Name: Zewei Chen
Date: 5/8/2017
CS 3200 Assignment 1

Problem 1:

DDL is data definition language which is a database schema specified by a set of definitions. It is used to define a schema or to modify an existing one such as describe and name entities, attributes, and relationships required for the application, plus integrity and security, and it cannot be used to manipulate data. DDL statements is a set of tables stored in system catalog which integrates the metadata. In the three level architecture, each level has different DDL, such as external schema, conceptual schema, and internal schema.

a. CREATE: to create new objects.
b. ALTER: to modify the definition of existing objects.
c. DROP: to move entities.
d. TRUNCATE TABLE: to remove rows.

DML is data manipulation language, and it is used to read and update the database. For internal level, we should define complex procedure which allows easy data access. However, at the higher levels, we should focus on user interaction. DML involves data retrieval is called query language. Procedural DML allows user to tell system exactly how to manipulate data. Non-procedural DML allows user to state what data is needed rather than how it is to be retrieved.

a. SELECT: to retrieve data in the database.
b. INSERT: to add new data.
c. UPDATE: to modify existing data.
d. DELETE: to remove data from the database.

Problem 2:

Data Security is about preventing unexpected actions happen on the database, such as data leaked, data lost, etc. Using both software and hardware to encrypt the data, have backup tools, data masking, and data erasure to keep data safe.

Data Integrity is about data accuracy and data reliability. It’s for preventing nonsense data or invalid data which does not match its definition. Data integrity mainly has 4 categories:

a. Entity Integrity: The primary key should be unique and minimal value, and it cannot be any empty or NULL.
b. Domain Integrity: The constrains of attributes. Check and prevent data does not match the rules.
c. Referential Integrity: All values in a column of a table must have their corresponding values in another relational table.
d. User-defined Integrity: Rules defined by a user.

Problem 3:

Data-base approach, that data is embedded in the application program, and no longer need to retrieve the file before getting the data:

a. DDL: the data definition language helps to create the schema of a database.
b. DML: the data manipulation language helps to read and write data.
c. Controlled access to a database include:
    i. Security system: protect data from leaked, lost, and modified unexpectedly
    ii. Integrity system: prevent invalid data
    iii. Concurrency control system: prevent error when multiple users are using the database
    iv. Recovery control system: can recover lost or modified data
    v. User-accessible catalog: allow user to access data which can be accessed under this user's privilege.

File-based approach has limitations:

a. Separation and isolation of data: each file is holding its own data. When a user needs the data, he/she may get redundant data from different files or ignore some useful data in other files.
b. Duplication of data: same data may stay in different files. Space wasting.
c. Data dependence: file structure is defined in the program code.
d. Incompatible file formats: programs are written in different languages and different version of languages, so it's possible to become incompatible, when the version of code is out of date or not supported.
e. Fixed queries/ proliferation: functions are associated with programs, and new functions will require new programs.

Problem 4:

“Data Administrator is responsible for the management of the data resource, including database planning, development and maintenance of standards, policies and procedures, and conceptual/logical database design.” Operations are check the data and schema’s correctness.

“The Database Administrator is responsible for the physical realization of the database, including physical database design and implementation, security and integrity control, maintenance of the operational system, and ensuring satisfactory performance of the applications for users” Operations are such as backup, day-to-day operations, create user, etc.

Problem 5:

a. Relational Data Model: “is based on the concept of mathematical relations, data and relationships are represented as tables, each of which has a number of columns with a unique name. The relational data model requires only that the database be perceived by the user as tables, However, this perception applies only to the logical structure of the database, the external and conceptual levels of the ANSI-SPARC architecture. It does not apply to the physical structure of the database, which can be implemented using a variety of storage structures.”
b. Network Data Model: “In the network model, data is represented as collections of records, and relationships are represented by sets. Compared with the relational model, relationships are explicitly modeled by the sets, which become pointers in the implementation. The records are organized as generalized graph structures with records appearing as nodes (also called segments) and sets as edges in the graph.”
c. Hierarchical Data Model: data is also represented as collections of sets. However, in this model, each node has only a parent. The shape of hierarchical model is a tree shape.

Problem 6:

The relational database:

a. Avoid separation and isolation of data: instead of each file is holding its own data, the data has relations linked to each other in the relational database.
b. Avoid duplication of data: data only stored once in the relational database, and it's spacing friendly.
c. Avoid data dependence: file structure is defined in the program code, but the database isn't
d. Avoid incompatible file formats: programs are written in different languages and different version of languages, so it's possible to become incompatible, when the version of code is out of date or not supported, but the relational database handles the data only.
e. Avoid fixed queries/ proliferation: functions are associated with programs, and new functions will require new programs. The relational database isn't.

Problem 7:

Concurrency Control is the tool that DBMS using to prevent interference occur when multiple users are accessing the database. The concurrency control is important because DBMS enable many users access shared data concurrently, when more than one of them are making changes, this process will interfere other users which will cause inconsistency.

Problem 8:

"A transaction is a series of actions, carried out by a single user or application program, which accesses or changes the contents of the database." Any unexpected actions could cause transaction error, such as system crashed, or some bad actions create inconsistency for the database. An example transaction can be a person A, goes to an ATM, deposit $100, then withdraw $10.

Problem 9:

Client-Server Architecture: The client- server architecture refers to software components interact to form a system. Client-Server architecture has two different types. Client which requires some resource, and the server will provide the resource.

The traditional two-tier architecture provides the basic separation of different application components. Tier 1 is responsible for the data presentation to the client, which handles the UI actions the main business and data application logic, and tier 2 is for supplying data services to the client, which provided limited business application logic, such as user information verification. The two-tier architecture has advantages like:

a. Enables wider access to existing database.
b. Increased performance (client and server are in different computers)
c. Hardware costs may be reduced (only needs hardware which is able to support the server running)
d. Communication costs are reduced (less data being sent across the network)
e. Increased consistency (the server can handle integrity checks, so all the validation happens in one place instead of each application checks its own)
f. It naturally maps on to open systems architecture.

The three-tier architecture: first tier is the user interface layer on end-user's computer. The second tier is business logic and data processing layer runs on server called application server. The third tier has the DBMS called the database server. The advantages are:

a. The need for less expensive hardware because the client is thin.
b. Application maintenance is centralized, since it's having the application server which take over the responsibilities of end-user's applications.
c. The modularity makes it easier to modify and replace without affecting the other tiers.
d. Load balancing is easier with the separation of the core business logic from the database functions.

Teleprocessing: Teleprocessing is the traditional multi-user architecture. There is only one computer with a single CPU connected to a number of dumb (incapable of functioning on their own) terminals. Messages sent through the communication control subsystem to the user's application program, and messages get routed back in the same but reverse order. The drawback of this architecture is, the central computer must run its own application programs, the database, and help to some work of the terminals, which cause its bad performance.

File-Server Architecture: The File-Server architecture distributed about the network, typically a LAN. The file server holds all the files and the database management system. However, the application and database management system are running on each workstation. In this case, the file server is acting as a shared hard disk. The DBMS of each workstation sends a request to the file server, and the server will return a file contains all the data. In this architecture, it's very easy to cause a network traffic, since files considerably large, and some data in the file may not be necessary, which will cause extra efforts to filer them out. Also, when multiple users are making changes to the same file, there is possibility of concurrency, inconsistency, and integrity issue.

Problem 10:

"The TP monitor is a program that controls data transfer between clients and servers in order to provide a consistent environment, particularly for online transaction processing."

Advantages:

a. Transaction routing: the TP monitor increase scalability by directing transactions to specific DBMSs.
b. Managing distributed transactions
c. Load balancing: The TP monitor balances client requests across multiple DBMSs on one or more computer.
d. Funneling: allow a large number of users to access with a much smaller number of connections.
e. Increased reliability: to maintain the consistency of the database.