DBMS Important Questions PDF (Unit-wise)

The most important DBMS questions are the three-schema architecture and data independence, ER diagrams with keys, relational algebra queries, SQL queries with joins and GROUP BY, functional dependencies, closure and candidate keys, normalisation up to BCNF, ACID properties, conflict serializability, two-phase locking, timestamp ordering and log-based recovery.

Key takeaways

- Normalisation and serializability numericals are asked in almost every DBMS paper.
- Practise writing the same query in relational algebra and SQL.
- Draw ER diagrams with correct notation for weak entities, cardinality and participation.
- These are practice questions, not past papers; check your own syllabus and official previous papers.

About these important questions

This page lists the most important Database Management Systems questions for B.Tech CSE and IT exams, arranged unit by unit. The units follow the standard syllabus that most Indian technical universities share, so the order may differ slightly from yours.

Please note: these are practice questions based on common syllabus topics. They are not verified past papers, and we do not claim that any question will appear in your exam. Check your own university syllabus first, then solve the official previous year papers. AKTU students can download them from the AKTU old question paper archive.

Unit 1: Introduction and ER Model

Covers DBMS basics, architecture, data models and ER modelling.

Short questions (2 marks)

- List four advantages of a DBMS over a file system.
- Define physical and logical data independence.
- What is a weak entity set?
- Differentiate between a super key, candidate key and primary key.

Long questions

1. Explain the three-schema architecture of a DBMS with a diagram.

2. Explain the roles of the DBA and the component modules of a DBMS.
3. Draw an ER diagram for a university, hospital or library database, showing keys, cardinality and participation.
4. Explain generalisation, specialisation and aggregation with examples.
5. Convert an ER diagram into relational tables, step by step.

Unit 2: Relational Model, Relational Algebra and SQL

Covers constraints, relational algebra, relational calculus and SQL.

Short questions (2 marks)

- What is referential integrity?
- Differentiate between DELETE, TRUNCATE and DROP.
- What is a view?

Long questions

1. Explain the relational algebra operations (select, project, union, set difference, Cartesian product, join, division) with examples.
2. For a given schema such as Employee(eid, name, salary, dept), write queries in relational algebra and SQL. *Hint:* expect questions on "highest salary", "employees in each department" and "employees who work on all projects" (division).
3. Explain tuple and domain relational calculus with examples.
4. Explain the types of joins in SQL with output tables.
5. Explain triggers, cursors and stored procedures with an example.
6. Write SQL using GROUP BY, HAVING, nested queries and EXISTS.

Unit 3: Database Design and Normalisation

Covers functional dependencies, normal forms and decomposition. This is the most numerical unit.

Short questions (2 marks)

- Define a functional dependency and a trivial FD.
- State Armstrong's axioms.
- What is a lossless join decomposition?

Long questions

1. Given R(A, B, C, D, E) and a set of FDs, find the closure of an attribute set and all candidate keys.
2. Find the canonical (minimal) cover of a set of FDs.
3. Explain 1NF, 2NF, 3NF and BCNF with an example, and normalise a given relation up to BCNF.
4. Check whether a decomposition is lossless and dependency preserving.
5. Explain multivalued dependency and 4NF, and join dependency and 5NF.

Unit 4: Transaction Processing and Concurrency Control

Covers ACID, schedules, serializability and concurrency protocols.

Short questions (2 marks)

- Define ACID properties.
- What is a cascading rollback?
- Differentiate between a shared and an exclusive lock.

Long questions

1. Draw the state diagram of a transaction and explain each state.
2. Check whether a schedule is conflict serializable using a precedence graph, and test view serializability.
3. Explain recoverable, cascadeless and strict schedules.
4. Explain the two-phase locking protocol and its variants (strict and rigorous 2PL). Does it prevent deadlock?
5. Explain timestamp-based protocols and Thomas' write rule.
6. Explain deadlock prevention with wait-die and wound-wait.

Unit 5: Recovery, Indexing and Distributed Databases

Covers recovery techniques, file organisation, indexing and an introduction to distributed databases. Some universities put indexing in Unit 3 instead.

Short questions (2 marks)

- What is a checkpoint?
- Differentiate between dense and sparse indexes.

Long questions

1. Explain log-based recovery with deferred and immediate database modification.
2. Explain shadow paging and its advantages and drawbacks.
3. Explain the ARIES recovery algorithm in brief.
4. Explain B+ tree indexing, and insert keys into a B+ tree of a given order.
5. Explain static and dynamic hashing.
6. Explain fragmentation, replication and the two-phase commit protocol in distributed databases.

Sample answer outline: Normalise a relation up to BCNF

A normalisation question usually gives a relation and a set of functional dependencies. Answer in this order:

1. **Find the candidate keys.** List the attributes that never appear on the right side of any FD. Compute their closure and grow the set until it covers every attribute. Write each key you find.
2. **Mark prime and non-prime attributes.** Prime attributes are part of some candidate key.

3. **Check each normal form in turn.** 2NF fails if a non-prime attribute depends on part of a key. 3NF fails if a non-prime attribute depends on another non-prime attribute (transitive dependency). BCNF fails if the left side of any non-trivial FD is not a super key.
4. **Decompose.** For each violating FD X → Y, create a relation with X and Y, and keep X in the remaining relation. Name every new relation and its key.
5. **Verify.** State whether the decomposition is lossless (the common attributes form a key of one relation) and whether all FDs are preserved. BCNF does not always preserve dependencies, and saying so earns marks.

Marks weightage and how to write answers

A common pattern, used by AKTU and many other universities, has a compulsory short-answer section that covers every unit, followed by long-answer questions with a choice inside each unit. Marks per question change between universities and years, so confirm the scheme on a recent official paper.

- **Short answers (about 2 marks):** a crisp definition, one example or diagram, and nothing more. Three to five lines is enough.
- **Long answers (7 to 10 marks):** start with a definition, add a labelled diagram or table, explain the steps, and finish with an example, advantages or a comparison.
- For normalisation, write the FDs, find the keys, state which normal form is violated and then decompose. Each step earns marks.
- For serializability, draw the precedence graph and say clearly whether it has a cycle.

A 7-day revision plan for DBMS

1. **Days 1 to 5:** one unit a day. Read your notes, then write the long answers for that unit from memory and check them against your notes.
2. **Day 6:** redo the numericals: closure and candidate keys, normalisation to BCNF, precedence graphs and B+ tree insertion.
3. **Day 7:** solve one full official previous year paper in three hours, then check which units cost you time.

Keep a single sheet of formulas, definitions and diagrams for the last hour before the exam.

Frequently asked questions

Which DBMS topics are most important for exams?

Normalisation with candidate keys, SQL and relational algebra queries, ER diagrams, serializability and two-phase locking appear most often. Recovery and indexing are also common long answers.

How do I find candidate keys quickly?

Find the attributes that never appear on the right side of any FD; they must be in every key. Compute their closure, then add other attributes one at a time until the closure covers the whole relation.

Is DBMS important for placements?

Yes. SQL queries, normalisation, joins, indexing and transactions are common in technical interviews for developer, data and testing roles.

Are these DBMS questions from previous year papers?

No. They are practice questions built from topics that appear in most B.Tech Database Management Systems syllabi. Use them with your own university's official previous papers, for example the AKTU archive at https://aktu.ac.in/old-question-paper.html.

Is this PDF free?

Yes. Fill in the short form on this page and the PDF downloads right away.

Sources and further reading

- AKTU: old question papers (official archive)
- PostgreSQL documentation: SQL language
- NPTEL: online courses from IITs (search for Database Management System)

About this PDF

These are practice questions for revision, prepared by the Uncodemy trainer team from common B.Tech syllabus topics. They are not past papers, a certificate or a promise of marks. Spot an error? Write to info@uncodemy.com and the team will review it.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550