



SQL & Database Design

Section A – Database Concepts & Design Theory

#	Question	Short Answer
1	What is a database?	Structured collection of data managed by DBMS.
2	What is DBMS vs RDBMS?	DBMS = data store, RDBMS = tables with relationships (e.g., MySQL).
3	What is a schema?	Logical structure or blueprint of database.
4	Define a table and a tuple.	Table = entity, Tuple = row/record.
5	What is a primary key?	Uniquely identifies each record.
6	Difference between primary and unique key	Primary → 1 per table, no NULL; Unique → multiple allowed, allows NULL.
7	What is a foreign key?	Establishes relationship between tables.
8	What is a composite key?	Combination of two or more fields as key.
9	What is normalization?	Process to reduce redundancy & maintain integrity.
10	Explain 1NF, 2NF, 3NF	1NF: atomic values; 2NF: full key dependency; 3NF: no transitive dependency.

Section B – ER Diagram & Relationships

#	Question	Short Answer
11	What is an ER diagram?	Visual model of entities and their relationships.
12	Define entity, attribute, and relationship.	Entity = object; Attribute = property; Relationship = association.
13	What is cardinality?	Defines relationship count (1:1, 1:N, M:N).
14	Example of 1:M relationship	Department → Employees.
15	How do you convert ERD to schema?	Entities → tables, relationships → foreign keys.
16	What is weak entity?	Depends on another entity for existence (e.g., OrderItem).
17	What is a junction table?	Resolves many-to-many relationships.
18	What is referential integrity?	Ensures foreign keys point to valid parent records.
19	Explain cascade delete/update.	Auto apply parent changes to child records.
20	Real use of ER diagram	Visualize project database before implementation.



Section C – SQL Queries (Core DML & Filtering)

#	Query / Question	Answer / Example
21	Select all columns from Employees table.	SELECT * FROM Employees;
22	Select EmployeeName and Salary.	SELECT EmployeeName, Salary FROM Employees;
23	Filter employees with salary > 50000.	SELECT * FROM Employees WHERE Salary > 50000;
24	Sort employees by joining date.	SELECT * FROM Employees ORDER BY JoinDate;
25	Find distinct job titles.	SELECT DISTINCT JobTitle FROM Employees;
26	Use alias for column.	SELECT Salary*12 AS AnnualSalary FROM Employees;
27	Find employees hired in 2023.	SELECT * FROM Employees WHERE YEAR(JoinDate)=2023;
28	Limit top 5 highest salaries.	SELECT * FROM Employees ORDER BY Salary DESC LIMIT 5;
29	Count employees in each department.	SELECT DeptID, COUNT(*) FROM Employees GROUP BY DeptID;
30	Use HAVING to filter groups.	SELECT DeptID, AVG(Salary) FROM Employees GROUP BY DeptID HAVING AVG(Salary)>60000;

Section D – JOINS & Subqueries

#	Question	Short Answer
31	What is a JOIN?	Combines rows from two tables based on condition.
32	INNER JOIN example	SELECT e.Name, d.DeptName FROM Emp e INNER JOIN Dept d ON e.DeptID=d.DeptID;
33	LEFT JOIN example	Returns all left + matched right records.
34	RIGHT JOIN example	Returns all right + matched left records.
35	FULL JOIN example	Combines all records from both sides (use UNION).
36	What is a subquery?	Query inside another query.
37	Example of subquery	SELECT Name FROM Emp WHERE Salary>(SELECT AVG(Salary) FROM Emp);
38	Difference between correlated and non-correlated subquery	Correlated = inner query depends on outer; Non-correlated = independent.
39	Nested subquery example	2-level subquery → (SELECT MAX(Salary) FROM (SELECT ...))
40	Self Join example	SELECT a.Name, b.Name FROM Emp a, Emp b WHERE a.ManagerID=b.EmpID;



Section E – Constraints & Integrity Rules

#	Question	Short Answer
41	What are SQL constraints?	Rules to maintain data integrity.
42	Types of constraints	PRIMARY, UNIQUE, CHECK, DEFAULT, FOREIGN.
43	CHECK constraint example	CHECK (Age >= 18)
44	DEFAULT constraint	Provides default value if not specified.
45	UNIQUE constraint use	Ensures no duplicate value.
46	NOT NULL constraint	Prevents NULL value insertion.
47	Referential integrity	Maintains parent-child consistency.
48	ON DELETE CASCADE	Deletes child when parent deleted.
49	Entity integrity	Ensures each row has unique non-null key.
50	Domain integrity	Data must follow allowed type/range.

Section F – Aggregation & Group Functions

#	Question	Short Answer
51	What are aggregate functions?	Operate on sets of values (SUM, COUNT, AVG, MAX, MIN).
52	Find total employees.	SELECT COUNT(*) FROM Employees;
53	Find max salary.	SELECT MAX(Salary) FROM Employees;
54	Find avg salary in dept.	SELECT DeptID, AVG(Salary) FROM Employees GROUP BY DeptID;
55	Use COUNT DISTINCT	SELECT COUNT(DISTINCT DeptID) FROM Employees;
56	Group by multiple columns	SELECT DeptID, Job, COUNT(*) FROM Employees GROUP BY DeptID, Job;
57	Having vs Where	HAVING filters after grouping; WHERE before.
58	Find departments with >10 employees.	SELECT DeptID FROM Employees GROUP BY DeptID HAVING COUNT(*)>10;
59	Find department with max total salary.	SELECT DeptID, SUM(Salary) FROM Employees GROUP BY DeptID ORDER BY SUM(Salary) DESC LIMIT 1;
60	Count employees joined per year.	SELECT YEAR(JoinDate), COUNT(*) FROM Employees GROUP BY YEAR(JoinDate);



Section G – Transactions & Indexing

#	Question	Short Answer
61	What is a transaction?	A unit of work (commit/rollback).
62	What are ACID properties?	Atomicity, Consistency, Isolation, Durability.
63	COMMIT vs ROLLBACK	COMMIT = save; ROLLBACK = undo.
64	What is SAVEPOINT?	Marks partial rollback point.
65	What is an index?	Improves query speed using sorted reference structure.
66	Clustered vs Non-clustered index	Clustered = rearranges data; Non-clustered = separate structure.
67	When to use index	On columns frequently used in WHERE/JOIN.
68	Drawback of too many indexes	Slow INSERT/UPDATE performance.
69	What is deadlock?	Two transactions waiting on each other's lock.
70	How to avoid deadlock	Consistent lock order or short transactions.

Section H – Integration & Real-World Scenarios

#	Question	Expected Answer
71	How do you connect database in Java?	JDBC → DriverManager → Connection → Statement.
72	How do you integrate React front-end with SQL backend?	React → API (Spring Boot / Flask) → DB query execution.
73	What is ORM?	Object Relational Mapping (e.g., Hibernate).
74	Advantage of ORM	Reduces SQL boilerplate, automatic mapping.
75	Example of schema for e-commerce	Tables: Users, Products, Orders, OrderItems, Payments.
76	Example join query for Orders & Users	SELECT u.Name, o.Total FROM Users u JOIN Orders o ON u.UserID=o.UserID;
77	Why normalization before integration?	Prevent redundancy and ensure data consistency.
78	When is denormalization useful?	For analytics or read-heavy operations.
79	Common SQL injection prevention method	Use parameterized queries / ORM.
80	Why transactions in backend integration	To ensure consistent update of related records.