

# TRAVAILLE A RENDRE

TP1 : BASE DE DONNEES

NOM ET PRENOM :IZIKKI Hajar

PROF ENCADRANT : M. OUKDACH

## Question1 :

La connexion à oracle :

```
C:\oracle\app\oracle\product\10.2.0\server\BIN\sqlplus.exe
SQL*Plus: Release 10.2.0.1.0 - Production on Sam. D  c. 2 21:22:37 2023

Copyright (c) 1982, 2005, Oracle. All rights reserved.

Entrez le nom utilisateur : system
Entrez le mot de passe :

Connect   :
Oracle Database 10g Express Edition Release 10.2.0.1.0 - Production

SQL> alter user system identified by Hajar1234;

Utilisateur modifi  .

SQL> disconnect
```

## Question2 :

Pour cr  er un utilisateur on tape cette requ  te :

```
SQL> Create user adia identified by idia123;

Utilisateur cr    .

SQL>
```

## Question3 :

Pour le droit d'acc  s a l'utilisateur :

```
SQL> grant create table to adia;

Autorisation de privil  ges (GRANT) accept  e.
```

#### Question4:

La connexion de l'utilisateur :

```
SQL> connect adia/idia123;
Connecté.
SQL>
```

#### Question 5 :

Chaque tableau possède une clé primaire unique :

Table AUTEUR : (nauteur PRIMARY KEY )

TABLE LIVRE : (nlivre PRIMARY KEY )

TABLE MEMEBRE : (nmembre PRIMARY KEY)

TABLE POSSEDE : ( nlivre,nauteur PRIMARY KEY , fk\_nlivre et fk\_nauteur FOREIGN KEY)

TABLE PRET : ( npret PRIMARY KEY , fk\_nmembre et fk\_nlivre FOREIGN KEY)

#### Question 6 :

Pour créer les tables on utilise la commande CREATE TABLE :

```
SQL> CREATE TABLE LIVRE (
  2   nlivre varchar2(20),
  3   title varchar2(20),
  4   prix number(8,2),
  5   nbpages number(4),
  6   annee number(4),
  7   CONSTRAINT pk_nlivre PRIMARY KEY (nlivre)
  8 );

Table créée.
```

```
SQL> CREATE TABLE AUTEUR (
  2   nauteur varchar2(10),
  3   nom varchar2(20),
  4   prenom varchar2(20),
  5   nationalite varchar2(30),
  6   constraint pk_auteur PRIMARY KEY (nauteur)
  7 );

Table créée.
```

```
SQL> CREATE TABLE MEMBRE (
  2   nmembre varchar2(20),
  3   nom varchar2(10),
  4   prenom varchar2(10),
  5   dateinscription date,
  6   CONSTRAINT pk_nmembre PRIMARY KEY (nmembre)
  7 );

Table créée.
```

```
SQL> CREATE TABLE POSSEDE (
  2   nlivre varchar2(20),
  3   nauteur varchar2(20),
  4   CONSTRAINT pk_possede PRIMARY KEY (nlivre, nauteur),
  5   CONSTRAINT fk_nlivre FOREIGN KEY (nlivre) REFERENCES LIVRE(nlivre),
  6   CONSTRAINT fk_nauteur FOREIGN KEY (nauteur) REFERENCES AUTEUR(nauteur)
  7 );
```

Table créée.

```
SQL> CREATE TABLE PRET (
  2   npret varchar2(10),
  3   nmembre varchar2(10),
  4   nlivre varchar2(10),
  5   datepret date,
  6   dateretour date,
  7   CONSTRAINT pk_npret PRIMARY KEY (npret),
  8   CONSTRAINT fk_nmembre FOREIGN KEY (nmembre) REFERENCES MEMBRE(nmembre),
  9   CONSTRAINT fk_nlivree FOREIGN KEY (nlivre) REFERENCES LIVRE(nlivre)
 10 );
```

Table créée.

### Question 7 :

La vue `user_objects` contient des informations sur tous les objets appartenant à l'utilisateur connecté. Vous pouvez l'utiliser pour lister tous les objets que vous avez créés dans votre schéma utilisateur, tels que les tables, les vues, les indexes, etc.

```
SQL> select*from user_objects;
```

OBJECT_NAME	SUBOBJECT_NAME	OBJECT_ID	DATA_OBJECT_ID	OBJECT_TYPE	CREATED	LAST_DDL	TIMESTAMP	STATUS	T	G	S
SYSCATALOG		2563		SYNONYM	07/02/06	07/02/06	2006-02-07:22:12:24	VALID	N	N	N
CATALOG		2565		SYNONYM	07/02/06	07/02/06	2006-02-07:22:12:25	VALID	N	N	N
TAB		2567		SYNONYM	07/02/06	07/02/06	2006-02-07:22:12:25	VALID	N	N	N

### Question 8/9:

Pour ajouter un champ ou une colonne on utilise ALTER puis la requête ADD

```
SQL> ALTER TABLE MEMBRE  
2 ADD email varchar2(20);  
  
Table modifiée.
```

```
SQL> ALTER TABLE MEMBRE  
2 ADD adresse varchar2(20);  
  
Table modifiée.
```

### Question 10 :

Pour supprimer (DROP) :

```
SQL> ALTER TABLE PRET  
2 DROP constraint fk_nembre;  
  
Table modifiée.
```

```
SQL> ALTER TABLE PRET  
2 DROP constraint fk_nlivree;  
  
Table modifiée.
```

### Question 11 :

```
SQL> ALTER TABLE PRET  
2 ADD constraint fk_nembre FOREIGN KEY(nembre) references MEMBRE(nembre);  
  
Table modifiée.  
  
SQL> ALTER TABLE PRET  
2 ADD constraint fk_nlivree FOREIGN KEY(nlivre) references LIVRE(nlivre);
```

### Question 13 :

```
SQL> ALTER TABLE LIVRE
  2  ADD nauteur varchar2(20);

Table modifiée.

SQL> ALTER TABLE LIVRE
  2  ADD constraint fk_nauteurs FOREIGN KEY(nauteur) references AUTEUR(nauteur);

Table modifiée.
```

### Question 14 :

```
SQL> DROP TABLE POSSEDE;

Table supprimée.
```

### Question 15 :

```
SQL> ALTER TABLE LIVRE
  2  ADD constraint un_title check (title IS NOT NULL);

Table modifiée.
```

### Question 16 :

```
SQL> ALTER TABLE LIVRE
  2  ADD langue varchar2(20);

Table modifiée.

SQL> ALTER TABLE LIVRE
  2  ADD nbreexemplaire varchar2(20);

Table modifiée.
```

### Question 17 :

La contrainte d'unicité garantit qu'aucune valeur en double n'est autorisée dans la colonne spécifiée, ce qui signifie qu'aucune entrée ne peut avoir la même valeur dans cette colonne pour plusieurs lignes de la table.

```
SQL> ALTER TABLE MEMBRE
  2  ADD constraint un_nom_prenom UNIQUE (nom,prenom);

Table modifiée.
```

### Question 18 :

Pour créer un index dans une base de données Oracle, vous pouvez utiliser la commande

**CREATE INDEX**

```
SQL> CREATE INDEX in_pret ON pret (npret);
CREATE INDEX in_pret ON pret (npret)
      *
ERREUR Ó la ligne 1 :
ORA-01408: cette liste de colonnes est d'já indexée
```

```
SQL> CREATE INDEX in_membre ON PRET(nmembre);

Index créé.
```

### DEDUCTION :

On remarque que la clé primaire ne peut pas être indexée car elle est déjà indexée par défaut.

### Question 19 :

```
SQL> CREATE INDEX in_nomm ON MEMBRE(nom);

Index créé.

SQL> CREATE INDEX in_prenom ON MEMBRE(prenom);

Index créé.
```

```
SQL> CREATE INDEX in_title ON LIVRE(title);

Index créé.
```

### Question 20 :

Pour afficher les index créés dans votre schéma utilisateur en utilisant la vue **user\_indexes** dans Oracle :

```
SQL> SELECT index_name, table_name
2  FROM user_indexes
3  WHERE table_name = 'LIVRE';
```

INDEX_NAME	TABLE_NAME
PK_LIVRE	LIVRE
IN_TITLE	LIVRE

### Question 21 :

```
SQL> DROP INDEX in_nom;  
  
Index supprimé.
```

### Question 22 :

Pour ajouter une contrainte dans une table existante spécifiant que la colonne `dateRetour` doit être supérieure ou égale à la colonne `datePret`, vous pouvez utiliser une contrainte de vérification (CHECK constraint) dans une commande ALTER TABLE.

```
SQL> ALTER TABLE PRET  
2 ADD constraint ck_pret_date CHECK(dateretour >= datepret);  
  
Table modifiée.
```

### Question 23 :

Pour afficher les contraintes d'une base de données Oracle en utilisant la vue `USER_CONSTRAINTS` :

```
SQL> SELECT constraint_name FROM user_constraints;  
  
CONSTRAINT_NAME  
-----  
UN_TITLE  
FK_NAUTEURE  
FK_NMEMBRE  
FK_NLIVREE  
CK_PRET_DATE  
SYS_C002798  
SYS_C002799  
SYS_C002800  
REPCAT$_SITES_NEW_FULL_INST  
REPCAT$_SITES_NEW_FK1  
REPCAT$_SITES_NEW_FK2
```