Advanced DataBase - Lab 1

Issam Falih & Hanen Ochi

Part I: Hands on SQL

A- Download from Teams on `M1 - ST2ADB - CM` team (assignment for Lab 1) and run the `ecole_oracle.sql` script to create and fill the tables

B- Formulate and test the following queries in SQL, offering at least two different solutions for each question if possible:

1. List the names, first names and dates of birth of all the students.
2. Provide full information on all activities.
3. Obtain the names of the students whose weight is between 60 and 80 kilos.
4. Obtain the names of professors whose specialty is ”poésie” or SQL.
5. Obtain the names of students whose names begin with ”L”.
6. Obtain the names of professors whose specialties are unknown.
7. Obtain, for each professor, his name and his specialty. If the latter is unknown, we want to display the character string: ‘****’.
8. What are the first and last names of the students who practice surfing at level 1. Write this request in at least **three** different ways.
9. Obtain peer names of professors who have the same specialty.
10. For each professor, display his draft date, his date of last promotion as well as the number of years elapsed between these two dates.
11. Display the average age of students. This average age will be expressed in years.
12. Obtain the list of students who will be at least 24 years old in less than 4 months.
13. Obtain his name and his average for each 1st year pupil.
14. Which 1st year students have an average higher than the 1st year average?
15. Obtain the name and weight of grade 1 students heavier than any grade 2 student.
16. Obtain the name, weight and grade of students weighing more than the average weight of students in the same grade.
17. Obtain the names of teachers who are not teaching class 1.

18. Obtain the names of grade 1 students who have obtained more than 60% and who play tennis.
19. Teachers who take charge of all the second year courses; we ask for the Number and the name.
20. Students who practice all the activities; we ask for the Number and the name.

Note: Save all you're queries in a sql file and upload it on Moodle using the below format.
`GroupeNumber_LastName_Lab1PartI.sql`

Part II: PL/SQL

Delete all the table created in Part I and re lunch the `ecole_oracle.sql` file.

C- Constraints

1. Try changing tables to add the following constraints in SQL:
2. The score a student must be between 0 and 20.
3. The sex of a student must be in the list: 'm', 'M', 'f', 'F' or Null.
4. The base salary of a teacher must be below the current salary. (horizontal constraint)
5. The salary of a teacher should not exceed twice the average teacher salary in the same specialty. (vertical constraint)

D- Triggers

1. Create a trigger to check the constraint: "The salary of a teacher can not decrease."
2. Create the following table:

```
CREATE TABLE PROF_SPECIALITE (SPECIALTE VARCHAR2 (20) NB_PROFESSEURS NUMBER);
```

3. Create a trigger to populate and update automatically the table `PROF_SPECIALITE` after each updating operation (`insertion`, `deletion`, `modification`) on the Table of teacher
4. Test the trigger on examples of update.
5. Create a trigger that updates the table `charge` when we delete a professor in the Professeur table or when we change his number. (Cascaded Update)
6. Create the `audit_resultats` table

```
CREATE TABLE AUDIT_RESULTATS ( USER VARCHAR2 (50), DATE_MAJ date,
DESC_MAJ VARCHAR2 (20), NUM_ELEVE NUMBER (4) NOT NULL,
NUM_COURS NUMBER (4) NOT NULL, NUMBER POINTS);
```

7. Create a trigger that update the table `audit_resultats` while changing table `Resultat`. (Security: access record)

   We must know the user who made the modification (USER), the date and a description of the modification (INSERT ',' DELETE ',' NEW','OLD').

   An example of a record on table `audit_resultats` could be:

   ```
   INSERT INTO audit_resultats VALUES (USER, SYSDATE, 'INSERT',
          :NEW.NUM_ELEVE,:NEW.NUM_COURS,:NEW.POINTS);
   ```

8. We wish that only the `GrandChef` user may increase teachers' salaries by more than 20%. The trigger must return an error (No. -20002) and the message `No modification is authorized` if the condition is not respected.

E- Functions and Procedures

1. Create a `fn_moyenne` function calculating the average of a student given as a parameter.
2. Create a `pr_resultat` procedure to display the average of each student with the appropriate marks: failure, passable, pretty good, good, very good.
3. Create a `package` containing these functions and procedures