**SQL Déclaratif**

Du standard à la pratique

**Manuel d’exercices**

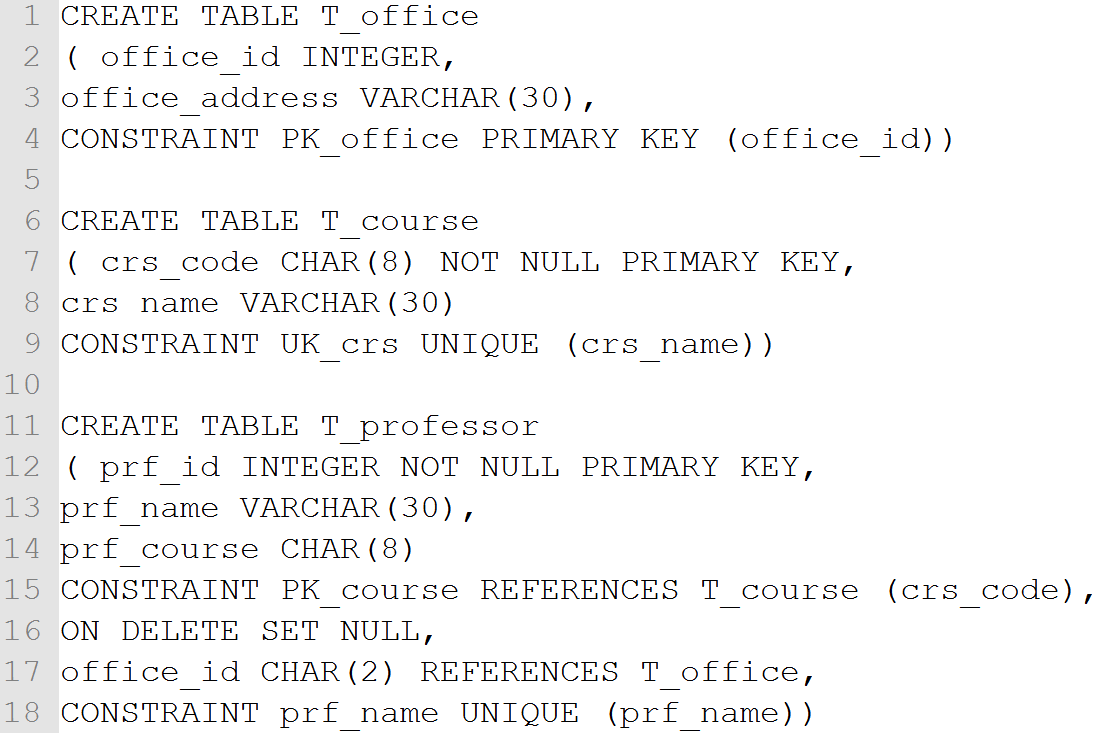
*MODULE I :*

DDL (Data Definition Language - Langage de définition de données)

**Exercice 1.1 – La syntaxe des ordres suivants est-elle correcte ? Si non, pourquoi ?**

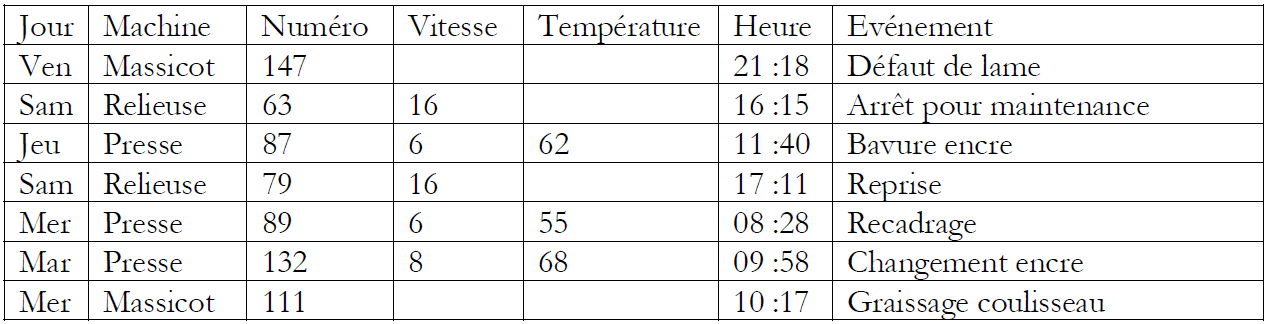
**Attention, les tables sont peut-être liées… !**

**N’hésitez pas à tester les requêtes directement !**



**Exercice 1.2 – A partir des données présentées dans le tableau suivant, proposer le code de la table T\_MAINTENANCE\_MTN.**

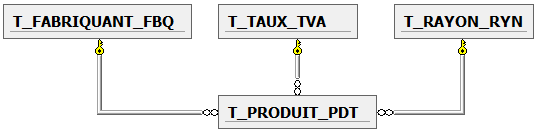
**Cette table devra contenir les 4 contraintes suivantes : contrainte de clé primaire, contrainte d’unicité, contrainte check et contrainte NOT NULL. Ces contraintes porteront sur 4 colonnes ou combinaisons de colonnes distinctes.**



**Exercice 1.3 – Créer une table pour y stocker les produits à vendre, avec les rubriques suivantes : identifiant, référence magasin, référence fabricant, code EAN13, prix de vente.**

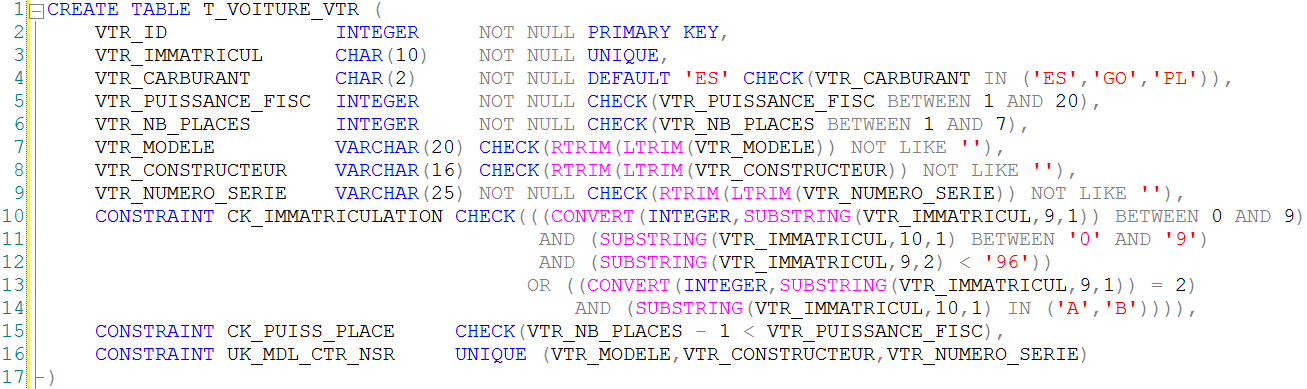
**Cette table fera en outre référence aux tables T\_TAUX\_TVA, T\_RAYON\_RYN, T\_FABRICANT\_FBQ.**

**Mettez en place toutes les contraintes nécessaires. La table produit contiendra au minimum les colonnes proposées, mais peut en contenir d’autres au besoin.**

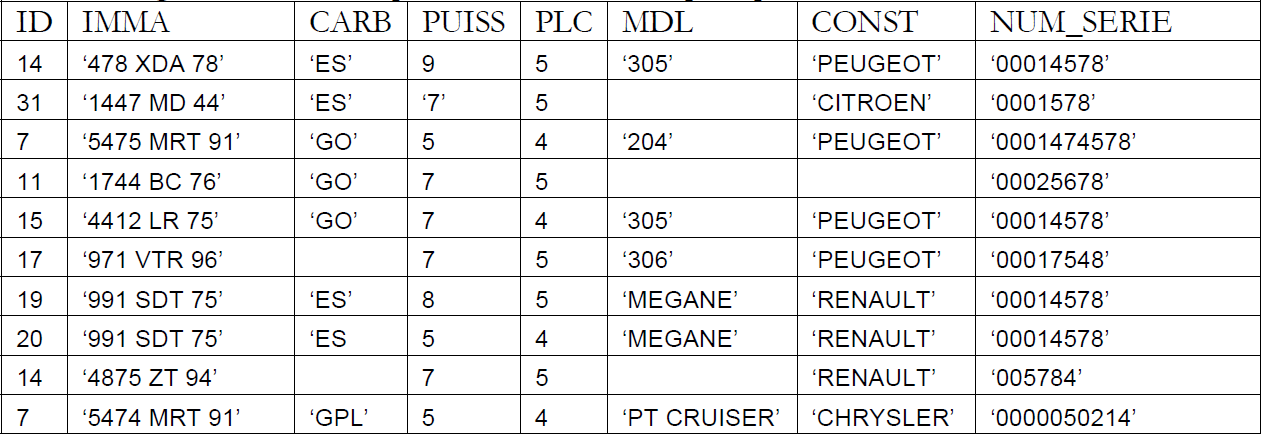


**Exercice 1.4 – Soit le code de création de table repris ci-après et pour lequel les annotations suivantes concernant les fonctions utilisées pourront être utiles (sous Oracle, demander le script et des explications au formateur) :**

* **« RTRIM(…) » et « LTRIM(…) » enlèves les espaces blancs respectivement à droite et à gauche de l’élément entre parenthèses**
* **« SUBSTRING(…,x,y) » renvoi la chaine de caractère commençant à « x » et se terminant « y » caractères après « x », à partir de la chaine de caractères donnée entre parenthèses**
* **« CONVERT(TYPE,…) » renvoi la valeur fournie dans le « TYPE » demandé**



**Parmi les lignes suivantes, lesquelles seront refusées et pourquoi ?**



**Exercice 1.5 – Deux scripts vous sont fournis : « DBSlide\_LoadDB.sql » et « DBSlide\_LoadData.sql ».**

**Créer une base de données que l’on appellera « DBSlide ». Tenter d’exécuter les scripts fournis… Cela ne devrait pas fonctionner. A vous de les corriger !**

**Exercice 1.6 – Une fois les scripts de l’exercice précédent corrigés, les tables créées et remplies, réaliser les modifications suivantes :**

* **Autoriser la table « SECTION » à accepter des valeurs NULL pour la colonne « delegate\_id »**
* **Ajouter à la table « SECTION » une clé étrangère faisant pointer la colonne « delegate\_id » vers la colonne « student\_id » de la table « STUDENT »**
* **Supprimer la colonne « course\_id » de la table « STUDENT »**
* **Faire en sorte que les données de la colonne « student\_id » de la table « STUDENT » soient auto-incrémentées**
* **En ne supprimant aucune donnée, modifier le type de la colonne « section\_id » de la table « section » afin qu’il soit en CHAR(4). Cela impliquera peut-être d’autres modifications…**

**Exercice 1.7 – Améliorer le script « DBSlide\_LoadDB.sql » afin qu’il commence par supprimer les tables, pour ensuite les recréer sans leurs clés étrangères. Une fois chaque table créée, leur rajouter les clés étrangères**

**Exercice 1.8 – Afin de partir sur des bases communes pour les exercices à venir, exécuter les scripts « DBSlide\_LoadDB\_OK.sql » et « DBSlide\_loadData\_OK.sql » (ou « DBSlide\_LoadDB\_Oracle.sql » et « DBSlide\_LoadData\_Oracle.sql », sous Oracle)**

**Exercice 1.9 – Ceci clôture la partie DDL du cours. Avant de passer à la suite de la matière, nous vous invitons à prendre un peu de temps afin d’évaluer personnellement votre niveau de compréhension de la matière en vous référant aux derniers slides du module (slides d’auto-évaluation)**

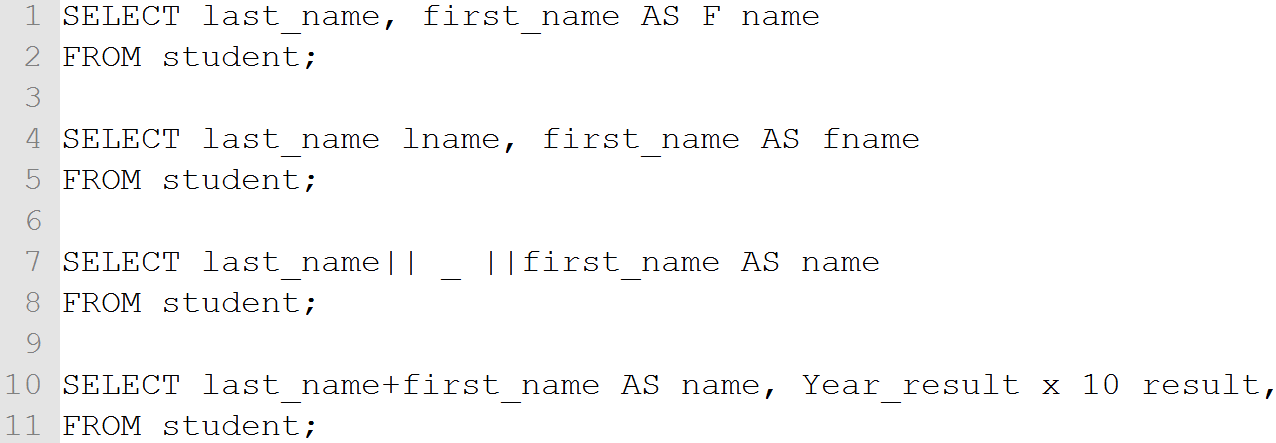
*MODULE 2 :*

DRL (Data Retrieval Language - Langage d’extraction de données)

*Partie I : SELECT … FROM …*

**Exercice 2.1.1 – Les requêtes suivantes fonctionnent-elles sous SQL-Server ? Si non, comment les corriger ?**

**N’hésitez pas à tester vos requêtes directement sous Management Studio !**



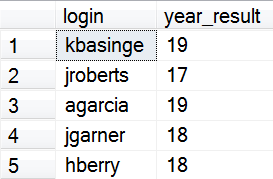
**Exercice 2.1.2 – Ecrire une requête pour présenter, pour chaque étudiant, le nom de l’étudiant, la date de naissance, le login et le résultat pour l’année de l’ensemble des étudiants.**

**Exercice 2.1.3 – Ecrire une requête pour présenter, pour chaque étudiant, son nom complet (nom et prénom séparés par un espace), son id et sa date de naissance.**

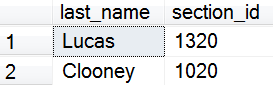
**Exercice 2.1.4 – Ecrire une requête pour présenter, pour chaque étudiant, dans une seule colonne (nommée « Info Étudiant ») l’ensemble des données relatives à un étudiant séparées par le symbole « | ». Sous SQL Server, il est nécessaire d’avoir recours à la fonction de conversion CONVERT(type, champs).**

*Partie II : SELECT … FROM … WHERE … ORDER BY*

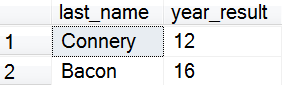
**Exercice 2.2.1 – Ecrire une requête pour présenter le login et le résultat de tous les étudiants ayant obtenu un résultat annuel supérieur à 16**



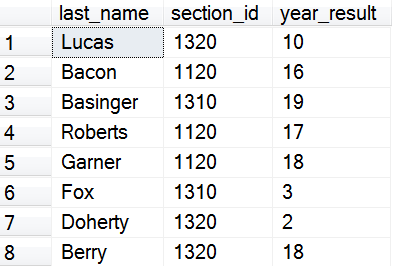
**Exercice 2.2.2 – Ecrire une requête pour présenter le nom et l’id de section des étudiants dont le prénom est Georges**



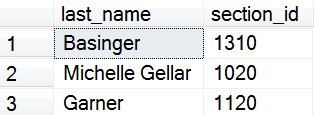
**Exercice 2.2.3 – Ecrire une requête pour présenter le nom et le résultat annuel de tous les étudiants ayant obtenu un résultat annuel compris entre 12 et 16**



**Exercice 2.2.4 – Ecrire une requête pour présenter le nom, l’id de section et le résultat annuel de tous les étudiants qui ne font pas partie des sections 1010, 1020 et 1110**



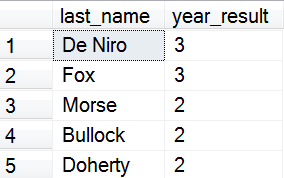
**Exercice 2.2.5 – Ecrire une requête pour présenter le nom et l’id de section de tous les étudiants qui ont un nom de famille qui termine par « r »**



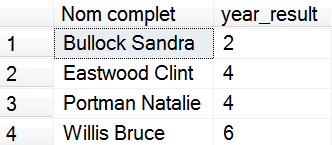
**Exercice 2.2.6 – Ecrire une requête pour présenter le nom et le résultat annuel de tous les étudiants qui ont un nom de famille pour lequel la troisième lettre est un « n » et qui ont obtenu un résultat annuel supérieur à 10**



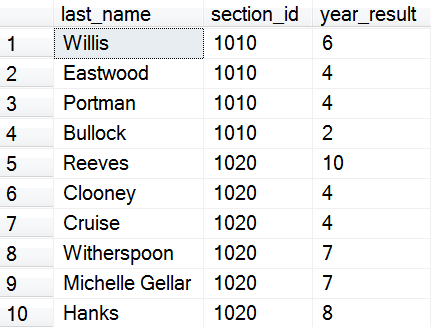
**Exercice 2.2.7 – Ecrire une requête pour présenter le nom et le résultat annuel classé par résultats annuels décroissants de tous les étudiants qui ont obtenu un résultat annuel inférieur ou égal à 3**



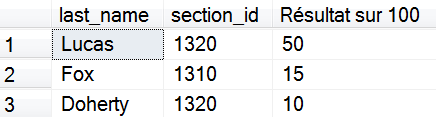
**Exercice 2.2.8 – Ecrire une requête pour présenter le nom complet (nom et prénom séparés par un espace) et le résultat annuel classé par nom croissant sur le nom de tous les étudiants appartenant à la section 1010**



**Exercice 2.2.9 – Ecrire une requête pour présenter le nom, l’id de section et le résultat annuel classé par ordre croissant sur la section de tous les étudiants appartenant aux sections 1010 et 1020 ayant un résultat annuel qui n’est pas compris entre 12 et 18**



**Exercice 2.2.10 – Ecrire une requête pour présenter le nom, l’id de section et le résultat annuel sur 100 (nommer la colonne « Résultat sur 100 ») classé par ordre décroissant du résultat de tous les étudiants appartenant aux sections commençant par 13 et ayant un résultat annuel sur 100 inférieur ou égal à 60**



**Exercice 2.2.11 – Ceci clôture la première partie DRL du cours. Avant de passer à la suite de la matière, nous vous invitons à prendre un peu de temps afin d’évaluer personnellement votre niveau de compréhension de la matière en vous référant aux derniers slides du module (slides d’auto-évaluation)**

*Partie III : Les Fonctions*

**Exercice 2.3.1 – Pourquoi lorsque l’on utilise la fonction « MAX » ou « MIN » les valeurs « NULL » sont-elles ignorées ?**

**Exercice 2.3.2 – Pourquoi le type des données n’a-t-il pas d’importance lorsque l’on utilise la fonction « COUNT » ?**

**Exercice 2.3.3 – La fonction « AVG » renvoie la moyenne de toutes les lignes résultantes d’une requête SELECT sur une colonne incluant toutes les valeurs « NULL ». (Vrai/Faux ?)**

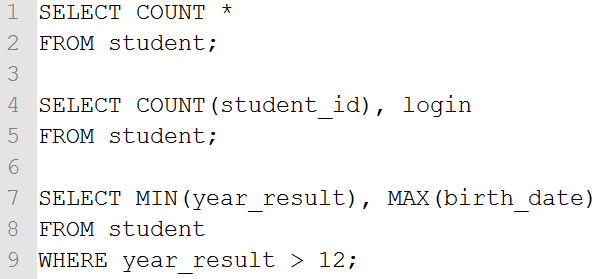
**Exercice 2.3.4 – La fonction « SUM » est utilisée pour ajouter des totaux aux colonnes.**

**(Vrai/Faux ?)**

**Exercice 2.3.5 – La fonction « COUNT(\*) » compte toutes les lignes d’une table. (Vrai/Faux ?)**

**v**

**Exercice 2.3.6 – Les requêtes suivantes sont-elles valides ?**



**Exercice 2.3.7 – Donner le résultat annuel moyen pour l’ensemble des étudiants**

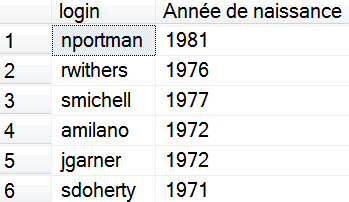
**Exercice 2.3.8 – Donner le plus haut résultat annuel obtenu par un étudiant**

**Exercice 2.3.9 – Donner la somme des résultats annuels**

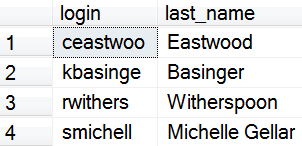
**Exercice 2.3.10 – Donner le résultat annuel le plus faible**

**Exercice 2.3.11 – Donner le nombre de lignes qui composent la table « STUDENT »**

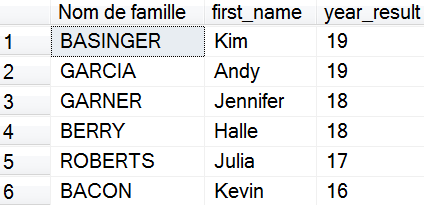
**Exercice 2.3.12 – Donner la liste des étudiants (login et année de naissance) nés après 1970**



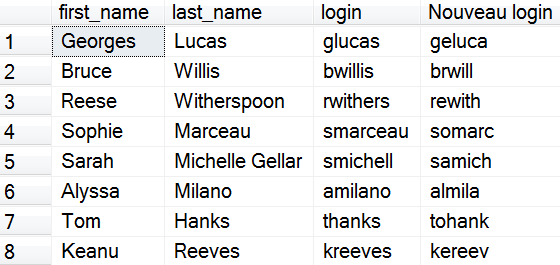
**Exercice 2.3.13 – Donner le login et le nom de tous les étudiants qui ont un nom composé d’au moins 8 lettres**



**Exercice 2.3.14 – Donner la liste des étudiants ayant obtenu un résultat annuel supérieur ou égal à 16. La liste présente le nom de l’étudiant en majuscules (nommer la colonne « Nom de Famille ») et le prénom de l’étudiant dans l’ordre décroissant des résultats annuels obtenus**

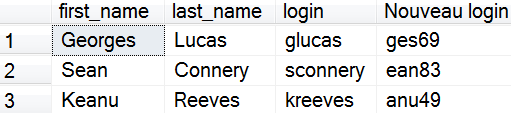


**Exercice 2.3.15 – Donner un nouveau login à chacun des étudiants ayant obtenu un résultat annuel compris entre 6 et 10. Le login se compose des deux premières lettres du prénom de l’étudiant suivi par les quatre premières lettres de son nom le tout en minuscule. Le résultat reprend pour chaque étudiant, son nom, son prénom l’ancien et le nouveau login (colonne « Nouveau login »)**

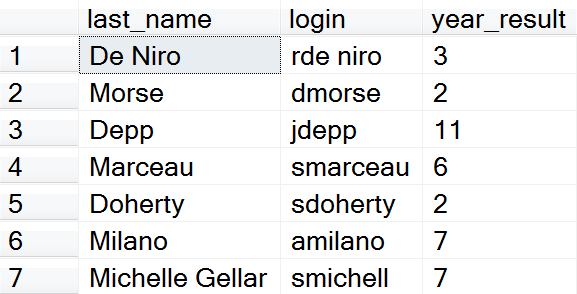


**Exercice 2.3.16 – Donner un nouveau login à chacun des étudiants ayant obtenu un résultat1 annuel égal à 10, 12 ou 14. Le login se compose des trois dernières lettres de son prénom suivi du chiffre obtenu en faisant la différence entre l’année en cours et l’année de leur naissance. Le résultat reprend pour chaque étudiant, son nom, son prénom l’ancien et le nouveau login (colonne**

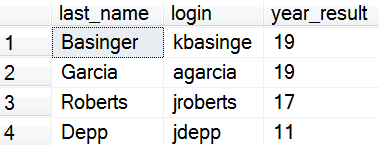
**« Nouveau login »)**



**Exercice 2.3.17 – Donner la liste des étudiants (nom, login, résultat annuel) qui ont un nom commençant par « D », « M » ou « S ». La liste doit présenter les données dans l’ordre croissant des dates de naissance des étudiants**



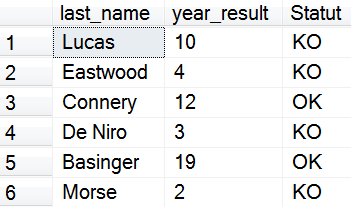
**Exercice 2.3.18 – Donner la liste des étudiants (nom, login, résultat annuel) qui ont obtenu un résultat impair supérieur à 10. La liste doit être triée du plus grand résultat au plus petit**



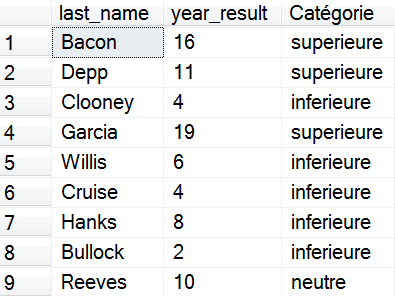
**Exercice 2.3.19 – Donner le nombre d’étudiants qui ont au moins 7 lettres dans leur nom de famille**



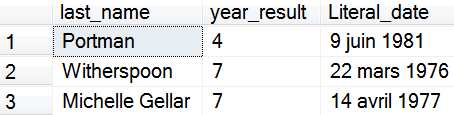
**Exercice 2.3.20 – Pour chaque étudiant né avant 1955, donner le nom, le résultat annuel et le statut. Le statut prend la valeur « OK » si l’étudiant à obtenu au moins 12 comme résultat annuel et « KO » dans le cas contraire**



**Exercice 2.3.21 – Donner pour chaque étudiant né entre 1955 et 1965 le nom, le résultat annuel et la catégorie à laquelle il appartient. La catégorie est fonction du résultat annuel obtenu : un résultat inférieur à 10 appartient à la catégorie « inférieure », un résultat égal à 10 appartient à la catégorie « neutre », un résultat autre appartient à la catégorie « supérieure »**



**Exercice 2.3.22 – Donner pour chaque étudiant né entre 1975 et 1985, son nom, son résultat annuel et sa date de naissance sous la forme: jours en chiffre, mois en lettre et années en quatre chiffres (ex : 11 juin 2005)**



**Exercice 2.3.23 – Donner pour chaque étudiant né en dehors des mois d’hiver et ayant obtenu un résultat inférieur à 7, son nom, le mois de sa naissance (en chiffre) son résultat annuel et son résultat annuel corrigé (« Nouveau résultat ») tel que si le résultat annuel est égal à 4, le valeur proposée est « NULL »**



**Exercice 2.3.24 – Ceci clôture la deuxième partie DRL du cours. Avant de passer à la suite de la matière, nous vous invitons à prendre un peu de temps afin d’évaluer personnellement votre niveau de compréhension de la matière en vous référant aux derniers slides du module (slides d’auto-évaluation)**

*Partie IV : GROUP BY … HAVING*

**Exercice 2.4.1 – L’utilisation de « GROUP BY » peut être considérée comme une forme de boucle dans une requête SQL ? (Vrai/Faux)**

**Exercice 2.4.2 – La répartition en groupe se fait avant de prendre en compte les restrictions imposées par un « WHERE » ? (Vrai/Faux)**

**faux**

**Exercice 2.4.3 – Un « GROUP BY » doit impérativement porter sur une colonne non alliacée ?**

**vrai**

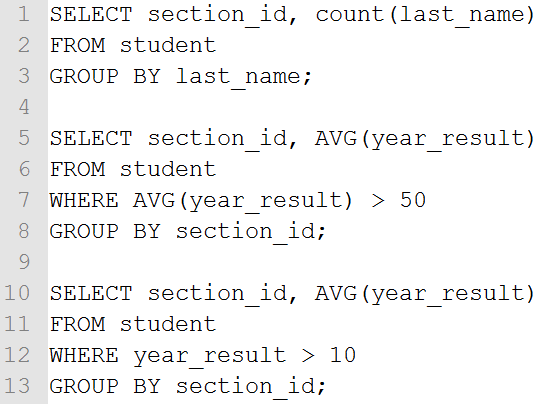
**Exercice 2.4.4 – L’utilisation d’un « GROUP BY » a pour effet de trier les résultats dans l’ordre croissant de la colonne incluse dans le « GROUP BY » ? (Vrai/Faux)**

**vrai**

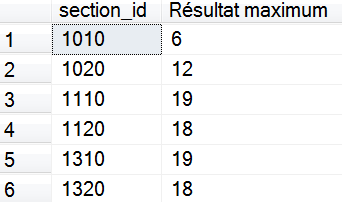
**Exercice 2.4.5 – La colonne sur laquelle porte le « GROUP BY » doit impérativement être présente dans la clause « SELECT » ? (Vrai/Faux)**

**faux**

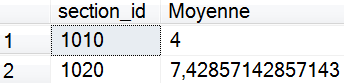
**Exercice 2.4.6 – Les requêtes suivantes sont-elles valides ?**



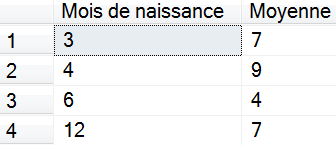
**Exercice 2.4.7 – Donner pour chaque section, le résultat maximum (dans une colonne appelée « Résultat maximum ») obtenu par les étudiants**



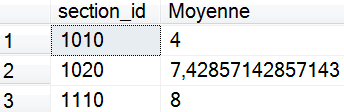
**Exercice 2.4.8 – Donner pour toutes les sections commençant par 10, le résultat annuel moyen PRÉCIS (dans une colonne appelée « Moyenne ») obtenu par les étudiants**



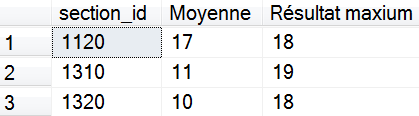
**Exercice 2.4.9 – Donner le résultat moyen (dans une colonne appelée « Moyenne ») et le mois en chiffre (dans une colonne appelée « Mois de naissance ») pour les étudiants nés le même mois entre 1970 et 1985**



**Exercice 2.4.10 – Donner pour toutes les sections qui comptent plus de 3 étudiants, la moyenne PRÉCISE des résultats annuels (dans une colonne appelée « Moyenne »)**



**Exercice 2.4.11 – Donner le résultat maximum obtenu par les étudiants appartenant aux sections dont le résultat moyen est supérieur à 8**



*Partie V : CUBE et ROLLUP*

**Exercice 2.5.1 – L’utilisation de « ROLLUP » crée des groupes de données en se déplaçant dans une seule direction, partant de la gauche vers la droite par rapport aux colonnes sélectionnées ? (Vrai/Faux)**

**faux**

**Exercice 2.5.2 – Le résultat produit par un « ROLLUP » présente les résultats du plus agrégé au moins agrégé ? (Vrai/Faux)**

**Exercice 2.5.3 – L’opérateur « CUBE » permet de produire moins de sous-totaux qu’avec l’opérateur « ROLLUP » ? (Vrai/Faux)**

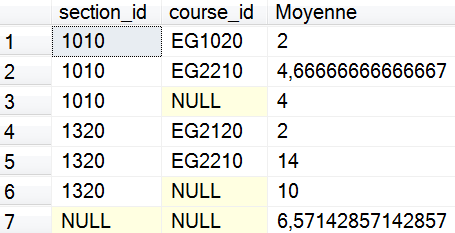
**faux**

**Exercice 2.5.4 – Avec l’opérateur « CUBE », le nombre de groupes dans le résultat est indépendant du nombre de colonnes sélectionnées dans le « GROUP BY » ? (Vrai/Faux)**

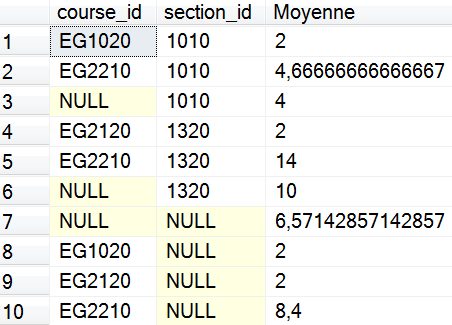
**Exercice 2.5.5 – L’opérateur « CUBE » ne peut pas être appliqué à la fonction d’agrégation « SUM » ? (Vrai/Faux)**

faux

**Exercice 2.5.6 – Donner la moyenne exacte des résultats obtenus par les étudiants par section et par cours, ainsi que la moyenne par section uniquement et enfin, la moyenne générale. La liste ainsi produite reprend l’id de section, de cours le résultat moyen (dans une colonne appelée « Moyenne »). Se baser uniquement sur les sections 1010 et 1320**



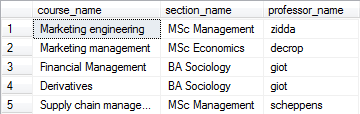
**Exercice 2.5.7 – Donner la moyenne exacte des résultats obtenus par les étudiants par cours et par section, ainsi que la moyenne par cours uniquement, puis par section uniquement et enfin, la moyenne générale. La liste ainsi produite reprend l’id de section, de cours le résultat moyen (dans une colonne appelée « Moyenne »). Se baser uniquement sur les sections 1010 et 1320**



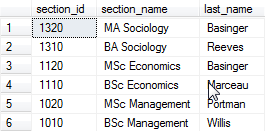
**Exercice 2.5.8 – Ceci clôture la troisième partie DRL du cours. Avant de passer à la suite de la matière, nous vous invitons à prendre un peu de temps afin d’évaluer personnellement votre niveau de compréhension de la matière en vous référant aux derniers slides du module (slides d’auto-évaluation)**

*Partie VI : Jointures*

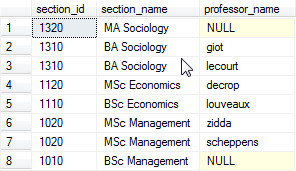
**Exercice 2.6.1 – Donner pour chaque cours le nom du professeur responsable ainsi que la section dont le professeur fait partie**



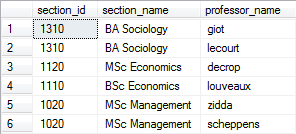
**Exercice 2.6.2 – Donner pour chaque section, l’id, le nom et le nom de son délégué. Classer les sections dans l’ordre inverse des id de section. Un délégué est un étudiant de la table « STUDENT »**



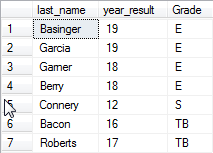
**Exercice 2.6.3 – Donner pour chaque section, le nom des professeurs qui en sont membre**



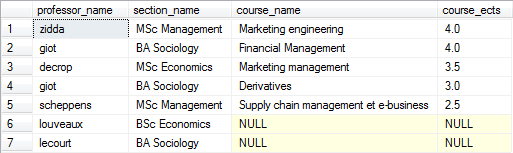
**Exercice 2.6.4 – Même objectif que la question 3 mais seuls les sections comportant au moins un professeur doivent être reprises**



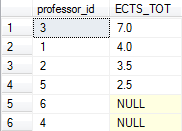
**Exercice 2.6.5 – Donner à chaque étudiant ayant obtenu un résultat annuel supérieur ou égal à 12 son grade en fonction de son résultat annuel et sur base de la table grade. La liste doit être classée dans l’ordre alphabétique des grades attribués**



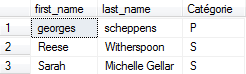
**Exercice 2.6.6 – Donner la liste des professeurs et la section à laquelle ils se rapportent ainsi que le(s) cour(s) (nom du cours et crédits) dont le professeur est responsable. La liste est triée par ordre décroissant des crédits attribués à un cours**



**Exercice 2.6.7 – Donner pour chaque professeur son id et le total des crédits ECTS (« ECTS\_TOT ») qui lui sont attribués. La liste proposée est triée par ordre décroissant de la somme des crédits alloués**



**Exercice 2.6.8 – Donner la liste (nom et prénom) de l’ensemble des professeurs et des étudiants dont le nom est composé de plus de 8 lettres. Ajouter une colonne pour préciser la catégorie (S pour « STUDENT », P pour « PROFESSOR ») à laquelle appartient l’individu**



**Exercice 2.6.9 – Donner l’id de chacune des sections qui n’ont pas de professeur attitré**



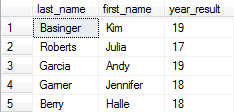
**Exercice 2.6.10 – Ceci clôture la quatrième partie DRL du cours. Avant de passer à la suite de la matière, nous vous invitons à prendre un peu de temps afin d’évaluer personnellement votre niveau de compréhension de la matière en vous référant aux derniers slides du module (slides d’auto-évaluation)**

*Partie VII : Requêtes imbriquées*

**Exercice 2.7.1 – Donner la liste des étudiants (nom et prénom) qui font partie de la même section que mademoiselle « Roberts ». La liste doit être classée par ordre alphabétique sur le nom et mademoiselle « Roberts » ne doit pas apparaitre dans la liste**



**Exercice 2.7.2 – Donner la liste des étudiants (nom, prénom et résultat) de l’ensemble des étudiants ayant obtenu un résultat annuel supérieur au double du résultat moyen pour l’ensemble des étudiants**



**Exercice 2.7.3 – Donner la liste de toutes les sections qui n’ont pas de professeur**



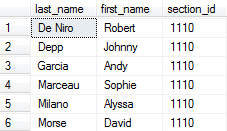
**Exercice 2.7.4 – Donner la liste des étudiants qui ont comme mois de naissance le mois correspondant à la date d’engagement du professeur « Giot ». Classer les étudiants par ordre de résultat annuel décroissant**



**Exercice 2.7.5 – Donner la liste des étudiants qui ont obtenu le grade « TB » pour leur résultat annuel**



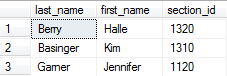
**Exercice 2.7.6 – Donner la liste des étudiants qui appartiennent à la section pour laquelle Mademoiselle « Marceau » est déléguée**



**Exercice 2.7.7 – Donner la liste des sections qui se composent de plus de quatre étudiants**



**Exercice 2.7.8 – Donner la liste des étudiants premiers de leur section en terme de résultat annuel et qui n’appartiennent pas aux sections dont le résultat moyen est inférieure à 10**



**Exercice 2.7.9 – Donner la section qui possède la moyenne la plus élevée. Le résultat présente le numéro de section ainsi que sa moyenne**



**Exercice 2.7.10 – Ceci clôture la cinquième partie DRL du cours. Avant de passer à la suite de la matière, nous vous invitons à prendre un peu de temps afin d’évaluer personnellement votre niveau de compréhension de la matière en vous référant aux derniers slides du module (slides d’auto-évaluation)**

*MODULE III :*

DML (Data Manipulation Language - Langage de manipulation de données)

**Exercice 3.1 – Inscrivez-vous comme étudiant dans la base de données DBSlide sans spécifier les noms de colonnes dans lesquelles on insère les données**

**Exercice 3.2 – Inscrivez votre voisin comme étudiant dans la base de données DBSlide. Votre voisin n’aura ni nom de famille, ni login, ni résultat annuel (valeurs NULL)**

**Exercice 3.3 – Créer une table « section\_archives » qui contiendra une copie des données contenues dans la table section**

**Exercice 3.4 – Insérer un nouvel étudiant dans la base de données. Cet étudiant sera inscrit dans la même section que Keanu Reeves, assistera au cours donné par le professeur Zidda (les lettres ‘EG’ suivies des 4 derniers caractères du cours en question) mais n’aura pas de login. Les valeurs de l’id de section et du cours devront être récupérées là où elles se trouvent dans la base de données, sans les renseigner directement**

**Exercice 3.5 – Insérer une nouvelle section dans la table section qui portera l’ID de section 1530, qui aura l’intitulé « Administration des SI » et qui aura le même délégué que la section dont l’ID et 1010 (vous ne connaissez pas la valeur de l’ID de ce délégué)**

**Exercice 3.6 – Mettre à jour vos propres données pour vous inscrire au cours EG2210**

**Exercice 3.7 – Mettre à jour les données de votre voisin pour qu’il ait un nom. Ensuite, refaire une mise à jour de la même ligne de données et attribuer à votre voisin un résultat de 18/20 et un login correspondant à la concaténation de la première lettre de son prénom et de la totalité de son nom, le tout en minuscules (sans connaître les valeurs réelles du nom et du prénom utilisés)**

**Exercice 3.8 – Mettre à jour les données de la table « student » pour que tous les étudiants de la section 1010 aient 15/20**

**Exercice 3.9 – Nommer Keanu Reeves délégué de la section 1530 (sans connaître la valeur réelle de l’ID de M. Reeves)**

**Exercice 3.10 – Donner à la section 1530 le même nom de section et le même délégué que la section 1320 (en allant rechercher ces valeurs via la requête, pas en les renseignant directement)**

**Exercice 3.11 – Nommer Alyssa Milano déléguée de sa section. On ne connait pas la valeur réelle de la section dans laquelle Mlle Milano est inscrite**

**Exercice 3.12 – Supprimer votre voisin de la base de données**

**Exercice 3.13 – Retirez-vous ainsi que Kim Basinger de la base de données. Comment se fait-il que le système accepte cette manipulation alors que Mlle. Basinger est déléguée de section ?**

**Exercice 3.14 – Supprimer tous les étudiants qui ont moins de 8/20**

**Exercice 3.15 – Supprimer tous les cours qui n’ont pas de professeur**

**Exercice 3.16 (bonus DDL-DML) – Sans supprimer les clés étrangères au préalable, supprimer les données de toutes les tables dans l’ordre suivant : sections => professeurs => étudiants => cours => grades. Il est possible qu’il faille d’abord modifier la structure des tables (ALTER TABLE) afin d’accepter des valeurs nulles à certains endroits… Une modification des données des tables sous-jacentes avant la suppression de certaines données sera sûrement nécessaire également**

**Exercice 3.17 – Ceci clôture la partie DDL du cours. Avant de passer à la suite de la matière, nous vous invitons à prendre un peu de temps afin d’évaluer personnellement votre niveau de compréhension de la matière en vous référant aux derniers slides du module (slides d’auto-évaluation)**