ITMD 523 *Advanced Topics in Data Management* HW 5

Student Name **Sivaranjani Prabasankar** Section

Instructor ***Luke Papademas*** Due Date **3/30/2019**

Part	1	2	3	4	**TOTAL**	**Score**
Maximum Points	25 points	25 points	25 points	25 points	**100** points	

Textbook Reading Assignment Thoroughly read Week 1 - 9 course lecture notes.

Part 1 Concepts, Topics, Glossary Terms - Advanced Topics in Data Management

Comment and expound, in detail, on exactly any five of these concept distinctions from the realm of database management and design computer topics, in general. If applicable, use examples to support your definitions and indicate when and / or where the individual concepts would apply.

NoSQL versus ANSI SQL
https://www.sitepoint.com/sql-vs-nosql-differences/

Activity Diagram versus Sequence Diagram
http://www.ibm.com/developerworks/rational/library/2802.html

http://www.ibm.com/developerworks/rational/library/3101.html

Primary Key versus Unique Index
http://www.dotnet-tricks.com/Tutorial/sqlserver/V2bS260912

http://www.differencebetween.com/difference-between-primary-key-and-vs-unique-key/

http://docs.oracle.com/cd/B19306_01/server.102/b14200/clauses002.htm

Commit versus Rollback
https://social.msdn.microsoft.com/Forums/en-US/home

ACID versus BASE
Article: "Abandoning Acid in Favor Of Base"
http://databases.about.com/od/otherdatabases

Transitive Functional Dependency versus Trivial Functional Dependency
Article: "Transitive Dependency"
http://databases.about.com/od

http://databases.about.com/od/specificproducts/g/transitive-dependency.htm

Linear Regression versus Multiple Regression
Article: "what is the Difference Between Linear Regression and Multiple Regression"
http://www.investopedia.com/ask/answers/060315/

Distributed Database versus Graph Database
http://whatis.techtarget.com/definition/graph-database

B - Tree versus Tree
http://searchsqlserver.techtarget.com/definition/B-tree

Business Intelligence versus Data Science
http://www.ibmbigdatahub.com/technology/business-intelligence

http://www.ibm.com/analytics/us/en/

(a) Concept Comparison 1

NoSQL	ANSI SQL
Dynamic schema for unstructured data	Predefined schemas to determine the structure of data
Data is stored in many ways: it can be column-oriented, document-oriented, graph-based or organized as a Key Value store	Data is stored in Tables
Horizontally scalable. ➔ handles more traffic by adding more servers	Vertically scalable ➔ increase the load on a server by increasing CPU, RAM or SSD capacity
Doesn't require a database administrator	Require a database administrator
MongoDB, BigTable, Redis, RavenDB Cassandra, HBase, Neo4j and CouchDB	MySQL, Oracle, PostgreSQL, and Microsoft SQL Server.

(b) Concept Comparison 2

Transitive Functional Dependency

- ⇒ A transitive dependency in a database is an indirect relationship between values in the same table that causes a functional dependency.
- ⇒ To achieve the normalization standard of Third Normal Form (3NF), you must eliminate any transitive dependency.
- ⇒ A transitive dependency requires three or more attributes (or database columns) that have a functional dependency between them
- ⇒ **Example:**
  Column A in a table relies on Column B through an intermediate Column C.
  Book → Author and Author → Author_Nationality
  Book →Author_Nationality determine the nationality via the Author column.

Trivial Functional Dependency

- ⇒ Trivial functional dependency exists when one attribute determines another attribute uniquely in a database.
- ⇒ Occurs when describing a functional dependency of an attribute or of a collection of attributes that includes the original attribute.
- ⇒ This kind of dependency is called trivial because it can be derived from common sense. If one "side" is a subset of the other, it's considered trivial. The left side is considered the determinant and the right the dependent.
- ⇒ **Example:**
  {A,B} –> B is a trivial functional dependency because B is a subset of A,B.
  Since {A,B} –> B includes B, the value of B can be determined.
  {Employee_ID, Employee_Name} –> Employee_ID is also a trivial functional dependency

(c) Concept Comparison 3

	Distributed Database	Graph Database
Description	Logically interrelated with each other, and they often represent a single logical database	collection of nodes and edges
Used	Well suited for vertical scalability and horizontal partitions Blockchain or (Database sharding)	well-suited for analyzing interconnections
Flexibility	Require many communications and additional calculations to provide uniformity in data across the sites.	The data captured can be easily changed and extended for additional attributes and objects
High Availability	Identical copy of the physical database in a separate hardware instance	Most common graph databases stores all the data on *one* server and are limited to a single node and can't scale beyond a certain point
Indexing	The need for updating data in multiple sites pose problems of data integrity and slower update	Naturally indexed by relationships and comparatively provides faster access to relational data
Volume of data	cluster database - multiple physical copies of the entire database are kept synchronized	Not optimized for large-volume analytics queries typical of data warehousing
Search	Improper data distribution often leads to very slow response to user requests	Runs fast relationship-based searches

(d) Concept Comparison 4

Description	COMMIT	ROLL BACK
Basic	COMMIT validates the modifications made by the current transaction	ROLLBACK erases the modifications made by the current transaction
Effect	After execution of COMMIT statement, the transaction cannot be ROLLBACK	Once ROLLBACK is executed database reaches its previous state, i.e. before the execution of the first statement of the transaction.
Occurrence	COMMIT occurs when the transaction gets executed successfully	ROLLBACK occurs when the transaction is aborted in middle of the execution
Syntax	COMMIT;	ROLLBACK;

(e) Concept Comparison 5

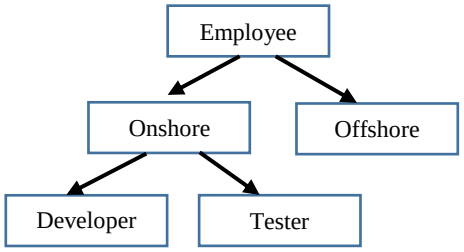
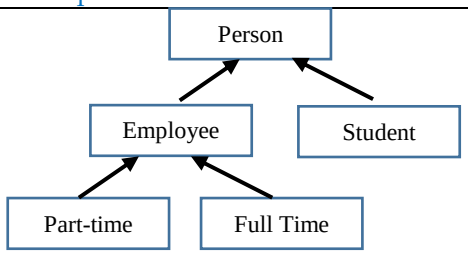
Business Intelligence	Data Science
Simpler Version	More Complex
About dashboards	Arranging data and produce information from it
Explore past trends of data	Finds predictors and significance behind the trends
Aim of the job is to assist in strategic business decisions & would require proficiency in data handling tools.	Aim of it is to derive decisions based on predictive algorithms and may require more technical skillsets in statistics, machine learning and programming
They help in viewing the relationships between various variables	More explicitly relies on predictive analytics, using the statistical method.

Part 2 DBMS Concepts - Advanced Topics in Data Management

(1) (Advanced Data Modeling)

In the realm of advanced database modeling, differentiate between specialization and generalization. Provide some examples to support your discussion of this distinction.

Specialization	Generalization
Top-Down approach	Bottom-Up approach
Uses to identify the subset of an entity set that shares some same characteristics	Extract the common features of multiple entities to form a new entity
Forms the multiple entity from a single entity	Forms a single entity from multiple entities

The superclass is defined first, the subclass and its related attributes are defined next	Subclasses are combined to make a superclass.
Increases the number of entities, and thus increases the size of the schema	Reduces the size of the schema
Example	Example
	

(2) (Advanced Data Modeling)

Explain the " no change over time " characteristic of a primary key. Provide some examples to support your discussion of this important characteristic.

Consider an attribute in table has semantic meaning, then it might subject to updates.
Example: Consider Name as the primary key in a database
what if a person changes their name or spelling? Then,

- ⇒ A primary key subject to change
- ⇒ The foreign key values must be updated
- ⇒ Adds to the database work load

But it is **not possible** in database as the PK should be **permanent and unchangeable** which we can call it as **No change over time.**
Furthermore, changing a primary key value means that you are basically changing the identity of an entity.

Part 3 Data Modeling Concepts - Advanced Topics in Data Management

(1) (Entity Relationship Diagrams with MS Visio)

Create ER - Diagrams using MS Visio 2016

https://www.youtube.com/watch?v=knvE3L57qrI

Review the video given at the link above and discuss some procedures that you would use to create and design an Entity - Relationship Diagram. How does an ERD differ from an Data Flow Diagram (DFD) ?

DFD and ERD are different data models that are mainly used for organizing business data for proper communication between members of a group.

ERD – Entity Relationship Diagram

- The ERD model represents the system data and includes an elaborate description of the relation between the data.

- It represents the entity model and will show what a system or a database will look like but not explain how to implement it. All the entities should represent a group of similar things.
- All the definitions in ERD should be unambiguous.

DFD – Data Flow Diagram

- The DFD model is a multi-level representation that commences with abstract information and includes multiple decomposed levels.
- It shows how data enter a system, are transformed in that system, and how it is stored in it.
- Each of the processes and the storing should have at least one data flow going towards it and one leaving it. All the data must have to go through a certain process, and all the processes in a system should be linked to a data store or another process.

Difference between DFD and ERD

⇒ DFD shows how data enter a system, are transformed in that system, and how it is stored in it whereas ERD represents the entity model and will show what a system or a database will look like but not explain how to implement it.

⇒ With DFD, each of the processes and storing should have at least one data flow going towards it and one leaving it but with ERD, all the entities should represent a group of similar things. All the definitions in ERD should be unambiguous.

⇒ DFD is represented by ovals, rectangles, or circles and is named with a single word and the ERD is represented by a rectangular box.

(2) (Process Mapping)

Review the video given at the link below and discuss how process mapping can be used in a database system.
https://www.youtube.com/watch?v=LJwKZuQUb7g

Process Mapping

The flow of work in a database is generally described in a process map. It is a planning and management tool that visually describes a flow in a database. It depicts a series of events that usually end up in the result of those events. A process map is also called a flowchart, process flowchart, process chart, functional process chart, functional flowchart, process model, workflow diagram, business flow diagram or process flow diagram.

Process maps shows who and what is involved in a process that can be used in any business and organization. It gives an overall perspective and hence can support in understanding where process improvements can take place.
The primary advantage of using process mapping in a database system is to see the areas of improvement and visualize ideas put into perspective and actionable items.

Flowcharts and process maps can be used to the following cases:

- Increase process understanding
- Analyze process improvements
- Show process steps
- Improve communication between individuals engaged in the same process
- Provide process documentation
- Plan projects efficiently

Student Name **Sivaranjani Prabasankar** Section _____

Part 4 Data Design Concepts - Advanced Topics in Data Management

Review the script associated with this assignment that has the following tables (with data) .

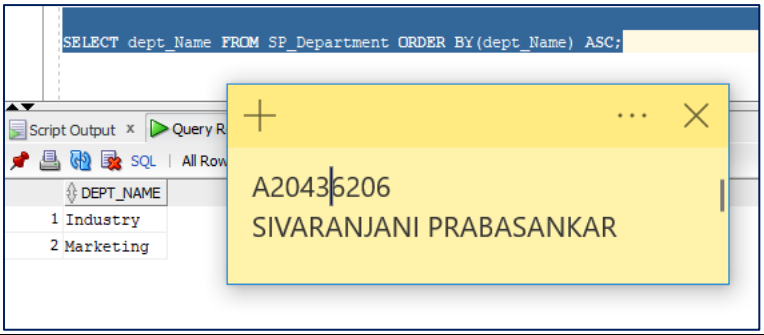
{ Departments, Employees, Bonus, Salary Grade }

(1) (DDL and DML Operations)

Consider the above table structures. The tables store department and employee information. Then, respond to the following exercises, which are based on these tables. Use SQL statements, when applicable.

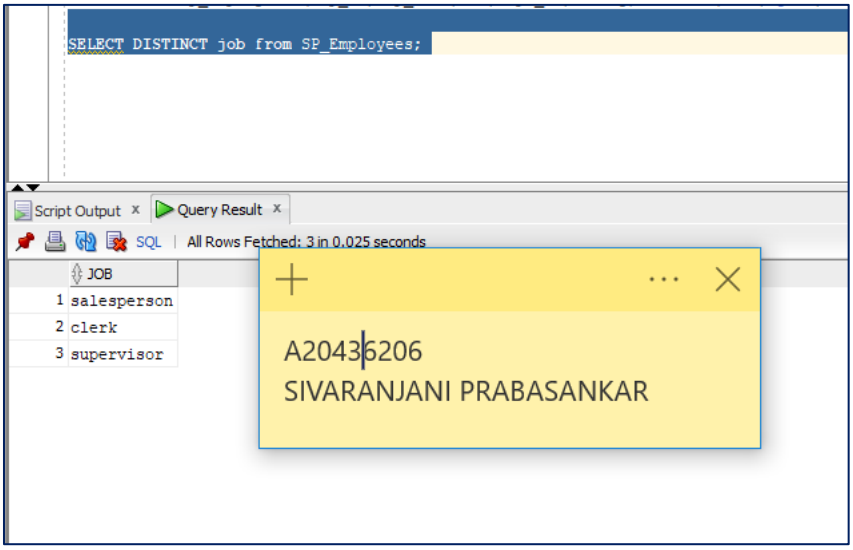
(a) List the names of all departments, in an alphabetical sequence.

```
SELECT dept_Name FROM SP_Department ORDER BY (dept_Name) ASC;
```



(b) Show a list of the different jobs. Eliminate any repeating values.

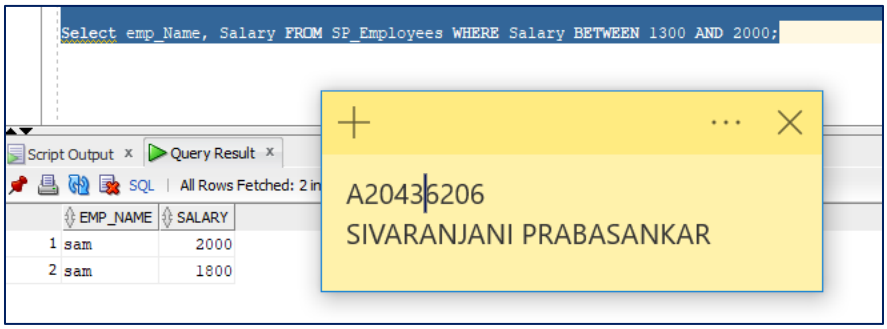
```
SELECT DISTINCT job FROM SP_Employees;
```



Student Name **Sivaranjani Prabasankar** Section _____

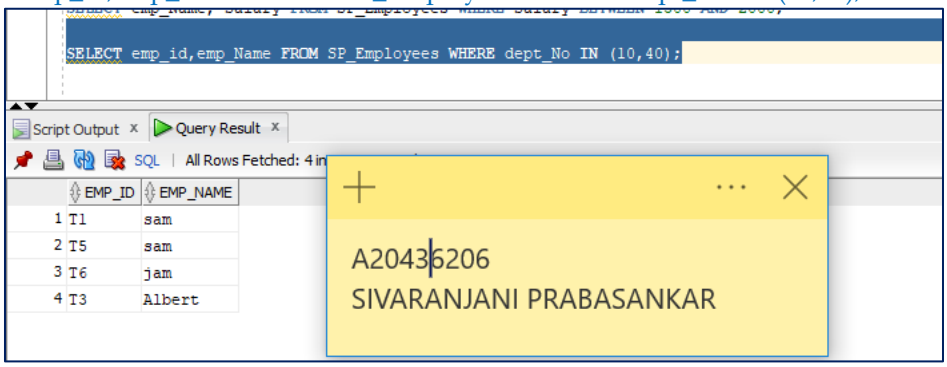
(c) Show all employee names and their salaries when they earn between $ 1,300 and $ 2,000 . Use the `BETWEEN` operator to accomplish this.

```
SELECT emp_Name, Salary FROM SP_Employees WHERE Salary BETWEEN 1300 AND 2000;
```



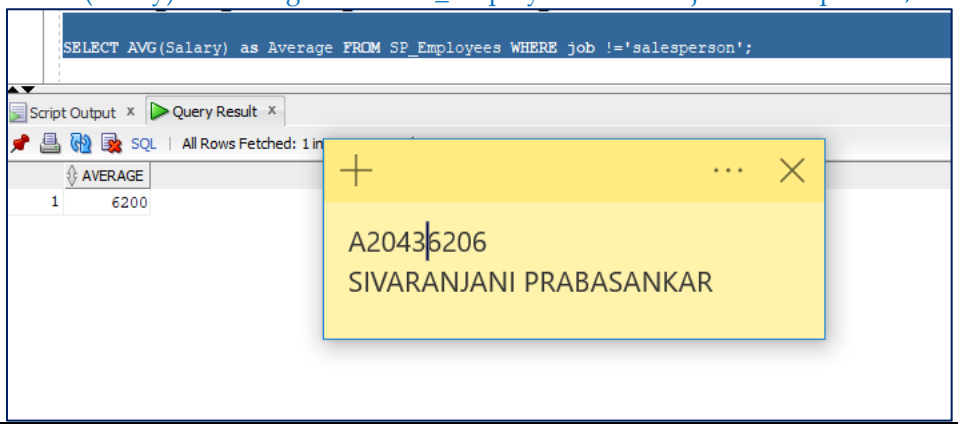
(d) Select all employees that are in department 10 or 40 . Use the `IN` operator to accomplish this task.

```
SELECT emp_id,emp_Name FROM SP_Employees WHERE dept_No IN (10,40);
```



(e) Show the average salary for all the employees, which are not salespersons.

```
SELECT AVG(Salary) as Average FROM SP_Employees WHERE job !='salesperson';
```

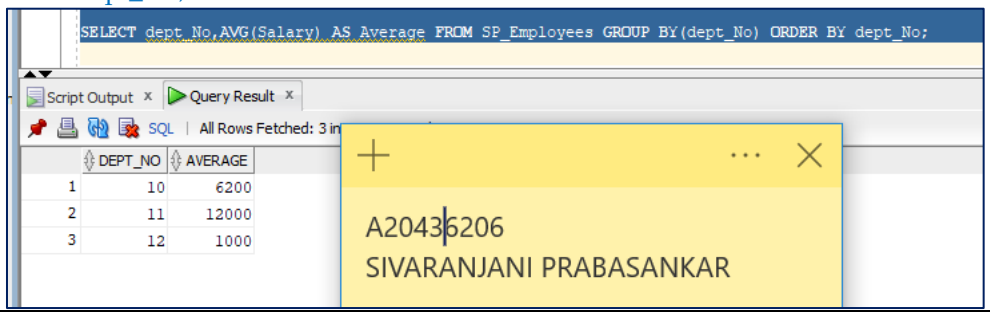


(2) (DDL and DML Operations)

Consider the above table structures. The tables store department and employee information. Then, answer the following exercises, which are based on these tables.

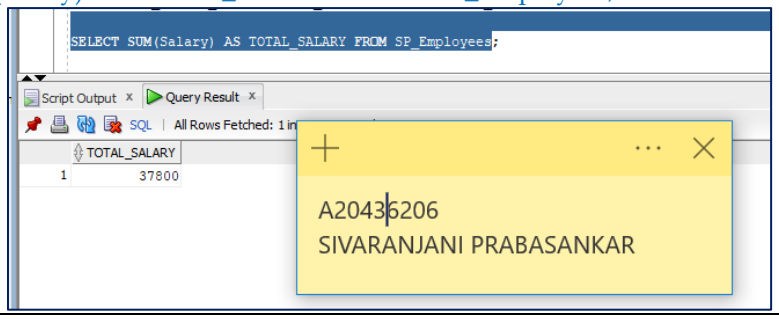
(a) List the average salary for each department.

```
SELECT dept_No,AVG(Salary) AS Average FROM SP_Employees GROUP BY(dept_No)
ORDER BY dept_No;
```



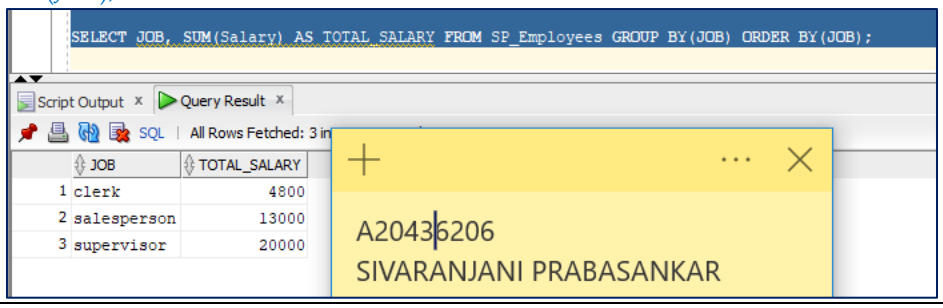
(b) Display the total salary being paid to all employees.

```
SELECT SUM(Salary) AS TOTAL_SALARY FROM SP_Employees;
```



(c) Display the various jobs and the total salary for each job.

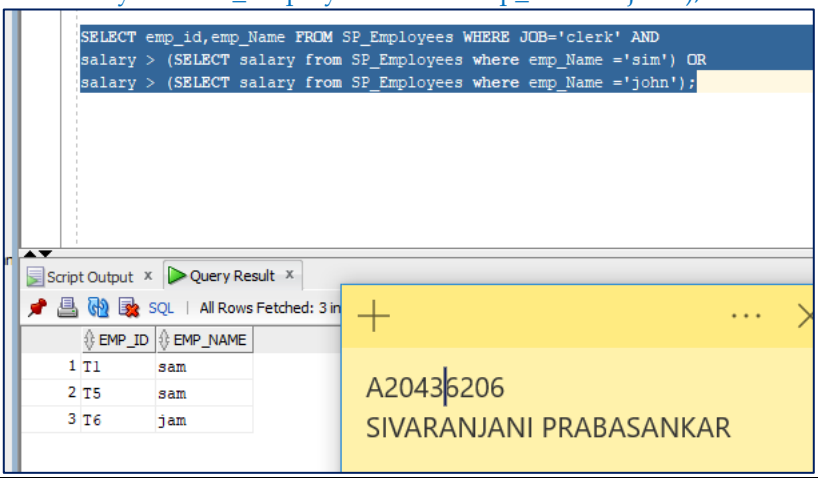
```
SELECT JOB, SUM(Salary) AS TOTAL_SALARY FROM SP_Employees GROUP BY(JOB)
ORDER BY(JOB);
```



Student Name **Sivaranjani Prabasankar** Section _____

(d) Display the names of clerks who earn a salary more than that of Sims or a salary less than that of John.

```
SELECT emp_id,emp_Name FROM SP_Employees WHERE JOB='clerk' AND
salary > (SELECT salary from SP_Employees where emp_Name ='sim') OR
salary > (SELECT salary from SP_Employees where emp_Name ='john');
```



(e) Display the names of employees whose name is exactly six characters in length.

```
SELECT emp_Name FROM SP_Employees WHERE LENGTH(emp_name) = 6 ;
```