



Ressourcesinformatiques

SQL

Les fondamentaux du langage

2^{ème} édition

Eric GODOC

Téléchargement
www.editions-eni.fr



Les éléments à télécharger sont disponibles à l'adresse suivante :

<http://www.editions-eni.fr>

Saisissez la référence ENI de l'ouvrage **RI2SQL** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Préambule

Chapitre 1

Introduction

1. Un peu d'histoire.	11
2. Les normes SQL	12
3. Description rapide du modèle relationnel	14
3.1 Principaux concepts du modèle relationnel	15
3.2 Principales règles	16
4. Les opérateurs en algèbre relationnelle	19
4.1 Union.	19
4.2 Intersection	20
4.3 Différence	21
4.4 Division	22
4.5 Restriction	23
4.6 Projection	24
4.7 Produit cartésien	24
4.8 Jointure	24
4.9 Calculs élémentaires	26
4.10 Calcul d'agrégats	27
5. Les systèmes de gestion de bases de données utilisant SQL	27

Chapitre 2

La définition des données (LDD)

1. Les types de données	29
1.1 Numériques.....	29
1.2 Caractères	30
1.3 Dates et heures	32
1.4 Les autres types de données.....	33
1.4.1 Les autres types numériques.....	33
1.4.2 Les autres types caractères	34
1.4.3 Les types binaires.....	34
2. La création de tables.....	35
2.1 L'ordre CREATE	35
2.2 Les commentaires (COMMENT)	38
2.3 Créer une table à partir d'une autre.....	40
2.4 Utilisation des synonymes.....	43
2.5 Les SEQUENCES	44
3. La suppression de tables.....	48
3.1 L'ordre DROP	48
4. La modification de tables.....	49
4.1 L'ordre ALTER.....	49
4.2 Renommer une table (RENAME)	50
5. Les vues	51
5.1 Pourquoi utiliser des vues.....	51
5.2 La création de vues	52
5.3 La suppression de vues	55
6. Les index	56
6.1 Les index et la norme SQL.....	56
6.2 Les différentes méthodes d'organisation des index	57
6.3 La création d'index	59
6.4 La suppression d'index	61

7. L'intégrité des données	62
7.1 La PRIMARY KEY	62
7.2 La FOREIGN KEY	64
7.3 Les valeurs par défaut (DEFAULT)	66
7.4 La valeur NULL	67
7.5 La clause UNIQUE	68
7.6 CHECK	69
7.7 Quelques conseils	70
7.8 Exercices d'application	71
7.9 Correction des exercices d'application.	72

Chapitre 3

La manipulation des données (LMD)

1. Introduction	77
2. La sélection de données	77
2.1 L'ordre de sélection de données SELECT	78
2.2 Les options DISTINCT et ALL	80
2.3 L'utilisation des alias	81
2.4 La clause WHERE	83
2.5 Les jointures	85
2.5.1 La jointure interne	87
2.5.2 La jointure externe	89
2.5.3 La jointure naturelle	92
2.5.4 La jointure croisée	93
2.5.5 Syntaxes des différentes formes de jointures.	95
2.6 Les tris (ORDER BY)	96
2.7 Les regroupements (GROUP BY)	98
2.8 Les fonctions utilisées lors d'un regroupement.	100
2.8.1 COUNT (compter des lignes).	101
2.8.2 SUM (additionner des valeurs).	103
2.8.3 MAX et MIN (valeurs maximum et minimum)	103
2.8.4 AVG (moyenne).	105

2.8.5	HAVING	105
2.9	L'instruction CASE	106
2.10	Résumé des syntaxes possibles du SELECT	109
2.11	Les opérateurs ensemblistes	110
2.11.1	L'opérateur UNION	110
2.11.2	L'opérateur INTERSECT	114
2.11.3	L'opérateur EXCEPT	116
2.12	Comment construire une requête : quelques conseils	117
2.13	Exercices sur la sélection de données	121
2.13.1	Questions générales	121
2.13.2	Exercices d'application	121
3.	L'insertion de données	130
3.1	L'ordre INSERT	130
3.2	L'insertion à partir d'une autre table	133
3.3	Résumé des syntaxes de l'INSERT	137
3.4	Exercices sur l'insertion de données	138
4.	La suppression de données	138
4.1	L'ordre DELETE	139
4.2	L'ordre TRUNCATE	140
4.3	Exercices sur la suppression de données	141
5.	La modification de données	141
5.1	L'ordre UPDATE	141
5.2	Exercices sur la modification de données	143
6.	Solutions des exercices	144
6.1	Solution des exercices sur la sélection de données	144
6.1.1	Questions générales	144
6.1.2	Exercices d'application	144
6.2	Solution des exercices sur l'insertion de données	156
6.3	Solutions des exercices sur la suppression de données	159
6.4	Solution des exercices sur la modification de données	160

Chapitre 4 Les fonctions

1. Introduction	163
2. Les fonctions numériques	163
2.1 ABS : valeur absolue	164
2.2 ASCII : valeur ASCII d'un caractère	165
2.3 COS : cosinus - SIN : sinus	165
2.4 LOG (<numéro base>, <colonne>) : logarithme de la colonne sélectionnée dans la base indiquée	166
2.5 MOD(<colonne>, <valeur>) : modulo	166
2.6 ROUND(<colonne>, [<précision>]) : arrondi	166
2.7 SQRT : racine carrée	167
3. Les fonctions de comparaison et de tests	168
3.1 IN - NOT IN	168
3.2 EXISTS - NOT EXISTS (existence ou non)	173
3.3 BETWEEN (entre deux valeurs)	174
3.4 LIKE (qui contient une partie de valeur)	176
4. Les fonctions de gestion des dates et heures	178
4.1 Date du jour : CURRENT_DATE	178
4.2 Heure actuelle	180
4.3 Date et heure du jour : CURRENT_TIMESTAMP	181
4.4 Les différents formats d'affichage des dates	182
4.5 La manipulation des dates et des heures	187
5. Les fonctions sur les chaînes de caractères	192
5.1 LOWER / UPPER / UCASE / LCASE (minuscules et majuscules)	192
5.2 Supprimer les espaces à droite ou à gauche d'une chaîne de caractères : TRIM / LTRIM / RTRIM	193
5.3 Transformer un numérique ou une date en caractères : TO_CHAR	195
5.4 Trouver la position d'une chaîne de caractères dans une chaîne : INSTR	196

5.5	Ajouter des caractères avant ou après une chaîne : LPAD / RPAD	197
5.6	Extraire une partie d'une chaîne de caractères : SUBSTR	198
6.	Les autres fonctions	198
6.1	NVL : tester une colonne à null	198
6.2	Tester plusieurs valeurs : COALESCE	199
6.3	Comparer deux colonnes : NULLIF	200
6.4	Changer le type d'une colonne : CAST	201
6.5	Test du contenu d'une donnée avec DECODE	202
7.	Exercices sur les fonctions	208
8.	Solution des exercices sur les fonctions	209

Chapitre 5

La sécurité des données (DCL)

1.	Introduction	215
2.	Pourquoi définir des droits	215
3.	Créer un utilisateur	222
4.	Changer le mot de passe d'un utilisateur	223
5.	Attribuer des droits (GRANT)	224
5.1	Attribuer des droits sur la manipulation d'une table	224
5.2	Attribuer des droits sur les objets de la base	229
5.3	Les autres droits possibles	232
6.	Supprimer des droits (REVOKE)	235
6.1	Supprimer des droits sur la manipulation d'une table	235
6.2	Supprimer des droits sur les objets de la base	236
7.	Utilisation des rôles	236
8.	Supprimer un rôle	237
9.	Exercices sur la sécurité	238
10.	Solution des exercices	238

Chapitre 6 Le contrôle de transactions (TCL)

1. Problématique des accès concurrents	241
1.1 Illustration des accès concurrents	241
1.1.1 Exemple 1 : mises à jour simultanées	241
1.1.2 Exemple 2 : incohérence des données suite à une modification d'un autre utilisateur	243
1.2 Le mécanisme de verrouillage	244
2. Notion de transaction	245
2.1 Définition d'une transaction	246
2.2 Comment éviter les incohérences de données	246
2.3 Mise en œuvre d'un verrouillage	246
2.3.1 READ UNCOMMITTED	247
2.3.2 READ COMMITTED	247
2.3.3 REPEATABLE-READ	247
2.3.4 SERIALIZABLE	247
2.3.5 Syntaxes	248
2.4 Mise en œuvre d'un verrouillage applicatif	248
2.4.1 Comment connaître les verrous posés sur une table ..	250
2.4.2 Comment supprimer des verrous posés sur une table ..	251
2.5 Validation des modifications (COMMIT)	253
2.6 Abandon des modifications (ROLLBACK)	254
2.7 Les points de synchronisation (SAVEPOINT)	254
2.8 Exemple d'utilisation des points de synchronisation	255

Chapitre 7 En allant plus loin

1. Charger des données en masse avec SQL*Loader	261
2. Les imports et exports de tables avec Oracle	269
2.1 Les exports de tables	269
2.2 Les imports de tables	273

3. Les sous-SELECT	274
4. Quelques notions de performances	280
4.1 Utilisation de EXPLAIN PLAN	281
4.2 Utilisation du package DBMS_XPLAN.DISPLAY	285
4.3 Optimisation des requêtes par l'utilisation des HINTS	286
4.4 Conclusion	288
5. Présentation de SQL*Plus	288
5.1 Utilisation standard	288
5.2 La commande SPOOL	291
5.3 Les commandes SET	292
5.3.1 Les commandes SET qui concernent l'affichage des données	292
5.3.2 Les commandes SET d'environnement	295
5.4 L'utilisation de variables dans SQL*Plus	296
5.5 L'appel de procédures stockées	297
6. Présentation des TRIGGER	298
6.1 Création d'un TRIGGER de contrôle et mise à jour dans une table	301
6.2 Création d'un TRIGGER de suivi des mises à jour	304
7. Les tables système (Oracle)	308
7.1 Tables système pour les tables et colonnes	308
7.1.1 Oracle	308
7.1.2 MySQL	308
7.2 Tables système pour les index et les vues	308
7.2.1 Oracle	308
7.2.2 MySQL	309
7.3 Les autres tables système	309
7.3.1 Oracle	309
7.3.2 MySQL	309

8. Quelques scripts bien utiles	310
8.1 Connaître la taille réelle d'une colonne	310
8.2 Rechercher et supprimer des doublons dans une table	311
8.3 Extraire les données d'une table dans un fichier à plat	312
8.4 Afficher le contenu d'une table sans connaître sa structure . .	314
8.5 Générer les ordres d'insertion dans une table à partir d'un fichier Excel	316
8.6 Procédure stockée pour supprimer des lignes dans une table .	319

Chapitre 8 Présentation de PL/SQL

1. Introduction	321
2. Syntaxe générale	321
3. Les curseurs	325
4. Les boucles FOR, WHILE, LOOP et la structure conditionnelle CASE	329
4.1 Le WHILE	329
4.2 Le FOR	331
4.3 Le LOOP	334
4.4 Le CASE	335
5. Les exceptions les plus utilisées	337
6. Création d'une procédure stockée	337
7. Création d'une fonction stockée	342
8. Les packages	344
9. Compilation d'une procédure, d'une fonction ou d'un package . .	348
10. Suppression d'une procédure, d'une fonction ou d'un package . .	348

Chapitre 9

Les erreurs les plus couramment rencontrées

1. Introduction 349
2. Sur les accès aux données (LDD/LMD) 350
3. Sur les transactions et les sessions (TCL/DCL) 364
4. Sur les composants internes (mémoire, système) 366

Annexes

1. Récapitulatif des ordres principaux. 369
 - 1.1 Les principaux ordres du LDD (langage de définition de données) ou DDL (Data Definition Language) 369
 - 1.2 Les principaux ordres du LMD (langage de manipulation de données) ou DML (Data Manipulation Language) 372
 - 1.3 Les principaux ordres du LCD (langage de contrôle des données) ou DCL (Data Control Language)..... 376
 - 1.4 Les principaux ordres du LCT (langage de contrôle des transactions) ou TCL (Transaction Control Language) .. 378
 - 1.5 La création de procédures et de fonctions. 380
2. Fonctions SQL présentées dans ce livre. 382
3. Glossaire 383

- Index 385

Chapitre 3

La manipulation des données (LMD)

1. Introduction

Le langage de manipulation de données permet aux utilisateurs et aux développeurs d'accéder aux données de la base, de modifier leur contenu, d'insérer ou de supprimer des lignes.

Il s'appuie sur quatre ordres de base qui sont SELECT, INSERT, DELETE et UPDATE.

Ces quatre ordres ne sont pas toujours autorisés par l'administrateur de la base qui est le seul à pouvoir attribuer ou non les droits d'utilisation sur ces ordres.

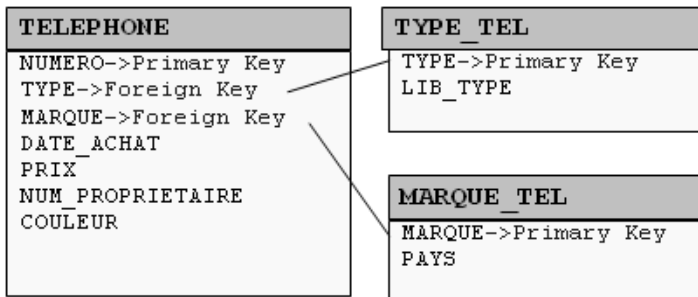
Pour l'utilisateur lambda, il pourra indiquer que seul l'ordre SELECT est utilisable. Les ordres de modification de la base ne sont pas accessibles pour certains utilisateurs pour des raisons évidentes de sécurité.

2. La sélection de données

L'ordre SELECT permet de réaliser des requêtes simples assez rapidement même sans connaissance approfondie en langage de programmation.

Il peut également être très puissant si l'on connaît toutes les fonctions et toutes les possibilités du langage. On peut réaliser des requêtes complexes, avec de nombreuses tables mais il faut toujours faire attention aux performances qui peuvent se dégrader très rapidement sur un ordre SQL mal construit ou n'utilisant pas les bons index dans les tables. Il faut être vigilant et utiliser les outils d'analyse de requête (cf. chapitre En allant plus loin - Quelques notions de performances) avant d'exécuter une requête sur une base réelle avec des tables conséquentes.

Les tables de base qui sont utilisées dans les paragraphes ci-dessous sont celles-ci :



Modèle de données utilisé pour illustrer la sélection de données

2.1 L'ordre de sélection de données SELECT

Le SELECT est l'ordre le plus important et le plus utilisé en SQL. Avec cet ordre nous pouvons ramener des lignes de une ou plusieurs tables mais également transformer des données par l'utilisation de fonction ou encore réaliser des calculs.

Nous allons décrire progressivement les possibilités de cet ordre dans les paragraphes suivants.

L'utilisation la plus courante consiste à sélectionner des lignes dans une table comme ceci :

```
■ SELECT DATE_ACHAT, TYPE, MARQUE FROM TELEPHONE;
```

Dans cet exemple, nous avons sélectionné trois colonnes de la table TELEPHONE.

L'ordre va donc nous ramener toutes les lignes de la table pour ces trois colonnes.

Si on avait voulu toutes les colonnes et toutes les lignes de la table, l'ordre aurait été celui-ci :

```
■ SELECT * FROM TELEPHONE;
```

NUMERO	TYPE	MARQUE	DATE_ACHAT	PRIX	NUM_PROPRIETAIRE	COULEUR
1	SP	1	15/01/10	159	190120	ROUGE
2	SP	1	14/03/10	99	190215	
3	CL	3	02/05/10	49	190001	NOIR
4	CO	4	25/07/10	89	190222	BLANC
5	IP	5	30/09/10	359	190561	

L'étoile est pratique lorsque l'on ne connaît pas le nom des colonnes, mais le résultat est rarement lisible avec des tables contenant un nombre important de colonnes. Pour connaître le nom des colonnes, faites un DESC de la table auparavant (DESC <nom table>).

La syntaxe simple est donc :

```
■ SELECT <colonne 1>, <colonne 2> ... | * FROM <table1>, <table2> ...
```

Si certaines colonnes ont le même nom mais appartiennent à des tables différentes, il faudra ajouter le nom de la table devant la colonne afin que le système sache quelle colonne prendre.

Il n'est pas obligatoire de mettre le nom des tables devant chaque colonne, mais pour une question de lisibilité et de maintenance il est préférable de les mettre sur des sélections complexes.

```
SELECT TELEPHONE.DATE_ACHAT,  
       TELEPHONE.TYPE,  
       TELEPHONE.MARQUE,  
       TYPE_TEL.LIB_TYPE,  
       MARQUE_TEL.LIB_MARQUE  
FROM TELEPHONE, TYPE_TEL, MARQUE_TEL;
```

Nous verrons dans la section L'utilisation des alias de ce chapitre que pour alléger la lecture il est également possible de donner un alias à chaque table. Cet alias est souvent simple et permet de retrouver facilement la table concernée. TELEPH pour TELEPHONE par exemple, ou TYP pour TYPE_TEL.

2.2 Les options DISTINCT et ALL

Par défaut lors de l'exécution d'un SELECT toutes les lignes sont ramenées (l'option ALL est automatique). Si l'on veut supprimer les doublons, il faut ajouter l'ordre DISTINCT.

L'ordre DISTINCT s'applique à toutes les colonnes présentes.

Exemple :

```
SELECT TYPE, MARQUE, PRIX FROM TELEPHONE;
```

et

```
SELECT DISTINCT TYPE, MARQUE, PRIX FROM TELEPHONE;
```

Les deux SELECT ci-dessus ont le même résultat car il y a un doublon sur les deux premières lignes. Ces deux premières colonnes sont identiques mais pas la troisième.

TYPE	MARQUE	PRIX
SP	1	<u>159</u>
SP	1	<u>99</u>
CL	3	49
CO	4	89
IP	5	359

En revanche, si l'on réduit la sélection à deux colonnes comme :

```
■ SELECT TYPE, MARQUE FROM TELEPHONE;
```

TYPE	MARQUE
SP	1
SP	1
CL	3
CO	4
IP	5

Si on ajoute un ordre DISTINCT, une des deux lignes contenant 'SP' et '1' sera supprimée.

```
■ SELECT DISTINCT TYPE, MARQUE FROM TELEPHONE;
```

TYPE	MARQUE
SP	1
CL	3
CO	4
IP	5

■ Remarque

La clause DISTINCT ne peut pas être utilisée avec des opérateurs de regroupement (voir le GROUP BY). En effet les opérateurs de type COUNT ou SUM éliminent automatiquement les doublons.

2.3 L'utilisation des alias

Dans une requête SQL qui comporte plusieurs tables, il est souhaitable d'attribuer un diminutif à chaque nom de table que l'on appelle un alias. Toujours dans l'optique de rendre les requêtes plus lisibles pour tous les programmeurs.

Cet alias peut aussi être utilisé pour un nom de colonne mais également pour un résultat de fonction ou de SELECT imbriqué.

Un alias suit directement l'élément qu'il va supplanter.

Exemple d'alias sur une table :

```
■ SELECT TYPE, MARQUE, PRIX FROM TELEPHONE TELEPH;
```

Lorsqu'il existe plusieurs tables, il faut donner un alias différent pour chacune des tables comme ceci :

```
■ SELECT TELEPH.DATE_ACHAT,  
        TELEPH.MARQUE,  
        MARQ.LIB_MARQUE  
FROM TELEPHONE TELEPH, MARQUE_TEL MARQ;
```

L'alias sur une colonne remplace lors de l'affichage le nom de la colonne, initiale. Cela peut permettre de rendre le résultat plus explicite pour un utilisateur ou de cacher le nom réel de la colonne.

Exemple d'alias sur une colonne :

```
■ SELECT TYPE C1, MARQUE C2, PRIX FROM TELEPHONE TELEPH;
```

L'alias est aussi intéressant pour nommer un résultat de calcul. Ainsi l'utilisateur comprendra immédiatement le contenu de cette colonne.

Exemple d'alias sur un résultat de calcul :

```
■ SELECT TYPE C1, MARQUE C2, PRIX*6.55957 PRIX_FRANC FROM TELEPHONE  
TELEPH;
```

L'alias est aussi utilisé lorsque deux colonnes ont le même nom dans deux tables différentes. Il faut préciser au système l'origine de la colonne.

Par exemple, la colonne TYPE existe dans la table TELEPHONE et la table TYPE_TEL. C'est assez logique étant donné qu'il s'agit de la FOREIGN KEY de la table TELEPHONE qui pointe sur la table TYPE_TEL.

Donc si l'on veut écrire une requête qui utilise ces deux tables, la jointure s'effectuera sur cette clé, il faut alors différencier les deux colonnes.