

Mini-Projet : Entrepôts de Données

AirBNB

Groupe: Leonie Berwanger, Tahani Qattan,
Leila Rekkab, Khady Ajna Thiam

Rendu le: 23 octobre 2018

I Considérations préliminaires

I.1 Quelles sont les actions/opérations importantes pour la plateforme de locations de vacances AirBNB ?

- Réservation d'un logement par un client
- Offre d'un logement par un hôte
- Annulation d'une réservation

I.2 Pour chaque opération : trois traitements possibles permettant d'aider à la prise de décision sur le sujet.

Réservation d'un logement par un client

- Le prix moyen (/nuit) des différents logements réservés par type
- Le prix moyen (/nuit) des différents logements réservés par ville
- Nombre de réservations par ville et par type de logement

Offre d'un logement par un hôte

- La disponibilité des logements par ville
- Le type de logements le plus offert
- Le prix moyen par nuit proposé par type de logement offert

Annulation d'une réservation

- Les raisons d'annulations les plus fréquentes
- Les hôtes les plus exposés aux annulations

I.3 L'ordre d'importance

1. Réservation d'un logement par un client : Permet d'augmenter la rentabilité de la plateforme
2. Offre d'un logement par un Hôte : Permet d'augmenter le nombre de logement offerts et par conséquent l'augmentation du nombre de réservations
3. Annulation d'une réservation par un client : Permet de retirer de l'espace de stockage de la plateforme les logements qui ne sont pas fiables

II Conception de l'entrepôt

II.1 Les deux actions les plus importantes à analyser

1. Réservation d'un logement par un client
2. Offre d'un logement par un hôte

II.2 Réservation d'un logement



FIGURE 1 – Modèle en étoile des réservations

La table des faits

Pour enregistrer les réservations, on choisit des faits transactionnels, parce qu'on s'intéresse aux caractéristiques des réservations spécifique et il ne sert à rien de les regrouper à l'avance. La table des faits suit le modèle relationnel suivant :

Reservations(
idLogement ChaîneCar,

idLocalisation ChaîneCar,
idHote ChaîneCar,
idClient ChaîneCar,
idPeriode ChaîneCar,
nbVoyageurs Integer,
prixReservation Nombre(10,2))

Les mesures

La table des faits contient les mesures *nbVoyageurs* et *prixReservation*. Ces mesures sont additives, parce qu'elles sont exprimées en nombres et liées à une certaine transaction.

II.3 Offre d'un logement par un hôte

Pour les logements offerts on crée une table de snapshots (une période raisonnable serait 1 jour). À la fin d'une période fixe on enregistre le nombre de logements par localisation et type de logement.

La table des faits

Un fait (**idLoc**, **typeLog**, **idDispo**, **idCap**, **j**, **x**) existe pour chaque combinaison de localisation idLoc, type de logement typeLog, disponibilité idDispo, capacité maximale de voyageurs cap et jour j. La mesure x représente le nombre de logements du type typeLog, au lieu idLoc, disponibles/indisponibles, pouvant recevoir cap voyageurs le jour j.

Les mesures

La mesure x est semi-additive, parce qu'on peut p.e. prendre la somme de logements dans toutes les localisations et de tous les types, mais on ne peut pas additionner à travers de la dimension du temps.

II.4 Les dimensions

Les tables des dimensions suivent le modèle relationnel suivant :

Logement(rf.II.2, 12 attributs)

Localisation(rf.II.2, 9 attributs)

Client(9 attributs)

Hôte(9 attributs)

Periode(6 attributs)

Abstract Type; TypeLogement(typeName ChaîneCar, MotsClés Array<ChaîneCar>,

termesLégaux Array<ChaineCar>, dateRajout Date) dans une table d'objets
Disponibilité(idDisponibilite ChaineCar, nomDisponibilite ChaineCar, limité Integer[0,1],
dernierChangement Integer[0,..,2])
Capacité(idCap, description ChaineCar, nbMinVoyageurs Integer, nbMaxVoyageurs Integer)

II.5 Répondre aux traitements

Le schéma défini permet de répondre aux traitements.

Réservation d'un logement par un client

- Un boucle imbriqué qui calcule le prix par nuit pour les réservations,après un GROUP BY typeLog est l'application de la mesure AVG()
- idem, GROUP BY ville
- COUNT(*),GROUP BY typeLog, ville

Offred'un logement par un hôte

-
- SELECT (typeLog,x) FROM FactTableOffres;
-

II.6 Un exemple d'instance

```
CREATE TABLE Localisation (
    idLocalisation VARCHAR2(10) PRIMARY KEY,
    Rue VARCHAR2(100),
    Region VARCHAR2(100),
    Ville VARCHAR2(100),
    codePostal VARCHAR2(20),
    pays VARCHAR2(100),
    latitude NUMBER(10,7),
    CONSTRAINT lat CHECK(latitude BETWEEN 0.0 AND 90.0),
    longitude NUMBER(10,7),
    CONSTRAINT long CHECK(longitude BETWEEN -180.0 AND 180.0),
    codePays VARCHAR2(100)
);
```

```
INSERT INTO Localisation
VALUES( '1', '26_Avenue_Emile_Diacon',
'Occitanie', 'Montpellier', '34090', 'France',
43.6286172, 3.8609373, 'FRA');
```

```

INSERT INTO Localisation
VALUES( '1', '280_Broadway',
       'New_York', 'New_York_City', '10007', 'United_States_of_America',
       40.7143528, -74.0059731, 'USA' );

```

```

CREATE TYPE typeLog_T AS OBJECT (
    typeName VARCHAR2(50),
    ...
);
/

```

```

CREATE TABLE typesLogement OF typeLog_T;

```

```

CREATE TABLE Logement (
    idLogement VARCHAR2(10) PRIMARY KEY,
    typeLog typeLog_T,
    nbChambres INTEGER,
    nbVoyageurs INTEGER,
    chezHabitant VARCHAR2(1),
    CONSTRAINT hab CHECK (chezHabitant IN ('0', '1')),
    nbLits INTEGER,
    ascenseur VARCHAR2(1),
    CONSTRAINT ascens CHECK (ascenseur IN ('0', '1')),
    parking VARCHAR2(1),
    CONSTRAINT park CHECK (parking IN ('0', '1')),
    television VARCHAR2(1),
    CONSTRAINT tele CHECK (television IN ('0', '1')),
    cuisine VARCHAR2(1),
    CONSTRAINT cui CHECK (cuisine IN ('0', '1')),
    wifi VARCHAR2(1),
    CONSTRAINT wf CHECK (wifi IN ('0', '1'))
);

```

```

CREATE TYPE Date_T AS OBJECT (
    jour INTEGER CHECK (jour BETWEEN 1 AND 31),
    mois INTEGER CHECK (mois BETWEEN 1 AND 12),
    an INTEGER CHECK (an BETWEEN 1905 AND 2018)
);
/

```

```

CREATE TYPE Periode_T AS OBJECT (
    idPeriode VARCHAR2(10),
    DateDeb Date_T,
    DateFin Date_T,
    HeureEntree INTEGER CHECK (HeureEntree BETWEEN 0 AND 23),
    HeureSorte INTEGER CHECK (HeureSorte BETWEEN 0 AND 23),

```

```

        Saison VARCHAR2(20)
    );
/

CREATE TABLE Periode OF Periode_T (
    CONSTRAINT pk PRIMARY KEY(idPeriode));

CREATE TABLE Client (
    idClient VARCHAR2(10) PRIMARY KEY,
    nomC VARCHAR2(50),
    prenomC VARCHAR2(50),
    sexe VARCHAR2(1) CHECK (sexe IN ('0','1','2')),
    dateNaiss Date_T,
    email VARCHAR2(100),
    langueParlee VARCHAR2(50),
    dateInscription Date_T,
    situationFamiliiale VARCHAR2(50)
);

CREATE TABLE Hote (
    idHote VARCHAR2(10) PRIMARY KEY,
    nomH VARCHAR2(50),
    prenomH VARCHAR2(50),
    sexe VARCHAR2(1) CHECK (sexe IN ('0','1','2')),
    dateNaiss Date_T,
    email VARCHAR2(100),
    langueParlee VARCHAR2(50),
    dateInscription Date_T,
    situationFamiliiale VARCHAR2(50)
);

CREATE TABLE Reservation (
    idLogement VARCHAR2(10),
    idLocalisation VARCHAR2(10),
    idHote VARCHAR2(10),
    idClient VARCHAR2(10),
    idPeriode VARCHAR2(10),
    nbVoyageurs INTEGER,
    prixReservation NUMBER(10,2)
);

```