**SCENARIO-BASED EXPERT MCQs – DATABASE NORMALIZATION**

**1️⃣ You are designing a table to store students and their enrolled courses. Each student can have multiple courses. Best normalized design?**

A. Single table with CSV list of courses  
B. STUDENT + COURSE + STUDENT\_COURSE ✅  
C. Single table with multiple course columns (course1, course2…)  
D. STUDENT + COURSE only

**Explanation:** M:N relationship → junction table. Avoid repeating groups (violates 1NF).

**2️⃣ An employee table has columns: EmpID, EmpName, DeptID, DeptName. What is violated?**

A. 1NF  
B. 2NF  
C. 3NF ✅  
D. BCNF

**Explanation:** DeptName depends on DeptID, not EmpID (transitive dependency). Violates 3NF.

**3️⃣ A table has a composite PK (OrderID, ProductID) and an attribute ProductName. Is it normalized?**

A. Yes, it’s fine  
B. No, violates 1NF  
C. No, violates 2NF ✅  
D. No, violates BCNF

**Explanation:** ProductName depends only on ProductID, not the whole primary key → partial dependency → violates 2NF.

**4️⃣ You remove all transitive dependencies but still get data anomalies. Why?**

A. Not in 1NF  
B. Missing primary key  
C. Must normalize to BCNF ✅  
D. Too many indexes

**Explanation:** 3NF solves most cases, but BCNF handles more complex functional dependency issues like when non-key attributes determine keys.

**5️⃣ You have a table with attributes A, B, C, D. A → B, B → C, C → A. What is the highest normal form?**

A. 1NF  
B. 2NF  
C. 3NF  
D. Not even 1NF ✅

**Explanation:** Circular dependencies mean no real key. Without a valid primary key, normalization cannot even start.

**6️⃣ In a normalized schema, queries require 6 joins and performance drops. What is the normalized vs performance trade-off action?**

A. Add redundant columns intentionally ✅  
B. Further normalize  
C. Remove all indexes  
D. Use composite keys everywhere

**Explanation:** Controlled denormalization improves performance in read-heavy systems. Normalization is logical; performance is physical.

**7️⃣ A table has attributes: (City, State, Country). The primary key is City. What is wrong?**

A. City is not a good key ✅  
B. Violates 1NF  
C. Too many columns  
D. Nothing, it's fine

**Explanation:** City is not unique globally. Incorrect primary key leads to normalization illusions but data is incorrect.

**8️⃣ You have a table CUSTOMER with: (CustomerID PK, Name, Address, City, State, Country). To achieve 3NF:**

A. Keep as is  
B. Move Address to separate table  
C. Move City, State, Country to a City table ✅  
D. Create multiple columns

**Explanation:** City → State → Country is functional dependency. Must remove transitive dependencies.

**9️⃣ You decompose a table to achieve BCNF, but now you lost some valid data combinations. What happened?**

A. Bad indexing  
B. Non-lossless decomposition ✅  
C. You didn’t create correct foreign keys  
D. Too many NULLs

**Explanation:** Normalization must ensure lossless join. If that property breaks, decomposition is incorrect.

**🔟 A table is in 3NF but a non-key column still determines a key column. What NF is violated?**

A. 2NF  
B. 3NF  
C. BCNF ✅  
D. 4NF

**Explanation:** BCNF requirement: every determinant must be a candidate key. Non-key determining key violates BCNF.