Étude de cas consultations

h) Le coût des consultations pour toutes les patientes pour le mois d'octobre

- Slice sexe = Femme
- Drill up sur Temps de Date à mois
- Slice mois = octobre
- Agréger *(somme)* le total des coûts

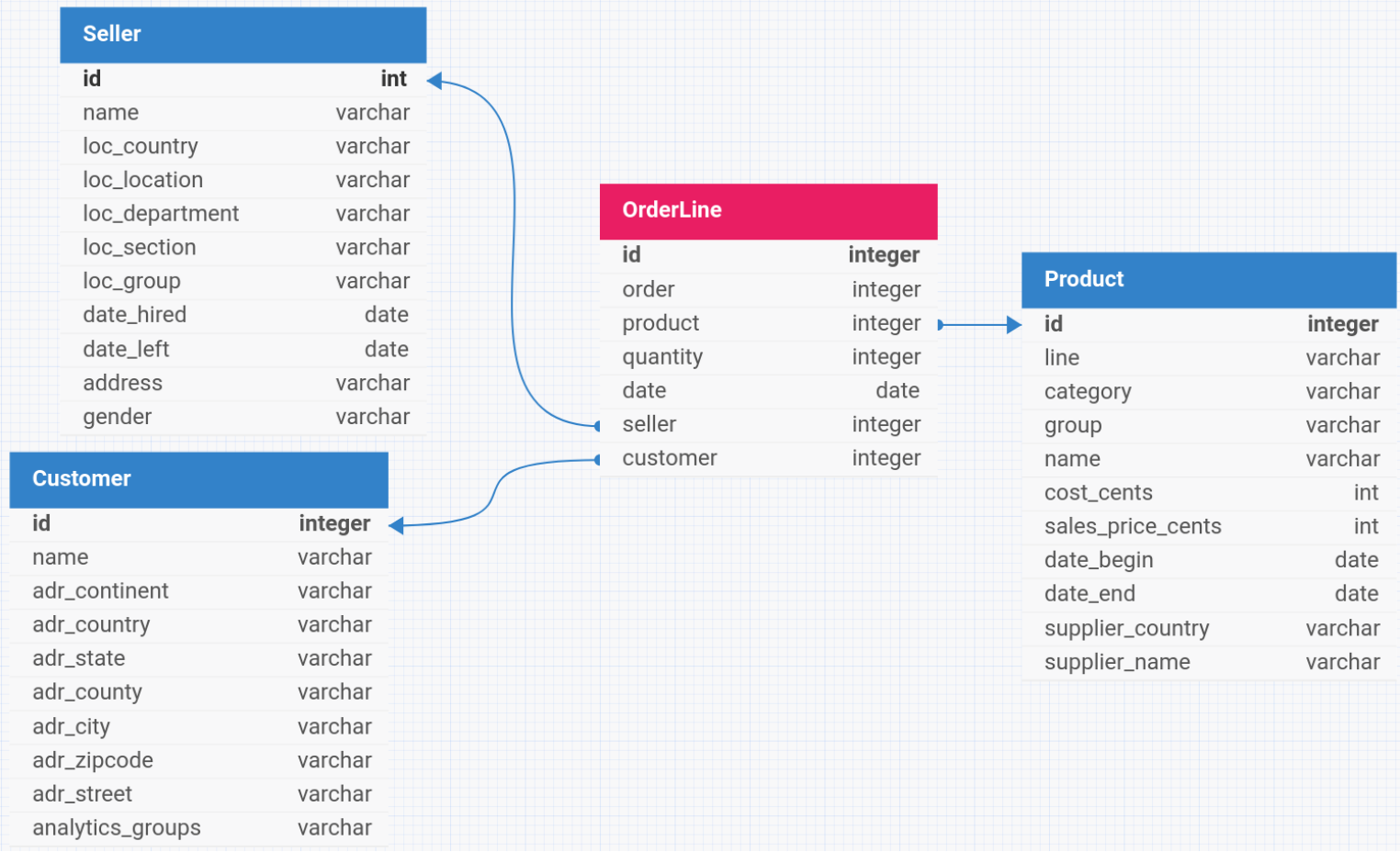
i) Le coût des consultations par patiente pour le mois d'octobre

- Slice sexe = Femme
- Drill up sur Temps de Date à mois
- Slice mois = octobre
- Agrérer *(somme)* les coûts par patient

Étude de cas Orion

1: Schéma de l'entrepôt

A. Proposez un schéma en étoile pour l'entrepôt de données de la société Orion.



B. À partir du schéma en étoile proposé, écrivez les commandes SQL nécessaires pour créer les tables de l'entrepôt de données de la société Orion. Assurez-vous d'inclure les clés primaires et les clés étrangères appropriées pour chaque table.

```
CREATE TABLE "Seller" (
    "id"         INTEGER NOT NULL,
    "name"       TEXT    NOT NULL,
    "country"    TEXT    NOT NULL,
    "location"   TEXT    NOT NULL,
    "department" TEXT    NOT NULL,
    "section"    TEXT    NOT NULL,
    "group"      TEXT    NOT NULL,
    "date_hired" TEXT    NOT NULL,
    "date_left"  TEXT,
```

```
    "address"    TEXT    NOT NULL,
    "gender"     TEXT    NOT NULL,
    PRIMARY KEY("id")
);

CREATE TABLE "Customer" (
    "id"              INTEGER NOT NULL,
    "name"            TEXT    NOT NULL,
    "continent"       TEXT    NOT NULL,
    "country"         TEXT    NOT NULL,
    "state"           TEXT,
    "county"          TEXT,
    "city"            TEXT    NOT NULL,
    "zipcode"         TEXT    NOT NULL,
    "street"          TEXT    NOT NULL,
    "analytics_group" TEXT,
    PRIMARY KEY("id")
);

CREATE TABLE "Product" (
    "id"                INTEGER NOT NULL,
    "line"              TEXT    NOT NULL,
    "category"          TEXT    NOT NULL,
    "group"             TEXT    NOT NULL,
    "name"              TEXT    NOT NULL,
    "cost_cents"        INTEGER NOT NULL,
    "sales_price_cents" INTEGER NOT NULL,
    "date_begin"        TEXT    NOT NULL,
    "date_end"          TEXT,
    "supplier_country"  TEXT    NOT NULL,
    "supplier_name"     TEXT    NOT NULL,
    PRIMARY KEY("id")
);

CREATE TABLE "OrderLine" (
    "id"       INTEGER NOT NULL,
    "order"    INTEGER NOT NULL,
    "product"  INTEGER NOT NULL,
    "quantity" INTEGER NOT NULL,
    "date"     TEXT    NOT NULL,
    "seller"   INTEGER NOT NULL,
    "customer" INTEGER NOT NULL,
    PRIMARY KEY("id"),
    FOREIGN KEY("customer") REFERENCES "Customer",
    FOREIGN KEY("product") REFERENCES "Product",
    FOREIGN KEY("seller") REFERENCES "Seller"
);
```

C. Utilisez des commandes SQL pour répondre aux questions suivantes à partir des données de l'entrepôt :

Quels sont les produits qui se vendent le mieux ?

```
SELECT P.id AS "product_id", P.name AS "product_name", SUM(O.quantity) AS "total_sales"
FROM OrderLine O LEFT JOIN Product P ON O.product = P.id
GROUP BY P.id, P.name
ORDER BY SUM(O.quantity) DESC;
```

Quels sont les commerciaux qui font le plus de ventes ?

```
SELECT S.id AS "seller_id", S.name as "seller_name", SUM(O.quantity) AS "total_sales"
FROM OrderLine O LEFT JOIN Seller S ON O.seller = S.id
GROUP BY S.id, S.name
ORDER BY SUM(O.quantity) DESC;
```

Quels sont les clients les plus rentables ?

```sql
SELECT C.id AS "customer_id", C.name AS "customer_name", SUM(O.quantity*(P.sales_price_cents-cost_cents))
AS "total_benefit_cents"
FROM OrderLine O
    LEFT JOIN Product P ON O.product = P.id
    LEFT JOIN Customer C ON O.customer = C.id
GROUP BY C.id, C.name
ORDER BY SUM(O.quantity*(P.sales_price_cents-cost_cents)) DESC;
```

Suite

Exercice 1

1: Quelles opérations OLAP faut-il appliquer pour obtenir la note moyenne des cours en Informatique pour chaque étudiant ?

- Slice `Course_name` pour garder seulement les cours d'informatique
- Agréger *(moyenne)* les notes moyennes par étudiant

2: Quelles opérations OLAP faut-il appliquer pour obtenir la charge totale payée par un étudiant spectateur à "GM Place" en 2010 ?

- Slice de `game` sur `game_name` pour garder `GM Place`
- Slice de `spectator` sur `status` pour garder `student`
- Roll up de `date` à `year`
- Slice de `date` sur `year` pour garder `2010`
- Agréger *(moyenne)* le `Charge_rate` de `spectator`

3: Équivalents OLAP

3.1

SQL:

```sql
SELECT TIME, LOCATION, PRODUCT, SUM(REVENUE) AS PROFIT
FROM SALES
GROUP BY ROLLUP(TIME, LOCATION, PRODUCT);
```

Opérations OLAP:

- Rollup de `Time`
- Rollup de `Location`
- Rollup de `Product`
- Agréger *(somme)* de `revenue` par `Time`, `Location`, et `Product`

3.2

SQL:

```sql
SELECT TIME, LOCATION, PRODUCT, SUM(REVENUE) AS PROFIT
FROM SALES
GROUP BY ROLLDOWN (TIME, LOCATION, PRODUCT);
```

Opérations OLAP:

- Drill down de `Time`
- Drill down de `Location`
- Drill down de `Product`
- Agréger *(somme)* de `revenue` par `Time`, `Location`, et `Product`

3.3

SQL:

```sql
SELECT PRODUCTS, SUM(REVENUE)
FROM SALES
WHERE PRODUCTS = 'OPV'
GROUP BY PRODUCTS;
```

Opérations OLAP:

- Slice de `Products` pour garder seulement `OPV`
- Agrérer *(somme)* de `revenue` par `Products`

3.4

SQL:

```sql
SELECT PRODUCTS, SUM(REVENUE)
FROM SALES
WHERE PRODUCTS = 'EL' AND LOCATION = 'EUROPE'
GROUP BY PRODUCTS;
```

Opérations OLAP:

- Slice de `Product` pour garder seulement `EL`
- Slice de `Location` pour garder seulement `Europe`
- Agrérer *(somme)* de `revenue` par `Products`

4: Déterminer les dimensions et hiérarchies

- Temps
  - Année
  - Trimestre
  - Mois
  - Jour
- Location
  - Continent
  - Pays
  - Région
  - Ville
- Produit
  - Thème
  - Type
  - Référence

5: Donner le détail des opérations pour avoir les cubes

1:

- Roll up de `Temps` vers `Mois`
- Roll up de `Produit` vers `Type`
- Dice
  - Sur `Temps`: Mois = Mars
  - Sur `Location`: Ville = Niort ou La Rochelle
  - Sur `Produit`: Type = Table ou Bureau

2:

- Pivot: `Produit` ↔ `Location`

3:

- Agréation *(somme)* par `Produit`

4:

- Agrégation *(somme)* totale

5:

- Slice de `Location`: Ville = La Rochelle ou Mauzé
- Pivot: `Location` ↔ `Temps`

6:

- Roll up de `Location` à `Pays`
- Slice de `Location`: Pays = `Algérie`
- Drill down de `Location` à `Région`
- Slice de `Location`: Région = `Centre_ALG` ou `Est_ALG`
- Slice ou roll up de `Temps`

7:

- Slice de `Location`: Ville = `Nantes`
- Pivot: `Location` ↔ `Produit`