

Relational schemas for DWHs

Chapter content:

- Schémas
- Implémentation relationnelle: modélisation dimensionnelle
- Table de faits
- Hiérarchies de dimension
- Attributs de dimension
- Changements dans les dimensions
- Implémentation des dimensions sous Oracle

Table des matières

Relational schemas for DWHs

- Schémas
 - Approches alternatives pour stocker le cube
 - Schéma en étoile et schéma en flocons
 - Autres variantes de schémas
- Implémentation relationnelle: modélisation dimensionnelle
- Table de faits
- Hiérarchies de dimension
- Attributs de dimension
- Changements dans les dimensions
- Implémentation des dimensions sous Oracle

2018-2019

Schémas

Storage

BD Opérationnelle	Entrepôt
schéma en 3 ^e forme normale, pas de redondance	infos redondantes
schéma indépendant des requêtes	modèle multidimensionnel: dimensions et mesures basées sur requêtes utilisateur
données détaillées	données pré-aggrégées
MB,GB	TB,PB

Implémenter le cube

3 approches pour implémenter le modèle multidimensionnel

- **ROLAP:** stocke le cube dans BD relationnelle. Extensions de SQL pour implémenter opérations OLAP
- **MOLAP:** stocke le cube dans structure de donnée spécifique (array)
- **HOLAP:** combine les 2 technologies de stockage

ROLAP

Avantages:

- stocke seulement des cellules non-vides
- tables de dimension petites: sont chargées 'en mémoire'
- bénéficie d'une technologie relationnelle mûre.

Inconvénients:

- coût de la traduction en SQL

MOLAP

Avantages:

- pas besoin de stocker les coordonnées des cellules
- accéder à une cellule particulière est facile (position dans vecteur)
- bonne performance pour requêtes d'agrégation (car stocke aussi les résultats agrégés)
- certaines requêtes analytiques ne sont pas disponibles sous SQL
- permet de détailler finement les options de contrôle d'accès

Inconvénients:

- absence de standard, offre limitée (propriétaire)
- stockage d'agrégats génère surcoût
- capacité de stockage limitée par rapport aux technos relationnelles les plus mûres
- passe mal à l'échelle avec le nombre de dimensions

ROLAP

Objectif: implémenter le cube; dimensions, mesures... dans un système relationnel.

- préserve l'information sémantique (hiérarchies de dimensions)
- faciliter la maintenance
- permettre d'effectuer efficacement les requêtes multidimensionnelles (traduction en SQL, exécution...)
- attentes traditionnelles des SGBD: performance, accès concurrent, sécurité...

ROLAP: caractéristiques

- stocke les données dans des relations
- séparation entre structure and contenu:
 - une relation stocke tous les faits
 - autre relation stocke attribut de dimension
 - la relation des faits stocke les mesures and fait références aux tables de dimension
- ne stocke que les faits existant (pas de 'cellules vides' ici)

Typiquement, schéma adopte une organisation caractéristique:

- Schéma en étoile
- Schéma en flocon
- (galaxy schema)
- starflake schema

Schéma en étoile (Star schema)

Schéma en étoile

Un **Schéma en étoile** est défini par une table des *faits* et un ensemble de tables de *dimension*

- Chaque dimension $(D_i.k_1, \dots, D_i.k_l, \text{Top}_D, \rightarrow)$ est une relation D_i de schéma $(\underline{PK}, k_1, \dots, k_l)$
- La relation des faits a schéma:
 $(FK_1 \rightarrow D_1.PK, \dots, FK_n \rightarrow D_n.PK, M_1, \dots, M_m)$

PK est une clé de D_i . Un attribut par niveau dans le schéma.

La combinaison de toutes les clés étrangères est la clé de la relation des faits. Un attribut par mesure.

Schéma en étoile: propriétés

- relations de dimension généralement en 2NF: redondance
- relations de dimension contient peu de tuples par rapport aux faits: redondance peu coûteuse
- relation des faits en 3NF
- en général, clé artificielle pour la clé de dimension
- requête efficace

Schéma en étoile



Schéma en étoile example

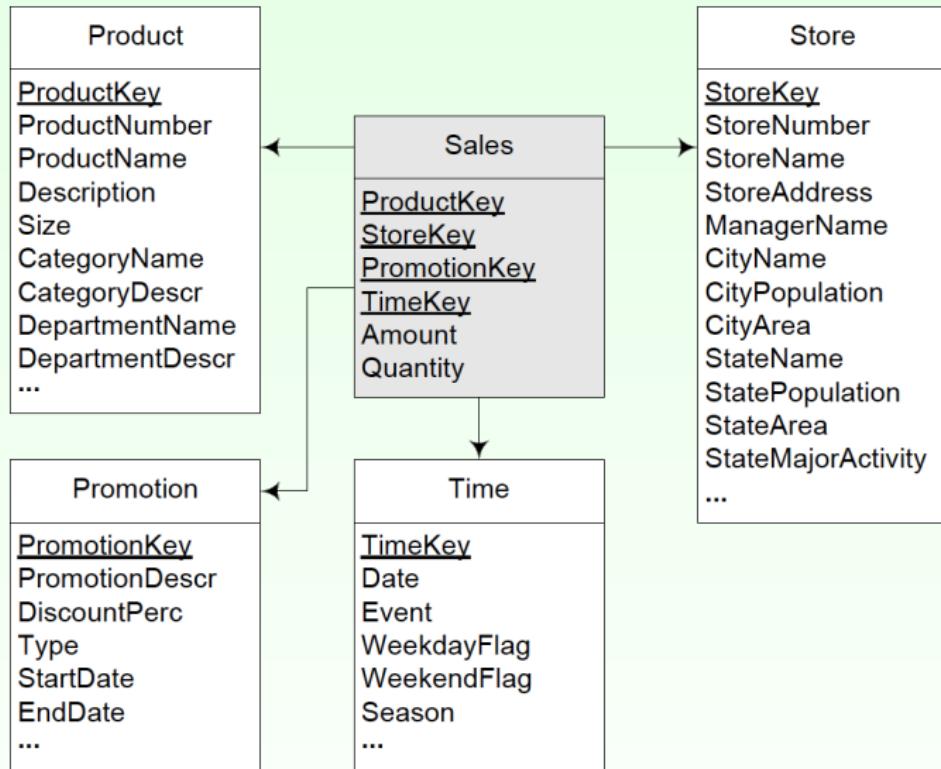


Schéma en flocon (Snowflake schema)

Schéma en flocon

Un **Schéma en flocon** est défini par table de *faits* et un ensemble de tables de *dimension*

- Chaque dimension $(D_i.k_1, \dots, D_i.k_l, \text{Top}_D, \rightarrow)$ est stockée dans l relations D_i^1, \dots, D_i^l , où la relation D_i^j a schéma $(\underline{\text{PK}}, A_1, A_2, \dots, \underline{\text{FK}} \rightarrow D_i^{j+1})^\dagger$
- La relation des faits a schéma:
 $(\underline{\text{FK}}_1 \rightarrow D_1.\text{PK}, \dots, \underline{\text{FK}}_n \rightarrow D_n.\text{PK}, M_1, \dots, M_m)$

PK désigne la clé de D_i . Un attribut par niveau du schéma.

La combinaison des clés étrangères est la clé de la relation des faits. Un attribut par mesure.

[†]évidemment PK, FK et les attributs $A_1, A_2 \dots$ sont distincts en général entre 2 relations

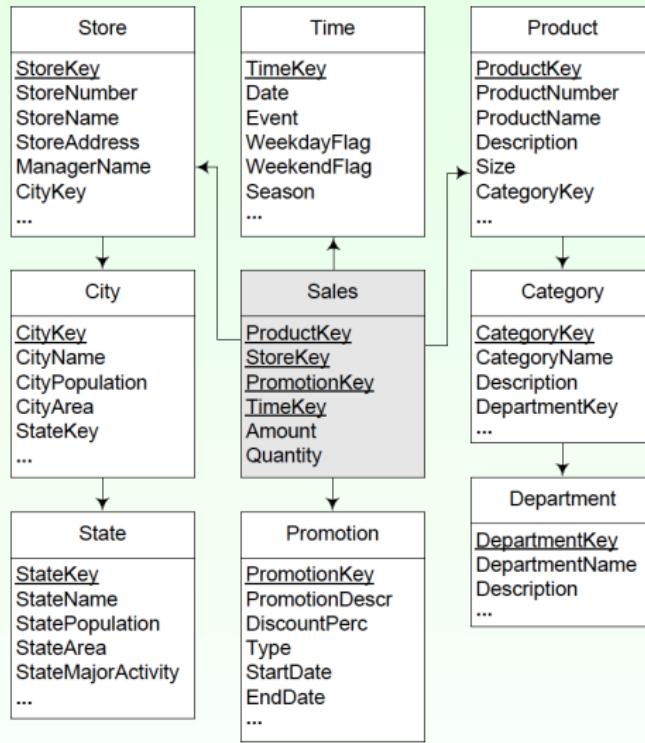
Schéma en flocon: propriétés

- redondance évitées grâce aux tables de dimension normalisées
- tables de dimension petites (peut accélérer certaines requêtes)
- n niveaux de hiérarchies dans une dimension $\implies n$ tables de dimension
- relation des faits en 3ème forme normale
- table des faits fait références aux table de dimension à la granularité la plus fine
- requêtes effectuent de nombreuses jointures: performance dégradée

Schéma en flocon



Schéma en flocon example



[Data Warehouse Course, Vaisman Zimanyi]

Flocons

Avantages:

- Normalisation réduit la taille des tables de dimension
(may improve query time)
- Normalisation facilite maintenance
- représente mieux les hiérarchies

Inconvénients:

- Espace gagné par la normalisation négligeable
- Normalisation rend l'évaluation des requêtes plus difficile (jointures)

⇒ *Faire le choix en fonction de la granularité des requêtes typiques, de la taille des dimensions, fréquence des mises à jour, des matérialisations.*

Dimension déportée (Outrigger dimension)

... (ce n'est pas un type de schéma à part entière!)



Outrigger

Une table qui est jointe à d'autres tables de dimensions (au lieu de la table de fait) dans un schéma en étoile.

À la différence d'une dimension en étoile, on n'a qu'un second niveau, et qui n'est pas forcément en 3NF.

Utilisation typique: table regroupant des attributs utilisés par plusieurs dimensions.

Schéma galaxie (Galaxy schema)

Schéma galaxie

Un **schéma galaxie** est un ensemble de schémas en étoile contenant plusieurs tables de faits qui partagent des dimensions.

!On parle parfois de “constellation de faits”.

Schéma galaxie

Schéma galaxie

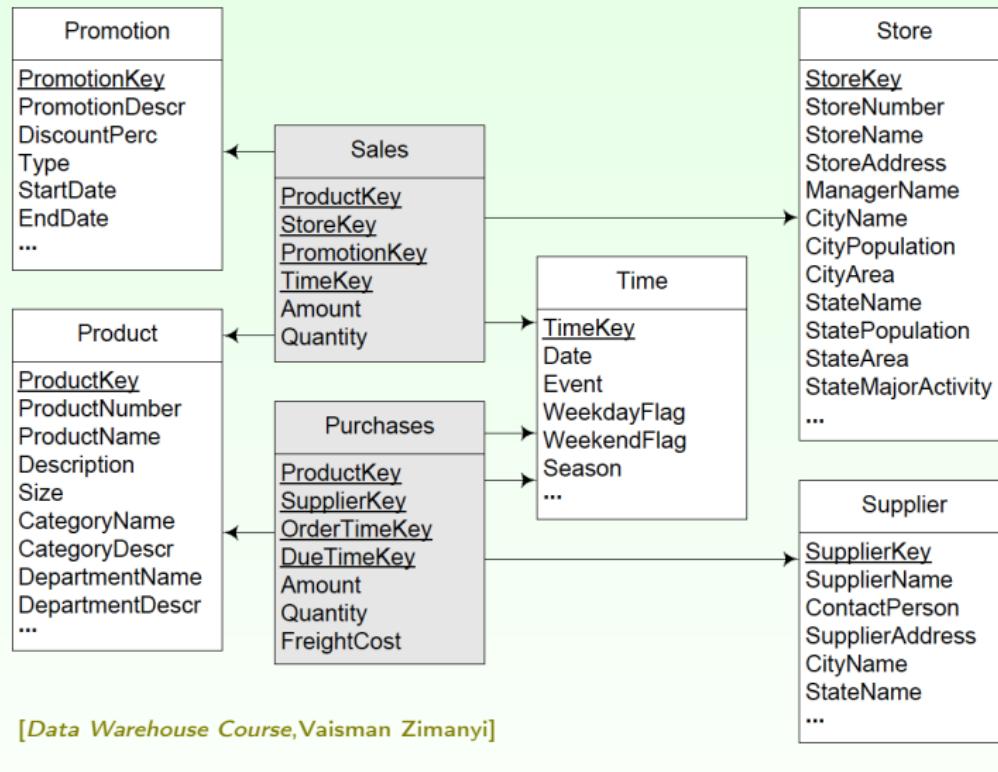


Schéma galaxie: constellation de faits pour optimiser les agrégats

Fact constellation example



Peut permettre d'améliorer les performances sur les agrégats (voire sur les données détaillées).

Hybride flocon-étoile

Schéma hybride flocon-étoile

Un **schéma hybride flocon-étoile** est une combinaison entre un Schéma en étoile et Schéma en flocon: certaines dimensions sont normalisées mais pas toutes.

but: associer les avantages des deux types de schéma.

Hybride flocon-étoile

Exemple de schéma hybride flocon-étoile

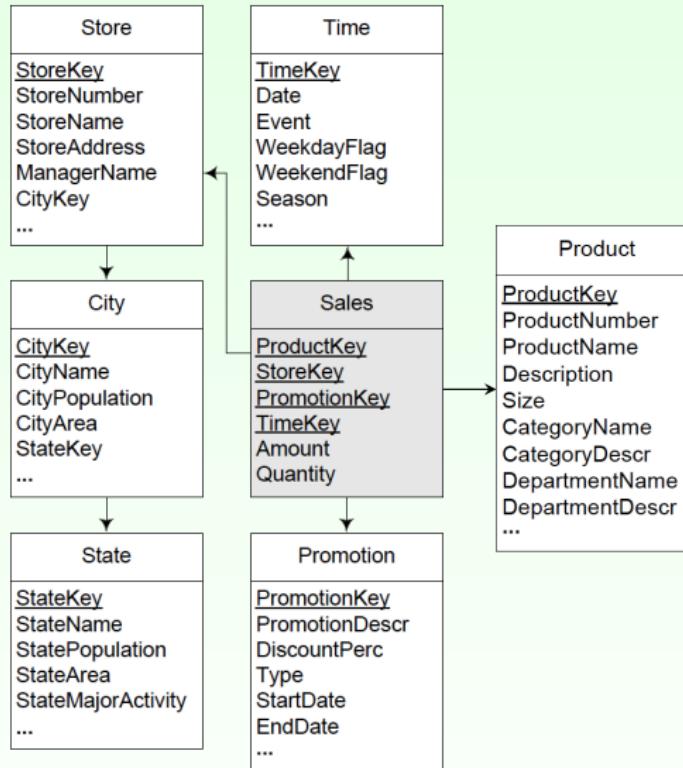


Table des matières

Relational schemas for DWHs

- Schémas
 - **Implémentation relationnelle: modélisation dimensionnelle**
 - Table de faits
 - Hiérarchies de dimension
 - Attributs de dimension
 - Changements dans les dimensions
 - Implémentation des dimensions sous Oracle

2018/2019

Table des matières

Relational schemas for DWHs

- Schémas
- Implémentation relationnelle: modélisation dimensionnelle
- Table de faits**
- Hiérarchies de dimension
- Attributs de dimension
- Changements dans les dimensions
- Implémentation des dimensions sous Oracle

2018/2019

Nulls

[Kimball]

- NULL ok pour les *mesures*
- NULL à éviter pour les clés étrangères de la table de faits.

Pourquoi?

- introduire une ligne par défaut dans les tables de dimension.
- NULL dans les attributs de dimension: préférable de les remplacer par une explication

Les différents types de tables de faits

[Kimball]

- transaction fact tables
- periodic snapshot fact table (instantané périodique)
- accumulating snapshot fact table (instantané cumulé)
- factless fact table
- aggregate fact table
- consolidated fact table

Table des matières

Relational schemas for DWHs

- Schémas
- Implémentation relationnelle: modélisation dimensionnelle
- Table de faits
- Hiérarchies de dimension**
- Attributs de dimension
- Changements dans les dimensions
- Implémentation des dimensions sous Oracle

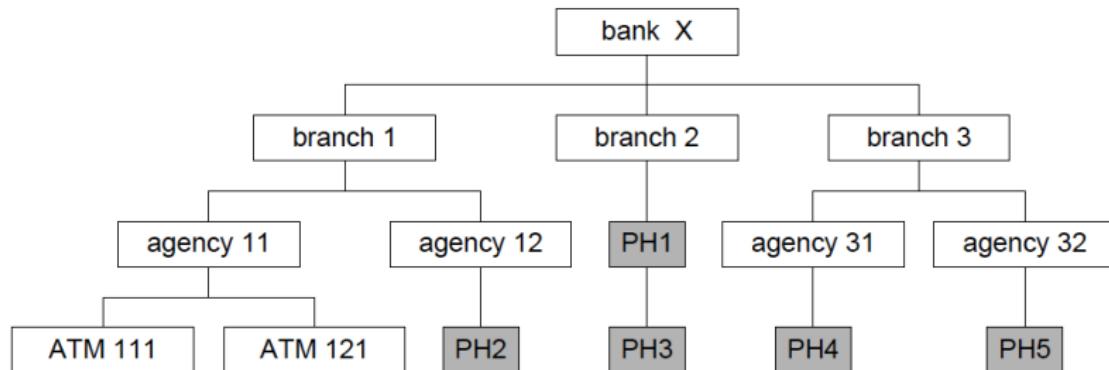
2018/2019

Unbalanced Hierarchies

Unbalanced hierarchy

Still a unique path, but members of some (non-leaf) levels may have no children.

E.g; bank needs not have an ATM, nor branch a bank.



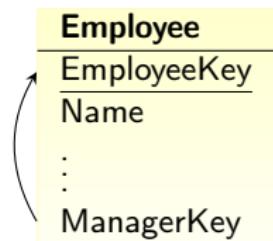
[Data Warehouse Course, Vaisman Zimanyi]

How to build a Schéma en flocon?

→ introduce placeholders/nulls for missing levels.

N.B. like star-schema (NULLs)

Unbalanced hierarchies: recursive hierarchy



Quite challenging!

Recursive hierarchy queries supported by some relationnelle DBMS like Oracle SQL, DB2, PostgresQL or SQL Server (CONNECT BY, CTE) and OLAP languages.

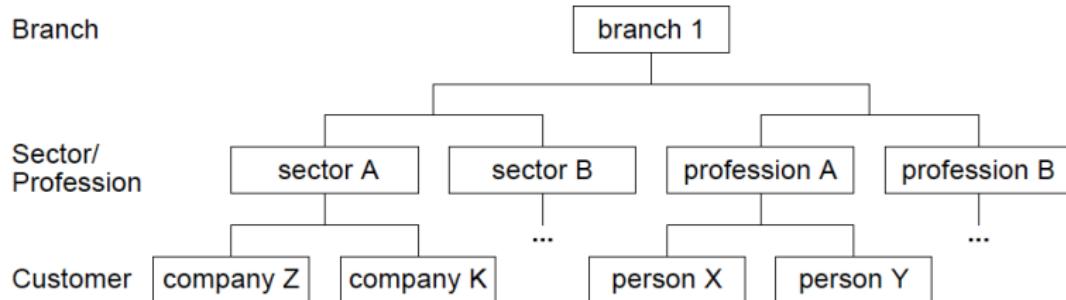
Multiple techniques:

- store transitive closure (descendants) in a bridge table, with distance (possibly make this table a new dimension)
- fix the depth
- record paths to root

Generalized hierarchies

Generalized hierarchy

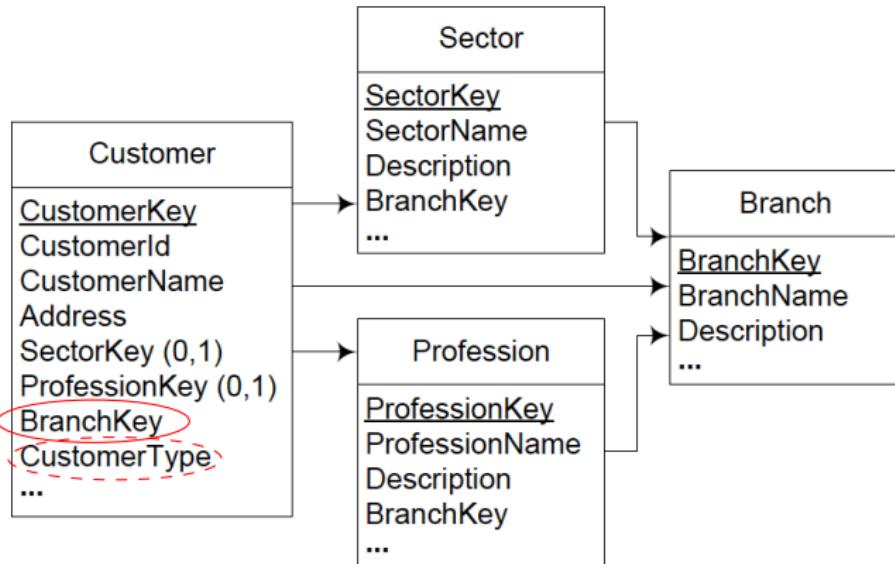
Multiple exclusive path sharing the leaf level at least.



[Data Warehouse Course, Vaisman Zimanyi]

Problem: design a schema allowing roll-up aggregation at common levels?

Generalized Hierarchies



[Data Warehouse Course, Vaisman Zimanyi]

→ additional attribute with next shared level.

Generalized Hierarchies

Other options include:

- Schéma en étoile with NULLs (Not Applicable) for non-applicable values,
- for ragged hierarchy: placeholders or allocation from upper or lower levels (business rules)
- separate fact and tables de dimension for each path

Particular case:

Ragged hierarchy (facts with diverse granularities)

Generalized hierarchy where paths obtained by skipping some levels.

Ex: City→State→Region→Country.

Alternative Hierarchies

Alternative hierarchy

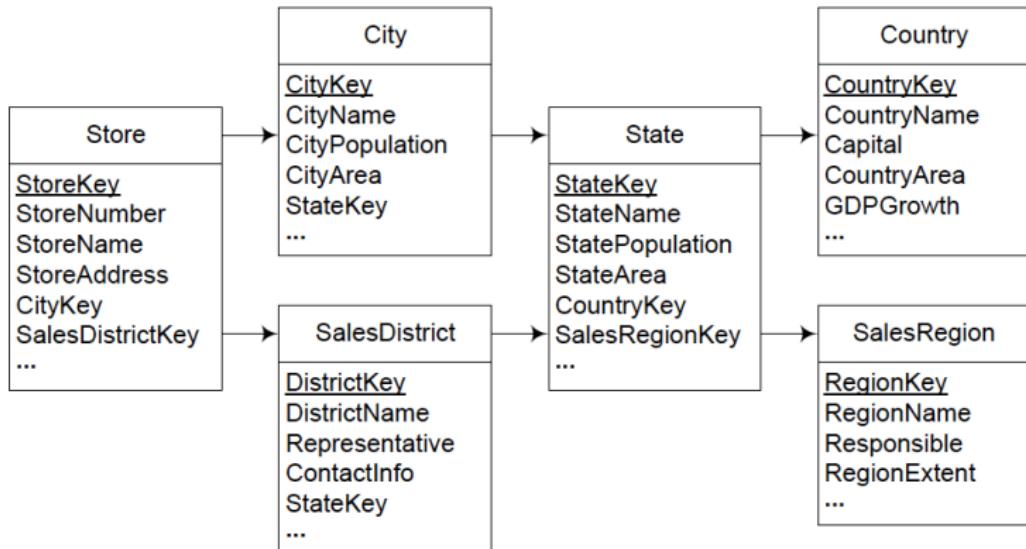
Multiple non-exclusive paths sharing the leaf level at least.



Parallel hierarchies

Parallel hierarchy

Multiple paths corresponding to different analysis criteria

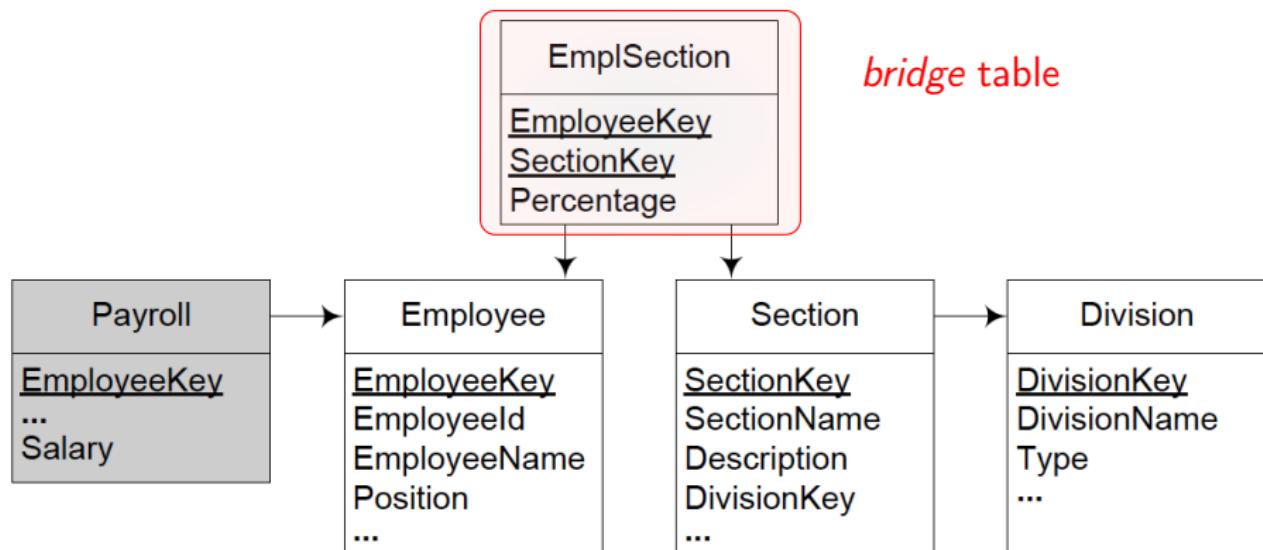


[Data Warehouse Course, Vaisman Zimanyi]

Non-strict hierarchies

Strict hierarchy

Each member has at most one parent along each path.

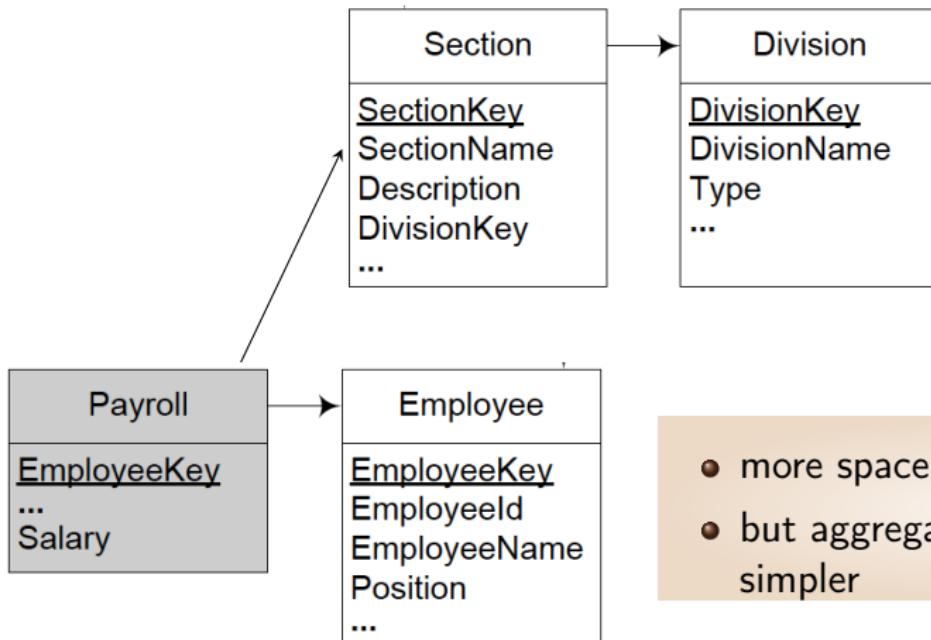


[Data Warehouse Course, Vaisman Zimanyi]

Issue: double counting when aggregation.

Non-strict hierarchies(2)

Alternative solution: transform into strict hierarchy with additional (independent) dimension (requires distribution attribute!).



- more space than bridge table
- but aggregation along hierarchy simpler

Non-strict hierarchies(3)

Other solutions to transform into strict hierarchy:

- create new members by grouping parents of a member. Each child has a single group parent.
- choose one parent, drop all others

Table des matières

Relational schemas for DWHs

- Schémas
- Implémentation relationnelle: modélisation dimensionnelle
- Table de faits
- Hiérarchies de dimension
- Attributs de dimension**
- Changements dans les dimensions
- Implémentation des dimensions sous Oracle

2018/2019

Choix des attributs

Attributs de dimension = critère qu'on va pouvoir utiliser dans les clauses WHERE et GROUP BY des requêtes SQL.

⇒ choix de ces attributs détermine les capacités d'analyse!

Attribute vs measure?

Rule of thumb:

- continuous → measure
- small list → dimension.

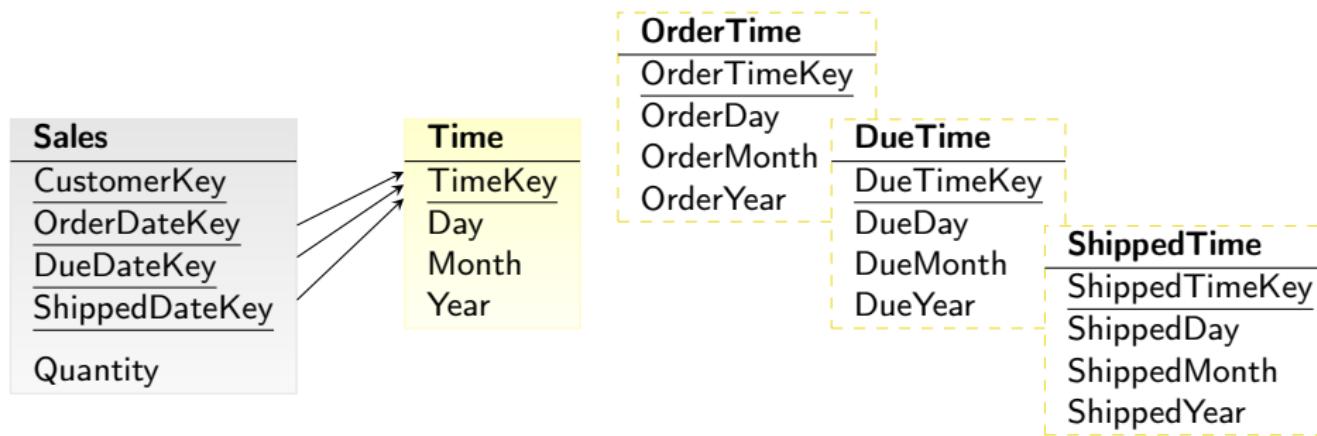
Role-playing dimensions (dimension à jeu de rôle)

Role-playing dimension

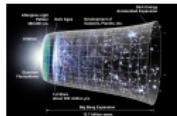
Dimension qui apparaît plusieurs fois dans une même table de fait

- certains outils OLAP ne le permettent pas: en ce cas créer une dimension par rôle.
- SQL aliases (souvent proposés par les outils BI)
- garder une **seule dimension** mais créer **une vue par rôle**.

Réétiqueter les attributs pour permettre distinction entre les vues.



Dimension temporelle



- Pratique courante: définir une dimension temporelle:
 1. certaines informations ne sont pas facilement obtenues avec les fonctions de dates SQL (vacances, périodes fiscales, jours ouvrés)
 2. plus facile de stocker que de calculer à la volée, surtout pour des utilisateurs peu familiers de SQL
- en général *l'instant précis dans la journée* est une mesure dans la table de faits
- si l'on souhaite faire des rollups sur time of day, on peut en faire une dimension à part

Flags/indicator attributes

Keep textual indicators explicit in the tables de dimension (avoid 'Y/N').

Ex:

Date

DateKey	Date	Day of Week	Calendar Month	Quarter	Calendar Year	Holiday Indicator	Weekday Indicator
20150101	01/01/2015	Thursday	January	Q1	2015	Holiday	Weekday
20150102	02/01/2015	Friday	February	Q1	2015	non-Holiday	Weekday
:	:	:	:	:	:	:	:

If many low cardinality indicators, store them all in a single *junk dimension* (dimension fourre-tout■ ■).

Clés

Clés Primaires des dimensions devraient être des *clés artificielles*

- indépendance des sources de données/intégration
- performance
- permet de suivre les changements effectués dans les dimensions



Table des matières

Relational schemas for DWHs

- Schémas
- Implémentation relationnelle: modélisation dimensionnelle
- Table de faits
- Hiérarchies de dimension
- Attributs de dimension
- **Changements dans les dimensions**
- Implémentation des dimensions sous Oracle

2018/2019

Dimensions à évolution lente (Slowly changing dimension)

En pratique, les dimensions sont parfois modifiées (instance et schéma).

- en cas d'erreur \implies il faut remplacer les données (au niveau de l'instance)
- le contexte peut changer

Mais on considère que ces changements restent occasionnels.

Slowly changing dimension

dimension conçue pour pouvoir subir des changements

 on ne parle évidemment pas ici de l'insertion de nouveaux faits (pas d'impact sur le schéma)

Plusieurs approches possibles. La première est “scd type 0”:

- interdit toute mise à jour des valeurs des attributs
- permet l'insertion de nouvelles valeurs
- utilisée (rarement) pour des attributs persistents; dépôt de crédit initial, clé durable, date de création.

SCD : exemple illustratif

Sale

Date Key	Customer Key	Product Key	Sales Amount
t1	c1	1	100
t2	c2	1	100
t3	c3	3	100
t4	c4	4	100

Dimension change:

- nouveaux produits
- modifications des infos (erreur)
- mise à jour de catégorie (évolution)

Product

Product Key	Product Name	Category
1	Cinnamon Roll	Bread
2	Baked Whole Wheat Bread	Bread
3	Treacle Tart 320 g	Desserts
4	Pecan Butter 64 fl oz	Desserts

À partir de l'instant t4, la catégorie de *Cinnamon Roll* est changé à *Dessert*.

SCD (1)

Attribut Type 1: écraser la valeur (overwrite)

- permet de mettre à jour la valeur de l'attribut (ancien tuple est remplacé).
- utilisé pour corriger les valeurs erronées.

Product

Product Key	Product Name	Category
1	Cinnamon Roll	Dessert
2	Baked Whole Wheat Bread	Bread
3	Treacle Tart 320 g	Desserts
4	Pecan Butter 64 fl oz	Desserts

SCD (2)

Dimension Type 2: versionnement (par lignes)

- garde trace des anciens tuples en associant un intervalle de validité à chaque version (ou un n° de version)
- quand la valeur d'attribut change, insère nouveau tuple avec même clé naturelle (mais nouvelle clé artificielle)
- nombre d'anciens tuples sauvegardés pas limité
- table des faits non modifiée (les faits anciens continuent à faire référence aux anciens tuples)

Product

Product Key	Product Name	Category	Row Effective Date	Row Expiration Date	Current Row Indicator
1	Cinnamon Roll	Bread	2010-01-01	2015-08-20	Expired
5	Cinnamon Roll	Dessert	2015-08-21	9999-12-31	Current
2	Baked Whole Wheat Bread	Bread	2010-01-01	9999-12-31	Current
3	Treacle Tart 320 g	Desserts	2010-01-01	9999-12-31	Current
4	Pecan Butter 64 fl oz	Desserts	2010-01-01	9999-12-31	Current

SCD (3)

Dimension Type 3: versionnement (par attribut)

- met en parallèle l'ancienne et la nouvelle valeur d'un attribut avec une (des) colonne(s) additionnelle(s)
- Remplace la valeur de l'attribut, sauvegarde l'ancienne valeur dans la colonne "valeur précédente" column
- historique limité

Product

Product Key	Product Name	Category	Prior Category
1	Cinnamon Roll	Dessert	Bread
2	Baked Whole Wheat Bread	Bread	NULL
3	Treacle Tart 320 g	Desserts	NULL
4	Pecan Butter 64 fl oz	Desserts	NULL

SCD (4)

Dimension Type 4: mini-dimension

- stocker les attributs les plus souvent modifiés (ou interrogés) dans une mini-dimension
- un tuple par combinaison de valeurs apparaissant dans les données = "profils"
- permet de traiter les grandes dimensions (*monster dimensions*)
- quand attribut mis à jour: insère nouveau profil (si nécessaire), les faits ultérieurs utilisent le nouveau profil.

Mini-dimension

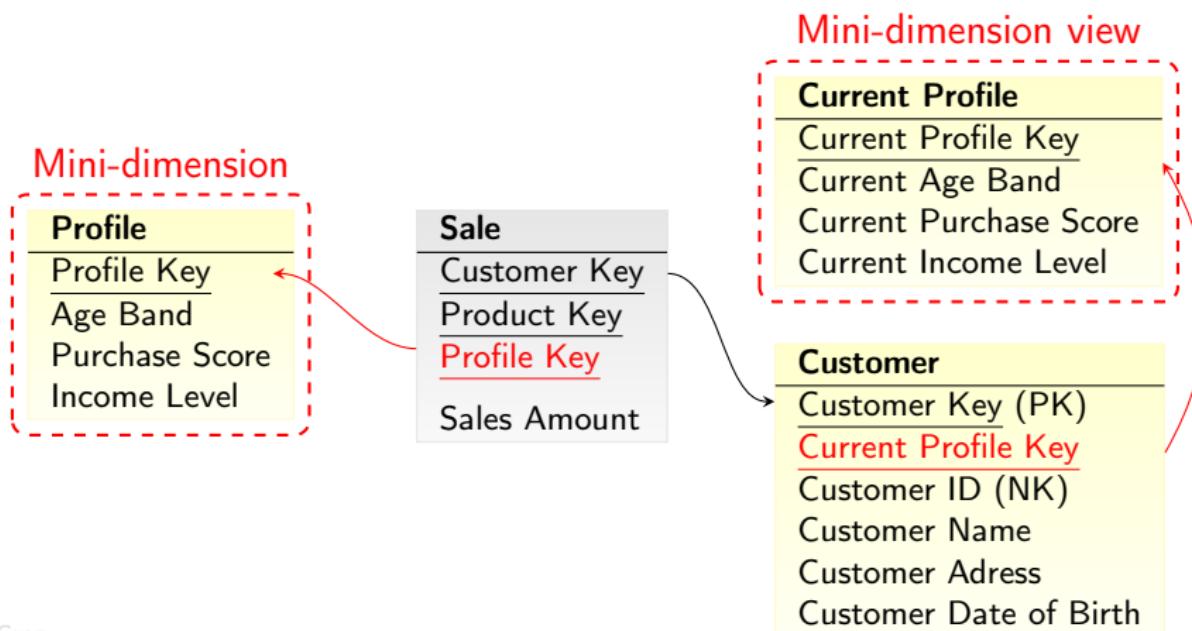


⚠ Other type 4 definition in literature: keep current data in the dimension, store separate *history table* as in type 2.

SCD (5)

Dimension Type 5: 4 + 1

- Type4 avec une clé étrangère depuis la table de dimension vers le profil courant dans la mini-dimension.
- La clé étrangère est un attribut type 1.



SCD (6)

Dimension Type 6: 1 + 2 + 3

- pareil que type 2 mais maintient la valeur courante de l'attribut dans une colonne à part
- ce nouvel attribut est de type 1, et permet de comparer passé et présent comme type 3

Product

Product Key	Product Name	Historic Category	Current Category	Row Effective Date	Row Expiration Date	Current Row Indicator
1	Cinnamon Roll	Bread	Dessert	2010-01-01	2015-08-20	Expired
5	Cinnamon Roll	Dessert	Dessert	2015-08-21	9999-12-31	Current
2	Baked Whole Wheat Bread	Bread	Bread	2010-01-01	9999-12-31	Current
3	Treacle Tart 320 g	Desserts	Desserts	2010-01-01	9999-12-31	Current
4	Pecan Butter 64 fl oz	Desserts	Desserts	2010-01-01	9999-12-31	Current

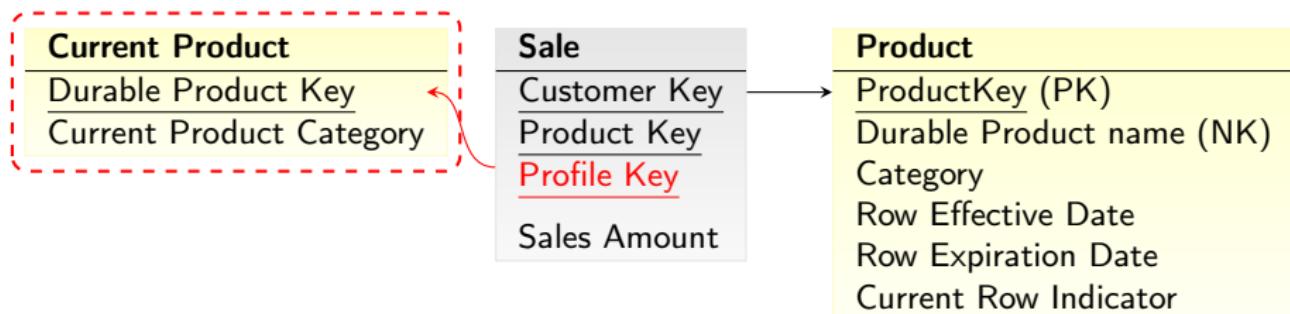
SCD (7)

Objectif: avoir un Type 6 sur plusieurs attributs sans ajouter de multiple colonnes.

- Solution Type 2 mais garde en plus la cleee naturelle vers les attributs dans les Faits
- la clé naturelle doit être durable
- pour accéder aux données courantes, joindre sur la clé durable et sélectionner la ligne avec la valeur “current”
- pour accéder aux données type 2, joindre sur la clé artificielle.

⇒ pour les utilisateurs BI: 2 vues distinctes sur les données.

View of Product...



SCD: Résumé

Type	Impact sur dimension	Impact sur le fait associé
Type 0	pas de changement de l'attribut	valeur originale
Type 1	attribut réécrit	valeur courante
Type 2	ajoute nouveau tuple avec nouvelle valeur de l'attribut	valeur en cours quand le fait a lieu
Type 3	seconde colonne sauvegarde la valeur précédente	valeur courante et précédente
Type 4	mini-dimension contient les attributs changeant souvent	valeur (de l' attr.) au moment du fait
Type 5	mini-dimension Type 4 référencée par clé étr. Type 1 dans la dimension	
Type 6	Dimension Type 2 avec attribut Type 1 additionnel: tuple antérieur remplacé	
Type 7	Dimension Type 2 avec vue limitée aux valeurs courantes	

[Kimball]

SCD pour les flocons

Même idée.

Product

Product Key	Product Name	Category Key
1	Cinnamon Roll	1
2	Baked Whole Wheat Bread	1
3	Treacle Tart 320 g	2
4	Pecan Butter 64 fl oz	2

Category

Category Key	Category Name	Descr.
1	Bread	desc 1
2	Desserts	desc 2

Le 2015-08-21, la catégorie de *Cinnamon Roll* est changée à *Dessert*:

Product

Product Key	Product Name	Category Key	Row Effective Date	Row Expiration Date	Current Row Indicator
1	Cinnamon Roll	1	2010-01-01	2015-08-20	Expired
5	Cinnamon Roll	2	2015-08-21	9999-12-31	Current
2	Baked Whole Wheat Bread	1	2010-01-01	9999-12-31	Current
3	Treacle Tart 320 g	2	2010-01-01	9999-12-31	Current
4	Pecan Butter 64 fl oz	2	2010-01-01	9999-12-31	Current

SCD pour les flocons

Changements des niveaux supérieurs propagés vers le bas:

Ex: le 2015-08-21, la description de catégorie *Bread* est changée:

Category Key	Category Name	Descr.	Row Effective Date	Row Expiration Date	Current Row Indicator
1	Bread	desc 1	2010-01-01	2015-08-20	Expired
3	Bread	desc 3	2015-08-21	9999-12-31	Current
2	Desserts	desc 2	2010-01-01	9999-12-31	Current

Product Key	Product Name	Category Key	Row Effective Date	Row Expiration Date	Current Row Indicator
1	Cinnamon Roll	1	2010-01-01	2015-08-20	Expired
5	Cinnamon Roll	3	2015-08-21	9999-12-31	Current
2	Baked Whole Wheat Bread	1	2010-01-01	2015-08-20	Expired
6	Baked Whole Wheat Bread	3	2015-08-21	9999-12-31	Current
3	Treacle Tart 320 g	2	2010-01-01	9999-12-31	Current
4	Pecan Butter 64 fl oz	2	2010-01-01	9999-12-31	Current

Table des matières

Relational schemas for DWHs

- Schémas
- Implémentation relationnelle: modélisation dimensionnelle
- Table de faits
- Hiérarchies de dimension
- Attributs de dimension
- Changements dans les dimensions
- **Implémentation des dimensions sous Oracle**

2018/2019

Dimensions sous Oracle

[Oracle Database Online Documentation 12c (Data Warehousing)]

Définition d'une dimension (Oracle)

```
CREATE DIMENSION name_dimension
  <level definition clauses>
  <hierarchy definition clauses>
  <attribute dependencies clauses>
```

Dimensions sous Oracle SQL: étoile

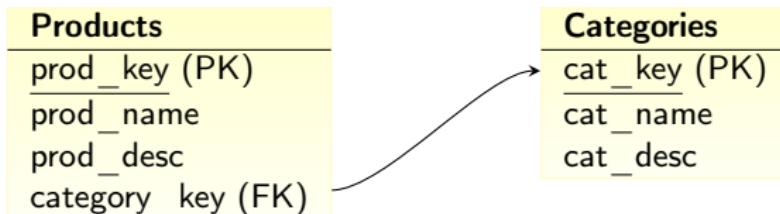
Définition d'une dimension

```
CREATE DIMENSION date_dim
  LEVEL day IS time.date
  LEVEL month IS time.calendar_month
  LEVEL year IS time.calendar_year
  LEVEL fis_month IS time.fiscal_month
  LEVEL fis_year IS time.fiscal_year
  HIERARCHY cal_rollup (
    day CHILD OF
    month CHILD OF
    year
  )
  HIERARCHY fis_rollup (
    day CHILD OF
    fis_month CHILD OF
    fis_quarter
  )
  ATTRIBUTE day DETERMINES
    (day_of_week, holiday_flag)
  ATTRIBUTE month DETERMINES
    (calendar_month_number_in_year)
;
```

Time

date
day_of_week
holiday_flag
calendar_month
calendar_month_number_in_year
calendar_year
fiscal_month
fiscal_year

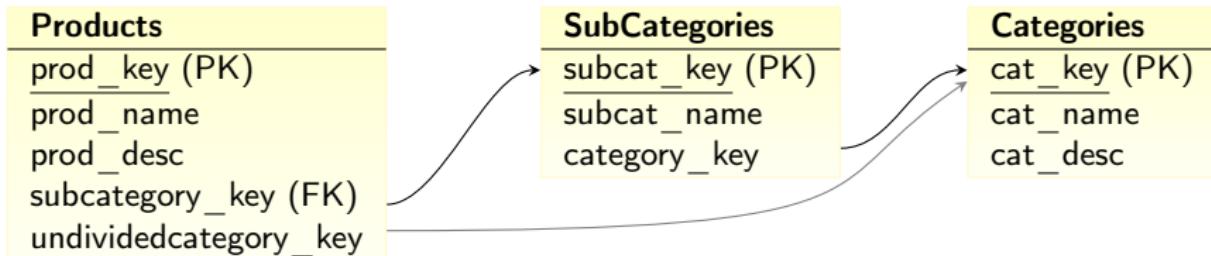
Dimensions sous Oracle SQL: flocons



Définition d'une dimension

```
CREATE DIMENSION products_dim
  LEVEL product IS products.prod_key
  LEVEL category IS categories.cat_key
  HIERARCHY prod_rollup (
    product CHILD OF
    category
  JOIN KEY (product.category_key) REFERENCES category
)
ATTRIBUTE product DETERMINES
  (prod_name, prod_desc)
ATTRIBUTE category DETERMINES
  (cat_name, cat_desc)
;
```

Dimensions sous Oracle SQL: flocons (2)



Définition d'une dimension (Oracle)

```
CREATE DIMENSION products_dim
  LEVEL product IS products.prod_key
  LEVEL subcategory IS subcategories.subcat_key SKIP WHEN NULL
  LEVEL category IS categories.cat_key
  HIERARCHY prod_rollup (
    product CHILD OF
    category
    JOIN KEY (products.subcategory_key) REFERENCES subcategory
    JOIN KEY (subcategories.category_key) REFERENCES category
    JOIN KEY (product.category_key) REFERENCES category
  )
;
```

Dimensions sous Oracle SQL

Modifier une dimension (Oracle)

```
ALTER DIMENSION products_dim DROP HIERARCHY prod_rollup;
ALTER DIMENSION products_dim ADD LEVEL brand IS categories.brand;
ALTER DIMENSION products_dim COMPILE; -- to re-validate
DROP DIMENSION products_dim;
...
```

Dimensions sous Oracle SQL: utilisation

Dimensions sont vues par Oracle comme des contraintes, utilisées par la

réécriture de requêtes.

Validation de dimensions

```
BEGIN  
    DBMS_DIMENSION.VALIDATE_DIMENSION(  
        dimension => 'products_dim',  
        incremental => false,  
        check_nulls => true,  
        statement_id=>'validation run 1');  
END;  
/
```

- incremental=true: valide seulement les nouveaux tuples
- check_nulls=true: pas de NULLS sauf dans les colonnes avec level 'SKIP WHEN NULL'
- statement_id: sert à identifier les tuples créés par cette validation dans la table dimension_exceptions

Dimensions sous Oracle SQL: utilisation

Afficher les informations de dimensions

```
EXECUTE DBMS_DIMENSION.DESCRIBE_DIMENSION('products_dim');
```