Langage SQL
&
Algèbre relationnelle

info@orditech.be

SQL et Algèbre relationnelle

- **La gestion des données**
- **Le modèle relationnel**
- **L'algèbre relationnelle**
- **Généralités sur le SQL**
- **Description des objets**
- **Manipulation des données**
- **Traduction de l'algèbre relationnelle**
- **SQL avancé**

NB: ces slides se basent sur un SGBDR Oracle

info@orditech.be

La gestion des données

A. Généralités sur les fichiers

B. Organisations classiques de fichiers

1. Séquentiel
2. Séquentiel indexé
3. Bases navigationnelles

SQL

info@orditech.be

La gestion des données

Généralités sur les fichiers

Un fichier informatique est un ensemble d'informations de même nature (texte, code exécutable, données, …) portant un nom et situé sur un support physique (disque dur, disquette, bande, CD-Rom, …) et peut avoir différentes fonctions :

- **Fichier programme**
- **Fichier texte**
- **Bibliothèque de fonctions**
- **Fichier périphérique (Unix)**
- **Fichier de données**

SQL

info@orditech.be

La gestion des données

Organisations classiques de fichiers

1. **Séquentiel**
2. **Séquentiel indexé**
3. **Bases navigationnelles**

SQL

info@orditech.be

Organisations classiques de fichiers

1. Séquentiel

Le principe de cette organisation est de gérer les enregistrements comme des suites d'octets structurées (par des caractères délimiteurs ou de tailles fixes).

Organisations classiques de fichiers

2. Séquentiel indexé

C'est une amélioration de l'organisation séquentielle, par ajout d'un fichier de clés (ou d'index) lié au fichier séquentiel.

info@orditech.be

Organisations classiques de fichiers

3. Bases navigationnelles

Ce sont des collections de fichiers, logiquement appareillés entre eux. Ces bases ont été créées sur des systèmes propriétaires afin de compenser la faiblesse des organisations précédentes, en ce qui concerne la traduction du dictionnaire des données de l'analyse.

info@orditech.be

Le modèle relationnel

Le modèle relationnel repose sur des concepts de bases simples (domaine, relation, attribut) auxquels s'appliquent des règles précises. La mise en œuvre est facilitée par un langage assertionnel (non procédural) simple, basé sur une logique ensembliste.

A. Concepts et définitions

1. Domaine
2. Produit cartésien
3. Relation

B. Principales règles

SQL

info@orditech.be

Le modèle relationnel

Concepts et définitions

1. Domaine

C'est un ensemble de valeurs caractérisées par un nom.

Cardinal : c'est le nombre d'éléments d'un domaine.

SQL

info@orditech.be

Concepts et définitions

2. Produit cartésien

Le produit cartésien **P** entre plusieurs domaines **D1, D2, … , Dn** noté **P** = **D1 X, D2 X, … X**. **Dn** est l'ensemble des n-uplets (tuples) (**d1, d2, …, dn**) où chaque **di** est un élément du domaine **Di**.

Le modèle relationnel

Concepts et définitions

3. Relation

Une relation définie sur les domaines **D1, D2, … , Dn** est un sous-ensemble du produit cartésien de ces domaines caractérisé par un nom.

Attribut : C'est une colonne d'une relation caractérisée par un nom.

Degré : C'est le nombre d'attributs d'une relation.

SQL

info@orditech.be

Le modèle relationnel

Principales règles

Le modèle relationnel gère donc un objet principal, la relation, associée aux concepts de domaine et d'attribut. Des règles s'appliquent à cette relation afin de respecter les contraintes liées à l'analyse.

Quelques-unes de ces règles sont :

Cohérence : Toute valeur prise par un attribut doit appartenir au domaine sur lequel il est défini

Unicité : Tout les éléments d'une relation doivent être distincts.

Identifiant : Attribut ou ensemble d'attributs permettant de caractériser de manière unique chaque élément de la relation.

Clé primaire : Identifiant minimum d'une relation.

info@orditech.be

Le modèle relationnel

Principales règles

Quelques-unes de ces règles sont (suite) :

Clés secondaires : Autres identifiants de la relation.

Intégrité référentielle : Cette règle impose qu'un attribut ou un ensemble d'attributs d'une relation apparaisse comme clé primaire dans une autre relation.

Clé étrangère : Attribut ou ensemble d'attributs vérifiant la règle d'intégrité référentielle.

Valeur nulle : Dans le model relationnel, la notion de nullité est admise. C'est une valeur représentant une information inconnue ou inapplicable NULL

Contrainte d'entité : Toute valeur participant à une clé primaire doit être non NULL.

info@orditech.be

L'algèbre relationnelle

A. Opérateurs

1. Union
2. Intersection
3. Différence
4. Restriction
5. Projection
6. Produit cartésien
7. Jointures
8. Calculs élémentaires
9. Calcul d'agrégats

B. Etapes de résolution d'un problème

1. Analyser le besoin
2. Etablir la « vue »
3. Ordonnancer et exprimer les opérations

info@orditech.be

L'algèbre relationnelle

Opérateurs

1. Union

L'union entre deux relations de même structure (degré et domaines) donne une table résultante de même structure ayant comme éléments l'ensemble des éléments distincts des deux relations initiales.

Notation : Rx = R1 ∪ R2

SQL

info@orditech.be

Opérateurs

2. Intersection

L'intersection entre deux relations de même structure (degré et domaines) donne une table résultante de même structure ayant comme éléments l'ensemble des éléments communs aux deux relations initiales.

Notation : Rx = R1 ∩ R2

L'algèbre relationnelle

Opérateurs

3. Différence

La différence entre deux relations de même structure (degré et domaines) donne une table résultante de même structure ayant comme éléments l'ensemble des éléments de la première relation qui ne sont pas dans la deuxième.

Notation : Rx = R1 - R2

info@orditech.be

L'algèbre relationnelle

Opérateurs

4. Restriction

La restriction selon une condition produit, a partir d'une relation, une relation de même schéma n'ayant que des éléments de la relation initiale qui répondent à la condition.

Notation : Rx = σ (condition) R1

info@orditech.be

L'algèbre relationnelle

Opérateurs

5. Projection

La projection d'une relation sur un groupe d'attributs donne une relation résultante ayant comme schéma uniquement ces attributs, et comme éléments les n-uplets distincts composés par les valeurs associées de ces attributs.

Notation : Rx = π R (A1, A2, … ,An).

SQL

info@orditech.be

Opérateurs

6. Produit cartésien

Le produit cartésien entre deux relations produit une relation ayant comme schéma tous les attributs des deux relations existantes et comme éléments l'association de chaque ligne de la première table avec chaque ligne de la deuxième.

Notation : Rx = S1 X S2

L'algèbre relationnelle

Opérateurs

7. Jointures

La jointure entre deux relations selon une condition est produite par la restriction sur le produit cartésien.

Notation : Rx = S1 JOIN (condition) R2

SQL

info@orditech.be

Opérateurs

8. Calculs élémentaires

Projection sur une relation associée à un calcul portant sur chaque ligne pour créer un ou plusieurs nouveaux attributs.

Notation : Rx = π S (A1, … , N1 = expression calculée …)

info@orditech.be

L'algèbre relationnelle

Opérateurs

9. Calcul d'agrégats

Projection sur une relation associée à un ou des calculs statistiques portant sur un attribut pour tous les éléments de la relation ou du regroupement lié à la projection afin de créer un ou plusieurs nouveaux attributs.

Notation : Rx = π S (A1, … , N1 = fonction statistique (Ax) …)

SQL

info@orditech.be

L'algèbre relationnelle

Etape de résolution d'un problème

1. **Analyser le besoin**

- Transcrire sous forme de relation résultante des besoins exprimés par le prescripteur.
- Déterminer les attributs et les relations à utiliser.
- Exprimer les calculs élémentaires et d'agrégats pour créer les

info@orditech.be

L'algèbre relationnelle

Etape de résolution d'un problème

2. Etablir la « vue »

La vue est une relation intermédiaire contenant tous les attributs permettant de réaliser l'extraction, avec leur relations d'origines, leurs classes d'utilité, et les opérations à appliquer.

Classes d'attribut:

- **Classe a** : attribut participant à la relation résultante.
- **Classe b** : attribut participant à un calcul.
- **Classe c** : attribut participant à une restriction.
- **Classe d** : attribut participant à une jointure.

info@orditech.be

L'algèbre relationnelle

Etape de résolution d'un problème

3. Ordonnancer et exprimer les opérations

D'une façon générale, et sauf exception, l'ordonnancement des opérations peut s'exprimer de la façon suivante :

1. Relation concernées.
2. Restrictions (pour éliminer les lignes inutiles).
3. Jointures, Produits cartésiens, Unions, intersections, différences (pour associer les lignes restantes).
4. Calcul élémentaires (pour créer les nouvelles colonnes).

info@orditech.be

L'algèbre relationnelle

Etape de résolution d'un problème

3. Ordonnancer et exprimer les opérations (suite)

5. Calcul d'agrégats (pour les colonnes statiques).
6. Jointure entre la table obtenue en 5 et la table initiale en 4 (pour ajouter les colonnes statistiques aux autres)..
7. Répéter les étapes du 5 pour les autres regroupements.
8. Restrictions par rapport aux attributs calculés.
9. Projections pour éliminer les doublons.
10. Projection finale pour éliminer les attributs inutiles dans la table résultante.

Généralités sur le SQL

A. Composants de la base logique : objets SQL

1. La gestion des données
2. Le stockage physique
3. Le stockage d'instructions
4. La gestion des utilisateurs
5. Dénomination des objets

B. Catégories d'instructions

1. DDL (Data Definition Language)
2. DML (Data Manipulation Language)
3. Transaction Control Language
4. Session Control Language
5. Embedded SQL

info@orditech.be

Généralités sur le SQL

Le SQL (Strucrured Query Language) est un langage de requête utilisé pour la manipulation des bases de données relationnelles.

L'intérêt du SQL réside dans les caractéristiques suivantes :

Normalisation : Il implémente le modèle relationnel, et les principaux organismes de normalisation le décrivent.

Standard : Du fait de cette normalisation, la plupart des éditeurs de SGBDR, intègrent le SQL à leurs produits (MySQL, INFORMIX, DB2, MS SQL Server, SYBASE, …)

Non procédural : Le SQL est un langage de requête qui permet à l'utilisateur de demander un résultat sans se préoccuper des moyens techniques pour trouver ce résultat.

Universel : Le SQL peut être utilisé à tous niveaux dans la gestion d'une base de données (*administration système, administration de la base, développement et application, gestion de données simple*). Tous les utilisateurs de la base ont donc un langage commun.

info@orditech.be

Composants de la base logique : objets SQL

1. La gestion des données

table : Ensemble de lignes (row) et de colonnes (column).

index : Colonne ou ensemble de colonnes permettant l'accélération des recherches.

view : Requête pouvant être manipulée comme une table (table virtuelle).

synonym : Nom alternatif pour une table ou une view.

sequence : Générateur de série de nombres.

snapshots : Table contenant le résultat d'une requête faite sur une table gérée dans une base distante.

database links : Lien avec des bases distantes.

info@orditech.be

Composants de la base logique : objets SQL

2. Le stockage physique

cluster : Regroupement physique de tables ayant des colonnes communes.

tablespace : Regroupement logique de fichiers.

directory : Représentation dans la base de données d'un répertoire du système d'exploitation hôte.

info@orditech.be

Composants de la base logique : objets SQL

3. Le stockage d'instructions

schema : Ensemble des objets de la base logique appartenant à un même utilisateur.

procedure : Ensemble de code procédural nommé.

function : Ensemble de code procédural nommé retournant une valeur.

database trigger : Ensemble de code procédural associé à une table.

packages : Collection d'objets (procédure, fonction, …) stockés ensemble.

library : Représentation d 'une collection de procédure externes à Oracle, stockées dans des bibliothèques partagées du système hôte.

info@orditech.be

Composants de la base logique : objets SQL

4. Le gestion des utilisateurs

profile : Ensemble nommé de limites système.
role : Ensemble de privilèges pouvant être attribués à des utilisateurs.
user : Utilisateur pouvant se connecter et accéder aux ressources de la base.

Composants de la base logique : objets SQL

5. Dénomination des objets

Les objets créés par les utilisateurs doivent être nommés.

- 1 à 30 caractères, sauf database (8) et database links (128).
- Pas de distinction majuscule-minuscule.
- Commencent par une lettre.
- Caractères alphanumérique plus _ $ et # (plus @ et . Pour les database links).
- Ne doit pas être un mot réservé.
- Doit être unique, à l'intérieur du SCHEMA d'un utilisateur, même pour des types d'objets différents.

Convention : Les noms utilisés doivent être significatifs, sans être trop longs. Utiliser des noms identiques pour désigner les mêmes entités dans des objets différents (nom de colonne)

Catégories d'instructions

1. DDL (Data Definition language)

Il permet de gérer les « structures » des objets :

- Créer, modifier, supprimer des objets.
- Autoriser ou interdire l'accès aux données.
- Activer, désactiver l'audit.
- Commenter le dictionnaire des données.

Les instructions du **DDL** sont : **CREATE, ALTER, DROP, GRANT, REVOKE, AUDIT, NOAUDIT, ANALYZE, RENAME, TRUNCATE**.

info@orditech.be

Catégories d'instructions

2. DML (Data Manipulation Language)

Il permet la gestion des données des objets existants :

- Ajout, suppression, modification des lignes.
- Visualisation du contenu des tables.
- Verrouillage des tables.

Les instructions du **DML** sont : **INSERT, UPDATE, DELETE, SELECT, EXPLAIN PLAN, LOCK TABLE.**

info@orditech.be

Catégories d'instructions

3. Transaction Control Language

Il gère les modification faites par le DML :

- Caractéristiques des transactions.
- Validation, annulation des modifications.

Les instructions du Transaction Control sont : **COMMIT, SAVEPOINT, ROLL-BACK, SET TRANSACTION.**

info@orditech.be

Catégories d'instructions

4. Session Control Language

Il permet la gestion d'une session utilisateur :

- Modification des caractéristiques de session.
- Activation, désactivation des privilèges utilisateurs.

Les instructions du Session Control sont : **ALTER SESSION, SET ROLE.**

Catégories d'instructions

5. Embedded SQL

Il permet d'intégrer le DDL, le DML, et le Transaction Control à un langage de programmation:

- Déclarations d'objets ou d'instructions.
- Exécutions d'instructions.
- Gestions des variables et des cursors.
- Traitement des erreurs.

Les instructions du Embedded **SQL** sont : **DECLARE, TYPE, DESCRIBE, VAR, CONNECT, PREPARE, EXECUTE, OPEN, FETCH, CLOSE, WHENEVER**.

info@orditech.be

Description des objets (1/2)

A. **Les types de données**
B. **Création d'une table**
    1. Contraintes de colonne
    2. Contraintes de table (portant sur plusieurs colonnes)
    3. Complément sur les contraintes
    4. Dénomination des contraintes
C. **Suppression d'une table**

SQL

info@orditech.be

Description des objets (2/2)

D. Modification d'une table

1. Ajout ou modification de colonnes
2. Ajout d'une contrainte de table
3. Suppression d'une contrainte
4. Activation, désactivation d'une contrainte
5. Modification d'une contrainte
6. Suppression de colonnes
7. Changement de nom d'une table

E. Restauration d'une table

F. Gestion des index

1. Création d'un index
2. Suppression d'un index

info@orditech.be

Description des objets

Les types de données

CHAR (n) : Chaine de caractères de longueur fixe : n octets complétés à droite par des espaces (n <= 2000).

VARCHAR2 (n) : Chaine de caractères de longueur variable : n octets au maximum (n <= 4000).

NCHAR (n) : Chaine de caractères de longueur fixe : n octets complétés à droite par des espaces (n <= 2000). Les caractères sont codés suivant le jeu de caractères national actif.

NVARCHAR2 (n) : Chaine de caractères de longueur variable : n octets au maximum (n <= 4000). Les caractères sont codés suivant le jeu de caractères national actif.

NUMBER (p,s) : Numérique avec une précision de p chiffres dont s décimales avec (1 <= p <= 38 et -84 <= s <= + 127).

info@orditech.be

Description des objets

Les types de données (suite)

DATE : Date comprise rntre le 1er janvier 4712 avant JC et le 31 décembre 9999 après JC.

TIMESTAMP (p) : Données de type date (Année, mois, jour, heure, minute et seconde) dans laquelle il est possible de préciser, à l'aide de la précision p, le nombre de chiffres significatifs pour les fractions de secondes. Par défaut ce nombre est 6.

TIMESTAMP (p) WITH TIME ZONE : Données de type TIMESTAMP avec décalage horaire.

ITIMESTAMP (p) WITH LOCAL TIME ZONE : Données de type TIMESTAMP WITH TIME ZONE qui sont stockées sur le serveur en tenant compte de la plage horaire du serveur, mais ces données sont affichées sur le poste client en tenant compte de la zone horaire définie au niveau de la session.

Description des objets

Les types de données (suite)

BLOB : Données binaires non structurée (4 Go maximum).

CLOB : Chaine de caractères de longueur variable (4 Go maximum). Ne peut contenir que des caractères codés sur 1 octet.

NCLOB : Chaine de caractères de longueur variable (4 Go maximum). Prend en charge les jeux des caractères de longueur variable.

BFILE : Données binaires stockées dans des fichiers extérieurs à la base de données (4 Go maximum).

SQL

info@orditech.be

Description des objets

Les types de données (suite)

LONG : Chaine de caractères de longueur variable (2 Go maximum).

RAW (n) : Données binaires de longueur variable (n octets maximum ; n <= 2000). Contenu non interprété par le moteur de base de données.

LONG RAW (n) : Données binaires de longueur variable (4 Go maximum). Contenu non interprété par le moteur de base de données.

Description des objets

Création d'une table

Une table est un ensemble de ligne et de colonnes. La création consiste à définir (en fonction de l'analyse) le nom de ces colonnes, leur format d'utilisation (type), une valeur par défaut à la création de la ligne (default), les règles de gestion s'appliquant à la colonne (CONSTRAINT). Si une règle de gestion concerne plusieurs colonnes de la ligne, on définit une contrainte de table. Il est possible de définir jusqu'à 1000 colonnes pour chaque table.

Syntaxe :

```
CREATE TABLE nom (nom_colonne type [DEFAULT expr]
[[ CONSTRAINT nom contrainte_colonne … ], …
[ , CONSTRAINT nom contrainte_table … ]);
```

info@orditech.be

Description des objets

Création d'une table

1. Contraintes de colonnes

NULL / NOT NULL : Autorise (NULL) ou interdit (NOT NULL) l'insertion de valeur NULL pour cet attribut.

PRIMARY KEY : Désigne l'attribut comme clé primaire de la table. Cette contrainte ne peut apparaitre qu'une seul fois dans l'instruction.

UNIQUE : Désigne l'attribut comme clé secondaire de la table. Dans le cas de contrainte UNIQUE portant sur une seule colonne, l'attribut peut prendre la valeur NULL. Cette contrainte peut apparaitre plusieurs fois dans l'instruction.

REFERENCES table [(colonne)] [ON DELETE CASCADE] : Contrainte d'intégrité référentielle pour l'attribut dans la table détail en cours de définition. Les valeurs prises par cet attribut doivent exister dans l'attribut colonne qui possède une contrainte PRIMARY KEY dans la table maitre.

CHECK (condition) : Vérifie lors de l'insertion de lignes que l'attribut réalise la condition.

info@orditech.be

Description des objets

Création d'une table

2. Contraintes de table (portant sur plusieurs colonnes)

PRIMARY KEY (colonne, ...) : Désigne l'attribut comme clé primaire de la table. Cette contrainte ne peut apparaitre qu'une seul fois dans l'instruction.

UNIQUE (colonne, ...) : Désigne l'attribut comme clé secondaire de la table. Dans le cas de contrainte UNIQUE portant sur une seule colonne, l'attribut peut prendre la valeur NULL. Cette contrainte peut apparaitre plusieurs fois dans l'instruction.

FOREIGN KEY (colonne, ...) REFERENCES table [(colonne)] [ON DELETE {CASCADE | SET NULL}] : Contrainte d'intégrité référentielle pour l'ensemble des attributs dans la table détail en cours de définition. Les valeurs prises par ces attributs doivent exister dans l'attribut colonne qui possède une contrainte PRIMARY KEY ou UNIQUE dans la table maitre table.

CHECK (condition) : Cette contrainte permet d'exprimer une condition qui doit exister entre plusieurs attributs de la ligne.

info@orditech.be

Description des objets

Création d'une table

3. Complément sur les contraintes

ON DELETE CASCADE : Demande la suppression des lignes dépendantes dans la tables en cours de définition, si la ligne contenant la clé primaire correspondante dans la table maitre est supprimée. Si cette option n'est pas indiquée, la suppression sera impossible dans la table mitre s'il existe des lignes référençant cette valeur de clé primaire.

ON DELETE SET NULL : Demande la mise à NULL des colonnes constituant la clé étrangère qui font référence à la ligne supprimée. Si cette option n'est pas indiquée, la suppression sera impossible dans la table maitre s'il existe des lignes référençant cette valeur de clé primaire.

[NOT] DEFERRABLE : Repousse ou non (NOT) la vérification de la contrainte au moment de la validation de la transaction.

info@orditech.be

Création d'une table

4. Complément sur les contraintes

Les contraintes peuvent être nommées afin d'être plus facilement manipulées ultérieurement (activation, suppression). Dans le cas où aucun nom n'est affecté explicitement à une contrainte, le moteur de base de données génère automatiquement un nom de la forme SYS_Cn (n est un nombre entier unique).

Mnémonique associé au type de contrainte :

PK Clé primaire.

UQ Unique.

NN Not NULL.

CK Check.

RF Référence.

FK Clé étrangère.

Description des objets

Suppression d'une table

Supprimer une table revient à éliminer sa structure et toute les données qu'elle contient ; les index associé sont également supprimé ; les vues construites directement ou indirectement sur cette table sont désactivées automatiquement par le moteur de base de données.

Si la clé primaire de la table est référencée dans d'autres tables par des contraintes REFERENCES ou FOREIGN KEY, la clause CASCADE CONSTRAINTS permet de supprimer ces contraintes d'intégrité référentielle dans les tables « enfants ».

La clause PURGE permet de supprimer la table et de libérer l'espace physique occupé par cette table. En l'absence de cette clause, la table est supprimée logiquement mais pas physiquement. Il est alors possible d'utiliser l'instruction FLASHBACK TABLE pour retrouver la table et ses données au moment de la suppression.

Syntaxe :

```
DROP TABLE nom [ CASCADE CONSTRAINTS ] [ PURGE ]
```

info@orditech.be

Description des objets

Modification d'une table

Il est possible de modifier la structure d'une table à plusieurs niveaux.

- Ajout de colonnes (nom, type, valeur par défaut, contrainte NOT NULL).
- Ajout de contraintes de colonne (contrainte NOT NULL uniquement).
- Ajout de contraintes de table.
- Redéfinition d'une colonne (type, valeur par défaut).
- Activation, désactivation de contraintes de colonne ou de table.
- Suppression de contraintes de colonne ou de table.
- Changement de nom de table.

SQL

info@orditech.be

Description des objets

Modification d'une table

1. Ajout ou modification de colonnes

Syntaxe :

```
ALTER TABLE nom { ADD / MODIFY } ([colonne type [ contrainte ]
, … ])
```

SQL

info@orditech.be

Modification d'une table

2. Ajout d'une contrainte de table

Syntaxe :

```
ALTER TABLE nom_table ADD [ CONSTRAINT nom ] contrainte
```

La syntaxe de déclaration de contrainte est identique à celle vue lors de la création de la table.

Description des objets

Modification d'une table

3. Suppression d'une contrainte

Syntaxe :

```
ALTER TABLE nom_table DROP { PRIMARY KEY / UNIQUE (colonne) /
CONSTRAINT } nom
```

SQL

info@orditech.be

Modification d'une table

4. Activation, désactivation d'une contrainte

La commande ALTER TABLE permet également d'activer et de désactiver les contraintes d'intégrité. Cette opération peut être intéressante lors d'un import massif de données afin, par exemple, de limiter le temps nécessaire à cette importation.

Syntaxe :

```
ALTER TABLE nom_table DROP { ENABLE VALIDATE / ENABLE
NOVALIDATE / DISABLE} nom_contrainte
```

Modification d'une table

5. Modification d'une contrainte

Il n'est pas possible de modifier la définition d'une contrainte. Par contre, il est tout à fait possible de modifier l'état d'une contrainte.

Syntaxe :

```
ALTER TABLE nom_table MODIFY CONSTRAINT nom_contrainte
etat_contrainte
```

Modification d'une table

5. Modification d'une contrainte (suite)

Les états possibles pour la contrainte sont:

DEFERRABLE : La validation de la contrainte peut être reportée à la fin de la transaction.

NOT DEFERRABLE : Pour chaque nouvelle transaction, le fonctionnement par défaut consiste à vérifier la contrainte à la fin de chaque ordre du DML. Cette option est sélectionnée par défaut.

INITIALLY IMMEDIATE : Pour chaque nouvelle transaction, le fonctionnement par défaut consiste à vérifier la contrainte à la fin de chaque ordre du DML. Cette option est sélectionnée par défaut.

info@orditech.be

Modification d'une table

5. Modification d'une contrainte (suite)

INITIALLY DEFERRED : Implique que la contrainte est en mode DEFERRABLE et précise que par défaut cette contrainte est vérifiée uniquement à la fin de la transaction.

RELY ou **NORELY** : Permet de préciser si lors de l'activation d'une contrainte, elle doit être vérifiée sur les données déjà présentes dans la table. Par défaut c'est le mode NORELY qui est choisi. Ce mode stipule que la contrainte doit être vérifiée par toutes les lignes lors de son activation. Le mode RELY permet quant à lui de réactiver une contrainte sans s'assurer au préalable que les informations contenues dans la tables vérifient la contrainte.

USING INDEX : Permet de préciser les paramètres des index utilisés pour mettre en place les contraintes de clés primaires et d'unicité.

Modification d'une table

6. Suppression de colonnes

Il est possible de supprimer une colonne en utilisant la clause DROP COLUMN de l'instruction ALTER TABLE. Cette opération permet de récupérer l'espace disqueoccupé par chaque colonne supprimée.

Syntaxe :

```
ALTER TABLE nom_table DROP COLUMN nom_colonne
```

ou

```
ALTER TABLE nom_table DROP (nom_colonne, nom_colonne [,…])
```

info@orditech.be

Modification d'une table

6. Suppression de colonnes (suite)

Une suppression de colonne dans une table existante et possédant de nombreuses lignes peut s'avérer une opération très longue. Il peut parfois être plus opportun de rendre la colonne inutilisable à l'aide de la clause SET UNUSED. Cette option ne permet pas de libérer l'espace disque occupé par la colonne, mais elle permet de planifier l'opération de suppression de la colonne à un moment où la charge de travail pour le serveur est moindre.

Syntaxe :

```
ALTER TABLE nom_table SET UNUSED (nom_colonne[,…]);
```

ou

```
ALTER TABLE nom_table DROP UNUSED COLUMNS [CHECKPOINT
nombre_lignes];
```

info@orditech.be

Modification d'une table

7. Changement de nom d'une table

L'instruction RENAME permet de renommer une table mais également les vues, les séquences et les synonymes privés.

> *Les synonymes publics ne peuvent pas être renommés et doivent être supprimés puis recréés.*

Syntaxe :

```
RENAME ancien_nom TO nouveau_nom;
```

Description des objets

Restauration d'une table

L'instruction FLASHBACK TABLE permet de restaurer de façon automatique une table modifiée de façon accidentelle. Le laps de temps pendant lequel il est possible de revenir à la version précédente de la table est fonction de l'espace d'annulation réservé à la base.

Syntaxe :

```
FLASHBACK TABLE nom_table TO BEFORE DROP;
```

SQL

info@orditech.be

Description des objets

Gestion des index

Les index ont pour but l'amélioration des performances des requêtes. Ils sont utilisés implicitement par l'optimiseur de requêtes SQL et sont mis à jour automatiquement avec les lignes.

On crée des index, de manière générale, sur toute les clés étrangères et sur les critères de recherche courants.

Les index concernant les clés primaires et secondaires (index UNIQUE) sont créés automatiquement au CREATE TABLE avec comme nom, le nom de la contrainte. Pour les autres index, il faut utiliser le CREATE INDEX.

> *La mise en place des contraintes de clés primaires et d'unicité passe par la création implicite d'un index.*

info@orditech.be

Description des objets

Gestion des index

1. Création d'un index

Syntaxe :

```
CREATE INDEX nom ON (colonne [DESC][,…]);
```

Gestion des index

2. Suppression d'un index

Syntaxe :

```
DROP INDEX nom;
```

Manipulation des données (1/2)

A. **Les instructions**
    1. Expressions
    2. Opérateurs
    3. Conditions
    4. Fonctions
B. **Création de lignes**
C. **Suppression de lignes**
D. **Modification de lignes**
E. **Extraction de données**

SQL

info@orditech.be

Manipulation des données (2/2)

F. Contrôle des transactions

1. Validation de la transaction
2. Annulation des modifications
3. Déclaration d'un point de contrôle
4. Accès simultané aux données
5. Vérification des contraintes en fin de transaction

info@orditech.be

Les instructions

Les instructions SQL manipulent des expressions. Ces expressions font référence à des noms d'objets de la base, à des constantes, comportent des appels à des fonctions standardisées et composent ces éléments avec des opérateurs.

Des expressions logiques (conditions) permettent également de définir la portée des instructions.

info@orditech.be

Les instructions

1. Expressions

Les termes des expressions peuvent être:

- Constantes caractères.
- Constantes littérales date.
- Constantes numériques.
- Noms d'attribut de table.
- Fonctions
- Pseudo-colonnes.

info@orditech.be

Les instructions

2. Opérateurs

- Arithmétique.
- Concaténation.

info@orditech.be

Les instructions

3. Conditions

Les conditions mettent en jeu des expressions, des opérateurs de comparaison et des opérateurs logiques.

Manipulation des données

Les instructions

3. Conditions (suite)

Opérateurs de comparaison.

- Comparaison simple.
  = != <> < <= > >=
- Appartenance à un ensemble de valeurs.
  expression1 IN (expression2, ...)
- Appartenance à un intervalle de valeurs.
  expression1 BETWEEN expression2 AND expression3
- Comparaison par rapport à un format de chaine de caractères.
  expression1 LIKE 'format'

info@orditech.be

Les instructions

3. Conditions (suite)

Opérateurs logiques.

- Négation d'une expression logique.
  NOT expression
- Combinaison d'expressions logiques.
  expression1 (AND / OR) expression2

info@orditech.be

Les instructions

4. Fonctions

Il existe deux types de fonctions :

- **Les fonctions scalaires** qui portent sur une seule ligne : la fonction est exécutée et retourne un résultat pour chaque ligne de la requête.
- **Les fonctions sur un regroupement de lignes** : la fonction est exécutée une fois et retourne un résultat pour un ensemble de lignes de la requête. Ces fonctions sont appelées fonctions d'agrégat.

info@orditech.be

Manipulation des données

Création des lignes

L'ajout d'une ligne à la table est réalisé si les cpontraintes sont respectées. Si le nom des colonnes à valoriser ne sont pas cités, une expression doit être donnée pour chaque colonne dans l'ordre des définitions de colonne faites lors de la création de la table. Aucune colonnes ne peut être omise.

Syntaxe :

```
INSERT INTO table [(colonne,…)] VALUE (expression,…);
```

ou

```
INSERT INTO table [(colonne,…)] SELECT colonne,… FROM table … ;
```

info@orditech.be

Manipulation des données

Suppression de lignes

L'instruction DELETE supprime toute les lignes d'une table. Si la classe WHERE est utilisée, seules les lignes pour lesquelles la condition est vraie sont supprimées.

L'instruction TRUNCATE permet de supprimer toutes les lignes d'une table et de reprendre les conditions de stockage adoptées lors de la création de la table.

Syntaxe :

```
DELETE FROM table [WHERE condition);
```

ou

```
TRUNCATE TABLE table;
```

info@orditech.be

Manipulation des données

Modification de lignes

L'instruction UPDATE permet de remplacer dans une table, la valeur des colonnes spécifiées par des expressions. Si aucune clause WHERE n'est spécifiée, la mise à jour est réalisée pour toutes les lignes de la table. Dans le cas contraire, seules les lignes pour lesquelles la condition spécifiée dans la clause WHERE est vérifiée sont mises à jour.

Syntaxe :

```
UPDATE table SET colonne = expression,… [WHERE condition);
```

SQL

info@orditech.be

Manipulation des données

Extraction des données

L'instruction SELECT permet d'afficher les données d'une table, vue ou synonyme.

Syntaxe

```
SELECT {* | expression,… } FROM table [WHERE condition);
```

SQL

info@orditech.be

Manipulation des données

Contrôle de transactions

Une transaction est un ensemble d'instructions du DML (INSERT, UPDATE, DELETE) exécutées entre deux commande CONNECT, COMMIT, et ROLLBACK.

La transaction permet de s'assurer que l'ensemble des instructions inclus dans celle-ci sera réalisé dans sa totalité ou pas du tout.

Contrôle de transactions

1. Validation de la transaction

Les modifications de lignes depuis le dernier COMMIT ou CONNECT deviennent définitives.

Syntaxe

```
COMMIT;
```

info@orditech.be

Contrôle de transactions

2. Annulation des modifications

Les modifications de lignes faites depuis le dernier COMMIT CONNECT ou le SAVEPOINT mentionné sont annulées.

Si le ROLLBACK indique un SAVEPOINT, la transaction en cours est toujours active après l'exécution de l'instruction.

Syntaxe

```
ROLLBACK (TO savepoint);
```

Contrôle de transactions

3. Déclaration d'un point de contrôle

Un point de contrôle permet de mémoriser un état de données au cours de la transaction. La pose d'un SAVEPOINT permet d'annuler les modifications faites après la pose de celui-ci.

La transaction en cours est toujours active après la pose du SAVEPOINT.

Syntaxe

```
SAVEPOINT savepoint;
```

Contrôle de transactions

4. Accès simultané aux données

Lors d'accès aux mêmes données (tables) à partir de plusieurs transactions, SQL garantit la cohérence des données dans la transaction par rapport à la lecture initiale.

Tant qu'une transaction n'est pas validée par une instruction COMMIT, les modifications faites sur les données à l'intérieur de la transaction sont invisibles des autres utilisateurs.

Contrôle de transactions

5. Vérification des contraintes

Il peut être nécessaire de vérifier de vérifier les contraintes d'intégrité en fin de transaction. Il est nécessaire de préciser lors de la mise en place de la contrainte d'intégrité, que sa vérification est différée en fin de transaction (INITIALLY DEFERRED) ou bien si au contraire il est possible de demander dans certains cas la validation de la contrainte en fin de transaction (DEFERRABLE). Par défaut les contraintes d'intégrité sont vérifiées dès la mise en place des valeurs dans les tables et il n'est pas possible de déplacer cette vérification en fin de transaction.

Traduction de l'algèbre relationnelle (1/2)

A. Opérations

1. Restriction
2. Calculs élémentaires
3. Projection
4. Calcul d'agrégats
5. Fonctions de GROUPE
6. Fonctions analytiques
7. Restrictions sur agrégats
8. Produits cartésien
9. Jointures
10. Jointures externes
11. Union, intersection, différences

info@orditech.be

Traduction de l'algèbre relationnelle (2/2)

B. Traitement du résultat

1. Le tri
2. La sauvegarde
3. Enumérer toutes les possibilités d'un calcul d'agrégats

C. MERGE

info@orditech.be

Traduction de l'algèbre relationnelle

La méthode de l'algèbre relationnelle permet de résoudre des extractions de données en créant des tables intermédiaires par l'utilisation d'opérateurs (UNION, RESTRICTION, JOINTURE, etc ...).

Cette méthode peut être traduite en SQL grâce à l'instruction SELECT qui permet toutes les opérations par ses différentes clauses (WHERE, GROUP BY, UNION, etc ...), et par les instructions CREATE et INSERT qui permettent la gestion des tables intermédiaires.

info@orditech.be

Traduction de l'algèbre relationnelle

Opérations

1. Restriction

La restriction permet de n'obtenir que les lignes répondant à une condition.
L'opération σ s (cond) se traduit par :.

Syntaxe

```
SELECT * FROM S WHERE cond;
```

info@orditech.be

Traduction de l'algèbre relationnelle

Opérations

2. Calculs élémentaires

Le calcul élémentaire permet d'obtenir des colonnes calculées pour chaque ligne.
L'opération π S (col, …, nvcol = exp) se traduit par :

Syntaxe

```
SELECT col,…, expression FROM S;
```

info@orditech.be

Traduction de l'algèbre relationnelle

Opérations

3. Projection

La projection a pour but d'éliminer les colonnes inutiles. Elle se fait en SQL en ne citant que les colonnes voulues dans le SELECT.

Syntaxe

```
SELECT liste de colonnes FROM table GROUP BY liste de colonnes;
```

Ou

```
SELECT DISTINCT {* | liste de colonnes) FROM table;
```

Traduction de l'algèbre relationnelle

Opérations

4. Calculs d'agrégats

Les projections de calcul d'agrégats permettent des calculs statistiques sur des regroupements introduits par GROUP BY.

L'opération π S (col, …, nvcol = calcul statistique) se traduit par :

Syntaxe

```
SELECT liste_colonnes, fonction_groupe FROM S GROUP BY liste_colonnes;
```

SQL

info@orditech.be

Traduction de l'algèbre relationnelle

Opérations

5. Fonctions de GROUPE

La commande GROUP BY est utilisée en SQL pour grouper plusieurs résultats et utiliser une fonction de totaux sur un groupe de résultat. Sur une table qui contient toutes les ventes d'un magasin, il est par exemple possible de liste regrouper les ventes par clients identiques et d'obtenir le coût total des achats pour chaque client.

info@orditech.be

Opérations

6. Fonctions analytiques

Les fonctions analytiques permettent d'extraire des résultats à partir d'un regroupement de lignes effectué à l'aide de l'instruction GROUP BY. Ces fonctions se différencient des fonctions d'agrégat car au lieu de retourner une seule valeur pour chaque groupe de valeurs, elles retournent plusieurs lignes de résultats pour chaque groupe. Ces lignes de résultats sont appelées windows (fenêtres). Pour chaque lignes analysée une fenêtre glissante est définie. La taille de la fenêtre détermine le nombre de lignes à prendre en compte pour le calcul de la ligne actuelle.

Traduction de l'algèbre relationnelle

Opérations

7. Restrictions sur agrégats

Lorsque l'on souhaite restreindre le nombre de lignes renvoyées par une requête comportant un calcul d'agrégats, il est possible d'utiliser la clause HAVING.

Syntaxe

```
SELECT liste_colonnes, fonction_groupe FROM S GROUP BY
liste_colonnes HAVING condition;
```

SQL

info@orditech.be

Traduction de l'algèbre relationnelle

Opérations

8. Produit cartésien

Le produit cartésien permet d'obtenir l'association de chaque ligne d'une table avec chaque ligne d'autres tables.

Syntaxe

```
SELECT liste_colonnes FROM S, T;
```

Traduction de l'algèbre relationnelle

Opérations

9. Jointures

La jointure est une restriction sur le produit cartésien afin de lier chaque ligne d'une table avec des lignes d'une autre table en respectant une condition.

Syntaxe

```
SELECT liste_colonnes FROM S, T WHERE cond;
```

info@orditech.be

Traduction de l'algèbre relationnelle

Opérations

10. Jointures externes

La jointure externe (outer join) est une extension de la jointure, qui permet d'obtenir en plus des lignes répondant à la condition, les lignes de l'une des tables qui n'y répondent pas. Il suffit d'ajouter l'opérateur (+) dans la condition, derrière la colonne de la table dont les lignes ne peuvent apparaître dans le résultat.

Traduction de l'algèbre relationnelle

Opérations

11. Union, intersection, différence

Ces opérateurs permettent d'obtenir dans une requête des lignes provenant de requêtes distinctes mais de même forme (même nombre de colonnes, même type dans le même ordre).

Syntaxe

```
SELECT liste_colonnes FROM S {UNION/UNION ALL/INTERSECT/MINUS}
SELECT liste_colonnes FROM T;
```

SQL

info@orditech.be

Traduction de l'algèbre relationnelle

Traitement du résultat

2. La sauvegarde

Il est possible de sauvegarder le résultat d'une requête dans une table, afin de pouvoir l'utiliser dans une autre requête.
Deux méthodes sont utilisables:

- Créer la table puis y insérer des lignes résultant de la requête.

Syntaxe

```
CREATE TABLE nom (colonne type [DEFAULT expr][contrainte],…);
INSERT INTO nom [(colonne, colonne, …)] SELECT … ;
```

- Créer la table à partir de la requête.

Syntaxe

```
CREATE TABLE nom [(colonne, colonne, …)] AS SELECT … ;
```

info@orditech.be

Traduction de l'algèbre relationnelle

Traitement du résultat

3. Enumérer toutes les possibilités d'un calcul d'agrégats

Un calcul d'agrégats porte toujours sur le regroupement indiqué par la clause CROUP BY. Il est parfois nécessaire d'effectuer un regroupement plus large afin de connaitre d'autres valeurs. Par exemple on souhaite connaitre le montant de chaque commande et le montant de toute les commandes passées par un client. Pour mener à bien ces calculs en une seule étape, il faut utiliser les mots clés ROLLUP et CUBE.

ROLLUP : permet de faire des regroupements de plus en plus généraux.

CUBE : permet de réaliser le calcul demandé sur tout les regroupements possibles.

Syntaxe

```
SELECT liste_colonne, calcul_agrégat FROM table GROUP BY
{ROLLUP|CUBE} (liste_colonne);
```

info@orditech.be

Traduction de l'algèbre relationnelle

MERGE

Cette instruction permet de fusionner, en une seule requête, une opération de mise à jour (INSERT, UPDATE, DELETE) sur une table.

Les informations, qui servent de base à l'exécution de l'ordre DML, peuvent provenir d'une ou de plusieurs tables.

Toutefois la même ligne d'informations ne peutpas participer en tant que source et destination de l'instruction MERGE.

Syntaxe

```
MERGE hint INTO nom_table USING nom_table ON (condition) [WHEN
MATCHED THEN UPDATE SET nom_colonne = valeur [WHERE condition]
[DELETE WHERE condition][WHEN NOT MATCHED THEN INSERT
(nom_colonne, …) values (valeur, …)[WHERE condition]];
```

info@orditech.be

SQL avancé

A. **Les objets**
    1. Les objets View (vue)
    2. Les objets Schéma
    3. Les objets Synonym
    4. Les objets Séquence
B. **Les requêtes complexes**
    1. Eléments de syntaxe
    2. Les sous-requêtes
    3. Les requêtes hiérarchiques
C. **Verrouillage des tables**
D. **Les commentaires**
E. **Informations sur les objets du schéma**
F. **Fonctionnalités spécifiques**
G. **Les expressions régulières**

info@orditech.be

SQL avancé

Le SQL permet l'utilisation d'autres objets que les tables et les index, afin de gérer les données ou de manipuler de requêtes.

La puissance de l'instruction SELECT permet, d'autre part, de combiner les différentes clauses en une seule commande, et également d'imbriquer les requêtes.

Enfin, le SQL permet dans un environnement multiutilisateur, de verrouiller les tables afin de préserver l'intégrité des données.

info@orditech.be

Les objets

1. Les objets View (vue)

Les vues sont des tables virtuelles « contenant » le résultat d'un SELECT.

L'un des intérêts de l'utilisation des vues vient du fait que la vue ne stocke pas les données, mais fait référence à une ou plusieurs tables d'origine à travers une requête SELECT, requête qui est exécutée chaque fois que la vue est référencée. De ce fait, toute modification de données dans les tables d'origine est immédiatement visible dans la vue dès que celle-ci est à nouveau exécutée..

info@orditech.be

Les objets

1. Les objets View (suite)

Création :

Syntaxe

```
CREATE [OR REPLACE] [FORCE | NO FORCE ] VIEW nom [(colonnes, …)] AS SELECT … [WITH CHECK OPTION | WITH READ ONLY];
```

Les objets

1. Les objets View (suite)

Compilation :

Syntaxe

```
ALTER VIEW nom COMPILE;
```

info@orditech.be

Les objets

1. Les objets View (suite)

Suppression :

Syntaxe

```
DROP VIEW nom;
```

info@orditech.be

Les objets

2. Les objets Schema

Un schéma est un ensemble de tables, vues et privilèges regroupés sous un même nom (celui de l'utilisateur).

Syntaxe

```
CREATE SCHEMA AUTHORIZATION nom {CREATE {TABLE/VIEW} … /GRANT …
} …;
```

info@orditech.be

Les objets

3. Les objets Synonym

Un synonyme est le nom alternatif donné à un objet TABLE, VIEW, SEQUENCE, SNAPSHOT, PROCEDURE, FUNCTION ou PACKAGE.

Création :

Syntaxe

```
CREATE [PUBLIC] SYNONYM nom FOR objet;
```

Suppression :

Syntaxe

```
DROP SYNONYM nom;
```

info@orditech.be

Les objets

4. Les objets Sequence

La création d'un objet SEQUENCE met à disposition de l'utilisateur un générateur de nombres.

Syntaxe

```
CREATE SEQUENCE nom [paramètres];
ALTER SEQUENCE nom paramètres;
DROP SEQUENCE nom;
```

info@orditech.be

Les requêtes complexes

1. Eléments de syntaxe

Alias : Nom alternatif donné à une colonne ou une table dans une requête.

Syntaxe

```
SELECT colonne [AS] alias_colonne … FROM table alias_table …;
```

info@orditech.be

Les requêtes complexes

1. Eléments de syntaxe (suite)

Any : Compare suivant l'opérateur donné (=,<>,<,<=,>,>=), les valeurs des colonnes spécifiées avec chacune des valeurs de la liste. L'expression est vraie si au moins une des comparaisons est vraie. La liste des valeurs peut être une liste de constantes littérales ou des valeurs retournées par une sous-requête.

Syntaxe

```
SELECT … WHERE [(colonne, colonne, … )] opérateur ANY ({SELECT … / expression, … });
```

info@orditech.be

Les requêtes complexes

1. Eléments de syntaxe (suite)

All : Compare suivant l'opérateur donné (=,<>,<,<=,>,>=), les valeurs des colonnes spécifiées avec chacune des valeurs de la liste. L'expression est vraie si toutes les comparaisons sont vraie. La liste des valeurs peut être une liste de constantes littérales ou des valeurs retournées par une sous-requête.

Syntaxe

```
SELECT … WHERE [(colonne, colonne, … )] opérateur ALL (SELECT …
/ expression, … );
```

info@orditech.be

Les requêtes complexes

1. Eléments de syntaxe (suite)

Exists : La condition est vraie si la sous-requête retourne au moins une ligne.

Syntaxe

```
SELECT … WHERE [(colonne, colonne, … )] EXISTS (SELECT … /
expression, … );
```

info@orditech.be

Les requêtes complexes

2. Les sous-requêtes

Dans l'écriture d'une requête complexe, on distingue la requête externe de la requête interne, la sous requête.

Les sous-requêtes peuvent être divisée en deux catégories : les sous requêtes imbriquées et les sous-requêtes corrélées.

info@orditech.be

Les requêtes complexes

2. Les sous-requêtes (suite)

Sous-requêtes imbriquées

Dans une sous-requête imbriquée il n'y a pas de lien explicite entre la requête interne et la requête externe. La requête interne est exécutée une seule fois pour construire la liste de valeurs, avant l'exécution de la requête externe (quel que soit le nombre de lignes ramenées par celle-ci).

info@orditech.be

Les requêtes complexes

2. Les sous-requêtes (suite)

Sous-requêtes corrélées

Dans une sous-requête corrélée, la condition spécifiée dans la clause WHERE de la requête interne fait référence à une ou plusieurs colonnes de la requête externe. La requête interne est donc ré exécutée pour chaque ligne retournée par la requête externe.

info@orditech.be

Les requêtes complexes

3. Les requêtes hiérarchiques

Il est parfois nécessaire d'interroger les données suivant un ordre hiérarchique bien précis. Pour cela SQL met à notre disposition des commandes pour extraire les données en utilisant un ordre. Ces commandes sont CONNECT BY PRIOR et START WITH. Lors de l'exécution d'une telle requête, SQL procède de la façon suivante :

a) SQL sélectionne la racine de l'arbre, ce sont les lignes qui satisfont la condition indiquée dans la classe START WITH.

b) SQL sélectionne les enfants de chaque père. Tous doivent satisfaire la condition indiquée dans la clause CONNECT BY.

c) Tous les enfants sont sélectionnés les uns après les autres.

d) Si la requête contient une clause WHERE, SQL supprime de la hiérarchie toutes les lignes qui ne respectent pas la clause WHERE.

info@orditech.be

Informations sur les objets du schéma

Le dictionnaire de données possède de nombreuses vues qui donnent une information détaillée des différents éléments présents dans le schéma.

info@orditech.be

Fonctionnalités spécifiques

NLS : Certaines fonctions doivent réagir différemment suivant le langage sélectionné au niveau du server SQL. Au niveau SQL, tous ces critères sont fixés par les paramètres NLS (National Language Support). Ces paramètres ont une incidence principalement sur l'affichage des donnée de type date et sur l'affichage des données de type numérique.

Case : L'instruction CASE permet de mettre en place une condition d'instruction conditionnelle directement dans la requête SELECT, sans faire appel à un bloc d'instructions procédurales.

L'intérêt principal de cette instruction est de faciliter la présentation et la mise en forme des résultats directement dans les requêtes SELECT. L'instruction CASE permet également de limiter le nombre de fois où il est nécessaire de faire appel à un bloc PL/SQL pour résoudre le problème.

info@orditech.be

Les expressions régulières

Les expressions régulières représentent un outil puissant pour travailler avec les chaines de caractères. Cette fonctionnalité est déjà présente sur de nombreux langages de programmation et sous Unix.

Les requêtes d'extraction de données avec des critères très précis de sélection sur les données de type caractère vont pouvoir être écrites plus facilement.

Pour pouvoir travailler avec les expressions régulières, SQL propose un opérateur REGEXP_LIKE et trois fonctions : REGEXP_INSTR, REGEXP_SUBSTR et REGEXP_REPLACE.

Les expressions régulières vont décrire à l'aide de méta-caractères la structure que doit posséder la chaine de caractère avec laquelle on souhaite travailler.

info@orditech.be