Name ___Christopher Holmes_____

Objective:

- design a warehouse database using star schema design
- load a data warehouse table using sql against the production table
- creating a pivot table report in EXCEL
- using VISUAL EXPLAIN and QUERY STATISTICS to understand the steps involved in processing a select

Additional References

In asnwering the questions in part 6 you may want to read the additional references posted to this week's module on iLearn about business intelligence and OLAP, and Big Data and NOSQL.

Part 1: Create the HSD tables and load data

Download files

- **`Lab12 HSD CreateTablesWithData.sql`**, the production tables for the HSD (Heather Sweeney Dodge Company) and
- **`Lab12 HSDDW CreateTablesWithData.sql`,** the data warehouse tables.

Execute the Lab12 HSD CreatetablesWithData script.

Part 2: Create the HSDDW (HSD Data Warehouse) tables

Execute the script Lab 12 HSDDW CreateTablesWithData which createx the warehouse tables and will loads the HSDDW tables with data from the HSD tables. Study the script carefully to understand how the data from HSD tables is copied and transformed and loaded into the HSDDW tables. This is known as ETL (Extract – Transform -Load). ETL may be done by sql scripts in simple cases (such as here). In more complex cases it needs special application programs to filter and "scrub" the data.

Part 3: Modifying the HSD_DW warehouse tables

1. What transformations of data were made before HSD-DW was loaded with data? [List all the transformations, showing the original format of the HSD data and how they appear in the HSD-DW database.

For the customer table in the DW, the customer name was concatenated between the first name and the last name from the HSD; the domain of the email address was parsed from the email address; the area code was parsed

from the full phone number. For the Product Sales table, the amount of items that the person purchased on the same day are added to a single entry with the total of all the sales. In the timeline table, the data is merged together to have the month, quarter id, quarter string, and the year.

2. Create the SALEs_FOR_RFM table to the HSD-DW database using the create table statement below.

The HSD-DW SALES_FOR_RFM SQL Statements

```
CREATE TABLE SALES_FOR_RFM(
    TimeID              Int                 NOT NULL,
    CustomerID          Int                 NOT NULL,
    InvoiceNumber       Int                 NOT NULL,
    PreTaxTotalSale     Numeric(9,2)        NOT NULL,
    CONSTRAINT          SALES_FOR_RFM_PK
                        PRIMARY KEY (TimeID, CustomerID, InvoiceNumber),
    CONSTRAINT          SRFM_TIMELINE_FK FOREIGN KEY(TimeID)
                          REFERENCES TIMELINE(TimeID)
                            ON UPDATE NO ACTION
                            ON DELETE NO ACTION,
    CONSTRAINT          SRFM_CUSTOMER_FK FOREIGN KEY(CustomerID)
                          REFERENCES CUSTOMER(CustomerID)
                            ON UPDATE NO ACTION
                            ON DELETE NO ACTION
);
```

3. What transformations of data are necessary to load the SALES_FORM_RFM table?

4. What data will be used to load the SALES_FOR_RFM fact able? Write the complete set of SQL statements necessary to load the data.

5. Populate the SALES_FOR_RFM fact table using the SQL statement you wrote in the last question.

6. A query to summarize product units sold by Customer (CustomerName) , City, and Product(ProductName) and Year would be :

```
SELECT c.CustomerId, c.CustomerName, c.City,
       p.ProductNumber, p.ProductName,
       t.Year,t.QuarterText,
       SUM(ps.Quantity) AS TotalQuantity
 FROM customer c, product_sales ps, product p,
      timeline t
WHERE c.CustomerId = ps.CustomerID
      AND p.ProductNumber = ps.ProductNumber
```

```
        AND t.TimeId = ps.TimeID
GROUP BY c.CustomerId, c.CustomerName, c.City,
         p.ProductNumber, p.ProductName,
         t.QuarterText, t.Year
ORDER BY c.CustomerName, t.Year, t.QuarterText;
```

In the above query, product_sales table is called the ***fact table***; customer, timeline and product tables are called ***dimension tables***.

Modify the above query to report the total dollar amount of each product for each year instead of the quantity sold for each product.

```
select c.CustomerId, c.CustomerName, c.City, p.ProductNumber,
p.ProductName,
t.Year, t.QuarterText, SUM(ps.Quantity)*(ps.unitprice) AS
TotalDollarAmount
from customer c, product_sales ps, product p, timeline t
where c.CustomerID = ps.CustomerID and p.ProductNumber =
ps.ProductNumber and t.TimeId - ps.TimeID
group by c.CustomerId, c.CustomerName, c.City, p.ProductNumber,
p.ProductName, t.QuarterText, t.Year, ps.unitprice
order by c.CustomerName, t.Year, t.QuarterText
```

7. Write an SQL View equivalent of the SQL query you wrote in question 6

```
create view total_per_year as
select c.CustomerId, c.CustomerName, c.City, p.ProductNumber,
p.ProductName,
t.Year, t.QuarterText, SUM(ps.Quantity)*(ps.unitprice) AS
TotalDollarAmount
from customer c, product_sales ps, product p, timeline t
where c.CustomerID = ps.CustomerID and p.ProductNumber =
ps.ProductNumber and t.TimeId - ps.TimeID
group by c.CustomerId, c.CustomerName, c.City, p.ProductNumber,
p.ProductName, t.QuarterText, t.Year, ps.unitprice
order by c.CustomerName, t.Year, t.QuarterText
```

8. Create the view that you defined in the last question.

9. Modify the design of the HSD-DW dimensional database to include a PAYMENT_TYPE dimension table.

10. What data will be used to load the PAYMENT_TYPE table? Write the complete set of SQL necessary to load these data.

Part 4: Create a pivot table in EXCEL

[Note: your version of EXCEL may be slightly different from what is described in the instructions below]

You must be running a Windows computer to do part 4. You need to have ODBC connector installed and configured. Watch the video on insalling and configuring ODBC.

If you do not have a Windows computer, skip to Part 4a.

Using the DATA tab in EXCEL, select "Get External Data" and then "Microsoft query" and then select the odbc data source for the hsddw database.

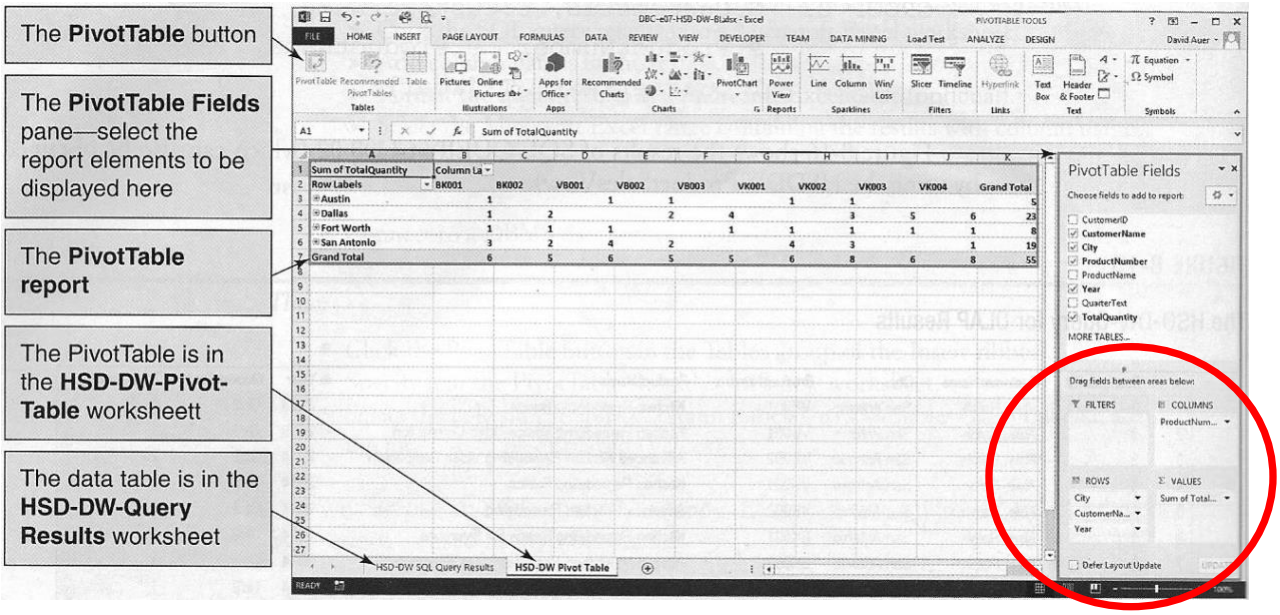
Select the table and columns for the query in Part 3 #6 that summarizes by TotalQuantity. You can select the tables, column and specify the ordering using the Microsoft Query dialogs but you will have to finish editing the query. The final query should look like the query in problem 6 in part 3.

Execute the query and then return data to the EXCEL Worksheet as TABLE.

Go the INSERT tab and use the PIVOT table tool.

Drag fields City, CustomerName and Year into the ROWS panel and ProductNumber into the COLUMNS panel and TOTALQUANTITY field into the VALUES panel. Your pivot table should look like the following.

OLAP ProductNumber by City Report



You can expand (called drilling down on) each City to get details on CustomerName and Year.

The City = San Antonio data are also showing customer data

The Customer = Able, Ralph data are also showing year data

Sum of TotalQuantity	Column La									
Row Labels	BK001	BK002	VB001	VB002	VB003	VK001	VK002	VK003	VK004	Grand Total
Austin	1			1	1		1	1		5
Pearson, Bobbi	1			1	1		1	1		5
Dallas	1	2			2	4		3	5	6 23
Foxtrot, Kathy		1				1		1	1	1 5
George, Sally	1	1							1	2 5
Hullett, Shawn						1			1	1 3
Tyler, Jenny					2	2		2	2	2 10
Fort Worth	1	1	1			1	1	1	1	1 8
Jacobs, Nancy	1		1				1			3
Wayne, Joan		1				1		1	1	1 5
San Antonio	3	2	4	2		4	3		1	19
Able, Ralph	1	1	2	2		2	2			10
2013			1			1				2
2014	1	1	1	2		1	2			8
Baker, Susan	1	1	1			1	1		1	6
Eagleton, Sam	1		1			1				3
Grand Total	6	5	6	5	5	6	8	6	8	55

Take a screenshot of your pivot table and paste here.

Part 4a: Create a pivot table in Google Sheets

Using MySql Workbench perform the query in Part 3 #6 that summarizes by customer, city, quarter and total dollar amount sold. Export the result into a csv file.

Create a blank google sheet and do File → Import. Select the upload option and upload the csv file.

Next do Data → Pivot Table.

Click Add Rows and add City, CustomerName and Year.

Click Add Columns and add ProductNumber.

Finally Click Add Values and add TOTALQUANTITY.

Your pivot table should look like the following.

OLAP ProductNumber by City Report

Untitled spreadsheet

File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive

Jacobs Nancy

SUM of TotalQuantity			ProductNumber								
City	CustomerName	Year	BK001	BK002	VB001	VB002	VB003	VK001	VK002	VK003	VK004	Grand Total
Auston	Pearson Bobbi Total		1			1	1		1	1		5
Auston Total			1			1	1		1	1		5
Dallas Total			1	2			2	4		3	5	6 23
Fort Worth	Jacobs Nanc	2014	1			1			1			3
	Jacobs Nancy Total		1			1			1			3
	Wayne Joan Total			1				1		1	1	1 5
Fort Worth Total			1	1	1			1	1	1	1	1 8
San Antonio Total			3	2	4	2		4	3		1	19
Grand Total			6	5	6	5	5	6	8	6	8	55

You can expand (called drilling down on) or collapse each City to get details on CustomerName, Year and Product.

Take a screenshot of your pivot table and paste here.

SUM of TotalDollarAmount			ProductNumber									
City	CustomerName	Year	BK001	BK002	VB001	VB002	VB003	VK001	VK002	VK003	VK004	Grand Total
Auston	Pearson Bobl	2013	74.85		23.97	23.97		44.85	44.85			212.49
		2014	199.6		63.92	63.92		119.6	119.6			566.64
	Pearson Bobbi Total		274.45		87.89	87.89		164.45	164.45			779.13
Auston Total			274.45		87.89	87.89		164.45	164.45			779.13
Dallas	Foxtrot Kathy	2013		74.85			29.97		44.85	59.85	74.85	284.37
		2014		199.6			79.92		119.6	159.6	199.6	758.32
	Foxtrot Kathy Total			274.45			109.89		164.45	219.45	274.45	1042.69
	George Sally	2013	74.85	74.85						59.85	124.75	334.3
		2014	199.6	199.6						159.6	424.15	982.95
	George Sally Total		274.45	274.45						219.45	548.9	1317.25
	Hullett Shawr	2013					29.97			59.85	74.85	164.67
		2014					79.92			159.6	199.6	439.12
	Hullett Shawn Total						109.89			219.45	274.45	603.79
	Tyler Jenny	2013				47.94	59.94		89.7	119.7	149.7	466.98
		2014				127.84	159.84		239.2	319.2	399.2	1245.28
	Tyler Jenny Total					175.78	219.78		328.9	438.9	548.9	1712.26
Dallas Total			274.45	548.9		175.78	439.56		493.35	1097.25	1646.7	4675.99
Fort Worth	Jacobs Nanc	2013	74.85		23.97			44.85				143.67
		2014	199.6		63.92			119.6				383.12
	Jacobs Nancy Total		274.45		87.89			164.45				526.79
	Wayne Joan	2013		74.85			29.97		44.85	59.85	74.85	284.37
		2014		199.6			79.92		119.6	159.6	199.6	758.32
	Wayne Joan Total			274.45			109.89		164.45	219.45	274.45	1042.69
Fort Worth Total			274.45	274.45	87.89		109.89	164.45	164.45	219.45	274.45	1569.48
San Antonio	Able Ralph	2013	74.85	74.85	39.95	47.94		74.75	89.7			402.04
		2014	199.6	199.6	135.83	127.84		254.15	239.2			1156.22
	Able Ralph Total		274.45	274.45	175.78	175.78		328.9	328.9			1558.26

Part 5: Visual Explain and Query statistics

Copy and paste the query from part 3 question 6 into a query tab in MySQL Workbench. Execute the query scroll down the right-hand side of the result panel and click "Execution Plan" and "Query Stats".

Paste a screenshot of the execution plan here.

id	select_type	table	partitions	type	possible_keys	key	key_len	ref
1	SIMPLE	p		ALL	PRIMARY			
1	SIMPLE	ps		ref	customerid_fk,product_fk	product_fk	35	hsddw.p.productnumber
1	SIMPLE	c		eq_ref	PRIMARY	PRIMARY	4	hsddw.ps.customerid
1	SIMPLE	t		ALL				

Explain in words what the execution plan shows and the query stats. [Consult the MySQL online reference or ask in the Q&A forum if you need help reading the execution plan]

For your information:

When a query is taking a very long time, one of the responsibilities of a DBA often will be to investigate how the server is processing the query and why it is taking so long and what can be done to decrease the time. The visual explain and query stats are used to analyze how the server is processing the select statement and may lead some ideas to improve performance. Some things that a DBA might do to improve performance include:

- *reorganize the table using OPTIMIZE TABLE command so the table rows are in physical sequence on the HDD by primary key*
- *Make use of VARCHAR datatype (instead of CHAR), use COMPRESSED or COMPACT options to reduce size of rows and tables to reduce the number of IOs and time needed to scan the table*
- *Create additional indexes or rearrange the column order of a multi-column index.*
- *How a select statement is processed (which indexes are used, how multiple indexes are used, how much of the table has to scanned and filtered, is the table and index files accessed in sequential or random order) depends on*
  - *the sql statement*
  - *the table statistics*
  - *DBMS server type and version*
  - *DBMS Configuration options*

Part 6 : Additional questions.

1. What is a BI (Business Intelligence) system?

   A Business Intelligence system is a system that utilizes a technology-driven process to be able to analyze data and create actionable information that decision makers in a business can use to make informed decisions.

2. How does a BI system differ from a transaction processing system?

   A BI System does not support the same operational activities that a transaction processing system would; such as recording and processing orders

3. What is an ETL system, and what functions does it perform?

   An ETL system is a system that will read data from an operational database.

4. What is the enterprise data warehouse (EDW) architecture?

   Is a combination of the data mart structure and data warehouse architecture. The data warehouse is the authority and maintains all enterprise BI data. The data mart gets all the data from the data warehouse and does not add any data.

5. What is a star schema?

A star schema is a design that will use a denormalized design that stores historical data. There is a fact table at the center, and all the other dimensional tables radiating out from that center. A fact table will always be fully normalized, but a dimension table may be not.

6. What does OLAP stand for?

Online Analytical Processing

7. What are the distinguishing features of a OLAP reports?

-Produce OLAP reports -Dynamic allowing a user to change the format

8. What is *Big Data*?

Big data is the term that is used for extremely large datasets that can be generated by web applications; Facebook, Google, Twitter.

9. What is the *NOSQL* movement?

The NOSQL movement is the movement to have the big data stored in nonrelational dmb's.