Back to Previous Page

Quiz Result For "Database Quiz" Is Here -

Total Question - 17

Total Attempts - 17

Total Correct - 16

Total Incorrect - 1

Duration - 17Minutes

Passing Percentage - 70%

Your Percentage - 94.12%

Q1. Which of the following SQL statements correctly adds a new column named <code>discount</code> to the <code>products</code> table with a numeric data type?

1)

```
ALTER TABLE products

ADD COLUMN discount INT;
```

2)

```
ADD COLUMN discount TO products NUMERIC;
```

3)

```
UPDATE products

SET discount = 0.1;
```

4)

```
INSERT INTO products (discount) VALUES (0.1);
```

Your Answer: 1) ✔

```
ALTER TABLE products

ADD COLUMN discount INT;
```

The answer appears to be incorrect because it calls for INT not NUMERIC

Correct Answer: 1)

```
ALTER TABLE products

ADD COLUMN discount INT;
```

Q2. What does this query do?

1) Deletes all customers.

2) Removes customers registered after '2023-01-01'.

3) Deletes customers registered before '2023-01-01'.

4) Updates the registration date for all customers.

Your Answer: 3) ✔

Deletes customers registered before '2023-01-01'.

Correct Answer: 3)

Deletes customers registered before '2023-01-01'.

Q3. In <b>SQL</b>, what is the purpose of the <code>LIKE</code> operator?

1) Performs mathematical operations.

2) Combines rows from different tables.

3) Compares two columns for equality.

4) Filters data based on a pattern.

Your Answer: 4) ✔

Filters data based on a pattern.

Correct Answer: 4)

Filters data based on a pattern.

Q4. Which of the following SQL JOINs retrieves all records from the <code>orders</code> table and the matching records from the <code>customers</code> table?

1)

```
SELECT *

FROM orders

LEFT JOIN customers ON orders.customer_id = customers.customer_id;
```

2)

```
SELECT *

FROM orders

INNER JOIN customers ON orders.customer_id = customers.customer_id;
```

3)

```
SELECT *

FROM orders

RIGHT JOIN customers ON orders.customer_id = customers.customer_id;
```

4)

```
SELECT *

FROM orders

FULL JOIN customers ON orders.
```

Your Answer: 2) ✔

```
SELECT *

FROM orders

INNER JOIN customers ON orders.customer_id = customers.customer_id;
```

Correct Answer: 2)

```
SELECT *

FROM orders

INNER JOIN customers ON orders.customer_id = customers.customer_id;
```

Q5. What does the following **SQL** query do?

1) Deletes all employees hired before '2022-01-01'.

2) Deletes employees in department 3 hired before '2022-01-01'.

3) Deletes all employees in department 3.

4) Deletes employees hired before '2022-01-01' in any department.

Your Answer: 2) ✔

Deletes employees in department 3 hired before '2022-01-01'.

Correct Answer: 2)

Deletes employees in department 3 hired before '2022-01-01'.

Q6. What is the purpose of the following SQL query?

1) Retrieves the average salary for all employees.

2) Calculates the average salary for employees in department 2.

3) Updates the salary of employees in department 2.

4) Deletes employees with a salary less than the average salary.

Your Answer: 2) ✔

Calculates the average salary for employees in department 2.

Correct Answer: 2)

Calculates the average salary for employees in department 2.

Q7. What does this query retrieve?

1) Products with names starting with 'C'.

2) Products with names containing 'C'.

3) Products with names ending with 'C'.

4) Products with names exactly 'C'.

Your Answer: 1) ✔

Products with names starting with 'C'.

Correct Answer: 1)

Products with names starting with 'C'.

Q8. Which of the following SQL queries retrieves the top 5 highest-priced products in the <code>electronics</code> category, ordered from highest to lowest price?

1)

```
SELECT product_name, price

FROM products

WHERE category = 'Electronics'

ORDER BY price DESC

LIMIT 5;
```

2)

```
SELECT product_name, price

FROM products

WHERE category = 'Electronics'

ORDER BY price ASC

LIMIT 5;
```

3)

```
SELECT product_name, price

FROM products

WHERE category = 'Electronics'

ORDER BY price DESC

OFFSET 5;
```

4)

```
SELECT product_name, price

FROM products

WHERE category = 'Electronics'

ORDER BY price ASC

OFFSET 5;
```

Your Answer: 1) ✔

```
SELECT product_name, price

FROM products

WHERE category = 'Electronics'

ORDER BY price DESC

LIMIT 5;
```

Correct Answer: 1)

```
SELECT product_name, price

FROM products

WHERE category = 'Electronics'

ORDER BY price DESC

LIMIT 5;
```

Q9. Which SQL statement correctly inserts three records into the <code>courses</code> table with sample data?

1)

```
INSERT INTO courses (course_name, credit_hours)

VALUES
 ('Mathematics', 3),
 ('History', 4),
 ('Computer Science', 5);
```

2)

```
INSERT INTO courses (name, credit_hours)

VALUES
 ('Mathematics', 3),
 ('History', 4),
 ('Computer Science', 5);
```

3)

```
INSERT INTO courses (id, course_name, credit_hours)

VALUES
 (1, 'Mathematics', 3),
 (2, 'History', 4),
 (3, 'Computer Science', 5);
```

4)

```
INSERT INTO courses (course_id, name, credit_hours)

VALUES
 (1, 'Mathematics', 3),
 (2, 'History', 4),
 (3, 'Computer Science', 5);
```

Your Answer: 1) ✔

```
INSERT INTO courses (course_name, credit_hours)

VALUES
 ('Mathematics', 3),
 ('History', 4),
 ('Computer Science', 5);
```

Correct Answer: 1)

```
INSERT INTO courses (course_name, credit_hours)

VALUES
 ('Mathematics', 3),
 ('History', 4),
 ('Computer Science', 5);
```

Q10. What does this query do?

1) Adds 10 to the stock quantity of all electronic products.

2) Decreases the stock quantity by 10 for all products.

3) Sets the stock quantity to 10 for all electronic products.

4) Updates the category of electronic products.

Your Answer: 4) ✗

Updates the category of electronic products.

Correct Answer: 2)

Decreases the stock quantity by 10 for all products.

Both of these answers appear to be incorrect because it only reduces the stock quantity for electronic products by 10. I put each of these on my . two tries

Q11. What does the following SQL query do?

1) Retrieves the highest salary for each department where the maximum salary is over 50,000.

2) Updates the salary of employees to 50,000.

3) Inserts a new employee with a salary of 50,000.

4) Deletes employees with a salary below 50,000.

Your Answer: 1) ✔

Retrieves the highest salary for each department where the maximum salary is over 50,000.

Correct Answer: 1)

Retrieves the highest salary for each department where the maximum salary is over 50,000.

Q12. Retrieve the names of products in the <code>Electronics</code> category with a price greater than $500.

1)

```
SELECT product_name

FROM products

WHERE category = 'Electronics' AND price > 500;
```

2)

```
SELECT product_name

FROM products

WHERE category = 'Electronics' OR price > 500;
```

3)

```
SELECT product_name

FROM products

WHERE category = 'Electronics' AND price < 500;
```

4)

```
SELECT product_name

FROM products

WHERE category = 'Electronics' OR price < 500;
```

Your Answer: 1) ✔

```
SELECT product_name

FROM products

WHERE category = 'Electronics' AND price > 500;
```

Correct Answer: 1)

```
SELECT product_name

FROM products

WHERE category = 'Electronics' AND price > 500;
```

Q13. Retrieve the product names and quantities for orders placed by customers with IDs 1, 2, and 3.

1)

```
SELECT product_name, quantity

FROM orders

JOIN order_details ON orders.order_id = order_details.order_id

JOIN products ON order_details.product_id = products.product_id

WHERE customers.customer_id IN (1, 2, 3);
```

2)

```
SELECT product_name, quantity

FROM orders

JOIN order_details ON orders.order_id = order_details.order_id

JOIN products ON order_details.product_id = products.product_id

WHERE customers.customer_id = 1 OR customers.customer_id = 2 OR customers.customer_id = 3;
```

3)

```
SELECT product_name, quantity

FROM orders

JOIN order_details ON orders.order_id = order_details.order_id

JOIN products ON order_details.product_id = products.product_id

WHERE customers.customer_id = 1 AND customers.customer_id = 2 AND customers.customer_id = 3;
```

4)

```
SELECT product_name, quantity

FROM orders

JOIN order_details ON orders.order_id = order_details.order_id

JOIN products ON order_details.product_id = products.product_id

WHERE customers.customer_id = 1 AND customers.customer_id = 2 OR customers.customer_id = 3;
```

Your Answer: 1) ✔

```
SELECT product_name, quantity

FROM orders

JOIN order_details ON orders.order_id = order_details.order_id

JOIN products ON order_details.product_id = products.product_id

WHERE customers.customer_id IN (1, 2, 3);
```

Correct Answer: 1)

```
SELECT product_name, quantity

FROM orders

JOIN order_details ON orders.order_id = order_details.order_id

JOIN products ON order_details.product_id = products.product_id

WHERE customers.customer_id IN (1, 2, 3);
```

Q14. Consider the following table structures: Which SQL statement retrieves the names of authors along with the titles of their books?

1)

```
SELECT author_name, book_title

FROM authors

JOIN books ON authors.author_id = books.author_id;
```

2)

```
SELECT author_name, book_title

FROM authors

JOIN books ON authors.author_id = books.author_id

GROUP BY author_name;
```

3)

```
SELECT author_name, book_title

FROM authors

LEFT JOIN books ON authors.author_id = books.author_id;
```

4)

```
SELECT author_name, book_title

FROM authors

RIGHT JOIN books ON authors.author_id = books.
```

Your Answer: 1) ✔

```
SELECT author_name, book_title

FROM authors

JOIN books ON authors.author_id = books.author_id;
```

Correct Answer: 1)

```
SELECT author_name, book_title

FROM authors

JOIN books ON authors.author_id = books.author_id;
```

Q15. Consider the following table structures: Which SQL statement retrieves the names of students along with the names of courses they are enrolled in?

1)

```
SELECT student_name, course_name

FROM students

JOIN student_courses ON students.student_id = student_courses.student_id

JOIN courses ON student_courses.course_id = courses.course_id;
```

2)

```
SELECT student_name, course_name

FROM students

LEFT JOIN student_courses ON students.student_id = student_courses.student_id

JOIN courses ON student_courses.course_id = courses.course_id;
```

3)

```
SELECT student_name, course_name

FROM students

RIGHT JOIN student_courses ON students.student_id = student_courses.student_id

JOIN courses ON student_courses.course_id = courses.course_id;
```

4)

```
SELECT student_name, course_name

FROM students

JOIN student_courses ON students.student_id = student_courses.student_id

LEFT JOIN courses ON student_courses.course_id = courses.course_id;
```

Your Answer: 1) ✔

```
SELECT student_name, course_name

FROM students

JOIN student_courses ON students.student_id = student_courses.student_id

JOIN courses ON student_courses.course_id = courses.course_id;
```

Correct Answer: 1)

```
SELECT student_name, course_name

FROM students

JOIN student_courses ON students.student_id = student_courses.student_id

JOIN courses ON student_courses.course_id = courses.course_id;
```

Q16. Consider the following table structures: Retrieve the names of customers who have placed at least two orders in the year 2023.

1)

```
SELECT customer_name

FROM customers

JOIN orders ON customers.customer_id = orders.customer_id

WHERE EXTRACT(YEAR FROM order_date) = 2023

GROUP BY customer_name

HAVING COUNT(order_id) >= 2;
```

2)

```
SELECT customer_name

FROM customers

LEFT JOIN orders ON customers.customer_id = orders.customer_id

WHERE EXTRACT(YEAR FROM order_date) = 2023

GROUP BY customer_name

HAVING COUNT(order_id) >= 2;
```

3)

```
SELECT customer_name

FROM customers

JOIN orders ON customers.customer_id = orders.customer_id

WHERE EXTRACT(YEAR FROM order_date) = 2023

GROUP BY customer_name

HAVING COUNT(*) >= 2;
```

4)

```
SELECT customer_name

FROM customers

JOIN orders ON customers.customer_id = orders.customer_id

WHERE EXTRACT(YEAR FROM order_date) = 2023

GROUP BY customer_name

HAVING COUNT(1) >= 2;
```

Your Answer: 1) ✔

```
SELECT customer_name

FROM customers

JOIN orders ON customers.customer_id = orders.customer_id

WHERE EXTRACT(YEAR FROM order_date) = 2023

GROUP BY customer_name

HAVING COUNT(order_id) >= 2;
```

Correct Answer: 1)

```
SELECT customer_name

FROM customers

JOIN orders ON customers.customer_id = orders.customer_id

WHERE EXTRACT(YEAR FROM order_date) = 2023

GROUP BY customer_name

HAVING COUNT(order_id) >= 2;
```

Q17. Consider a table named <code>products</code> with columns (product_id, product_name, price). Retrieve the names of products, sorted by price in descending order, skipping the first 5 records.

1)

```
SELECT product_name

FROM products

ORDER BY price DESC

OFFSET 5;
```

2)

```
SELECT product_name

FROM products

ORDER BY price ASC

OFFSET 5;
```

3)

```
SELECT product_name

FROM products

ORDER BY price DESC

OFFSET 0;
```

4)

```
SELECT product_name

FROM products

ORDER BY price ASC

OFFSET 0;
```

Your Answer: 1) ✔

```
SELECT product_name

FROM products

ORDER BY price DESC

OFFSET 5;
```

Correct Answer: 1)

```
SELECT product_name

FROM products

ORDER BY price DESC

OFFSET 5;
```