1. The bank manager has decided to activate all those accounts which were previously marked as inactive for performing no transaction in last 365 days. Write a PL/SQ block (using implicit cursor) to update the status of account, display an approximate message based on the no. of rows affected by the update. (Use of %FOUND, %NOTFOUND, %ROWCOUNT)
2. Organization has decided to increase the salary of employees by 10% of existing salary, who are having salary less than average salary of organization, Whenever such salary updates takes place, a record for the same is maintained in the increment\_salary table.
3. Write PL/SQL block using explicit cursor for following requirements: College has decided to mark all those students detained (D) who are having attendance less than 75%. Whenever such update takes place, a record for the same is maintained in the D\_Stud table. create table stud21(roll number(4), att number(4), status varchar(1));

a) PL/SQL Block to Activate Inactive Accounts:

```sql

DECLARE

-- Declare a cursor to select inactive accounts

CURSOR inactive\_accounts\_cur IS

SELECT account\_id

FROM accounts

WHERE last\_transaction\_date < SYSDATE - 365;

v\_account\_id accounts.account\_id%TYPE;

v\_rows\_updated NUMBER := 0;

BEGIN

OPEN inactive\_accounts\_cur;

LOOP

FETCH inactive\_accounts\_cur INTO v\_account\_id;

EXIT WHEN inactive\_accounts\_cur%NOTFOUND;

-- Activate the account

UPDATE accounts

SET status = 'Active'

WHERE account\_id = v\_account\_id;

v\_rows\_updated := v\_rows\_updated + SQL%ROWCOUNT;

END LOOP;

CLOSE inactive\_accounts\_cur;

DBMS\_OUTPUT.PUT\_LINE('Approximately ' || v\_rows\_updated || ' accounts have been activated.');

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An error occurred: ' || SQLERRM);

END;

/

```

In this PL/SQL block, we use an implicit cursor to select inactive accounts based on the last transaction date and activate them. We also display a message indicating the approximate number of accounts that have been activated.

b) PL/SQL Block to Increase Salary and Maintain Increment Records:

```sql

DECLARE

v\_avg\_salary NUMBER;

v\_increment\_amount NUMBER := 0.1; -- 10% increase

BEGIN

-- Calculate the average salary of the organization

SELECT AVG(salary)

INTO v\_avg\_salary

FROM employees;

-- Update the salary of employees with salary less than the average

UPDATE employees

SET salary = salary + (salary \* v\_increment\_amount)

WHERE salary < v\_avg\_salary;

-- Insert records into the increment\_salary table

INSERT INTO increment\_salary (employee\_id, increment\_date, increment\_percentage)

SELECT employee\_id, SYSDATE, v\_increment\_amount

FROM employees

WHERE salary < v\_avg\_salary;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Salary increment has been applied to eligible employees.');

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An error occurred: ' || SQLERRM);

END;

/

```

In this PL/SQL block, we first calculate the average salary of the organization and then increase the salary of employees with salaries less than the average. We also insert records into the increment\_salary table to maintain the increment history.

c) Execute in Ubuntu Terminal:

To execute the PL/SQL block in the Ubuntu terminal, you can use SQL\*Plus or any other Oracle Database command-line tool.

1. Open a terminal window.

2. Launch SQL\*Plus by running the following command and provide the necessary login credentials:

```bash

sqlplus username/password@database

```

Replace `username`, `password`, and `database` with your actual database login credentials and database identifier.

3. Once you are connected to the database, you can paste the PL/SQL block and execute it by entering a forward slash `/` on a new line.

The PL/SQL block will be executed, and the results or messages will be displayed in the terminal.

Make sure you have the necessary privileges to perform updates and inserts in the database.