SQL Question Bank with Answers

🕒 Duration: 2 Sessions (60 + 60 minutes)

🎯 Focus: Concept clarity + Query logic + Output prediction

Layer 1 – ER Diagram & Normalization

No	Question	Answer
1	Draw an ER diagram for Customers, Orders, and Products.	Customer (1) — (M) Order; Product (1) — (M) Order.
2	Define cardinality with examples.	1:1 (Person–Aadhar), 1:N (Customer–Order), M:N (Student–Course).
3	Identify PK and FK in Customer–Order.	PK: Customer_ID; FK: Customer_ID in Orders.
4	Why normalize a database?	To remove redundancy and anomalies.
5	Convert table to 3NF.	Split into Customer(Cust_ID, Name, City) and Order(Order_ID, Cust_ID, Product, Price).
6	1NF vs 2NF vs 3NF	1NF → Atomic values; 2NF → No partial dependency; 3NF → No transitive dependency.
7	Where did you apply normalization in your project?	Separated user and login tables to avoid duplication.
8	Define update anomaly.	Inconsistent data during redundant updates.

Layer 2 – Database Integrity Rules

No	Question	Answer
9	What is Domain Integrity?	Valid data types and ranges (e.g., Age > 0).
10	What is Entity Integrity?	Each row uniquely identified (Primary Key).
11	What is Referential Integrity?	FK ensures referenced record exists in parent table.
12	Which integrity prevents orphan records?	Referential Integrity.
13	Write constraint examples.	PK, FK, CHECK(Age>18), UNIQUE(Email).
14	Why use constraints?	To maintain accurate and consistent data.

Layer 3 – Query Clauses

No	Question	Answer
15	Display all customers from Chennai.	SELECT * FROM Customer WHERE City='Chennai';
16	Show all unique cities.	SELECT DISTINCT City FROM Customer;
17	Orders > 20000.	SELECT * FROM Orders WHERE Amount>20000;
18	Total sales per customer.	SELECT Cust_ID,SUM(Amount) FROM Orders GROUP BY Cust_ID;
19	Customers with total > 40000.	SELECT Cust_ID,SUM(Amount) FROM Orders GROUP BY Cust_ID HAVING SUM(Amount)>40000;
20	Top 3 customers by amount.	SELECT Cust_ID,SUM(Amount) FROM Orders GROUP BY Cust_ID ORDER BY SUM(Amount) DESC LIMIT 3;
21	Exclude Delhi customers.	SELECT * FROM Customer WHERE City!='Delhi';
22	Sort by name asc & amount desc.	SELECT * FROM Orders ORDER BY Name ASC, Amount DESC;
23	Orders between 10000–50000.	SELECT * FROM Orders WHERE Amount BETWEEN 10000 AND 50000;
24	Customers with at least one order.	SELECT * FROM Customer WHERE EXISTS (SELECT * FROM Orders WHERE Customer.Customer_ID=Orders.Customer_ID);

Layer 4 – Joins

No	Question	Answer
25	INNER JOIN customer–order.	SELECT c.Name,o.Product FROM Customer c JOIN Orders o ON c.Customer_ID=o.Customer_ID;
26	LEFT JOIN customers with orders.	SELECT c.Name,o.Product FROM Customer c LEFT JOIN Orders o ON c.Customer_ID=o.Customer_ID;
27	RIGHT JOIN orders with customers.	SELECT c.Name,o.Product FROM Customer c RIGHT JOIN Orders o ON c.Customer_ID=o.Customer_ID;
28	FULL JOIN both tables.	SELECT * FROM Customer c FULL JOIN Orders o ON c.Customer_ID=o.Customer_ID;
29	Difference between INNER & OUTER JOIN.	INNER → common data only; OUTER → all rows + nulls.
30	Product with buyer name.	SELECT p.ProductName,c.Name FROM Product p JOIN Orders o ON p.PID=o.PID JOIN Customer c ON c.CID=o.CID;

Layer 5 – Subqueries

No	Question	Answer
31	Customers who spent > 40000.	SELECT Name FROM Customer WHERE Customer_ID IN (SELECT Customer_ID FROM Orders WHERE Amount>40000);
32	Orders = max amount.	SELECT * FROM Orders WHERE Amount=(SELECT MAX(Amount) FROM Orders);
33	Customers with no orders.	SELECT * FROM Customer WHERE Customer_ID NOT IN (SELECT Customer_ID FROM Orders);
34	Products above average price.	SELECT * FROM Product WHERE Price>(SELECT AVG(Price) FROM Product);
35	Customers above average spending.	SELECT Name FROM Customer WHERE Customer_ID IN (SELECT Customer_ID FROM Orders GROUP BY Customer_ID HAVING SUM(Amount)>(SELECT AVG(SUM(Amount)) FROM Orders GROUP BY Customer_ID));

Layer 6 – Transactions & ACID

No	Question	Answer
36	What is a transaction?	A logical unit of work (set of SQL statements).
37	ACID properties?	Atomicity, Consistency, Isolation, Durability.
38	TCL commands?	BEGIN; COMMIT; ROLLBACK; SAVEPOINT;
39	What if a transaction fails midway?	Rollback occurs (Atomicity).
40	How does isolation help?	Prevents conflicts in concurrent transactions.