

4<sup>ème</sup> année - Options SIB & SEN

**TD INITIATION AUX BASES DE DONNEES**

Mars 2016

Erwan TRANVOUEZ

Vous pouvez utiliser le logiciel JMerise (<http://www.jfreesoft.com/JMerise/>)

**1. Modélisation :**

- **Modéliser la Base de données permettant de produire les informations suivantes :**

Samsung Galaxy S7 Edge Smartphone débloqué 4G (Ecran : 5,5 pouces ·  
Android Marshmallow 6.0) Noir - 32 Go - 4 Go RAM - Simple Nano-SIM -

[Offres de Pâques : Retrouvez toutes les Ventes Flash High Tech](#)

de Samsung

★★★★☆ 10 commentaires client | 5 questions ayant reçu une réponse

Prix : **EUR 799,00** LIVRAISON GRATUITE Détails

Tous les prix incluent la TVA.

En stock.

Expédié et vendu par Amazon. Emballage cadeau disponible.

Note: cet article est éligible au click and collect. [Détails](#)

8 neufs à partir de **EUR 620,00**

Taille: **Version France**

**Version France**

Version Italie

Style: **Galaxy S7 Edge**

Galaxy S7

**Galaxy S7 Edge**

Couleur: **Noir**

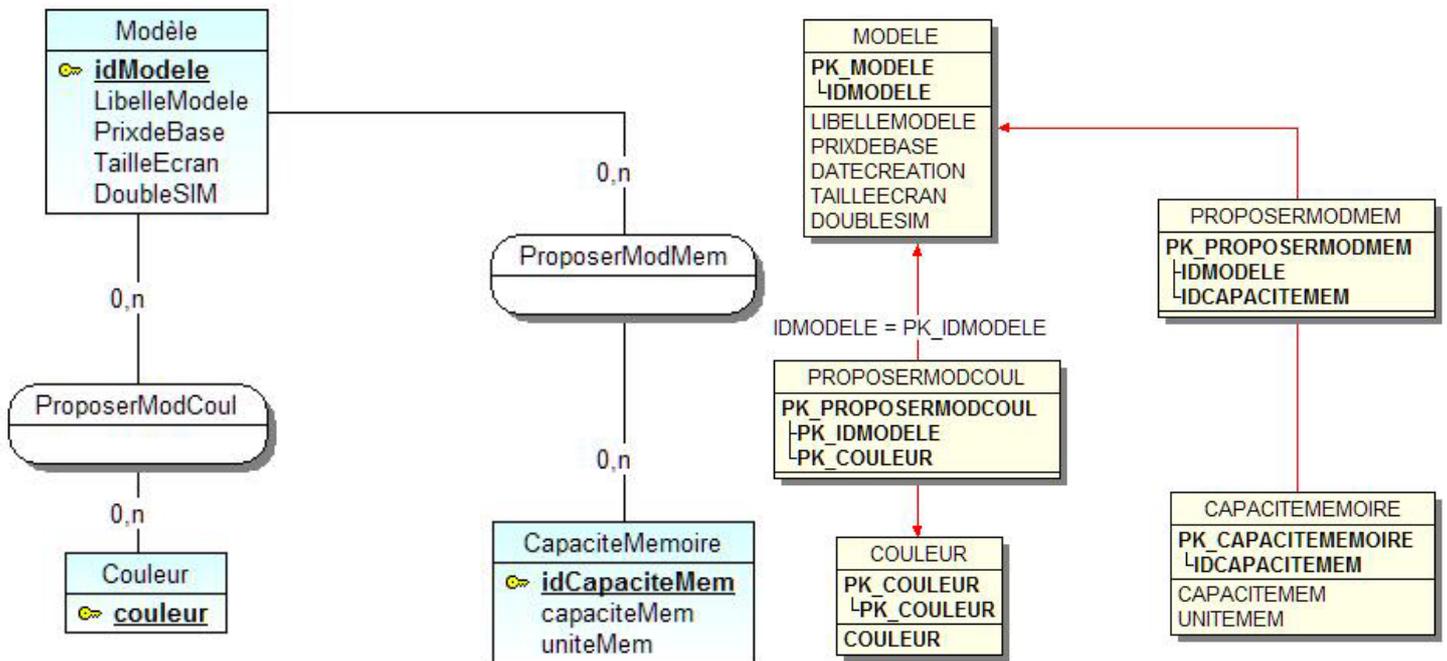


- Écran aux bords incurvés de 5,5" Super AMOLED Quad HD
  - Appareil photo 12MPix avec technologie Dual PIXEL
  - Port microSD pour plus de stockage : jusqu'à 200Go
  - Étanchéité : Certification IP68
  - Processeur Octo Core 2,3 GHz
  - Charge rapide et à induction sans fil
  - Garantie Constructeur : 2 ans
- > [Voir plus de détails](#)



☞ la souris sur l'image pour zoomer

## MCD simplifié pour partie SQL



1. Ecrire les requêtes SQL permettant de créer les tables dans un fichier texte

2. Charger le script SQL ainsi créé avec la commande SQLite `.read scriptCreation.sql`

3. Insérer des données.

A noter que SQLite enregistre les dates comme un texte (string) avec le format suivant : "AAAA-MM-JJ" , "AAAA-MM-JJ HH:MM"

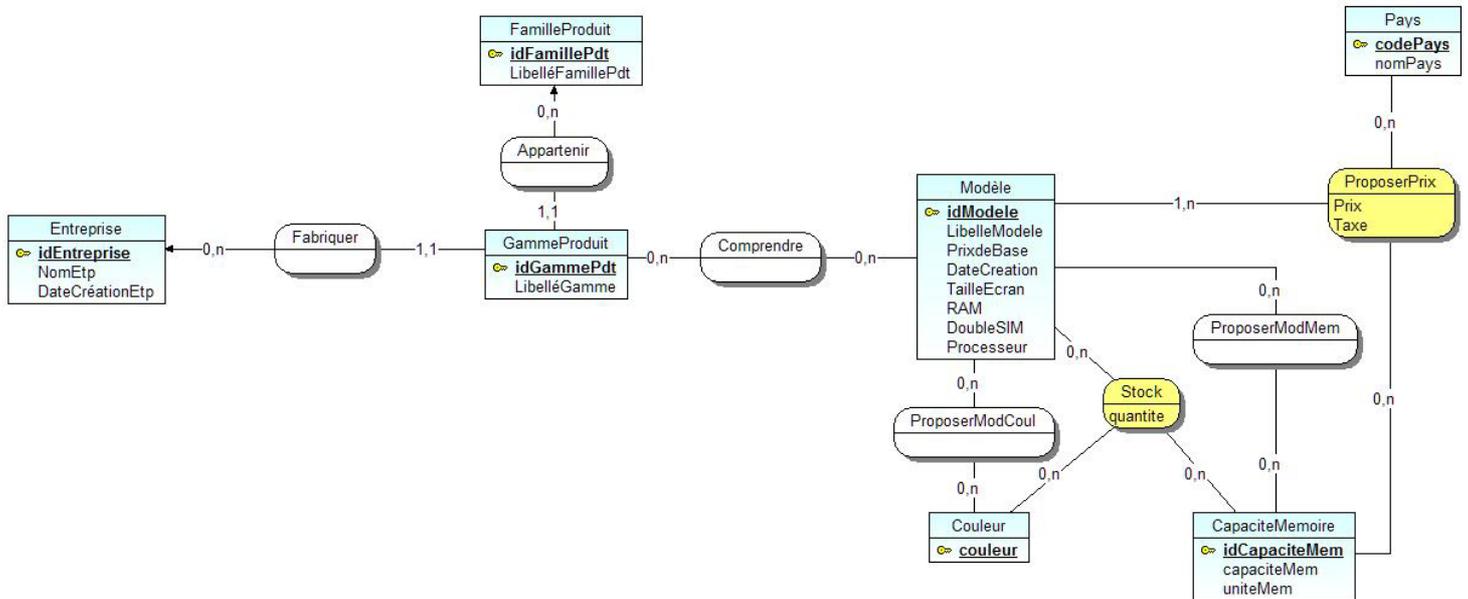
De même le format boolean ne reconnaît pas les mots clés true ou false. Il faut utiliser leur équivalent : 0 (false) et 1 (true)

4. Donner les scripts SQL permettant de répondre aux questions suivantes :

- Liste des modèles
- Liste des modèles avec double sim
- liste des modèles d'une taille particulière
- liste des modèles créés après une date
- liste un modèle et ses différentes configurations par couleur
- liste un modèle et ses différentes configurations par capacité
- liste un modèle et ses différentes configurations
  
- compter le nombre de modèles
- donner le prix moyen des modèles
- compter le nombre de modèles de couleur blanche
- compter le nombre de modèles par couleur
- compter le nombre de modèles par capacité
- compter le nombre de modèles ayant une couleur particulière et une capacité choisie

# Corrigé

MCD :



Le catalogue de produit peut être distingué en 2 catégories :

- Ce qui relève de la description des produits proposés par le site internet (équivalent à ce qu'on aurait dans un catalogue papier)
- Ce qui relève de la gestion commerciale (information spécifique au site internet)

A noter que ne figure pas toute la partie commerciale liée aux ventes (notion de client, commande etc.)

**Discussion sur l'entité Couleur :** on pourrait penser que le gain est réduit par rapport à la solution d'ajouter couleur directement comme un attribut dans Modèle.

Cela pourrait être corrigé en ajoutant un identifiant court (si nombre de couleurs réduit, on peut utiliser un entier court) => au lieu d'avoir (ce qui deviendra au niveau du MLD) une clef étrangère de 16 caractères par exemples, quelque chose qui occupe 8 fois moins de mémoire.

Mais cette solution se justifie surtout par la volonté de lister **les** (donc plusieurs) **couleurs** proposées par **un** modèle, sachant que cela peut varier selon les modèles. La discussion ci-après détaille

Ainsi, il est possible de restreindre, lors de la définition de la table en SQL (CREATE TABLE), de restreindre la liste des valeurs autorisées (ex. ajouter à la suite de la définition de la colonne couleur CHAR (16) CHECK (couleur IN ('Blanc', 'Noir', 'Bleu' ) MAIS 2 problèmes sont alors soulevés :

- si l'on veut ajouter a posteriori (après déploiement du site internet par exemple) de nouvelles couleurs il faudra modifier la définition de la table (avec la requête ALTER TABLE) ce qui suppose des droits d'accès plus fort pour l'utilisateur de la Base de données, alors que l'ajout d'un enregistrement dans la table Couleur est alors plus facile à déléguer.
- par ailleurs, la liste de valeur est valable pour tous les enregistrements ET SURTOUT il ne sera possible de saisir **qu'une seule valeur de couleur par modèle**. Donc pour avoir toutes les couleurs il faudrait créer ... autant de modèles que de couleur ("Modele Bleu", "Modele Blanc" etc...). Cela aura un coût mémoire important avec la duplication de tout l'enregistrement initial avec comme seul changement la couleur...

Pour conclure sur cette question : il faut faire la part des choses entre l'approche conceptuelle qui décrit une solution "propre" (pour ne pas dire "pure") de la structuration des informations et les solutions techniques qui peuvent être mises en oeuvre... pour des raisons techniques (simplification du nombre de table, choix de relâcher la contrainte d'optimisation de la mémoire pour réduire les coûts de traitement des requêtes...) le développeur pourra faire des choix



- **Modéliser la Base de données permettant de gérer les inscriptions à un établissement d'enseignement supérieur.**

