Babu Banarasi Das
National Institute of Technology and Management

Department of Computer Applications

Question Bank
(Short-to-Medium-Answer Type Questions)

Masters of Computer Applications (MCA) NEW Syllabus
(Affiliated to U. P. Technical University, Lucknow.)

III Semester

MCA313: Database Management System

UNIT I

1. Define the concept of "database schema." Describe the types of schemas that exist in a database complying with the three levels ANSI/SPARC architecture.

2. List four significant differences between a file-processing system and a DBMS.

3. List five responsibilities of a database management system. For each responsibility, explain the problems that would arise if the responsibility were not discharged.

4. What do understand by the term 'data model'? Explain the difference between a data model and its implementation. Why is the difference important?
5. What three main types of actions involve databases? Briefly discuss each.
6. What is the difference between logical data independence and physical data independence?
7. What is the difference between procedural and nonprocedural DMLs?
8. Discuss the different types of user-friendly interfaces and the types of users who typically use each.
9. Discuss some types of database utilities and tools and their functions.

10. Distinguish between strong and weak entity?
11. Should a real world object be modelled as an entity or as an attribute?
12. Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.
13. Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted.
14. We can convert any weak entity set to a strong entity set by simply adding appropriate attributes. Why, then, do we have weak entity sets?

UNIT II

1. Describe the differences in meaning between the terms *relation* and *relation schema*. Illustrate your answer by an example.

2. Discuss the characteristics of relations that make them different from ordinary tables and files.

3. Why are duplicate tuples not allowed in a relation?

4. Why are tuples in a relation not ordered?

5. What is the difference between a key and a superkey?

6. Discuss the various reasons that lead to the occurrence of null values in relations.

7. Define *foreign key.* What is this concept used for? How does it play a role in the join operation?

8. Does the relational model, as seen by an SQL query writer, provide physical and logical data independence? Explain.

9. What is the difference between a candidate key and the primary key for a given relation? What is a superkey?

10. Answer each of the following questions briefly. The questions are based on the following relational schema:
Emp(*eid:* integer, *ename:* string, *age:* integer, *salary:* real)
Works(*eid:* integer, *did:* integer, *pct time:* integer)
Dept(*did:* integer, *dname:* string, *budget:* real, *managerid:* integer)
a. Give an example of a foreign key constraint that involves the Dept relation. What are the options for enforcing this constraint when a user attempts to delete a Dept tuple?
b. Write the SQL statements required to create the above relations, including appropriate versions of all primary and foreign key integrity constraints.

11. Given two relations $R1$ and $R2$, where $R1$ contains N1 tuples, $R2$ contains N2 tuples, and $N2 > N1 > 0$, give the minimum and maximum possible sizes (in tuples) for the result relation produced by each of the following relational algebra expressions. In each case, state any assumptions about the schemas for $R1$ and $R2$ that are needed to make the expression meaningful:
(1) $R1 \cup R2$, (2) $R1 \cap R2$, (3) $R1 - R2$, (4) $R1 \times R2$, (5) $\sigma_{a=5}(R1)$, (6) $\pi_a(R1)$, and (7) $R1 = R2$

12. What is *relational completeness*? If a query language is relationally complete, can you write any desired query in that language?

13. What is an *unsafe* query? Give an example and explain why it is important to disallow such queries.

UNIT III

1. Explain what is meant by *repetition of information* and *inability to represent information*. Explain why each of these properties may indicate a bad relational database design.

2. Suppose that we decompose the schema $R = (A, B, C, D, E)$ into

(A, B, C)
(A, D, E)

Show that this decomposition is a lossless-join decomposition if the following set F of functional dependencies holds:

$A \rightarrow BC$
$CD \rightarrow E$
$B \rightarrow D$
$E \rightarrow A$

3. Use the definition of functional dependency to argue that each of Armstrong's axioms (reflexivity, augmentation, and transitivity) is sound.

4. Give a set of FDs for the relation schema *R(A,B,C,D)* with primary key *AB* under which *R* is in 1NF but not in 2NF.

5. Give a set of FDs for the relation schema *R(A,B,C,D)* with primary key *AB* under which *R* is in 2NF but not in 3NF.

6. Consider the relation schema *R(A,B,C)*, which has the FD $B \rightarrow C$. If *A* is a candidate key for *R*, is it possible for *R* to be in BCNF? If so, under what conditions? If not, explain why not.

7. Suppose that we have a relation schema *R(A,B,C)* representing a relationship between two entity sets with keys *A* and *B*, respectively, and suppose that *R* has (among others) the FDs $A \rightarrow B$ and $B \rightarrow A$. Explain what such a pair of dependencies means (i.e., what they imply about the relationship that the relation models).

8. Consider a relation *R* with five attributes *ABCDE*. You are given the following dependencies: $A \rightarrow B$, $BC \rightarrow E$, and $ED \rightarrow A$.

a. List all keys for *R*.
b. Is *R* in 3NF?
c. Is *R* in BCNF?

9. Describe the purpose of normalizing the data? Describe the types of anomalies that may occur on a relation that has redundant data?

10. Give an example of a relation which is in 3NF but not in BCNF? How will you convert that relation to BCNF?

11. Given a relation R(S, B, C, D) with key={S, B, D} and F={$S \rightarrow C$}. Identify the normal form of the relation R?

12. List the three design goals for relational databases, and explain why each is desirable.

13. Give an example of a relation schema *R* and a set of dependencies such that *R* is in BCNF, but is not in 4NF.

14. Explain why 4NF is a normal form more desirable than BCNF.

15. Explain how dangling tuples may arise. Explain problems that they may cause.

UNIT IV

1. List the ACID properties. Explain the usefulness of each.

2. Suppose that there is a database system that never fails. Is a recovery manager required for this system?

3. Database-system implementers have paid much more attention to the ACID properties than have file-system implementers. Why might this be the case?

4. Explain the distinction between the terms *serial schedule* and *serializable schedule*.

5. Since every conflict-serializable schedule is view serializable, why do we emphasize conflict serializability rather than view serializability?

6. Under what conditions is it less expensive to avoid deadlock than to allow deadlocks to occur and then to detect them?

7. If deadlock is avoided by deadlock avoidance schemes, is starvation still possible? Explain your answer.

8. What is meant by the concurrent execution of database transactions in a multiuser system? Discuss why concurrency control is needed, and give informal examples.

9. Discuss the different types of failures. What is meant by catastrophic failure?

10. What is the system log used for? What are the typical kinds of records in a system log? What are transaction commit points, and why are they important?

11. What is a schedule (history)? Define the concepts of recoverable, cascadeless, and strict schedules, and compare them in terms of their recoverability.

12. Discuss the different measures of transaction equivalence. What is the difference between conflict equivalence and view equivalence?

13. Discuss the different measures of transaction equivalence. What is the difference between conflict equivalence and view equivalence?

14. Discuss how serializability is used to enforce concurrency control in a database system. Why is serializability sometimes considered too restrictive as a measure of correctness for schedules?

UNIT V

1. Explain how the following differ: fragmentation transparency, replication transparency, and location transparency.

2. When is it useful to have replication or fragmentation of data? Explain your answer.

3. Explain the notions of transparency and autonomy. Why are these notions desirable from a human-factors standpoint?

4. To build a highly available distributed system, you must know what kinds of failures can occur.
a. List possible types of failure in a distributed system.
b. Which items in your list from part a are also applicable to a centralized system?

5. Explain the difference between data replication in a distributed system and the maintenance of a remote backup site.

6. What is strict 2PL? Explain its role in Lock-Based Concurrency Control.

7. Show that the two-phase locking protocol ensures conflict serializability, and that transactions can be serialized according to their lock points.

8. Consider the following two transactions:

T31: read(A);
read(B);
if A = 0 **then** B := B + 1;
write(B).
T32: read(B);
read(A);
if B = 0 **then** A := A + 1;
write(A).

Add lock and unlock instructions to transactions *T*31 and *T*32, so that they observe the two- phase locking protocol. Can the execution of these transactions result in a deadlock?

9. What benefit does strict two-phase locking provide? What disadvantages result?

10. What benefit does rigorous two-phase locking provide? How does it compare with other forms of two-phase locking?

11. Most implementations of database systems use strict two-phase locking. Suggest three reasons for the popularity of this protocol.

12. When a transaction is rolled back under timestamp ordering, it is assigned a new timestamp. Why can it not simply keep its old timestamp?

13. In multiple-granularity locking, what is the difference between implicit and explicit locking?

14. Show that there are schedules that are possible under the two-phase locking protocol, but are not possible under the timestamp protocol, and vice versa.