Assignment 4

Chapter14:

1. A(n) _______ is a process or set of operations in a calculation.

a. algorithm
b. feedback loop
c. stream
d. structure

2. Big Data

a. relies on the use of structured data
b. captures data in whatever format it naturally exists
c. relies on the use of unstructured data
d. imposes a structure on data when it is captured

3. In the context of Big Data, ______ refers to the trustworthiness of a set of data.

a. value
b. variability
c. veracity
d. viability

4. In the context of Big Data, ______ relates to differences in meaning.

a. variety
b. variability
c. veracity
d. viability

5. Modeling and storing data about relationships is the focus of:

a. key-value databases
b. column-oriented databases
c. document databases
d. graph databases

6. To query the value component of the pair when using a key-value database, use get or:

a. store
b. fetch
c. retrieve
d. gather

7. When using MapReduce, a ________ function takes a collection and data and sorts and filters it into a set of key-value pairs.

a. reduce
b. map
c. data
d. block

8. When using MapReduce, best practices suggest that the number of mappers on a given node should be:

a. 100 or more
b. 100 or less
c. 50 or less
d. at least 300

9. When using a HDFS, a heartbeat is sent every _____ to notify the name node that the data mode is still available.

a. 3 hours
b. 3 seconds
c. 6 hours
d. 6 seconds

10. Which of the following is NOT a key assumption of the Hadoop Distributed File System?

a. High volume
b. Write many, read-once
c. Streaming access
d. Fault-tolerance

11. Which of the following is NOT one of the standard NoSQL categories?

a. document databases
b. column-oriented databases
c. graph databases
d. chart databases

12. _____ is NOT one of the "3 Vs" of Big Data.

a. Volume
b. Velocity
c. Validation
d. Variety

13. ______ minimizes the number of disk reads necessary to retrieve a row of data.

a. Column-oriented database
b. Row-centric storage
c. Column-family database
d. Column-centric storage

14. ______ uses statistical analysis to answer questions about the how and why of relationships.

a. Explanatory analytics
b. Data mining
c. Predictive analytics
d. Knowledge acquisition

15. ______ uses statistical tools to answer questions about future data occurrences.

a. Explanatory analytics
b. Data mining
c. Predictive analytics
d. Knowledge acquisition

16. ________ processing occurs when a program runs from beginning to end without any user interaction.

a. Hadoop
b. Block
c. Hive
d. Batch

17. Document databases group documents into logical groups called:

a. buckets
b. sets
c. collections
d. blocks

18. Most BI vendors are dropping the term “data mining” and replacing it with the term:
a. explanatory analytics
b. data analytics
c. predictive analytics
d. knowledge acquisition

19. When using a HDFS, the _____ node creates new files by communicating with the ____ node.
a. client, name
b.name, client
c. client, data
d. data, client

20. By default, Hadoop uses a replication factor of:
a. one
b. two
c. three
d. four

21. _______ is keeping the same number of systems, but migrating each system to a larger system.
a. Clustering
b. Scaling up
c. Streaming
d. Scaling out

22. The goal of the _____ phase of data mining is to identify common data characteristics or patterns.
a. data preparation
b. data analysis and classification
c. knowledge acquisition
d. prognosis

23. Two of the most popular applications to simplify the process of creating MapReduce jobs are Hive and
a. Flume
b. Pig
c. Sqoop
d. Impala

24. ______ focuses on filtering data as it enters the system to determine which data to keep and which to discard.
a. Scaling up
b. Feedback loop processing
c. Stream processing
d. Scaling out

25. ______ is a tool for converting data back and forth between a relational database and the HDFS.
a. Flume
b. Pig
c. Sqoop
d. Impala

26. ______ was the first SQL-on-Hadoop application.
a. Flume
b. Pig
c. Sqoop
d. Impala

Deliverables

RemoteLabs MS SQL Server
http://labs.cci.fsu.edu/
Important: A5 requires the completion of A4!

Business Rules:
A high-volume home office supply company contracts a database designer to develop a system in order to track its day-to-day business operations. The CFO needs an updated method for storing data, running reports, and making business decisions based upon trends and forecasts, as well as maintaining historical data due to new governmental regulations. Here are the mandatory business rules:

- A sales representative has at least one customer, and each customer has at least one sales rep on any given day (as it is a high-volume organization).
- A customer places at least one order. However, each order is placed by only one customer.
- Each order contains at least one order line. Conversely, each order line is contained in exactly one order.
- Each product may be on a number of order lines. Though, each order line contains exactly one product id (though, each product id may have a quantity of more than one included, e.g., "**oln_qty**").
- Each order is billed on one invoice, and each invoice is a bill for exactly one order (by only one customer).
- An invoice can have one (full), or can have many payments (partial). Though, each payment is made to only one invoice.
- A store has many invoices, but each invoice is associated with only one store.
- A vendor provides many products, but each product is provided by only one vendor.
- Must track yearly history of sales reps, including (also, see Entity-specific attributes below): yearly sales goal, yearly total sales, yearly total commission (in dollars and cents).
- Must track history of products, including: cost, price, and discount percentage (if any).

Notes:
- A customer's contact (in-store or online) is made through a sales rep.
- A customer buys or potentially buys products from the company, but does not have to.
- An order is a purchase of one or more products by a customer. If an order is cancelled, it is deleted (optional participation).
- An order line contains the details about each product sold on a particular customer order, and includes data such as quantity and price.
- A product is an item that the company sells that was initially bought from an outside vendor (which may also be the manufacturer).
- A sales rep receives a 3% commission based upon the amount of year-to-date sales.
- A sales reps's current yearly sales goal is 8% more than their previous year's total sales.

Additional Notes:
- Social security numbers, should be unique, and hashed **and** salted for security purposes.
- ERD MUST include relationships, though, **not** cardinalities.
- Appropriate attributes *__are__* required (e.g., name, ssn (for sales rep and customer), dob, address, phone, email, url… also, see Assignment Guidelines, and Notes above),

Entity-specific attributes (apart from attributes required in Assignment Guidelines)**:**
Sales rep: ssn, current year sales goal (stored derived attribute, 8% of previous year's total sales), year-to-date sales, year-to-date commission (stored derived attribute, 3% of year-to-date sales).
Customer: ssn, balance and total sales
Order: placed and filled dates
Order line: quantity and price
Product: name, description, weight, quantity-on-hand, cost, price, discount (as percentage)
Invoice: store id, date, total, paid (bit value), **ord_id** must contain a unique index to create a one-to-one relationship with **order**!
Payment: date, amount
Sales rep history: action (insert, update, delete), modified (timestamp), modifier (user making modification), date (only year), yrly sales goal, yrly total sales, yrly total comm.
Product history: date (datetime), cost, price, discount (as percentage).

MS SQL Server

1. **Using RemoteLabs, log into SQL Server:** http://labs.cci.fsu.edu/
2. **Must** populate tables using **T-SQL *or* Table Designer (if chosen, research how to use it)**
   NOTES: Tables *must* include the following constraints and defaults:

- per_ssn: must be unique (see indexes/keys), and **SHA2_512** hashed and salted
- per_gender: m or f
- per_type: c or s

Example: ([per_gender]='f' OR [per_gender]='m')

- state: default = FL
- zip: require entries in zip column to be 9 digits

Example: ([per_zip] like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')

- phone num: require entries in phone column to be 10 digits

Example: ([phn_num] like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9] [0-9]')

- phone type: home, cell, work, fax

Example: ([phn_type]='f' OR [phn_type]='w' OR [phn_type]='c' OR [phn_type]='h')

- ***all*** numeric values: >= 0

Example: ([srp_yr_sales_goal] >= (0))

3. **FK:** Must require ON DELETE CASCADE, ON UPDATE CASCADE

4. Include at least **10 *unique*** records in the **person** table, and at least **5 *unique* records** in all other tables.

Video helper files:
1. LIS3781 A4a: https://youtu.be/acr4eQ610BY
2. LIS3781 A4b: https://youtu.be/YK242RlPEjc
3. LIS3781 A4c: https://youtu.be/sB_LCKaE4Ao
4. LIS3781 A4d: https://youtu.be/skSlheq4LeQ

Note: In the videos, data for the phone table is not reviewed, ***be sure*** to add phone data!

Changes to Videos:

1. ***Be sure*** to use the updated **person** table and **stored procedure** files uploaded as images to the A4 module in Canvas. Use the code in the updated files, in lieu of the similar code shown in the project videos: **per_ssn** should be salted and hashed, as per the image files.
   The files demonstrate how to auto-generate data for testing purposes.
   More importantly, they demonstrate how to obfuscate data using unique salt and hash values for each record.
2. **Except for the person table (see images online), ***BE SURE*** ** TO USE THE DATA IN THE VIDEOS!!! (**No debugging assistance** will be provided if other data values are used!)

Note: the stored procedure could be easily modified to obscure real data (e.g., social security numbers, credit card numbers, drivers' license numbers, etc.). Lastly, ***always*** use a secure connection!

Deliverables

1. **ERD (tables *must* be populated using RemoteLabs – MS SQL Server)**
2. **SQL Statement Questions**

No Credit will be given if tables and data are not populated in RemoteLabs MS SQL Server.

Note:

README.md file should include the following items:
1. Screenshot of ***your*** ERD;
3. Optional: SQL code for the required reports.
4. Bitbucket repo links: ***Your*** lis3781 Bitbucket repo link

Deliverables (see screenshots below):

1. Provide **Bitbucket** read-only access to **course** repo, using Markdown syntax, (**README.md** must also include screenshots per above.)
   (**DO NOT create README in Bitbucket**—ALWAYS do it locally, then push it to Bitbucket.)
2. **FSU's Learning Management System**: include course **Bitbucket** repo link

SQL Statements for A4

The following items are *required* (use RemoteLabs – MS SQL Server):

Optional: All dollar amounts must be formatted to two decimal places, including a dollar sign ($). All phone numbers and zip codes must include proper hyphens (-).

1) Create a view that displays the sum of all paid invoice totals for each customer, sort by the largest invoice total sum appearing first.
2) Create a stored procedure that displays all customers' outstanding balances (unstored derived attribute based upon the difference of a customer's invoice total and their respective payments). List their invoice totals, what was paid, and the difference.
3) Create a stored procedure that populates the sales rep history table w/sales reps' data when called.
4) Create a trigger that automatically adds a record to the sales reps' history table for every record added to the sales rep table.
5) Create a trigger that automatically adds a record to the product history table for every record added to the product table.

*****Extra Credit*****

Create a stored procedure that updates sales reps' yearly_sales_goal in the **slsrep** table, based upon 8% more than their previous year's total sales (**sht_yr_total_sales**), name it **sp_annual_salesrep_sales_goal**. (See Notes above.)

MS SQL Server Notes:

Save (Not Permitted) Dialog Box:
The Save (Not Permitted) dialog box warns you that saving changes is not permitted because the changes you have made require the listed tables to be dropped and re-created.

To change this option, on the Tools menu, click Options, expand Designers, and then click Table and Database Designers. Select or clear the Prevent saving changes that require the table to be re-created check box.

Creating Tables in MS SQL Server Using Management Studio:
http://databases.about.com/od/sqlserver/ss/sqlservertables.htm

http://www.youtube.com/watch?v=m0-vZZI0QFA

http://www.cs.trinity.edu/~thicks/Tutorials/MSSQL-Server-Management-Studio-DB-Construction/MSSQL-Server-Management-Studio-DB-Construction.html