DBMS internal prep 2

1. Find a Minimum of Two Numbers Using Procedures

Q: Write a program to find a minimum of two numbers using procedures.

A:

```
CREATE OR REPLACE PROCEDURE find_minimum(
    num1 IN NUMBER,
    num2 IN NUMBER,
    min_num OUT NUMBER
) AS
BEGIN
    IF num1 < num2 THEN
        min_num := num1;
    ELSE
        min_num := num2;
    END IF;
END;
/
```

Execution Example:

```
DECLARE
    result NUMBER;
BEGIN
    find_minimum(10, 20, result);
    DBMS_OUTPUT.PUT_LINE('Minimum: ' || result);
END;
/
```

2. Insert Values into a Table Using Procedures

Q: Write a program to insert values into a table using procedures.

A:

```
CREATE OR REPLACE PROCEDURE insert_into_table(
    table_name IN VARCHAR2,
    col1_value IN NUMBER,
    col2_value IN VARCHAR2
) AS
BEGIN
    EXECUTE IMMEDIATE 'INSERT INTO ' || table_name || ' VALUE
S(:1, :2)'
    USING col1_value, col2_value;
END;
/
```

Execution Example:

```
BEGIN
    insert_into_table('customer', 11, 'test_name');
END;
/
```

---

3. Find the Factorial of a Given Number Using Functions

Q: Write a program to find the factorial of a given number using functions.

A:

```
CREATE OR REPLACE FUNCTION factorial(n IN NUMBER) RETURN NUMB
ER IS
    result NUMBER := 1;
BEGIN
    FOR i IN 1..n LOOP
        result := result * i;
    END LOOP;
    RETURN result;
```

```
END;
/
```

Execution Example:

```
DECLARE
    fact NUMBER;
BEGIN
    fact := factorial(5);
    DBMS_OUTPUT.PUT_LINE('Factorial: ' || fact);
END;
/
```

---

4. Exception Handling (Zero Error)

Q: Implement PL/SQL program to print a given error statement (Exception handling for divide-by-zero).

A:

```
BEGIN
    DECLARE
        num NUMBER := 10;
        denom NUMBER := 0;
        result NUMBER;
    BEGIN
        result := num / denom;
    EXCEPTION
        WHEN ZERO_DIVIDE THEN
            DBMS_OUTPUT.PUT_LINE('Error: Division by zero is
not allowed.');
    END;
END;
/
```

---

5. Exception Handling (No Rows Found)

Q: Implement PL/SQL program to print a given error statement (Exception handling for no rows found).

A:

```
BEGIN
    DECLARE
        v_name VARCHAR2(50);
    BEGIN
        SELECT cname INTO v_name FROM customer WHERE cid = 999;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            DBMS_OUTPUT.PUT_LINE('Error: No rows found for the given ID.');
    END;
END;
/
```

---

6. Increment Salary of Employees Using Cursors

Q: Write a program to increment the salary of employees in the table using cursors.

A:

```
DECLARE
    CURSOR emp_cursor IS SELECT cid, sal FROM customer WHERE sal IS NOT NULL;
    v_cid customer.cid%TYPE;
    v_sal customer.sal%TYPE;
BEGIN
    OPEN emp_cursor;
    LOOP
        FETCH emp_cursor INTO v_cid, v_sal;
        EXIT WHEN emp_cursor%NOTFOUND;
        UPDATE customer SET sal = sal + 1000 WHERE cid = v_ci
d;
    END LOOP;
    CLOSE emp_cursor;
    COMMIT;
END;
/
```

7. Implement Triggers for Update

Q: Write a program to implement triggers (Update).

A:

```
CREATE OR REPLACE TRIGGER update_trigger
AFTER UPDATE ON customer
FOR EACH ROW
BEGIN
    DBMS_OUTPUT.PUT_LINE('Customer ID ' || :OLD.cid || ' was
updated.');
END;
/
```

8. Create a View from Two Tables

Q: Write a SQL query to create a view from two tables.

A:

```
CREATE OR REPLACE VIEW orders_customer_view AS
SELECT
    o.sno, o.orderdate, o.amount, c.cname, c.age
FROM
    orders o
```

```
JOIN
    customer c ON o.cid = c.cid;
```

---

9. Update a Row in the View

Q: Write a SQL query to update a row in the view.

A:

```
UPDATE orders_customer_view
SET amount = 1000
WHERE sno = 1;
```

---

10. Inner Join

Q: Display cname, age, sid, and amount using an inner join.

A:

```
SELECT
    c.cname, c.age, o.sid, o.amount
FROM
    customer c
INNER JOIN
    orders o ON c.cid = o.cid;
```

---

11. Left Outer Join

Q: Display cname, cid, order_date, and amount using left outer join.

A:

```
SELECT
    c.cname, c.cid, o.orderdate, o.amount
FROM
    customer c
```

```
LEFT OUTER JOIN
    orders o ON c.cid = o.cid;
```

12. Full Join

Q: Display sno, cname, age, and sal using a full join.

A:

```
SELECT
    o.sno, c.cname, c.age, c.sal
FROM
    orders o
FULL OUTER JOIN
    customer c ON o.cid = c.cid;
```

13. Nested Queries

Q: Nested query example for observations.

A:

```
SELECT
    cname, age
FROM
    customer
WHERE
    cid = (SELECT cid FROM orders WHERE amount = (SELECT MAX
(amount) FROM orders));
```