**TDs et TPs à faire**

## QCM (10pts)

## Voir le dossier QCM contenant les fichiers (\*.ppt)

## Questions de cours (10pts)

1. Il existe quatre types de modèles utilisés couramment en modélisation de données, recensez ces modèles et donnez un exemple d’application pour chaque modèle de données ? (2 pts)
2. Qu’est ce qu’un Système d’information décisionnel et quel est son objectif ? (2 pts)
3. Dessine l’architecture d’un DataWare House et décrivez chacune de ses couches ? (2 pts)
4. Pour concevoir des data marts on peut utiliser trois types de BD, lesquels ? (3 pts)
5. Le modèle dimensionnel peut être représenté par 2 types de schéma, lesquels ? (1 pts)
6. Donne la définition de : (3pts)
   1. table de faits  transactionnelle
   2. table de faits périodique
   3. table de faits  cumulative
7. Selon l’additivité, il y a trois types de faits, lesquels ? (1.5 pts)
8. SCD est l’acronyme de quoi ? (1pts)
9. Donne une définition de SCD ? (1pts)
10. Quelle est la différence entre clé de substituion et une clé naturelle ? (1.5pts)
11. Il existe plusieurs types de tables SCD, donne une définition de chaque type ? (2pts)s

## Exercice 1 (10 points) :

Vous êtes chargé de créer un Data Warehouse comme étant un consultant BI, Les données sources proviennent directement du système des ventes utilisé par l’entreprise. La structure du tableau des sorties est illustrée dans la figure :

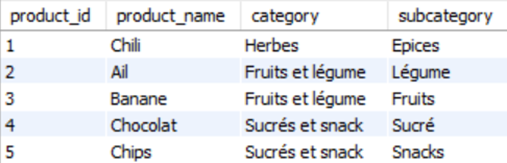
**Exigences :**

Les responsables veulent analyser la quantité des ventes et les montants des transactions. Ils veulent aussi analyser ces mesures par mois, trimestre et par année. Vous pouvez ajouter des tables supplémentaire si c’est nécessaire.

**Travail demandé :**

1. Identifier les faits et les dimensions (5 pts)
2. Dessiner la table de faits et les tables de dimensions en mentionnant pour chacune des tables les clés primaires, étrangères, et les autres colonnes. (5 pts)

## Exercice 2 (10 points) Modèle en étoile vers Modèle en flocoon :

Soit une table produits comme illustré dans la figure :

Donne les requêtes SQL qui répondent aux questions suivantes :

1. Afficher seulement les catégories depuis la table produits (éviter les répétitions) (2 pts)
2. Créer une deuxième table Categorie contenant uniquement une seule colonne category. category ressemble à la colonne category de la table Produits. (2 pts)
3. En utilisant la requête insert into et select insérer les catégories de la table produits dans la table Categorie (2 pts)
4. Donner la syntaxe SQL pour créer une troisième table Categorie\_Table ayant deux colonnes (category\_id,category) (2 pts)
5. Utiliser la combinaison de la requête Insert into et la requête « select row\_number() over (order by catégorie) as id\_catégorie, catégorie from Categorie », pour insérer les enregistrements de la table categorie dans la nouvelle table Categorie\_Table. (2 pts)

## Exercice 3 (10 points) Remplissage préalable de la dimension date:

On veut créer une table dimension *date\_dim* qui va contenir les lignes représentant tous les jours d’une année (de 2024-01-01 jusqu’à 2024-12-31) afin d’utiliser cette dimension avec une table de fait gérant les ventes.

Pour faire cela, donne les requêtes SQL qui répondent aux questions suivantes :

1. Créez la table date\_dim sans oublier les contraintes not null, clé primaire et l’index pour la colonne Date. (1 pts)
2. Créée une table ‘nombres’ avec une seule colonne ‘id’ qui va représenter après les id naturels des jours de l’année. (1 pts)
3. Créer une procédure genererNombre(start idDepart, idFin) qui va inserer les id de idDepart jusqu’à idFin dans la table nombres. (1 pts)
4. Appeler la procédure en lui passant 0 et 400 comme parametres. (1 pts)
5. Soit la requete "*SELECT id, DATE\_ADD('2024-01-01', INTERVAL id DAY) AS dateGeneree FROM numbers;*" Adapte la requête pour retenir seulement les dates inférieures ou égales à 2024-12-31. (1 pts)
6. Rappelez vous de l’une des syntaxes des requêtes imbriquées : "*select columnsNames from (select \* from tableName) as resultatSousRequete ;"* donne maintenant la requete qui affichera les colonnes id, dateGeneree et Jour\_Semaine (le nom du jour de la semaine). (1 pts)
7. Adapter la requête précédente pour afficher aussi *Jour\_Mois, Mois\_num, Mois\_nom, Année.* (2 pts)
8. Utilise la combinasion de requete insert et select pour inserer le resultat de la requête précédente dans la table date\_dim. (2 pts)

## Exercice 4 (10 points) Dimensions à évolution lente :

Soit une table produits comme illustré dans la figure. Nous aimons mettre en place une table SCD pour gérer le changement de category des produits au fil du temps.

Pour faire cela, donne les requêtes SQL qui répondent aux questions suivantes :

1. Créez une nouvelle table de dimension "Produits\_SCD" avec les mêmes colonnes que la table "Produit", ainsi que trois colonnes supplémentaires : 1- ID\_Historique (clé primaire), 2- Date\_debut\_validite 3- Date\_fin\_validite. (2 pts)
2. Copiez les données de la table " Produits " vers la table "Produits \_SCD" en initialisant la colonne "Date\_debut\_validite" avec la date "01/02/2024" pour tous les enregistrements. (2 pts)
3. On veut changer la catégorie du produit Chips ayant l’id 5 (de "Sucrés et snack" vers "Aliments apéritifs") dans la table "Produits \_SCD", pour cela :
4. Mettez à jour la colonne "Date\_fin\_validite" de l’enregistrement actuel avec la date "31/02/2024". (2 pts)
5. Insérez un nouvel enregistrement dans la table "Produits\_SCD" avec les informations de la nouvelle catégorie. (2 pts)
6. Imaginez que nous avons changer la catégorie du produits (ex. Chips) dans la table "Produits\_SCD" plusieurs fois. Pour récupérer la dernière catégorie valide du produit (ex. Chips) à une date donnée, utilisez une requête SQL avec une clause WHERE spécifiant la date et sélectionnez la ligne correspondante avec la date la plus récente et une date de fin de validité nulle. (2 pts)

## Exercice 5 (10 points) modèles dimensionnels :

Considérons un système de suivi des performances sportives. Les données disponibles sont les suivantes :

1. ID de match
2. Date du match
3. Nom de l'équipe à domicile
4. Nom de l'équipe à l'extérieur
5. Score de l'équipe à domicile
6. Score de l'équipe à l'extérieur
7. ID du joueur
8. Nom du joueur
9. Position du joueur
10. Temps de jeu du joueur dans le match

**Questions :**

1. Identifiez les faits et les dimensions.
2. Proposez une modélisation dimensionnelle du problème.
3. Créez la table de faits "Performances" avec les colonnes nécessaires pour stocker les données en utilisant SQL.
4. Identifiez les colonnes qui serviront de clés étrangères pour les dimensions.
5. Proposez les définitions des tables de dimensions associées en utilisant SQL.
6. Ajoutez des contraintes de clé étrangère pour les colonnes correspondantes dans la table de faits en utilisant SQL.