

# IT Skill Test GIC Myanmar

Duration: 30 Minutes

Total Questions: 8

---

1. You need to find all employees who earn more than their department's average salary. Which of the following SQL queries would accomplish this task?

- A.SELECT e.employee\_id, e.salary FROM employees e WHERE e.salary > (SELECT AVG(salary) FROM employees)
- B.SELECT e.employee\_id, e.salary FROM employees e WHERE e.salary > (SELECT AVG(salary) FROM employees WHERE department\_id = e.department\_id)
- C.SELECT e.employee\_id, e.salary FROM employees e JOIN (SELECT department\_id, AVG(salary) AS avg\_salary FROM employees GROUP BY department\_id) d ON e.department\_id = d.department\_id WHERE e.salary > d.avg\_salary
- D.SELECT e.employee\_id, e.salary FROM employees e WHERE e.salary > AVG(salary) GROUP BY department\_id

2. Which of the following statements about subqueries in Oracle SQL is TRUE?

- A.Subqueries can only be used in the WHERE clause of a SELECT statement
- B.Subqueries always return multiple rows
- C.Subqueries can be used in SELECT, FROM, and WHERE clauses
- D.Correlated subqueries are always more efficient than non-correlated subqueries

3. You need to update the salaries of all employees who earn less than the average salary in their department by increasing them by 10%. Which SQL statement would you use?
- A.UPDATE employees e SET salary = salary \* 1.1 WHERE salary < (SELECT AVG(salary) FROM employees WHERE department\_id = e.department\_id)
- B.UPDATE employees SET salary = salary \* 1.1 WHERE salary < AVG(salary)  
GROUP BY department\_id
- C.UPDATE employees e SET e.salary = e.salary \* 1.1 WHERE e.salary < (SELECT AVG(d.salary) FROM employees d WHERE d.department\_id = e.department\_id)
- D.UPDATE (SELECT e.employee\_id, e.salary, AVG(d.salary) OVER (PARTITION BY e.department\_id) AS avg\_salary FROM employees e JOIN employees d ON e.department\_id = d.department\_id) SET salary = salary \* 1.1 WHERE salary < avg\_salary
4. What is the purpose of the EXISTS operator in a subquery?
- A.To check if a subquery returns any rows
- B.To compare a value with a list of values returned by a subquery
- C.To join two tables based on a condition
- D.To calculate aggregate functions in a subquery
5. You're working on a query to find all departments that have at least one employee earning more than \$100,000. Which of the following queries would correctly accomplish this task?
- A.SELECT department\_id FROM employees WHERE salary > 100000
- B.SELECT DISTINCT department\_id FROM employees WHERE salary > 100000
- C.SELECT department\_id FROM departments WHERE EXISTS (SELECT 1 FROM employees WHERE employees.department\_id = departments.department\_id AND salary > 100000)
- D.SELECT department\_id FROM departments d WHERE d.department\_id IN (SELECT department\_id FROM employees GROUP BY department\_id HAVING

$\text{MAX}(\text{salary}) > 100000$

6. You need to write a query to find the employees who earn more than 50% of their department's total salary. Which of the following queries would correctly accomplish this task?
- A.`SELECT e.employee_id, e.salary FROM employees e WHERE e.salary > 0.5 * (SELECT SUM(salary) FROM employees WHERE department_id = e.department_id)`
- B.`SELECT e.employee_id, e.salary FROM employees e JOIN (SELECT department_id, SUM(salary) AS total_salary FROM employees GROUP BY department_id) d ON e.department_id = d.department_id WHERE e.salary > 0.5 * d.total_salary`
- C.`SELECT employee_id, salary FROM employees WHERE salary > 0.5 * SUM(salary) OVER (PARTITION BY department_id)`
- D.`SELECT e.employee_id, e.salary FROM employees e WHERE e.salary > (SELECT AVG(salary) * 0.5 FROM employees WHERE department_id = e.department_id)`
7. Your team is analyzing customer behavior across different sales channels. You need to identify customers who have made purchases both online and in-store. Which SQL query would be most appropriate for this task?
- A.`SELECT customer_id FROM Online_Sales UNION SELECT customer_id FROM Store_Sales`
- B.`SELECT customer_id FROM Online_Sales INTERSECT SELECT customer_id FROM Store_Sales`
- C.`SELECT customer_id FROM Online_Sales MINUS SELECT customer_id FROM Store_Sales`
- D.`SELECT customer_id FROM Online_Sales UNION ALL SELECT customer_id FROM Store_Sales`

8. You are tasked with writing a query to find the second highest salary in each department. The database has an 'employees' table with columns: employee\_id, department\_id, and salary. Which of the following queries would correctly solve this problem?
- A.SELECT department\_id, MAX(salary) AS second\_highest\_salary FROM employees WHERE salary < (SELECT MAX(salary) FROM employees e2 WHERE e2.department\_id = employees.department\_id) GROUP BY department\_id
- B.SELECT department\_id, salary AS second\_highest\_salary FROM (SELECT department\_id, salary, DENSE\_RANK() OVER (PARTITION BY department\_id ORDER BY salary DESC) AS rank FROM employees) WHERE rank = 2
- C.SELECT department\_id, MIN(salary) AS second\_highest\_salary FROM (SELECT department\_id, salary FROM employees e WHERE 2 > (SELECT COUNT(DISTINCT salary) FROM employees WHERE department\_id = e.department\_id AND salary > e.salary)) GROUP BY department\_id
- D.SELECT department\_id, MAX(salary) AS second\_highest\_salary FROM employees GROUP BY department\_id HAVING COUNT(DISTINCT salary) >= 2