

**Malla Reddy College of Engineering & Technology**

(Autonomous Institution- UGC, Govt. of India)

(Affiliated to JNTUH, Hyderabad, Approved by AICTE, NBA &NAAC with ‘A’ Grade)

**DBMS Guide**

1. **Define a database and its importance**.  
   *Answer:* A database is a structured collection of data. It allows efficient data storage, retrieval, management, and sharing.
2. **What are database applications?**  
   *Answer:* Systems such as banking, telecom, inventory, airline reservation, social networks—where data persistence and integrity are essential.
3. **Explain data abstraction levels.***Answer:* Three levels—physical (how data is stored), logical (structure of data), and view (user-specific access).
4. **What is the difference between instance and schema?***Answer:* Schema is the database structure; instance is snapshot data at a point in time.
5. **Who are database users and administrators?***Answer:* Users (naïve, sophisticated) interact via apps; DBA manages schema, storage, security, performance.
6. **Explain the storage manager.***Answer:* Manages disk space, indexing, caching, I/O, and file organization.
7. **What is the role of a query processor?***Answer:* Transforms queries into executable plans, optimizing performance.
8. **Describe the relational model.***Answer:* Data represented in tables (relations) with rows (tuples) and columns (attributes), using keys for integrity.
9. **What are primary and foreign keys?***Answer:* Primary key uniquely identifies a row; foreign key refers to primary key in another table (ensures referential integrity).
10. **Draw and explain a basic ER diagram.***Answer:* Entities (rectangles), attributes (ovals), relationships (diamonds). Example: Student —<enrolls>— Course.
11. **What are ER model design issues?***Answer:* Naming, completeness, redundancy, relationships arity, weak entities.
12. **Compare ER modeling and relational schema**.  
    *Answer:* ER is conceptual; relational is logical schema derived post-design.
13. **Explain selection and projection operations.**  
    *Answer:* Selection (σ) filters rows; Projection (π) selects columns.
14. **What are set operations?***Answer:* Union, intersection, difference—require union-compatible relations.
15. **Describe join types.***Answer:* Natural join, equi-join, outer joins (left, right, full), theta-join.
16. **What is division in relational algebra?***Answer:* Division (÷) finds tuples in R associated with all values in S (e.g., students who took all required courses).
17. **Differentiate relational algebra and calculus.***Answer:* Algebra is procedural; calculus (TRC and DRC) is declarative.
18. **How do you express "students older than 20" in TRC?***Answer:* {t | Student(t) ∧ t.age > 20}.
19. **Explain SQL basic SELECT structure.***Answer:* SELECT ... FROM ... WHERE ... GROUP BY ... HAVING ... ORDER BY ...
20. **What are aggregate functions?***Answer:* SUM, AVG, MAX, MIN, COUNT, used with GROUP BY.
21. **When are nested subqueries used?**  
    *Answer:* In WHERE or FROM clauses to filter based on another query.
22. **What is a view in SQL?**  
    *Answer:* Virtual table built from a SELECT query for abstraction/security.
23. **Explain triggers, their types and use.***Answer:* Automatic procedures on INSERT, UPDATE, DELETE; Types: BEFORE, AFTER.
24. **Write a stored procedure to update student marks.  
    *Answer:***

CREATE PROCEDURE update\_marks(IN sid INT, IN newm INT)

BEGIN

UPDATE Student SET marks = newm WHERE id = sid;

END;

1. **Discuss DML vs. DDL statements.***Answer:* DML manipulates data (SELECT, INSERT); DDL defines schema (CREATE, ALTER).
2. **What is normalization?***Answer:* Eliminates redundancy and anomalies through normal forms (1NF–5NF).
3. **Describe functional dependency.***Answer:* Attribute A uniquely determines B (A → B).
4. **Explain 1NF with example.***Answer:* 1NF requires atomic values; no repeating groups.
5. **What is 2NF?***Answer:* 1NF + no partial dependency on part of a composite key.
6. **Define 3NF.***Answer:* 2NF + no transitive dependency on non-key attributes.
7. **Explain BCNF.***Answer:* Every determinant must be a candidate key.
8. **What is multi-valued dependency and 4NF?***Answer:* A multi-valued dependency X →→ Y and Z exists independently; 4NF requires no MVD except by key.
9. **Explain join dependencies and 5NF.**  
   *Answer:* Relation is decomposed into joins of projections; 5NF ensures lossless joins of irreducible schemas.
10. **What is lossless decomposition?***Answer:* Decomposition where joining yields original relation without spurious tuples.
11. **What is dependency preservation?***Answer:* All original functional dependencies can be enforced in the decomposed schema.
12. **Why normalization matters in DB design?***Answer:* It reduces anomalies, improves consistency and maintainability.
13. **Define a database transaction.**  
    *Answer:* A logical unit of work satisfying ACID properties.
14. **List ACID properties.***Answer:* Atomicity, Consistency, Isolation, Durability.
15. **Explain serializability.***Answer:* Concurrent execution schedule is correct if equivalent to a serial schedule.
16. **Differentiate conflict vs. view serializability.***Answer:* Conflict ensures ordering of conflicting operations; view focuses on final read outputs, a broader criterion.
17. **Give examples of lock-based protocols.***Answer:* 2PL, Strict 2PL for ensuring serializability.
18. **What are time-stamp protocols?***Answer:* Use timestamps to order transactions, avoiding deadlock (e.g., Thomas write rule).
19. **Explain validation-based concurrency control.**  
    *Answer:* Transactions execute optimistically and validate before commit.
20. **Discuss multiple granularity locking.***Answer:* Lock at different levels in hierarchy (e.g., database, table, tuple).
21. **Define recoverable and cascade-free schedules.**  
    *Answer:* Recoverable avoids committing before dependencies commit; cascade-free avoids rollbacks propagated.
22. **What is the Write-Ahead Log protocol?**  
    *Answer:* Ensures changes are logged before actual data disk write to enable recovery.
23. **How does checkpointing improve recovery?***Answer:* Periodic checkpoints limit redo and undo operations to expedite crash recovery.
24. **Explain undo and redo operations.**  
    *Answer:* Used for restoring database to a consistent state; undo backs off uncommitted actions, redo replays committed ones.
25. **What is the ARIES recovery algorithm?***Answer:* Advanced ARchitectural Information System approach with repeating history, logging changes and fuzzy checkpoints.
26. **Describe buffer management in DBMS.***Answer:* Manages in-memory pages using strategies like LRU, MRU, Clock for efficient I/O.

**SQL Coding Questions**

1. **Create a table for STUDENT with appropriate data types and a primary key.**

CREATE TABLE Student (

sid INT PRIMARY KEY,

name VARCHAR2(100),

dob DATE,

dept VARCHAR2(50)

);

1. **Insert multiple rows into STUDENT**.

INSERT INTO Student VALUES (1,'Alice',DATE '2000-05-20','CS');

INSERT INTO Student VALUES (2,'Bob',DATE '1999-10-15','IT');

1. **Select student name and department.**

SELECT name, dept FROM Student;

1. **Find students born after 1999.**

SELECT \* FROM Student WHERE dob > DATE '1999-12-31';

1. **Rename output column in SELECT.**

SELECT name AS student\_name FROM Student;

1. **Count students in each department.**

SELECT dept, COUNT(\*) AS cnt FROM Student GROUP BY dept;

1. **Find students whose name starts with 'A'.**

SELECT \* FROM Student WHERE name LIKE 'A%';

1. **List students ordered by name ascending.**

SELECT \* FROM Student ORDER BY name;

1. **Delete student with sid 2.**

DELETE FROM Student WHERE sid = 2;

1. **Update student department.**

UPDATE Student SET dept = 'ECE' WHERE sid = 1;

1. **Create COURSE and ENROLL tables with foreign keys.**

CREATE TABLE Course (

cid INT PRIMARY KEY,

title VARCHAR2(100)

);

CREATE TABLE Enroll (

sid INT,

cid INT,

grade CHAR(2),

PRIMARY KEY (sid, cid),

FOREIGN KEY (sid) REFERENCES Student(sid),

FOREIGN KEY (cid) REFERENCES Course(cid)

);

1. **Find students enrolled in course id=10.**

SELECT s.name

FROM Student s JOIN Enroll e

ON s.sid = e.sid

WHERE e.cid = 10;

1. **List students not enrolled in any course.**

SELECT name FROM Student

WHERE sid NOT IN (SELECT sid FROM Enroll);

1. **Relational division: find students who took both course 10 and 20.**

SELECT sid

FROM Enroll

WHERE cid IN (10,20)

GROUP BY sid

HAVING COUNT(DISTINCT cid)=2;

1. **Create a view showing student-course enrollments.**

CREATE VIEW StudentCourse AS

SELECT st.name, c.title

FROM Student st

JOIN Enroll e ON st.sid=e.sid

JOIN Course c ON e.cid=c.cid;

1. **Use TRIGGER to audit deletes on ENROLL.**

CREATE TABLE EnrollAudit (

sid INT, cid INT, deleted\_at TIMESTAMP

);

CREATE OR REPLACE TRIGGER trg\_delete\_enroll

BEFORE DELETE ON Enroll

FOR EACH ROW

BEGIN

INSERT INTO EnrollAudit VALUES (:OLD.sid, :OLD.cid, SYSTIMESTAMP);

END;

1. **Create stored procedure to update grade.**

CREATE OR REPLACE PROCEDURE UpdateGrade(p\_sid INT, p\_cid INT, p\_grade CHAR) IS

BEGIN

UPDATE Enroll SET grade = p\_grade WHERE sid=p\_sid AND cid=p\_cid;

END;

1. **Invoke the procedure to update grade.**

BEGIN

UpdateGrade(1,10,'A');

END;

1. **Drop view, table, and procedure.**

DROP VIEW StudentCourse;

DROP PROCEDURE UpdateGrade;

DROP TABLE Enroll;

1. **Demonstrate UNION, INTERSECT, MINUS.**

SELECT dept FROM Student WHERE dept='CS'

UNION

SELECT dept FROM Student WHERE dept='ECE';

-- INTERSECT, MINUS can similarly be used

1. **Use aggregate functions MAX, MIN, AVG.**

SELECT MAX(salary), AVG(salary) FROM Employee;

1. **Nested subquery example.**

SELECT name FROM Student

WHERE sid IN (SELECT sid FROM Enroll WHERE grade='A');

1. **Tuple relational calculus equivalent.**

{ s.name | Student(s) AND s.sid IN (e.sid WHERE e.grade='A') }

1. **Domain relational calculus equivalent.**

{ n | ∃sid, ∃cid, Student(sid,n,d,dept) AND Enroll(sid,cid,'A') }

1. **Rename a column in query using ALIAS.**

SELECT s.sid AS student\_id, s.name AS student**\_name FROM Student s;**