Data Retrieval with SELECT Statement

Concept-Based

Q1. What is the purpose of the SELECT statement?
A1. To retrieve data from one or more tables.

Q2. What does SELECT * mean?
A2. It selects all columns from the specified table(s).

Q3. What is the difference between DISTINCT and ALL?
A3. DISTINCT returns unique values, ALL (default) includes duplicates.

Q4. Can we use expressions in SELECT statements?
A4. Yes, arithmetic and string operations are allowed.

Q5. What is a column alias?
A5. A temporary name given to a column using the AS keyword.

Code-Based

Q6. Select all data from the `` table.**

```
SELECT * FROM employees;
```

Q7. Select ** and ** from the `` table.**

```
SELECT first_name, salary FROM employees;
```

Q8. Display full names as ``.

```
SELECT CONCAT(first_name, ' ', last_name) AS 'Employee Name' FROM employees;
```

Q9. Add 1000 to each employee's salary in the output.

```
SELECT first_name, salary + 1000 AS updated_salary FROM employees;
```

Q10. Select unique job titles from the employees table.

```
SELECT DISTINCT job_title FROM employees;
```

Scenario-Based

Q11. How would you retrieve all employee names and add a title "Mr./Ms." depending on gender?

```
SELECT
  CASE gender
    WHEN 'M' THEN CONCAT('Mr. ', first_name)
    ELSE CONCAT('Ms. ', first_name)
  END AS titled_name
FROM employees;
```

Filtering and Sorting Data

Concept-Based

Q1. What is the WHERE clause used for?
A1. To filter rows based on specific conditions.

Q2. What operators are used in WHERE clause?
A2. =, <>, >, <, >=, <=, BETWEEN, LIKE, IN, IS NULL, AND, OR, NOT.

Q3. What does ORDER BY do?
A3. It sorts the result set in ascending (ASC) or descending (DESC) order.

Q4. What is the difference between WHERE and HAVING?
A4. WHERE filters rows before grouping, HAVING filters groups.

Q5. Can we filter data based on pattern matching?
A5. Yes, using the LIKE operator with wildcards (% and _).

Code-Based

Q6. Select employees with salary > 50000.

```
SELECT * FROM employees WHERE salary > 50000;
```

Q7. List employees from 'New York' city.

```
SELECT * FROM employees WHERE city = 'New York';
```

Q8. Find employees whose names start with 'J'.

```
SELECT * FROM employees WHERE first_name LIKE 'J%';
```

Q9. Display employees ordered by salary descending.

```
SELECT * FROM employees ORDER BY salary DESC;
```

Q10. Retrieve employees not from the 'HR' or 'IT' departments.

```
SELECT * FROM employees WHERE department NOT IN ('HR', 'IT');
```

Scenario-Based

Q11. How would you get the top 3 highest paid employees?

```
SELECT * FROM employees ORDER BY salary DESC LIMIT 3;
```