Souop Fotso Jocelyn Axel

LAB 2

The following relations are given (primary keys are underlined, optional attributes are denoted with *):

ATHLETE(AthleteCode, TeamName)
ATHLETE_ARRIVAL(AthleteCode, Time)
TEAM_ARRIVAL(TeamName, NumberArrivedAthletes)
RANKING(AthleteCode, Position, Time)

Write triggers to *update TEAM_ARRIVAL and RANKING tables* when a new row is inserted in ATHLETE_ARRIVAL table. For the update of the TEAM_ARRIVAL table, consider also the case of a team not yet inserted in the table. For the update of RANKING table, consider that the Time field can assume the same value for two different athletes.

Draft solution

```
CREATE OR REPLACE TRIGGER UPDATE_RANKING
AFTER INSERT ON ATHLETE_ARRIVAL
FOR EACH ROW
DECLARE
        APos NUMBER;
        X NUMBER;
        ATime NUMBER;
        Draw BOOLEAN;
BEGIN
        -- verify if the ranking is empty and compute the position for the new athlete
        SELECT MAX(Time), MAX(Position) INTO ATime, X
        FROM RANKING
        WHERE Time <= :NEW.Time;

        IF (X IS NULL) THEN
        --- ranking empty or athlete in the first position without any other athlete with the same time
                APos := 1;
                Draw := FALSE;
        ELSE
        --- verify if there is another athlete with the same time
                IF (ATime < :NEW.Time) THEN
                        APos := X +1;
                        Draw := FALSE;
                ELSE
                        APos := X;
                        Draw := TRUE;
                END IF;
        END IF;
```

```
        INSERT INTO RANKING (Position, AthleteCode, Time)
        VALUES (APos, :NEW.AthleteCode, :NEW.Time);

        IF (Draw = FALSE) THEN
                UPDATE RANKING SET Position=Position+1
                WHERE Time > :NEW.Time;
        END IF;
END;


CREATE OR REPLACE TRIGGER UPDATE_TEAM_ARRIVAL
AFTER INSERT ON ATHLETE_ARRIVAL
FOR EACH ROW
DECLARE
        Team VARCHAR(10);
        X NUMBER;
BEGIN
        ---read the team of the new athlete
        SELECT TeamName INTO Team
        FROM ATHLETE
        WHERE AthleteCode = :NEW.AthleteCode;

        ---check if the new athlete is the first athlete for the team
        SELECT COUNT(*) INTO X
        FROM TEAM_ARRIVAL
        WHERE TeamName = Team;

        IF (X=0) THEN
        ---the new athlete is the first athlete for the team
                INSERT INTO TEAM_ARRIVAL (TeamName, NumberArrivedAthletes)
                VALUES (Team,1);
        ELSE
                UPDATE TEAM_ARRIVAL
                SET NumberArrivedAthletes = NumberArrivedAthletes + 1
                WHERE TeamName= Team;
        END IF;
END;
```