**Q. 1**  
Which **SQL code snippet** correctly creates a table with **foreign key constraints** and **checks** for valid data?

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

CustomerID INT,

OrderDate DATE,

CONSTRAINT FK\_Customer FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),

CONSTRAINT CHK\_OrderDate CHECK (OrderDate >= '2023-01-01')

);

**a)** This code correctly creates the Orders table with a foreign key and a check constraint.   
**b)** The foreign key constraint is incorrectly referencing the Customers table.  
**c)** The CHECK constraint should be placed after the PRIMARY KEY definition.  
**d)** Foreign key constraints cannot be defined in the CREATE TABLE statement.

**Q. 2**  
Which **SQL code snippet** demonstrates a correct use of **ALTER TABLE** to **add a new column** and **set a default value** in **MySQL**?

ALTER TABLE Employees

ADD COLUMN Department VARCHAR(100) DEFAULT 'General';

**a)** This code correctly adds a Department column with a default value.

**b)** Default values should be defined inside the CREATE statement, not with ALTER TABLE.  
**c)** The column type should be VARCHAR(50) instead of VARCHAR(100).  
**d)** The ALTER TABLE command cannot add a default value to a new column.

**Q. 3**  
Which **SQL code snippet** demonstrates **correct usage** of **JOIN** between the Customers and Orders tables to retrieve **customer names** and their **order details**?sql

Copy

SELECT Customers.CustomerName, Orders.OrderID, Orders.OrderDate

FROM Customers

INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID;

**a)** This code correctly uses INNER JOIN to retrieve customer details along with their orders.   
**b)** INNER JOIN should be replaced by LEFT JOIN to include customers with no orders.  
**c)** The INNER JOIN should use WHERE instead of ON.  
**d)** The ON condition should use Orders.CustomerID = Customers.CustomerID instead.

**Q. 4**  
Which **SQL code snippet** correctly drops a table Products if it exists, and prevents errors if the table is not found?

DROP TABLE IF EXISTS Products;

**a)** This code correctly drops the Products table if it exists.   
**b)** The DROP TABLE statement cannot use IF EXISTS in MySQL.  
**c)** The DROP statement should include CASCADE to handle dependent objects.  
**d)** The DROP TABLE command must always be executed with CASCADE to avoid errors.

**Q. 5**  
Which **SQL code snippet** demonstrates the **correct use** of **Codd's 12th rule** (View Updating Rule) in a **relational database**?sql

Copy

CREATE VIEW EmployeeDetails AS

SELECT EmployeeID, EmployeeName, Department FROM Employees

WHERE Active = 1;

**a)** This view is valid under Codd’s 12th rule because it can be updated through the underlying tables.

**b)** Views defined with WHERE clauses cannot be updated under Codd’s 12th rule.  
**c)** Codd’s 12th rule does not allow the creation of any views.  
**d)** Views should only contain JOIN operations to satisfy Codd’s 12th rule.

**Q. 6**  
Which **SQL code snippet** correctly creates a **table** with **unique constraints** on the Email field and Phone field, ensuring no duplicates?

CREATE TABLE Employees (

EmployeeID INT PRIMARY KEY,

Name VARCHAR(100),

Email VARCHAR(100) UNIQUE,

Phone VARCHAR(15) UNIQUE

);

**a)** This code correctly creates the Employees table with unique constraints on Email and Phone.   
**b)** The UNIQUE constraint cannot be applied to multiple columns in the CREATE TABLE statement.  
**c)** The Phone column should be indexed, not constrained as unique.  
**d)** The UNIQUE constraint is not valid for Phone in the CREATE statement.

**Q. 7**  
Which **SQL code snippet** correctly uses **GROUP BY** and **HAVING** to find **the number of orders per customer** that have **more than 5 orders**?

SELECT CustomerID, COUNT(OrderID) AS NumberOfOrders

FROM Orders

GROUP BY CustomerID

HAVING COUNT(OrderID) > 5;

**a)** This code correctly counts the number of orders per customer and filters customers with more than 5 orders.   
**b)** The HAVING clause should be used before the GROUP BY clause.  
**c)** The HAVING clause is not needed if COUNT is used in the SELECT statement.  
**d)** COUNT should be replaced with SUM to get the correct number of orders.

**Q. 8**  
Which **SQL code snippet** correctly implements **a trigger** to **update the LastModified field** in the Employees table whenever an Employee record is updated?

CREATE TRIGGER UpdateEmployeeTimestamp

BEFORE UPDATE ON Employees

FOR EACH ROW

SET NEW.LastModified = NOW();

**a)** This code correctly creates a trigger that updates LastModified whenever an employee record is updated.   
**b)** The SET command should be placed inside the BEGIN and END block.  
**c)** The trigger should be AFTER UPDATE instead of BEFORE UPDATE.  
**d)** NEW.LastModified should be replaced with OLD.LastModified to refer to the previous value.

**Q. 9**  
Which **SQL code snippet** demonstrates **correct use** of **foreign key constraints** to ensure **referential integrity** between Orders and Customers?

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

CustomerID INT,

OrderDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) ON DELETE CASCADE

);

**a)** This code correctly creates a foreign key with ON DELETE CASCADE to maintain referential integrity.   
**b)** The foreign key should not include the ON DELETE CASCADE clause.  
**c)** The foreign key constraint should reference the Orders table, not the Customers table.  
**d)** The CustomerID should not be a foreign key in the Orders table.

**Q. 10**  
Which **SQL code snippet** demonstrates **correct usage** of **DDL** commands to modify an existing table by **adding** a new column for Age and **setting a default value**?

ALTER TABLE Employees

ADD COLUMN Age INT DEFAULT 30;

**a)** This code correctly adds a column Age with a default value of 30 to the Employees table.   
**b)** The ALTER TABLE statement should be used with CREATE COLUMN, not ADD COLUMN.  
**c)** The DEFAULT value should be assigned after the column definition.  
**d)** The ALTER TABLE command cannot set default values for columns in MySQL.

**Q. 11**  
Which of the following is **NOT a feature** of **Relational Databases** as per **Codd's 12 Rules**?

**a)** Data should be represented in tables.  
**b)** Operations should be based on set theory.  
**c)** The DBMS should support hierarchical data storage.   
**d)** Every column in a table must contain atomic values.

**Q. 12**  
Which **SQL command** would you use to remove all rows from a table without removing the table structure in **MySQL**?

TRUNCATE TABLE Employees;

**a)** DELETE FROM Employees;  
**b)** DROP TABLE Employees;  
**c)** TRUNCATE TABLE Employees;   
**d)** REMOVE ALL FROM Employees;

**Q. 13**  
Which of the following **commands** will allow you to **change the name of a column** in an existing table in **MySQL**?

ALTER TABLE Students CHANGE COLUMN old\_name new\_name VARCHAR(255);

**a)** ALTER COLUMN  
**b)** MODIFY COLUMN  
**c)** CHANGE COLUMN   
**d)** RENAME COLUMN

**Q. 14**  
Consider the **Entity-Relationship Diagram (ERD)** where a Department has many Employees, and Employee works for one Department. Which relationship should be represented?

**a)** One-to-one  
**b)** One-to-many   
**c)** Many-to-many  
**d)** Many-to-one

**Q. 15**  
Which of the following is a **valid example of a NoSQL Database**?

**a)** MySQL  
**b)** PostgreSQL  
**c)** MongoDB   
**d)** Oracle

**Q. 16**  
In a **Conceptual Data Model**, which of the following is represented?

**a)** Detailed structure of data  
**b)** Logical structure and relationships among entities  
**c)** High-level representation of the business environment   
**d)** Implementation-specific constraints

**Q. 17**  
Which of the following **data models** is **NOT** one of the major **three types** of database models?

**a)** Relational model  
**b)** Hierarchical model  
**c)** Object-relational model  
**d)** Flat-file model

**Q. 18**  
Which **SQL command** is used to **create a new table** in **MySQL** with a **primary key** constraint?

CREATE TABLE Students (

StudentID INT PRIMARY KEY,

Name VARCHAR(100),

Age INT

);

**a)** INSERT INTO Students  
**b)** CREATE DATABASE Students  
**c)** CREATE TABLE Students   
**d)** ALTER TABLE Students ADD PRIMARY KEY (StudentID)

**Q. 19**  
Which of the following is **true about DML (Data Manipulation Language)** in **SQL**?

**a)** DML commands are used for defining database structures.  
**b)** DML commands include CREATE, ALTER, and DROP.  
**c)** DML commands are used for querying and modifying data in tables.   
**d)** DML commands affect the database schema.

**Q. 20**  
Which **SQL query** will **select all records** from the Customers table where the Country is either 'USA' or 'Canada'?

SELECT \* FROM Customers WHERE Country IN ('USA', 'Canada');

**a)** SELECT \* FROM Customers WHERE Country LIKE 'USA' OR Country LIKE 'Canada';  
**b)** SELECT \* FROM Customers WHERE Country = 'USA' AND Country = 'Canada';  
**c)** SELECT \* FROM Customers WHERE Country IN ('USA', 'Canada');   
**d)** SELECT \* FROM Customers WHERE Country = 'USA' OR Country = 'Canada';

**Q. 21**  
Which of the following is an example of **data redundancy** in a **relational database**?

**a)** Storing customer address in multiple tables without normalization.  
**b)** Using a primary key to enforce unique rows in a table.  
**c)** Using IN operator to filter records.  
**d)** Using DISTINCT to remove duplicate records.

**Q. 22**  
Which of the following **functional dependencies** is a violation of **3rd Normal Form (3NF)**?

**a)** EmployeeID → EmployeeName  
**b)** StudentID → CourseID  
**c)** StudentID → InstructorName (where InstructorName depends on CourseID) **(Correct)**  
**d)** StudentID → StudentName

**Q. 23**  
Which **normal form** does this table violate?

| **StudentID** | **CourseID** | **InstructorName** |
| --- | --- | --- |
| 1 | C101 | Prof. A |
| 1 | C102 | Prof. B |
| 2 | C101 | Prof. A |
| 2 | C103 | Prof. C |

**a)** 1st Normal Form (1NF)  
**b)** 2nd Normal Form (2NF)  
**c)** 3rd Normal Form (3NF) **(Correct)**  
**d)** Boyce-Codd Normal Form (BCNF)

**Q. 24**  
Which of the following **normal forms** removes **transitive dependencies**?

**a)** 1st Normal Form (1NF)  
**b)** 2nd Normal Form (2NF)  
**c)** 3rd Normal Form (3NF) **(Correct)**  
**d)** Boyce-Codd Normal Form (BCNF)

**Q. 25**  
What will this **SQL query** do?

INSERT INTO Students (StudentID, StudentName)

VALUES (1, 'John Doe');

**a)** Insert a new row with StudentID = 1 and StudentName = 'John Doe'.   
**b)** Update an existing row with StudentID = 1 to have StudentName = 'John Doe'.  
**c)** Delete the row where StudentID = 1.  
**d)** Return the number of rows in the Students table.

**Q. 26**  
What is the result of this **SQL query**?

UPDATE Students

SET StudentName = 'Jane Doe'

WHERE StudentID = 1;

**a)** It will insert a new record with StudentName = 'Jane Doe'.  
**b)** It will update the StudentName of the student with StudentID = 1 to 'Jane Doe'.   
**c)** It will delete the record where StudentID = 1.  
**d)** It will change all StudentName values to 'Jane Doe'.

**Q. 27**  
Which **MySQL data type** would you use to store **decimal numbers** with high precision?

**a)** INT  
**b)** FLOAT  
**c)** DECIMAL   
**d)** VARCHAR

**Q. 28**  
Which of the following is **NOT** a valid **SQL constraint** in MySQL?

**a)** PRIMARY KEY  
**b)** UNIQUE  
**c)** CHECK   
**d)** FOREIGN KEY

**Q. 29**  
What will the following **SQL query** do?

DELETE FROM Students WHERE StudentID = 1;

**a)** It will delete all rows in the Students table.  
**b)** It will delete the row where StudentID = 1.   
**c)** It will remove the column StudentID from the Students table.  
**d)** It will update the StudentName where StudentID = 1.

**Q. 30**  
What will be the **result** of the following SQL query?

SELECT COUNT(DISTINCT CourseID) FROM Enrollments;

**a)** Count of all CourseID values in the Enrollments table.  
**b)** Count of unique CourseID values in the Enrollments table.

**c)** Count of all rows in the Enrollments table.  
**d)** Sum of all CourseID values in the Enrollments table.

**Q. 31**  
Which **SQL function** will return the **average** value of a numeric column in a table?

**a)** MAX()  
**b)** SUM()  
**c)** AVG()   
**d)** COUNT()

**Q. 32**  
What is the correct **SQL syntax** to **group** records in the Sales table by the ProductID column?

SELECT ProductID, SUM(SaleAmount)

FROM Sales

GROUP BY ProductID;

**a)** GROUP BY groups the records based on the ProductID column.   
**b)** GROUP BY groups the records based on the SaleAmount column.  
**c)** GROUP BY is not valid in SQL.  
**d)** GROUP BY is only valid for non-numeric columns.

**Q. 33**  
What is the result of the following **SQL query**?

SELECT \* FROM Orders WHERE OrderDate BETWEEN '2022-01-01' AND '2022-12-31';

**a)** It will select orders placed before January 1, 2022.  
**b)** It will select orders placed after December 31, 2022.  
**c)** It will select orders placed between January 1, 2022, and December 31, 2022.   
**d)** It will select all orders from the Orders table.

**Q. 34**  
What does the **DISTINCT** keyword do in this SQL query?

SELECT DISTINCT Department FROM Employees;

**a)** Returns only the first department listed.  
**b)** Returns all departments, including duplicates.  
**c)** Returns unique department names, excluding duplicates.   
**d)** Returns only employees from one department.

**Q. 35**  
Which of the following **SQL operators** is used to test for **null values**?

**a)** BETWEEN  
**b)** LIKE  
**c)** IS NULL   
**d)** IN

**Q. 36**  
Which of the following **Relational Algebra operations** returns rows that satisfy a given condition?

**a)** Projection  
**b)** Selection   
**c)** Union  
**d)** Cartesian Product

**Q. 37**  
In a **Join operation**, which type of join only returns rows where there is a matching row in both tables?

**a)** Cross Join  
**b)** Outer Join  
**c)** Inner Join

**d)** Natural Join

**Q. 38**  
Which SQL query syntax will produce the **same result** for the following operation in **Relational Algebra**: A ∪ B?

SELECT column1 FROM A

UNION

SELECT column1 FROM B;

**a)** UNION only returns distinct rows.  
**b)** UNION ALL includes duplicate rows.  
**c)** INTERSECT returns common rows.  
**d)** MINUS returns rows in A but not in B.

**Q. 39**  
Given the **SQL query**:

SELECT \* FROM Employees

INNER JOIN Departments

ON Employees.DeptID = Departments.DeptID;

What type of **Join** is being used?

**a)** Equi Join  
**b)** Inner Join   
**c)** Outer Join  
**d)** Cross Join

**Q. 40**  
Which **SQL command** is used to automatically assign a unique value to a column when a new record is inserted?

**a)** SEQUENCE  
**b)** AUTO\_INCREMENT   
**c)** SERIAL  
**d)** IDENTITY