# *HW5 – Data Warehouse*

**Overview:**

- 100 points (out of 1000)
- Due on Sunday April 4th, at 11:59 p.m.
- Submit as a PDF file via the link found in the assignment section of “Week Eleven” in Canvas

This assignment gives students hands-on practice in working with a realistic data warehouse.

# *HW5 – Data Warehouse*

**Objectives**

- Create a connection to the Adventureworks Data Warehouse (“aw”) on a virtual shared server in our CU CS Private Cloud
- Examine the Adventureworks Data Warehouse to become familiar with the structure of a realistic star schema data warehouse
- Using a SQL query editor (such as MySQL Workbench)
  1. Study the database and answer questions about its structure
  2. Create and run queries against the warehouse to answer questions about the content of the warehouse
  3. Create and run analytical queries against the warehouse to analyze Adventureworks’ business

# *HW5 – Data Warehouse*

**Why?**

Challenge you to get a glimpse of what it feels like when you step into an organization and are required to figure out their databases

Study and understand an unfamiliar database

Using the tools and skills you've learned in class,

- figure out how their database is structured
- what data is in what tables
- how a realistic data warehouse is actually constructed

# *HW5 – Data Warehouse*

**Submission Requirements:**

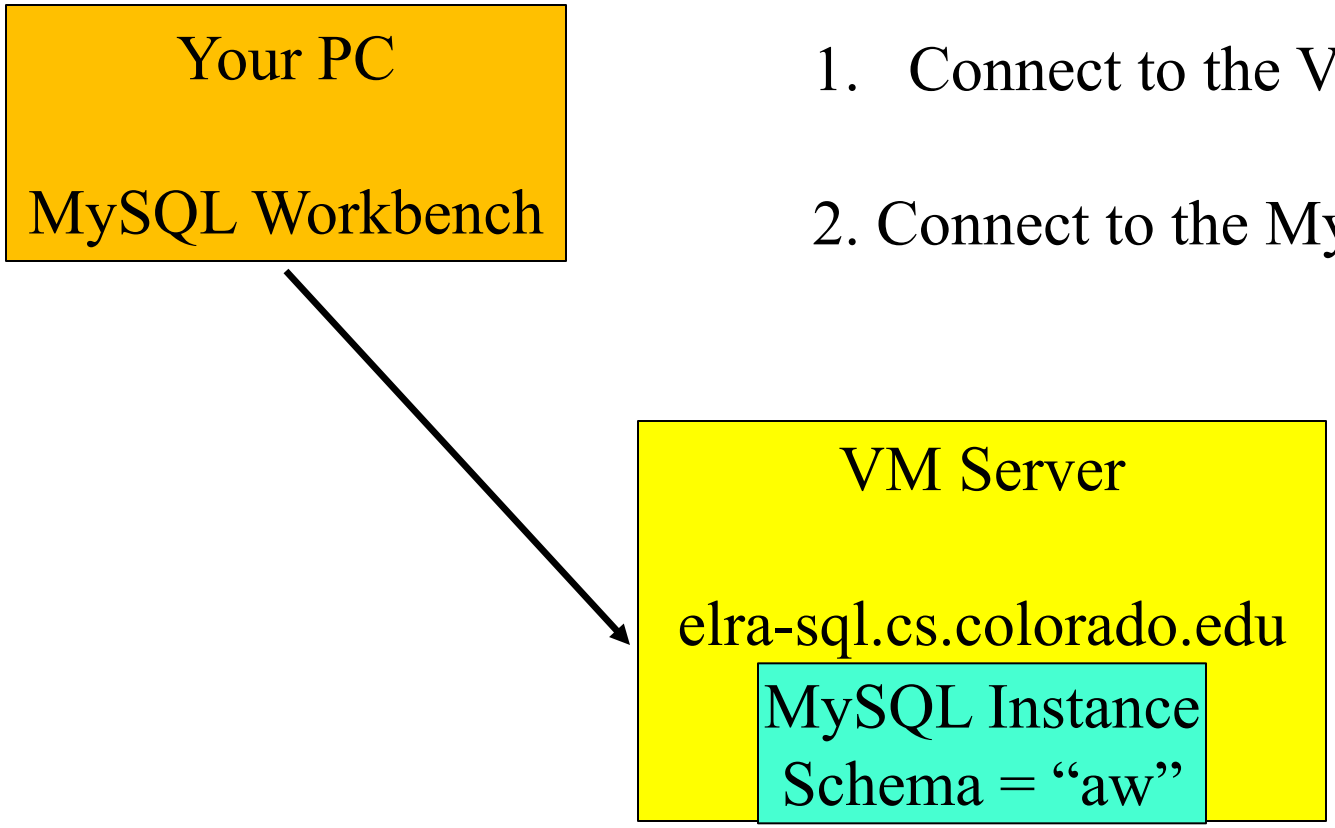
- Answers must be clearly identified and numbered according to the questions
- Where the question requires SQL, submit ***both*** your SQL code and your answer set.
- Where the question requires a written answer, submit proper sentences (spelling and grammar)
- Save the document as a PDF and submit your PDF using the Homework # 5 link in Week Eleven module in Canvas

# *HW5 – Data Warehouse*

**“Pair Programming”**

- You may work with **one** partner on completing this assignment.
- However, this is an individual assignment: each one must submit your own final deliverable
- Be sure to specify your partner’s name on the document you submit.

## Connecting to the Data Warehouse




1. Connect to the VM via SSH
2. Connect to the MySQL instance

# *HW5 – Data Warehouse*

This database is managed by MySQL running on a Linux VM server.

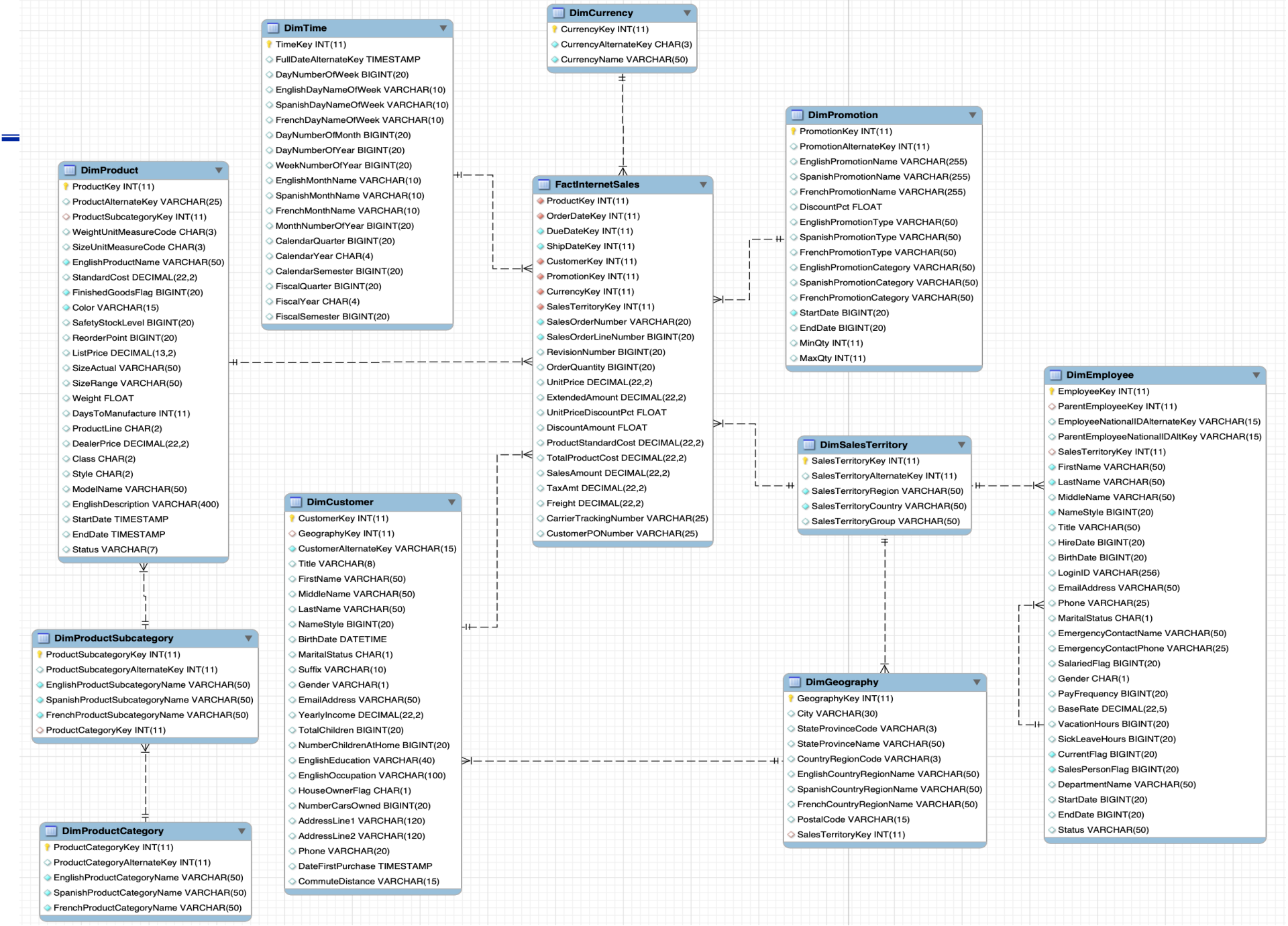
Because it is Linux, the names of database objects are case sensitive:

DimProduct ≠ Dimproduct

# *HW5 – Data Warehouse*

**Adventureworks**

- The AdventureWorks data warehouse is based on a fictitious bicycle manufacturing company named Adventure Works Cycles.
- Microsoft created this company and its databases (an OLTP database *and* a Star Schema Data warehouse) to assist their customers in learning about database technologies (in order to sell more MS SQL Server!!)
- It was built for MS SQL Server, but I converted part of it to MySQL
- Multinational: descriptions in several languages


DimTime
TimeKey INT(11)
FullDateAlternateKey TIMESTAMP
DayNumberOfWeek BIGINT(20)
EnglishDayNameOfWeek VARCHAR(10)
SpanishDayNameOfWeek VARCHAR(10)
FrenchDayNameOfWeek VARCHAR(10)
DayNumberOfMonth BIGINT(20)
DayNumberOfYear BIGINT(20)
WeekNumberOfYear BIGINT(20)
EnglishMonthName VARCHAR(10)
SpanishMonthName VARCHAR(10)
FrenchMonthName VARCHAR(10)
MonthNumberOfYear BIGINT(20)
CalendarQuarter BIGINT(20)
CalendarYear CHAR(4)
CalendarSemester BIGINT(20)
FiscalQuarter BIGINT(20)
FiscalYear CHAR(4)
FiscalSemester BIGINT(20)
DimCurrency
CurrencyKey INT(11)
CurrencyAlternateKey CHAR(3)
CurrencyName VARCHAR(50)
DimPromotion
PromotionKey INT(11)
PromotionAlternateKey INT(11)
EnglishPromotionName VARCHAR(255)
SpanishPromotionName VARCHAR(255)
FrenchPromotionName VARCHAR(255)
DiscountPct FLOAT
EnglishPromotionType VARCHAR(50)
SpanishPromotionType VARCHAR(50)
FrenchPromotionType VARCHAR(50)
EnglishPromotionCategory VARCHAR(50)
SpanishPromotionCategory VARCHAR(50)
FrenchPromotionCategory VARCHAR(50)
StartDate BIGINT(20)
EndDate BIGINT(20)
MinQty INT(11)
MaxQty INT(11)
DimProduct
ProductKey INT(11)
ProductAlternateKey VARCHAR(25)
ProductSubcategoryKey INT(11)
WeightUnitMeasureCode CHAR(3)
SizeUnitMeasureCode CHAR(3)
EnglishProductName VARCHAR(50)
StandardCost DECIMAL(22,2)
FinishedGoodsFlag BIGINT(20)
Color VARCHAR(15)
SafetyStockLevel BIGINT(20)
ReorderPoint BIGINT(20)
ListPrice DECIMAL(13,2)
SizeActual VARCHAR(50)
SizeRange VARCHAR(50)
Weight FLOAT
DaysToManufacture INT(11)
ProductLine CHAR(2)
DealerPrice DECIMAL(22,2)
Class CHAR(2)
Style CHAR(2)
ModelName VARCHAR(50)
EnglishDescription VARCHAR(400)
StartDate TIMESTAMP
EndDate TIMESTAMP
Status VARCHAR(7)
FactInternetSales
ProductKey INT(11)
OrderDateKey INT(11)
DueDateKey INT(11)
ShipDateKey INT(11)
CustomerKey INT(11)
PromotionKey INT(11)
CurrencyKey INT(11)
SalesTerritoryKey INT(11)
SalesOrderNumber VARCHAR(20)
SalesOrderLineNumber BIGINT(20)
RevisionNumber BIGINT(20)
OrderQuantity BIGINT(20)
UnitPrice DECIMAL(22,2)
ExtendedAmount DECIMAL(22,2)
UnitPriceDiscountPct FLOAT
DiscountAmount FLOAT
ProductStandardCost DECIMAL(22,2)
TotalProductCost DECIMAL(22,2)
SalesAmount DECIMAL(22,2)
TaxAmt DECIMAL(22,2)
Freight DECIMAL(22,2)
CarrierTrackingNumber VARCHAR(25)
CustomerPONumber VARCHAR(25)
DimSalesTerritory
SalesTerritoryKey INT(11)
SalesTerritoryAlternateKey INT(11)
SalesTerritoryRegion VARCHAR(50)
SalesTerritoryCountry VARCHAR(50)
SalesTerritoryGroup VARCHAR(50)
DimEmployee
EmployeeKey INT(11)
ParentEmployeeKey INT(11)
EmployeeNationalIDAlternateKey VARCHAR(15)
ParentEmployeeNationalIDAltKey VARCHAR(15)
SalesTerritoryKey INT(11)
FirstName VARCHAR(50)
LastName VARCHAR(50)
MiddleName VARCHAR(50)
NameStyle BIGINT(20)
Title VARCHAR(50)
HireDate BIGINT(20)
BirthDate BIGINT(20)
LoginID VARCHAR(256)
EmailAddress VARCHAR(50)
Phone VARCHAR(25)
MaritalStatus CHAR(1)
EmergencyContactName VARCHAR(50)
EmergencyContactPhone VARCHAR(25)
SalariedFlag BIGINT(20)
Gender CHAR(1)
PayFrequency BIGINT(20)
BaseRate DECIMAL(22,5)
VacationHours BIGINT(20)
SickLeaveHours BIGINT(20)
CurrentFlag BIGINT(20)
SalesPersonFlag BIGINT(20)
DepartmentName VARCHAR(50)
StartDate BIGINT(20)
EndDate BIGINT(20)
Status VARCHAR(50)
DimCustomer
CustomerKey INT(11)
GeographyKey INT(11)
CustomerAlternateKey VARCHAR(15)
Title VARCHAR(8)
FirstName VARCHAR(50)
MiddleName VARCHAR(50)
LastName VARCHAR(50)
NameStyle BIGINT(20)
BirthDate DATETIME
MaritalStatus CHAR(1)
Suffix VARCHAR(10)
Gender VARCHAR(1)
EmailAddress VARCHAR(50)
YearlyIncome DECIMAL(22,2)
TotalChildren BIGINT(20)
NumberChildrenAtHome BIGINT(20)
EnglishEducation VARCHAR(40)
EnglishOccupation VARCHAR(100)
HouseOwnerFlag CHAR(1)
NumberCarsOwned BIGINT(20)
AddressLine1 VARCHAR(120)
AddressLine2 VARCHAR(120)
Phone VARCHAR(20)
DateFirstPurchase TIMESTAMP
CommuteDistance VARCHAR(15)
DimGeography
GeographyKey INT(11)
City VARCHAR(30)
StateProvinceCode VARCHAR(3)
StateProvinceName VARCHAR(50)
CountryRegionCode VARCHAR(3)
EnglishCountryRegionName VARCHAR(50)
SpanishCountryRegionName VARCHAR(50)
FrenchCountryRegionName VARCHAR(50)
PostalCode VARCHAR(15)
SalesTerritoryKey INT(11)
DimProductSubcategory
ProductSubcategoryKey INT(11)
ProductSubcategoryAlternateKey INT(11)
EnglishProductSubcategoryName VARCHAR(50)
SpanishProductSubcategoryName VARCHAR(50)
FrenchProductSubcategoryName VARCHAR(50)
ProductCategoryKey INT(11)
DimProductCategory
ProductCategoryKey INT(11)
ProductCategoryAlternateKey INT(11)
EnglishProductCategoryName VARCHAR(50)
SpanishProductCategoryName VARCHAR(50)
FrenchProductCategoryName VARCHAR(50)

**Adventureworks Data Model**

- FactInternetSales Granularity: one row represents the online sale of one item by AdventureWorks to a customer.
- One sale = bike, article of clothing, accessory (like a helmet), or a repair/replacement part for a bike.
- We will only look at **bike sales**.

`UnitPrice` = the dollar amount of the sale.

`OrderQuantity` = number of items sold. (always = 1)

`OrderDateKey` = date of the sale

`ProductStandardCost` = Adventureworks' cost

`UnitPrice` - `ProductStandardCost` = profit or margin

**Adventureworks Data Model**

There is a hierarchy within the product dimensions

A specific **product** sold belongs to a **subcategory**

A **subcategory** belongs to a **category**

A category consists of many subcategories
A subcategory consists of many products

**Adventureworks Data Model**

There is an interesting design regarding geography data

- One DimCustomer row = one customer.
- A customer has a geography key
- A geography key represents one row in the DimGeography table.
- The sale of an item (one row in the fact table) is credited to an employee (sales person) who reports to one Sales Territory

## Adventureworks Data Model

- `DimProductSubcategory.EnglishProductSubcategoryName`
  = type of item sold (Bike, Component, Accessory, etc.)

- `DimProduct.EnglishProductName`
  = more detailed product/model information

- `DimProduct.ModelName`
  = specific model of bike

**Information_schema**

**The MySQL "catalog"**

```
use information_schema;
show tables;
desc tables;
Select table_name, table_rows
     from tables
     where table_name like 'nw%';
```

**Information_schema**

```
desc columns;
```

If a column is a Primary Key,

then `columns.column_key` contains “PRI”