# Travail pratique 1 :

Situation : **Gestion des stocks**

Modalité : En équipe de deux

Pondération du travail : 20%, L’addition des parties donne 100.

## Partie 1 : Modélisation 10 points

Nous faisons appel à vous pour faire un programme qui nous permettra de faire une **gestion simple des stocks**.

Vous devez choisir le sujet de votre entreprise et permettre la gestion de stock qui s’y rattache.

Par exemple une Entreprise qui utilise :

1. des composants d’ordinateur pour fabriquer des ordinateurs sur mesure
2. des pièces pour automatiser des machines industrielles
3. des pièces pour la construction de réseau informatique
4. des pièces pour un système de surveillance domiciliaire

Votre entreprise fabrique des produits. Les projets que vous réalisez demandent plusieurs pièces. Chaque projet porte un nom et inclut une description détaillée. Comme beaucoup d’autres entreprises, vous disposez d’un magasin. Une opération récurrente y a lieu : l’imputation. Celle-ci désigne le processus par lequel un employé se rend au magasin pour récupérer une certaine quantité d’une pièce destinée à un projet spécifique.

Pour chaque employé, nous conservons les informations suivantes : nom, prénom et adresse email. Pour un projet, nous conservons le nom du projet, la description et la compagnie pour qui est réalisé ce projet.

Chaque pièce est enregistrée avec un numéro unique, une description et un numéro couramment utilisé sur le marché. Nous comptabilisons la quantité en stock de chaque pièce pour chaque projet.

Pour chaque projet, une quantité prévue est définie par pièce, représentant le nombre maximum de pièces attribuées à ce projet. Il est possible que la quantité en stock soit inférieure à la quantité prévue si la pièce n’a pas encore été reçue ou si des pièces ont déjà été imputées.

Lorsqu’un employé se rend au magasin pour récupérer des pièces, il doit indiquer la pièce souhaitée et le projet pour lequel il souhaite faire une imputation. En plus des informations sur l'employé, nous enregistrons la date de l’imputation et la quantité imputée de chaque pièce. Le système vérifie que les pièces demandées sont en stock et que la quantité maximale prévue pour ce projet n’est pas atteinte. Notre inventaire se met automatiquement à jour après chaque opération.

Vous n’avez pas à faire la réception des pièces ici.

### Faites la modélisation de ce cas :

* Remettez le schéma **conceptuel** fait avec looping (fichier.loo). (Rappel : pas de clé dupliquée au conceptuel).

N.B. Pour vous aider dans votre modélisation, imaginez faire l’ajout de données dans votre schéma physique (MLD). Pouvez-vous retrouver les informations du cas décrit ? Pouvez-vous répondre aux demandes de notre usager ?

### remise : Le **3 février** avant minuit

# Travail pratique 1

Situation : **Gestion des stocks**

## **Partie 2 :** schéma de la base de données

1. Vous devez créer la base de données et les tables avec le **langage SQL**. Votre travail doit être dans **un script (.SQ).**

Vérifiez si votre base de données existe à l’exécution de votre script et si c’est le cas détruisez là et permettez de la recréer.

Les instructions doivent s’exécuter au complet (GO entre chacune et sans erreur). Partez de **mon script de base** pour ne rien oublier.

Bien entendu, vous devez gérer les contraintes d’intégrité (clé primaire, null, not null, contrainte check, défaut, contraintes de références, contrainte d’unicité…). Vous devez respecter ces consignes :

1. Il n’y aura pas de destruction en cascade ni de mises à jour en cascade.
2. Les clés qui peuvent l’être seront **générées automatiquement**. Vous devez utiliser les types **unicode** lorsque possible (**n**char ou **n**varchar). Pour être constant, en général, vos champs caractères seront longs de **200**. Le nom de vos tables commencera par **tbl\_.** Le nom des clés gardera le même format exemple no...ou ID…
3. Vous devrez avoir des tables qui sont **entièrement définies dans un seule create** (pas de alter table sur cette table par la suite, donc toutes les contraintes sont dans le create). La table que vous choisirez pour ce cas doit inclure une clé étrangère. Mettez un commentaire pertinent pour m’indiquer cette table.
4. De même certaines tables seront **définies en un create et plusieurs alter table** (c’est à dire la clé primaire sera définie après la création de la table (alter table), de même pour les contraintes de références et autres contraintes demandées.) La table choisie doit inclure une clé étrangère. Mettez un commentaire pertinent pour m’indiquer cette table.
5. Vous ajouterez ce qu’il faut pour s’assurer que la **quantité** en stock de votre pièce pour un projet ne sera pas plus petite que zéro, de plus, la **quantité prévue** ne pourra pas être de zéro ou moins. La quantité imputée ne pourra pas être de zéro et moins. Bien entendu ces champs doivent avoir une valeur (pas null). Faites ces trois cas en utilisant ces deux manières différentes : 1- directement dans le create et 2- dans un deuxième temps après la création de la table.
6. Assurez-vous que la date d’imputation doit être plus petite ou égale à la date d’aujourd’hui.
7. Le champ **email** devra avoir minimalement un @. Assurez-vous qu’il puisse être de 200 de long comme tous vos champs caractères. Et il devra être unique.
8. Volontairement, laissez un **champ null** lors de sa création dans une table. Puis avec une instruction, corriger ce champ en le mettant not null. Indiquez-moi clairement ce cas dans vos commentaires.
9. Bien entendu, pour un projet, on ne peut pas lui attribuer 2 fois la même pièce (on donnera plutôt des quantités plus grandes qu’ un !). Soyez-en sure en ajoutant ce qu’il faut. (indice : unicité des valeurs).
10. Le **nom du projet** ne pourra pas être identique à un autre projet. Faites l’intervention nécessaire sur ce champ (indice : **unicité** des valeurs). De plus le nom de compagnie ne peut pas être identique à une autre compagnie. Et les numéros d’une pièce de l’industrie doivent être tous différents.

## Partie 3 : Jeu d’essai

1. Afin de vous assurer que votre travail est bon, vous pouvez vérifier avec un **diagramme** dans SQL serveur. Visuellement, il vous sera possible de vérifier vos clés primaires et vos références demandées.
2. Nous avons utilisé un générateur de données ensemble pour les noms, prénoms de nos **employés** ([generatedata.com](https://generatedata.com/generator) ). Ces données ont été placées dans un fichier Excel. Gemini vous a généré des **pièces réelles** selon votre cas. Vous devez utiliser les données du fichier Excel fourni (format .xls). Nous avons fait ensemble les manipulations pour importer les données dans une BD partagée du serveur au cégep.
   1. Pour travailler sur ton serveur à la maison ou sur ton portable, fais le **même import** sur ton serveur (voir fichier Excel .xls) . Ainsi lors de l’exécution de ton script au cégep ou à la maison, tes données pourront s’ajouter à ta BD grâce à ton script.
   2. Vous devez **insérer les données de la BDDonneesTP** en batch. Vous devez faire l’insertion de ces données dans votre bd du travail pratique **en utilisant une seule instruction SQL pour les employés et une seule pour les pièces**. Cette instruction SQL est placée dans votre script de création de votre BD. N’utilisez que les colonnes nécessaires.
   3. Faites une **instruction SQL** qui vous affiche les employés qui ont un nom et prénom identique à d’autres. Nous voulons voir le nom, prénom et son email. Assurez-vous qu’il n’y ait pas de doublons. (select avancée, indice Cross Apply)
3. Vous ferez l’ajout **d’au moins trois** données par table en utilisant le langage SQL.

* Ces données doivent être réalistes.
* Lors d’ajout en SQL, même si les **noms des champs** peuvent être omis, indiquez-les pour une meilleure compréhension. Dans le cas d’un numéro de clé étrangère, mettre un select entre parenthèses où vous placerez dans le where une valeur significative de cette autre table. (Exemple, on verra le nom de la pièce et son numéro dans l’industrie au lieu du numéro de pièce dans votre commande SQL). Il est possible que vous deviez utiliser des inner join par moment.

Exemple à la place d’une valeur entière :

select no\_piece

from tbl\_piece

where piece = 'Capteur de Proximité Inductif' and

no\_pieceIndustrie = '872C-D5N18-B4')

* 1. Pour la table de **projet**, ajouter **4 données,** dont 2 pour la **même compagnie**.
  2. Pour la table des **projets-pièces**, faites des ajouts pour **2 projets** **différents** et pour chacun utiliser au **moins 3** pièces différentes. Une même pièce sera **dans les 2 projets**.
  3. Pour la table **d’imputation**, faites :
     1. Utiliser comme employé, un employé ayant le **même nom** qu’un autre. Pour un même projet, un des employés aura 2 imputations et l’autre une.
     2. Ajouter 3 imputations pour le même projet.
     3. Ajouter 2 pièces **différentes** parmi celles importées et une **pareille**. Bien entendu, ce doit être les pièces déjà associées à ce projet (**projets-pièces**).
     4. Assurez-vous que ce soit toujours la **date d’aujourd’hui** lorsque vous exécutez votre script.
     5. Répéter la même chose pour un 2ème projet.

1. Donnez-moi **UNE** seule instruction select pour garantir des bons ajouts demandés. (Affichez tous les champs 1 seule fois et prenez toutes les tables, n’affichez pas les numéros automatiques). Servez-vous du menu requête, concevoir une requête. Vérifier que vos données correspondent à ce qui était demandé.
2. Faites une **instruction SQL** qui vous affiche, pour chaque pièce, le nombre d’imputations réalisé en tout dans le magasin, la quantité totale de pièces imputées, et le nombre de projets dont elle fait partie. (Select avancé, indice table dérivée).
3. Testez vos contraintes pour vous assurer de leur bon fonctionnement. Vous devez placer les instructions prouvant le **non-fonctionnement** de certaines valeurs. **Mettez-les en commentaire** dans votre script pour **chacune** des contraintes (voir fichier **réponse**). (email, quantitePrevu , quantiteStock, quantitePieceImpute , date d’imputation, nombre de pièces différentes par projet, nom de compagnie unique , no\_piece de l’industrie unique). Vous pouvez utiliser les numéros de clé étrangère directement dans l’instruction SQL. Attention le message d’erreur doit correspondre à ce qui est vérifié)

### **Remise**: partie 2-3 (25%) Le **25 février** avant minuit