Base de données Avancée et données biologiques Examen final

DJENNAOUI RAOUF 2014 0000 2338

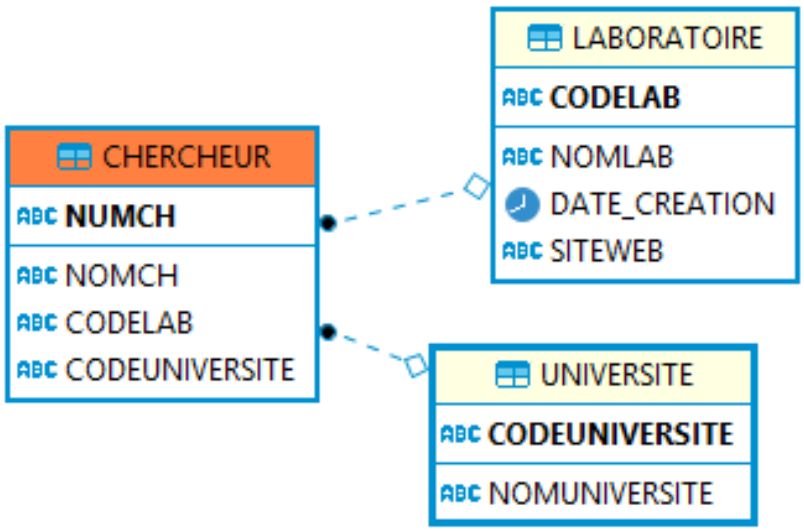
Exercice 1 :

1)

```
CREATE TABLE Laboratoire(
CodeLab  varchar2(20) NOT NULL PRIMARY KEY,
NomLab varchar2(30),
Date_creation date,
Siteweb varchar2(50)
);

CREATE TABLE Universite(
CodeUniversite varchar2(30) NOT NULL PRIMARY KEY,
NomUniversite  varchar2(30)
);

CREATE TABLE Chercheur(
NumCh varchar2(20) NOT NULL PRIMARY KEY,
NomCh varchar2(30),
CodeLab varchar2(20) NOT NULL,
CodeUniversite varchar2(30) NOT NULL,
CONSTRAINT FK_1 FOREIGN KEY (CodeLab) REFERENCES Laboratoire(CodeLab) ON
DELETE CASCADE,
CONSTRAINT FK_2 FOREIGN KEY (CodeUniversite) REFERENCES
Universite(CodeUniversite) ON DELETE CASCADE
);
```



2)

```
SELECT NumCh,NomCh FROM Chercheur WHERE
CodeLab IN(SELECT CodeLab FROM Laboratoire WHERE NomLab='LSI')
AND CodeUniversite IN(SELECT CodeUniversite FROM Universite WHERE
NomUniversite='USTHB');
```

3)

```
ALTER TABLE Laboratoire ADD nbr_ch int DEFAULT 0 ;
```

4)

```
CREATE TRIGGER maj
AFTER INSERT OR DELETE
ON Chercheur FOR EACH ROW
BEGIN
IF INSERTING THEN
update Laboratoire
set nbr_ch:= nbr_ch+1
where CodeLab = :NEW.CodeLab ;
ELSIF DELETING THEN
update Laboratoire
set nbr_ch:= nbr_ch-1
where CodeLab= :OLD.CodeLab ;
END IF ;
END;
```

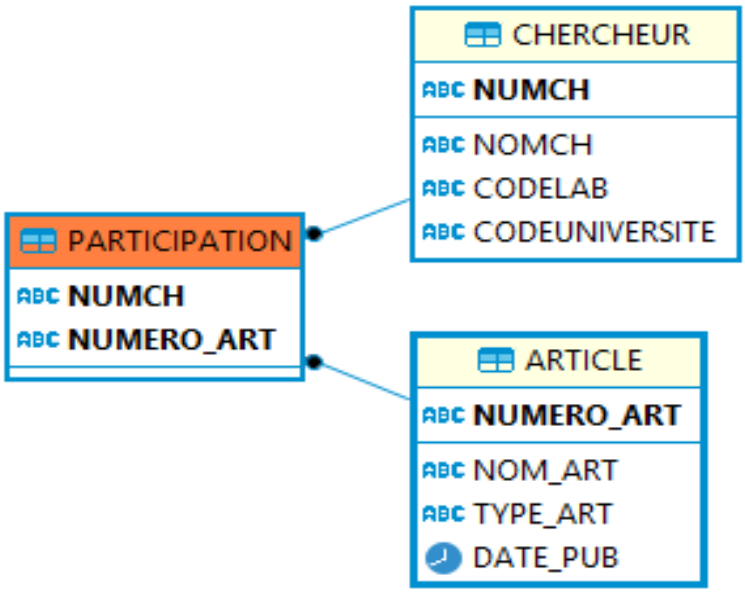
5)

```
Select CodeLab from Laboratoire
where nbr_ch > (select AVG(nbr_ch) from Laboratoire);
```

6)

```
CREATE TABLE Article(
numero_art varchar2(50) NOT NULL PRIMARY KEY,
nom_art varchar2(50),
type_art varchar2(60) CHECK (type_art IN ('conference','revue','livre')),
date_pub DATE
);

CREATE TABLE Participation(
NumCh varchar2(20) NOT NULL,
numero_art varchar2(50) NOT NULL,
CONSTRAINT PK_1 PRIMARY KEY (NumCh,numero_art),
CONSTRAINT FK_3 FOREIGN KEY (NumCh) REFERENCES Chercheur(NumCh) ON DELETE
CASCADE,
CONSTRAINT FK_4 FOREIGN KEY (numero_art) REFERENCES Article(numero_art) ON
DELETE CASCADE
);
```



Exercice 2 :

1)

```
db.createCollection("people")
db.people.insert( {id:"id_1", user_id:"user_id_1",age:18,status:"1" } )
```

2)

```
db.people.update({},{$set: {name:"name_1"}},{multi: true})
```

3)

```
ALTER TABLE people  DROP COLUMN name;
```

4)

```
db.people.createIndex({user_id:1})
```

5)

```
db.people.insert( {user_id:"bcd001",age:45,status:"A" } )
```

6)

```
db.people.aggregate(
    [
    {$unwind:"$status"},
    {$match:{status:{$eq:"A"}}},
    {$project : {_id: 0,id:"$id",
   user_id:"$user_id",age:"$age",status:"$status"}}
    ]
   );
```

Ou bien

```
db.people.find({"status":"A"},{"id":1,"age":1,"user_id":1,"status":1})
```

7)

```
SELECT * FROM people WHERE age >25 ;
```

8)

```
db.people.find(
    {$or:[{status:{$eq:"A"}},{age:{$eq:50}}]}
   )
```

9)

```
db.people.update(
      { status: "A" },
      { $inc: { age: 3 } }
    )
```

10)

```
db.people.remove({"status":"D"})
```

11)

```
db.people.aggregate([{
        "$group": {
        "_id": "$status",
        "Moy": {"$avg":"$age" }
         }
}])
```

Exercice 3 :

1) Trois concept d'un SGBD No SQL :

No SQL orienté graph

No SQL orienté document

No SQL clé valeur

2) blast :

Utilisée pour la recherche de similitudes entre des séquences de nucléotides ou d'acides aminés

Elle a pour rôle de tenter de déterminer la fonction d'un gène ou d'une protéine par comparaison avec des s´séquences dont le rôle est bien connu

On distingue deux types d'alignements qui diffèrent suivant leur complexité

L’alignement par paires
L’alignement multiple

3) Le fonctionnement de blast :

- Découpe la requête en mots élémentaires
- Compare ces mots a ceux d’une BDD de référence, elle aussi indexée en mots élémentaires
- A partir de mots communs, sépares par une des distances similaires, essaye d’étendre l’alignement des deux côtés
- Calcul un score d’alignement