

## TP 1

### Création d'une base de données sous ORACLE

#### A-Lancer SQLPLUS

1. Connectez-vous sous linux.
2. Mettez éventuellement à jour votre fichier **.bashrc** pour permettre le lancement d'ORACLE
3. Créez un répertoire BD et allez dans ce répertoire (dans un terminal).
4. Lancez l'interface SQLPLUS d'oracle (il suffit de taper **sqlplus**).
5. Identifiez-vous avec votre **Login** et votre **Mot de Passe**.

**Il faudra garder ou modifier le mot de passe lors de la première connexion.**

#### B-Première création de table

La commande SQL : CREATE TABLE permet de créer une table, voici un exemple :

CREATE TABLE ETUDIANTS		
(	NUMERO	NUMBER(4)
	NOM	VARCHAR2(25)
	PRENOM	VARCHAR2(25)
	SEXE	CHAR(1)
	DATENAISSANCE	DATE
	POIDS	NUMBER,
	ANNEE	NUMBER);
		<b>PRIMARY KEY,</b>
		<b>NOT NULL,</b>
		<b>NOT NULL,</b>
		<b>CHECK (SEXE IN ('F', 'M')),</b>
		<b>NOT NULL,</b>

**Remarque** : cette description de la table ETUDIANTS ne suit pas la convention (norme) SQL ! Les contraintes sont mal définies.

Cette commande décrit le nom des colonnes, le type de donnée de chaque colonne et quelques contraintes à vérifier lors de l'insertion de nouvelles lignes.

1. Créez la table ETUDIANTS.
2. Vérifiez que la table est correctement créée avec « **desc nom\_table ;** » et « **select \* from nom\_table ;** ». Quelle est la différence entre ces deux commandes.
3. Quel attribut est défini comme clé primaire ici ?

4. Quelles contraintes ont été définies ?
5. On peut ajouter des lignes avec la commande « **INSERT INTO ETUDIANTS VALUES** (valeur1, valeur2, ...) ; ». Insérer les lignes (individus) suivantes :

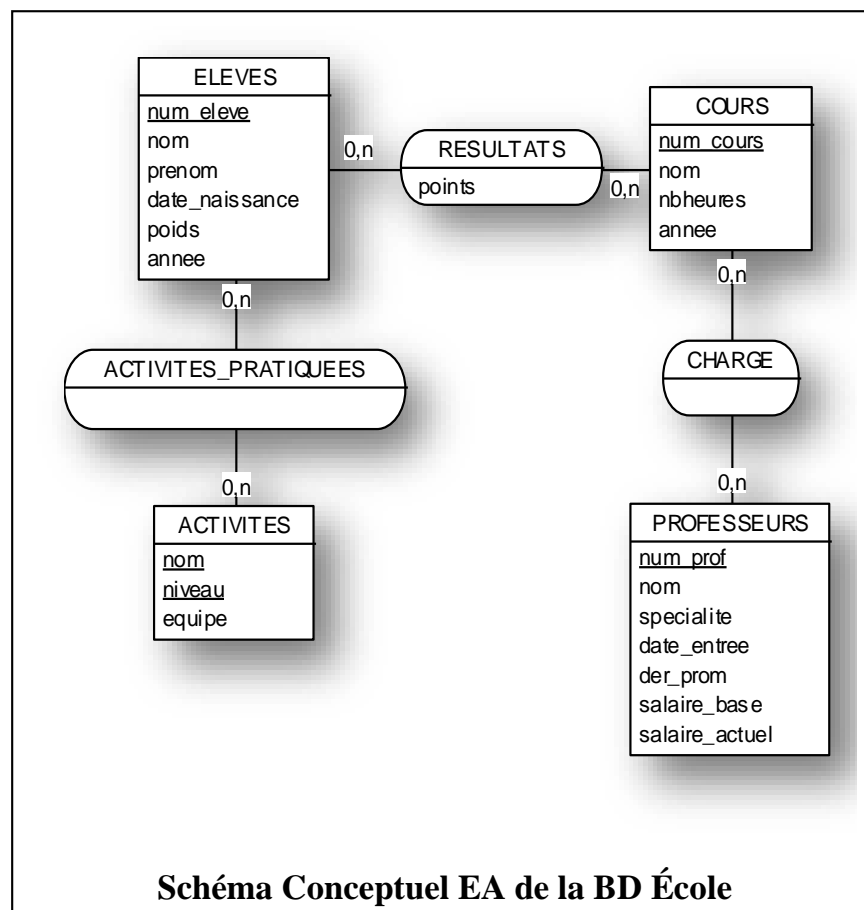
Contenu de la table : ETUDIANTS

NUMERO	NOM	PRENOM	SEXE	DATENAISSANCE	POIDS	ANNEE
71	Traifor	Benoît	M	10/12/1978	77	1
72	Génial	Clément	M	10/04/1978	72	1
73	Paris	Adam	M	28/06/1974	72	2
74	Paris	Clémence	F	20/09/1977	72	
69	Saitout	Inès	F	22/11/1969	69	2
55	Serafoub	Izouaf	M	19/09/2013	1	0

6. Essayez d'insérer de nouvelles lignes qui violent les contraintes définies pour cette table. Que se passe-t-il ? Toutes les contraintes sont bien vérifiées ?

## C- Base de données avec plusieurs tables

Soit l'exemple de gestion d'une école dont le **schéma conceptuel de données**, selon le modèle **Entité-Association (EA)**, est le suivant :



Le **schéma relationnel** est représenté par les **tables** ci-dessous :

```
ELEVES(NUM_ELEVE, NOM, PRENOM, DATE_NAISSANCE, POIDS, ANNEE)
COURS(NUM_COURS, NOM, NBHEURES, ANNEE)
PROFESSEURS(NUM_PROF, NOM, SPECIALITE, DATE_ENTREE, DER_PROM,
SALAIRE_BASE, SALAIRE_ACTUEL)
ACTIVITES (NIVEAU,NOM , EQUIPE)

RESULTATS(NUM_ELEVE, NUM_COURS, POINTS )
CHARGE(NUM_PROF, NUM_COURS)
ACTIVITES_PRATIQUEES(NUM_ELEVE, NIVEAU, NOM)
```

**Schéma relationnel de la BD École**

1. Créez les tables dans un fichier texte *ecole.sql*. N'oubliez pas les contraintes.
2. Si les tables existent déjà, il est nécessaire de supprimer les tables et leurs contraintes avant de les créer à nouveau. Pour cela utilisez « **DROP TABLE** table\_name **CASCADE CONSTRAINTS;** » dans le script.
3. Pour définir des clés primaires avec plusieurs attributs, utilisez « **CONSTRAINT** constraint\_name **PRIMARY KEY** (attribut\_1, attribut\_2, ... attribut\_n) » lors de la création de la table.
4. N'oubliez pas les clés étrangères : « **CONSTRAINT** constraint\_name **FOREIGN KEY** (attributs) **REFERENCES** table\_name (attributs) ».
5. Lancez le script avec la commande « **@ecole** », puis vérifiez que les tables ont bien été créées.
6. Ajoutez des lignes aux tables, vérifiez que les contraintes (en particulier de clés étrangères) sont bien respectées.