Assignment1 (Chapter 1 -3 Connolly & Begg) Each question is worth 10 points each

1. Discuss the differences between DDL and DML. What operations would you typically expect to be available in each language?

   DDL is Data definition language. Permits specification of data types, structures and any data constraints. All specifications are stored in the database. It allows users to define the database and specify data types, structures and constraints on the data. Operations can be CREATE TABLE, CREATE INDEX. ALTER, and DROP

   DML Is Data manipulation language. General enquiry facility (query language) of the data. It is used for selecting, inserting, deleting and updating data in a database. Therefore, operations can be SELECT, INSERT, DELETE or UPDATE.

2. Describe the difference between data security and data integrity.

   Data security is protecting data in database from destructive forces and from the unwanted actions of unauthorized users. Without suitable security measures, integration makes the data more vulnerable than file-based systems. However, integration allows the DBA to define database security, and the DBMS to enforce it. This security may take the form of user names and passwords to identify people authorized to use the database. The access that an authorized user is allowed on the data may be restricted by the operation type (retrieval, insert, update, delete).

   Data integrity refers to maintaining and assuring the accuracy and consistency of data in database, and is a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves data. It refers to the validity and consistency of stored data. Integrity is usually expressed in terms of constraints, which are consistency rules that the database is not permitted to violate. Constraints may apply to data items within a single record or to relationships between records.

3. Describe the main characteristics of the database approach and contrast it with the file-based approach.

   The file-based systems have no database, so each program defines and manages its own data. It has some limitations that data of each program is separated, isolated and duplicate. Also, it causes problem of data dependence and Fixed queries/proliferation of application programs. On the other hand, file structure is defined in the program code; and they are written in different language, so it takes extra works and time to manage data of the system or life-cycle (Incompatible file formats and). According to the textbook, these limitations can be summarized into two factors:

(1) The definition of the data is embedded in the application programs, rather than being stored separately and independently.
(2) There is no control over the access and manipulation of data beyond that imposed by the application programs.

Therefore, we use database to manage data. A database needs to have characteristics including, security system, integrity system, concurrency control system, recovery control system and user-accessible catalog. With security and integrity system, it helps maintain safe and accurate data. Concurrency controlling can reduce extra works of programs in file-based system and handle the conflict of data. Recovery controlling can avoid losing data, which can also reduce works of programs and improve the data integrity. And user-accessible catalog is used to improve Data security. It provides a way to track user behavior to check whether there is unauthorized action from users. Comparing it with file-based approach, it has some advantages, including control of data redundancy, data consistency, more information from the same amount of data, sharing of data, improved data integrity, improved security and enforcement of standards. However, it also has disadvantages, such as complexity, size, cost of DBMSs, additional hardware costs, cost of conversion, performance and greater impact of a failure.

4. Provide a definition for a data administrator and a database administrator. What types of interactions would these two users of the database have?

Data Administrator (DA) 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. The DA consults with and advises senior managers, ensuring that the direction of database development will ultimately support corporate objectives. Data administrator works on conceptual/logic level who is responsible for defining data elements, data names and their relationship with the database analyst. He must know how to describe what data is stored in database and relationships among the data. However, he shouldn't require changes to external schema or rewrites of application programs.

The Database Administrator (DBA) 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. Database administrator works on internal/physical level who store and manage the information in the database. He is responsible for designing and implementing database, so he knows how the data is stored in the database. A database administrator shouldn't require change to conceptual or external schemas.

Data administrator and database administrator should have some interactions that Data administrator should know from database administrator that how data is stored in the database. The role of the DBA is more technically oriented than the role of the DA, requiring detailed knowledge of the target DBMS and the system environment. In some organizations there is no distinction between these two roles; in others, the importance of the corporate resources is reflected in the allocation of teams of staff dedicated to each of these roles.

5. Name three record-based data models. Discuss the main differences between these data models.
   Three record-based data models are relational data model, network data model and hierarchical data model.

   Relational data model Represents data as a collection of relations. The relational data model is based on the concept of mathematical relations. In the relational model, data and relationships are represented as tables, each of which has a number of columns with a unique name.

   Network model is a database model conceived as a flexible way of representing objects and their relationships. Its distinguishing feature is that the schema, viewed as a graph in which object types are nodes and relationship types are arcs, is not restricted to being a hierarchy or lattice. 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.

   Hierarchical database model is a data model in which the data is organized into a tree-like structure. The data is stored as records which are connected to one another through links. A record is a collection of fields, with each field containing only one value. The hierarchical model is a restricted type of network model. Again, data is represented as collections of records and relationships are represented by sets. However, the hierarchical model allows a node to have only one parent. A hierarchical model can be represented as a tree graph, with records appearing as nodes (also called segments) and sets as edges.

6. What are the advantages of a relational database when compared to the file-based approach to storing data?

   Comparing with file-based approach, relational database has advantage that data is only stored once, so no multiple record changes are needed. It is simple and efficient to delete or modify data. Secondly, Complex queries can be carried out. Developers can use SQL to

do complex modifications. Also, it provides better security and integrity. data change and authority can be uniformly managed in DBMS with login.

Generally, relational database has some advantages, including:

1. Control of data redundancy (traditional file-based systems waste space by storing the same information in more than one file.)
2. Data consistency (By eliminating or controlling redundancy, we reduce the risk of inconsistencies occurring.)
3. More information from the same amount of data (With the integration of the operational data, it may be possible for the organization to derive additional information from the same data.)
4. Sharing of data (Typically, files are owned by the people or departments that use them. On the other hand, the database belongs to the entire organization and can be shared by all authorized users.)
5. Improved data integrity (Constraints may apply to data items within a single record or to relationships between records.)
6. Improved security (The access that an authorized user is allowed on the data may be restricted by the operation type)
7. Enforcement of standards (integration allows the DBA to define and the DBMS to enforce the necessary standards.)
8. Economy of scale (Combining all the organization's operational data into one database and creating a set of applications that work on this one source of data can result in cost savings.)
9. Balance of conflicting requirements (the database is under the control of the DBA)
10. Improved data accessibility and responsiveness (data that crosses departmental boundaries is directly accessible to the end users.)
11. Increased productivity (the DBMS provides many of the standard functions that the programmer would normally have to write in a file based application)
12. Improved maintenance through data independence (a DBMS separates the data descriptions from the applications, thereby making applications immune to changes in the data descriptions.)
13. Increased concurrency (DBMS manage concurrent database access and ensure that concurrency problems cannot occur.)
14. Improved backup and recovery services (DBMSs provide facilities to minimize the amount of processing that is lost following a failure.)

However, it also has disadvantages, such as complexity, size, cost of DBMSs, additional hardware costs, cost of conversion, performance and greater impact of a failure.

7. What is concurrency control and why does a DBMS need a concurrency control facility?

Concurrency control is the activity of coordinating concurrent accesses to a data- base in a multiuser database management system (DBMS), which concurrency control. It ensures that database transactions are performed concurrently without violating the data

integrity of the respective databases. Thus concurrency control is an essential element for correctness in any system where two database transactions or more, executed with time overlap, can access the same data. In some file-based systems, if two or more users are allowed to access the same file simultaneously, it is possible that the accesses will interfere with each other, resulting in loss of information or even loss of integrity. Many DBMSs manage concurrent database access and ensure that such problems cannot occur. Concurrent access is relatively easy if all users are only reading data as there is no way that they can interfere with one another. However, when two or more users are accessing the database simultaneously and at least one of them is updating data, there may be interference that can result in inconsistencies.

8. What is a transaction? Give an example of a transaction.

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. For example, some simple transactions for the *Dream Home* case study might be to add a new member of staff to the database, to update the salary of a member of staff, or to delete a property from the register. If the transaction fails during execution, perhaps because of a computer crash, the database will be in an inconsistent state: some changes will have been made and others will not. Consequently, the changes that have been made will have to be undone to return the database to a consistent state again. A transaction symbolizes a unit of work performed within a database management system (or similar system) against a database, and treated in a coherent and reliable way independent of other transactions. A transaction generally represents any change in database, such as BEGIN and END in SQL. The process between them is a typical transaction.

9. What is meant by the term 'client-server architecture' and what are the advantages of this approach? Compare the client-server architecture with two other architectures.

There are three DBMS architectures: teleprocessing, file-server and client-server.

Client-server architecture is a computing model in which the server hosts, delivers and manages most of the resources and services to be consumed by the client. This type of architecture has one or more client computers connected to a central server over a network or Internet connection. This system shares computing resources. Comparing it with other two architectures, client-server architecture has advantages, including wider access to databases, increased performance, possible reduction in hardware costs, reduction in communication costs and increased consistency.

10. What is a Transaction Processing Monitor? What advantages does a TP Monitor bring to an OLTP environment?

Transaction Processing Monitor is a program that controls data transfer between clients and servers in order to provide a consistent environment, particularly for online transaction

processing (OLTP). Complex applications are often built on top of several resource managers (such as DBMSs, operating systems, user interfaces, and messaging software). A Transaction Processing Monitor, or TP Monitor, is a middleware component that provides access to the services of a number of resource managers and provides a uniform interface for programmers who are developing transactional software. A TP Monitor forms the middle tie of a three-tier architecture. The advantages of this include increased system robustness as well as throughput.

TP Monitors provide significant advantages, including:

1. Transaction routing: The TP Monitor can increase scalability by directing transactions to specific DBMSs.
2. Managing distributed transactions: The TP Monitor can manage transactions that require access to data held in multiple, possibly heterogeneous, DBMSs.
3. Load balancing: The TP Monitor can balance client requests across multiple DBMSs on one or more computers by directing client service calls to the least loaded server. In addition, it can dynamically bring in additional DBMSs as required to provide the necessary performance.
4. Funneling: In environments with a large number of users, it may sometimes be difficult for all users to be logged on simultaneously to the DBMS. In many cases, we would find that users generally do not need continuous access to the DBMS. Instead of each user connecting to the DBMS, the TP Monitor can establish connections with the DBMSs as and when required, and can funnel user requests through these connections. This allows a larger number of users to access the available DBMSs with a potentially much smaller number of connections, which in turn would mean less resource usage.
5. Increased reliability: The TP Monitor acts as a transaction manager, performing the necessary actions to maintain the consistency of the database, with the DBMS acting as a resource manager. If the DBMS fails, the TP Monitor may be able to resubmit the transaction to another DBMS or can hold the transaction until the DBMS becomes available again.