***Rapport du mini-projet :***

Gestion d’ecole

module : base de données avancée

***Encadré par :***

* Mme. Karima KHALIL

**Année universitaire : 2020/2021**

***Présenté par :***

* Aimrane ESSAKHI

SOMMAIRE

[Introduction 5](#_Toc59362770)

[Création et manipulation de la base de données (requêtes SQL) : 6](#_Toc59362771)

[** Création des tables : 6**](#_Toc59362772)

[*Table employees : 6*](#_Toc59362773)

[*Table etudiants : 6*](#_Toc59362774)

[*Table emplois : 7*](#_Toc59362775)

[*Table filiers : 7*](#_Toc59362776)

[*Table clubs : 7*](#_Toc59362777)

[** Inclure des contraintes : 8**](#_Toc59362778)

[*Table employees : 8*](#_Toc59362779)

[*Table etudiants : 9*](#_Toc59362780)

[*Table emplois : 9*](#_Toc59362781)

[*Table filiers : 10*](#_Toc59362782)

[*Table clubs : 10*](#_Toc59362783)

[** Vérification de la création des tables : 11**](#_Toc59362784)

[*Table employees : 11*](#_Toc59362785)

[*Table etudiants : 11*](#_Toc59362786)

[*Table emplois : 11*](#_Toc59362787)

[*Table filiers : 12*](#_Toc59362788)

[*Table clubs : 12*](#_Toc59362789)

[** Insertion des données : 13**](#_Toc59362790)

[*Table employees : 13*](#_Toc59362791)

[*Table etudiants : 14*](#_Toc59362792)

[*Table emplois : 16*](#_Toc59362793)

[*Table filiers : 16*](#_Toc59362794)

[*Table clubs : 17*](#_Toc59362795)

[** Ecriture des instructions SQL SELECT élémentaire : 18**](#_Toc59362796)

[*Table employees : 18*](#_Toc59362797)

[*Table etudiants : 19*](#_Toc59362798)

[*Table emplois : 21*](#_Toc59362799)

[*Table filiers : 21*](#_Toc59362800)

[*Table clubs : 22*](#_Toc59362801)

[** Limiter et trier des données : 23**](#_Toc59362802)

[*Table employees : 23*](#_Toc59362803)

[*Table etudiants : 24*](#_Toc59362804)

[*Table emplois : 25*](#_Toc59362805)

[*Table filiers : 26*](#_Toc59362806)

[*Table clubs : 26*](#_Toc59362807)

[** Afficher les données issues de plusieurs tables : 27**](#_Toc59362808)

[** Utilisation des fonctions monoligne : 29**](#_Toc59362809)

[** Utilisation des fonctions de group et des sous interrogations : 31**](#_Toc59362810)

[** Utilisation des vues : 32**](#_Toc59362811)

[*VIEW enseignants : 32*](#_Toc59362812)

[*VIEW etudiants\_GI : 32*](#_Toc59362813)

[*VIEW pauvre\_employees : 32*](#_Toc59362814)

[** Inclure des contraintes : 33**](#_Toc59362815)

[*VIEW Enseignants : 33*](#_Toc59362816)

[*VIEW etudiants\_GI : 33*](#_Toc59362817)

[*VIEW pauvre\_employees : 33*](#_Toc59362818)

[** Vérification de la création des views : 34**](#_Toc59362819)

[*VIEW Enseignants : 34*](#_Toc59362820)

[*VIEW etudiants\_GI : 34*](#_Toc59362821)

[*VIEW pauvre\_employees : 34*](#_Toc59362822)

[** Vérification de l’affichage des views : 35**](#_Toc59362823)

[*VIEW Enseignants : 35*](#_Toc59362824)

[*VIEW etudiants\_GI : 35*](#_Toc59362825)

[*VIEW pauvre\_employees : 35*](#_Toc59362826)

[Conclusion 36](#_Toc59362827)

**Liste des figures**

[Figure 1 Création de la table employees 5](#_Toc58720072)

[Figure 2 Création de la table etudiants 5](#_Toc58720073)

[Figure 3 Création de la table emplois 6](#_Toc58720074)

[Figure 4 Création de la table filiers 6](#_Toc58720075)

[Figure 5 Création de la table clubs 6](#_Toc58720076)

[Figure 6 ajouter des contraintes à la table employees à l'aide de la clause ADD 7](#_Toc58720077)

[Figure 7 ajouter des contraintes à la table employees à l'aide de la clause MODIFY 7](#_Toc58720078)

[Figure 8 ajouter des contraintes à la table client à l'aide de la clause ADD 8](#_Toc58720079)

[Figure 9 ajouter des contraintes à la table etudiants à l'aide de la clause MODIFY 8](#_Toc58720080)

[Figure 10 ajouter des contraintes à la table emplois à l'aide de la clause ADD 8](#_Toc58720081)

[Figure 11 ajouter des contraintes à la table emplois à l'aide de la clause MODIFY 8](#_Toc58720082)

[Figure 12 ajouter des contraintes à la table filiers à l'aide de la clause ADD 9](#_Toc58720083)

[Figure 13 ajouter des contraintes à la table filiers à l'aide de la clause MODIFY 9](#_Toc58720084)

[Figure 14 ajouter des contraintes à la table clubs à l'aide de la clause ADD 9](#_Toc58720085)

[Figure 15 ajouter des contraintes à la table clubs à l'aide de la clause MODIFY 9](#_Toc58720086)

[Figure 16 vérification de la création de la table employees à l'aide de la clause DESCRIBE 10](#_Toc58720087)

[Figure 17 vérification de la création de la table etudiants à l'aide de la clause DESCRIBE 10](#_Toc58720088)

[Figure 18 vérification de la création de la table emplois à l'aide de la clause DESCRIBE 10](#_Toc58720089)

[Figure 19 vérification de la création de la table filiers à l'aide de la clause DESCRIBE 11](#_Toc58720090)

[Figure 20 vérification de la création de la table clubs à l'aide de la clause DESCRIBE 11](#_Toc58720091)

[Figure 21 insertion des données dans la table employees 12](#_Toc58720092)

[Figure 22 insertion des données dans la table etudiants 14](#_Toc58720093)

[Figure 23 Création de la table emplois 15](#_Toc58720094)

[Figure 24 Création de la table filiers 15](#_Toc58720095)

[Figure 25 Création de la table clubs 16](#_Toc58720096)

[Figure 26 requête : sélection de tous les champs de la table employees 17](#_Toc58720097)

[Figure 27 requête : sélection des champs de la table employees en utilisant la concaténation 17](#_Toc58720098)

[Figure 28 requête : sélection de tous les champs de la table etudiants 18](#_Toc58720099)

[Figure 29 requête : sélection des champs de la table employees en utilisant la clause DISTINCT 18](#_Toc58720100)

[Figure 30 requête : sélection des champs de la table etudiants en utilisant la concaténation et les alias 19](#_Toc58720101)

[Figure 31 requête : sélection de tous les champs de la table emplois 20](#_Toc58720102)

[Figure 32 requête : sélection de quelque champ en utilisant les opérateurs arithmétiques et les alias 20](#_Toc58720103)

[Figure 33 requête : sélection de tous les champs de la table filiers 20](#_Toc58720104)

[Figure 34 requête : sélection de tous les champs de la table clubs 21](#_Toc58720105)

[Figure 35 requête : sélection des données limités en utilisant les conditions de comparaison 21](#_Toc58720106)

[Figure 36 requête : sélection des données limitées de la table employees en utilisant les condition de comparaison et les fonction de manipulation des majiscules/miniscules. 22](#_Toc58720107)

[Figure 37 requête : sélection des données limitées de la table employees en utilisant les conditions de comparaison 22](#_Toc58720108)

[Figure 38 requête : sélection des données de la table etudiants en utilisant NOT NULL 23](#_Toc58720109)

[Figure 39 requête : sélection des données limitées et triées en utilisant les conditions de comparaison 24](#_Toc58720110)

[Figure 40 requête : sélection des données limitées et triées dans la table commande 24](#_Toc58720111)

[Figure 41 requête : sélection des données triées de la table filiers en utilisant les conditions de comparaison 25](#_Toc58720112)

[Figure 42 requête : sélection des données de la table filiers en utilisant les conditions logiques 25](#_Toc58720113)

[Figure 43 requête : sélection des données limitées de la table clubs en utilisant les conditions logiques 25](#_Toc58720114)

[Figure 44 requête : sélection des données de la table employees et filiers en utilisant l'équijointure 26](#_Toc58720115)

[Figure 45 requête : Extraire des enregistrements de la table commande et produit à l'aide 27](#_Toc58720116)

[Figure 46 requête : jointure RIGHT OUTER JOIN 28](#_Toc58720117)

[Figure 47 requête : sélection des données de la table employees en utilisant les foncions de date 29](#_Toc58720118)

[Figure 48 requête : sélection des données de la table employees et filiers en utilisant l'équijointure et la conversion explicite de type de données 29](#_Toc58720119)

[Figure 49 requête : utilisation de l'expression conditionnelle CASE 30](#_Toc58720120)

[Figure 50 requête : utilisation de l'expression conditionnelle DECODE 30](#_Toc58720121)

[Figure 51 requête : utilisation de la fonction de group COUNT 31](#_Toc58720122)

[Figure 52 requête : utilisation de fonction de group COUNT 31](#_Toc58720123)

[Figure 53 requête : utilisation de fonction de group COUNT et limiter les données en utilisant les sous interrogations 31](#_Toc58720124)

[Figure 54 requête : utilisation de fonction de group et conversion explicite de type de données 31](#_Toc58720125)

[Figure 55 Création de la view enseignants 32](#_Toc58720126)

[Figure 56 Création de la view etudiants\_GI 32](#_Toc58720127)

[Figure 57 Création de la view complexe pauvre\_employees 32](#_Toc58720128)

[Figure 58 ajouter contraintes à la view Enseignants à l'aide de la clause CREATE OR REPLACE VIEW 33](#_Toc58720129)

[Figure 59 ajouter contraintes à la view etudiants\_GI à l'aide de la clause CREATE OR REPLACE VIEW 33](#_Toc58720130)

[Figure 60 ajouter contraintes à la view pauvre\_employees à l'aide de la clause CREATE OR REPLACE VIEW 33](#_Toc58720131)

[Figure 61 vérification de la création de la view Enseignants à l'aide de la clause DESCRIBE 34](#_Toc58720132)

[Figure 62 vérification de la création de la view etudiants\_GI à l'aide de la clause DESCRIBE 34](#_Toc58720133)

[Figure 63 vérification de la création de la view pauvre\_employees à l'aide de la clause DESCRIBE 34](#_Toc58720134)

# **Introduction**

Le SQL (Structured Query Language) est un langage permettant de communiquer avec une base de données. Ce langage informatique est notamment très utilisé par les développeurs web pour communiquer avec les données d’un site web.

Ce mini projet a pour objectif de faire une base de données de l’école AL WIFAK à l'aide d’Oracle.

Oracle Database est un système de gestion de base de données (SGBD) fourni par Oracle Corporation. C’est l’un des SGBD les plus populaire au niveau mondial. Il fonctionne sur un système relationnel et même sur un modèle relationnel objet depuis la version 8.

# **Création et manipulation de la base de données (requêtes SQL) :**

### Création des tables :

Une table : est un objet de base de données, est une unité de stockage élémentaire, composée de lignes et de colonnes.

### Table employees :

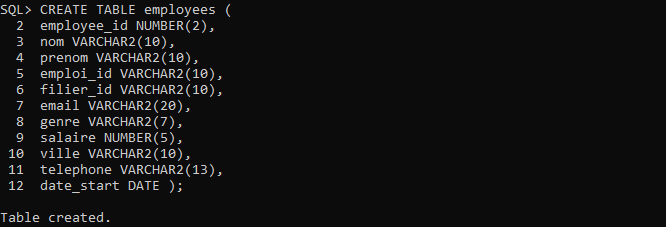


Figure 1 Création de la table employees

### Table etudiants :

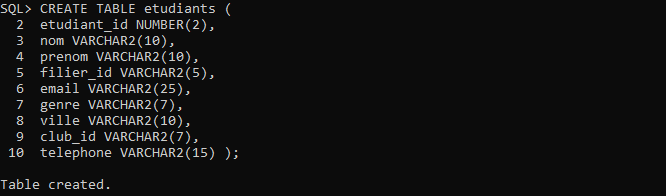


Figure 2 Création de la table etudiants

### Table emplois :



Figure 3 Création de la table emplois

### Table filiers :



Figure 4 Création de la table filiers

### Table clubs :



Figure 5 Création de la table clubs

### Inclure des contraintes :

Une contrainte d'intégrité est une clause permettant de contraindre la modification de tables, faite par l'intermédiaire de requêtes d'utilisateurs, afin que les données saisies dans la base soient conformes aux données attendues. Ces contraintes doivent être exprimées dès la création de la table. Les types de constraints:

* NOT NULL
* UNIQUE
* PRIMARY KEY
* FOREIGN KEY
* CHECK

### Table employees :

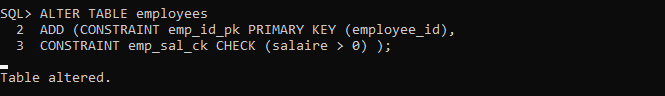
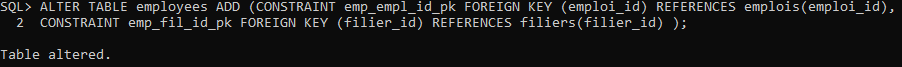
 

Figure 6 ajouter des contraintes à la table employees à l'aide de la clause ADD

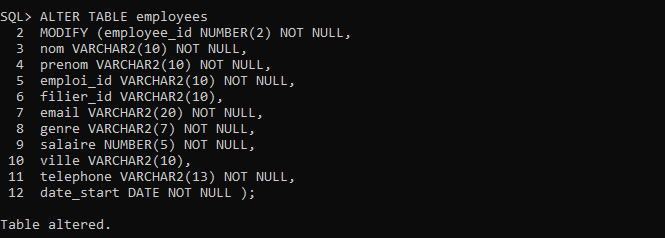


Figure 7 ajouter des contraintes à la table employees à l'aide de la clause MODIFY

### Table etudiants :

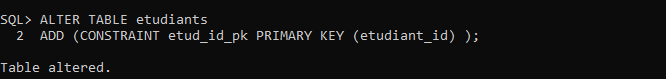
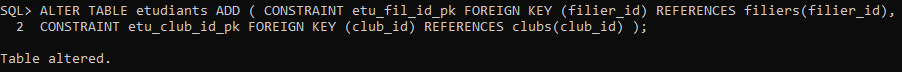
 

Figure 8 ajouter des contraintes à la table client à l'aide de la clause ADD

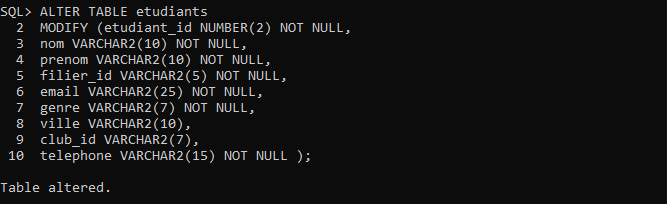


Figure 9 ajouter des contraintes à la table etudiants à l'aide de la clause MODIFY

### Table emplois :

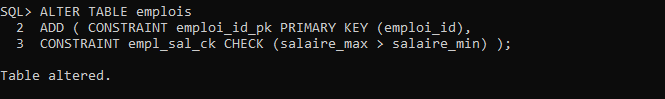


Figure 10 ajouter des contraintes à la table emplois à l'aide de la clause ADD

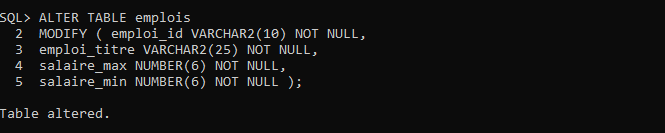


Figure 11 ajouter des contraintes à la table emplois à l'aide de la clause MODIFY

### Table filiers :

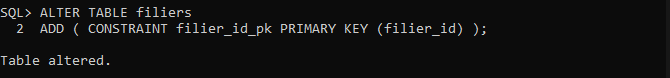


Figure 12 ajouter des contraintes à la table filiers à l'aide de la clause ADD

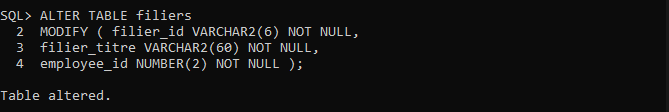


Figure 13 ajouter des contraintes à la table filiers à l'aide de la clause MODIFY

### Table clubs :

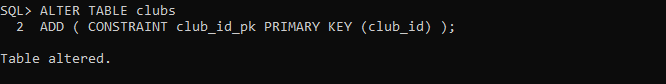


Figure 14 ajouter des contraintes à la table clubs à l'aide de la clause ADD

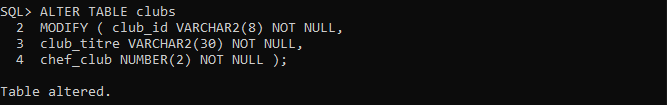


Figure 15 ajouter des contraintes à la table clubs à l'aide de la clause MODIFY

### Vérification de la création des tables :

### Table employees :

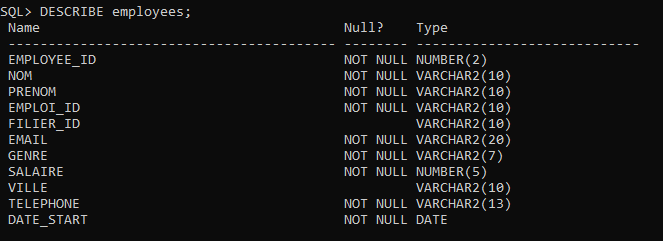


Figure 16 vérification de la création de la table employees à l'aide de la clause DESCRIBE

### Table etudiants :

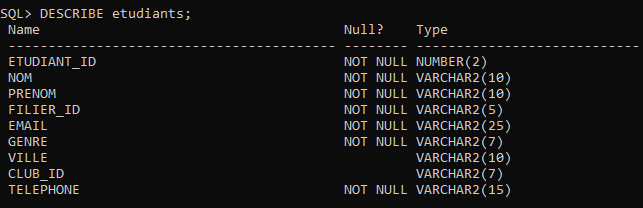


Figure 17 vérification de la création de la table etudiants à l'aide de la clause DESCRIBE

### Table emplois :

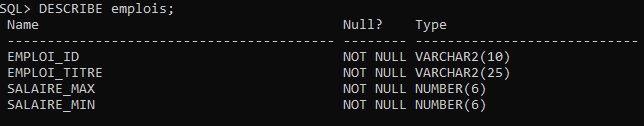


Figure 18 vérification de la création de la table emplois à l'aide de la clause DESCRIBE

### Table filiers :

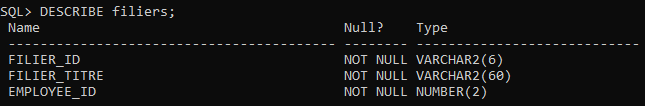


Figure 19 vérification de la création de la table filiers à l'aide de la clause DESCRIBE

### Table clubs :

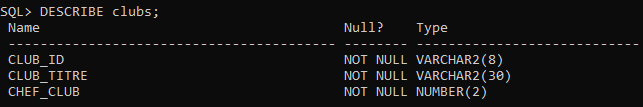


Figure 20 vérification de la création de la table clubs à l'aide de la clause DESCRIBE

### Insertion des données :

L’insertion des données faite à l’aide de syntaxe suivant : SELECT INTO table\_name VALUES …

### Table employees :

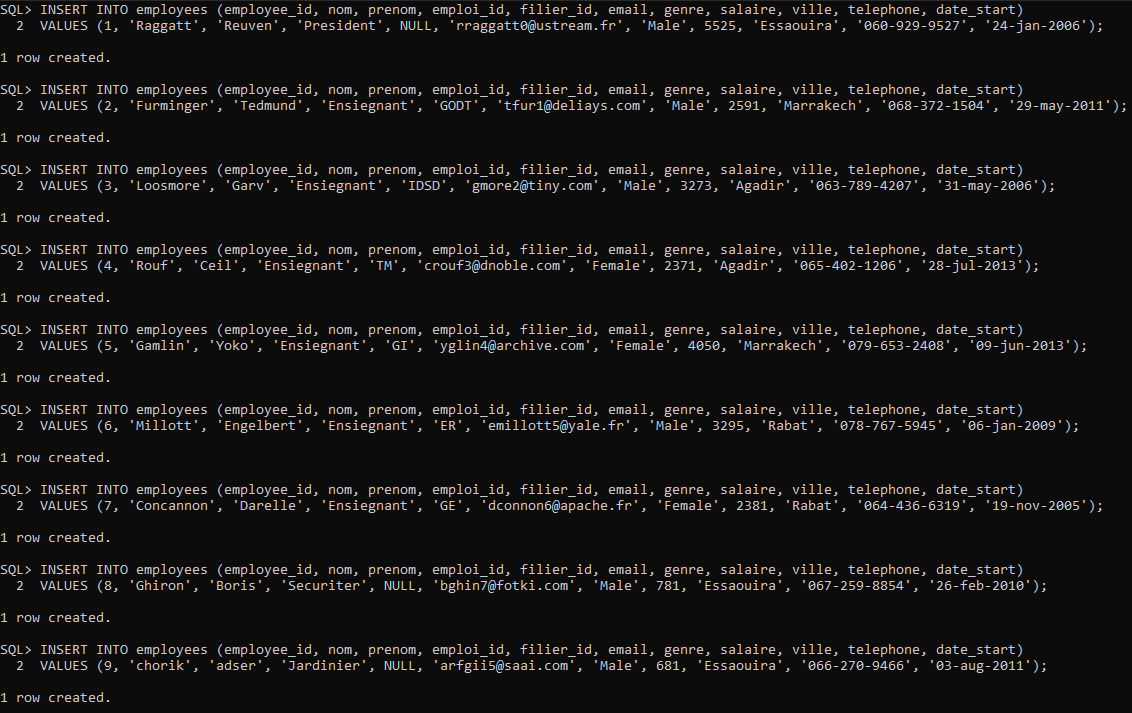


Figure 21 insertion des données dans la table employees

### Table etudiants :

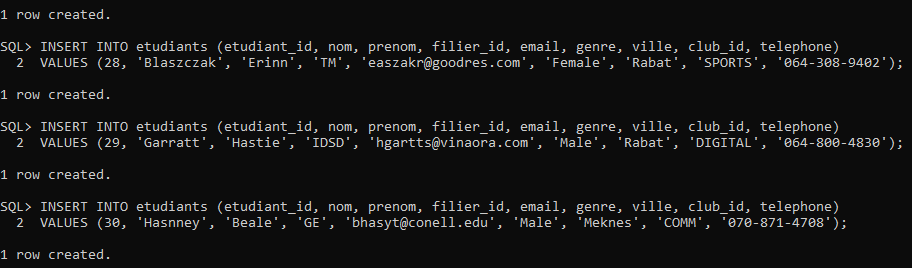
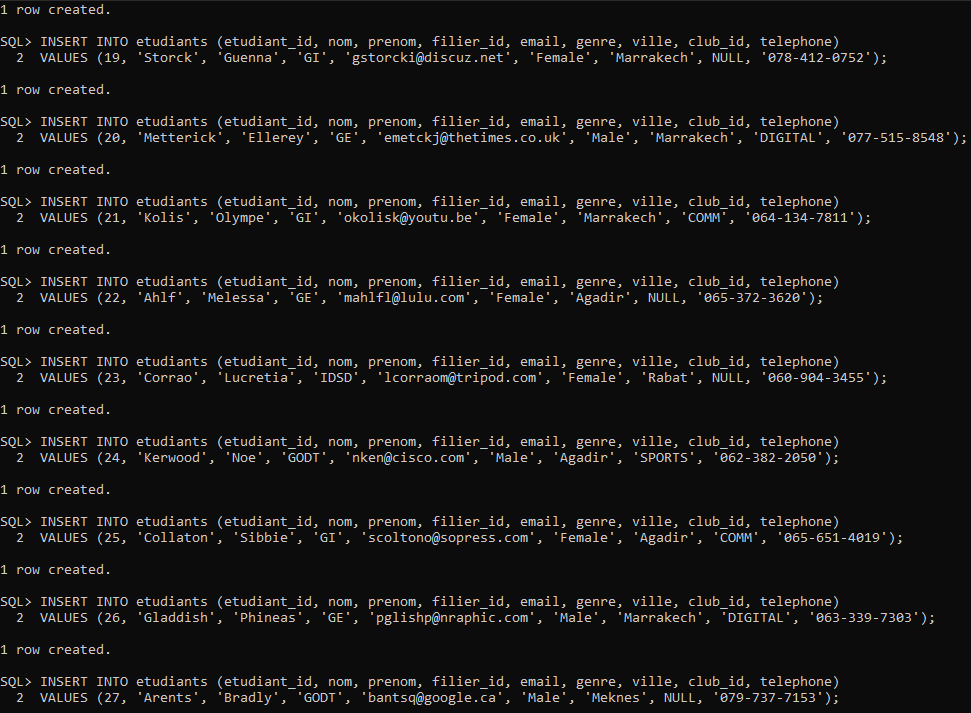
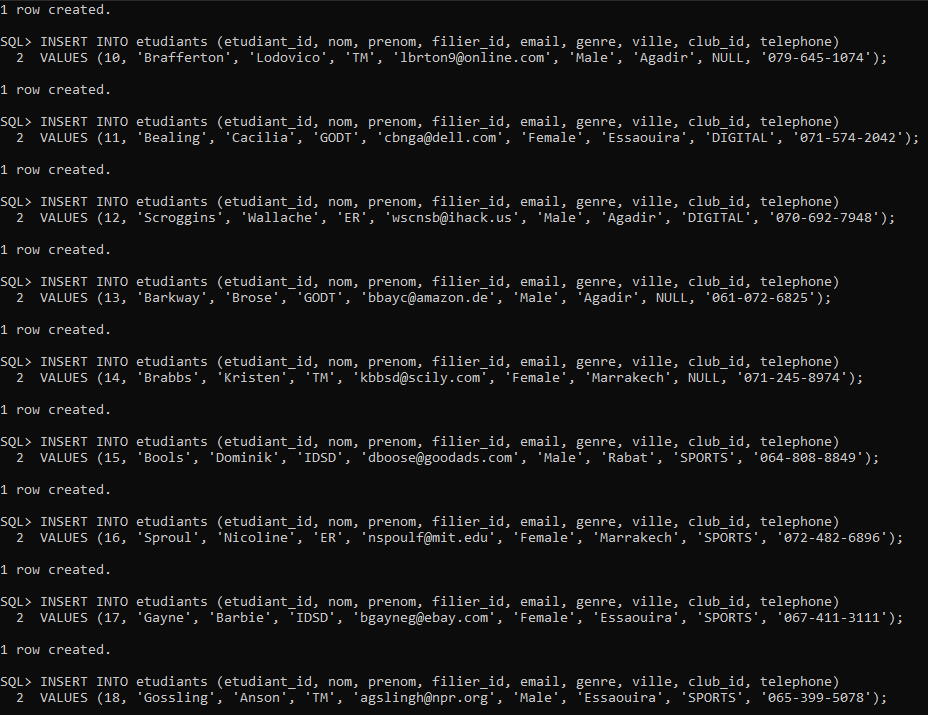
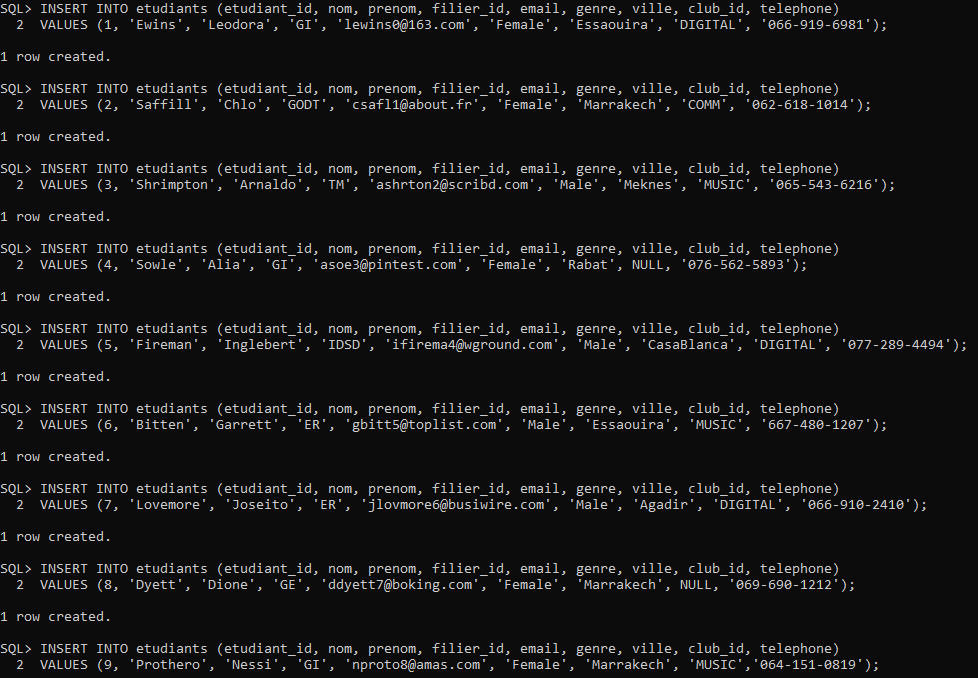


Figure 22 insertion des données dans la table etudiants

### Table emplois :

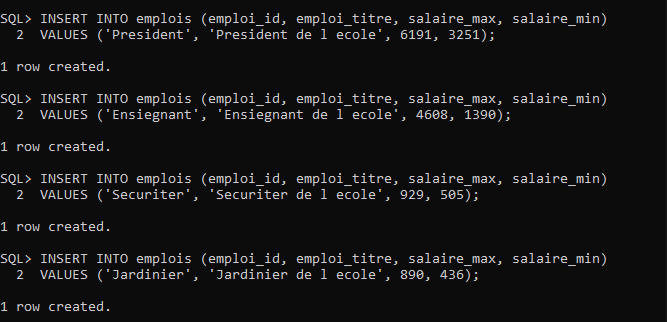


Figure 23 Création de la table emplois

### Table filiers :

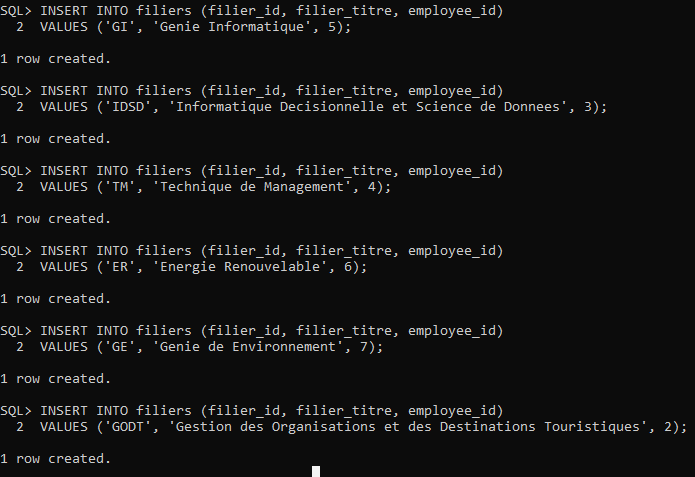


Figure 24 Création de la table filiers

### Table clubs :

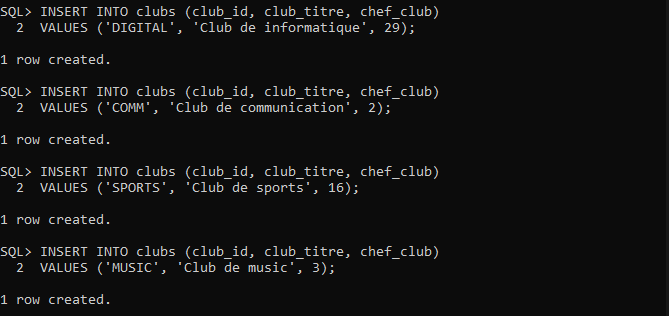


Figure 25 Création de la table clubs

### Ecriture des instructions SQL SELECT élémentaire :

L’instruction SQL SELECT permet d’extraire des données de la base.

### Table employees :

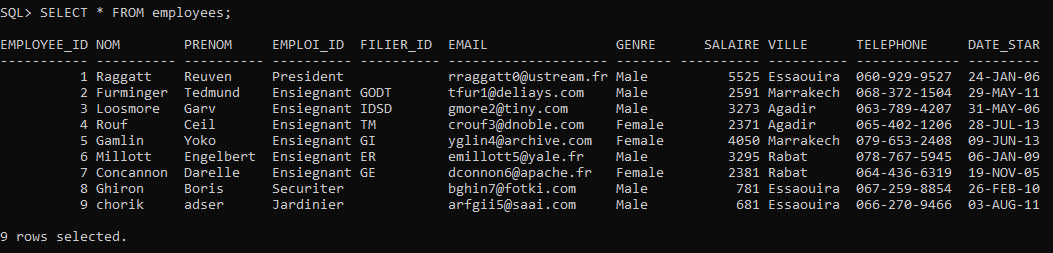
Cette requête permet de sélectionner toutes les colonnes de la table (employees).

Figure 26 requête : sélection de tous les champs de la table employees

Cette requête permet de sélectionner le nom concaténé avec le prénom de tous les employees, sous l’alias : « Employee ».

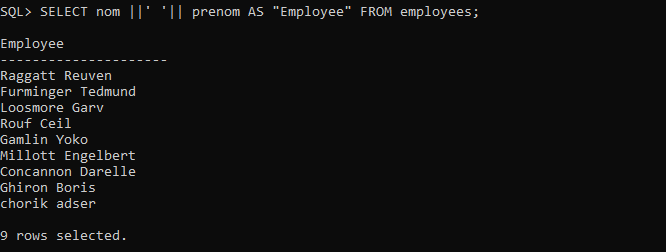


Figure 27 requête : sélection des champs de la table employees en utilisant la concaténation

### Table etudiants :

Cette requête permet de sélectionner toutes les colonnes de la table (etudiants).

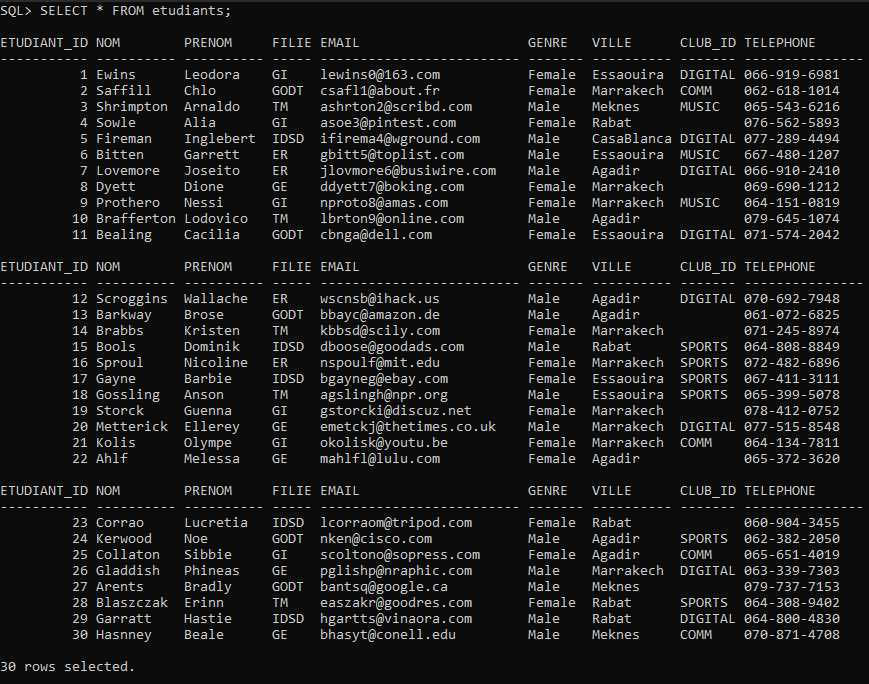


Figure 28 requête : sélection de tous les champs de la table etudiants

Cette requête ci-dessus permet d’afficher les différents emplois des employees.



Figure 29 requête : sélection des champs de la table employees en utilisant la clause DISTINCT

Cette requête ci-dessus permet d’afficher une colonne nommée « etudiants » contient la chaîne suivante : « la filier de » concaténé avec le nom et prenom, qui est concaténé aussi avec la chaîne « est » concaténé avec le filier\_id.

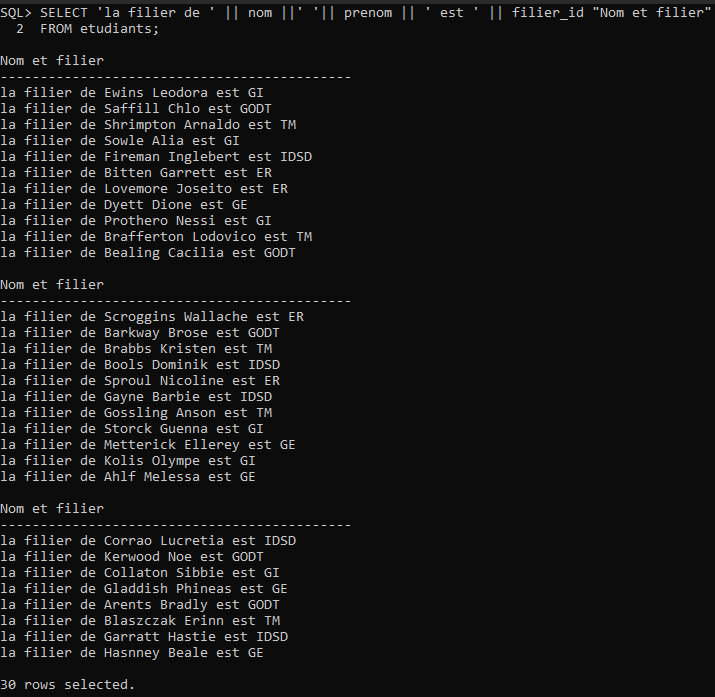


Figure 30 requête : sélection des champs de la table etudiants en utilisant la concaténation et les alias

### Table emplois :

Cette requête ci-dessus permet de sélectionner toutes les colonnes de la table (produit).

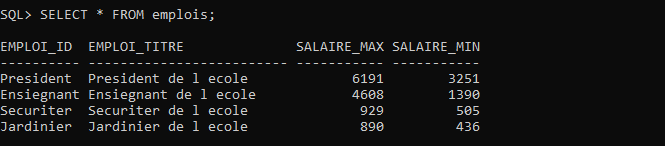


Figure 31 requête : sélection de tous les champs de la table emplois

Cette requête ci-dessus permet de sélectionner l’emploi, et le salaire moyen, sous le nom « Salaire moyen », de tous les produits.

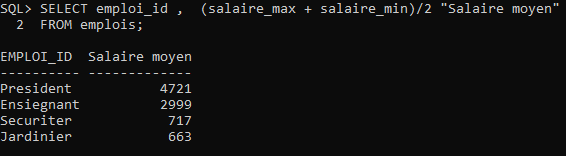


Figure 32 requête : sélection de quelque champ en utilisant les opérateurs arithmétiques et les alias

### Table filiers :

Cette requête ci-dessus permet de sélectionner toutes les colonnes de la table (filiers).

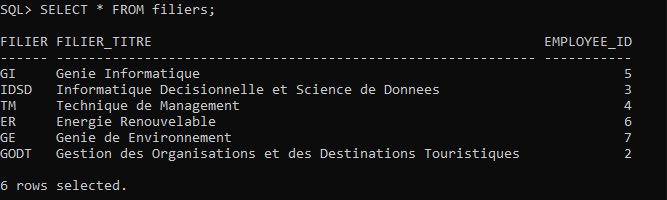


Figure 33 requête : sélection de tous les champs de la table filiers

### Table clubs :

Cette requête ci-dessus permet de sélectionner toutes les colonnes de la table (clubs).

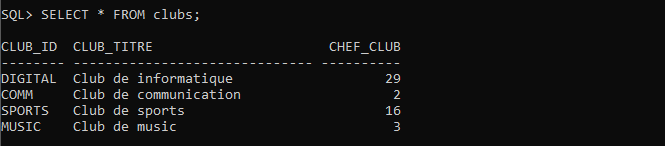


Figure 34 requête : sélection de tous les champs de la table clubs

### Limiter et trier des données :

Lors de l'extraction de données de la base, il est parfois nécessaire de restreindre le nombre de lignes renvoyées ou de préciser l'ordre d'affichage de ces lignes.

On peut limiter le nombre de lignes renvoyées par l'interrogation à l'aide de la clause WHERE. Cette clause comporte une condition à satisfaire

### Table employees :

Cette requête ci-dessus permet d’extraire le numéro, le nom, le prix, et la catégorie de tous les produits qui ont la quantité égale à 100.

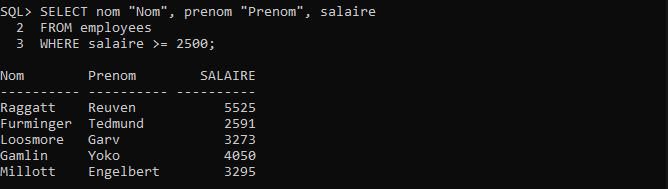


Figure 35 requête : sélection des données limités en utilisant les conditions de comparaison

Cette requête ci-dessus permet d’afficher le nom, le prenom et l’emploi de toutes les employées qui en enseignant.

**LOWER** : permet de Convertir tous les caractères d'une chaîne en minuscules.

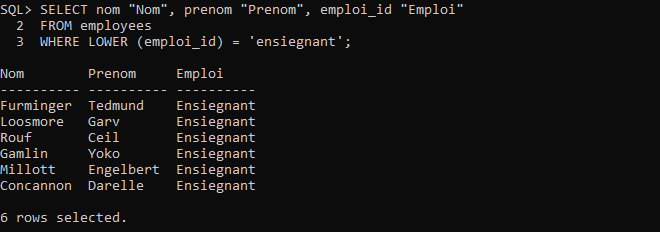


Figure 36 requête : sélection des données limitées de la table employees en utilisant les condition de comparaison et les fonction de manipulation des majiscules/miniscules.

Cette requête ci-dessus affiche toutes les employées dont la filier est « GI » ou « IDSD » ou « ER » en utilisant la condition d’appartenance **IN**.

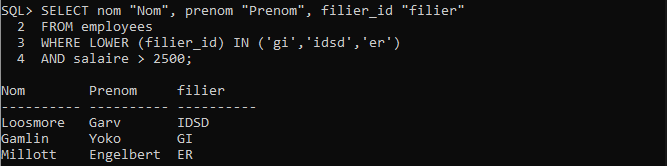


Figure 37 requête : sélection des données limitées de la table employees en utilisant les conditions de comparaison

### Table etudiants :

Cette requête ci-dessus affiche toutes les etudiants qui intégrer sur un club.

Une valeur NULL est une valeur non disponible non affecté ou inapplicable.

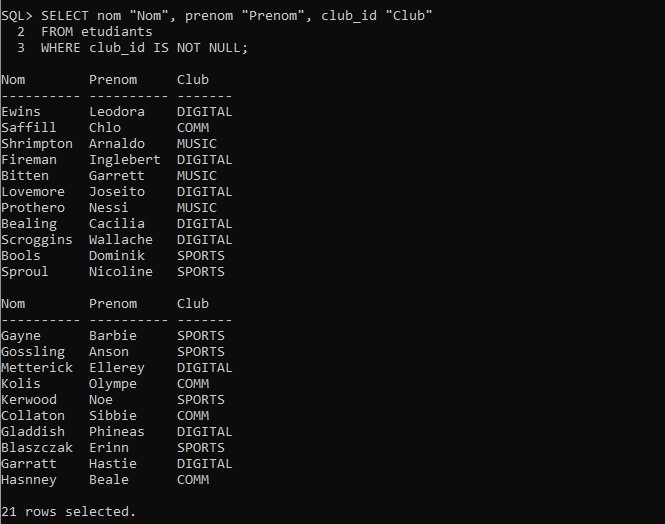


Figure 38 requête : sélection des données de la table etudiants en utilisant NOT NULL

Cette requête ci-dessus permet d’afficher le nom, le prenom et le club de chaque etudiant qui n’a pas sur les clubs de « MUSIC » ou « SPORTS », en utilisant la condition NOT IN.

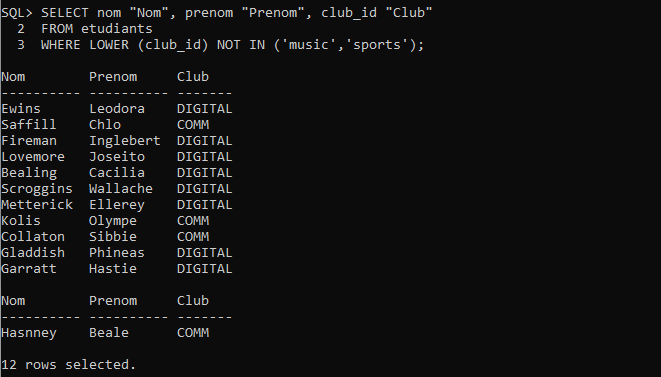


Figure 39 requête : sélection des données limitées et triées en utilisant les conditions de comparaison

### Table emplois :

Cette requête ci-dessus permet d’afficher l’emploi, le main salaire et le max salaire de chaque emploi, trier par la date le max salaire, et tout ça où le min salaire supérieure à 500.

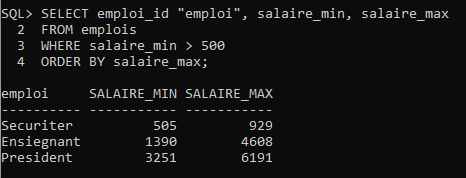


Figure 40 requête : sélection des données limitées et triées dans la table commande

### Table filiers :

Cette requête ci-dessus permet d’afficher la filier et l’enseignant de la filier, trier par date numéro de l’enseignant, dont le numéro de l’enseignant est compris entre 2 et 6. On a utilisé la condition **BETWEEN … AND ….**

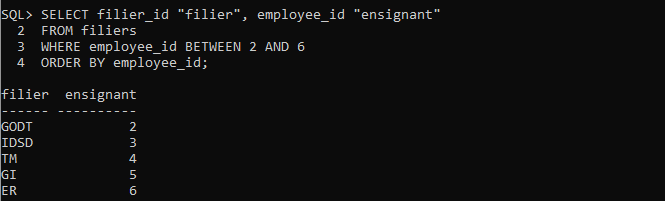


Figure 41 requête : sélection des données triées de la table filiers en utilisant les conditions de comparaison

Cette requête ci-dessus permet d’afficher la description des filiers, et le nom de filier qui a le nom « TM » ou bien qui a le titre « Technique de Management ». On a utilisé l’opérateur **OR** qui renvoie true si l’une des conditions est vraie.

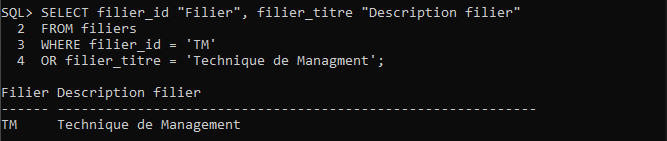


Figure 42 requête : sélection des données de la table filiers en utilisant les conditions logiques

### Table clubs :

Cette requête ci-dessus affiche le nom de club et la description de club de tous les clubs qui ont le club\_id se termine par « L » ou qui ont le titre se commence par « c » et le club\_id contient « %i% ». On a utilisé l’opérateur suivant :

**AND** qui renvoie true si les deux conditions sont vraies.

**LOWER** qui converti tous les caractères d’une chaîne en minuscules.

**UPPER** qui converti tous les caractères d’une chaîne en majuscules.

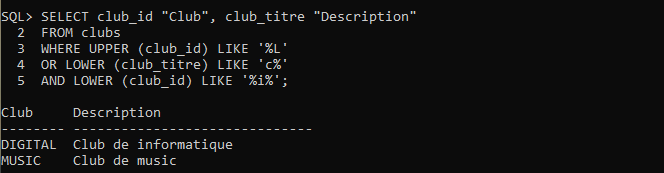


Figure 43 requête : sélection des données limitées de la table clubs en utilisant les conditions logiques

### Afficher les données issues de plusieurs tables :

Les jointures en SQL permettent d’associer plusieurs tables dans une même requête. Cela permet d’exploiter la puissance des bases de données relationnelles pour obtenir des résultats qui combinent les données de plusieurs tables de manière efficace.

Cette requête ci-dessus affiche le numéro, le nom et le prenom chaque employee avec le titre de sa filier.

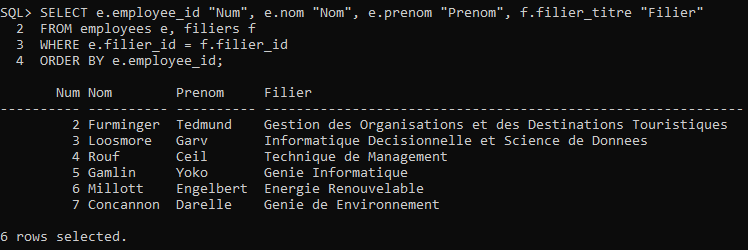


Figure 44 requête : sélection des données de la table employees et filiers en utilisant l'équijointure

Cette requête ci-dessus permet d’afficher tous les filiers avec leurs employees. En utilisant la clause **ON**

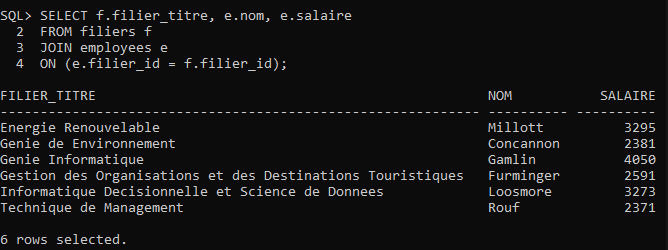


Figure 45 requête : Extraire des enregistrements de la table commande et produit à l'aide

Cette requête ci-dessus permet de faire la jointure externe entre les deux tables « etudiants » et « clubs » par la colonne « club\_id », et d’extraire toutes les lignes de la table « produit » même s'il n'existe aucune correspondance dans la table « commande ».

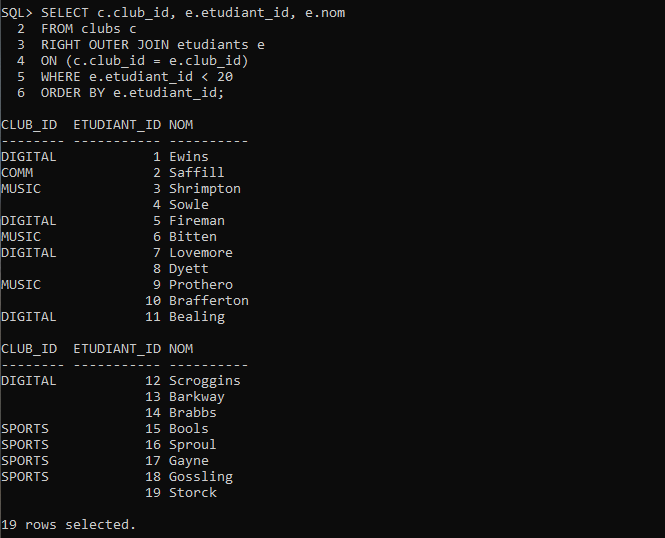


Figure 46 requête : jointure RIGHT OUTER JOIN

### Utilisation des fonctions monoligne :

Cette requête ci-dessus affiche le nombre des mois d’ancienneté de chaque employee. En utilisant la fonction de date : **MONTHS\_BETXEEN**.

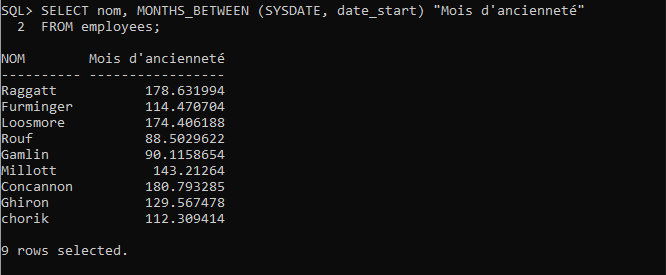


Figure 47 requête : sélection des données de la table employees en utilisant les foncions de date

Cette requête ci-dessus permet d’afficher la filière et le numéro de l’employee et le salaire de chaque employee.

Le salaire sera affiché sous une forme en utilisant **TO\_CHAR**.

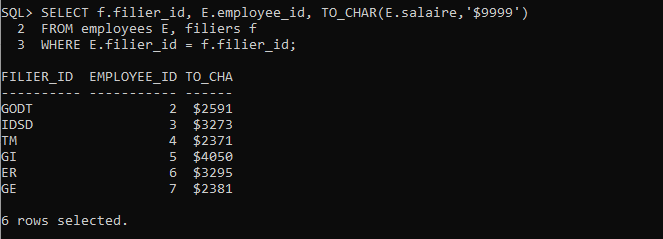


Figure 48 requête : sélection des données de la table employees et filiers en utilisant l'équijointure et la conversion explicite de type de données

Cette requête ci-dessus affiche le nom d’etudiant, le club et la ID\_club pour chaque club et chaque etudiant.

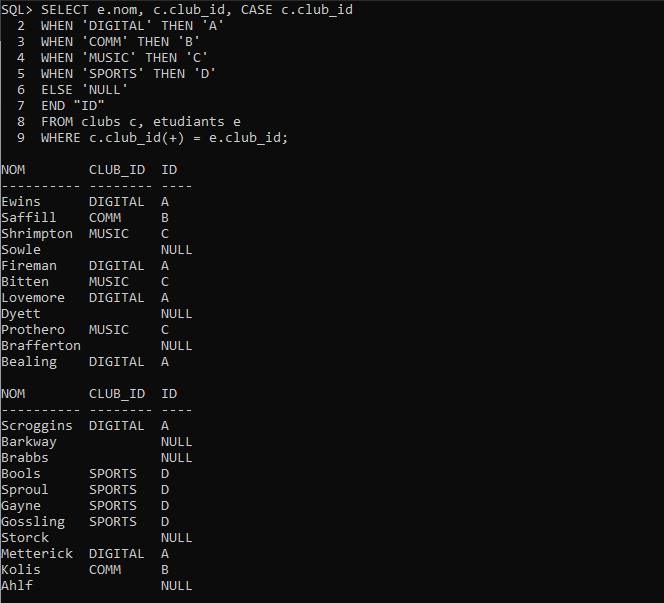


Figure 49 requête : utilisation de l'expression conditionnelle CASE

Cette requête ci-dessus affiche le nom et l’emploi de chaque employée et en plus d’une colonne nommée « Ordre » qui poser l’ordre de chaque emploi en utilisant l’expression **DECODE**.

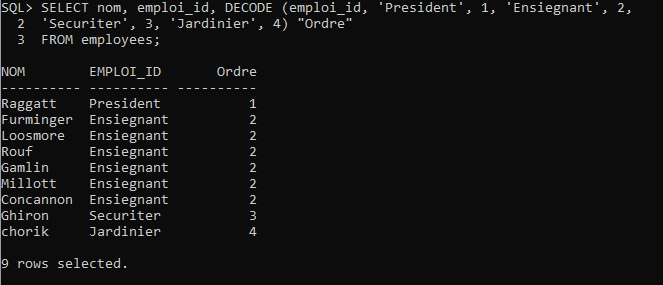


Figure 50 requête : utilisation de l'expression conditionnelle DECODE

### Utilisation des fonctions de group et des sous interrogations :

Cette requête affiche le nombre des filiers.

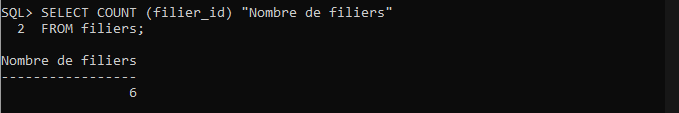


Figure 51 requête : utilisation de la fonction de group COUNT

Cette requête renvoie le nombre des étudiants.

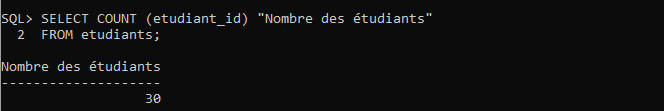


Figure 52 requête : utilisation de fonction de group COUNT

Cette requête ci-dessus affiche le nombre des étudiants qui sont habité à Essaouira et Marrakech.

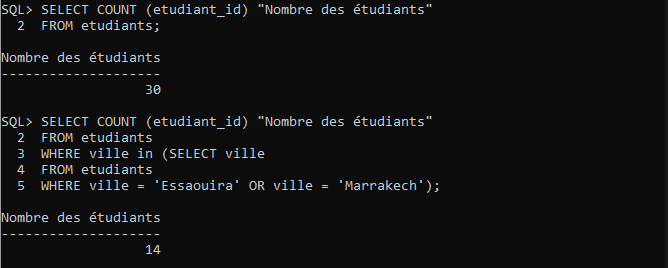


Figure 53 requête : utilisation de fonction de group COUNT et limiter les données en utilisant les sous interrogations

Cette requête ci-dessus affiche le nombre des étudiants pour chaque club.

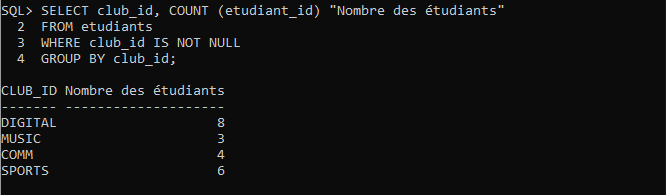


Figure 54 requête : utilisation de fonction de group et conversion explicite de type de données

### Utilisation des vues :

Une vue : est une table virtuelle basée sur l'ensemble de résultats d'une instruction SQL.

### VIEW enseignants :

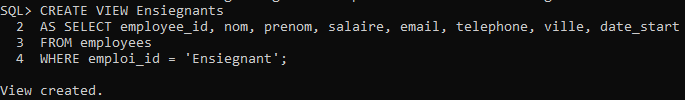


Figure 55 Création de la view enseignants

### VIEW etudiants\_GI :

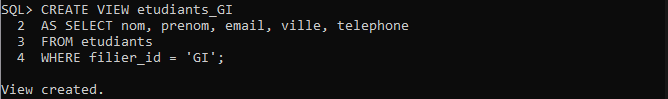


Figure 56 Création de la view etudiants\_GI

### VIEW pauvre\_employees :

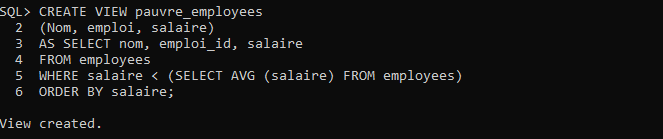


Figure 57 Création de la view complexe pauvre\_employees

### Inclure des contraintes :

* WITH CHECK OPTION
* WITH READ ONLY

### V*IEW* Enseignants :

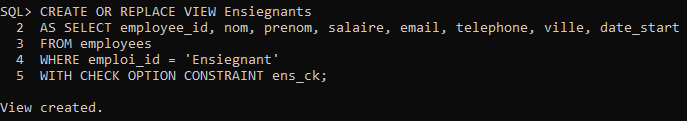


Figure 58 ajouter contraintes à la view Enseignants à l'aide de la clause CREATE OR REPLACE VIEW

### V*IEW* etudiants\_GI :

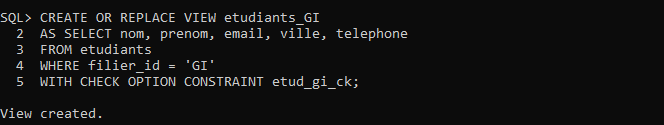


Figure 59 ajouter contraintes à la view etudiants\_GI à l'aide de la clause CREATE OR REPLACE VIEW

### V*IEW* pauvre\_employees :

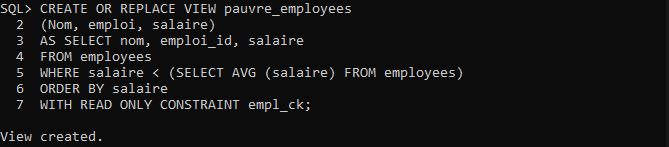


Figure 60 ajouter contraintes à la view pauvre\_employees à l'aide de la clause CREATE OR REPLACE VIEW

### Vérification de la création des views :

### V*IEW* Enseignants :

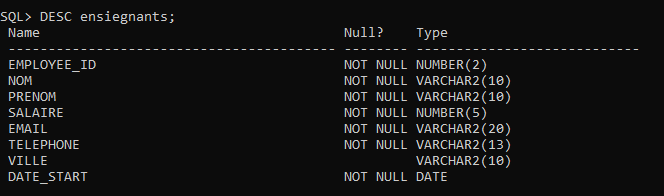


Figure 61 vérification de la création de la view Enseignants à l'aide de la clause DESCRIBE

### V*IEW* etudiants\_GI :

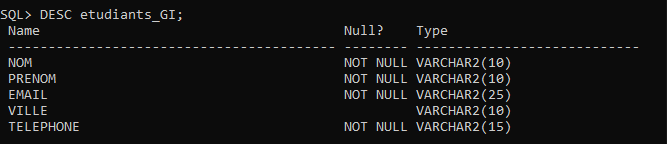


Figure 62 vérification de la création de la view etudiants\_GI à l'aide de la clause DESCRIBE

### V*IEW* pauvre\_employees :

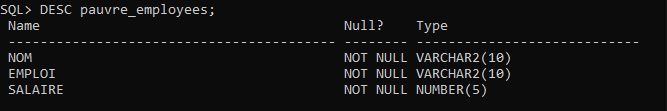


Figure 63 vérification de la création de la view pauvre\_employees à l'aide de la clause DESCRIBE

### Vérification de l’affichage des views :

### V*IEW* Enseignants :

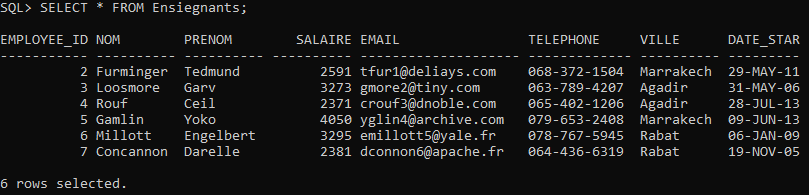


Figure 64 requête : sélection de tous les champs de la view Enseignants

### V*IEW* etudiants\_GI :



Figure 65 requête : sélection de tous les champs de la view Enseignants

### V*IEW* pauvre\_employees :

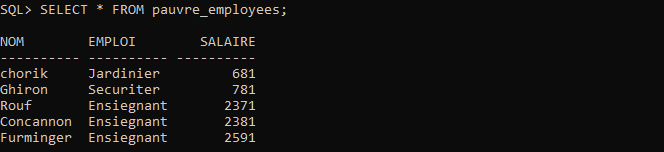


Figure 66 requête : sélection de tous les champs de la view Enseignants

# Conclusion

Finalement, ce projet nous a permet d’appliquer tous ce qu’on a pris durant les cours de base données afin de réaliser un projet finalisé, et de développer des connaissances techniques et pratique du langage SQL Oracle.