Question 1

Correct

Mark 1.00 out of 1.00

Flag question

The NOT operator can be used with _____ operators.

Select one or more:

- [x] a. ALL ✔
- [x] b. ANY ✔
- [x] c. IN ✔
- [ ] d. >

Question 2

Correct

Mark 1.00 out of 1.00

Flag question

```
SELECT last_name, first_name
FROM employee
WHERE salary IN
(SELECT salary
FROM employee
WHERE dept_no = 3 OR dept_no = 5);
```

Which values are displayed?

Select one:

- (●) a. last name and first name of all employees with the same salary as employees in department 3 or 5 ✔
- ( ) b. last name and first name of only the employees whose salary falls in the range of salaries from department 3 or 5
- ( ) c. last name and first name of only the employees in department number 3 or 5
- ( ) d. last name and first name of all employees except those working in department 3 or 5

Question 4

Correct

Mark 1.00 out of 1.00

Flag question

Which operator is NOT appropriate in the join condition of a non-equi join SELECT statement?

Select one:

- (●) a. equal operator ✔
- ( ) b. BETWEEN x AND y operator
- ( ) c. IN operator
- ( ) d. greater than or equal to operator
- ( ) e. LIKE operator

Question 5

Correct

Mark 1.00 out of 1.00

Flag question

In which two cases would you use the USING clause? (Choose two)

Select one or more:

- [ ] a. You want to create a nonequijoin.
- [x] b. The tables to be joined have columns with the same name and compatible data types. ✔
- [x] c. The tables to be joined have columns of the same name and different data types. ✔
- [ ] d. The tables to be joined have multiple NULL columns.

Question 6

Correct

Mark 1.00 out of 1.00

Flag question

Which SELECT statement displays all the employees who do not have any subordinates?

Select one:

- a. SELECT e.ename
  FROM emp e
  WHERE e.mgr IS NOT NULL;
- b. SELECT e.ename
  FROM emp e
  WHERE e.empno IN (SELECT m.mgr
  FROM emp m);
- c. SELECT e.ename
  FROM emp e
  WHERE e.empno NOT IN (SELECT m.mgr
  FROM emp m);
- d. SELECT e.ename
  FROM emp e
  WHERE e.empno NOT IN (SELECT m.mgr
  FROM emp m
  WHERE m.mgr IS NOT NULL); ✔

Question 7

Correct

Mark 1.00 out of 1.00

Flag question

To display the names of employees who earns more than the average salary of all employees.

SELECT last_name, first_name
FROMemployee
WHEREsalary > AVG(salary);

Which change should you make to achieve the desired results?

Select one:

- a. Move the function to the SELECT clause and add a GROUP BY clause.
- b. Change the function in the WHERE clause.
- c. Move the function to the SELECT clause and add a GROUP BY clause and a HAVING clause.
- d. Use a subquery in the WHERE clause to compare the average salary value. ✔

Question 8

Correct

Mark 1.00 out of 1.00

Flag question

In which cases would you use an outer join?

Select one:

- a. The tables being joined have NOT NULL columns.
- b. The tables being joined have both matched and unmatched data. ✔
- c. The tables being joined have only unmatched data.
- d. Only when the tables have a primary key/foreign key relationship.
- e. The tables being joined have only matched data.

Question 9

Correct

Mark 1.00 out of 1.00

Flag question

Evaluate this SQL statement:
SELECT first_name, commission
FROM employee
WHERE commission =
(SELECTcommission
FROMemployee
WHEREUPPER(first_name) = 'SCOTT')

What would cause this statement to fail?

Select one:

- a. There is no employee with the first name Scott.
- b. Scott has a NULL commission value.
- c. The FIRST_NAME values in the database are in lowercase.
- d. There is more than one employee with the first name Scott. ✔
- e. Scott has a zero commission value.

Question 10

Correct

Mark 1.00 out of 1.00

Flag question

Consider the below tables:

Employee Table

Column Name	DataType	Constraint
Name	Varchar2(20)	
Empno	Number(10)	PK
salary	Number(10,2)	

Tax Table

Column Name	DataType	Constraint
Taxgrade	Number	
Lowsal	Number(10)	
highsal	Number(10,2)	

We want to create a report that displays the employee details along with the tax category of each employee. The tax category is determined by comparing the salary of the employee from the EMP table to the lower and upper salary values in the TAX table.

Which SELECT statement produces the required results?

Select one:

a. SELECT e.name, e.salary, t.taxgrade
FROM emp e, tax t
WHERE e.salary <= t.lowsal AND e.salary >= t.highsal;

b. SELECT e.name, e.salary, t.taxgrade
FROM emp e, tax t
WHERE e.salary >= t.lowsal AND <= t.highsal;

c. SELECT e.name, e.salary, t.taxgrade
FROM emp e, tax t
WHERE e.salary IN t.lowsal AND t.highsal;

d. SELECT e.name, e.salary, t.taxgrade
FROM emp e, tax t
WHERE e.salary BETWEEN t.lowsal AND t.highsal; ✔

Question 3

Correct

Mark 1.00 out of 1.00

Flag question

Consider the following table:

Product Table

Column Name	DataType	Constraint
prod_name	Varchar2(20)	
prod_id	Number(10)	PK

Customer Table

Column Name	DataType	Constraint
cust_last_name	Varchar2(20)	
cust_id	Number(10)	PK
cust_city	Varchar2(20)	

Sales Table

Column Name	DataType	Constraint
prod_id	Number(10)	FK
cust_id	Number(10)	FK
quantity_sold	Number(10,2)	

Generate a report that gives details of the customer's last name, name of the product and the quantity sold for all customers in 'Tokyo'.

Which two queries give the required result? (Choose two.)

Select one or more:

- [ ] a. SELECT c.cust_last_name,p.prod_name,s.quantity_sold
  FROM products p JOIN sales s JOIN customers c
  ON(p.prod_id=s.prod_id)
  ON(s.cust_id=c.cust_id)
  WHERE c.cust_city='Tokyo';
- [ ] b. SELECT c.cust_last_name,p.prod_name,s.quantity_sold
  FROM products p JOIN sales s
  USING (prod_id)
  ON(p.prod_id=s.prod_id)
  JOIN customers c
  USING(cust_id)
  WHERE c.cust_city='Tokyo';
- [x] c. SELECT c.cust_last_name,p.prod_name,s.quantity_sold
  FROM sales s JOIN products p
  USING (prod_id)
  JOIN customers c
  USING (cust_id)
  WHERE c.cust_city='Tokyo'; ✔
- [x] d. SELECT c.cust_last_name,p.prod_name,s.quantity_sold
  FROM products p JOIN sales s
  ON(p.prod_id=s.prod_id)
  JOIN customers c
  ON(s.cust_id=c.cust_id)
  WHERE c.cust_city='Tokyo'; ✔