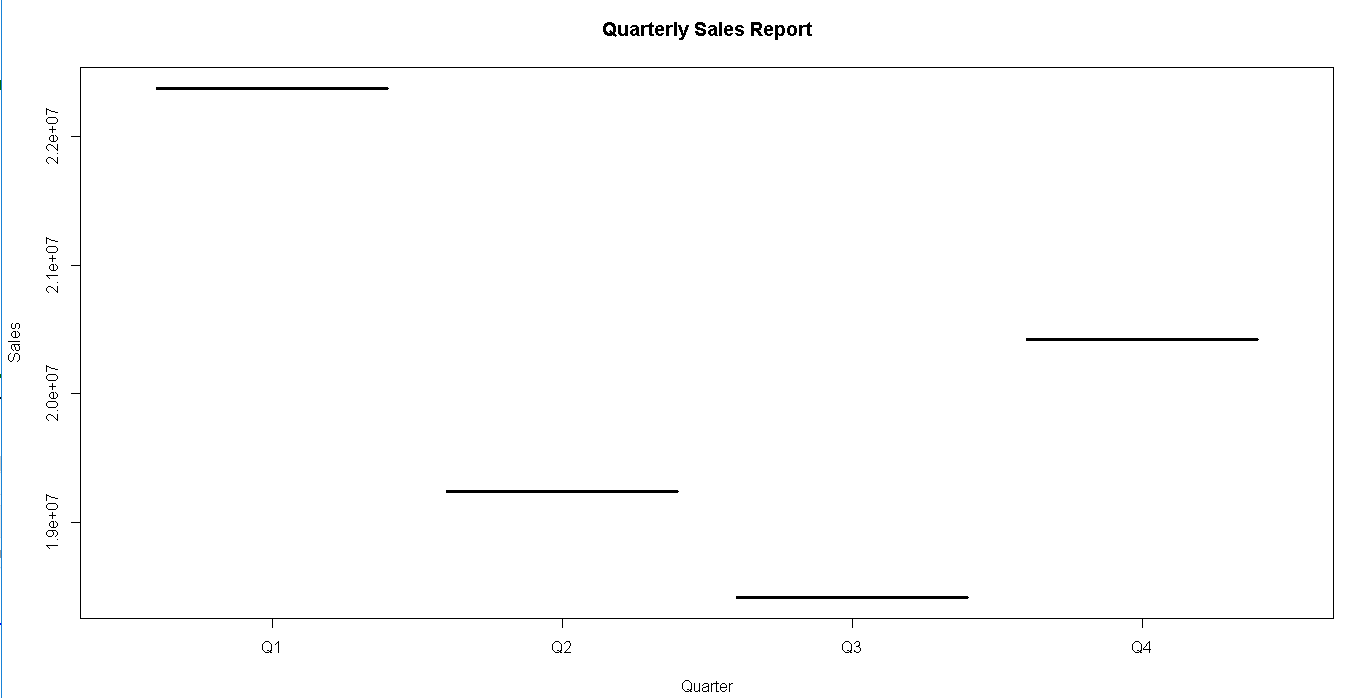
### 5.1.3 DISCOUNT RANGE IN EACH PRODUCT LINE



M – Mountain, R – Road, S – Sport, T – Touring

The least number of discounts were offered in the Sports range indicating a strong selling product. The touring range had most number of discounts outside the normal range indicating more number of products which were found difficult to sell. Even Road range had a few number of discounts outside the normal.

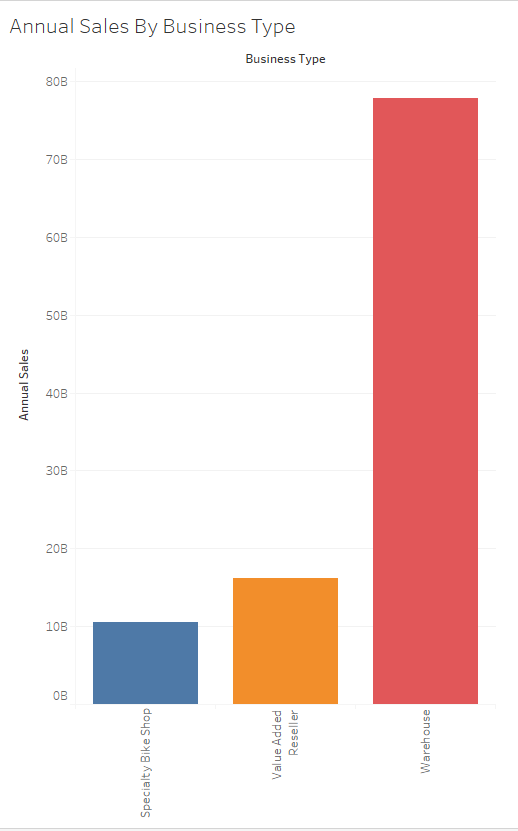
### 5.1.4 QUARTERLY SALES REPORT



The quarterly sales report divides the total sales into four quarters and as can be seen the first quarter shows the highest performance which far exceeds the lowest selling quarter (Q3) by more than a million units.

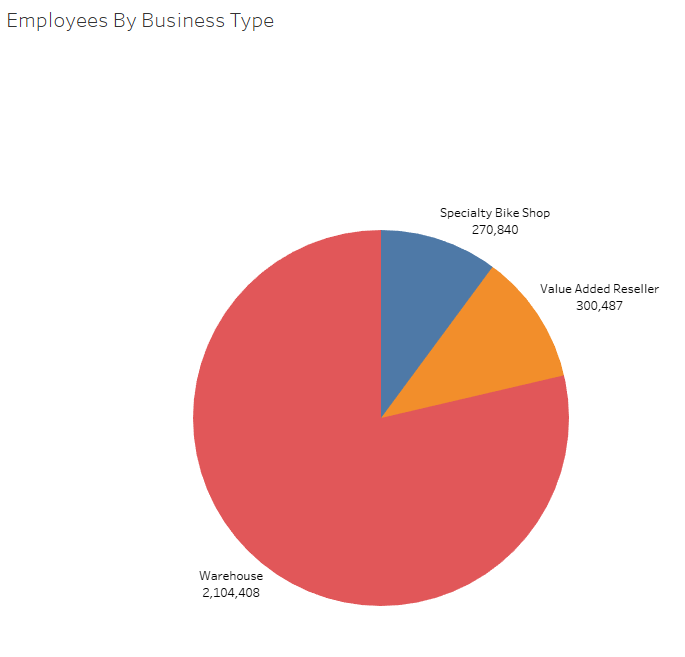
## 5.2 VISUALISATIONS IN TABLEAU

### 5.2.1 Annual sales by business type



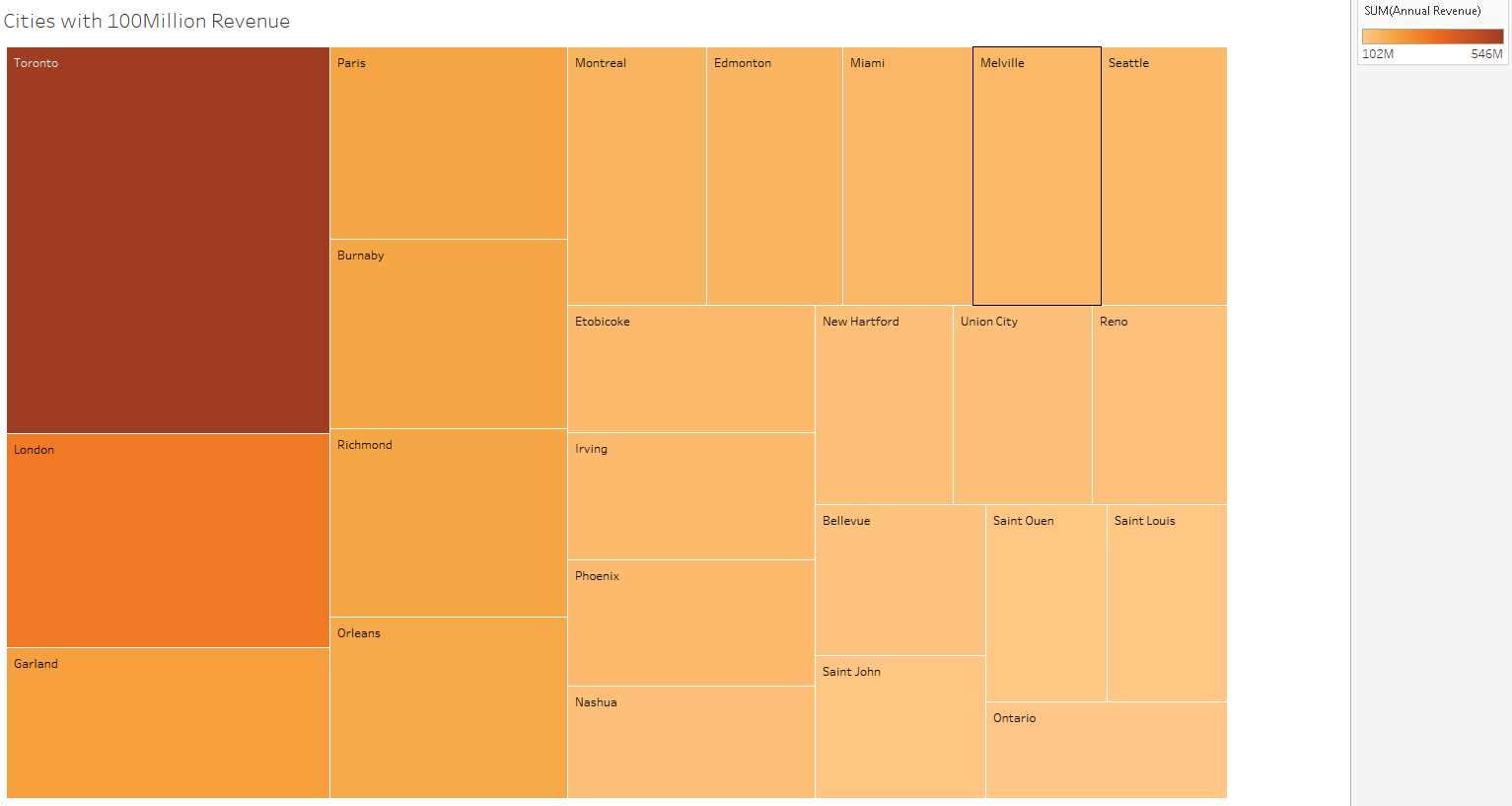
The above plot shows that the direct sales from Speciality Bike Shops and Value Added Resellers is far outnumbered by the sales directly from the warehouse. The sales from Speciality Bike Shops is atleast 65Billion units less than from the Warehouse. This can be used as an index to focus sales strategies and discounts.

### 5.2.2 Employees by business type



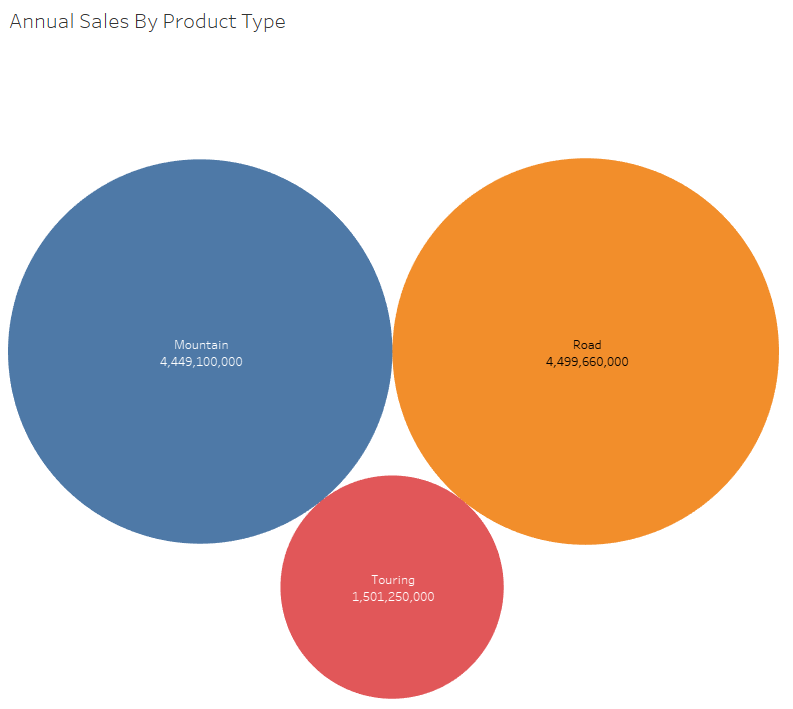
Warehouse employs the most number of people followed by Value Added Resellers and Specialty Shops . This shows a direct relationship with the annual sales of each type of entity wherein the Warehouse had to employ more due to the amount sales they have to process annually.

### 5.2.3 Cities with 100 million revenue



Toronto leads the cities with London, Garland and Paris following. This can be used to offer promotions to low performing cities and bring up the sales.

### 5.2.4 Annual sales by product type



The above bubble chart shows the annual sale of cycles by their type. This can be used to forecast and plan the production plan and selling patterns for the future. Mountain Bikes and Road going versions show more or less similar selling patterns.

# 6. IMPLEMENTING A WAREHOUSE USING XML SCHEMA

The Data Warehouse developed using SQL as mentioned above can also be implemented using XML Schema. XML Schema along with the XSD file can be used to implement a Data Warehouse. The advantage of using XML Schema is that it is widely accepted and is particularly useful when migrating your data.

A screenshot of a cell phone

Description automatically generated

# 7. IMPLEMENTATING A WAREHOUSE USING NEO4J

The primary difference is that in a graph database, the relationships are stored at the individual record level, while in a relational database, the structure is defined at a higher level (the table definitions).

This has important ramifications:

* A relational database is much faster when operating on huge numbers of records. In a graph database, each record has to be examined individually during a query in order to determine the structure of the data, while this is known ahead of time in a relational database.
* Relational databases use less storage space, because they don't have to store all of those relationships.

Storing all of the relationships at the individual-record level only makes sense if there is going to be a lot of variation in the relationships; otherwise you are just duplicating the same things over and over. This means that graph databases are well-suited to irregular, complex structures. But in the real world, most databases require regular, relatively simple structures. This is why relational databases predominate.

The Neo4j diagrams and queries are attached as document called Neo4j Document implementing the “FactReseller” Table**.**

# 8. CONCLUSIONS

From the analysis, it can be concluded that the data varies largely across the country. There are many cities which generate revenue more than 100 Million. It can also be seen that the Mountain and Road range accounts for the most number of products sold. The warehouses sell more products than speciality bike shops and value added resellers. All the above information can be used to devise sales strategies and it should be noted that these strategies should aim at improving the sales at Value Added Resellers and Speciality Shops.

# 9. BIBLIOGRAPHY

https://neo4j.com/developer/graph-db-vs-rdbms/ **B**

https://stackoverflow.com/questions/12739208/how-to-use-rank-in-sql-server**Y**

http://www.utilities-online.info/xsdvalidation/#.xduqovz7suk**.CustomerI**

# Appendix – visualization and r code

**install.packages("RODBC")**

**require(RODBC)**

**conn <- odbcDriverConnect("driver={SQL Server};server=LAPTOP-72BLL3OI\\SQLEXPRESS;database=AdventureWorksDW2014; trusted\_connection=true;")**

**conn**

**q1 <- sqlQuery(conn,"select top 5 sum(f.OrderQuantity) value, g.City, r.ResellerName**

**from FactResellerSales f**

**inner join DimProduct d**

**on f.ProductKey = d.ProductKey**

**inner join DimReseller r**

**on f.ResellerKey = r.ResellerKey**

**inner join DimGeography g**

**on r.GeographyKey = g.GeographyKey**

**group by g.City, r.ResellerName**

**order by sum(f.OrderQuantity) desc")**

**q1**

**plot(q1$ResellerName, q1$value, xlab= "Reseller", ylab = "Number of Orders",**

**main = "Reseller's Order Summary")**

**# --Most sold products in a particular city**

**q2 <- sqlQuery(conn,"select top 5 sum(f.OrderQuantity) OrderCount, g.City, r.ResellerName**

**from FactResellerSales f**

**inner join DimProduct d**

**on f.ProductKey = d.ProductKey**

**inner join DimReseller r**

**on f.ResellerKey = r.ResellerKey**

**inner join DimGeography g**

**on r.GeographyKey = g.GeographyKey**

**where g.City = 'Toronto'**

**group by g.City, r.ResellerName**

**order by sum(f.OrderQuantity) desc")**

**q2**

**pie(q2$OrderCount, labels = paste(q2$ResellerName,q2$OrderCount),col = sample(colors()),main = "Most Popular Resellers in Toronto")**

**# --Products with max discount**

**q3 <- sqlQuery(conn,"select d.EnglishProductName, f.DiscountAmount, d.ProductLine**

**from FactResellerSales f**

**inner join DimProduct d**

**on f.ProductKey = d.ProductKey**

**where f.DiscountAmount > 100**

**order by f.DiscountAmount desc")**

**View(q3)**

**boxplot(q3$DiscountAmount~q3$ProductLine, xlab = "Product Line", ylab = "Discount Amount",**

**main= "Discount Range in each Product Line")**

**#**

**# --Sales in every Quarter -- Trend**

**q4 <- sqlQuery(conn,"select concat('Q',d.CalendarQuarter) Quarter, sum(SalesAmount) Sales**

**from FactResellerSales f**

**inner join DimDate d**

**on f.OrderDateKey = d.DateKey**

**group by d.CalendarQuarter")**

**plot(q4$Quarter,q4$Sales,**

**type = "l", xlab = "Quarter", ylab = "Sales", main = "Quarterly Sales Report")**