2022-2023
Advanced Data Base

Lab session2:
SQL3

Objective: Using object-relational introduced in SQL3, We are looking to Implement the database school in SQL3.

1 Creation of types and tables

1. create type adresse_type with a number of streets, street name and city name.

   NB: It must end by a final line containing only a "/". Do not end the definition by ";".

2. Create the type: *personne_type* with a name and a surname.
3. Create the types: *activites_type* and *cours_type*.
4. Create the tables, *personnes, activites* and *cours* associated to the pervious types (*personne_type, activites_type* and *cours_type*). Don’t forgot to add the necessary integrity constraints (at least primary keys).
5. Use *describe* to see descriptions of the types and tables that you just create.

2 Adding and modifying data and queries

1. Add data in the three tables (*personnes, activites* and *cours*) using the same data as the base SQL2.
2. Check if it is about object tables and not relational tables by consulting *user_tables* and *user_object_tables*
3. Write the following queries:
   (a) Course list with all associated information
   (b) Number of Team by activity
   (c) Course list with the number of hours that is greater than or equal to 25

4. Add a *ski* activity for Team *Ace Club* (Level 1)
5. set *Avs80* team to Level 3 on *Volleyball*

3 Inheritance

We would like to define two types *eleve_type* and *professeur_type*. They will inherit the type *personne_type*.

1. The *personne_type* type must be modified (NOT FINAL clause must be added to allow the inheritance - the type's specialization). We are looking also to respect the coverage constraint (no person can be neither student nor teacher).

   Since a type can be modified if tables or other types use it, you must delete the *personne* table before people modifying *personne_type*.

   Suggest a new definition of the *personne_type* type
2. Create the type *professeur_type* that inherits *personne_type* and has these attributes *specialite*, *date_entree,* der_prom, salaire_base and salaire_actuel
3. Create a type *eleve_type* that inherits *personne_type* and which has as attributes *birth_date*, *poids, annee* and *address (adresse* having the *adresse_type* type).
4. Create the *eleves* and *professeurs* tables
5. Write triggers to ensure the partition constraint on the *eleves* and *professeurs* tables (a teacher and a student should not have the same number). Test the trigger with insert clauses.
6. Create the *eleves* and *professeurs* tables
7. Display the list of teachers with all the associated information.

4 collections

1. We would like to define *UE* that is composed by a name and a set of courses (maximum 5).

   Create the *UE_type* type using the pre-dimensioned arrays (VARRAY)
2. Create the UE table and insert data into it.

3. Modify the type *eleve_type* and the *students* table to have a new attribute named *result*. This attribute is a nested table (*nested table*) composed of two *nom_cours* and *point* attributes.

4. Insert the data into the new table *eleves* using the data base in SQL2.

5. Add to each student, one course named “*Web Service*” and a note associated to this course

6. A among students whose the mark of the course named *“Analyse”* is greater than or equal to 10.