

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;