# **Creer et requeter la base de données d'un site de e-commerce**

## Contexte du projet

Votre mission est divisée en trois temps:

* Complétez l'ébauche de modèle conceptuel élaboré par un de vos collègue (les relations sont bonnes mais il n'y a pas les cardinalités, il n'y a pas non plus les attributs et leurs type). Créez également le modèle physique
* Créez une BDD SQlite, créez les tables correspondants au sujet et ajoutez les données depuis les fichiers csv.
* Faites les requetes demandées pour mener à bien une première analyse de la base

## Modalités pédagogiques

Vous êtes par équipe de 2, vous avez 1 journée et demi pour réaliser ce brief

## Modalités d'évaluation

Correction en classe.

## Livrables

Un github avec:

- vos modèle conceptuel et physique au format jpeg

- les fichiers csv et votre db dans le gitignore

- les requetes permettant de répondre à chaque question dans des fichiers .sql

Plan marketing d’un site de e-commerce brésilien.

Vous avez obtenu les différentes tables d’un site de e-commerce.

# I Importation des tables et comprehension des tables

1. Commencer par importer les tables dans un nouvel ensemble de données créé ad hoc dénommé “Olist” (ce qui correspond au nom de l’entreprise.
2. Pour chacune des tables indiquez quelle est la clef primaire et quelles sont les clefs étrangères (ainsi que la table vers laquelle elles pointent) comprises dans chaque table.

**Informations sur les tables:**

Customers:

* customer\_id: key to the orders dataset. Each order has a unique customer\_id.
* customer\_unique\_id: unique identifier of a customer.

order\_items

* shipping\_limit\_date: Shows the seller shipping limit date for handling the order over to the logistic partner.
* freight\_value: item freight value item (if an order has more than one item the freight value is splitted between items) (frais de livraison)

reviews:

* review\_creation\_date: Shows the date in which the satisfaction survey was sent to the customer.
* review\_answer\_timestamp: Shows satisfaction survey answer timestamp.

orders

* order\_delivered\_carrier\_date: Shows the order posting timestamp. When it was handled to the logistic partner.
* order\_delivered\_customer\_date: Shows the actual order delivery date to the customer.
* order\_estimated\_delivery\_date: Shows the estimated delivery date that was informed to customer at the purchase moment.

order\_payments

* payment\_sequential: a customer may pay an order with more than one payment method. If he does so, a sequence will be created to
* payment\_installements: number of installments chosen by the customer. (nombre de versements)

# II Analyse de la base customers:

* Les clients viennent de combien de villes et d’état différents?
* Dans l’état de Sao Paolo (SP), combien y a t il de ville différentes?
* La table présente-t-elle des valeurs nulles?
* La table présente deux id, (un de jointure, l’autre comme clef primaire). Est ce que l’un de ces ID est en doublon?
* Quelle est la relation théorique entre customers et orders (one to many ou many to one)?
* Quelle relation est encodée dans les bases?
* A partir de l’analyse précédente, calculez le nombre réel de client par état (state)

# III Analyse de la base orders:

* On cherche à calculer le nombre de commandes, le nombre de produits vendus et le chiffre d'affaires (qui n’intègre pas les frais de port) pour chaque mois et chaque état. Pour cela, à l’aide d’une requête imbriquée ou d’une table temporaire procédez en trois temps:
  + à partir de la table orders items calculez les indicateurs souhaités
  + faire une jointure ensuite avec la table order pour récupérer la date de vente
  + faire une jointure enfin sur la table consumers pour récupérer le pays.

# IV Analyse de la base sellers / produits locaux:

* Réunissez sur une même table dont chaque ligne correspond à un order item:
  + le prix du produit: order\_item
  + la catégorie du produit en anglais : categorie\_name\_translation
  + le pays du vendeur: seller
  + le pays de l’acheteur: custumer
  + la date de vente: order
* A partir de la table précédente créée deux nouvelles colonnes.
* Une colonne “local\_value” Si le pays du vendeur est le même que le pays de l’acheteur alors cette nouvelle colonne prendra comme valeur “local” s’ils sont différents la valeurs sera “inter-étatique”
* Une colonne “local\_indicatrice” qui suit la même règle mais qui prendra la valeur 1 si l’état du vendeur et de l’acheteur sont les mêmes ou 0 autrement.

# V Qualité de la description des produits

On cherche à mesurer l’effet de la description des produits sur les ventes (taille description, nombre d’images).

Pour cela on va modifier les variables “product\_description\_lenght” et “product\_photos\_qty” afin que les différentes “valeurs” qu'elles prennent aient toutes un nombre consistant de ligne. Par exemple, on ne veut pas savoir combien de description font exactement 138 caractères mais plutôt entre 100 et 150 caractères. Avant de faire cela il faut trouver les regroupements optimaux.

* Commencez par déterminer combien il y a de commandes pour chaque quantité de photos dans la description
* On voit qu'au-delà de 6 photos le nombre de commandes est marginal. Préparer la requête que vous intégrerez dans la table finale qui à l’aide d’une fonction IF() permettra de changer les valeurs de product\_photos\_qty supérieures à 7 pour les remplacer par 7 (on pourrait aussi les remplacer par “7 et plus” à condition de changer le type de product\_photos\_qty)

Pour la longueur de la description on ne peut pas utiliser la même technique. On va calculer les seuils à l’aide cette requête:

SELECT

quantile(product\_description\_lenght, 0.25) AS first\_quartile,

quantile(product\_description\_lenght, 0.50) AS second\_quartile,

quantile(product\_description\_lenght, 0.75) AS third\_quartile

FROM order\_items A

LEFT JOIN products B

ON A.product\_id = B.product\_id;

* Expliquez ce que fait cette requête:
* A présent en utilisant ces seuils et la fonction CASE, créer une nouvelle colonne “longueur\_description” qui prendra les valeurs “très courtes”, “courte”, “moyenne” et “longue”

Rappel de l’écriture d’une clause [CASE](https://cloud.google.com/bigquery/docs/reference/standard-sql/conditional_expressions):

SELECT

CASE new\_column

WHEN condition1 THEN valeur1

WHEN condition2 THEN valeur2

WHEN condition3 THEN valeur3

…

END

FROM …

Intégrez les deux colonnes créées dans une table dont chaque ligne correspond à l’objet d’une commande:

* On veut comparer la qualité de la description des produits en moyenne par rapport à la qualité des produits vendus. Créer donc ces deux variables sur une tables dont chaque ligne correspond à un produit