

## R504 - Mission 1

comment allons nous travailler :

- en binôme
- utiliser un outil de suivi de projet
- ce TD va s'étaler sur les séances suivantes :
  - jeudi 23 novembre
  - mercredi 6 décembre
  - jeudi 7 décembre
  - mercredi 20 décembre
- rendu : toutes les séances, un rapport d'activité sur votre séance qui doit contenir :
  - réponses aux questions
  - explication des choix
  - organisation du travail
  - explication des problématiques
  - screenshot trello (ou autres)
  - 2 pages recto/verso maximum - possibilité d'ajouter des annexes

Vous êtes embauché en tant qu'analyste administratif & financier dans l'entreprise Ma formation.

L'outil d'e-learning MaformationCom est mis en place et votre responsable vous attribue la mission de gestionnaire back office.

Le gestionnaire back office assure le traitement administratif et comptable des opérations à partir des ordres de transactions transmis par le front office, en intégrant les phases de confirmation, validation, règlement et livraison.

Le schéma entité association est donné en annexe : schéma.jpg

la base de données est donnée en annexe : MaFormationCom.accdb

# Mission 1

## Remplir la base de données, effectuer les tests.

définition d'un test unitaire : Le test unitaire est un processus de vérification d'une unique unité de logiciel, elle-même définie comme étant la plus petite partie non divisible. Dans notre cas, la plus petite partie sera une table.

### Actions 1 : créer des entrées dans la base de données et générer des jeux de tests.

Pour remplir la Base de données, plusieurs méthodes s'offrent à vous, je vous laisse les soin de décrire votre méthode dans votre rendu.

Vous pourrez utiliser le tableau test suivant :

Quoi	Qui	Réponse test	Problématique	Réponse à la problématique	test validé
table personne	david	ok			oui
table formation	david	défaut	pas d'entrée pour type formation	d'abord remplir la table Typeformation	oui

Table Type de formation : *initiale, continue, alternance*

Table Type personne : *administratif, prof, élève tuteur, prof-tuteur*

Table formation

Liste des bts proposés :

*BTS SUPPORT A L'ACTION MANAGÉRIALE - SAM*

*BTS MANAGEMENT COMMERCIAL OPÉRATIONNEL - MCO*

*BTS NÉGOCIATION DIGITALISATION RELATION CLIENT - NDRC*

Liste des bac +3

*Responsable Marketing et Communication*

*Chargé(e) de Développement des RH*

Poursuivez le remplissage de la BD avec :

- 9 professeurs
- 1 session par formation (soit 5 au total)
- 6 élèves par session (création de la personne + la liaison à une session)
- Lier les professeurs aux matières et à la session.

**Actions 2 : répondez aux questions ci dessous**

- 1) Qu'est ce qu'une clé primaire ?
- 2) A quoi sert une clé étrangère. Donner 3 exemples de clé étrangères de la base de données MaFormationCom
- 3) A quoi sert le format .csv ?
- 4) A quoi sert la table suiviSession ?
- 5) Expliquez le champ CpVille de la table personne
- 6) Expliquez les champs (attributs) de la table Evaluation.

**Actions 3 : modifier la BD**

- 1) Votre responsable vous indique qu'il veut bloquer l'accès aux profs qui ne font plus partie de l'équipe ou aux élèves qui démissionnent ou qui finissent leur étude.

Modifier la table Personne pour pouvoir répondre à ce besoin. Expliquez votre choix

- 2) Il serait intéressant que les sessions aient un délégué.

Modifier la table Session pour pouvoir répondre à ce besoin.

- 3) Les suivis de sessions ne doivent pas être visibles par tout le monde.

Modifier la table suiviSession pour pouvoir répondre à ce besoin.

**Actions 4 : ajout du module tutorat**

La plateforme souhaite proposer des tutorats. Pour cela un élève peut bénéficier d'un tuteur prof ou élève avec lequel il pourra avoir des séances d'accompagnement. Modifier la base de données afin de permettre ce service.

Vous devrez créer deux tables.

# Mission 2

## Interroger la base de données

Vous allez utiliser votre base de données.

L'objectif de ce TD est la création de requêtes.

Vous utiliserez le logiciel ACCESS. Pour créer une requête, cliquer sur le menu "Créer" puis "Création de requête".

ensuite sur le bandeau en haut à gauche vous aurez la possibilité de choisir SQL

### **Rendu :**

### **Synthèse le 7 décembre incluant mission 2 et mission 3**

Votre synthèse doit contenir pour chaque requête, la requête SQL et la screenshot de la réponse.

## Annexe : langage SQL

Le SQL, ou Structured Query Language, est un langage de programmation utilisé pour gérer et manipuler des bases de données relationnelles. Voici une synthèse de ses principaux aspects, accompagnée d'exemples:

Fondamentaux de SQL : SQL permet de lire, mettre à jour, insérer et supprimer des données dans une base de données. C'est un langage standardisé, mais des variantes existent selon les systèmes de gestion de base de données (SGBD) comme MySQL, PostgreSQL, SQL Server, etc.

Requêtes de Sélection (SELECT) :

- Exemple: `SELECT nom, prenom FROM utilisateurs;`
- Cette commande sélectionne les colonnes nom et prenom de la table utilisateurs.

Filtrage avec WHERE :

- Exemple: `SELECT * FROM utilisateurs WHERE age > 30;`
- Cette requête sélectionne tous les champs des utilisateurs dont l'âge est supérieur à 30 ans.

Jointures :

- Exemple: `SELECT utilisateurs.nom, commandes.date FROM utilisateurs JOIN commandes ON utilisateurs.id = commandes.id_utilisateur;`
- Cette requête effectue une jointure entre les tables utilisateurs et commandes pour sélectionner les noms d'utilisateurs et les dates de leurs commandes.

Insertion de Données (INSERT INTO) :

- Exemple: `INSERT INTO utilisateurs (nom, prenom, age) VALUES ('Dupont', 'Jean', 25);`
- Cette commande ajoute un nouvel utilisateur dans la table utilisateurs.

Mise à Jour de Données (UPDATE) :

- Exemple: `UPDATE utilisateurs SET age = 26 WHERE nom = 'Dupont' AND prenom = 'Jean';`
- Cette commande met à jour l'âge de l'utilisateur Jean Dupont.

Suppression de Données (DELETE) :

- Exemple: `DELETE FROM utilisateurs WHERE nom = 'Dupont';`

- Cette commande supprime de la table utilisateurs les entrées où le nom est 'Dupont'.

Gestion des Tables : SQL permet également de créer, modifier et supprimer des tables.

Fonctions Agrégées : Pour effectuer des calculs sur des ensembles de données, comme COUNT, SUM, MAX, MIN.

Transactions : SQL permet de gérer des transactions pour assurer l'intégrité des données, avec des commandes comme BEGIN, COMMIT, ROLLBACK.

SQL est un outil puissant pour la gestion des données, largement utilisé dans divers secteurs pour sa flexibilité et son efficacité. Il est essentiel pour les développeurs de bases de données, les analystes de données, et tout professionnel travaillant avec des données stockées de manière relationnelle.

### définition fonction d'agrégation

Une fonction d'agrégation en SQL est une fonction utilisée pour effectuer un calcul sur un ensemble de valeurs et retourner une valeur unique. Ces fonctions sont essentielles dans le traitement des données, en particulier lorsqu'il s'agit de synthétiser de grandes quantités d'informations. Voici les principales caractéristiques des fonctions d'agrégation :

- Opération sur des Groupes de Données : Les fonctions d'agrégation traitent des groupes de valeurs et produisent un résultat unique par groupe.
- Types de Fonctions d'Agrégation Courantes :
  - SUM() : Calcule la somme totale d'une colonne numérique.
  - AVG() : Calcule la moyenne des valeurs d'une colonne numérique.
  - COUNT() : Compte le nombre de lignes ou d'éléments distincts dans une colonne.
  - MAX() et MIN() : Trouvent respectivement la valeur maximale et minimale dans une colonne.
  - GROUP\_CONCAT() (dans certains SGBD) : Concatène les valeurs d'une colonne.
- Utilisation avec GROUP BY : Souvent utilisées en combinaison avec l'instruction GROUP BY pour regrouper les lignes ayant des valeurs communes dans une ou plusieurs colonnes, puis effectuer le calcul d'agrégation sur chaque groupe.
- Importance dans les Requêtes Analytiques : Les fonctions d'agrégation sont cruciales pour les requêtes analytiques et les

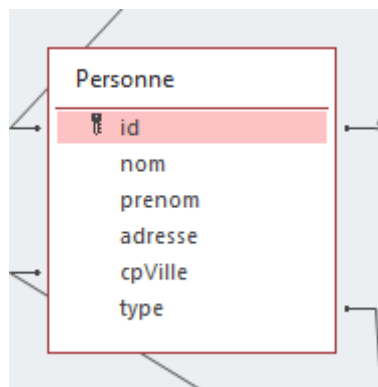
rapports, permettant de résumer des données, de faire des statistiques, et d'extraire des insights significatifs.

Les fonctions d'agrégation simplifient l'analyse des données en réduisant la complexité des ensembles de données et en présentant des résumés compréhensibles et utiles pour la prise de décision.

# Missions

## Action 1 : interroger une table.

Vous allez travailler sur la table Personne 👍



### Requêtes de sélection :

- 1) afficher la liste des personnes
- 2) afficher la liste des personnes comprenant id, nom, prenom
- 3) afficher la liste des professeurs
- 4) afficher la liste des professeurs par ordre alphabétique
- 5) affiche la liste des élèves
- 6) afficher tous les utilisateurs regroupés par type

### Requêtes de sélection avec fonction d'agrégat :

- 7) donner le nombre de personnes présents dans votre table Personne
- 8) donner le nombre de professeurs
- 9) donner le nombre d'élèves
- 10) donner le nombre d'utilisateurs regroupés par type

### Requêtes de sélection avec fonction d'agrégat :



- 11) écrire une requête qui insert dans la table personne l'élève Alex Dury
- 12) écrire une requête qui insert dans la table personne le professeur Alexandre Duri

**Action 2 : interroger plusieurs tables**

- 1) Donner la liste de sessions
- 2) Pour chaque session donner la liste des professeurs
- 3) Pour chaque session donner la liste des étudiants qui la compose
- 4) Pour chaque session donner la liste des professeurs avec leur.s matière.s
- 5) Donner la liste de tous les cours par professeurs

**Action 3 : requêtes à partir de votre module**

- 1) créer deux requêtes qui utilisent les tables de votre module. Vous utiliserez obligatoirement une jointure et une fonction d'agrégation.

**Action 4 : requêtes à partir d'un paramètre / requête paramétrée**

- 1) écrire une requête qui demande l'Id (clé primaire d'une évaluation) et qui renvoie la moyenne des notes.