Assignment 2: Data Warehousing

In this assignment, you will implement a data warehouse for a fictitious company and formulate queries to address their business needs.

Instructions

Deadline

Make sure to upload **all** your results **before December 30, 2016**!

How to hand in your results

Compress your solution folder and upload it on TUWEL.
Use the following naming convention:

```
VU_BI_2016_HandsOn2_DWH_<group no>.zip/.tar.gz
```

Please make sure to upload only a single solution per group. Furthermore, make sure to list your group members in the header of your solution document.

Tools

You will use a set of open source, cross-platform tools to develop a data mart and analyze sales data. Specifically, you will need to set up the following software:

- Java Runtime Environment
- MySQL (or MariaDB)
- MySQL Connector/J
- Pentaho Data Integration
- Pentaho Schema Workbench

The complete software stack is available for Linux, Windows, and MacOS. Compared to more full-fledged commercial BI solutions, the individual elements can be set up fairly quickly on any machine with minimal system requirements and good performance for basic needs. The key consideration in selecting these software products – apart from their Open Source status - was to provide an easy to install environment for basic experimentation in order focus on developing an understanding of concepts rather than learning particular graphical tools. Please refer to the Lab Setup section of this document for installation instructions.

Questions

Please post general questions in the TUWEL discussion forum of the course. You can also discuss problems and issues you are facing there. We appreciate if you help out other students tackling general problems or questions (and may take that into account in case you are short a few points for a better grade). For obvious reasons, however, please do not post any solutions there.

Assignment 2: Data Warehousing

Please direct other questions (particularly those that are not of general interest for other students) to this semester's BI tutor, Gerta Sheganaku:

`gerta.sheganaku@tuwien.ac.at`

You can also contact me directly at:

`elmar.kiesling@tuwien.ac.at`

Please use subject line `BI_2016_<your subject>` to minimize the likelihood of your Email getting lost in my inbox.

Assignment 2: Data Warehousing

Introduction

Bikes&More is an online bike and bike accessories reseller operating from the U.S. that ships its products to customers in the US and Canada as well as Great Britain and Australia. The company is aggressively expanding by targeting their customers directly and offering their products on line to reduce distribution cost.

Since its inception in 2000, Bikes&More has used a reliable, but increasingly outdated order transaction processing system, which management intends to upgrade to a new, more powerful system that not only supports daily business processes, but also facilitates strategic analysis of the collected business data. In later development stages, all relevant data will be collected in a central data warehouse which will feed the data into subject-specific analytic applications.

In a pilot project, you have been tasked with developing a data mart that provides management detailed access to sales data in order to analyze key business questions and support decision making.

In this pilot project, you will:
1. Implement a new schema for a transactional database and migrate legacy data to the new system.
2. Implement a star schema and create a data mart for sales analysis.
3. Define and execute an ETL process to extract, transform, and load data from the transactional database into the sales data mart.
4. Create an OLAP cube for multidimensional sales data analysis.
5. Implement a set of queries to inform management about critical business metrics.

Preliminaries: Lab Setup [0 points]

1. Make sure to have a Java runtime environment properly installed

2. Set up MySQL Community (using your package manager of choice or download it from http://dev.mysql.com/downloads/mysql/)

3. Download and extract Pentaho Data Integration Community Edition from https://sourceforge.net/projects/pentaho/

4. Download and extract Pentaho Schema Workbench Community Edition from https://sourceforge.net/projects/mondrian/files/schema%20workbench/

5. Download the MySQL JDBC driver (MySQL Connector/J) from http://dev.mysql.com/downloads/connector/j/

6. Copy the MySQL JDBC driver to the Pentaho Data integration lib folder (i.e., `data-integration/lib`) and to the Pentaho Schema Workbench drivers folder (i.e., `schema-workbench/drivers`).

Assignment 2: Data Warehousing

7. You can run the Pentaho tools using the provided shell scripts/batch files, i.e., `data-integration/spoon.sh` and `schema-workbench/workbench.sh`, respectively).

8. Download the csv data dump of the transactional database (needed in Step 1) from TUWEL.

Notes:
- If you are on MacOS or Linux, make sure that your class path does not contain any spaces or Pentaho Schema Workbench will not start (cf. http://jira.pentaho.com/browse/PSW-95)
- Pentaho Schema Workbench tends to have some (fairly annoying) GUI issues. Using the up and down arrows or saving and reopening the schema usually helps in case the tree hierarchy disappears.
- Pentaho Data Integration is quite powerful and straightforward. The Pentaho Schema Workbench, on the other hand, is not the prettiest piece of software, but it is an open source, cross-platform, lightweight, and self-contained OLAP engine and serves our purposes here.

Assignment 2: Data Warehousing

1. OLTP [5 points]

Business consultants have proposed a redesign of the logical schema of the company's operational database as part of the migration to a new system. The redesign process resulted in the database table layout depicted in Figure 1.
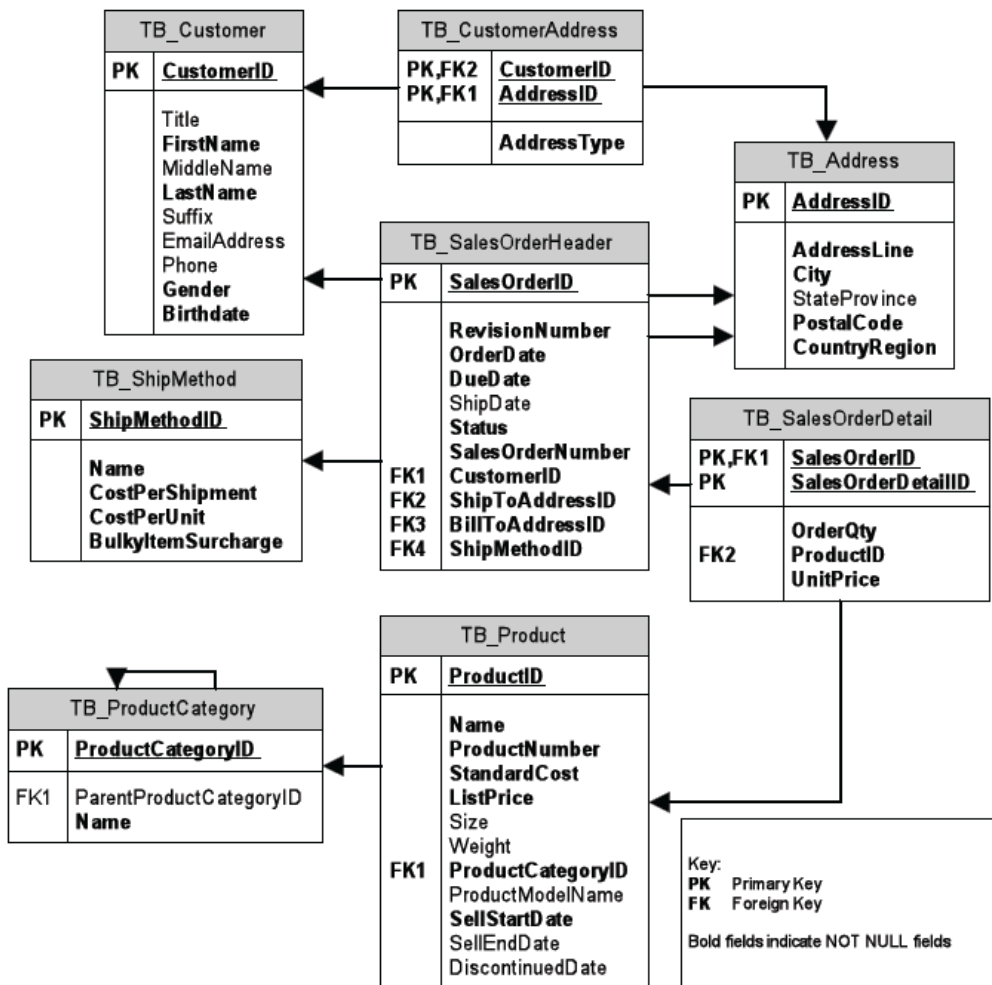

Figure 1 Logical OLTP Schema

Task description

Based on the logical relational data model depicted in Figure 1, your team is tasked with implementing the database (see Appendix: Data Dictionary) as follows:

1. **Create OLTP DB:**
   Name your MySQL transactional database `BI_OLTP_<groupNo>`
   (where `<groupNo>` is your group number)

2. **Table create scripts:**
   For each table, create an sql script `TB_<table_name>.sql` that creates the table according to the definition specified in Figure 1.

Assignment 2: Data Warehousing

a. Primary key constraints:

Define the necessary primary keys as table constraints and name them `PK_<table_name>`, where `<table_name>` is replaced with the respective name of the table, e.g. for table `TB_Address` the primary key is named `PK_Address`.

b. Foreign Key Constraints:

Name the foreign key constraints `FK_<column_name>`, where `<column_name>` is the name of the column without the 'ID' suffix that represents the reference – For example, the references between `TB_SalesOrderHeader` and for the foreign key relation between `TB_SalesOrderHeader` and `TB_Address` are represented by the columns `BillToAddressID` and `ShipToAddressID`. The corresponding foreign key constraints should be named `FK_BillToAddress` and `FK_ShipToAddress`, respectively.

Place the scripts in your solution folder under `task 1/oltp/TB_<tableName>.sql.`

3. **OLTP create job:**

   Create a Data Integration job in Pentaho Spoon to invoke the table create scripts from step 1 in the appropriate sequence.

   Detailed steps:
   a. Create a Data Integration Repository
      i. Tools → Repository → Connect..
      ii. Click the + to add a repository
      iii. Choose Kettle File repository
      iv. Select a base directory and assign `BI_<groupNo>` as a name (where <groupNo> is your group number)

   b. Create a Database connection
      i. Click the ⊕ symbol and select Database connection
      ii. Make sure you have installed the JDBC driver (MySQL connector/J)
      iii. Set the connection parameters
         1. Access: "Native (JDBC)"
         2. Host name: "localhost"
         3. Database Name: "BI_OLTP"
      iv. Name the connection `BI_OLTP_<groupNo>`

   c. Create an integration job and name it "OLTP create". Include it in your solution folder as OLTP `create.kjb.` (should work automatically if you create a Kettle file repository in your solution folder).

   d. Add the SQL create scripts to the integration job (using execute SQL steps) in an appropriate sequence.

   e. Execute the integration job to create your OLTP database

Assignment 2: Data Warehousing

4. **Load data**

The data of the legacy transactional database has already been converted to the new schema and dumped into tab-separated files (download them from TUWEL). To load the data into your OLTP database, create a new integration job named `OLTP load` in Spoon, add the database connection, and use "MySQL bulk" load steps to load the data dumps for table in the appropriate sequence (i.e., ensure that the database is in a consistent state after loading).

Note: The tsv files are tab-delimited, i.e., when setting up the MySQL bulk load steps, you need to remove the comma from the delimiter field and click "Insert TAB".

Deliverables

Put all files created in this task into the `task1` folder of your solution.

Resources

- Pentaho Data Integration (Kettle) Tutorial
  http://wiki.pentaho.com/display/EAI/Pentaho+Data+Integration+(Kettle)+Tutorial

Assignment 2: Data Warehousing

2. OLAP [25 points]

Your team has been tasked with the implementation of a sales analysis data mart as the first step towards the development of a comprehensive business intelligence system for the company. To this end, you have conducted extensive requirements elicitation interviews with the responsible managers of the sales, logistics and products departments. Your task now in this step is now to implement the corresponding star schema in a separate database and implement the process that extracts, transforms and loads the data from the transactional database.

In the following, you will find the definition of the required information package that should be implemented in a star schema (Table 1), as well as a specification of the transformations that need to be performed (Table 2).

Time	Customer	Product	Location
Date	CustomerID	ProductID	AddressID
Day Number of Month	Name*	Product Number	Postal Code
Month Number of	Birth Date	Name	City
Calendar Year	Age*	Model Name	State Province
	Gender*	Standard Cost	Country Region
	Email	List Price	
	Phone	Product Sub Category	
		Product Top Category	
		Sell Start Date	
		Sell End Date	
		Discontinued Date	
		Size	
		Weight	
		Is Bulky Item*	

FACTS: Sales Order Number, Sales Order Line Number*, *CustomerID*, *ProductID*, *ShipToAddressID*, *BillToAddressID*, Shipment Method*, Unit Price, Discount*, Order Quantity, Order Line Total*, Order Line Profit*, Tax Amount*, Order Line Freight Cost*, Order Status, *Order Date*, *Due Date*, *Ship Date*, Is Late Shipment*

Table 1 Information Package: Bikes & More sales. Underlined items denote the primary keys, italic items denote foreign keys, and items with an asterix (*) require transformations as specified in Table 2

Attribute Name	Transformation Instruction
Customer Name VARCHAR(150)	concatenate first, middle, last name and name suffix to form the customer's name in the following format: `FirstName [MiddleName.] LastName[, Suffix]`. Note that NULL values should be omitted.
Gender VARCHAR(10)	Convert 'M' to 'Male' and 'F' to 'Female'
Age INTEGER	Calculate the age from the customer's birth date. Use '2016-01-01' as end date.
Is Bulky Item (INTEGER: 0 = false	`true` if the Product is a bike, i.e. its top product category is 'Bikes', or if the

Assignment 2: Data Warehousing

Sales Order Line Number VARCHAR(10	the Sales Order Line Number is built by concatenating $'SOL' + SalesOrderID + ' - ' + SalesOrderDetailID$. Note that the resulting string must not contain any whitespaces.

Discount NUMERIC(10,4)	Calculate the discount according to the following formula: $Discount = OrderQuantity * UnitPrice * DiscountRate$ cf. Table 3 for the applicable discount rates by Top Product Category.
Order Line Total NUMERIC(10,4)	calculate the total payable amount by order line with the formula $LineTotal = OrderQuantity * UnitPrice - Discount)$
Order Line Profit NUMERIC(10,4)	calculate the profit per order line with the formula $Profit = LineTotal - OrderQuntity * StandardCost$
Tax Amount NUMERIC(10,4)	calculate the tax amount of an order line with the formula $TaxAmount = VAT * LineTotal$ cf. Table 4 for the applicable country-specific VAT rate values (use the country of the shipping address to look up the applicable tax rate)
Order Line Freight Cost NUMERIC(10,4)	calculate the freight cost as follows (the BulkyItemSurcharge should be applied only to bulky items): $Freight = OrderQuantity * PerItemCost + BulkyItemSurcharge$
Is Late Shipment (INTEGER: 0 = false <>0 = true)	`true` if the Ship Date lies past the Due Date, `false` otherwise

Table 2 OLTP to OLAP Transformations

Top Product Category		Minimum Order Quantity	Discount Rate
1	Bikes	...	none
2	Components	...	none
3	Clothing	5	5%
		10 and above	10%
4	Accessories	5	4%
		10 and above	11%

Table 3 Discount Rates per Top Product Category

Country	VAT Rate
United States	8%
Canada	13%
United Kingdom	17.5%
Australia	10%

Table 4 VAT rates by country/region

Shipment Method	Per Shipment Cost	Per Item Cost	Bulky Item Surcharge
Standard	$4	$1	+$7

Assignment 2: Data Warehousing

Cargo (International)	$7	$4	+$9
Overseas Deluxe	$30	$6	+$10

Table 5 Freight Cost by Shipment Method

Assignment 2: Data Warehousing

Task descriptions

2a) Data Mart creation
In this step, your team will design a star schema and implement the data mart.

1. Create a database named `BI_OLAP_<groupNo>`.

2. Write the SQL scripts that create the necessary fact and dimension tables and place them in a folder named `task2/DM/DM_<table name>.sql`.
   - Name the fact table `DM_FactSales`
   - Name the dimension tables `DM_<dimension_name>`
   - Omit the white space characters for the column names, i.e. `'Is Bulky Item'` becomes `'IsBulkyItem`
   - Make sure to define all foreign key constraints.

3. Create a data integration job that executes the create scripts in an appropriate sequence.

4. Run your job to create the data mart

   Notes:
   - "Date" is a keyword in SQL that you need to escape properly (i.e., `Date` in MySQL; "Date" can only be used if Ansi SQL mode is enabled)

2b) ETL
In this step, your team will develop the required ETL processes that extract the data from the operational database, transform the data according to transformation instructions specified in Table 2, and load the data into the data mart's fact and dimension tables.

You will use Pentaho Data Integration's Spoon to define the ETL process. For the actual transformations, you have two options: you can either define the transformations in plain SQL or define the ETL jobs directly in Spoon, using the built-in transformation steps. In either case, you should create a data integration job that loads the data into the data mart in an appropriate sequence to ensure consistency.

1. Implement the transformations in plain SQL or as Kettle transformations in spoon. Name the transformations or ETL scripts according to the fact/dimension name. Place the resulting files in `task 2/etl` in your solution folder.

2. Create a data integration job that executes the sql scripts or transformation in an appropriate sequence and loads all data into the data mart.

3. Execute your ETL process to populate the data mart from the transactional database.

Assignment 2: Data Warehousing

Notes: If you implement your SQL process in plain SQL, use `INSERT INTO <table_name>`
`SELECT …` commands.

2c) ROLAP Cube definition

Once the data mart has been populated, you will use Pentaho Schema Workbench to define an OLAP cube, including appropriate hierarchies for rollup and drill down analyses. In the final part of this hands on (task 3), you will then use the created OLAP cube to formulate Multidimensional Expression Query language (MDX) queries that you will report to management.

1. Start Pentaho Schema Workbench and establish a connection to your OLAP database (Options → Connection…; make sure the MySQL JDBC driver is placed in the schema workbench drivers folder)

2. Create a schema (`File → New → Schema`) and save it as `task2/OLAP/bike_sales.xml` in your solution folder

3. Add a cube and name it `bike_sales`. Add the following measures:

    a. Profit (column OrderLineProfit)
    b. Revenue (column OrderLineTotal)
    c. Quantity (column OrderQuantity)
    d. Discount
    e. FreightCost (column OrderLineFreightCost)
    f. TaxAmount
    g. IsLateShipment

    Choose appropriate aggregators for your measures and "Currency" formatString where appropriate.

4. Add the Date dimension

    a. Add the dimesion on the top level as a direct subnode of Schema
    b. Make sure to set it up as a time dimension (i.e., choose type `TimeDimension`).
    c. Add a single hierarchy named `Days` and set the `primaryKey` field to Date) and add the DM_Time dimension to it.
    d. And the three temporal levels `Year`, `Month`, and `Day`. For each level, choose the respective `table` (`CalendarYear, MonthNumberOf, DayNumberOfMonth`) and `column` and choose the correct `levelType` (`TimeYears, TimeMonths, TimeDays`).

5. Define the Location dimension:

    Add `Location` as a subnode of Schema and add a hierarchy `Areas` with levels `CountryRegion`, `State`, and `City`. Like in the previous step, set the respective tables, columns and keys.

Assignment 2: Data Warehousing

6. Define the Product dimension:

    a. Add Product as a subnode of Schema

    b. Add a hierarchy `ProductCategory` with levels TopCategory, and `SubCategory.` Again, set the respective tables, columns and keys.

    c. Add a second hierarchy `Name` with a single level `Name`

7. Define the Customer dimension with the following three single-level hierarchies and set the respective tables, columns and keys:

    a. Age

    b. Gender

    c. Name

8. Add dimension usages to the `bike_sales` cube:

- Customer (with foreign key `CustomerID`)
- Product (with foreign key `ProductID`)
- ShippedTo (using Location with foreign key `ShipToAddressID`)
- BilledTo (using Location with foreign key `BillToAddressID`)
- ShipDate (using Date with foreign key `ShipDate`)
- OrderDate (using Date with foreign key `OrderDate`)

Make sure you set the `foreignKey` and `source` fields of each dimension usage.

Notes

Schema Workbench does not warn you about missing definitions (e.g., table, primary key for dimensions) – if you don't get any values for your MDX query, check if you have set everything correctly.

Deliverables

Put all files created in this task into the *task2* folder (placing them in the appropriate DM/ETL/OLAP subfolders).

Resources

- Mondrian Documentation: http://mondrian.pentaho.com/documentation/index.php
- Youtube playlist: Pentaho Workbench OLAP Cubes – Using Pentaho Schema Workbench http://bit.ly/2gpEcP7

Assignment 2: Data Warehousing

3. Queries [20 points]

A number of users have contacted your team and requested various custom queries, which have been collected and are specified below. Your team is assigned the task to deliver the requested information using either plain SQL or MDX query language. You can choose which language is more appropriate to answer each of the questions, but you should formulate and **hand in at least three MDX queries** executed in Schema workbench.

Task description

Implement each of the following queries using either plain SQL or MDX targeting the data mart or the OLAP cube, respectively. For each query, example result set tables with sample data are provided. For MDX queries, the (albeit less useful) default result set format is fine. To develop your MDX queries, use File → New → MDX Query in the Schema Workbench.

A) Sales

1. What was the profit per country for calendar year 2013 sorted by profit in descending order (use BillToAddress and Order Date, sort by CountryRegion)?

Country Region	Profit
Australia	90000
Europe	85000
…	…

2. What was the revenue per top product category for calendar year 2014 (use the *Order Date*, sort by T*op Product Category*)?

Top Product Category	Revenue
Bikes	70000.00
Components	80000.00
…	…

3. What is the profit trend by the top product categories over the last available 24 month by month (use the *Bill to Address* and *Order Date*, sort by Y*ear*, M*onth, Top Product Category*)?

Year	Month	Top Product Category	Monthly Profit
2013	1	Accessories	3450
2013	1	Bikes	2960
2013	2	Accessories	3120
…	…	…	…

4. What are the top 5 most profitable customers and their 4 top most frequently purchased products (sort by *Customer Rank*, *Product Rank*)?

Assignment 2: Data Warehousing

Customer Rank	Customer Name	Product Rank	Product Name
1	Joe Short	1	Touring Rear Wheel
1	Joe Short	2	Touring Front Wheel
…	…	…	…
2	Sara North	1	Women's Tights, S
…	…	…	…

5. What are the top 10 most profitable customers in the first half year of 2014 (use the *Order Date*, sort by *Profit*)?

Customer Name	Profit
Joe Short	2430.00
Sara North	2350.00
Thomas Wayne	2160.00
…	…

B) Production

1. What are the 3 top selling products of each top product category (sort by T*op Product Category, Product Rank*)?

Top Product Category	Product Rank	Product Name
Bikes	1	Mountain-100 Silver, 44
Bikes	2	Road-250 Red, 44
…	…	…
Components	1	LL Crank set
…	…	…

C) Logistics

1. What is the percentage of late shipments compared to total shipments with respect to the shipment method by country/region (use the *Ship to Address*, sort by *Country/Region, Shipment Method*)?

Shipment Method	Country Region	% of Late Shipments
Cargo International	Australia	1.8000
Overseas Deluxe	Australia	0.5400
Standard Ground	Canada	1.2900
…	…	…

D) Accounting

1. What is the total of charged taxes for January through July of 2014 by month?

Month	Tax Payments
1	21000.00
2	19500.00
…	…

Assignment 2: Data Warehousing

Deliverables

Create a file for each query solution named according to the schema
`<department_numbering><#>.<query_type>` where `<department_numbering>` is either `A`, `B`,
`C`, `D` representing the Sales (`A`), Production (`B`), Logistics (`C`) or Accounting (`D`) department,
respectively, `<#>` the query number, and `<query_type>` the query format (SQL or MDX)
chosen to solve the query. E.g. if the second query of the Sales team was solved with a
MDX query, the file should be named `A2.mdx`, on the other hand, if it was solved with SQL,
it should be named `A2.sql`.

> **Note**: *Do not put any comments in these files!*
> Instead, create a separate text file named `comments.txt`.

Put all your files (including your *comments.txt* if present) into a subfolder
named *task3.*

Resources
- A Brief MDX Tutorial Using Mondrian http://cobb.typepad.com/files/mdx.pdf

Assignment 2: Data Warehousing

Appendix: Data Dictionary

Introduction/Instructions

This is a simple set of sample data files comprised of 8 files to be used to perform the lab examples. This data has already been extracted from the company's legacy order transaction systems and is ready to be loaded into the newly installed OLTP system. Each file is briefly described below. Most of the field names are self-explanatory, but in cases additional information is provided.

All source files were created as text flat files using:

- Code page 1252 (ANSI – Latin I)
- English Locale (United States)
- Data fields are Tab delimited

File: Address.csv

Fields	Key	Type	Relates to (File/Field)
AddressID	yes	Integer	CustomerAddress.csv/AddressID SalesOrderHeader.csv/ShipToAddressID SalesOrderHeader.csv/BillToAddressID
AddressLine		String	
City		String	
StateProvince		String	
CountryRegion		String	
PostalCode		String	

File: Customer.csv

Fields	Key	Type	Relates to (File/Field)	
CustomerID	yes	Integer	CustomerAddress.csv/CustomerID SalesOrderHeader.csv/CustomerID	
Title		String		
FirstName		String		
Title		String		
MiddleName		String		
LastName		String		
Suffix		String		
EmailAddress		String		

Assignment 2: Data Warehousing

Phone		String		
Gender		String		'M' for 'Male', 'F' for 'Female'
Birthdate		Date (YYYY-MM-DD)		

File: CustomerAddress.csv

Fields	Key	Type	Relates to (File/Field)
CustomerID	yes	Integer	Customer.csv/CustomerID
AddressID		Integer	Address.csv/AddressID
AddressType		String	

File: SalesOrderHeader.csv

Fields	Key	Type	Relates to (File/Field)	
SalesOrderID	yes	Integer	SalesOrderDetail.csv/SalesOrderID	
RevisionNumber		Integer		Incremental number to track changes to the sales order over time.
OrderDate		Timestamp		Dates the sales order was created.
DueDate		Timestamp		Date the order is due to the customer.
ShipDate		Timestamp		Date the order was shipped to the customer.
Status		Integer		Order current status. 1 = In process 2 = Approved 3 = Back ordered 4 = Rejected 5 = Shipped 6 = Canceled
SalesOrderNumber		String		Unique sales order identification number.
CustomerID		Integer	Customer.csv/CustomerID	
ShipToAddressID		Integer	CustomerAddress.csv/AddressID	Customer shipping address.
BillToAddressID		Integer	CustomerAddress.csv/AddressID	Customer billing address.
ShipMethodID		Integer	ShipMethod.csv/ShipMethodID	

File: SalesOrderDetail.csv

Fields	Key	Type	Relates to (File/Field)	
SalesOrderID	yes	Integer	SalesOrderHeader.csv/SalesOrderID	
SalesOrderDetailID		Integer		
OrderQty		Integer		Quantity ordered per product.
ProductID		Integer	Product.csv/ProductID	

Assignment 2: Data Warehousing

UnitPrice		Numeric		Selling price of a single product.

File: Product.csv

Fields	Key	Type	Relates to (File/Field)	
ProductID	yes	Integer	SalesOrderDetail.csv/ProductID	
Name		String		Name of the product.
ProductNumber		String		Unique product identification number.
StandardCost		Numeric		Standard cost of the product.
ListPrice		Numeric		Selling price.
Size		String		Product size.
Weight		Numeric		Product weight.
ProductCategoryID		Integer	ProductCategory.csv/ProductCastegoryID	Product is a member of this product category.
ProductModelName		String		Product is a member of this product model.
SellStartDate		Timestamp		Date the product was available for sale.
SellEndDate		Timestamp		Date the product was no longer available for sale.
DiscontinuedDate		Timestamp		Date the product was discontinued.

File: ProductCategory.csv

Fields	Key	Type	Relates to (File/Field)	
ProductCategoryID	yes	Integer	ProductCategroy.csv/ProductCategoryID Product.csv/ProductCategoryID	
ParentProductCategoryI		Integer		
Name		String		Category description.

File: ShipMethod.csv

Fields	Key	Type	Relates to (File/Field)	
ShipMethodID	yes	Integer	SalesOrderHeader.csv/ShipMethodID	
Name		String		
CostPerShipment		Numeric		
CostPerUnit		Numeric		
BulkyItemSurcharge		Numeric		