CS460 Spring, 2016

Homework #1
TA: Shuo Yang

---

1. Database approach is different with the file-based approach in three ways:

   (a) Unlike file-based approach, where the structure of data is embedded in the application programs, database approach decouples the structure of data from the application programs, thus providing data-program independence. In database approach, the structure of data is held in system catalog (or data dictionary) and programs can share data.

   (b) Database approach also reduces unnecessary data redundancy which is often the case in the file-base approach.

2.

	Point of view	Functions	Relationship with other levels
External level	User's view of data	Let users access data that is customized to their needs	Interfaces with conceptual level to provide logical independence
Conceptual level	Community view of database, seen by DBA	Describes the logical structure of the entire database, but is independent of any storage considerations	Interfaces with external level to support external views and interfaces with internal level to provide physical data independence
Internal level	Physical representation of the database	Describes how data is stored in the database	Interfaces with conceptual level to provide physical data independence

3. The three-tier client-server architecture maps naturally to the Web with a web browser acting as a thin client, an application server handling user requests and a database server managing data access. This architecture has better modularity (easier to modify or replace one tier without affecting the others), better load balancing (with the separation of core business logic from the database functions) compared to the two-tier client-server architecture for traditional DBMSs.

4. MySQL supports client-server architecture because it was designed to operate in a networking environment where multiple users can simultaneously connect to the database to make requests. In MySQL package, *mysqld* is the server program that manages database and handles client requests, *mysql* is the client program that connects to *mysqld* to access the database.

5. Mathematical relations are based on set theory.
   Let $D_1, D_2, \cdots D_n$ be n sets(domains). Their Cartesian product is defined as:

$$D_1 \times D_2 \times \cdots \times D_n = \{(d_1, d_2, \cdots, d_n) | d_1 \in D_1, d_2 \in D_2, \cdots d_n \in D_n\} \tag{1}$$

   A mathematical relation on the n sets is defined as any set of n-tuples from this Cartesian product.

   Now let $A_1, A_2, \cdots, A_n$ be attributes with values drawing from the domains $D_1, D_2, \cdots D_n$. Then the set $\{A_1 : D_1, A_2 : D_2, \cdots, A_n : D_n\}$ is a relation schema. In relational model, a relation defined by the relation schema is a set of mapping from the attribute names to their corresponding domains. Thus, a relation R is a set of n-tuples:

$$R = \{(A_1 : d_1, A_2 : d_2, \cdots, A_n : d_n) | d_1 \in D_1, d_2 \in D_2, \cdots d_n \in D_n\} \tag{2}$$

   In this way, we can think of a relation in the relational model as any subset of the Cartesian product of the domains of the attributes.

6. The primary key is the candidate key that is selected to identify tuples uniquely within a relation. A foreign key is an attribute or set of attributes within one relation that matches the candidate key of some (possibly the same) relation.

   For example, consider the following relations:

   Student(UANetID, firstName, lastName, CSID)
   Grade(CSID, Assignment#, score)
   Employee(SSN, firstName, lastName, superSSN)

   where *CSID* in the **Student** relation is a candidate key. Thus *CSID* in the **Grade** table is a foreign key refers to the candidate key CSID in the **Student** relation. *superSSN*(supervisor's SSN) is a foreign key refers to the primary key *SSN* in the same relation.

7. A relationship type is a set of associations between one or more participating entity types.

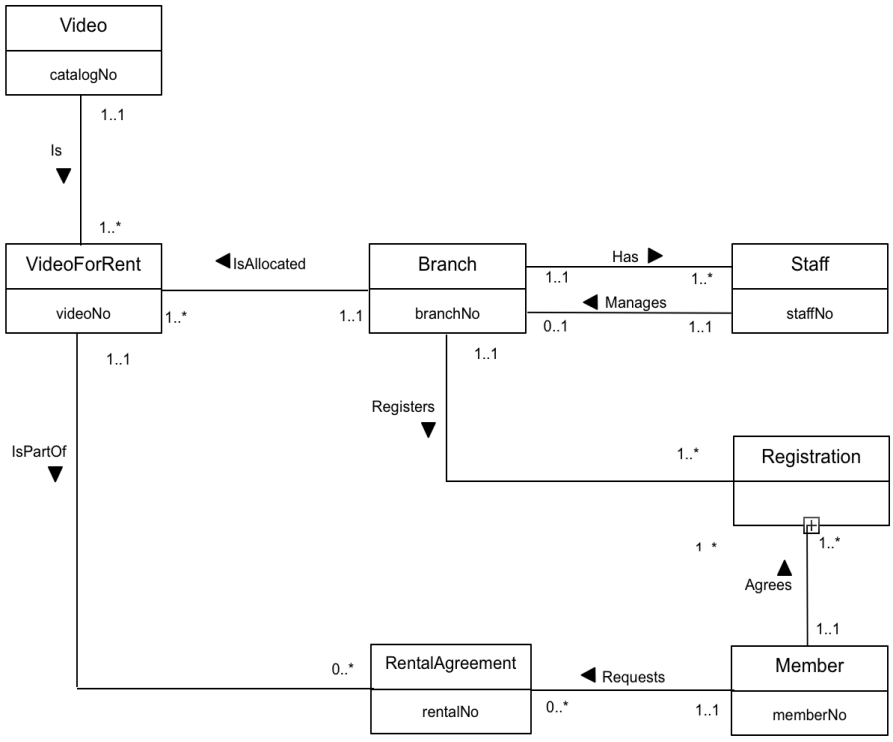
   (a) Unary relationship (recursive relationship):
   Employee *supervises* employee

   (b) Binary relationship:
   Doctor *works for* Hospital

   (c) Ternary relationship:
   Client *sees* Doctor *at* Hospital

8. The relationship **Newspaper** *Advertises* **PropertyForRent** consists of two attributes: dateAdvert (representing the date the advert took place) and cost (representing the cost of the advert).

9. ER diagram is shown below:



10. Consider Manager as a specialization of the Staff entity. This would move the Manages relationship from Staff to the Manager subclass. However, the attributes for both entities would be the same and there would, therefore, seem to be no obvious advantage to introducing the Manager specialization.

11. (a) The system fails if either one of the four hard drives fails. Put it in another way, the system works only if all the four hard drives do not fail. The probability of the system not fail is:

$$P_{nf} = (1 - 0.015)^2 \times (1 - 0.03)^2 \tag{3}$$
$$\approx 0.913 \tag{4}$$
$$= 91.3\% \tag{5}$$

Therefore the probability of failure of the system is:

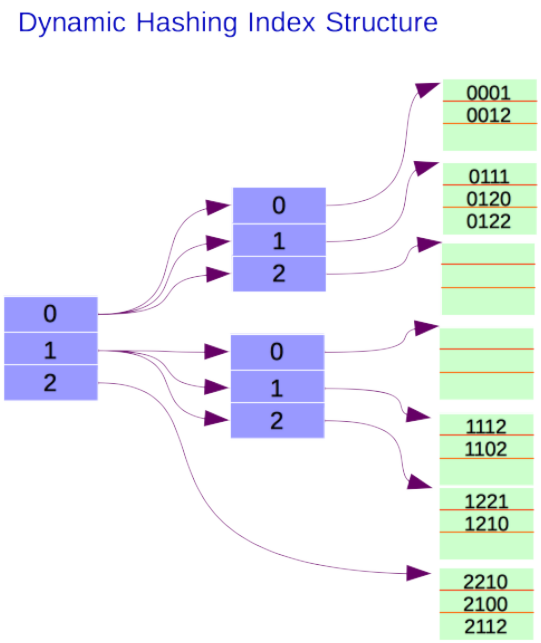
$$P_f = 1 - P_{nf} \tag{6}$$
$$= 0.087 \tag{7}$$
$$= 8.7\% \tag{8}$$

(b) In order to exceed a 50% failure rate, we need to satisfy the following condition:

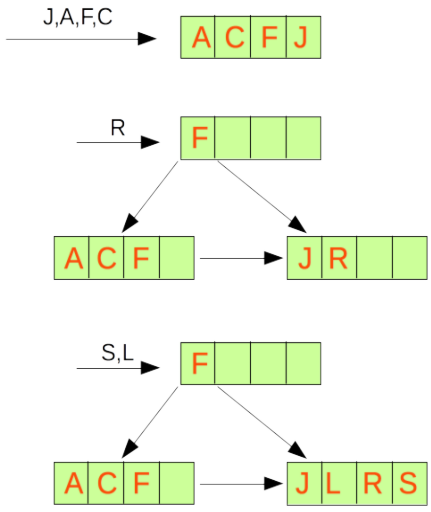
$$P_f = 1 - P_{nf} \tag{9}$$
$$= 1 - ((1 - 0.015)^n \times (1 - 0.03)^n) \tag{10}$$
$$> 50\% \tag{11}$$

When $n = 15$, $P_f = 49.5\%$. When $n = 16$, $P_f = 51.8\%$. Thus there must be at least 32 hard drives the RAID system would have to exceed a 50% failure rate.

12. Dynamic hashing index structure is shown below:



13. (a) Insertion process of B^+ tree is shown below:



(b) Deletion process of B^+ tree is shown below: