

Database Systems

SQL and Relational Model Quiz – Answer Key

1. **(C) Third Normal Form (3NF).** 3NF requires no transitive dependencies—non-key attributes must depend only on the primary key, not on other non-key attributes.
2. **(B) HAVING.** WHERE filters rows before grouping; HAVING filters groups after GROUP BY aggregation. Example: `HAVING COUNT(*) > 5`.
3. **(B) Values reference existing values in another table.** Foreign keys enforce referential integrity—a value must exist in the referenced table’s primary key column.
4. **(D) FULL OUTER JOIN.** Returns all rows from both tables: matching rows are combined; non-matching rows have NULLs for the other table’s columns.
5. **(C) Concurrent transactions don’t interfere with each other.** Isolation ensures transactions execute as if they were the only operation, preventing dirty reads, non-repeatable reads, and phantom reads.
6. **False.** Primary keys must be unique AND NOT NULL. They uniquely identify each row, so NULL values (representing unknown) are prohibited.
7. **True.** DELETE is a DML operation that can be rolled back if issued within an uncommitted transaction. TRUNCATE and DROP typically cannot be rolled back.
8. **False.** Indexes improve read performance but add overhead to write operations (INSERT, UPDATE, DELETE). They also consume storage space. Over-indexing can degrade overall performance.

9. Purpose of Normalization:

- Eliminate data redundancy
- Prevent update, insertion, and deletion anomalies
- Ensure data integrity
- Organize data efficiently

First Normal Form (1NF):

- Requirement: Atomic values only (no repeating groups or arrays)
- Violation: `Student(ID, Name, Courses)` where `Courses = “Math, Physics, Chemistry”`
- Solution: Create separate rows or a junction table for each course

Second Normal Form (2NF):

- Requirement: 1NF + no partial dependencies (non-key attributes depend on entire primary key)
- Violation: `OrderItem(OrderID, ProductID, ProductName, Quantity)` — `ProductName` depends only on `ProductID`

- Solution: Separate into Orders and Products tables

Third Normal Form (3NF):

- Requirement: 2NF + no transitive dependencies
- Violation: Employee(ID, DeptID, DeptName, DeptLocation) — DeptName and DeptLocation depend on DeptID, not ID
- Solution: Separate into Employee(ID, DeptID) and Department(DeptID, DeptName, DeptLocation)

10. ACID Properties:

Atomicity:

- Definition: Transaction executes completely or not at all (“all or nothing”)
- Mechanism: Transaction logs, rollback capability
- Example: Bank transfer—both debit and credit must succeed or neither happens

Consistency:

- Definition: Transaction brings database from one valid state to another
- Mechanism: Constraints, triggers, validation rules
- Example: Account balance cannot go negative; referential integrity maintained

Isolation:

- Definition: Concurrent transactions don’t interfere
- Mechanism: Locking (pessimistic), MVCC (optimistic), isolation levels
- Levels: Read Uncommitted, Read Committed, Repeatable Read, Serializable
- Example: Two users updating same account see consistent data

Durability:

- Definition: Committed transactions persist even after system failure
- Mechanism: Write-ahead logging (WAL), transaction logs, checkpointing
- Example: Confirmed purchase remains recorded after power outage

Implementation mechanisms: Logging and recovery systems, lock managers, buffer management, and careful protocol design ensure ACID compliance in modern RDBMS.