Problem 1: Procedure (20 Marks)

Write a Procedure named **RANK_JOBS** that takes one integer **MIN_HIRED_COUNT** as input. It ranks the jobs whose JOB_ID matches with the ID extracted from its JOB_TITLE. The extracted ID contains the first two letters of the first word of the title, followed by an underscore, followed by the first three letters of the second word (if any) of the title. For example, from the title 'Sales Representative', the extracted ID would be 'SA_REP'. If the actual JOB_ID also is 'SA_REP', then there is a match.

The jobs having this match, which also have to have their number of hired employees no less than *MIN_HIRED_COUNT*, are then ranked by the Procedure *RANK_JOBS*. The ranking is done according to the descending order of their average salaries, with the job having the highest average salary ranked as 1. Finally, insert the ranks of the jobs in the JOB_RANK table given below.

```
CREATE TABLE JOB_RANK (
    JOB_ID VARCHAR2(10),
    RANK NUMBER
);
```

Sample Output:

```
BEGIN
    RANK_JOBS(MIN_HIRED_COUNT => 5);
END;
```

```
RANK-1
Sales Manager(SA_MAN)
EMPLOYEES HIRED: 5
AVG SALARY: 12200

RANK-2
Sales Representative(SA_REP)
EMPLOYEES HIRED: 30
AVG SALARY: 8350

RANK-3
Stock Manager(ST_MAN)
EMPLOYEES HIRED: 5
AVG SALARY: 7280
```

Problem 2: Trigger (20 Marks)

Please write a trigger to validate updates of salary values in the TEMP_EMPLOYEES table, which is a copy of the HR.EMPLOYEES table). The trigger should only be invoked when the SALARY column is updated in the TEMP_EMPLOYEES Table.

This trigger should check if the updates are valid. The constraint for validating salary of an employee is as follows:
- Case 1: The salary cannot be less than the minimum salary of the same JOB_ID of other employees in the company.
- Case 2: If the employee is not a manager, then the salary must be less than the minimum salaries of the managers who manage the same JOB_ID.
- Case 3: If the employee is a manager, the salary must be less than the salary of the manager of the same department.

If the salary is not valid, then the trigger should throw an exception so that the update results in an error.
If the salary is valid, the update should be allowed.

First, create a TEMP_EMPLOYEES table that is a copy of the EMPLOYEES table (SQL given below). Then write your trigger on the test TEMP_EMPLOYEES table. To avoid the "mutating" trigger issue, all your PL/SQL statements should perform validation checks on the original EMPLOYEES table (instead of TEMP_EMPLOYEES table).

```
-- CREATE THE FOLLOWING TABLE
DROP TABLE TEMP_EMPLOYEES;

CREATE TABLE TEMP_EMPLOYEES AS SELECT * FROM EMPLOYEES;
SELECT * FROM TEMP_EMPLOYEES;

–TEST YOUR TRIGGER [use small and large values to check trigger conditions]
UPDATE TEMP_EMPLOYEES
SET SALARY = 100000
WHERE EMPLOYEE_ID = 198;
```