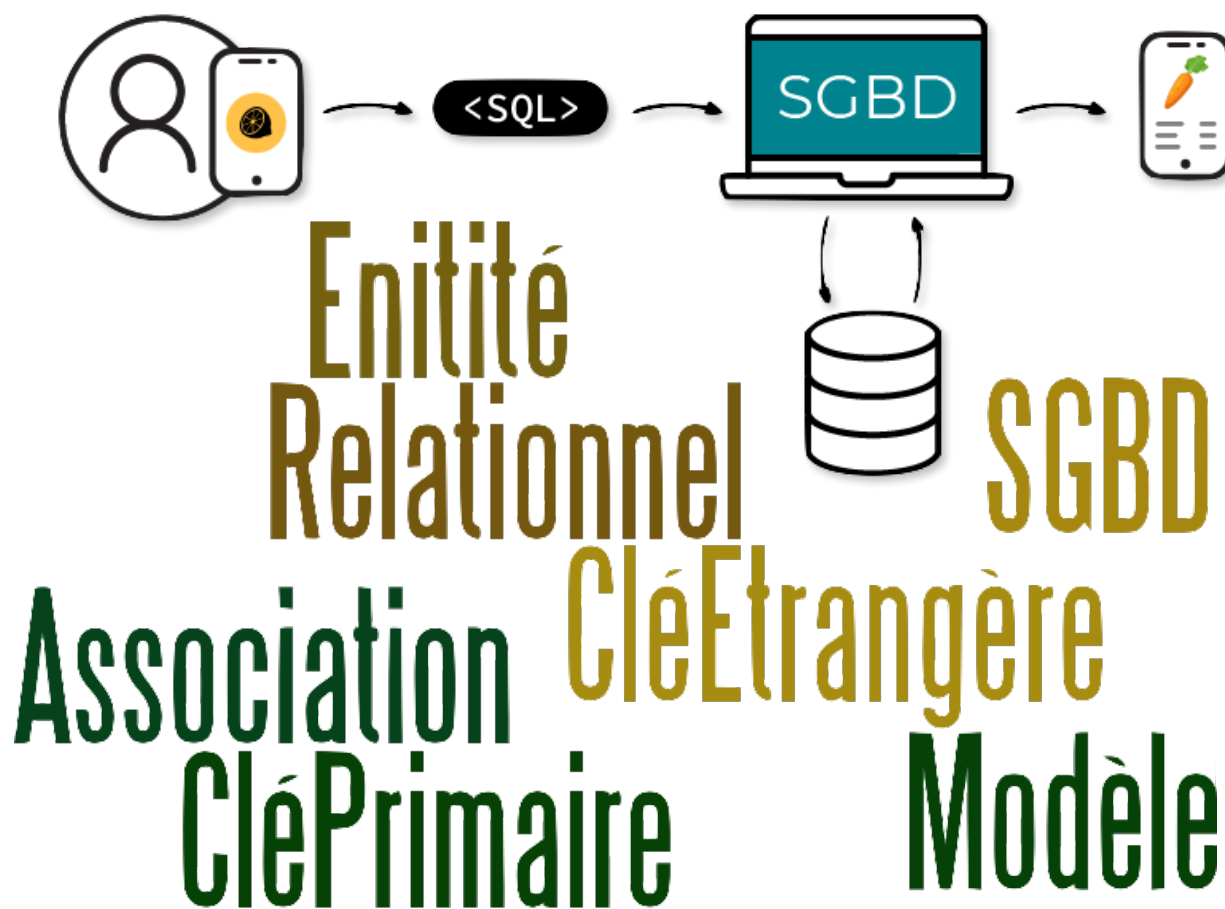


# CAHIER D'EXERCICES

Hatem Aziza



# Sommaire

TD N°1	2
TD N°2	6
TD N°3	10
TD N°4	12
TD N°5	15
TD N°6	17
TD N°7	20
TD N°8	23

## TD N°1

### Savoir Faire

- Créer des tables, leurs clés primaires et des contraintes de vérification (NOT NULL, CHECK)

La première partie des exercices (de 1 à 4) concerne la base Parc Informatique. Le dernier exercice traite la base Chantier.

### EXERCICE N°1 : Présentation de la base de données

Une entreprise désire gérer son parc informatique à l'aide d'une base de données. Le bâtiment est composé de trois étages. Chaque étage possède son réseau (ou segment distinct) Ethernet. Ces réseaux traversent des salles équipées de postes de travail. Un poste de travail est une machine sur laquelle sont installés certains logiciels. Quatre catégories de postes de travail sont recensées. (Stations Unix, terminaux X, PC Windows et PC NT). La base de données devra aussi décrire les Installations de logiciels.

Les noms et types des colonnes sont les suivants :

Colonnes	Commentaires	Types
<b>indIP</b>	Trois premiers groupes IP (exemple : 192.168.1)	VARCHAR(11)
<b>nomSegment</b>	Nom du segment	VARCHAR(20)
<b>etage</b>	Étage du segment	NUMERIC(2)
<b>nSalle</b>	Numéro de la salle	VARCHAR(7)
<b>nomSalle</b>	Nom de la salle	VARCHAR(20)
<b>nbPoste</b>	Nombre de postes de travail dans la salle	NUMERIC(2)
<b>nPoste</b>	Code du poste de travail	VARCHAR(7)
<b>nomPoste</b>	Nom du poste de travail	VARCHAR(20)
<b>ad</b>	Dernier groupe de chiffres IP (exemple : 3)	VARCHAR(3)
<b>typePoste</b>	Type du poste (Uix, TX, PCWS, PCNT)	VARCHAR(9)

<b>dateIns</b>	Date d'installation du logiciel sur le poste	DATE
<b>nLog</b>	Code du logiciel	VARCHAR(5)
<b>nomLog</b>	Nom du logiciel	VARCHAR(20)
<b>dateAch</b>	Date d'achat du logiciel	DATE
<b>version</b>	Version du logiciel	VARCHAR(7)
<b>typeLog</b>	Type du logiciel (Unix, TX, PCWS, PCNT)	VARCHAR(9)
<b>prix</b>	Prix du logiciel	NUMERIC(6,2)
<b>numIns</b>	Numéro séquentiel des installations	NUMERIC(5)
<b>dateIns</b>	Date d'installation du logiciel	DATE
<b>delai</b>	Intervalle entre achat et installation	INT
<b>typeLP</b>	Types des logiciels et des postes	VARCHAR(9)
<b>nomType</b>	Nom des types (Terminaux X, PC Windows, etc.)	VARCHAR(20)

## EXERCICE N°2 : Création des tables

Écrivez puis exécutez le script SQL de création des tables avec leurs clés primaires et les contraintes suivantes :

- Les noms des segments, des salles et des postes sont non nuls.
- Le domaine de valeurs de la colonne ad s'étend de 0 à 255.
- La colonne prix est supérieur ou égale à 0.
- La colonne dateIns est égale à la date du jour par défaut.

### Schéma des tables

Segment(indIP, nomSegment, etage)

Salle(nSalle, nomSalle, nbPoste, #indIP)

Poste(nPoste, nomPoste, #indIP, ad, typePoste, #nSalle)

Logiciel(nLog, nomLog, dateAch, version, typeLog, prix)

```
Installer(#nPoste, #nLog, numIns, dateIns, delai)  
Type(typeLP, nomType)
```

### **EXERCICE N°3 : Structure des tables**

Écrivez puis exécutez le script SQL qui affiche la description de toutes ces tables (en Utilisant DESC). Comparez avec le schéma.

### **EXERCICE N°4 : Destruction des tables**

Écrivez puis exécutez le script SOL de destruction des tables. Lancez ce script puis à nouveau celui de la création des tables.

### **EXERCICE N°5 : Schéma de la base Chantiers**

Une société désire Informatiser les visites des chantiers de ses employés. Pour définir cette base de données, une première étude fait apparaître les Informations suivantes:

- Chaque employé est modélisé par un numéro, un nom et une qualification.
- Un chantier est caractérisé par un numéro, un nom et une adresse.
- L'entreprise dispose de véhicules pour lesquels il est important de stocker pour le numéro d'immatriculation, le type (un code valant par exemple 0 pour une camionnette, 1 pour une moto et 2 pour une voiture) ainsi que le kilométrage en fin d'année.
- Le gestionnaire a besoin de connaître les distances parcourues par un véhicule pour chaque visite d'un chantier.
- Chaque jour, un seul employé sera désigné conducteur des visites d'un véhicule.
- Pour chaque visite, Il est Important de pouvoir connaître les employés transportés.

Les colonnes à utiliser sont les suivantes :

Colonnes	Commentaires	Types
<b>ki lometres</b>	Kilométrage d'un véhicule lors d'une Sortie.	NUMERIC
<b>n_conducteur</b>	Numéro de l'employé conducteur.	VARCHAR (4)
<b>n_transporte</b>	Numéro de l'employé transporté.	VARCHAR (4)

L'exercice consiste à compléter le schéma relationnel ci-après (ajout de colonnes et définition des contraintes de clé primaire et étrangère).

```
CREATE TABLE Employe (n_emp VARCHAR(4), nom_ch VARCHAR(20), qualif_emp
VARCHAR(12), CONSTRAINT pk_emp PRIMARY KEY (n_emp));

CREATE TABLE Chantier (n_chantier VARCHAR(10), nom_ch VARCHAR(10), adresse_ch
VARCHAR(15), CONSTRAINT pk_chan PRIMARY KEY(n_chantier));

CREATE TABLE Vehicule (n_vehicule VARCHAR(10), type_vehicule VARCHAR(1),
kilometrage NUMERIC, CONSTRAINT pk_vehi PRIMARY KEY(n_vehicule));

CREATE TABLE Visite (n_chantier VARCHAR(10), n_vehicule VARCHAR(10), date_jour
DATE, .....,
CONSTRAINT pk_visite PRIMARY KEY(...),
CONSTRAINT fk_depl_chantier FOREIGN KEY(n_chantier) .....,
CONSTRAINT fk_depl_vehicule FOREIGN KEY(n_vehicule) .....,
CONSTRAINT fk_depl_employe FOREIGN KEY(n_conducteur) ....);

CREATE TABLE Transporter (....
CONSTRAINT pk_transporter PRIMARY KEY(...),
CONSTRAINT fk_transp_visite FOREIGN KEY....,
CONSTRAINT fk_transp_employe FOREIGN KEY....);
```

## TD N°2

### Savoir Faire

- Insérer des données dans les tables du schéma Parc informatique et du schéma des chantiers ;
- Créer une séquence et d'insérer des données en utilisant une séquence ;
- Modifier des données.

### EXERCICE N°1 : Insertion de données

Écrivez puis exécutez le script SQL afin d'insérer les données dans les tables suivantes :

INDIP	NOMSEGMENT	ETAGE
<b>130.120.80</b>	Brin RDC	
<b>130.120.81</b>	Brin 1 <sup>er</sup> étage	
<b>130.120.82</b>	Brin 2 <sup>ème</sup> étage	

Table Segment

NSALLE	NOMSALLE	NBPOSTE	INDIP
<b>s01</b>	Salle 1	3	130.120.80
<b>s02</b>	Salle 2	2	130.120.80
<b>s03</b>	Salle 3	2	130.120.80
<b>s11</b>	Salle 11	2	130.120.81
<b>s12</b>	Salle 12	1	130.120.81
<b>s21</b>	Salle 21	2	130.120.82
<b>s22</b>	Salle 22	0	130.120.82
<b>s23</b>	Salle 23	0	130.120.82

Table Salle

NPOSTE	NOMPOSTE	INDIP	AD	TYPEPOSTE	NSALLE
--------	----------	-------	----	-----------	--------

<b>p1</b>	Poste 1	130.120.80	1	TX	s01
<b>p2</b>	Poste 2	130.120.80	2	UNIX	s01
<b>p3</b>	Poste 3	130.120.80	3	TX	s01
<b>p4</b>	Poste 4	130.120.80	4	PCWS	s02
<b>p5</b>	Poste 5	130.120.80	5	PCWS	s02
<b>p6</b>	Poste 6	130.120.80	6	UNIX	s03
<b>p7</b>	Poste 7	130.120.80	7	TX	s03
<b>p8</b>	Poste 8	130.120.81	1	UNIX	s11
<b>p9</b>	Poste 9	130.120.81	2	TX	s11
<b>p10</b>	Poste 10	130.120.81	3	UNIX	s12
<b>p11</b>	Poste 11	130.120.82	1	PCNT	s21
<b>p12</b>	Poste 12	130.120.82	2	PCWS	s21

*Table Poste*

<b>NLOG</b>	<b>NOMLOG</b>	<b>DATEACH</b>	<b>VERSION</b>	<b>TYPELOG</b>	<b>PRIX</b>
<b>log1</b>	Oracle 6	13/05/95	6.2	UNIX	3000
<b>log2</b>	Oracle 8	15/09/99	8i	UNIX	5600
<b>log3</b>	SQL Server	12/04/98	7	PCNT	2700
<b>log4</b>	Front Page	03/06/97	5	PCWS	500
<b>log5</b>	WinDev	12/05/97	5	PCWS	750
<b>log6</b>	SQL*Net		2.0	UNIX	500
<b>log7</b>	I.I.S.	12/04/02	2	PCNT	810
<b>log8</b>	DreamWeaver	21/09/03	2.0	BeOs	1400

*Table Logiciel*

<b>TYPELP</b>	<b>NOMTYPE</b>
<b>TX</b>	Terminal X-Window
<b>UNIX</b>	Système Unix
<b>PCNT</b>	PC Windows NT



<b>PCWS</b>	PC Windows
<b>NC</b>	Network Computer

Table Types

## EXERCICE N°2 : Gestion d'une séquence

Modifiez la table Installer pour que la colonne numIns soit auto incrémentale commençant à la valeur 1, d'incrément 1. (ALTER TABLE)

Insérez les enregistrements suivants :

NPOSTE	NLOG	NUMINS	DATEINS	DELAI
<b>p2</b>	log1	1	15/05/03	
<b>p2</b>	log2	2	17/09/03	
<b>p4</b>	log5	3		
<b>p6</b>	log6	4	20/05/03	
<b>p6</b>	log1	5	20/05/03	
<b>p8</b>	log2	6	19/05/03	
<b>p8</b>	log6	7	20/05/03	
<b>p11</b>	log3	8	20/04/03	
<b>p12</b>	log4	9	20/04/03	
<b>p11</b>	log7	10	20/04/03	
<b>p7</b>	log7	11	01/04/02	

Table Installer

## EXERCICE N°3 : Modification de données

- Modifiez (avec UPDATE) la colonne etage (pour l'instant nulle) de la table Segment afin d'affecter un numéro d'étage correct (0 pour le segment 130.120.80, 1 pour le segment 130. 120.81, 2 pour le segment 130. 120.82).
- Diminuez de 10% le prix des logiciels de type 'PCNT'.

## **EXERCICE N°4 : Insertion dans la base Chantiers**

Écrivez puis exécutez le script SQL afin d'insérer les données suivantes:

- une dizaine d'employés (numéros E1 à E10) en considérant diverses qualifications (OS, Assistant, Ingénieur et Architecte) ;
- quatre chantiers et cinq véhicules ;
- deux ou trois visites de différents chantiers durant trois jours ;
- la composition (de un à trois employés transportés) de chaque visite.

## TD N°3

### Savoir Faire

- Ajouter et de modifier des colonnes ;
- Renommer des tables et des colonnes ;
- Ajouter des contraintes ;

### EXERCICE N°1 : Ajout de colonnes

Écrivez un script qui contient les Instructions nécessaires pour ajouter les colonnes suivantes (avec ALTER TABLE). Le contenu de ces colonnes sera modifié ultérieurement.

Table	Nom, type et signification des nouvelles colonnes
<b>Segment</b>	nbSalle TINYINT(2) : nombre de salles par défaut = 0. nbPoste TINYINT(2) : nombre de postes par défaut = 0.
<b>Logiciel</b>	nbInstall TINYINT(2) : nombre d'installations par défaut = 0.
<b>Poste</b>	nbLog TINYINT(2) : nombre de logiciels installés par défaut = 0.

### EXERCICE N°2 : Modification de colonnes

Dans ce même script, rajouter les instructions nécessaires pour :

- augmenter la taille dans la table Salle de la colonne nomSalle (passer à VARCHAR(30)) ;
- diminuer la taille dans la table Segment de la colonne nomSegment à VARCHAR(15) ; Vérifier la structure et le contenu de chaque table avec DESCRIBE et SELECT.

### EXERCICE N°3 : Renommer des tables

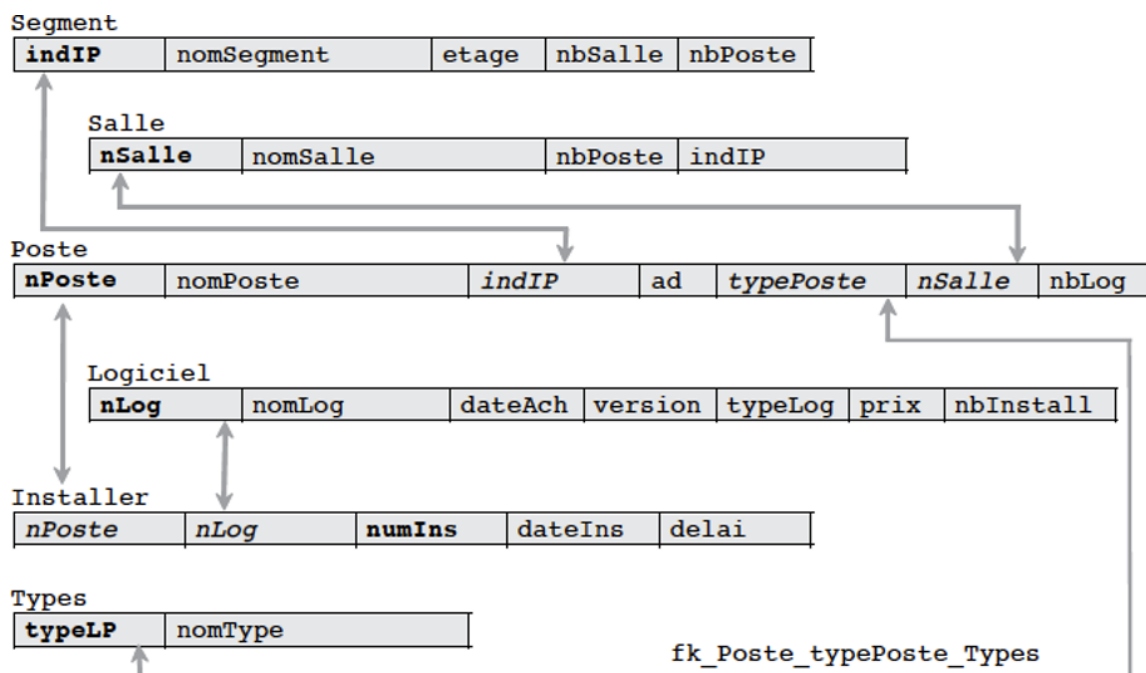
- Renommer la table **Segment** en **NewSegment** de deux façons différentes.
- Restaurer le nom de la table **Segment**

## EXERCICE N°4 : Renommer des colonnes

- Dans la table **Salle**, renommer la colonne **nbPoste** en **nombrePoste** en la positionnant avant la colonne **nomSalle**.
- Restaurer le nom et la position de la colonne **nbPoste**.

## EXERCICE N°5 : Ajout de contraintes

- Ajouter la contrainte afin de s'assurer qu'on ne puisse installer plusieurs fois le même logiciel sur un poste de travail donné.
- On suppose que les tables ne sont pas encore liées, ajouter les contraintes de clés étrangères pour assurer l'intégrité référentielle (avec ALTER TABLE... ADD CONSTRAINT...) entre les tables suivantes.
- Si l'ajout d'une contrainte référentielle renvoie une erreur, vérifier les enregistrements des tables «pères» et «fils» (notamment au niveau de la casse des chaînes de caractères, 'Tx' est différent de 'TX' par exemple).



## TD N°4

### Savoir Faire

- Créer dynamiquement des tables et leurs données ;
- Ecrire des requêtes monotables et multitable ;

### EXERCICE N°1 : Création dynamique de tables

Écrire un script permettant de créer les tables *Softs* et *PCSeuls* suivantes (en utilisant la directive *AS SELECT* de la commande *CREATE TABLE*). Vous ne poserez aucune contrainte sur ces tables. Penser à modifier le nom des colonnes.

Voici les Structures des nouvelles tables :

*Soft (nomSoft, version, prix)*

*PCSeuls (nP, nomP, seg, ad, typeP, salle)*

La table *Softs* sera construite sur la base de tous les enregistrements de la table *Logiciel* que vous avez créée et alimentée précédemment. La table *PCSeuls* doit seulement contenir les enregistrements de la table *Poste*, qui sont de type '*PCWS*' ou '*PCNT*'. Vérifier :

```
SELECT * FROM Softs;
```

```
SELECT * FROM PCSeuls;
```

### EXERCICE N°2 : Requêtes monotables

Écrire un script permettant d'extraire, à l'aide d'instructions *SELECT*, les données suivantes :

2.1. Type du poste '*p8*'.

2.2. Noms des logiciels '*UNIX*'.

2.3. Noms, adresses IP, numéros de salle des postes de type '*UNIX*' ou '*PCWS*'.

2.4. Même requête pour les postes du segment '*130.120.80*' triés par numéros de salles décroissants.

- 2.5. Numéros des logiciels installés sur le poste 'p6'.
- 2.6. Numéros des postes qui hébergent le logiciel 'log1'.
- 2.7. Noms et adresses IP complètes (ex : '130.120.80.01') des postes de type 'TX' (utiliser la fonction de concaténation).

### **EXERCICE N°3 : Fonctions et groupements**

- 3.1. Pour chaque poste, le nombre de logiciels installés (en utilisant la table Installer).
- 3.2. Pour chaque salle, le nombre de postes (à partir de la table Poste).
- 3.3. Pour chaque logiciel, le nombre d'installations sur des postes différents.
- 3.4. Moyenne des prix des logiciels 'UNIX'.
- 3.5. Plus récente date d'achat d'un logiciel.
- 3.6. Numéros des postes hébergeant 2 logiciels.
- 3.7. Nombre de postes hébergeant 2 logiciels (utiliser la requête précédente en faisant un SELECT dans la clause FROM).

### **EXERCICE N°4 : Requêtes multitables**

#### 4.1. Opérateurs ensemblistes

- 4.1.1. Types de postes non recensés dans le parc informatique (utiliser la table Types).
- 4.1.2. Types existant à la fois comme types de postes et de logiciels.
- 4.1.3. Types de postes de travail n'étant pas des types de logiciels.

#### 4.2. Jointures procédurales

- 4.2.1. Adresses IP complètes des postes qui hébergent le logiciel 'log6'.
- 4.2.2. Adresses IP complètes des postes qui hébergent le logiciel de nom 'Oracle 8'.

4.2.3. Noms des segments possédant exactement trois postes de travail de type 'TX'.

4.2.4. Noms des salles où l'on peut trouver au moins un poste hébergeant le logiciel 'Oracle 6'.

4.2.5. Nom du logiciel acheté le plus récent (utiliser la requête 3.5.).

#### 4.3. Jointures relationnelles

4.3.1. Écrire les requêtes 4.2.1, 4.2.2, 4.2.3, 4.2.4 avec des jointures de la forme relationnelle.

4.3.2. Installations (nom segment, nom salle, adresse IP complète, nom logiciel, date d'installation) triées par segment, salle et adresse IP.

#### 4.4. Jointures SQL2

4.4.1. Écrire les requêtes 4.2.1, 4.2.2, 4.2.3, 4.2.4 avec des jointures SQL2 (JOIN, NATURAL JOIN, JOIN USING).

## TD N°5

### Savoir Faire

- Réaliser des modifications synchronisées
- Composer des jointures et des divisions

### EXERCICE N°1 : Modifications synchronisées

Écrire un script pour ajouter les lignes suivantes dans la table Installer :

nPoste	nLog	dateIns	delai
<b>p2</b>	log6	Date courante	NULL
<b>p8</b>	log1	Date courante	NULL
<b>p10</b>	log1	Date courante	NULL

Rajouter au script les requêtes UPDATE synchronisées pour mettre à jour automatiquement les colonnes rajoutées :

- nbSalle dans la table Segment (nombre de salles traversées par le segment) ;
- nbPoste dans la table Segment (nombre de postes du segment) ;
- nbInstall dans la table Logiciel (nombre d'installations du logiciel) ;
- nbLog dans la table Poste (nombre de logiciels installés par poste).

### EXERCICE N°2 : Opérateurs existentiels

Écrire un script pour extraire les données suivantes :

#### 2.1. Sous-interrogation synchronisée

2.1.1. Noms des postes ayant au moins un logiciel commun au poste 'p6' (on doit trouver les postes p2, p8 et p10).

#### 2.2. Divisions

2.2.1. Noms des postes ayant les mêmes logiciels que le poste 'p6' (les postes peuvent avoir plus de logiciels que 'p6'). On doit trouver les postes 'p2' et 'p8' (division inexacte).



2.2.2. Noms des postes ayant exactement les mêmes logiciels que le poste 'p2' (division exacte), on doit trouver 'p8'.

### **EXERCICE N°3 : Extractions dans la base Chantiers**

Écrire un script pour extraire les données suivantes de la base Chantiers :

- 3.1. Numéro et nom des conducteurs qui étaient sur la route un jour donné.
- 3.2. Numéro et nom des passagers qui ont visités un chantier un jour donné.
- 3.3. Numéro des chantiers visités entre le 2 et le 3 d'un mois et d'une année donnés avec le nombre de visites pour chacun d'eux.
- 3.4. En déduire les chantiers les plus visités.
- 3.5. Nombre de visites de chaque employé (transporté ou conducteur) pour un mois donné.
- 3.6. Temps de conduite de chaque conducteur d'un mois donné.
- 3.7. Numéro du conducteur qui a fait le plus de kilométrage dans l'année avec le kilométrage total.

## TD N°6

### Savoir Faire

- Créer des rôles, des utilisateurs et des déclencheurs
- Attribuer les rôles et les privilèges

L'université YottaByte vient d'ouvrir ses portes et vous avez été chargé de créer et de gérer la base de données de l'université. Voici un ensemble de tâches que vous devez accomplir pour mettre en place le système de gestion de l'université:

#### 1. Création de la base de données et des tables:

- Créez une base de données nommée "universite".
- Créez les tables suivantes sans contraintes: "etudiant", "matiere", "enseignant" et "salle". Les détails de chaque table sont les suivants:
  - "etudiant" avec les colonnes "id", "nom", "prenom" et "date\_naissance".
  - "matiere" avec les colonnes "id", "nom" et "credit".
  - "enseignant" avec les colonnes "id", "nom", "prenom" et "specialite".
  - "salle" avec les colonnes "id", "numero" et "capacite".

#### 2. Ajout des champs des clés étrangères dans les tables:

- Ajoutez les colonnes suivantes avec les clés étrangères aux tables correspondantes:
  - "id\_matiere" à la table "etudiant".
  - "id\_enseignant" à la table "matiere".
  - "id\_matiere" à la table "salle".

3. Ajout des contraintes aux tables avec les clés étrangères:
  - Ajoutez des contraintes de clé étrangère aux colonnes correspondantes des tables "etudiant", "matiere" et "salle".
4. Création des rôles:
  - Créez trois rôles distincts: "gestionnaire", "enseignant" et "etudiant".
5. Affectation des privilèges aux rôles:
  - Accordez tous les privilèges sur la base de données "universite" au rôle "gestionnaire".
  - Accordez les privilèges SELECT, INSERT et UPDATE sur la base de données "universite" au rôle "enseignant".
  - Accordez le privilège SELECT sur la base de données "universite" au rôle "etudiant".
6. Création des utilisateurs:
  - Créez trois utilisateurs distincts: "gestion", "ens" et "etu", chacun avec un mot de passe approprié.
7. Attribution des rôles aux utilisateurs:
  - Attribuez le rôle "gestionnaire" à l'utilisateur "gestion".
  - Attribuez le rôle "enseignant" à l'utilisateur "ens".
  - Attribuez le rôle "etudiant" à l'utilisateur "etu".
8. Vérification des rôles attribués aux utilisateurs:
  - Vérifiez les privilèges accordés à chaque utilisateur à l'aide de la commande "SHOW GRANTS".

9. Activation des rôles pour les utilisateurs:

- Activez les rôles pour les utilisateurs en utilisant la commande "SET DEFAULT ROLE ALL TO".

10. Remplissage des tables avec des données factices:

- Insérez des données dans les tables "etudiant", "matiere", "enseignant" et "salle" pour simuler les informations d'étudiants, de matières, d'enseignants et de salles.

11. Révocation des privilèges de gestionnaire:

- Révoquez les privilèges de gestionnaire pour l'utilisateur "gestion".

## TD N°7

### Savoir Faire

- Créer des transactions

Vous êtes chargé de mettre en place le système de gestion des comptes bancaires pour la banque BM2I. Utilisez la base de données fournie et complétez les différentes étapes pour mettre en place ce système.

1. Créez la base de données :
  - Nommez la base de données "BM2I".
2. Créez les tables nécessaires :
  - Créez les tables "Clients", "Comptes" et "Transactions" avec les champs appropriés selon le schéma suivant:
    - Clients (client\_id, nom, prenom, solde)
    - Comptes (compte\_id, #client\_id, numero\_compte, solde)
    - Transactions (transaction\_id, #compte\_source\_id, #compte\_destination\_id, montant, date\_transaction)
3. Insérez des données initiales :
  - Ajoutez au moins trois clients avec leurs informations dans la table "Clients".
  - Ajoutez quelques comptes pour ces clients dans la table "Comptes", en associant chaque compte à un client.
  - Assurez-vous d'avoir suffisamment de données pour effectuer des opérations de virement.
4. Effectuez un virement entre deux comptes :
  - Utilisez une transaction pour effectuer un virement entre deux comptes.

- Choisissez deux comptes existants et définissez un montant pour le virement.
  - Débitez le montant du compte source, créditez le montant sur le compte destination et enregistrez la transaction dans la table "Transactions".
  - Assurez-vous que le virement est effectué de manière atomique en utilisant une transaction.
5. Affichez les détails des comptes :
- Écrivez une requête pour afficher les détails de tous les comptes des clients, y compris leur solde actuel.
  - Vérifiez que les soldes des comptes sont mis à jour correctement après le virement effectué.
6. Affichez l'historique des transactions :
- Écrivez une requête pour afficher l'historique des transactions effectuées, y compris les détails des comptes source et destination, le montant et la date de chaque transaction.
7. Ajoutez un nouveau client :
- Ajoutez un nouveau client à la base de données en insérant ses informations dans la table "Clients".
  - Assurez-vous que le nouveau client dispose également d'un compte associé dans la table "Comptes".
8. Effectuez un autre virement entre comptes :
- Répétez l'étape 4 pour effectuer un autre virement entre deux comptes différents.
  - Utilisez à nouveau une transaction pour garantir la cohérence des données.
9. Affichez à nouveau les détails des comptes :

- Répétez l'étape 5 pour afficher à nouveau les détails de tous les comptes des clients et vérifiez que les soldes ont été mis à jour après le deuxième virement.

10. Supprimez un client :

- Supprimez un client existant de la base de données, en supprimant également tous ses comptes associés.
- Assurez-vous que cette opération est bien exécutée dans le cadre d'une transaction pour maintenir la cohérence des données.

## TD N°8

### Savoir Faire

- Créer des procédures, des fonctions et des déclencheurs

#### EXERCICE N°1 :

Procédure pour insérer un nouveau segment.

```
InsertSegment(IN p_indIP VARCHAR(11), IN p_nomSegment  
VARCHAR(20), IN p_etage NUMERIC(2))
```

#### EXERCICE N°2 :

Procédure pour supprimer une salle spécifique.

```
DeleteSalle(IN p_nSalle VARCHAR(7))
```

#### EXERCICE N°3 :

Fonction pour calculer le prix total des logiciels installés sur un poste.

```
CalculateTotalPrice(nPosteArg VARCHAR(7)) RETURNS DECIMAL(10,2)
```

#### EXERCICE N°4 :

Fonction pour obtenir le nombre de postes dans une salle.

```
CountPostesInSalle(nSalleArg VARCHAR(7)) RETURNS INT
```

#### EXERCICE N°5 :

Déclencheur pour mettre à jour la date d'installation lorsqu'un nouvel enregistrement est inséré dans la table Installer.

#### EXERCICE N°6 :

Déclencheur pour supprimer automatiquement un poste de la table Installer lorsqu'il est supprimé de la table Poste.