

Projet MATSKI : Partie 1

Contexte

La société Matski est une société spécialisée dans la revente de matériel de ski et de matériel de montagne. Elle a été créée en 1984 à Annecy (Haute-Savoie) sous l'impulsion de 3 entrepreneurs-es. Matski propose une gamme de produits, principalement des articles et accessoires pour les skieurs et skieuses alpin-es et nordiques, ainsi que des luges en bois et en plastique. Depuis 2010, elle dispose d'un logiciel de gestion commerciale et d'un site B2B.

L'entreprise souhaite mieux exploiter les données existantes et développer une extension de sa base de données pour une meilleure gestion des ses contacts et des différentes catégories d'articles qu'elle gère. Pour cela, la qualification et l'extension de l'actuelle base de données est nécessaire.

Travail demandé

Ce projet est subdivisé en deux parties :

- **Partie 1** : une analyse qualitative de la base de données existante avec la proposition d'une solution corrective (**durée 4 semaines**)
- **Partie 2** : une extension de la base de données existante, ainsi que la proposition d'un nouveau jeu de données (**durée 2 semaines**)

Dans ce qui suit, nous allons détailler le travail demandé pour la première partie. Pour plus de détails sur l'organisation du projet et des rendus, consultez le document "*Fiche SAé S1.04.pdf*" mis à votre disposition dans l'espace moodle dédié à cette SAé.

1 Partie 1

L'entreprise souhaite qu'à partir de la structure de la base de données fournie (cf. script SQL *BaseMatski.sql* pour la création et le peuplement de la base de données) vous donniez votre avis et proposiez des modifications si nécessaire. Cette base ne contient que les données sur les commandes des clients, sachant que pour commander on renseigne également les quantités livrées qui correspondront aux quantités facturées.

La Figure 1 montre la représentation des catégories d'articles au sein de l'entreprise.

L'entreprise désire également que vous interrogiez les données afin de faire une analyse des données économiques de l'entreprise.

1. Il semble en effet que certains articles ou catégories ne soient jamais commandés, est-ce le cas ?

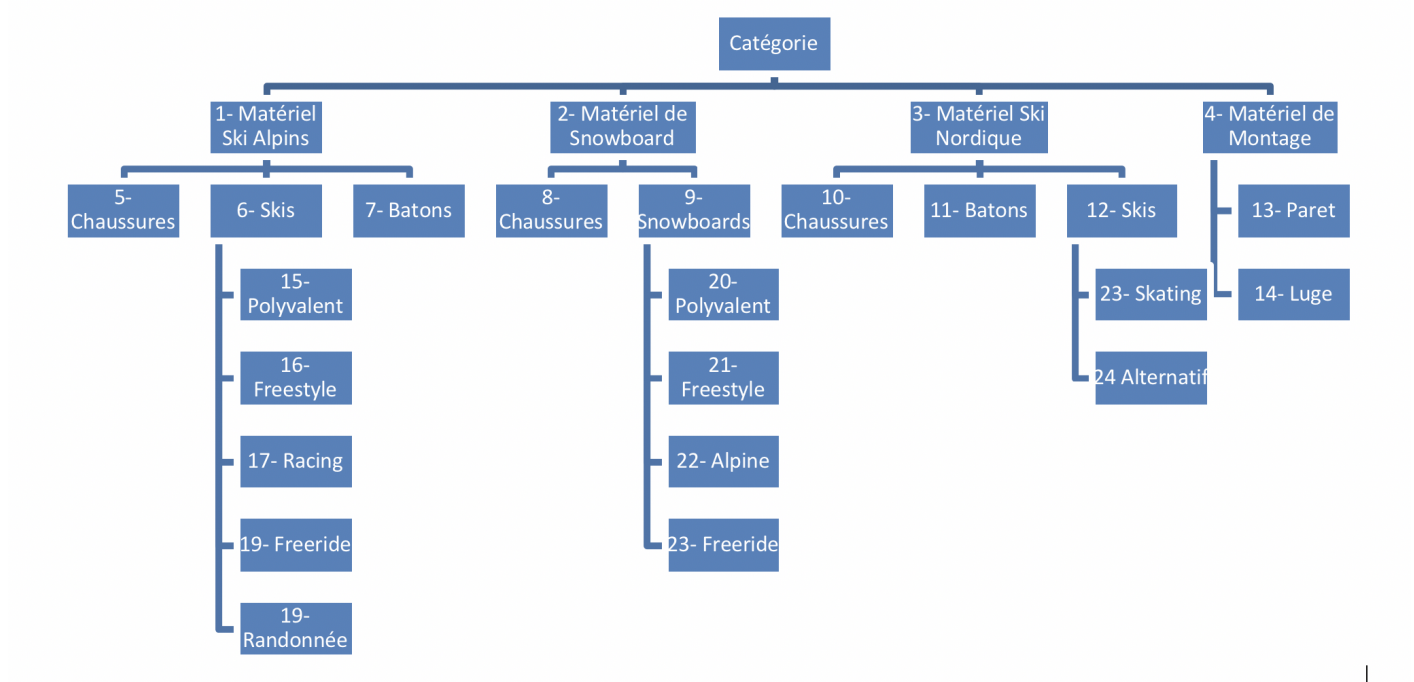


FIGURE 1 – Catégories d'articles

2. Une action commerciale a été menée il y a 3 semaines sur les client·es basé·es en Suisse, l'entreprise souhaiterait savoir s'il y a eu un effet sur les commandes
3. Existe-t-il des client·es qui n'ont jamais passé de commande ?
4. Il semble que certaines commandes ne soient pas livrées totalement. Le phénomène est-il inquiétant ?
5. La/Le comptable de l'entreprise sollicite votre aide pour l'aider à remplir la déclaration de TVA pour le dernier mois et notamment la TVA collectée sur les ventes. En fonction des types de client·es, elle/il a besoin du montant des ventes correspondant à trois ventilations :
 - (a) Vente à des client·es en France
 - (b) Vente à des client·es étrangers·ères hors UE
 - (c) Vente à des client·es étrangers·ères dans UE

L'information sur ces trois types est indiquée dans la table *Etiquette* sous la forme d'un code type TVA.

1.1 Ce qui est demandé

1. Exécutez le script *SQL BaseMatski.sql* après vous être connecté·e à votre base de données *Postgresql*. Analyser la base de données obtenue.
2. À partir des tables et jeux d'essais fournis, proposez une représentation du diagramme de classes (DC) correspondant (retro-ingénierie)
3. Vérifiez la qualité des informations présentes dans la base (anomalies sur certains attributs), et développez un questionnement sur la modélisation proposée (respect des règles de gestion et des contraintes, redondance, valeurs NULL, décomposition, etc.)

4. Proposez une solution (nouveau diagramme de classe et son implémentation en SQL) permettant de corriger les anomalies détectées dans l'étape précédente
5. Interrogez la base existante et corrigée via des requêtes SQL pour répondre aux différentes questions métiers du client·e. Plus précisément, traduire les questions métiers en requêtes SQL, et analyser les résultats obtenus. Chaque schéma de requête sera stocké dans une "Vue" (le concept de *Vue* sera abordé en séance de TD)

2 Contraintes et évaluation

2.1 Travail en binôme

Le projet est impérativement à réaliser en binômes. Toute exception devra avoir été acceptée par l'enseignant·e. Vous travaillerez sur une base de données commune (chaque binôme aura sa propre base de données).

Chaque membre du binôme doit s'investir à part égale dans le projet. Le travail doit donc être réparti équitablement entre ses membres. La participation de chacun·e doit être clairement identifiée dans les rapports qui seront rendus.

Les notes seront individuelles et prendront en compte la participation réelle de chacun·e au travail demandé, telle qu'elle aura pu être vérifiée lors des évaluations.

2.2 Consignes

La partie modélisation (diagramme de classe) sera réalisée à l'aide de l'outil "Visual Paradigm" ou "BOUML".

L'implémentation du schéma relationnel de la base de données corrigée sera réalisée avec le SGBD *Postgresql* à l'aide d'un nouveau script SQL que vous nommerez *BaseMatskiUpdate.sql*.

Les consignes suivantes devront obligatoirement être respectées :

- Les noms des schémas seront définis en minuscules
- Les relations (tables) seront définies au sein d'un schéma nommé "matski_update".
- Les noms des relations de base seront définis en minuscules et préfixés par "_".
Exemple : `_article`, `_client`, ...
- les contraintes seront nommées explicitement
- les noms d'attributs seront définis en minuscules et reproduiront les variables d'instance ou noms de rôle (au singulier) du diagramme de classes

2.3 Rendus

Deux rendus sont attendus dans cette première partie.

1. Rapport incluant la critique de la base de données existante, et la proposition d'un nouveau diagramme de classe (incluant les corrections) et son implémentation sous *Postgresql*. Chaque anomalie identifiée doit être expliquée, et une correction proposée. Le format du rendu sera : (1) un fichier *PDF* incluant les critiques, la correction proposée, et le diagramme de classe correspondant, et (2) le script SQL de la nouvelle base de données (rendu à **T0+ 2 semaines**)

2. Réponses aux différentes requêtes demandées. Le format du rendu sera un script SQL incluant toutes les requêtes et les résultats correspondants. Pour chaque requête, proposez une interprétation en 1 ou 2 phrases maximum du résultat obtenu (à inclure dans le script SQL sous forme de commentaire) (rendu à **T0+ 4 semaines**)