

28/09/2024

# TP1 Base de données Réparties

**Création de la Base de Données  
Centralisées**

**Mamadou Baïlo BARRY**

M1 MIAGE  
UNIVERSITE TOULOUSE 3 – PAUL SABATIER

## Sommaire

<b>1. Introduction .....</b>	<b>2</b>
<b>2. Scripts et résultats .....</b>	<b>2</b>
➤ <b>Scripts du fichier <i>createbase.sql</i> :.....</b>	<b>2</b>
➤ <b>Vérification du contenu de la base : .....</b>	<b>4</b>
➤ <b>En utilisant le dictionnaire de données, affichage de la : .....</b>	<b>4</b>
➤ <b>Proposition du schéma logique (relationnel) de la BD centralisée : .....</b>	<b>6</b>
➤ <b>Proposition du diagramme conceptuel (DC UML ou E/A) de la BD centralisée :.....</b>	<b>6</b>
<b>3. Conclusion .....</b>	<b>7</b>

## 1. Introduction

L'objectif de ce TP est de nous faire comprendre les différents modes de connexion à Oracle et, nous permettre aussi de mettre en place une base de données centralisée en créant les tables, insérer les données.

## 2. Scripts et résultats

### ➤ Scripts du fichier *createbase.sql* :

```
-- Suppression des tables s'ils existent --
DROP TABLE Gagner_I;
DROP TABLE Gagner_E;
DROP TABLE Pratiquer;
DROP TABLE Athlete;
DROP TABLE Discipline;
DROP TABLE Sport;
DROP TABLE Pays;

-- Création de La table Pays --
CREATE TABLE Pays(
    cio CHAR(3),
    nomp VARCHAR(30),
    ncomplet VARCHAR(50),
    CONSTRAINT pk_pays_cio PRIMARY KEY (cio)
);

-- Création de La table Sports --
CREATE TABLE Sport(
    cds CHAR(3),
    noms VARCHAR(30),
    urls VARCHAR(100),
    CONSTRAINT pk_sport_cds PRIMARY KEY (cds)
);

-- Création de La table Discipline --
CREATE TABLE Discipline (
    ndd NUMBER,
    nomd VARCHAR(50),
    typed CHAR(1),
    genred CHAR(1),
    cds CHAR(3),
    CONSTRAINT pk_discipline_ndd PRIMARY KEY (ndd),
    CONSTRAINT fk_discipline_cds FOREIGN KEY (cds) REFERENCES Sport(cds),
    CONSTRAINT chk_discipline_ndd CHECK (ndd>0),
    CONSTRAINT chk_discipline_typed CHECK (typed IN ('P', 'T')),
    CONSTRAINT chk_discipline_genred CHECK (genred IN ('M', 'W', 'X', 'O'))
);
```

```
-- Création de La table Athlete --
CREATE TABLE Athlete(
    nda NUMBER(7),
    ncomplet VARCHAR(50),
    genre CHAR(1),
    taille NUMBER(3),
    poids NUMBER(4, 1),
    daten DATE,
    villen VARCHAR(30),
    paysn VARCHAR(30),
    cio CHAR(3),
    CONSTRAINT pk_athlete_nda PRIMARY KEY (nda),
    CONSTRAINT fk_athlete_cio FOREIGN KEY (cio) REFERENCES Pays(cio),
    CONSTRAINT chk_athlete_genre CHECK (genre IN ('M', 'F')),
    CONSTRAINT chk_athlete_taille CHECK (taille >= 0),
    CONSTRAINT chk_athlete_poids CHECK (poids >= 0.0)
);

-- Création de La table Gagner une médaille Individuelle --
CREATE TABLE Gagner_I(
    ndd NUMBER,
    nda NUMBER(7),
    medaille CHAR(1),
    dateg DATE,
    CONSTRAINT pk_gagnerI PRIMARY KEY (ndd, nda),
    CONSTRAINT fk_gagnerI_ndd FOREIGN KEY (ndd) REFERENCES Discipline(ndd),
    CONSTRAINT fk_gagnerI_nda FOREIGN KEY (nda) REFERENCES Athlete(nda),
    CONSTRAINT chk_gagnerI_medaille CHECK (medaille IN ('G', 'S', 'B'))
);

-- Création de La table Gagner un médaille en Equipe --
CREATE TABLE Gagner_E(
    ndd NUMBER,
    cio CHAR(3),
    medaille CHAR(1),
    dateg DATE,
    CONSTRAINT pk_gagnerE PRIMARY KEY (ndd, cio, medaille),
    CONSTRAINT fk_gagnerE_ndd FOREIGN KEY (ndd) REFERENCES Discipline(ndd),
    CONSTRAINT fk_gagnerE_cio FOREIGN KEY (cio) REFERENCES Pays(cio),
    CONSTRAINT chk_gagnerE_medaille CHECK (medaille IN ('G', 'S', 'B'))
);

-- Création de La table Pratiquer --
CREATE TABLE Pratiquer(
    ndd NUMBER,
    nda NUMBER(7),
    CONSTRAINT pk_pratiquer PRIMARY KEY (ndd, nda),
    CONSTRAINT fk_pratiquer_ndd FOREIGN KEY (ndd) REFERENCES Discipline(ndd),
    CONSTRAINT fk_pratiquer_nda FOREIGN KEY (nda) REFERENCES Athlete(nda)
);
```

➤ **Vérification du contenu de la base :**

- **Sur « connexion etupre » :**

**Scripts :**

```
SELECT TABLE_NAME
FROM USER_TABLES
ORDER BY TABLE_NAME ASC;
```

**Résultat**

All rows fetched: 7 in 0.360 seconds

	TABLE_NAME
1	ATHLETE
2	DISCIPLINE
3	GAGNER_E
4	GAGNER_I
5	PAYS
6	PRATIQUER
7	SPORT

- **Sur « connexion etusec » et « connexion etuter » :**

On constate qu'il n'y a aucune table créée, les deux bases de données sont vides.

➤ **En utilisant le dictionnaire de données, affichage de la :**

- **Liste des tables (USER\_TABLES) :**

**Scripts :**

```
SELECT TABLE_NAME
FROM USER_TABLES
ORDER BY TABLE_NAME ASC;
```

**Résultat**

All rows fetched: 7 in 0.360 seconds

	TABLE_NAME
1	ATHLETE
2	DISCIPLINE
3	GAGNER_E
4	GAGNER_I
5	PAYS
6	PRATIQUER
7	SPORT

- **Liste des attributs (USER\_TAB\_COLUMNS) :**

**Script :**

```
SELECT TABLE_NAME, COLUMN_NAME, DATA_TYPE
FROM USER_TAB_COLUMNS
ORDER BY TABLE_NAME, COLUMN_ID ASC;
```

**Résultat**

	TABLE_NAME	COLUMN_NAME	DATA_TYPE
5	ATHLETE	POIDS	NUMBER
6	ATHLETE	DATEN	DATE
7	ATHLETE	VILLEN	VARCHAR2
8	ATHLETE	PAYSN	VARCHAR2
9	ATHLETE	CIO	CHAR
10	DISCIPLINE	NDO	NUMBER
11	DISCIPLINE	NOMD	VARCHAR2
12	DISCIPLINE	TYPED	CHAR
13	DISCIPLINE	GENRED	CHAR
14	DISCIPLINE	CDS	CHAR
15	GAGNER_E	NDO	NUMBER
16	GAGNER_E	CIO	CHAR
17	GAGNER_E	MEDAILLE	CHAR
18	GAGNER_E	DATEG	DATE
19	GAGNER_I	NDO	NUMBER
20	GAGNER_I	NDA	NUMBER
21	GAGNER_I	MEDAILLE	CHAR
22	GAGNER_I	DATEG	DATE
23	PAYS	CIO	CHAR
24	PAYS	NOMP	VARCHAR2
25	PAYS	NCOMPLET	VARCHAR2
26	PRATIQUER	NDO	NUMBER
27	PRATIQUER	NDA	NUMBER
28	SPORT	CDS	CHAR
29	SPORT	NOMS	VARCHAR2
30	SPORT	URLS	VARCHAR2

- liste des clés primaires (USER\_CONSTRAINTS & USER\_CONS\_COLUMNS), :

Script :

```
SELECT UC.TABLE_NAME, UCC.COLUMN_NAME
FROM USER_CONSTRAINTS UC
JOIN USER_CONS_COLUMNS UCC
ON UC.CONSTRAINT_NAME = UCC.CONSTRAINT_NAME
WHERE UC.CONSTRAINT_TYPE = 'P';
```

Résultat

	TABLE_NAME	COLUMN_NAME
1	PAYS	CIO
2	SPORT	CDS
3	DISCIPLINE	NDD
4	ATHLETE	NDA
5	GAGNER_I	NDA
6	GAGNER_I	NDD
7	GAGNER_E	MEDAILLE
8	GAGNER_E	CIO
9	GAGNER_E	NDD
10	PRATIQUER	NDA
11	PRATIQUER	NDD

- liste des clés étrangères (USER\_CONSTRAINTS & USER\_CONS\_COLUMNS)

Script :

```
SELECT UC.TABLE_NAME, UCC.COLUMN_NAME
FROM USER_CONSTRAINTS UC
JOIN USER_CONS_COLUMNS UCC
ON UC.CONSTRAINT_NAME = UCC.CONSTRAINT_NAME
WHERE UC.CONSTRAINT_TYPE = 'R';
```

Résultat

	TABLE_NAME	COLUMN_NAME
1	PRATIQUER	NDA
2	PRATIQUER	NDD
3	GAGNER_E	CIO
4	GAGNER_E	NDD
5	GAGNER_I	NDA
6	GAGNER_I	NDD
7	ATHLETE	CIO
8	DISCIPLINE	CDS

- liste des contraintes de domaines (USER\_CONSTRAINTS & USER\_CONS\_COLUMNS)

Script :

```
SELECT UC.TABLE_NAME, UC.SEARCH_CONDITION, UCC.COLUMN_NAME
FROM USER_CONSTRAINTS UC
JOIN USER_CONS_COLUMNS UCC
ON UC.CONSTRAINT_NAME = UCC.CONSTRAINT_NAME
WHERE UC.CONSTRAINT_TYPE = 'C';
```

## Résultat

	TABLE_NAME	SEARCH_CONDITION	COLUMN_NAME
1	DISCIPLINE	ndd>0	NDD
2	DISCIPLINE	typed IN ('P', 'T')	TYPED
3	DISCIPLINE	genred IN ('M', 'W', 'X', 'O')	GENRED
4	ATHLETE	genre IN ('M', 'F')	GENRE
5	ATHLETE	taille >= 0	TAILLE
6	ATHLETE	poids >= 0.0	POIDS
7	GAGNER_I	medaille IN ('G', 'S', 'B')	MEDAILLE
8	GAGNER_E	medaille IN ('G', 'S', 'B')	MEDAILLE

### ➤ Proposition du schéma logique (relationnel) de la BD centralisée :

**PAYS** (cio, nomp, ncomplet)

**SPORT** (cds, noms, urls)

**DISCIPLINE** (ndd, nomd, typed, genred, cds#)

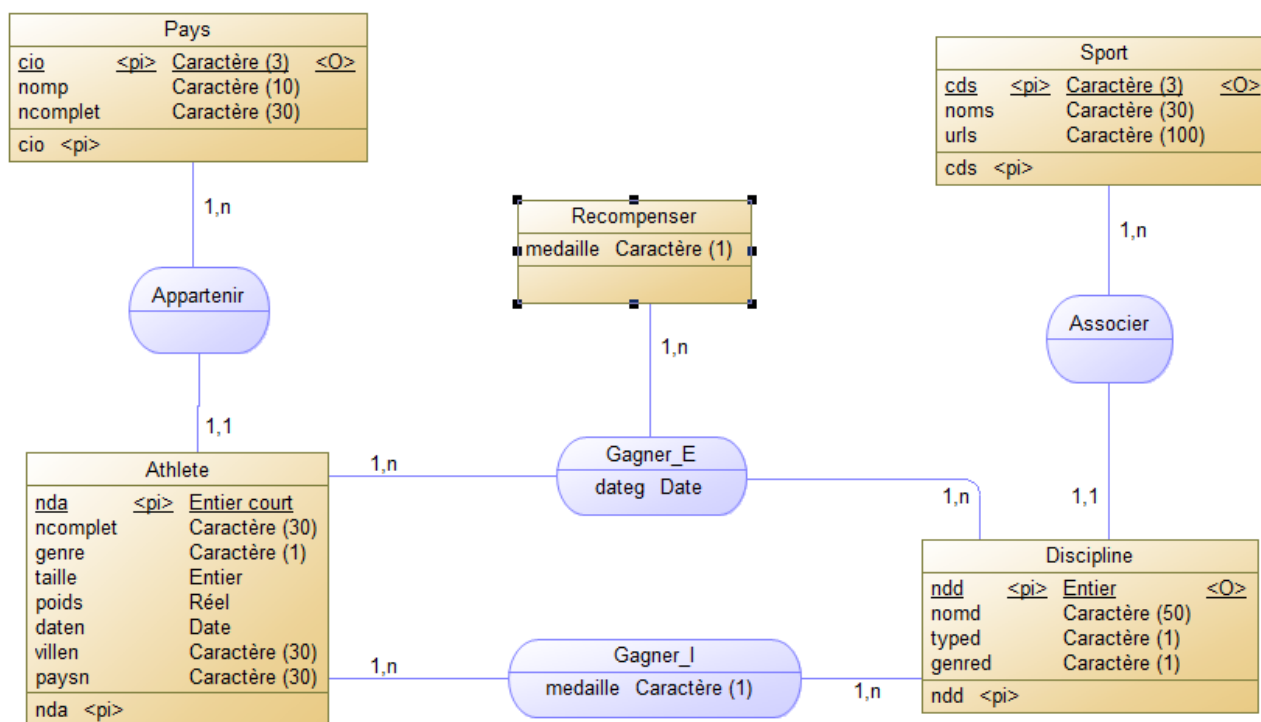
**ATHLETE** (nda, ncomplet, genre, taille, poids, daten, villen, paysn, cio#)

**GAGNER\_I** (ndd#, nda#, medaille, dateg)

**GAGNER\_E** (ndd#, cio#, medaille, dateg)

**PRATIQUER** (nda#, ndd#)

### ➤ Proposition du diagramme conceptuel (DC UML ou E/A) de la BD centralisée :



### 3. Conclusion

En conclusion, ce TP permet de rappeler les bases de la connexion à un serveur Oracle via SQL Developer et de rappeler la gestion des bases de données centralisées. Il met l'accent sur la création, la suppression de tables et l'insertion de données dans les tables. Il permet aussi de rappeler l'utilisation du dictionnaire de données permet de consulter les métadonnées, notamment les tables, les colonnes et les contraintes (primaires, étrangères et de domaines). Enfin, ce TP aborde les concepts de conception de bases de données avec un schéma logique relationnel (Modèle Relationnel de données) et un diagramme conceptuel (Modèle Conceptuel de données).