

# IT Skill Test GIC Myanmar

Duration: 30 Minutes

Total Questions: 8

---

1. You need to insert data into a table named 'EMPLOYEES' with columns (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, HIRE\_DATE, JOB\_ID, SALARY). Which of the following SQL statements correctly inserts a new employee record?
  - A.INSERT INTO EMPLOYEES VALUES (301, 'John', 'Doe', 'jdoe@example.com', SYSDATE, 'IT\_PROG', 75000);
  - B.INSERT EMPLOYEES (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, HIRE\_DATE, JOB\_ID, SALARY) VALUES (301, 'John', 'Doe', 'jdoe@example.com', SYSDATE, 'IT\_PROG', 75000);
  - C.INSERT INTO EMPLOYEES (EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, HIRE\_DATE, JOB\_ID, SALARY) VALUES (301, 'John', 'Doe', 'jdoe@example.com', SYSDATE, 'IT\_PROG', 75000);
  - D.INSERT VALUES (301, 'John', 'Doe', 'jdoe@example.com', SYSDATE, 'IT\_PROG', 75000) INTO EMPLOYEES;
  
2. You need to update the salary of all employees in the 'IT\_PROG' job category by increasing it by 10%. Which SQL statement would accomplish this?
  - A.UPDATE EMPLOYEES SET SALARY = SALARY \* 1.1 WHERE JOB\_ID = 'IT\_PROG';
  - B.UPDATE EMPLOYEES SET SALARY += SALARY \* 0.1 WHERE JOB\_ID = 'IT\_PROG';
  - C.UPDATE EMPLOYEES SET SALARY = SALARY + (SALARY \* 10%) WHERE JOB\_ID = 'IT\_PROG';
  - D.MODIFY EMPLOYEES SET SALARY = SALARY \* 110% WHERE JOB\_ID = 'IT\_PROG';

3. You need to delete all employees from the EMPLOYEES table who were hired before the year 2000. Which SQL statement would correctly accomplish this?
- A.DELETE EMPLOYEES WHERE HIRE\_DATE < '01-JAN-2000';  
B.DELETE FROM EMPLOYEES WHERE HIRE\_DATE < TO\_DATE('2000-01-01', 'YYYY-MM-DD');  
C.REMOVE FROM EMPLOYEES WHERE HIRE\_DATE < '2000-01-01';  
D.DELETE \* FROM EMPLOYEES WHERE HIRE\_DATE < '01/01/2000';
4. You need to insert data into a table named SALES\_HISTORY, copying all records from the SALES table where the sale date is older than one year. Which SQL statement would correctly accomplish this?
- A.INSERT INTO SALES\_HISTORY SELECT \* FROM SALES WHERE SALE\_DATE < ADD\_MONTHS(SYSDATE, -12);  
B.INSERT ALL INTO SALES\_HISTORY SELECT \* FROM SALES WHERE SALE\_DATE < SYSDATE - 365;  
C.COPY INTO SALES\_HISTORY FROM SALES WHERE SALE\_DATE < ADD\_MONTHS(SYSDATE, -12);  
D.INSERT INTO SALES\_HISTORY VALUES (SELECT \* FROM SALES WHERE SALE\_DATE < SYSDATE - 1 YEAR);
5. You have a table PRODUCTS (PRODUCT\_ID, PRODUCT\_NAME, PRICE, STOCK\_QUANTITY) and another table NEW\_PRODUCTS with the same structure. You need to update the PRODUCTS table with data from NEW\_PRODUCTS, inserting new products and updating existing ones. Which SQL statement would be most appropriate?
- A.UPDATE PRODUCTS SET (PRODUCT\_NAME, PRICE, STOCK\_QUANTITY) = (SELECT PRODUCT\_NAME, PRICE, STOCK\_QUANTITY FROM NEW\_PRODUCTS WHERE NEW\_PRODUCTS.PRODUCT\_ID = PRODUCTS.PRODUCT\_ID);  
B.INSERT INTO PRODUCTS SELECT \* FROM NEW\_PRODUCTS WHERE PRODUCT\_ID NOT IN (SELECT PRODUCT\_ID FROM PRODUCTS);  
C.MERGE INTO PRODUCTS p USING NEW\_PRODUCTS np ON (p.PRODUCT\_ID =

```
np.PRODUCT_ID) WHEN MATCHED THEN UPDATE SET p.PRODUCT_NAME =  
np.PRODUCT_NAME, p.PRICE = np.PRICE, p STOCK_QUANTITY =  
np STOCK_QUANTITY WHEN NOT MATCHED THEN INSERT (PRODUCT_ID,  
PRODUCT_NAME, PRICE, STOCK_QUANTITY) VALUES (np.PRODUCT_ID,  
np.PRODUCT_NAME, np.PRICE, np.STOCK_QUANTITY);
```

D.INSERT OR UPDATE INTO PRODUCTS SELECT \* FROM NEW\_PRODUCTS;

6. You have a table SALES (SALE\_ID, PRODUCT\_ID, SALE\_DATE, QUANTITY, TOTAL\_AMOUNT) and need to update the TOTAL\_AMOUNT based on a new price list in the PRODUCTS table (PRODUCT\_ID, PRICE). Which SQL statement would correctly update the SALES table?
- A.UPDATE SALES SET TOTAL\_AMOUNT = QUANTITY \* (SELECT PRICE FROM PRODUCTS WHERE PRODUCTS.PRODUCT\_ID = SALES.PRODUCT\_ID);
- B.UPDATE SALES s SET s.TOTAL\_AMOUNT = s.QUANTITY \* p.PRICE FROM PRODUCTS p WHERE p.PRODUCT\_ID = s.PRODUCT\_ID;
- C.MERGE INTO SALES s USING PRODUCTS p ON (s.PRODUCT\_ID = p.PRODUCT\_ID) WHEN MATCHED THEN UPDATE SET s.TOTAL\_AMOUNT = s.QUANTITY \* p.PRICE;
- D.UPDATE SALES SET TOTAL\_AMOUNT = QUANTITY \* PRODUCTS.PRICE FROM PRODUCTS WHERE SALES.PRODUCT\_ID = PRODUCTS.PRODUCT\_ID;
7. You need to insert data into a table EMPLOYEE\_SUMMARY (DEPARTMENT\_ID, TOTAL\_EMPLOYEES, TOTAL\_SALARY) by summarizing data from the EMPLOYEES table (EMPLOYEE\_ID, DEPARTMENT\_ID, SALARY). Which SQL statement would correctly accomplish this?
- A.INSERT INTO EMPLOYEE\_SUMMARY SELECT DEPARTMENT\_ID, COUNT(\*), SUM(SALARY) FROM EMPLOYEES GROUP BY DEPARTMENT\_ID;
- B.INSERT ALL INTO EMPLOYEE\_SUMMARY VALUES (DEPARTMENT\_ID, COUNT(\*), SUM(SALARY)) SELECT \* FROM EMPLOYEES GROUP BY DEPARTMENT\_ID;
- C.MERGE INTO EMPLOYEE\_SUMMARY es USING (SELECT DEPARTMENT\_ID, COUNT(\*) as TOTAL\_EMPLOYEES, SUM(SALARY) as TOTAL\_SALARY FROM

```
EMPLOYEES GROUP BY DEPARTMENT_ID) tmp ON (es.DEPARTMENT_ID =
tmp.DEPARTMENT_ID) WHEN MATCHED THEN UPDATE SET
es.TOTAL_EMPLOYEES = tmp.TOTAL_EMPLOYEES, es.TOTAL_SALARY =
tmp.TOTAL_SALARY WHEN NOT MATCHED THEN INSERT (DEPARTMENT_ID,
TOTAL_EMPLOYEES, TOTAL_SALARY) VALUES (tmp.DEPARTMENT_ID,
tmp.TOTAL_EMPLOYEES, tmp.TOTAL_SALARY);

D.UPDATE EMPLOYEE_SUMMARY SET (TOTAL_EMPLOYEES, TOTAL_SALARY) = (SELECT
COUNT(*), SUM(SALARY) FROM EMPLOYEES GROUP BY DEPARTMENT_ID);
```

8. You have a table ORDERS (ORDER\_ID, CUSTOMER\_ID, ORDER\_DATE, STATUS) and need to update the STATUS of all orders that are older than 30 days and still have a 'PENDING' status to 'CANCELLED'. Which SQL statement would correctly accomplish this?
- A.UPDATE ORDERS SET STATUS = 'CANCELLED' WHERE ORDER\_DATE < SYSDATE - 30 AND STATUS = 'PENDING';
- B.UPDATE ORDERS SET STATUS = 'CANCELLED' WHERE DATEDIFF(ORDER\_DATE, SYSDATE) > 30 AND STATUS = 'PENDING';
- C.MERGE INTO ORDERS o USING (SELECT ORDER\_ID FROM ORDERS WHERE ORDER\_DATE < SYSDATE - 30 AND STATUS = 'PENDING') tmp ON (o.ORDER\_ID = tmp.ORDER\_ID) WHEN MATCHED THEN UPDATE SET o.STATUS = 'CANCELLED';
- D.UPDATE ORDERS SET STATUS = CASE WHEN ORDER\_DATE < SYSDATE - 30 AND STATUS = 'PENDING' THEN 'CANCELLED' ELSE STATUS END;