EPITA

Bases de données 2ème partie
AppIng2-2015
Session 2014

Alexandra Champavert

Contenu du cours

- Le datawarehouse
- Principes de modélisation (flocon, étoile)
- Les ETL
- Les fonctionnalités propres à Oracle
- Le projet
- Soutenance

Datawarehouse
Introduction

Le datawarehouse – La base

- Datawarehouse = Entrepôt de données
- A opposer avec bases de données relationnelles « classiques »
- Vue multidimensionnelle des données
- Notion d'hyper-cube
- Notion de dimensions

Le datawarehouse - Modélisation

- Deux modélisations courantes possibles :
  - Flocon (modèle français)
  - Etoile (modèle anglo-saxon)

Datawarehouse vs OLTP – 1

- OLTP : OnLine Transaction Processing
  - Modèle éminemment relationnel
  - Reconstitution de l'information en posant les relations entre tables
  - Forte dépendance aux index
  - Calcul des relations lourd dans certains cas
  - Tout est fait pour pouvoir sauvegarder/restaurer rapidement la base de données

Datawarehouse vs OLTP – 2

- Le datawarehouse contient :
  - Une ou des tables de faits
  - Des tables de dimensions
  - Les relations se font entre dimensions et faits
  - Ce sont les uniques relations
  - La table de faits contient volontairement des doublons contrairement à un modèle OLTP
  - Très peu d'index
  - Aucun trigger de préférence
  - La base de données est extrêmement rarement faite pour être sauvegardée

Datawarehouse et systèmes décisionnels
Eléments de base

Tables de faits

- Une table de faits est une table contenant les données en provenance des sources hétérogènes de données, et dont les éventuelles relations ont été calculées afin de présenter des données brutes pré-calculées
- Un fait est une information unitaire, une ligne de cette table

Dimension

- La dimension est un ensemble de tables en relation les unes avec les autres, représentant un axe d'agrégation des données

Construction des agrégats – 1

- Les agrégats sont présentés dans des tables de résultats
- Ces tables de résultats contiennent les colonnes résultant des relations entre les tables de faits et les tables de dimension

Construction des agrégats – 2

- Les agrégats peuvent se construire les uns par rapport aux autres en cascade
- On peut parler de cascade d'agrégats
- Il s'agit d'un graphe de construction des agrégats entre eux

Datawarehouse Construction

Constituer un datawarehouse – 1
Sources de données

- On part généralement de données hétérogènes :
  - Extractions de fichiers plats provenant de sources diverses : applications de gestion, application de logistique…
  - Bases de données de tous éditeurs
  - Fichiers de tableurs
  - Documents papier (Bien sûr !!)
  - …

Constituer un datawarehouse – 2
Que fait-on des sources de données ?

- Obligation de reformater les données hétérogènes
- Se fonder sur le moteur de bases de données cible
- Prévoir un Staging Area

Constituer un datawarehouse – 3 Staging Area

- La Staging Area est :
  - Une base de données intermédiaire
  - Contient des tables et des ensembles de tables permettant l'importation des données hétérogènes
  - Contient des tables de travail
  - Assure l'interface avec le datawarehouse après traitement
  - Contient l'ensemble des procédures de retraitement des données brutes

Constituer un datawarehouse – 4 Chargement de la Staging Area

- Les procédures de chargement de la Staging Area sont diverses et diversement situées :
  - Serveur d'application permettant la mise en forme des données pour importation
  - Procédures internes à la base de données de la Staging Area de mise en forme des données une fois importées
  - Pseudo-tables liées à des fichiers externes permettant de présenter les données en interne pour retraitement

Constituer un datawarehouse – 5 Travail sur la Staging Area

- Une fois les tables chargées et remises en forme :
  - Processus de préparation pour chargement dans le datawarehouse incluant :
  - Vérification de la qualité des données
  - Préparation de vues intégrant le doublonnage volontaire des données
  - Création/Mises à jour des dimensions

Constituer un datawarehouse – 6 Chargement du datawarehouse

- Cycle de chargement :
  - Vidange du datawarehouse pour chargement
  - Destruction des index des tables du datawarehouse
  - Chargement du datawarehouse à partir de la Staging Area
  - Vérification de toutes les manières possibles, de la qualité du chargement
  - Création des index précédemment détruits
  - Traitement des données du datawarehouse pour préparation des données de restitution

Constituer un datawarehouse – 7 Préparation

- Préparation de la restitution :
  - Création de tables d'agrégats intermédiaires
  - Alimentations des tables d'agrégats
  - Flux d'alimentation donnant l'ordre d'alimentation des tables d'agrégats
  - Recalcul des agrégats pour présentation des données
  - Validation des agrégats pour utilisation dans les requêtes

Constituer un datawarehouse – 8 Restitution

- Restitution :
  - Au travers des agrégats recalculés
  - Au travers de vues sur les agrégats

Les éléments principaux du datawarehouse

Données hétérogènes

- Elles proviennent de différentes sources :
  - Mainframe
  - Logiciels de logistique
  - Logiciels comptables
  - Bases de données
- Elles sont extraites sous forme de
  - fichiers plats
  - SQL
- Ou intégrées en connexion directe avec la base de données

Staging Area

- C'est une base de données intermédiaire qui récupère les données brutes
- Elle transforme ces données afin de les charger
- On parle souvent de logiciel ETL → Extract, Transform, Load

Datawarehouse

- C'est l'endroit où sont stockées les données après leur transformation par la staging area
- Le datawarehouse contient la logique de création des agrégats
- Il permet les interrogations à vocation décisionnelle

Datawarehouse
Les outils Oracle

L'instance de base de données

- La mémoire allouée aux tris est plus conséquente
- La parallélisation des calculs des requêtes est essentielle à un traitement rapide
- Le nombre de process permettant le calcul des tables d'agrégats doit être supérieur à zéro
- La fonction de réécriture des requêtes doit être mise en service
- La mémoire doit être découpée en zones empêchant les tables de faits de provoquer des défauts de cache

La base de données

- La base de données doit être construite selon un modèle très différent d'une utilisation OLTP
- La taille du bloc de données est plus importante
- L'utilisation intensive des tablespaces est cruciale
- Les groupes de tablespaces pour les tablespaces temporaires devient important pour la parallélisation des opérations de tri à grande échelle

Le bloc Oracle

- Il représente l'unité de base de la base de données
- Il est exprimé en puissance de 2 x Ko : 2, 4, 8, 16 ou 32 Ko
- Il contient les lignes des tables, et les entrées des index
- Plus il est grand, plus il peut contenir de lignes de taille importante (cas des systèmes décisionnels)

Les tablespaces

- Dans un système décisionnel, la taille des données traitée dépasse très souvent la centaine de Go, voir affleure parfois la dizaine de To
- Le tablespace est l'unité logique de base de stockage de l'information
- Afin d'accélérer l'accès aux données non en mémoire, il est nécessaire de faciliter la parallélisation
- Les tablespaces permettent de répartir les données sur les stockages

Les tables

- Une table est constituée de segments
- Elle appartient à un seul et unique tablespace
- Elle est généralement indexée afin de rendre son parcours plus rapide

Limite de stockage en table classique

- Certaines tables peuvent avoir des volumes de données provoquant une lenteur d'accès à la donnée, y compris par les index
- Dans le monde décisionnel, il n'est pas rare de voir des tables de plusieurs centaines de Go, voire de plusieurs To
- La table de faits d'un système décisionnel représente souvent 99 % du stockage pris par le système complet

Le partitionnement de table et d'index

- Partitionner une table revient à la découper en unités de taille plus petite
- Chaque partition représente une table
- La table principale devient une méta-table, ou table encadrante
- Le partitionnement permet de stocker chaque unité de table dans un tablespace différent
- Il permet aussi de diminuer la concurrence d'accès à l'objet (table, index)

Partitionnement – Principe

- Une partition est un sous-ensemble d'objet
- Le partitionnement se fait sur une clé constituée d'une ou plusieurs colonnes
- Il est :
  - Sur intervalle
  - Sur ensemble de valeurs
  - Clé hash
- Le partitionnement peut comporter du sous-partitionnement
- Deux niveaux de partitionnement suffisent à classer correctement les données
- Le deuxième niveau de partitionnement est forcément une clé hash

Partitionnement par intervalle

- Le partitionnement par intervalle est utilisé majoritairement pour trier les données selon un axe-temps
- Il permet une historisation rapide des données au fil du temps

Le partitionnement par valeurs

- Lorsque la clé de partitionnement est régie par un code, le code n'est pas suivi
- Le partitionnement par valeurs consiste en une liste de valeurs décrivant chaque partition

Le partitionnement hash

- Lorsque la colonne de partitionnement dispose d'une « mauvaise » répartition des données ne permettant pas un bon équilibre des partitions, le partitionnement hash permet de rééquilibrer la répartition
- Ce type de répartition est utilisé pour la performance essentiellement, et non pour une éventuelle historisation ultérieure

Les index dans le partitionnement

- Un index est soit global, soit local
- Un index global ne suit pas les partitions de la table sous-jacente
- Un index local est indivisible des partitions de la table

Les index locaux

- Un index local s'appuie sur la clé de partitionnement de la table
- Chaque partition de l'index local correspond strictement à la partition de la table qu'il adresse
- L'index local permet une amélioration notable des performances de parcours de la table par index par rapport à un index global

Partitionnement et overflow

- Deux cas majeurs existent dans le stockage des tuples dans une table partitionnée :
  - L'ensemble des partitions permet de stocker toutes les valeurs de clé car celles-ci sont connues et représentent un ensemble fini
  - L'ensemble des valeurs n'est pas connu, il devient nécessaire de créer une partition stockant les valeurs inconnues ne rentrant pas dans les partitions connues

Partitionnement – attachement/ détachement de partitions

- Le partitionnement permet d'attacher/ détacher des tables
- Attachement : La table doit être bijective quant à ses colonnes et ses index « locaux ». La table devient une partition de la méta-table
- Détachement : La partition détachée devient une table à part entière
- Dans tous les cas, les index globaux doivent être recalculés

Dimension

- Une dimension est une table ou un ensemble de tables contenant des données filtrantes pour les tables de faits
- S'il existe plusieurs tables, elles sont liées les unes aux autres et chaque table de niveau supérieur encadre la table du niveau immédiatement inférieur
- Une dimension permet de « regarder » la donnée selon un axe déterminé
- Voir chapitre 10 du Datawarehousing Guide

Vue matérialisée

- Une vue matérialisée est une requête donnant lieu à la création d'une table non modifiable
- Elle peut être indexée
- Elle peut être partitionnée
- Elle est recalculable à chaque mise à jour des tables sous-jacentes
- … ou à la demande
- Voir chapitre 8 et chapitre 9 du Datawarehousing Guide

Cascade de vues matérialisées

- Une utilisation possible des vues matérialisées peut être la constitution d'une ou plusieurs chaînes de vues matérialisées
- Ces chaînes de vues matérialisées sont recalculées dans l'ordre de la chaîne permettant de mettre à disposition un ensemble d'agrégats recalculés les uns par rapport aux autres

Réécriture de requêtes

- La réécriture de requêtes implique plusieurs mécanismes Oracle :
  - Les dimensions
  - Les tables de faits
  - Les vues matérialisées mises à jour
  - Clés primaires
  - Clés étrangères
- La requête est réécrite à la volée par le moteur Oracle pour tenir compte des agrégats précalculés

Les outils Oracle d'interrogation

Windowing

- Il est possible d'extraire les données par fenêtre glissante
- Syntaxe et explication dans la documentation

Ratio To Report

- Connaître le ratio d'une valeur par rapport au total de l'ensemble des valeurs d'une colonne
- Nécessite :
  - La valeur qui peut être un agrégat
  - La somme de toutes les valeurs (utilisation de la fonction OVER())
  - La fonction RATIO_TO_REPORT(la valeur) OVER()
- Voir documentation

Valeur « précédente »/Valeur « suivante »

- Il peut être intéressant de connaître une valeur précédente ou une valeur suivante pour comparaison
- Les fonctions LAG et LEAD pourvoient à cette attente
- Syntaxe :
  - LAG(valeur/agrégat,<rang de la valeur avant>)
  - LEAD(valeur/agrégat,<rang de la valeur après>)

Classements

- Deux fonctions très utiles :
- RANK() : Donne des valeurs non-consécutives en cas d'ex-aequo
- DENSE_RANK() : Donne des valeurs consécutives malgré les ex-aequos

Agréger les données

- De multiple fonctions d'agrégats très puissantes existent directement dans le moteur de base de données
- Syntaxe et explication dans la documentation