Lu Ye

Cs460-Homework 1

February 22, 2017

2.4 Discuss the function and importance of conceptual modeling.

Conceptual modeling is the process of constructing a detailed architecture for a database, which is independent of implementation details such as the target DBMS, application programs, programming languages etc. Its function is mainly to represent data in an understandable way that enhance an individuals understanding of database. Also, it provides a complete and accurate representation of the data requirement of the enterprise. It is important because some information about the enterprise will be missing or incorrectly represented if the function is lost. Thus, it is simple the way we implement one or more of the external views.

3.2 Compare and contrast the two-tier client–server architecture for traditional DBMSs with the three-tier client–server architecture. Why is the latter architecture more appropriate for the Web?

Similarly, they are both client server architecture, which means it refers to the way in which software components interact to from a system. However, the two-tier client-server architecture (abbrev. 2TA as below) is makes use of two computer sets and three-tier client-server architecture (abbrev. 3TA as below) is makes use of three computer sets. Also, 2TA provides two layers including the presentation layer (the client) and the data services layer (the server supply data services to the client). On the contrary, 3TA proposed three layers including the user interface layer, application server layer (aka. business logic and data processing layer) and the database server layer (stores the data). What's more, the mainly advantage of 2TA is easy to modify and maintain. Based on the 2TA, 3TA got developed which has high performance, scalability and improved data integrity as well as improved security. The mainly disadvantage of 2TA is high cost, but for 3TA is its complexity.

The reason why three-tier client-server architecture more appropriate for the web is 1) the cost of hardware is lower. 2) Load balancing is easier with the separation of the core business logic from the database functions. 3) Application maintenance is centralized. 4) The added modularity makes it is easier to modify. 5) It is naturally to the web environment.

3.18(a) Examine the documentation sets of Microsoft SQL Server, Oracle, and IBM's DB2 system to identify their support for the following: (but use the Postgres DBMS (www.postgresql.org) as your subject)

(a) client–server architecture

PostgreSQl uses a client-server architecture and a postgreSQl session consistes of a server process and the user's client application. A server process manages the database files and it accepts connection to the database from client application. It is also performs database action on behalf of the clients. What's more the user's client application is usually performing database operations. A client can be a text-oriented tool, a graphical application, a web server or a specialized database maintenance tool. These client applications are usually supplied with the PostgreSQL and developed by users. Be more specific, the client and the server can be on different hosts, which means they communicate with each other through network connection.

4. What is the difference between a database file and that file's schema? And, what is the schema of the hash bucket file of a dynamic hashing index?

The database can be seen as a main container, it is a collection of logically related data. Database file contains data and log file, and all the schemas within it. However, the schemas are like folders within a database, which is a description of the database structure. What is more, schema represents the way of categorizing the objects in database and provides a logical view of the entire database. From the file's schema, we can generally know how the data is organized in the file and what the relations are within the file.

Based on the definition of dynamic hashing, we can generally know that dynamic hashing provides a mechanism in which data buckets can be added or removed dynamically on demand. And the schema of the hash bucket file of a dynamic hashing index can be included a collection of buckets. Buckets as the organization of the hash bucket file of a dynamic hashing index, it is considered a unit of storage and it typically stores one complete disk block, which stores one or more records. Then look at the function of the bucket file, in the hash bucket file, it should has several empty buckets at the beginning. As the buckets are filled based on the record being inserted, the new buckets will be added to the end of the file. Also, it performs hash function, which is a performance from search key to bucket address.

4.8 Identify the foreign keys in this schema. Explain how the entity and referential integrity rules apply to these relations.

A foreign key is an attribute, or set of attributes within one relation that is the candidate key of another relation. For this table, room is related to hotel according to the attribute hotelNo. Booking is related to hotel according to the attribute hotelNo, it is related to guest according to the attribute guestNo and it is also related to room according to roomNo. (The underlying words are identified as the foreign keys.)

Based on the definition, entity integrity is a constraint that states that in a base relation no attribute of a primary key can be null. Referential integrity states that foreign key values must match a candidate key value of some tuple in the home relation or be wholly null. Therefore, the more rules apply to these relations is that hotelNo in the Room should be either a null value or be with an existing value of hotel in Hotel relation. In this case, hotelNo should be with an existing value.

Similarly for the foreigner key hotelNo in the Booking, so a null value for this case is not accepted because hotelNo is also one of primary key. It is same for guestNo in this case. And for the roomNo in the Booking, the null value is accepted because it is not part of primary key in the Booking.

12.8 Describe how strong and weak entity types differ and provide an example of each.

Strong entity type is an entity type that is not existence-dependent on some other entity type. And weak entity is an entity type that is existence-dependent on some other entity type. For example, “Question” is an example of strong entity because it is not rely on other entity type, which means it exited by itself. And “Answer” is an example of weak entity because the answer exits only if there is a question, which means it exits that is dependent on the entity type of “Question.”

12.11 You are required to create an ER model of the data requirements for a company that specializes in IT training. The company has 30 instructors and can

handle up to 100 trainees per training session. The company offers five advanced technology courses, each of which is taught by a teaching team of two or more instructors. Each instructor is assigned to a maximum of two teaching teams or may be assigned to do research. Each trainee undertakes one advanced technology course per training session.

(a) Identify the main entity types for the company.

The main entity types for this company including instructors, trainees, training session, advanced technology courses, teaching teams and research.

(b) Identify the main relationship types and specify the multiplicity for each relationship. State any assumptions that you make about the data.

Each trainee undertakes one advanced technology course.

Each training session can handle many (100) trainees.

This company provides many (5) courses.

Each course has one training session.

Each courses is taught by a teaching team of two or more instructors (One teaching team).

One teaching team can teach many courses.

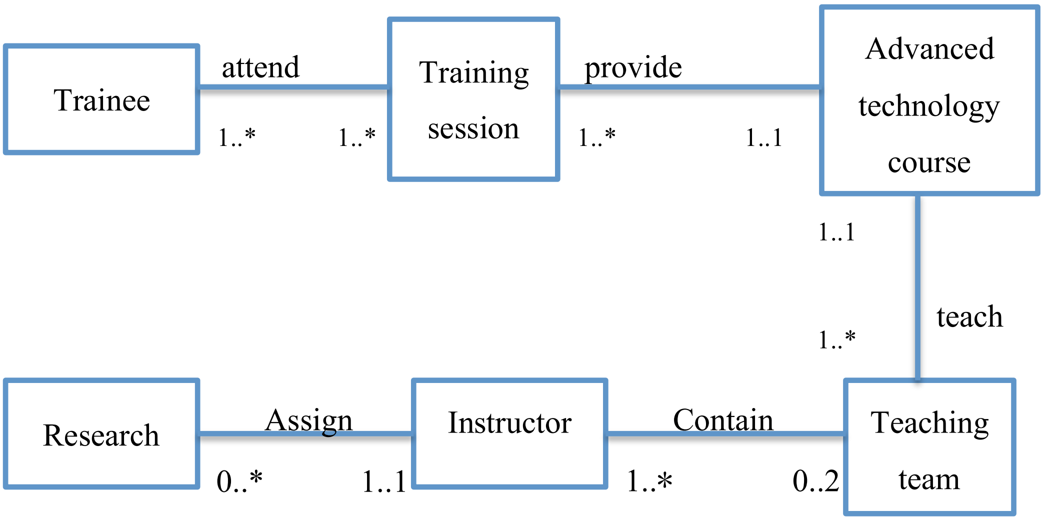
An instructor is assigned to a maximum of two teaching teams.

Each teaching team contains many instructors.

Each instructor may be assigned to do research.

One research may be assigned to one instructor.

(c) Using your answers for (a) and (b), draw a single ER model to represent the data requirements for the company



8. Probabilities of disk failure. For all parts, assume an AFR probability of pf = 0.025.

(a) What is the probability of failure of a striped 5-disk system?

$$P(D_{nf}) = 1 - p(D_f) = 1 - 0.025 = 0.975$$

$$p_{nf} = p\neg(D_{1f} \cup D_{2f} \cup D_{3f} \cup D_{4f} \cup D_{5f})$$

$$= (0.975)^5$$

$$\approx 0.88109$$

$$\mathrm{p}_f = 1 - 0.88109 \approx 0.11891$$

(b) When your boss hears this, he's appalled. "That's too high! Build another system, just like this one, put it in our branch office in Milwaukee, and use it to mirror this system. That'll reduce the probability of the whole system failing!" Will it or won't it? Explain your answer, showing the math so that even your boss will understand.

Mirror system will be fail when each disk of a striped 5-disk system will fail, which means if 1A disk fail, the mirror system will fail when 1B disk fails.

$$p_f = (p_{1A} \cap p_{1B}) \cup (p_{2A} \cap p_{2B}) \cup (p_{3A} \cap p_{3B}) \cup (p_{4A} \cap p_{4B}) \cup (p_{5A} \cap p_{5B})$$

9. (10 points) We recently covered just four of the standard RAID levels in class. One of the levels we skipped is RAID 2. Using reliable sources, learn the details of RAID 2, and write, in your own words, a description of no more than three paragraphs describing how RAID 2 differs from RAID 3, and why RAID 2 is almost never used.

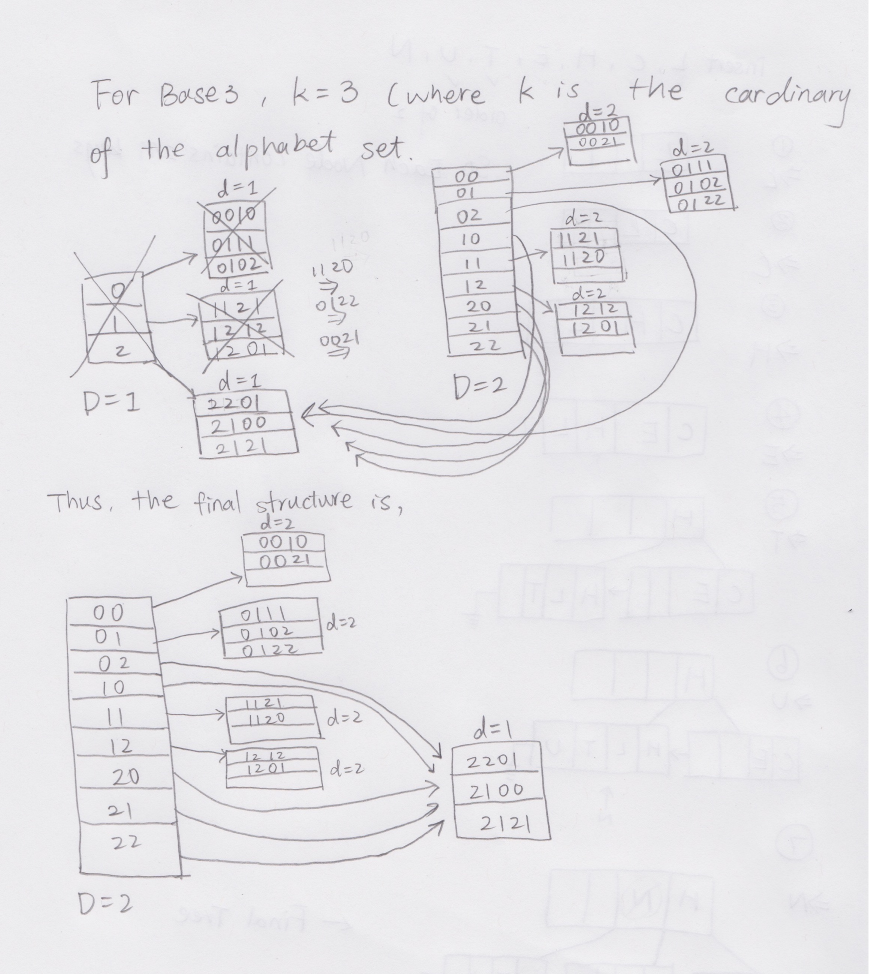
RAID is known as redundant arrays of independent disks, which supports DBMS keeps operating even if one of the hardware components fails. There are many different RAID levels and in particular RAID 2 is known as memory-style error-correcting code. It is using a Hamming code to detect errors in order to stripe data at the bit level.

From the conceptual perspectives, RAID 2 is similar to RAID 3 that they are both storing parity data on a single disk, and the unit of the striping is a single bit. However, different from RAID 3, it is not only used to detect errors but also used to correct information. However, it has no advantage over RAID 3 such that we don't use it any more.

For more information, RAID 2 cannot work for multiple requests simultaneously because the disks are synchronized by controller, which is required to spin at the same angular orientation. This is lead to require super high data transfer rates. Thus, compared to all other disks drives, the complexity of external and the lack of efficiency make RAID 2 be not used anymore.

10. (10 points) In Program #2, you dynamically-hashed using decimal digits. For this exercise, assume that our keys are in Base 3 instead of Base 10. Also assume that buckets are disk blocks that can hold at most three keys each. Build an extendible hashing (not dynamic hashing!) index structure using the keys listed below, and draw the final structure. Be sure to include all global and local depths.

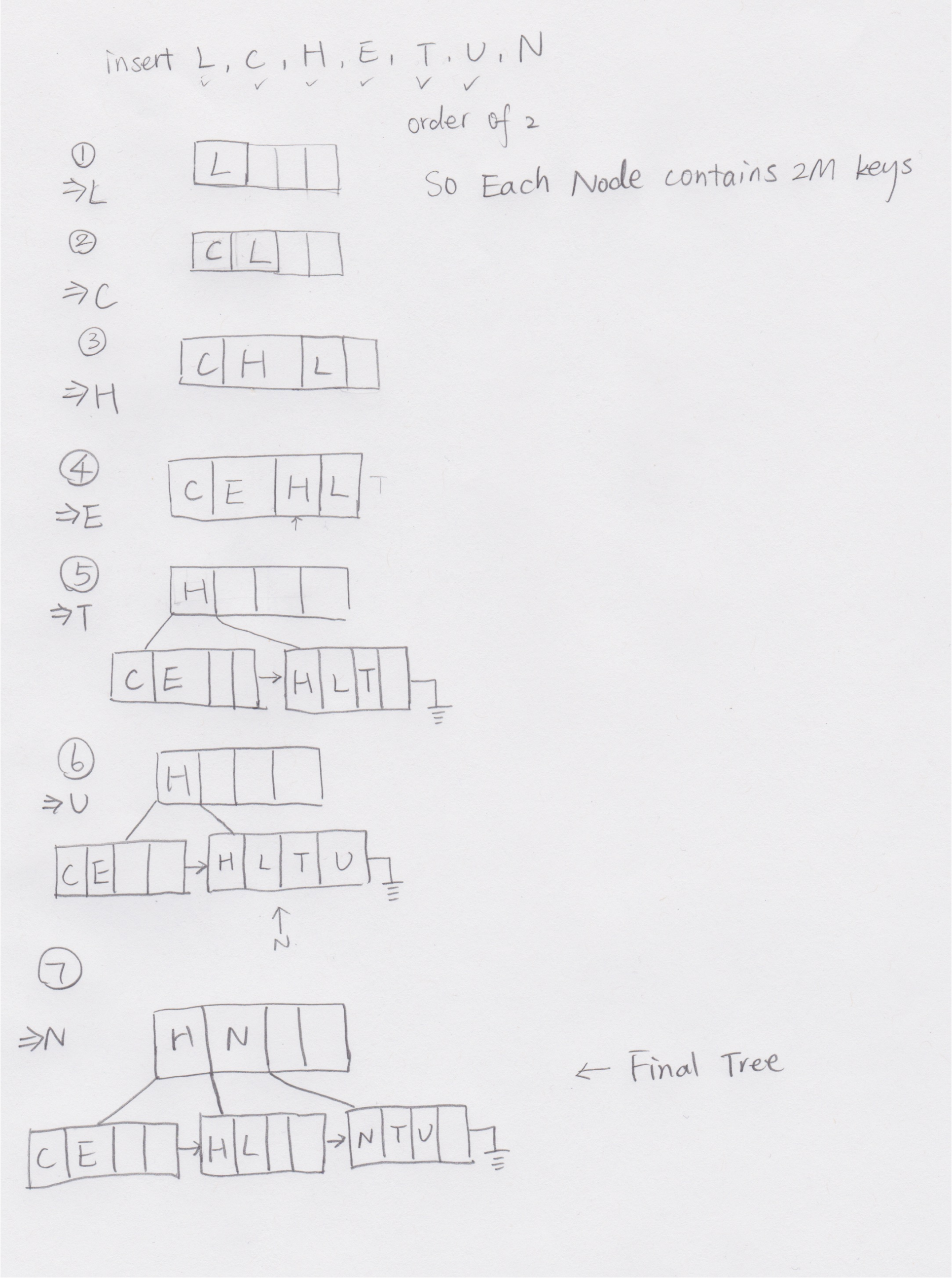
1121, 2201, 0010, 0111, 1212, 0102, 1201, 2100, 1120, 0122, 0021, 2121



For Base3, k=3 (where k is the cardinary of the alphabet set.

Thus, the final structure is,

11. (15 points) Assuming a B+-tree of Order 2 (using our in-class (Comer's) definition), insert the values L, C, H, E, T, U, and N, in the order presented. Show the tree after each insertion that causes the tree to grow by a level, and show the final tree.



insert L, C, H, E, T, U, N

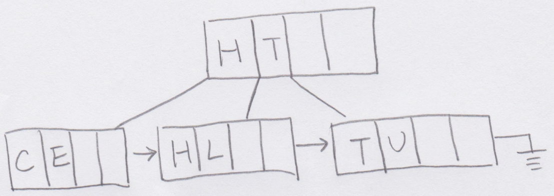
order of 2

So Each Node contains 2M keys

← Final Tree

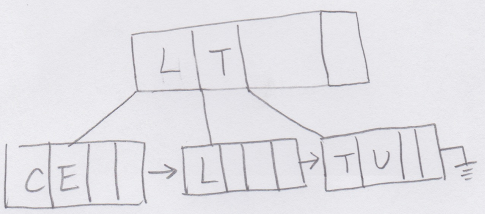
12. (10 points) Delete, from your final tree from the last problem, the following keys, and show the final tree: N and H (in that order). Again, you may show intermediate trees if you so desire

delete N

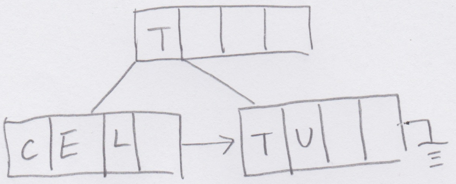


⇐ Final Tree after deleting N

delete H



merging Node



⇐ Final Tree after deleting H