Q2 (a) With the help of examples, explain the following terms briefly:
entity set, one-to-many relationship, participation constraint, weak entity set.

Answer

Entity set: A collection of similar entities such as all of the toys in the toy department

One-to-many relationship: A key constraint that indicates that one entity can be associated with many of other entity. An example of a one-to-many relationship is when an employee can work for only one department, and a department can have many employees.

Participation constraint: A participation constraint determines whether relationships must involve certain entities. An example is if every department entity has a manager entity. Participation constraints can either be total or partial. A total participation constraint says that every department has a manager. A partial participation constraint says that every employee does not have to be a manager.

Weak entity set: An entity that cannot be identified uniquely without considering some primary key attributes of another identifying owner entity. An example is including dependent information for employees for insurance purposes.

Q2 (b) Which of the following plays an important role in *representing* information about the real world in a database? Explain briefly.
(i) The data definition language.
(ii) The data manipulation language.

Answer

Let us discuss the choices in turn.

The data definition language is important in representing information because it is used to describe external and logical schemas.

The data manipulation language is used to access and update data; it is not important for representing the data. (Of course, the data manipulation language must be aware of how data is represented, and reflects this in the constructs that it supports.)

Q2 (c) List four significant differences between a file-processing system and a DBMS.

Answer

Some main differences between a database management system and a file-processing system are:

- Both systems contain a collection of data and a set of programs which access that data. A database management system coordinates both the physical and the logical access to the data, whereas a file-processing system coordinates only the physical access.

- A database management system reduces the amount of data duplication by ensuring that a physical piece of data is available to all programs authorized to have access to it, where as data written by one program in a file processing system may not be readable by another program.
- A database management system is designed to allow flexible access to data (i.e., queries), whereas a file-processing system is designed to allow predetermined access to data (i.e., compiled programs).
- A database management system is designed to coordinate multiple users accessing the same data at the same time. A file-processing system is usually designed to allow one or more programs to access different data files at the same time. In a file-processing system, a file can be accessed by two programs concurrently only if both programs have read only access to the file.

Q3 (a) Using suitable example, explain the following relational operations: SELECT, JOIN, DIVISION

Answer Page Number 175, 188 of Textbook

Q3 (b) Explain the entity integrity and referential integrity constraints.

Answer Page Number 157-158 of Textbook

Q3 (c) Define foreign key. What is this concept used for?

Answer Page Number 158 of Textbook

Q4 (a) What is a view? How is it different from a table?

Answer Page Number 285 of Textbook

Q5 (a) Explain with the help of examples, the concept of insertion anomalies and deletion anomalies.

Answer Page Number 344 of Textbook

Q6 (a) What is collision in hashing? What are the various ways of collision resolution?

Answer Page Number 491 of Textbook

Q6 (b) What is a B-tree? Describe the structure of B-tree nodes.

Answer Page Number 530 of Textbook

Q7 (b) With the help of a diagram, explain the steps required in processing of a high-level query.

Answer Page Number 552-553 of Textbook

Q8 (a) List the ACID properties. Explain the usefulness of each.

Answer

The ACID properties and the need for each of them are:-

1. Consistency: Execution of a transaction in isolation (that is, with no other transaction executing concurrently) preserves the consistency of the database. this is typically the responsibility of the application programmer who codes the the transaction .
2. Atomicity: either all operations of the transection are reflected properly in the database, or none are clearly lack of atomicity will lead to inconsistency in the database.
3. Isolation: when multiple transaction execute concurrently, it should be the case that for every pair of transaction Ti and Tj, it appears to Ti that either Tj finished execution before Ti started, or Tj started execution after Ti finished. Thus each transaction is unaware of other transactions executing concurrently with it. The user view of a transaction system requires the isolation property. and the property that concurrent schedules take system from one consistent state to another . These requirements are satisfied by ensuring that serializable schedules of individually consistency preserving transaction are allowed.
4. Durability: after transaction complete successfully, the change it has made to the database persists, even if there are system failures.

Q8 (b) During its execution, a transaction passes through several states, until it finally commits or aborts. List all possible sequences of states through which a transaction may pass. Explain why each state transition may occur.

Answer

a. active –partially committed-committed . this is the normal sequence a successful transaction will follow . After executing all its statement it enters the partially committed state .
b. active –partially committed –aborted . after executing the last statement of the transaction it enter the partially committed state. But before enough recovery information is written to disk, a hardware failure may occur destroying the memory contents. In this case the change which to disk, a hardware failure may occur destroying the memory contents. In this case the change which it made to the database are undone, and transaction enters the aborted state.
c. Active – failure –aborted. After the transaction start. if it is discovered at some points that normal execution cannot continue (either due to internet program errors or external errors) , it enters the failed state .it is then rolled back , after which it enters the enters the aborted state.

Q8 (c) What is a cascadeless schedule? Why is cascadelessness of schedules desirable? Are there any circumstances under which it would be desirable to allow non cascadeless schedules? Explain your answer.

Answer

A cascade less schedule is one where , for each pair of transaction Ti and Tj such that Tj reads data item previously written by Ti the commit operation of Ti the commit operation of Ti appears before the read operation of Tj . Cascade less schedules are desirable because the failure of a transaction does not lead to the aborting of any other transaction. Of course this comes at the cost of less concurrency. If failure occure rarely, so that we can pay the price of cascading aborts for the increased concurrency, non cascade less schedules might be desirable.

Q9 (a) Briefly explain recovery techniques based on immediate update.

Answer Page Number 672 of Textbook

Q9 (b) What is the difference between stable storage and disk?

Answer

Stable storage is guaranteed (with very high probability) to survive crashes and media failure .A disk might get corrupted or fail but the stable storage is still expected to retain whatever is stored in it. One of the ways of achieving stable storage is to store the information in a set of disk rather then in a single disk with some information duplicated so that the information is available even if one or two of the disks fail

Q9 (c) What are the roles of the Analysis, Redo, and Undo phases in ARIES?

(i) What is done during Analysis?

(ii) What is done during Redo?

(ii) What is done during Undo?

Answer

The Analysis phase starts with the most recent begin checkpoint record and proceeds forward in the log unit the log record. It determines.

LSN	Log
00	BEGIN CHECKPOINT
10	END_CHECKPOINT
20	UPDATE:TI AND P5
30	UPDATET2 AND P3
40	T2COMMIT
50	T2END
60	UPDATE:T3 AND P3
70	T1 ABORT
	CRASH, RE-START

a. The points in the log at which to start the redo pass. We will assume that the dirty page table and transaction. Tables were empty before the start of the log. Analysis determine s that the last begin checkpoint was LSN00 and start at the corresponding

end checkpoint (LSN10).We will denote transaction table records as (transID, lastLSN)and Dirty page table records as (page ID, recLSN)sets .

b. The dirty pages in the buffer pool at the time of the crash. Redo start at LSN20 (smallest recLSN in DPT).

c. Transaction those were active at the time of the crash which needs to be undone. Undone start at LSN70 (highest lastLSN in TT). The Loser set consists of LSNs70 and 60.LSN70: adds LSN20 to the Loser set .loser set= (60, 20): Undone the change on P5 and adds a CLR indicating this Undo .Loser set= (20).LSN20: Undoes the change on P5 and adds a CLR indicating this undo.

The Redo phase follows analysis and redoes all changes to any page that might have been dirty at time of the crash. The Undo phase follows Redo and undoes the change of all transaction that was active at the time of the crash.

Text Book

Fundamentals of Database Systems, Elmasri, Navathe, Somayajulu, Gupta, Pearson Education, 2006