SQL and NoSQL Reassessment BC-101

Points: 18/25

✓ **Correct** 1/1 Points

1. Which ACID property ensures that a transaction is treated as a single unit of work and is either fully completed or fully rolled back in case of failure?

- (●) Atomicity
- ( ) Consistency
- ( ) Isolation
- ( ) Durability

✓ **Correct** 1/1 Points

2. You want to create a view in your database to display only the names and salaries of employees in the "Sales" department. Which SQL statement should you use?

- (●) CREATE VIEW SalesEmployees AS SELECT name, salary FROM Employees WHERE department = 'Sales';
- ( ) CREATE VIEW SalesEmployees AS SELECT * FROM Employees WHERE department = 'Sales';

- CREATE VIEW SalesEmployees AS SELECT name, salary FROM Employees;
- CREATE VIEW SalesEmployees AS SELECT name, salary FROM Employees HAVING department = 'Sales';

✓ **Correct** 1/1 Points

3. You have a database with a products table and a suppliers table. You want to find the names of products that are supplied by "Supplier ABC." Which SQL query is appropriate?

- SELECT product_name FROM products WHERE supplier_id = (SELECT supplier_id FROM suppliers WHERE supplier_name = 'Supplier ABC');
- SELECT product_name FROM products WHERE product_id IN (SELECT product_id FROM suppliers WHERE supplier_name = 'Supplier ABC');
- SELECT product_name FROM products WHERE supplier_id IN (SELECT supplier_id FROM suppliers WHERE supplier_name = 'Supplier ABC');
- SELECT product_name FROM suppliers WHERE supplier_name = 'Supplier ABC';

✓ **Correct** 1/1 Points

4. In a university database, you have a students table and a grades table. You want to find the names of students who have a GPA higher than 3.0. Which SQL query should you use?

- SELECT student_name FROM students WHERE gpa > (SELECT MAX(gpa) FROM students);
- SELECT student_name FROM students WHERE student_id IN (SELECT student_id FROM grades WHERE gpa > 3.0);
- SELECT student_name FROM students WHERE gpa = (SELECT AVG(gpa) FROM students);
- SELECT student_name FROM students WHERE student_id = (SELECT student_id FROM grades WHERE gpa > 3.0);

✓ **Correct** 1/1 Points

5. You have two tables, "employees" and "departments," with the following structures:

   employees table:

   • employee_id (unique employee identifier)

   • employee_name (name of the employee)

   • department_id (identifier indicating which department the employee belongs to)

   departments table:

   • department_id (unique department identifier)

   • department_name (name of the department)

   • location (location of the department)

   You want to find all employees who belong to the "Marketing" department. Which SQL query should you use?

   - (●) SELECT employee_name FROM employees INNER JOIN departments ON employees.department_id = departments.department_id WHERE department_name = 'Marketing';
   - ( ) SELECT employee_name FROM employees LEFT JOIN departments ON employees.department_id = departments.department_id WHERE department_name = 'Marketing';
   - ( ) SELECT employee_name FROM employees RIGHT JOIN departments ON employees.department_id = departments.department_id WHERE department_name = 'Marketing';
   - ( ) SELECT employee_name FROM employees FULL JOIN departments ON employees.department_id = departments.department_id WHERE department_name = 'Marketing';

✓ **Correct** 1/1 Points

6. Which of the following statement is TRUE about LEFT JOIN created on two tables Table1 and Table2?

- (●) Retrieves all the unmatched rows of Table1
- ( ) Retrieves all the unmatched rows of Table2
- ( ) Retrieves both matched and unmatched rows of Table1 and Table2
- ( ) Retrieves only matched rows of Table1 and Table2

✗ **Incorrect** 0/1 Points

7. You have a table named employees with the following columns: employee_id, employee_name, department_id, and salary. You want to find the total salary expenses for each department. Which SQL query should you use?

- ( ) SELECT department_id, SUM(salary) FROM employees;
- ( ) SELECT department_id, SUM(salary) FROM employees GROUP BY employee_id;
- (●) SELECT department_id, AVG(salary) FROM employees GROUP BY department_id;
- ( ) SELECT department_id, COUNT(employee_id) FROM employees GROUP BY department_id;

✓ **Correct** 1/1 Points

8. You have a table named products with the following columns: product_id, category_id, and quantity_in_stock. You want to find the highest and lowest stock quantities for each product category. Which SQL query should you use?

- ( ) SELECT category_id, MAX(quantity_in_stock), MIN(quantity_in_stock) FROM products;

- (●) SELECT category_id, MAX(quantity_in_stock), MIN(quantity_in_stock) FROM products GROUP BY category_id;
- ( ) SELECT category_id, AVG(quantity_in_stock) FROM products GROUP BY category_id;
- ( ) SELECT category_id, COUNT(product_id) FROM products GROUP BY category_id;

✕ **Incorrect** 0/1 Points

9. Which of the following statements is true about subqueries in SQL?

- (●) Subqueries are always independent and never rely on other queries.
- ( ) Subqueries are used to retrieve data from a single table only.
- ( ) Subqueries can be used in SELECT, FROM, and WHERE clauses.
- ( ) Subqueries can only be used in conjunction with stored procedures.

✕ **Incorrect** 0/1 Points

10. Consider a database with two tables, employees and departments. Each employee has a department_id indicating the department they belong to. You want to find the names of employees who belong to the "Sales" department. Which SQL query would you use?

- [ ] SELECT employee_name FROM employees WHERE department_id IN (SELECT department_id FROM departments WHERE department_name = 'Sales');
- [✓] SELECT employee_name FROM employees JOIN departments ON employees.department_id = departments.department_id WHERE departments.department_name = 'Sales';
- [ ] SELECT employee_name FROM employees WHERE EXISTS (SELECT department_id FROM departments WHERE department_name = 'Sales');

- [x] SELECT employee_name FROM employees WHERE department_id = (SELECT department_id FROM departments WHERE department_name = 'Sales');

✓ **Correct** 1/1 Points

11. Which type of SQL join returns all rows from the left table and the matching rows from the right table, filling in with NULL values if there is no match?

- ( ) INNER JOIN
- (•) LEFT JOIN
- ( ) RIGHT JOIN
- ( ) FULL OUTER JOIN

✓ **Correct** 1/1 Points

12. What SQL statement retrieves the product names and their prices from a "products" table?

- (•) SELECT product_name, price FROM products;
- ( ) SELECT product_id, category FROM products;
- ( ) SELECT category, quantity_in_stock FROM products;
- ( ) SELECT product_name, quantity_in_stock FROM products;

✗ **Incorrect** 0/1 Points

13. What SQL query returns the total quantity of each product in stock?

    You have a "products" table with columns: product_id, product_name, price, stock_quantity.

- [x] SELECT SUM(stock_quantity) FROM products GROUP BY product_id;
- [ ] SELECT product_id, SUM(stock_quantity) FROM products;
- [ ] SELECT product_id, SUM(stock_quantity) FROM products GROUP BY product_name;
- [ ] SELECT AVG(stock_quantity) FROM products GROUP BY product_id;

✗ **Incorrect** 0/1 Points

14. You have a "sales" table with columns: sale_id, product_id, quantity_sold, sale_date. How would you find products that were sold between 2022-01-01 and 2022-12-31?

- [x] SELECT product_id, quantity_sold FROM sales WHERE sale_date BETWEEN '2022-01-01' AND '2022-12-31';
- [ ] SELECT product_id, quantity_sold FROM sales WHERE sale_date >= '2022-01-01' AND sale_date <= '2022-12-31';
- [x] SELECT product_id, quantity_sold FROM sales WHERE sale_date > '2022-01-01' AND sale_date < '2022-12-31';
- [ ] SELECT product_id, quantity_sold FROM sales WHERE sale_date NOT BETWEEN '2022-01-01' AND '2022-12-31';

✓ **Correct** 1/1 Points

15. You have a table named teachers with the following columns: teacher_id, teacher_name, subject_id, and teaching_experience. You want to find the average teaching experience for each teacher in each subject. Which SQL query should you use?

- [ ] SELECT teacher_id, AVG(teaching_experience) FROM teachers;
- [ ] SELECT teacher_id, AVG(teaching_experience) FROM teachers GROUP BY teacher_id;
- [ ] SELECT subject_id, AVG(teaching_experience) FROM teachers GROUP BY subject_id;

- SELECT teacher_id, subject_id, AVG(teaching_experience) FROM teachers GROUP BY teacher_id, subject_id;

✓ **Correct** 1/1 Points

16. You have two tables, "orders" and "customers," with the following structures:

orders table:

• order_id (unique order identifier)

• customer_id (customer identifier)

• order_date (date when the order was placed)

customers table:

• customer_id (customer identifier)

• customer_name (name of the customer)

• city (city where the customer resides)

You want to retrieve a list of customers who have not placed any orders. Which SQL query should you use?

- SELECT customer_name FROM customers LEFT JOIN orders ON customers.customer_id = orders.customer_id WHERE order_id IS NULL;
- SELECT customer_name FROM customers INNER JOIN orders ON customers.customer_id = orders.customer_id WHERE order_id IS NULL;
- SELECT customer_name FROM customers RIGHT JOIN orders ON customers.customer_id = orders.customer_id WHERE order_id IS NULL;
- SELECT customer_name FROM customers FULL JOIN orders ON customers.customer_id = orders.customer_id WHERE order_id IS NULL;

✓ **Correct** 1/1 Points

17. Which join is to be used between two tables A and B when the resultant table needs rows from A and B that match the condition and rows from A that do not match the condition?

- ○ Outer Join
- ○ Cross Join
- ○ Inner Join
- ◉ None of the above

✓ **Correct** 1/1 Points

18. When using the GROUP BY clause, which clause can be used to filter the grouped rows based on a condition?

- ◉ GROUP HAVING
- ○ GROUP WHERE
- ○ GROUP FILTER
- ○ GROUP BY

✗ **Incorrect** 0/1 Points

19. In an SQL JOIN operation, what is the common criteria used to connect rows from two tables?

- ○ Row number
- ◉ Primary key

- Index
- Column values

✓ **Correct** 1/1 Points

20. You have a "products" table with columns: product_id, product_name, category, quantity_in_stock. How would you find products that belong to the "Electronics" category and have a quantity in stock greater than 10?

- SELECT product_name FROM products WHERE category = 'Electronics' AND quantity_in_stock > 10;
- SELECT product_name FROM products WHERE category = 'Electronics' OR quantity_in_stock > 10;
- SELECT product_name FROM products WHERE category = 'Electronics' AND quantity_in_stock < 10;
- SELECT product_name FROM products WHERE category = 'Electronics' OR quantity_in_stock < 10;

✓ **Correct** 1/1 Points

21. You have a "courses" table with columns: course_id, course_name, and instructor_id. What SQL query lists the unique instructor IDs?

- SELECT DISTINCT course_name FROM courses;
- SELECT DISTINCT instructor_id FROM courses;
- SELECT DISTINCT course_id FROM courses WHERE instructor_id = 'instructor_id';
- SELECT DISTINCT instructor_id FROM courses WHERE course_name = 'course_name';

✗ **Incorrect** 0/1 Points

22. How can you find the employees who were hired before the year 2020?

You have an "employees" table with columns: employee_id, first_name, last_name, hire_date.

- ○ SELECT * FROM employees WHERE YEAR(hire_date) < 2020;
- ◉ SELECT * FROM employees WHERE hire_date < '2020-01-01';
- ○ SELECT * FROM employees WHERE hire_date <= '2020-01-01';
- ○ SELECT * FROM employees WHERE hire_date BETWEEN '2020-01-01' AND '2023-12-31';

✓ **Correct** 1/1 Points

23. What is an advantage of using indexes in a database?

- ○ Indexes reduce the storage space required for the database.
- ○ Indexes enforce data integrity constraints.
- ○ Indexes speed up data insertion operations.
- ◉ Indexes allow for faster data retrieval for specific queries.

✓ **Correct** 1/1 Points

24. What is a super key in a database?

- ○ A key that is used for authentication purposes.
- ○ A key that is used for data encryption.
- ◉ A key that uniquely identifies each record in the table.

- ○ None of the above

✓ **Correct** 1/1 Points

25. You have a view named "HighSalaryEmployees" that displays employees earning more than $50,000 per year. What SQL statement should you use to delete this view?

- ○ DROP TABLE HighSalaryEmployees;
- ○ DELETE VIEW HighSalaryEmployees;
- ○ REMOVE VIEW HighSalaryEmployees;
- ◉ DROP VIEW IF EXISTS HighSalaryEmployees;