TP2 Entrepôts de Données

Master 1 MIAGE

Conception et alimentation d'un entrepôt de données

Etat de l'existant: Une BD transactionnelle

Soit un extrait d'une base de données transactionnelle (SIO) servant à la gestion de facture dans le cadre de la vente de produits alimentaires pour une enseigne donnée. Les tables qui nous intéressent sont celles-là :

Client (Num, Nom, Prenom, Adresse, Date_nais, Sexe)
Produit (Num, Designation, Stock)
Prix_date (Num, Produit=>Produit, date, prix, remise)
Facture (Num, Client=>Client, date_etabli)
Ligne_facture (Facture=>Facture, Produit=>Produit, Qte, Id_prix=>Prix_date)

Il est à noter que pour chacune des tables ci-dessus, la clé primaire est soulignée et les clés étrangères sont indiquées par des flèches de référencement. Ces différentes tables sont créées par le script "script.sql" que vous trouverez sur Moodle.

Etat des besoins

La première étape pour lancer un projet décisionnel est de faire l'inventaire des questions pour lesquelles les décideurs ont besoin d'avoir des réponses. En effet, il est fréquent qu'un projet décisionnel avec un périmètre très restreint, par exemple, uniquement pour la force de ventes, fonctionne bien dans un premier temps. Mais ses extensions n'ayant pas été prévues au départ, plusieurs choix techniques bons pour un petit projet, s'avèrent être aujourd'hui des handicapes majeurs à l'évolution du système. De plus, en moyenne, le volume des données double tous les deux ans. On parle donc souvent de problème de «scalability » ou passage à l'échelle, soit la capacité d'une plateforme décisionnelle de monter en puissance. Une bonne phrase pour synthétiser la méthodologie d'un projet décisionnel est : « voir grand, mais commencer petit ».

Dans le cadre de ce projet, on cherche à analyser les ventes effectuées afin de les faire croître. Les caractéristiques intéressantes des ventes sont les prix et les quantités. On s'intéresse à des critères géographiques (où sont les clients qui achètent) afin de cibler des campagnes promotionnelles. La précision des analyses est variable, mais on notera que l'entreprise a à la fois une vocation locale (échelle de la ville) et internationale (échelle du pays). On s'intéresse à des critères temporels (quand se passent les achats) afin d'aider à l'organisation de l'entreprise (période de vacances, embauches saisonnières supplémentaires, etc.). La précision de ces critères est peu exprimée, on cherchera à adopter des critères classiques.

Voici les principales questions auxquels nous essayons de répondre, sachant que dans le cas d'un Data Warehouse, l'enjeu est de proposer des vues sur les données qui dépassent les simples questions formulées, pour faire apparaître des relations non encore envisagées par les utilisateurs

- Quels sont les produits les plus vendus, selon leurs désignations et catégories ?
- On s'intéresse aux profils des clients : quels sont leurs achats en fonction de leur âge, de leur groupe d'âge et de leur sexe

- Quels sont les chiffres d'affaire en fonction des jours, semaines et année (Quel est le chiffre d'affaire du jour 254, de la semaine 42 et de l'année 2003) ?
- Est ce qu'il existe une relation entre les chiffres d'affaire, les mois de l'année et les sexes des acheteurs (par exemple est-ce que les femmes achètent plus en novembre) ?
- Y a-t-il une relation entre le temps, l'espace et la vente de produit ?
- Quels sont les trois produits les plus vendus en général, et par catégorie ?
- Combien se classerait dans le top des ventes toutes catégories confondues un produit vendu à 50 exemplaires ?
- Quels sont les produits qui contribuent à moins de 0.05% du CA pour un pays ou pour une année donné(e)?
- Est-ce que ces produits peuvent bénéficier d'une remise ?
- Quels sont les produits qui sont achetés ensemble ? Par exemple afin de les rendre plus proches sur le site de vente?
- Quelle est la tendance des ventes pour l'année à venir?
- Est-ce que les remises font augmenter les ventes?
- etc.

Remarque : Nous avons ci-dessus une liste de questions non structurées, mal posées, pas claires. Il serait un comble de les considérer comme un cahier des charges.

La deuxième étape de notre projet décisionnel est d'identifier les sources d'informations, internes et externes à l'entreprise, permettant de construire le Data Warehouse qui répondra aux questions précédemment soulevées. Nous avons choisi ici un cas simple, où le SIO est suffisant.

Modélisation dimensionnelle

Question de modélisation

Proposez un modèle multidimensionnel (schéma en étoile avec une table de fait et des tables de dimensions) de Data Warehouse capable de répondre aux besoins des utilisateurs.
Représentez, sur un diagramme conceptuel, les données de l'entrepôt sous la forme du schéma en étoile proposé. Enoncez les hiérarchies pertinentes pour chaque dimension.

Implémentation et alimentation

Une fois la conception de votre entrepôt de données terminée, vous allez travailler maintenant sur son implémentation, et l'intégration des données.
Pour ce TP, l'entrepôt de données (fait et dimensions) sera vu comme un ensemble de **vues matérialisées** provenant d'un ensemble de sources sous-jacentes. De ce fait, les changements au niveau des sources doivent être répercutés périodiquement au niveau de l'entrepôt. La maintenance de l'entrepôt entraîne donc la consultation des sources sous-jacentes. La périodicité de mise à jour des vues matérialisées est fonction des besoins des données sur le serveur dédié à l'analyse. Le processus ETL qui permet l'intégration des données au sein de l'entrepôt sera donc réduit à la maintenance des vues matérialisées.

Les éléments suivants peuvent vous aider à la définition des vues matérialisées :

- La désignation d'un produit comporte son nom, sa catégorie suivie de sa sous-catégorie séparées les 3 par des ".". Si la sous-catégorie est absente, il faut la mettre à NULL dans l'entrepôt.
- L'adresse d'un client comporte son pays, son code postal et sa ville. Les deux premières lettres du code postal donnent le département.

Quelques rappels SQL :

- La fonction substr(X, A, B) renvoie les B caractères à partir du caractère A dans la chaîne X.
- La fonction instr(chaîne, sous-chaîne [,début [,nombre occurrences]]) recherche la position d'une sous-chaîne dans une chaîne.
- La fonction regexp_substr(X,P,pos,n) cherche la nième occurrence de l'expression régulière P dans la chaîne X, à partir de la position pos.
- Pour la gestion des dates, utilisez les fonctions extract, to_date et to_number.
- Month_between donne le nombre de mois entre 2 dates.
- La requête suivante permet de créer une liste de dates entre date_a et date_b

```
select level + date_a- 1 as date
from dual
connect by level < (date_b – date_a + 2)
```

- Pour le calcul du groupe d'âge pour chaque client utilisez le format suivant : Si l'âge est inférieur à 30, il écrit la chaine de caractère « <30 ans ». Si l'âge est compris entre 30 et 45, il écrit la chaine de caractère « 30-45 ans ». Si l'âge est compris entre 45 et 60, il écrit la chaine de caractère « 46-60 ans ». Sinon, il écrit la chaine de caractère « >60 ans ». La structure de contrôle **case when…then…end** dans le select de votre requête peut faire l'affaire.

Questions d'implémentation

1. Ecrivez les requêtes SQL de création de vues matérialisées qui permettent le transfert et la transformation des données depuis la base transactionnelle vers le Data Warehouse.
2. Créez les différentes clés primaires et étrangères pour les différentes vues.
3. Oracle permet de déclarer les dimensions, ainsi que leurs hiérarchies. Les dimensions sont optionnelles mais hautement recommandées du fait qu'elles constituent des informations sur les hiérarchies qui permettront ultérieurement au moteur d'Oracle d'optimiser les requêtes dimensionnelles (de type rollup par exemple).
   Pour la syntaxe, voici un exemple de création de dimension (extrait de la doc Oracle):

```
CREATE DIMENSION products_dim
LEVEL product IS(products.prod_id)
LEVEL subcategory IS (products.prod_subcategory) [SKIP WHEN NULL]
LEVEL category    IS (products.prod_category)
HIERARCHY prod_rollup (
  Product CHILD OF subcategory
    CHILD OF category)
ATTRIBUTE product DETERMINES (products.prod_name, products.prod_desc,
prod_weight_class, prod_unit_of_measure, prod_pack_size, prod_status,
prod_list_price, prod_min_price)
ATTRIBUTE subcategory DETERMINES (prod_subcategory, prod_subcategory_desc)
ATTRIBUTE category DETERMINES (prod_category, prod_category_desc);
```

A chaque création de dimension, il faut la valider (compilation) et regarder éventuellement les erreurs. Voici un exemple pour une dimension nommée « `products_dim`» :

```
EXECUTE DBMS_DIMENSION.VALIDATE_DIMENSION ('PRODUCTS_DIM', FALSE, TRUE, 'test
dim prod');

On regarde quelles lignes posent problème dans la vue matérialisée
« products » par rapport à la définition de cette dimension :
SELECT * FROM products
WHERE rowid IN (SELECT bad_rowid
```

```
FROM dimension_exceptions
WHERE statement_id = 'test dim prod';
```

Les dimensions n'apparaissent pas dans le navigateur de SQLDeveloper. Pour voir le descriptif d'une dimension, utiliser une autre procédure du paquetage DBMS_DIMENSION :

```
SET SERVEROUTPUT ON ;
EXECUTE DBMS_DIMENSION.DESCRIBE_DIMENSION('products_dim');
```

Questions d'exploitation – requêtes SQL

1. Quel est le chiffre d'affaire par produit ?
2. Quel est le chiffre d'affaire par catégorie et mois, par catégorie, et globalement ?
3. Quel est le chiffre d'affaire par tranche d'âge, en donnant le rang de chaque tranche d'âge (1 pour celle qui a le plus grand chiffre d'affaire) ?
4. Quels sont les 3 produits les plus vendus en quantité ?