

Q1) What SQL statement would you use to create an SQL database named Test?

Answer:

c) CREATE DATABASE Test;

Q2) What SQL statement would you use to remove an SQL database named Test?

Answer:

a) DROP DATABASE Test;

Q3) Assign individual relationships that one can find in ERD (Entity Relationship Diagram) to their appropriate descriptions.

Answer:

a) 3) one to one

b) 5) one to many (mandatory)

c) 1) zero or many (optional)

d) 4) one and only one (mandatory)

e) 2) one or more (mandatory)

f) 7) zero or one (optional)

g) 6) many

Q4) Which kind of relationship best describes the relationship between Products (A) and OrderDetails (B) tables?

Answer:

a) One-to-Many

b) Many-to-One

c) Many-to-Many

d) One-to-One

Correct Answer:

a) One-to-Many

(One product can appear in many order details.)

Q5) Fulfill missing parts in the SQL statement below to create the Products table.

Answer:

sql

Copy

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

```
    ProductName VARCHAR(255),  
    SupplierID INT,  
    CategoryID INT,  
    Unit VARCHAR(50),  
    Price DECIMAL(10, 2)  
);
```

---

Q6) Primary Key is a combination of two types of constraints. Find the correct one in the list below.

Answer:

c) UNIQUE & NOT NULL

Q7) Which field in the Orders table has a role of FOREIGN KEY in relation to the Customers table?

Answer:

a) CustomerID

Q8) What SQL statement would you use to put a new record into the Orders table?

Answer:

a) INSERT INTO Orders (OrderID, CustomerID, OrderDate) VALUES (200, 125, '2018-02-05');

---

Q9) What SQL statement would you use to modify the existing record of CustomerID to 10 for OrderID #10308 in the Orders table?

Answer:

d) UPDATE Orders SET CustomerID = 10 WHERE OrderID = 10308;

Q10) What SQL statement would you use to delete the order with OrderID #10308 from the Orders table?

Answer:

a) DELETE FROM Orders WHERE OrderID = 10308;

---

Q11) What SQL statement would you use to add an Age field to the Customers table?

Answer:

b) ALTER TABLE Customers ADD Age INT;

Q12) What SQL statement would you use to remove the Customers table from the Test database?

Answer:

d) DROP TABLE Customers;

Q13) Fulfill missing parts in the SQL statement below to extract all records from the Customers table.

Answer:

sql

Copy

```
SELECT * FROM Customers;
```

Q14) Fulfill missing parts in the SQL statement below to extract CustomerName and Address from the Customers table.

Answer:

sql

Copy

```
SELECT CustomerName, Address FROM Customers;
```

Q15) Fulfill missing parts in the SQL statement below to extract all distinct countries from the Customers table.

Answer:

sql

Copy

```
SELECT DISTINCT Country FROM Customers;
```

Q16) Fulfill missing parts in the SQL statement below to extract all records from the Products table that will include only products with a price higher than 20 EUR.

Answer:

sql

Copy

```
SELECT * FROM Products WHERE Price > 20;
```

Q17) Fulfill missing parts in the SQL statement below to extract all records from the Customers table that will include only those customers who have NULL values in the Address field.

Answer:

sql

Copy

```
SELECT * FROM Customers WHERE Address IS NULL;
```

Q18) Fulfill missing parts in the SQL statement below to extract all records from the Customers table that will include only those customers who are from Germany or the UK.

Answer:

sql

Copy

```
SELECT * FROM Customers WHERE Country IN ('Germany', 'UK');
```

Q19) Fulfill missing parts in the SQL statement below to extract all records from the Customers table that will include only those customers who are not from the USA.

Answer:

sql

Copy

```
SELECT * FROM Customers WHERE Country <> 'USA';
```

Q20) Fulfill missing parts in the SQL statement below to extract all records from the Products table that will include only those products that are supplied by supplier with SupplierID #1 and that belong to CategoryID #2.

Answer:

sql

Copy

```
SELECT * FROM Products WHERE SupplierID = 1 AND CategoryID = 2;
```

---

Q21) Fulfill missing parts in the SQL statement below to arrange records in the Products table according to Price in descending order.

Answer:

sql

Copy

```
SELECT * FROM Products ORDER BY Price DESC;
```

Q22) Fulfill missing parts in the SQL statement below to extract the first 50 records from the Customers table.

Answer:

sql

Copy

```
SELECT TOP 50 * FROM Customers;
```

(For MySQL, use LIMIT 50 instead of TOP 50.)

---

Q23) Fulfill missing parts in the SQL statement below to find the maximum Price for products listed in the Products table.

Answer:

sql

Copy

```
SELECT MAX(Price) FROM Products;
```

---

Q24) What statement will you use to count the number of records within the Customers table?

Answer:

c) SELECT COUNT(\*) FROM Customers;

---

Q25) Fulfill missing parts in the SQL statement below to find the average Price for products listed in the Products table.

Answer:

sql

Copy

```
SELECT AVG(Price) FROM Products;
```

---

Q26) Fulfill missing parts in the SQL statement below to find the overall number of ordered products using the Quantity field in the OrderDetails table.

Answer:

sql

Copy

```
SELECT SUM(Quantity) FROM OrderDetails;
```

---

Q27) Fulfill missing parts in the SQL statement below to find all customers listed in the Customers table whose name starts with the letter "b".

Answer:

sql

Copy

```
SELECT * FROM Customers WHERE CustomerName LIKE 'b%';
```

---

Q28) Fulfill missing parts in the SQL statement below to find all customers listed in the Customers table whose name starts with the letter "b" and ends with the letter "o".

Answer:

sql

Copy

```
SELECT * FROM Customers WHERE CustomerName LIKE 'b%o';
```

---

Q29) Fulfill missing parts in the SQL statement below to find all customers listed in the Customers table whose name has the letter "b" in the second position.

Answer:

sql

Copy

```
SELECT * FROM Customers WHERE CustomerName LIKE '_b%';
```

---

Q30) Fulfill missing parts in the SQL statement below to find all customers listed in the Customers table who live in Germany, the UK, and the USA.

Answer:

sql

Copy

```
SELECT * FROM Customers WHERE Country IN ('Germany', 'UK', 'USA');
```

---

Q31) Fulfill missing parts in the SQL statement below to find all products listed in the Products table whose price belongs to the range from 5 to 25 EUR, including the begin and end values.

Answer:

sql

Copy

```
SELECT * FROM Products WHERE Price BETWEEN 5 AND 25;
```

---

Q32) What statement would you use to temporarily change the name of the CustomerName field to Customer within the Customer table?

Answer:

a) SELECT CustomerName AS Customer FROM Customers;

---

Q33) Fulfill missing parts in the SQL statement below to select all orders with existing customer information.

Answer:

sql

Copy

```
SELECT Orders.OrderID, Customers.CustomerName
```

```
FROM Orders
```

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

---

Q34) Fulfill missing parts in the SQL statement below to select all customers and any orders they might have.

Answer:

sql

Copy

```
SELECT Customers.CustomerName, Orders.OrderID
```

```
FROM Customers
```

```
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

---

Q35) Fulfill missing parts in the SQL statement below to select all customers and any orders they might have.

Answer:

sql

Copy

```
SELECT Customers.CustomerName, Orders.OrderID
```

```
FROM Customers
```

```
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

---

Q36) Fulfill missing parts in the SQL statement below to select all customers and all orders.



Answer:

sql

Copy

```
SELECT Customers.CustomerName, Orders.OrderID
```

```
FROM Customers
```

```
FULL OUTER JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

---

Q37) What operator would you use to merge selects from two different tables with the same number of columns in the same order and with similar data types?

Answer:

d) UNION

---

Q38) Fulfill missing parts in the SQL statement below to calculate the overall Quantity for each ProductID and arrange the resulting list in descending order according to this new metric.

Answer:

sql

Copy

```
SELECT ProductID, SUM(Quantity) AS Overall_Quantity
```

```
FROM OrderDetails
```

```
GROUP BY ProductID
```

```
ORDER BY Overall_Quantity DESC;
```

---

Q39) Fulfill missing parts in the SQL statement below to filter products whose overall Quantity is higher than 100 and arrange the resulting list in descending order according to the overall Quantity.

Answer:

sql

Copy

```
SELECT ProductID, SUM(Quantity) AS Overall_Quantity
```

```
FROM OrderDetails  
  
GROUP BY ProductID  
  
HAVING SUM(Quantity) > 100  
  
ORDER BY Overall_Quantity DESC;
```

---

Q40) Fulfill missing parts in the SQL statement below to create a new field that will classify products listed in the Products table as "Cheap" when their Price is lower than 10 EUR or as "Expensive" otherwise.

Answer:

sql

Copy

```
SELECT ProductID, Price,  
  
    CASE  
  
        WHEN Price < 10 THEN 'Cheap'  
  
        ELSE 'Expensive'  
  
    END AS Price_Level  
  
FROM Products;
```