

Introduction à la modélisation dimensionnelle

Data warehouse
<http://dwh.crzt.fr>



Stéphane Crozat

Table des matières



Objectifs	4
Introduction	5
I - Principes de la modélisation dimensionnelle	6
1. Approche générale de modélisation	6
2. Table des faits	8
3. Table des dimensions	9
II - Projet Fantastique : Problème posé	11
III - Projet Fantastique : Données disponibles	12
IV - Étude des besoins utilisateurs	14
1. Requête décisionnelle	14
2. Rapport	15
3. Hiérarchie	15
V - Exercice : Projet Fantastique : Étude des besoins	16
VI - Étude des données	17
1. Étude séparée des sources données	17
2. Étude intégrée des sources de données	18
VII - Exercice : Projet Fantastique : Étude des données	19
VIII - Modélisation du datawarehouse	20
1. Intégration des besoins	20
2. Arbitrages pour le choix des données	21
3. Métadonnées	21
IX - Exercice : Projet Fantastique : Modélisation	23
X - Rappels	24

1. Rappels SQL pour l'étude des données	25
1.1. Fichier CSV	25
1.2. Agrégats	27
2. Rappels Oracle pour l'étude des données	28
2.1. Accès inter-schémas	28
2.2. Fenêtrage des données	28
2.3. Dictionnaire de données Oracle	29
3. Prise en main de Oracle SQL Developer	31
3.1. Installation de SQL Developer	31
3.2. Connexion avec SQL Developer	31
3.3. Naviguer dans le catalogue de SQL Developer	32
3.4. Exécuter des requêtes SQL avec SQL Developer	32
3.5. Écrire du PL/SQL avec SQL Developer	35
3.6. Exécution de fichiers SQL	37
Solutions des exercices	38
Abréviations	42
Bibliographie	43

Objectifs

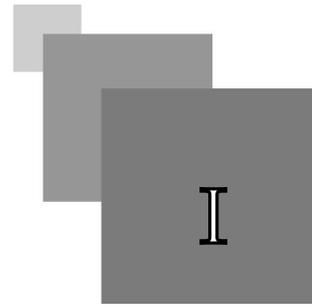
- Savoir faire une étude de besoins
- Savoir faire une étude des données existantes
- Savoir faire un modèle dimensionnel en étoile

Introduction



- Volume de cours : 2h
- Volume d'exercice : 4h

Principes de la modélisation dimensionnelle



Objectifs

Connaître les principes, les étapes et les méthodes de la modélisation dimensionnelle

1. Approche générale de modélisation

◆ Rappel

La modélisation en étoile (cf. p.)

🦋 Fondamental

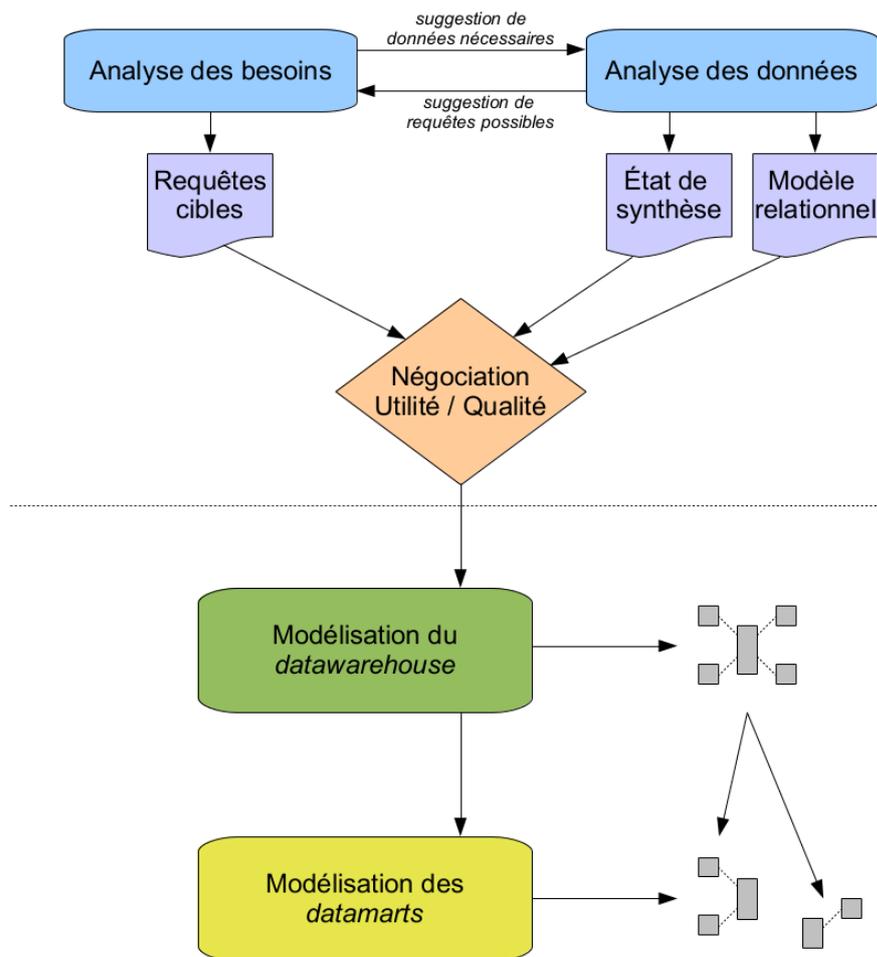
Un modèle dimensionnel est le résultat :

- d'une analyse des besoins : *ce que je souhaite étudier.*
- d'une analyse des données disponibles : *ce que je peux étudier.*

✂ Méthode : Méthode générale de modélisation

1. Analyse des données
 1. Étude des sources de données (quantification, analyses générales)
 2. Qualification des données (qualité et intérêt)
 3. Intégration logique des données (simulation d'un schéma relationnel virtuel)
 4. Normalisation du schéma virtuel en 3NF pour en avoir une vue cohérente
2. Analyse des besoins clients
 1. Exprimer les besoins sous la forme de requêtes décisionnelles
 2. Réaliser les vues hiérarchiques pour chaque requête
3. Sélectionner les requêtes qui seront effectivement réalisables en fonction des données disponibles
4. Conception du data warehouse et des data marts
 1. Séparer les requêtes en fonction de la granularité de la table des faits (grain fin des ventes, grain plus grossier du ticket de caisse, etc.)
 2. Créer un data warehouse intégrant toutes les requêtes de grain fin

3. Extraire un data mart par niveau de grain supérieur et/ou pour des thématiques particulières nécessitant par exemple une pré-agrégation



Éléments méthodologiques pour la modélisation dimensionnelle

Remarque

Il est généralement intéressant de paralléliser les tâches d'analyse des besoins et d'analyse des données.

En particulier il est inutile d'aller trop loin dans l'expression de besoins que l'on sait *a priori* impossibles à satisfaire pour cause d'absence de donnée ou d'absence de donnée exploitable.

Rappel : Informations

Il est conseillé de conserver certains champs d'information dans le modèle dimensionnel, même s'ils ne seront pas exploités pour les calculs ou les agrégats.

Cela permettra par exemple d'identifier des enregistrements, comme les désignations de produits.

On pourra noter en italique ces champs dans le modèle dimensionnel.

✂ Méthode : Granularité minimale



Preferably you should develop dimensional models for the most atomic information captured by a business process. Atomic data is the most detailed information collected; such data cannot be subdivided further.

(Kimball, Ross, 2008, p.34) *



✂ Méthode : Granularité des data marts

Pour un data mart on peut pré-agrégé sur un grain plus gros que le data warehouse : des colonnes d'agrégation (somme, moyenne, compte...) peuvent alors apparaître pour rendre compte statistiquement d'informations perdues à l'agrégation.

3. Table des dimensions

🔑 Définition



Dimension tables are the entry points into the fact table. [...] The dimension implement the user interface to the data warehouse.

(Kimball, Ross, 2008, p.20) *



👉 Exemple

Product Dimension Table
Product Key (PK)
Product Description
SKU Number (Natural Key)
Brand Description
Category Description
Department Description
Package Type Description
Package Size
Fat Content Description
Diet Type Description
Weight
Weight Units of Measure
Storage Type
Shelf Life Type
Shelf Width
Shelf Height
Shelf Depth
... and many more

Exemple de table de dimension (Kimball, Ross, 2008, p.20)

Conseil : Intelligibilité



The best attributes are textual and discrete. Attributes should consist of real words rather than cryptic abbreviations. (Kimball, Ross, 2008, p.20) *



Projet Fantastique : Problème posé

II

Vous travaillez en tant qu'ingénieur spécialisé dans les systèmes décisionnels au siège de l'entreprise française "Fantastique".

L'entreprise "Fantastique" vend principalement des ouvrages de divertissement de type science fiction, thriller, policier... Elle dispose pour cela de plusieurs magasins de vente dans les centres des grandes villes en France.

La direction de l'entreprise souhaite faire une étude large sur les ventes de l'année passée afin de prendre des orientations stratégiques nouvelles : ouverture de nouveaux magasins, fermeture ou transfert de magasins mal implantés, extension territoriale à de nouveaux départements français, réorganisation des directions, réorientation du marketing, élargissement ou réduction du catalogue, etc.

Fondamental

La question posée est donc : *quels sont les facteurs sur lesquels l'on pourrait jouer pour augmenter les ventes ?*

Elle vous charge dans ce cadre de mettre en place une solution logicielle permettant d'intégrer les données pertinentes et de pouvoir les interroger efficacement sous des angles divers.

Notons que bien entendu, la direction éclairée de l'entreprise ne compte pas se fier à ces seuls facteurs de ventes pour prendre ses décisions, mais bien privilégier les facteurs sociaux et territoriaux, en dialoguant avec ses salariés et ses clients, pour maintenir sa mission culturelle et son rôle d'entreprise citoyenne. Votre posture d'ingénieur est bien entendu de se préoccuper de ces dimensions fondamentales, même si elles seront largement ignorées dans le cadre de cet exercice à vocation essentiellement technique. Elle pourront néanmoins être brièvement abordées en marge de vos rapports d'analyse.

Projet Fantastic : Données disponibles



Catalogue des livres

Une base Oracle contient le catalogue complet de l'entreprise que chaque magasin a à sa disposition.

- Cette base, composée d'une seule table publique `catalogue`, est disponible sur le serveur Oracle `sme-oracle.sme.utc`, sous le schéma `nf26`.

Fichier des ventes

Un fichier contient une consolidation de l'ensemble des ventes de l'année passée réalisées dans chaque magasin.

- Ces données sont disponibles sous la forme d'un fichier CSV dans un répertoire du serveur `sme-oracle.sme.utc` : `/home/nf26/data`
- La structure du fichier est : Numéro de ticket, date de ticket, produit, magasin

Fichier des magasins

Un fichier ODS géré par la direction marketing contient pour chaque magasin l'organisation des rayonnages : `marketing.ods`

- Le responsable des ventes de chaque département décide de l'organisation des rayonnages des magasins de son département.
- Il existe 3 types de rayonnage : par Auteur (A), par Année (Y), par Éditeur (E)
- Le fichier est déposé dans un répertoire du serveur `sme-oracle.sme.utc` : `/home/nf26/fantastic`

Données géographique sur les départements

Un stagiaire a trouvé sur Internet un fichier permettant de connaître la population de chaque département, présageant que cette information sera utile.

- Le stagiaire parvient à trouver une information un peu datée qui pourra suffire sous la forme d'un fichier CSV : `departementsInsee2003.txt`.

[cf. `departementsInsee2003.txt` (population par département)]

Méthode

Inspecter les données pour chaque source :

1. Se connecter à la base Oracle avec SQL Developer et inspecter le schéma de la table Oracle.
2. Ouvrir un terminal et se connecter en `ssh` au serveur.
Utiliser la commande Unix `more` pour regarder les premières lignes du fichier `data.csv`.
3. Récupérer le fichier CSV `departementsInsee2003.txt` et l'ouvrir avec un éditeur de texte.
4. Récupérer le fichier ODS et l'ouvrir avec un traitement de texte.

Étude des besoins utilisateurs

IV

Objectifs

Savoir formaliser une étude de besoins sous forme de requêtes multidimensionnelles

1. Requête décisionnelle

Définition : Requête décisionnelle

Une requête décisionnelle exprime toujours la mesure d'une quantification de *faits* par rapport à des *dimensions*, sous une forme du type : "Quelle a été la quantité de ... en fonction de ...".

Synonyme : Vue, requête multidimensionnelle

Fondamental

- Les faits sont des grandeurs que l'on cherche à mesurer (prix, quantité...)
- Les dimensions sont des axes d'analyse (date, lieu, produit, personne...)

Syntaxe

```
1 quantité de faits
2 / dimension1
3 / dimension2
4 ...
```

Exemple

"Quelle a été la quantité de produits vendus en fonction des départements et des mois de l'année."

```
1 quantité de produits vendus
2 / département
3 / mois
```

2. Rapport

Définition : Rapport

La réponse à une requête décisionnelle est un rapport, généralement sous une forme tabulaire ou graphique.

Synonyme : État

Exemple

01												02												03											
J	F	M	A	M	J	J	A	S	O	N	D	J	F	M	A	M	J	J	A	S	O	N	D	J	F	M	A	M	J	J	A	S	O	N	D
04												05												06											
J	F	M	A	M	J	J	A	S	O	N	D	J	F	M	A	M	J	J	A	S	O	N	D	J	F	M	A	M	J	J	A	S	O	N	D
07												08												09											
J	F	M	A	M	J	J	A	S	O	N	D	J	F	M	A	M	J	J	A	S	O	N	D	J	F	M	A	M	J	J	A	S	O	N	D

Exemple de rapport

Méthode

Les besoins des utilisateurs s'expriment en général plus facilement sous la forme d'exemples de rapports recherchés.

3. Hiérarchie

Définition : Hiérarchie

Une hiérarchie est un ensemble de paramètres d'étude de granularité croissante appartenant à une même dimension au sein d'un modèle décisionnel.

Exemple

- 1 quantité de produits vendus
- 2 / lieu (département, région)
- 3 / date (jour, mois, trimestre, semestre)

Remarque

Une même dimension peut contenir plusieurs hiérarchies.

Étude des données

VI

Objectifs

Savoir faire une étude des données existantes

1. Étude séparée des sources données

✂ Méthode

Pour chaque source de données identifiée on proposera une synthèse avec les informations suivantes :

- Nom, origine précise, nombre de lignes de la source
- Nom, type et description des colonnes
- Qualité
 - 0 : données inexploitable
 - 1 : données peu exploitables (traitements incertains)
 - 2 : données exploitables après traitements
 - 3 : données exploitables sans traitement
- Utilité
 - 0 : données sans intérêt
 - 1 : données utiles pour la documentation uniquement
 - 2 : données utiles a priori
 - 3 : données utiles avec certitude
- Commentaires.

☞ Exemple

Nom	Description	Type	Qualité	Utilité	Commentaire
isbn	Identifiant international d'un livre publié	char(3) et char(13)	3	3	Certaines ISBN ne sont pas corrects (3 caractères au lieu de 13) ; clé de référencement dans le fichier de ventes
titre	Titre du livre	varchar(255)	2	1	
auteur	Auteur(s) du livre	varchar(255)	1	2	

					Peut contenir plusieurs auteurs ; certains auteurs sont inscrits différemment d'un livre à l'autre
...					

Table Oracle oracle.utc.fr/schema.table [1000 lignes]

2. Étude intégrée des sources de données

Méthode

Afin d'avoir une vision globale de l'ensemble des données, il est conseillé de rétro-concevoir :

- une représentation relationnelle des données telles qu'elles existent
- une représentation relationnelle des données idéalisées (telles qu'elles existeraient si elles étaient normalisées dans une même BD)
- une représentation conceptuelle en UML

Exercice : Projet Fantastique : Étude des données

VII

[1h]

Afin de réaliser votre travail, l'entreprise vous met à disposition les données suivantes.

Dans le contexte de cet exercice, les données vous sont livrées *a priori*, notez que dans un contexte réel, vous aurez la plupart du temps à rechercher vous même les données qui peuvent exister.

Données disponibles (cf. p.12)

Question 1

[solution n°3 p.39]

Établissez le modèle *relationnel* sous-jacent aux données présentes.

Indice :

Pour initier une connexion ssh : `ssh user@serveur`

Question 2

Étudiez les données dont vous disposez et proposez une synthèse des données disponibles pour chaque source.

Indice :

Pour compter les lignes d'un fichier texte sous Linux, on peut utiliser la commande `wc -l`

Question 3

[solution n°4 p.39]

Afin de clarifier les données et leur organisation, rétro-concevez un modèle relationnel *normalisé* en troisième forme normale unifiant toutes les données grâce à l'identification de clés primaires et l'expression de clé étrangères.

Question 4

Rétro-concevez le modèle *conceptuel* en UML correspondant au modèle relationnel normalisé (un modèle UML peut être plus facile à utiliser ensuite).

Modélisation du datawarehouse

VIII

Objectifs

Savoir faire un modèle dimensionnel en étoile pour un cas simple

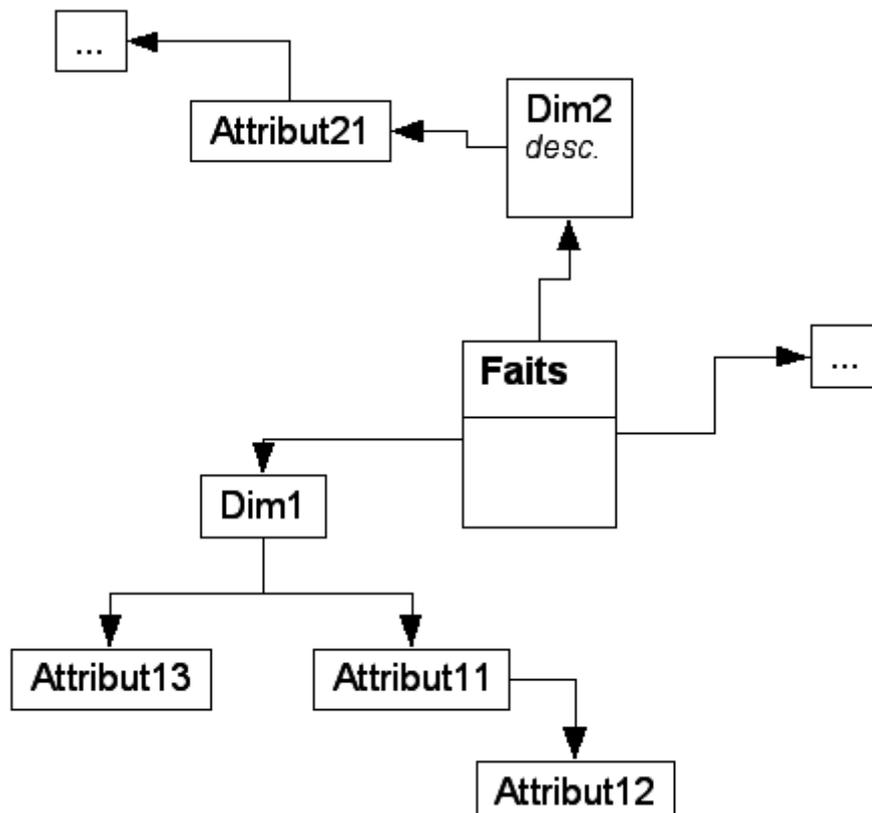
1. Intégration des besoins

✂ Méthode

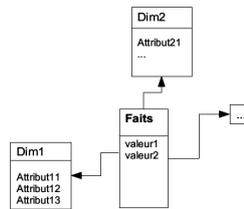
Les différentes vues formalisées lors de l'étude des besoins utilisateurs doivent être intégrées :

- une pour le data warehouse,
- et pour chaque data mart.

📦 Syntaxe



Vue intégrée



Modèle dimensionnel du data warehouse

2. Arbitrages pour le choix des données

✂ Conseil : Pilotage par les besoins

Un projet de *datawarehouse* est destiné à répondre à des besoins d'analyse, en cela il doit prioritairement prendre en considération les besoins réels d'analyse des utilisateurs. En particulier il est déconseillé de travailler uniquement à partir des données sources existantes, puisque les utilisateurs cherchent justement à sortir de la réalité transactionnelle (Entrepôts de données : guide pratique de modélisation dimensionnelle *, p.34).

Bien entendu, la possibilité de répondre aux besoins dépend des données disponibles et de leur qualité.

✂ Méthode : Diagramme Utilité / Qualité

Chaque données peut être qualifiée selon une utilité (donnée intéressante ou non) et une qualité (données facile à exploiter ou coûteuse à nettoyer).

Ces deux paramètres peuvent ensuite servir à choisir les données que l'on conserve (et que l'on va devoir éventuellement nettoyer) de celle que l'on abandonne.

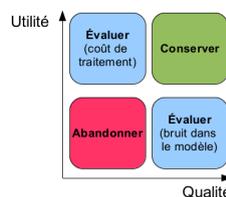


Diagramme Utilité / Qualité

3. Métadonnées

✂ Méthode

Documentez la table des faits ainsi que chaque dimension de votre data warehouse, avec pour chaque attribut :

- Le nom
- Le type précis
- Une description

On identifiera les clés candidates et la clé primaire de chaque dimension, les informations uniquement descriptives.

On pourra préciser lorsque ce n'est pas évident le mode de calcul des attributs en fonction des sources.

Exercice : Projet Fantastique : Modélisation

IX

À partir de l'étude des données et des besoins utilisateurs, proposez un modèle *dimensionnel* pour un *data warehouse* permettant de traiter la question générale qui a été posée.

Question 1

[solution n°5 p.39]

Proposez une modélisation dimensionnelle *en étoile* pour chaque contexte d'usage (directions marketing et éditoriale) :

1. identifiez la table des faits
2. identifiez les dimensions en intégrant les vues exprimées (utilisez le rapport qualité/utilité pour décider des données que vous souhaitez conserver)

Question 2

[solution n°6 p.40]

Intégrer vos deux sous-modèles pour proposer le modèle de votre *data warehouse*.

Vous pourrez augmenter vos dimensions de données disponibles bien que non identifiées a priori lors de l'analyse des besoins.

Question 3

[solution n°7 p.40]

Établissez les métadonnées du modèle dimensionnel : décrivez chaque données (type précis, description...) ; identifiez la clé primaire de chaque dimension, ainsi que les informations descriptives.

1. Rappels SQL pour l'étude des données

1.1. Fichier CSV

Définition : Fichier CSV

CSV * est un format informatique permettant de stocker des données tabulaires dans un fichier texte.

Chaque ligne du fichier correspond à une ligne du tableau. Les valeurs de chaque colonne du tableau sont séparées par un caractère de séparation, en général une *virgule* ou un *point-virgule*. Chaque ligne est terminée par un *caractère de fin de ligne* (*line break*).

Toutes les lignes contiennent *obligatoirement* le même nombre de valeurs (donc le même nombre de caractères de séparation). Les valeurs vides doivent être exprimées par deux caractères de séparation contigus.

La taille du tableau est le nombre de lignes multiplié par le nombre de valeurs dans une ligne.

La première ligne du fichier peut être utilisée pour exprimer le nom des colonnes.

Syntaxe

```
1 [NomColonne1;NomColonne2;...;NomColonneN]
2 ValeurColonne1;ValeurColonne2;...;ValeurColonneN
3 ValeurColonne1;ValeurColonne2;...;ValeurColonneN
4 ...
```

Exemple : Fichier CSV sans entête

```
1 Pierre;Dupont;20;UTC;NF17
2 Pierre;Dupont;20;UTC;NF26
3 Paul;Durand;21;UTC;NF17
4 Jacques;Dumoulin;21;UTC;NF29
```

Exemple : Fichier CSV avec entête

```
1 Prenom;Nom;Age;Ecole;UV
2 Pierre;Dupont;20;UTC;NF17
3 Pierre;Dupont;20;UTC;NF26
4 Paul;Durand;21;UTC;NF17
5 Jacques;Dumoulin;21;UTC;NF29
```

Exemple : Valeur nulle

```
1 Jacques;Dumoulin;;UTC;NF29
```

L'âge est inconnu (NULL).

 *Attention : Variations...*

La syntaxe des fichiers CSV n'est pas complètement standardisée, aussi des variations peuvent exister :

- Les chaînes de caractères peuvent être protégées par des guillemets (les guillemets s'expriment alors avec un double guillemet).
- Le caractère de séparation des nombres décimaux peut être le point ou la virgule (si c'est la virgule, le caractère de séparation doit être différent)
- ...

Un des problèmes les plus importants reste l'encodage des caractères qui n'est pas spécifié dans le fichier et peut donc être source de problèmes, lors de changement d'OS * typiquement.

 *Méthode : Usage en base de données*

Les fichiers CSV sont très utilisés en BD * pour échanger les données d'une table (export/import).

Les SGBD * contiennent généralement des utilitaires permettant d'exporter une table ou un résultat de requête sous la forme d'un fichier CSV, en spécifiant un certain nombre de paramètres (caractère de séparation de valeur, caractère de fin de ligne, présence ou non d'une ligne de définition des noms des colonnes, etc.). De même ils proposent des utilitaires permettant d'importer un fichier CSV dans une table (en spécifiant les mêmes paramètres), voire de créer directement une table à partir du fichier CSV (quand les noms des colonnes sont présents).

 *Complément : Fichiers à largeur de colonne fixe*

Les fichiers à largeur de colonne fixe n'utilisent pas de séparateur de colonne, mais imposent *le même nombre de caractères* pour chaque cellule. L'avantage est de ne pas avoir à spécifier le caractère de séparation, l'inconvénient est la taille de fichier supérieure si les valeurs ne font pas toutes la même largeur.

 *Complément : XML*

Les fichiers XML * tendent de plus en plus à remplacer les fichiers CSV car ils permettent d'être beaucoup plus expressifs sur le schéma d'origine. Ils sont également plus standards (encodage spécifié, principe de séparation des données par les *tags*, etc.). Leur seul inconvénient est d'être plus verbeux et donc plus volumineux.

 *Complément : Tables externes*

Certains SGBD, comme Oracle, permettent de créer des tables dites *externes*, qui autorisent de créer un schéma de table *directement sur un fichier CSV*, permettant ainsi un accès SQL standard à un fichier CSV, sans nécessité de l'importer d'abord dans une table.

1.2. Agrégats

Définition : Agrégat

Un agrégat est un partitionnement horizontal d'une table en sous-tables, en fonction des valeurs d'un ou plusieurs attributs de partitionnement, suivi de l'application d'une fonction de calcul à chaque attribut des sous-tables obtenues.

Syntaxe

```
1 SELECT <liste d'attributs de partitionnement à projeter et de fonctions de calcul>
2 FROM <liste de relations>
3 WHERE <condition à appliquer avant calcul de l'agrégat>
4 GROUP BY <liste ordonnée d'attributs de partitionnement>
5 HAVING <condition sur les fonctions de calcul>
```

Exemple

```
1 SELECT Societe.Nom, AVG(Personne.Age)
2 FROM Personne, Societe
3 WHERE Personne.NomSoc = Societe.Nom
4 GROUP BY Societe.Nom
5 HAVING Count(Personne.NumSS) > 10
```

Cette requête calcul l'âge moyen du personnel pour chaque société comportant plus de 10 salariés.

Remarque : Restriction

Une restriction peut être appliquée avant calcul de l'agrégat, au niveau de la clause WHERE, portant ainsi sur la relation de départ, mais aussi après calcul de l'agrégat sur les résultats de ce dernier, au niveau de la clause HAVING.

Attention : Projection

Si dans la clause SELECT, un attribut est projeté directement, sans qu'une fonction lui soit appliquée, alors il faut impérativement que cet attribut apparaisse dans la clause GROUP BY (car ce ne peut être qu'un attribut de partitionnement).

Remarque : Fonctions de calcul sans partitionnement

Si une ou plusieurs fonctions de calcul sont appliquées sans partitionnement, le résultat de la requête est un tuple unique.

Remarque : Intérêt de la clause GROUP BY

Pour que l'utilisation de la clause GROUP BY ait un sens, il faut qu'au moins une fonction de calcul soit utilisée, soit dans la clause SELECT, soit dans la clause HAVING.

Remarque : Contrôle imposé par quelques SGBDR

Notons que dans le cas de certains SGBDR * (par exemple Oracle), l'ensemble des attributs de l'agrégation (clause GROUP BY) doivent être préalablement projetés (donc déclarés dans la clause SELECT).

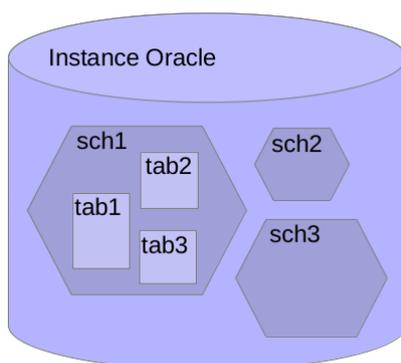
2. Rappels Oracle pour l'étude des données

2.1. Accès inter-schémas

Un schéma Oracle correspond au sein d'une base de données (instance Oracle) à un espace isolé comportant toutes les tables appartenant à d'un utilisateur du SGBD. Ainsi chaque utilisateur possède un schéma.

Pour simplifier on pourrait plutôt dire qu'une base Oracle est en fait une collection de BD et qu'un schéma est une BD.

Lorsqu'un utilisateur se connecte il se connecte généralement dans son propre schéma. Il lui est néanmoins possible de travailler sur les tables d'autres schémas, à condition d'avoir les droits associés bien entendu.



Organisation en schémas d'une instance Oracle

Syntaxe : Lecture des données d'une table d'un autre schéma

Il suffit de préfixer les noms des tables de leur nom de schéma.

```
1 SELECT * FROM schema.table;
```

2.2. Fenêtrage des données

Syntaxe : Rownum

```
1 SELECT ... FROM ... WHERE rownum<=N;
2 -- avec N le nombre de lignes désirées.
```

Rownum

La restriction ROWNUM <= N dans la clause WHERE permet filtrer les N premières lignes de la table.

 *Remarque*

rownum est une pseudo colonne qu'il est bien entendu possible de projeter : `SELECT rownum FROM ...`

 *Syntaxe : Utilisation avancée*

```
1 SELECT a1, ..., aN FROM
2 (SELECT a1, ..., aN, rownum AS rnum FROM t)
3 WHERE rnum BETWEEN n1 AND n2
```

Cette syntaxe permet de sélectionner une fenêtre sur les données et pas seulement les N premières lignes.

 *Méthode : Exploration de données massives*

Lorsque l'on est en présence de gros volumes de données, et que l'on veut se faire une idée du contenu de ces données, il n'est *pas souhaitable* de faire un simple `SELECT *`. En il serait trop long de rapatrier les dizaines de milliers de lignes et de plus cela serait inutile puisque seules quelques unes seraient effectivement lues.

L'usage de rownum permet de s'intéresser à des fenêtres de données représentatives, pour se faire une idée générale.

2.3. Dictionnaire de données Oracle

 *Rappel*

Le dictionnaire des données contient la description des objets créés et maintenus par le serveur Oracle.

 *Syntaxe : Lister tous les objets appartenant à l'utilisateur*

```
SELECT * FROM user_objects;
```

 *Syntaxe : Lister sélectivement certains objets*

```
- SELECT * FROM user_tables;
- SELECT * FROM user_sequences;
- SELECT * FROM user_views;
- SELECT * FROM user_procedures;
- SELECT * FROM user_indexes ;
- ...
```

 *Syntaxe : Décrire un objet*

```
- DESCRIBE nom_objet
```

Syntaxe : Lister les contraintes

```

1 SELECT table_name, constraint_type AS type, constraint_name, search_condition
2 FROM user_constraints
3 ORDER BY table_name, constraint_type

```

Complément : Recyclebin

Selon la configuration du serveur Oracle, il peut y avoir une gestion de corbeille (*recyclebin*) qui maintient des objets supprimés en les renommant avec un nom préfixé de *BIN\$*.

Pour exclure ces objets des requêtes au catalogue, ajouter une contrainte de type WHERE adéquate, par exemple :

```

1 SELECT table_name, constraint_type AS type, constraint_name, search_condition
2 FROM user_constraints
3 WHERE constraint_name NOT LIKE 'BIN$%'
4 ORDER BY table_name, constraint_type

```

Il est également possible de purger la corbeille avec la commande : `PURGE recyclebin.`

Complément

- Afficher les différents types d'objets appartenant à l'utilisateur :
`SELECT object_type FROM user_objects;`
- Afficher la liste des objets appartenant à l'utilisateur :
`SELECT * FROM user_catalog;`

Rappel : BD "Gestion des intervenants" : Schéma relationnel

```

1 tIntervenant (#pknom:vchar, prenom:vchar, poste:integer)
2 tCours (#pkannee:2000..2100, #pknum:integer, titre:vchar, type:C|TD|TP, fkintervenant=>tIntervenant,
   debut:date)

```

Exemple : BD "Gestion des intervenants" : Utilisation du catalogue

```

1 SELECT table_name FROM user_tables;
2
3 DESCRIBE tCours;
4
5 SELECT table_name, constraint_type AS type, constraint_name, search_condition
6 FROM user_constraints
7 WHERE constraint_name NOT LIKE 'BIN$%'
8 ORDER BY table_name, constraint_type
9

```

```

1 TABLE_NAME
2 -----
3 TINTERVENANT
4 TCOURS

```

3. Prise en main de Oracle SQL Developer

3.1. Installation de SQL Developer

Site Web

http://www.oracle.com/technology/products/database/sql_developer

(Téléchargement pour Windows, Mac et Linux)

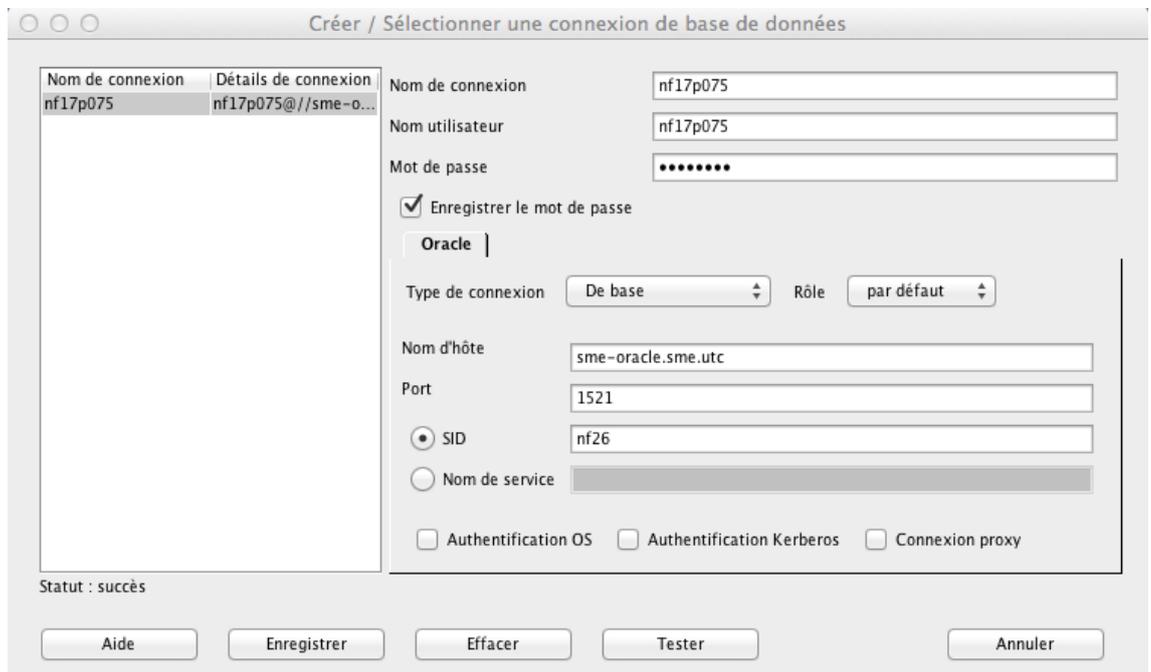
Complément

Documentation en ligne disponible à la même adresse.

3.2. Connexion avec SQL Developer

Créer une connexion à la base de données Oracle de l'UTC

1. Clic droit sur Connexions, puis Nouvelle connexion
2. Entrez vos paramètres :
 - Nom libre
 - Username / Password
 - Hôte : *sme-oracle.sme.utc*
 - SID : *nf26*



Nom de connexion	Détails de connexion
nf17p075	nf17p075@//sme-o...

Nom de connexion : nf17p075

Nom utilisateur : nf17p075

Mot de passe :

Enregistrer le mot de passe

Oracle |

Type de connexion : De base | Rôle : par défaut

Nom d'hôte : sme-oracle.sme.utc

Port : 1521

SID : nf26

Nom de service

Authentification OS Authentification Kerberos Connexion proxy

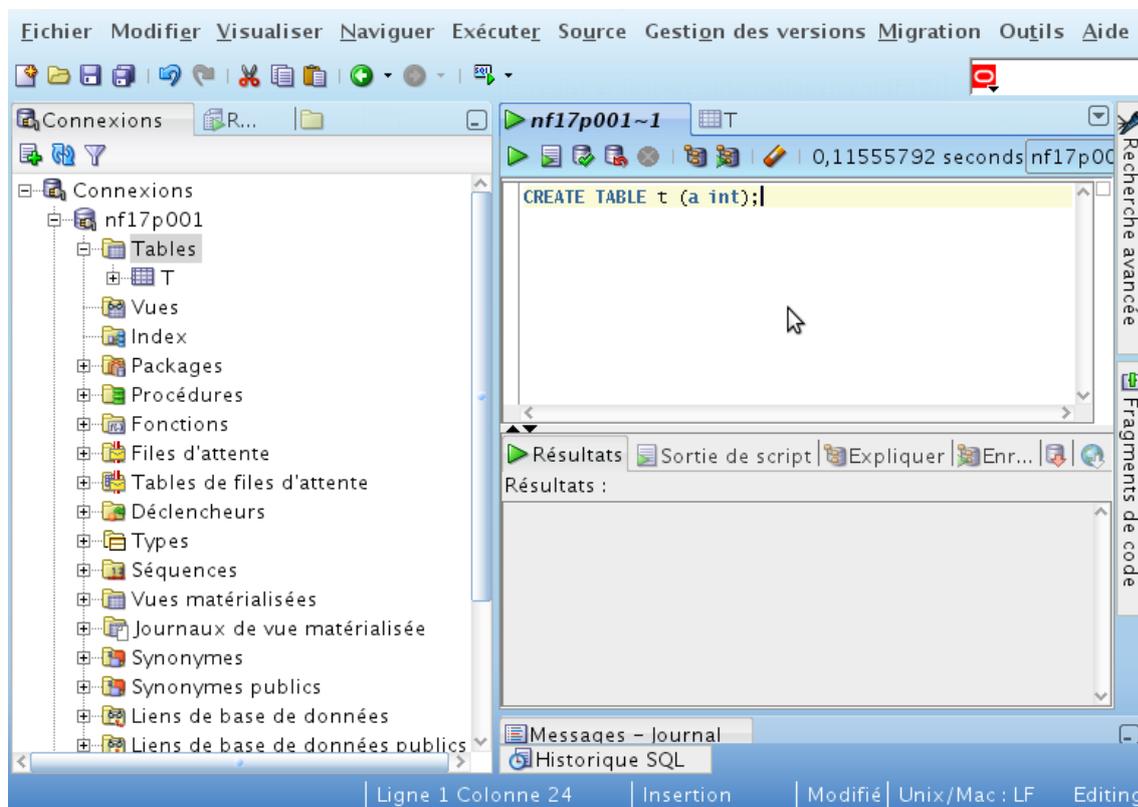
Statut : succès

Aide Enregistrer Effacer Tester Annuler

Fenêtre de connexion SQL Developer

3.3. Naviguer dans le catalogue de SQL Developer

L'espace de gauche permet de naviguer dans le catalogue de la base de données et donc de visualiser les tables, vues, index, etc.



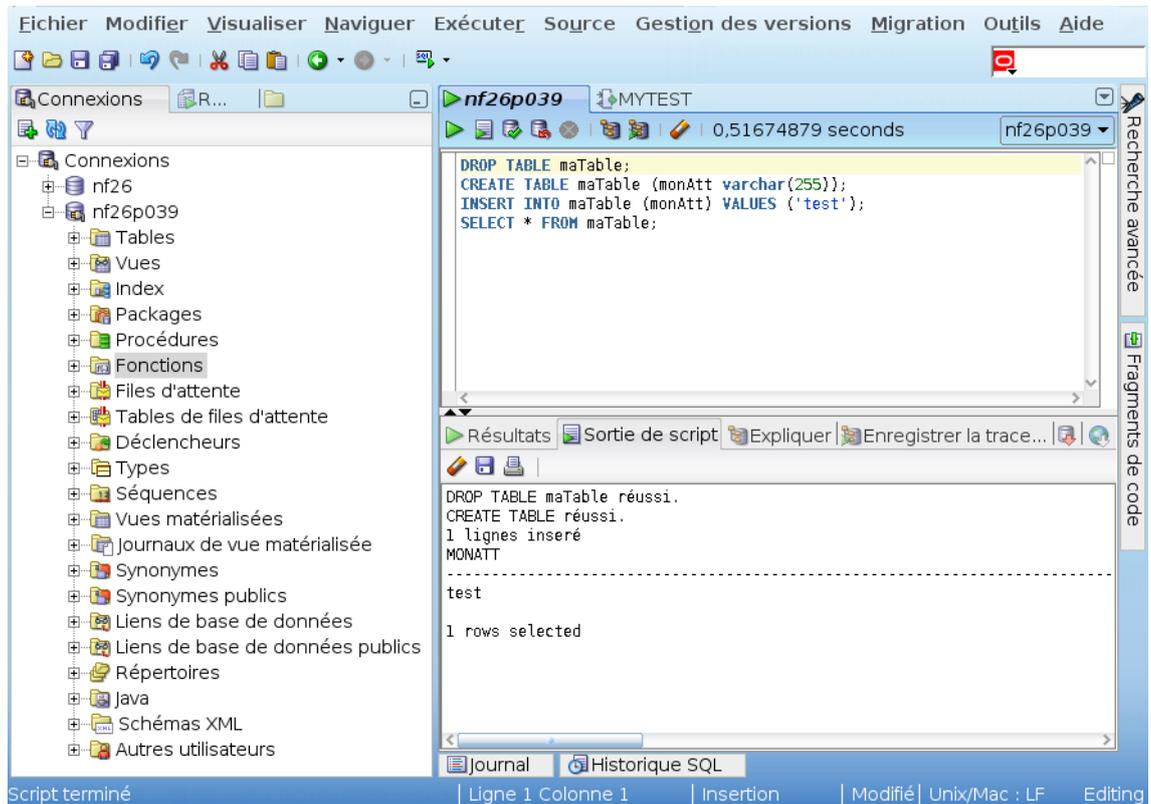
Fenêtre principale SQL Developer

⚠ Attention

Pour rafraîchir la vue de gauche (catalogue) après une requête LDD, il faut faire un clic droit sur l'élément du catalogue (par exemple Tables après une création de table) puis sélectionner Régénérer.

3.4. Exécuter des requêtes SQL avec SQL Developer

L'espace de droite permet d'écrire des requêtes SQL (en haut) et de visualiser le résultat de l'exécution (en bas). Appuyer sur le bouton Exécuter un script  ou faire F5 pour exécuter les requêtes saisies.



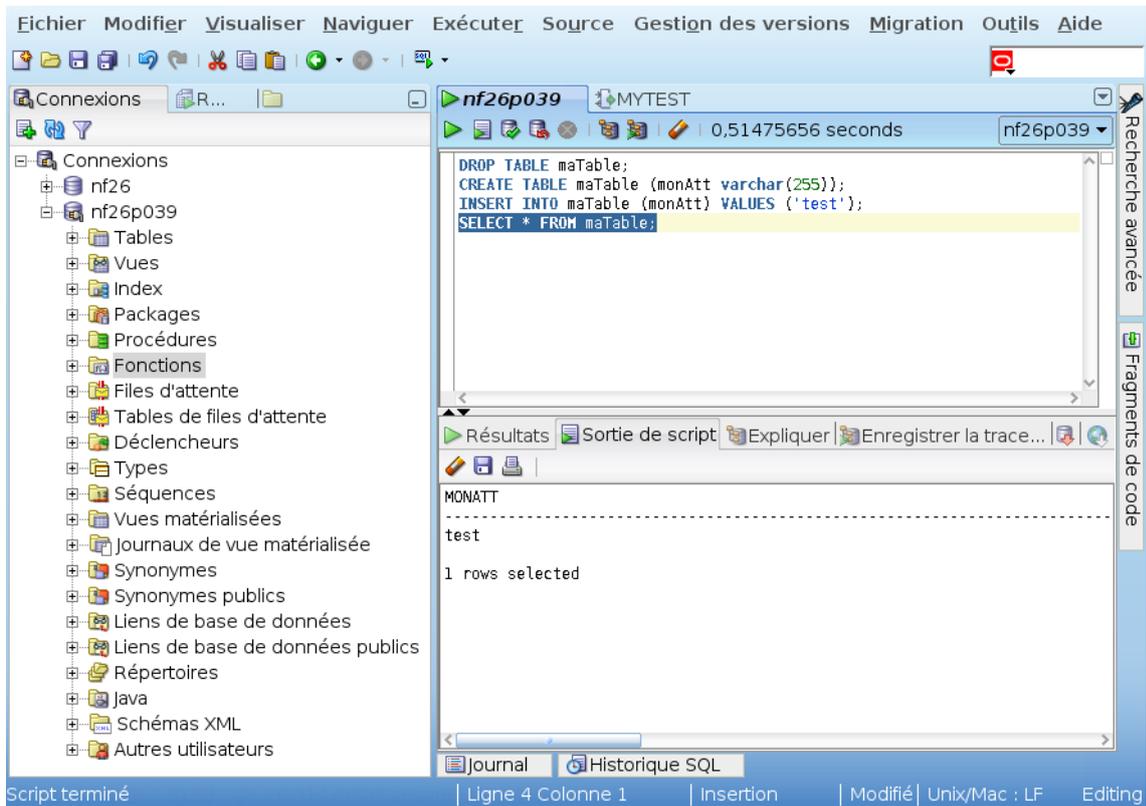
SQL (script)

Remarque

Pour effacer les résultats d'exécution précédents, cliquer sur Effacer .

Exécuter une seule requête au sein d'un script

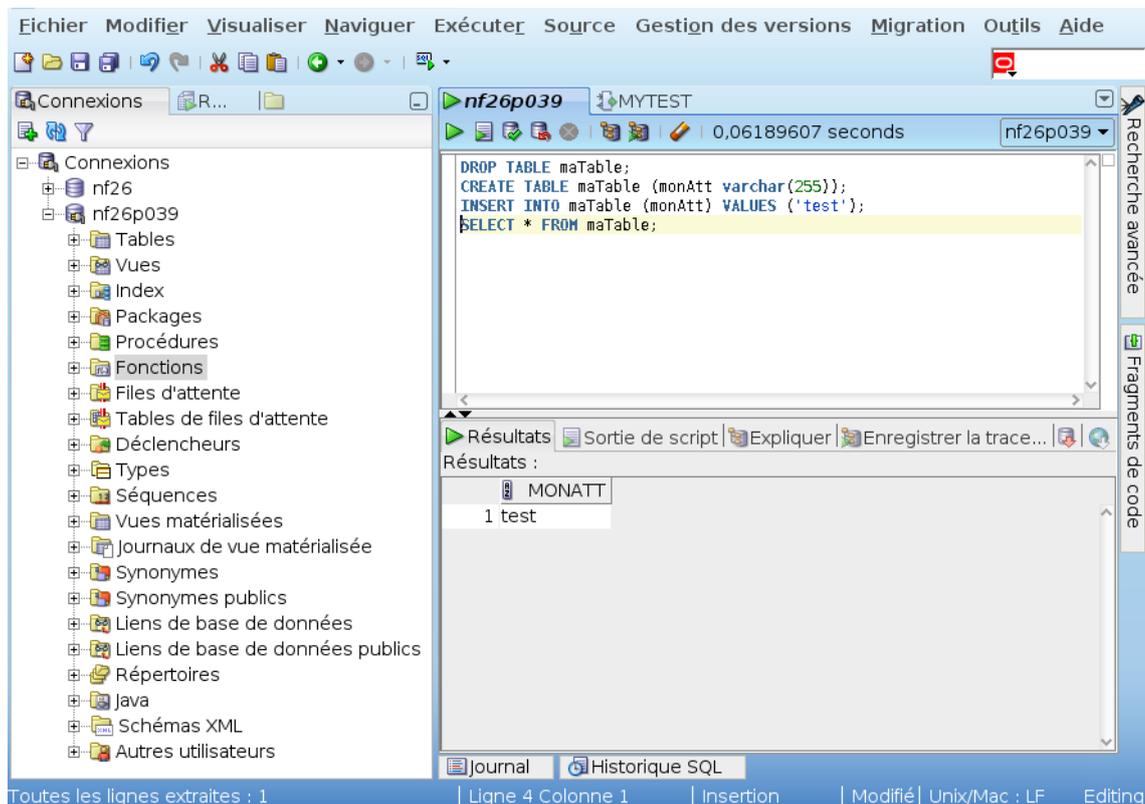
Pour n'exécuter qu'une seule requête parmi celle saisies dans la zone du haut, la sélectionner, puis cliquer sur Exécuter un script ou faire F5.



SQL (instruction par instruction)

Exécuter une seule requête SELECT

Pour les requêtes SELECT, SQL Developer propose un affichage sous forme de tableau, plus lisible que l'affichage texte. Pour exécuter une requête SELECT et obtenir un tel affichage, cliquer sur Exécuter l'instruction  ou faire F9.



SQL (une seule instruction SELECT)

⚠ Attention

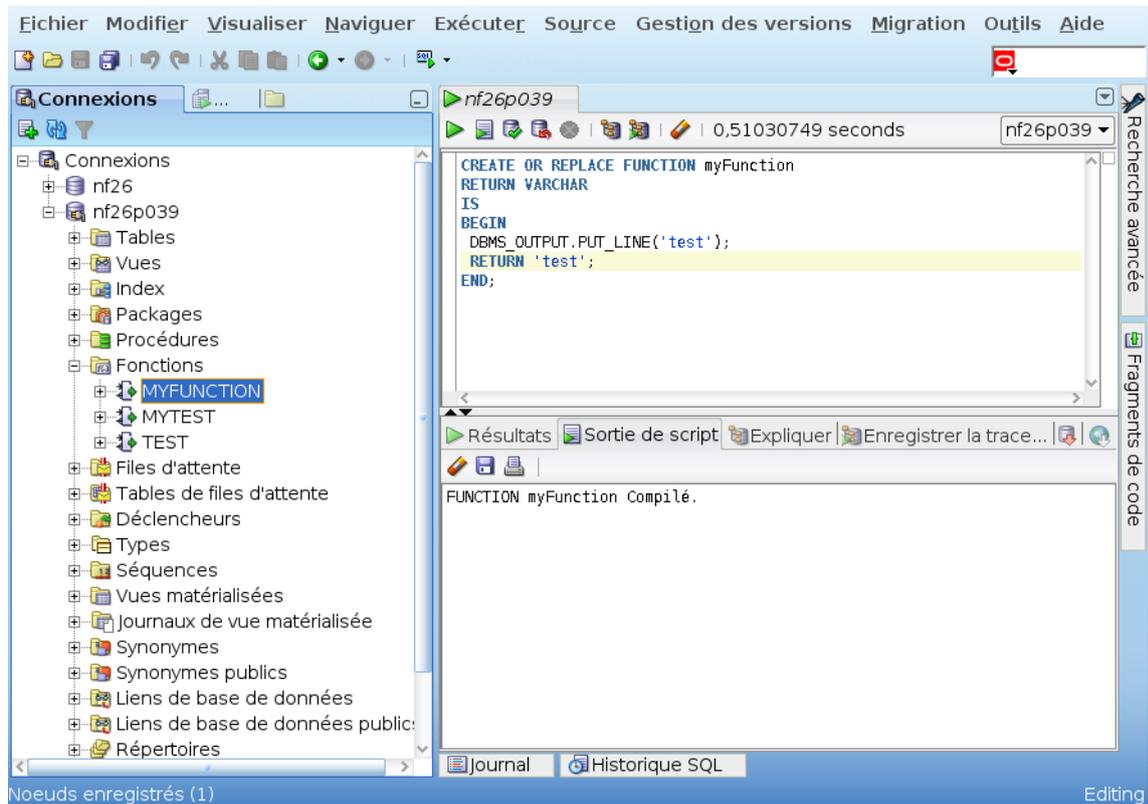
Le bouton **Exécuter** l'instruction n'affiche pas les erreurs d'exécution ou les confirmations de création ou insertion (requêtes CREATE, INSERT, UPDATE), il est donc à réserver aux requêtes SELECT valides (si le résultat n'est pas correct, utiliser **Exécuter un script**).

🔧 Conseil

Dans le doute utilisez toujours F5 et jamais F9.

3.5. Écrire du PL/SQL avec SQL Developer

Pour créer des blocs PL/SQL, procéder comme pour le SQL, avec le bouton **Exécuter un script**.



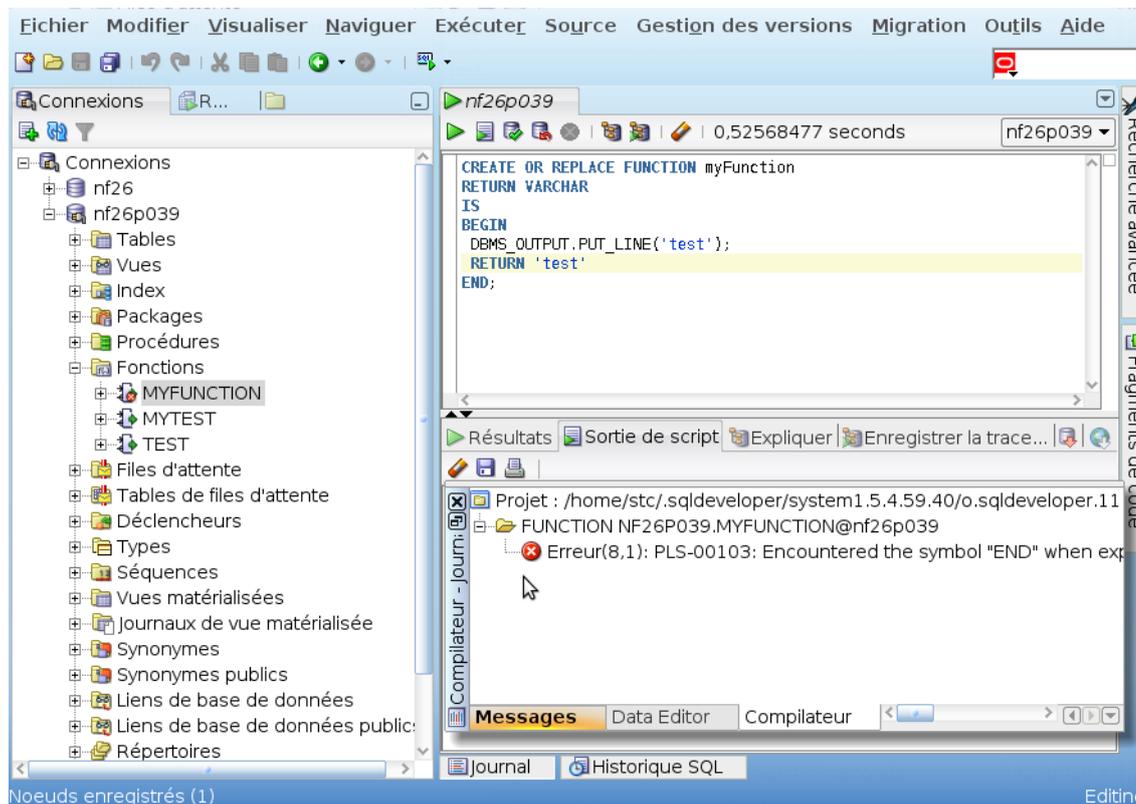
PL/SQL (Fonction compilée avec succès)

⚠ Attention : Show errors

Terminez vos blocs PL/SQL par l'instruction `show errors`, sinon le compilateur ne retourne pas de message.

🖨 Complément : Traiter les erreurs de compilation

Pour mieux voir les erreurs de compilation, il faut faire un clic droit sur le nom de la fonction ou de la procédure (zone de gauche), puis choisir `Compiler pour le débogage`.



PL/SQL (Fonction en erreur)

3.6. Exécution de fichiers SQL

✂ Méthode

Pour enregistrer un script SQL ou PL/SQL écrit dans *Oracle SQL Developer* sous la forme d'un fichier utiliser la fonction `file > save as`.

✂ Méthode

Pour exécuter un fichier SQL ou PL/SQL utiliser la commande `@fichier.sql`

☞ Exemple : *script.sql*

```
1 @file1.sql
2 @file2.sql
3 @file3.sql
```

Solutions des exercices



> Solution n°1

Exercice p. 16

Requêtes brutes

```
1 Quantite
2 / rayonnage
3 / rayon_bs
4 / rayon_recent
```

```
1 Quantite
2 / semaine
3 / mois, trimestre
4 / jds
5 / rayonnage
6 / rayon_bs
```

```
1 Quantite
2 / magasin, dpt
```

Requêtes organisées par dimension

```
1 Quantite
2 / date (semaine)
3 / date (mois, trimestre)
4 / date (jds)
5 / magasin (rayonnage)
6 / magasin (rayon_bs)
7 / magasin (recent)
```

```
1 Quantite
2 / magasin (dpt)
```

> Solution n°2

Exercice p. 16

Requêtes brutes

```
1 Quantite
2 / produit
3 / magasin, departement
4 / date, semaine
```

```
5 / date, mois, trimestre
```

```
1 Quantite
2 / auteur
3 / editeur
```

```
1 Quantite
2 / jds
3 / jds, semaine
4 / parution
```

Requêtes organisées par dimension

```
1 Quantite
2 / produit
3 / magasin (departement)
4 / date (semaine)
5 / date (mois, trimestre)
```

```
1 Quantite
2 / produit (auteur)
3 / produit (editeur)
```

```
1 Quantite
2 / date (jds)
3 / date (semaine)
4 / produit (parution)
```

> **Solution n°3**

Exercice p. 19

```
1 catalogue(isbn,titre,auteur,langue,parution,editeur, genre)
2 data(num,magasin,date,isbn)
3 marketing(dpt,rayonnage,ray_nom,magasin,rayon_bs, rayon_recent)
4 dpt(dpt,nom,pop)
```

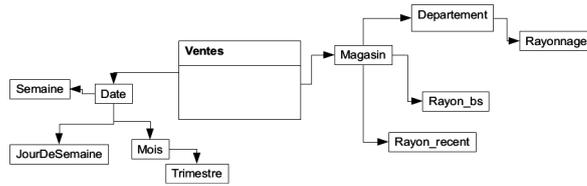
> **Solution n°4**

Exercice p. 19

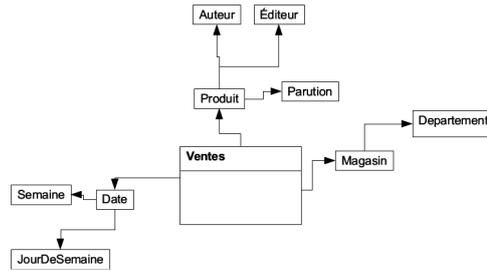
```
1 auteur(#num,nom,prenom)
2 langue(#langue)
3 editeur(#editeur)
4 catalogue(#isbn,titre,fkauteur=>auteur,langue=>langue,parution,editeur=>editeur,genre)
5 data(num,magasin=>magasin,date,isbn=>catalogue)
6 magasin(#magasin, dpt=>dpt,rayonnage,ray_nom,rayon_bs, rayon_recent)
7 dpt(#dpt,nom,pop)
```

> **Solution n°5**

Exercice p. 23



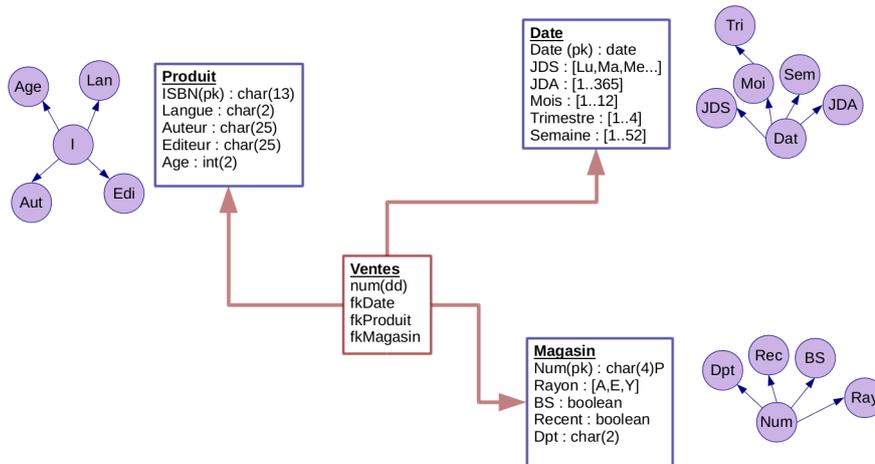
Vue direction marketing



Vue direction éditoriale

> **Solution n°6**

Exercice p. 23



Modèle dimensionnel du data warehouse Fantastic

> **Solution n°7**

Exercice p. 23

#ISBN	Char(13)	Clé primaire
Langue	Char(3)	Langue du livre sur trois lettres
Auteur	Varchar(50)	Auteur (Nom, Prénom) ; seul le premier auteur est conservé
Editeur	Varchar(50)	Type de produit
Parution	Char(4)	Année de parution sur 4 chiffres

Description de la dimension Produit (DataWarehouse)

#NumMag	Char(4)	Mxxx
Rayon_bs	0..1	0 : pas de rayon BS / 1 : un rayon BS
Dpt	Char(2)	Numéro du département
Population	1..20	Nombre d'habitants du département en centaine de milliers
Rayonnage	{A,Y,E}	Type de rayonnage dans le département

Description de la dimension Magasin (DataWarehouse)

#Date	Date (yyyy-mm-dd)	Date d'achat (clé primaire)
Mois	1..12	Mois d'achat
Trimestre	1..4	Trimestre d'achat
Semaine	1..52	Semaine d'achat
JDS	Lundi..Dimanche	Jour de la semaine d'achat

Description de la dimension Date (DataWarehouse)

Abréviations



BD : Base de Données

CSV : Comma Separated Values

OS : Operating Système (Système d'Exploitation)

SGBD : Système de Gestion de Bases de Données

SGBDR : Système de Gestion de Bases de Données Relationnelles

XML : eXtensible Markup Language

Bibliographie



Kimball R., Ross M., Thornthwaite W., Mundy J., Becker B. (2008, 1998). *The Data Warehouse Lifecycle Toolkit*. Wiley Publishing, second edition.

Kimball R., Ross M. (2008, 2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. Wiley Publishing, second edition.

Kimball R., Ross M. (2003). *Entrepôts de données : guide pratique de modélisation dimensionnelle*. Vuibert.