

Christian Soutou

Mis à jour avec MySQL 8.0

Programmer avec MySQL

SQL • Transactions • PHP • Java • Optimisations
Avec 40 exercices corrigés

5^e édition

EYROLLES

Programmer avec MySQL

5^e édition

Apprendre SQL par l'exemple

Particulièrement destiné aux débutants, cet ouvrage permet de découvrir tous les aspects de la programmation SQL (création de tables, évolution, mises à jour et extractions) par le biais du système de gestion de bases de données MySQL. Les concepts du langage procédural de MySQL y sont décrits avec précision : variables, structure de contrôle, interactions avec la base, sous-programmes, curseurs, transactions, gestion des exceptions, déclencheurs, SQL dynamique... L'auteur explique en outre comment exploiter une base MySQL (connexion et transactions) en programmant avec Java (JDBC) ou PHP. Chaque notion importante du livre est introduite à l'aide d'exemples simples et chaque chapitre se clôt par une série d'exercices, avec corrigés disponibles en ligne, qui permettront au lecteur de tester ses connaissances.

Une nouvelle édition mise à jour avec MySQL 8.0

Cette cinquième édition inclut les fonctionnalités de la version 8.0 de MySQL, notamment la gestion des espaces de stockage, les fonctions SQL pour JSON et la récursivité avec les CTE. L'optimisation des requêtes est également détaillée, en particulier le fonctionnement de l'optimiseur, l'utilisation des statistiques et les plans d'exécution.

À qui s'adresse ce livre ?

- À tous ceux qui souhaitent s'initier à MySQL
- Aux développeurs Java et PHP

Sur www.editions-eyrolles.com/dl/0067379

- Téléchargez le code source des exemples et le corrigé des exercices
- Consultez les mises à jour et les compléments
- Dialoguez avec l'auteur

Maître de conférences rattaché au département Réseaux et Télécoms de l'IUT de Blagnac, **Christian Soutou** intervient en licence et master professionnels. Il est aussi consultant indépendant chez Orsys et auteur de nombreux ouvrages aux éditions Eyrolles.

Au sommaire

Partie I : Les bases de SQL. Définition et manipulation des données. Évolution d'un schéma. Interrogation et contrôle des données. **Partie II : Programmation procédurale.** Bases du langage de programmation. Programmation avancée. **Partie III : Langages et outils.** Utilisation avec Java. Utilisation avec PHP. Optimisations.

Christian Soutou

Programmer avec MySQL

SQL • Transactions • PHP • Java • Optimisations

5^e édition

EYROLLES

The logo for EYROLLES, featuring the word "EYROLLES" in a bold, sans-serif font, centered above a horizontal line with a small circle in the middle.

© Groupe Eyrolles, 2006, 2011, 2013, 2015, 2017 pour la présente édition, ISBN : 978-2-212-67379-1

Éditions Eyrolles
61, bd Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans l'autorisation de l'Éditeur ou du Centre Français d'exploitation du droit de copie, 20, rue des Grands Augustins, 75006 Paris.

Avant-propos

Nombre d'ouvrages traitent de MySQL ; certains ressemblent à des bottins téléphoniques ou proviennent de la traduction hasardeuse de la documentation officielle. Les survivants ne sont peut-être plus vraiment à jour.

Ce livre a été rédigé avec une volonté de concision et de progression dans la démarche ; il est illustré, par ailleurs, de nombreux exemples et figures. Bien que la source principale d'informations fût la documentation officielle de MySQL (<http://dev.mysql.com/doc>), l'ouvrage ne constitue pas un condensé de commandes SQL. Chaque notion importante est introduite par un exemple simple et que j'espère démonstratif. En fin de chaque chapitre, des exercices vous permettront de tester vos connaissances.

La documentation en ligne des différentes versions de MySQL (*MySQL 5.x Reference Manual*) représente une dizaine de mégaoctets au format HTML. Tous les concepts et commandes s'y trouvant ne pourraient pas être ici décemment expliqués, sauf peut-être si cet ouvrage ressemblait à un annuaire. J'ai tenté d'en extraire seulement les aspects fondamentaux sous la forme d'une synthèse. Vous n'y trouverez donc pas des considérations à propos d'aspects avancés du langage ou du serveur comme l'administration, la mise en cluster ou la réplication.

Ce livre résulte de mon expérience de l'enseignement dans le domaine des bases de données en premier, deuxième et troisième cycles universitaires dans des cursus d'informatique à vocation professionnelle (IUT, licences et masters professionnels).

Cet ouvrage s'adresse principalement aux novices désireux de découvrir SQL en programmant sous MySQL.

- Les étudiants et enseignants trouveront des exemples pédagogiques pour chaque concept abordé, ainsi que des exercices thématiques.
- Les développeurs PHP ou Java découvriront des moyens de stocker leurs données.

Guide de lecture

Ce livre s'organise autour de trois parties distinctes mais complémentaires. La première intéressera le lecteur débutant en la matière, car elle concerne les instructions SQL et les notions de base de MySQL. La deuxième partie décrit la programmation avec le langage procédural de MySQL. La troisième partie attirera l'attention des programmeurs qui envisagent d'utiliser MySQL à l'aide d'outils natifs, ou tout en programmant avec des langages évolués ou via des interfaces Web (PHP ou Java). Enfin des aspects plus avancés de SQL sont abordés comme l'optimisation des requêtes.

Première partie : SQL de base

Cette partie présente les différents aspects du langage SQL de MySQL, en étudiant en détail les instructions de base. À partir d'exemples, j'explique notamment comment déclarer, manipuler, faire évoluer et interroger des tables avec leurs différentes caractéristiques et leurs éléments associés (contraintes, index, vues, séquences). Nous étudions aussi SQL dans un contexte multi-utilisateur (droits d'accès), et au niveau du dictionnaire de données.

Deuxième partie : programmation procédurale

Cette partie décrit les caractéristiques du langage procédural de MySQL. Le chapitre 6 traite des éléments de base (structure d'un programme, variables, structures de contrôle, interactions avec la base et transactions). Le chapitre 7 traite des sous-programmes, des curseurs, de la mise en œuvre d'exceptions, de déclencheurs et l'utilisation du SQL dynamique. La gestion des documents XML et JSON est également décrite.

Troisième partie : langages et outils

Cette partie intéressera les programmeurs qui envisagent d'exploiter une base MySQL en utilisant un langage de programmation. Le chapitre 8 détaille l'API JDBC qui permet de manipuler une base MySQL par l'intermédiaire d'un programme Java. Le chapitre 9 décrit les principales fonctions de l'API *mysql_i* qui permet d'interfacer un programme PHP avec une base MySQL.

Le chapitre 10 est enfin consacré à l'optimisation des requêtes et des schémas. Plusieurs aspects sont étudiés : le fonctionnement de l'optimiseur, l'utilisation de statistiques et les plans d'exécution. Enfin, différents mécanismes permettant d'optimiser les traitements sont présentés : contraintes, index, tables temporaires, partitionnement et dénormalisation.

Annexes

Les annexes contiennent une bibliographie, des adresses Web et un index complet.

Convention d'écriture

La police `courrier` est utilisée pour souligner les instructions SQL, noms de types, tables, contraintes, etc. (ex : `SELECT nom FROM Pilote`).

Les majuscules sont employées pour les directives SQL, et les minuscules pour les autres éléments. Les noms des tables, index, vues, fonctions, procédures... sont précédés d'une majuscule (exemple : la table `CompagnieAerienne` contient la colonne `nomComp`).

Les termes de MySQL (bien souvent traduits littéralement de l'anglais) sont notés en italique, exemple : *trigger*, *table*, *column*, etc.

Dans une instruction SQL, les symboles `{ }` désignent une liste d'éléments, et le symbole `< | >` un choix (exemple `CREATE { TABLE | VIEW }`). Les symboles `< [>` et `<] >` précisent le caractère optionnel d'une directive au sein d'une commande (exemple : `CREATE [UNIQUE|FULLTEXT|SPATIAL] INDEX index_name [USING index_type] ON table_name (index_col_name, ...)`).



Ce sigle introduit une définition, un concept ou une remarque importante. Il apparaît soit dans une partie théorique soit dans une partie technique pour souligner des instructions importantes ou la marche à suivre avec SQL.



Ce sigle annonce soit une impossibilité de mise en œuvre d'un concept soit une mise en garde. Il est principalement utilisé dans la partie consacrée à SQL.

J'en profite pour faire passer le message suivant : si vous travaillez en version 4 de MySQL, certaines instructions décrites dans ce livre ne fonctionneront pas. Cet ouvrage n'est pas un guide de référence ! Vous trouverez sur le Web des ressources pour connaître la compatibilité de telle ou telle fonction SQL.



Ce sigle indique à partir de quelle version du serveur MySQL la fonctionnalité (commande SQL ou mécanisme de programmation) est opérationnelle. Les versions 5.1 à 8.0 sont prises en compte.



Ce sigle signale une astuce ou un conseil personnel.

Contact avec l'auteur

Si vous avez des remarques à formuler sur le contenu de cet ouvrage, n'hésitez pas à m'écrire à l'adresse christian.soutou@gmail.com, mais seules les remarques relatives à l'ouvrage devraient trouver une réponse.

Par ailleurs, un site d'accompagnement de l'ouvrage (*errata*, corrigés des exercices, source des exemples et compléments) est disponible, accessible via www.editions-eyrolles.com sur la fiche de l'ouvrage.

Table des matières

Introduction	1
SQL, une norme, un succès	1
Modèle de données	2
Tables et données	2
Les clés	3
MySQL	3
Les produits	4
Licences	4
Les versions	4
Architecture	5
Moteurs de stockage	6
Oracle devient propriétaire	7
Notion de database [schéma]	8
Notion d'hôte	8
Aspects étudiés	9
Mise en œuvre de MySQL [sous Windows]	10
Installation	10
Premiers pas	12
L'interface de commande	12
Création d'un utilisateur	13
Connexion au serveur	13
Vérification de la version	14
Options de base	14
Batch	15
Votre prompt, et vite !	16
Commandes de base	17
Partie I SQL de base	19
1 Définition des données	21
Tables relationnelles	21
Création d'une table [CREATE TABLE]	21
Délimiteurs	22
Sensibilité à la casse	23
Commentaires	23
Premier exemple	24
Contraintes	25
Conventions recommandées	26
Types des colonnes	28

Structure d'une table [DESCRIBE]	32
Restrictions	32
Les collations et jeux de caractères	33
Index	34
Arbres balancés	34
Création d'un index [CREATE INDEX]	35
Bilan	36
Destruction d'un schéma	36
Suppression d'une table [DROP TABLE]	37
Ordre des suppressions	37
Exercices	38
2 Manipulation des données	41
Insertions d'enregistrements [INSERT]	41
Syntaxe	41
Les doublons	42
Renseigner toutes les colonnes	42
Renseigner certaines colonnes	43
Renseignez vos colonnes !	43
Plusieurs enregistrements	44
Ne pas respecter des contraintes	44
Insertions multilignes	45
Données binaires	46
Énumérations	46
Dates et heures	49
Séquences	53
Utilisation en tant que clé primaire	53
Modification d'une séquence	54
Utilisation en tant que clé étrangère	55
Modifications de colonnes	56
Syntaxe [UPDATE]	56
Modification d'une colonne	57
Modification de plusieurs colonnes	57
Modification de plusieurs enregistrements	57
Ne pas respecter les contraintes	58
Restrictions	59
Dates et intervalles	59
Remplacement d'un enregistrement	63
Suppressions d'enregistrements	63
Instruction DELETE	64
Instruction TRUNCATE	64
Intégrité référentielle	65
Syntaxe	65
Cohérences assurées	66
Contraintes côté « père »	67
Contraintes côté « fils »	67

	Clés composites et nulles	68
	Cohérence du fils vers le père	68
	Cohérence du père vers le fils	69
	En résumé	71
	Insertions à partir d'un fichier	71
	Exercices	74
3	Évolution d'un schéma	77
	Renommer une table [RENAME]	77
	Modifications structurelles [ALTER TABLE]	78
	Ajout de colonnes	78
	Renommer des colonnes	79
	Modifier le type des colonnes	79
	Valeurs par défaut	80
	Supprimer des colonnes	80
	Énumérations	81
	Colonnes virtuelles	82
	Modifications comportementales	84
	Ajout de contraintes	84
	Suppression de contraintes	86
	Désactivation des contraintes	88
	Réactivation des contraintes	89
	Contraintes différées	92
	Les collations et jeux de caractères	93
	Exercices	94
4	Interrogation des données	97
	Généralités	97
	Syntaxe [SELECT]	98
	Pseudotable	98
	Projection [éléments du SELECT]	99
	Extraction de toutes les colonnes	100
	Extraction de certaines colonnes	101
	Alias	101
	Duplicatas	102
	Expressions simples	102
	Ordonnancement	103
	Concaténation	104
	Insertion multiligne	104
	Limitation du nombre de lignes	105
	Restriction [WHERE]	105
	Opérateurs de comparaison	106
	Opérateurs logiques	107
	Opérateurs intégrés	107
	Alias	109
	Fonctions	109

Caractères	109
Numériques	113
Fonctions pour les bits	114
Dates	115
Fonctions pour les NULL	119
Conversions	120
Comparaisons	121
Énumérations	122
Fonctions pour les UUID	124
Autres fonctions	127
Regroupements	127
Fonctions de groupe	128
Étude du GROUP BY et HAVING	130
Opérateurs ensemblistes	132
Intersection	133
Opérateurs UNION et UNION ALL	134
Différence	135
Produit cartésien	136
Division	137
Ordonner des résultats	139
Bilan	140
Jointures	141
Classification	141
Jointure relationnelle	142
Jointures SQL2	142
Types de jointures	142
Équijointure	143
Autojointure	145
Inéquijointure	146
Jointures externes	148
Jointures procédurales	150
Tables dérivées (et CTE)	154
Sous-interrogations synchronisées	156
Autres directives SQL2	157
Récursivité avec les CTE	162
Transformations de résultats	167
Exercices	170
5 Contrôle des données	173
Gestion des utilisateurs	174
Classification	174
Création d'un utilisateur [CREATE USER]	174
Modification d'un utilisateur	176
Renommer un utilisateur [RENAME USER]	176
Verrouillage d'un utilisateur	177
Suppression d'un utilisateur [DROP USER]	177

Gestion des bases de données	178
Création d'une base [CREATE DATABASE]	178
Sélection d'une base de données [USE]	178
Modification d'une base [ALTER DATABASE]	179
Suppression d'une base [DROP DATABASE]	179
Création des espaces de stockage	180
Privilèges	181
Niveaux de privilèges	181
Tables de la base mysql	182
Table mysql.user	182
Attribution de privilèges [GRANT]	186
Table mysql.db	190
Table mysql.host	191
Table mysql.tables_priv	191
Table mysql.columns_priv	191
Table mysql.procs_priv	192
Révocation de privilèges [REVOKE]	193
Attributions et révocations « sauvages »	195
Rôles	196
Accès distants	199
Connexion par l'interface de commande	199
Table mysql.host	200
Vues	201
Création d'une vue [CREATE VIEW]	202
Classification	203
Vues monotables	204
Vues complexes	208
Autres utilisations de vues	212
Transmission de droits	215
Modification d'une vue [ALTER VIEW]	216
Visualisation d'une vue [SHOW CREATE VIEW]	216
Suppression d'une vue [DROP VIEW]	216
Dictionnaire des données	217
Constitution	217
Modèle graphique du dictionnaire des données	218
Démarche à suivre	219
Classification des vues	221
Moteurs du serveur	223
Bases de données du serveur	223
Composition d'une base	224
Détail de stockage d'une base	224
Structure d'une table	225
Les collations et jeux de caractères	227
Recherche des contraintes d'une table	228
Composition des contraintes d'une table	229
Détails des contraintes référentielles	229

Recherche du code source d'un sous-programme	231
Paramètres des sous-programmes stockés	232
Privilèges des utilisateurs d'une base de données	232
Commande SHOW	234
Exercices	236

Partie II Programmation procédurale 239

6 Bases du langage de programmation 241

Généralités	241
Environnement client-serveur	241
Avantages	242
Structure d'un bloc	242
Portée des objets	243
Casse et lisibilité	243
Identificateurs	244
Commentaires	244
Variables	244
Variables scalaires	245
Affectations	245
Restrictions	245
Résolution de noms	246
Opérateurs	246
Variables de session	247
Conventions recommandées	248
Test des exemples	248
Structures de contrôle	249
Structures conditionnelles	249
Structures répétitives	251
Interactions avec la base	254
Extraire des données	254
Manipuler des données	256
Gestion des transactions	258
Début et fin d'une transaction	259
Gestion des anomalies transactionnelles	262
Transactions en lecture seule	267
Le problème du verrou mortel [deadlock]	268
Verrouillage manuel	269
Contrôle des transactions	270
Quel mode adopter ?	271
Où placer les transactions ?	271
Modes d'exécution SQL	272
Le contexte	272
Programmation transactionnelle	272
Les dates	273

Les séquences	275
Chaînes de caractères	276
Les moteurs	277
Portabilité	278
Les combinaisons	279
Exercices	281
7 Programmation avancée	283
Sous-programmes	283
Généralités	283
Procédures cataloguées	284
Fonctions cataloguées	285
Structure d'un sous-programme	285
Exemples	286
Fonction n'interagissant pas avec la base	288
Compilation	289
Appel d'un sous-programme	289
Récursivité	291
Sous-programmes imbriqués	292
Modification d'un sous-programme	293
Destruction d'un sous-programme	293
Restrictions	294
Curseurs	294
Généralités	294
Instructions	295
Parcours d'un curseur	296
Accès concurrents [FOR UPDATE]	297
Restrictions	298
Erreurs [codes et messages]	298
Exceptions	299
Généralités	300
Exceptions avec EXIT	301
Exceptions avec CONTINUE	305
Gestion des autres erreurs [SQLEXCEPTION]	306
Même erreur sur différentes instructions	308
Exceptions nommées	311
Déroutements [SIGNAL et RESIGNAL]	313
Déclencheurs	320
Généralités	320
À quoi sert un déclencheur ?	320
Mécanisme général	321
Syntaxe	322
Déclencheurs LMD [de lignes]	323
Appel de sous-programmes	329
Dictionnaire des données	331
Programmation d'une contrainte de vérification	332

- Programmation dans un déclencheur 334
- Exceptions dans un déclencheur 334
- Tables mutantes 337
- Restrictions 338
- Suppression d'un déclencheur 338
- SQL dynamique** 338
 - Syntaxe 339
 - Exemples 340
 - Restrictions 341
 - Paramètres de retour 344
- Programmation d'événements** 345
 - Le contexte 345
 - Création d'une planification 345
 - Exemple 347
 - Dictionnaire des données 349
 - Modification 350
 - Restrictions actuelles 351
- Gestion de XML** 351
 - Fonctions XML 351
 - Gestion des erreurs 358
 - Limitations 359
 - Chargement XML [LOAD XML] 361
- Gestion de JSON** 366
 - Mise en place de l'environnement Document Store 366
 - Les documents JSON 368
 - Les collections JSON 369
 - Méthodes pour les documents JSON 375
 - Fonctions SQL pour JSON 379
- Exercices** 392

Partie III Langages et outils..... 395

8 Utilisation avec Java..... 397

- JDBC avec Connector/J** 397
 - Classification des pilotes [drivers] 398
 - Le paquetage java.sql 399
 - Structure d'un programme 399
 - Test de votre configuration 400
- Connexion à une base** 401
 - Base Access 402
 - Base MySQL 403
 - Interface Connection 403
- États d'une connexion** 404
 - Interfaces disponibles 404
 - Méthodes génériques pour les paramètres 404

États simples [interface Statement]	405
Méthodes à utiliser	406
Correspondances de types	406
Manipulations avec la base	408
Suppression de données	408
Ajout d'enregistrements	409
Modification d'enregistrements	409
Extraction de données	409
Curseurs statiques	410
Curseurs navigables	411
Curseurs modifiables	415
Suppressions	417
Modifications	418
Insertions	418
Gestion des séquences	419
Méthode <code>getGeneratedKeys</code>	420
Curseur modifiable	420
Interface <code>ResultSetMetaData</code>	421
Interface <code>DatabaseMetaData</code>	422
Instructions paramétrées [<code>PreparedStatement</code>]	424
Extraction de données [<code>executeQuery</code>]	424
Mises à jour [<code>executeUpdate</code>]	425
Instruction LDD [<code>execute</code>]	425
Procédures cataloguées	426
Exemple	427
Transactions	428
Points de validation	429
Traitement des exceptions	430
Affichage des erreurs	431
Traitement des erreurs	431
Exercices	433
9 Utilisation avec PHP	435
Configuration adoptée	435
Logiciels	436
Fichiers de configuration	436
Test d'Apache et de PHP	437
Test d'Apache, de PHP et de MySQL	437
API de PHP pour MySQL	438
Connexion	438
Interactions avec la base	439
Extractions	441
Instructions paramétrées	444
Gestion des séquences	446
Traitement des erreurs	447
Procédures cataloguées	448

	Métadonnées	450
	Style d'écriture objet	454
	Exercices	455
10	Optimisations	459
	Cadre général	459
	Les acteurs	459
	Contexte et objectifs	460
	Causes possibles	460
	Présentation du jeu d'exemples	461
	L'optimiseur	461
	L'estimateur	463
	Les statistiques destinées à l'optimiseur	464
	Collecte	465
	Visualisation des statistiques	466
	Quand mettre à jour les statistiques ?	467
	Outils de mesure de performances	468
	MySQL Query Analyzer	468
	Visualisation des plans d'exécution	470
	Organisation des données	475
	Les contraintes	475
	Indexation	477
	Jointures	490
	Index invisibles	492
	Configuration de l'optimiseur [les hints]	494
	Tables temporaires	496
	Partitionnement	497
	Annexe : bibliographie et webographie	509
	Index	511

Introduction

Dans cette introduction, nous présentons, tout d'abord, le cadre général dans lequel cet ouvrage se positionne (SQL, le modèle de données et l'offre MySQL). Nous décrivons, pour finir, la procédure d'installation de MySQL sous Windows et l'utilisation de l'interface de commande en ligne pour que vous puissiez programmer en SQL dès le chapitre 1.

SQL, une norme, un succès

C'est IBM, à *tout seigneur tout honneur*, qui, avec System-R, a implanté le modèle relationnel au travers du langage SEQUEL (*Structured English as QUery Language*), rebaptisé par la suite SQL (*Structured Query Language*).

La première norme (SQL1) date de 1987. Elle était le résultat de compromis entre constructeurs, mais elle était fortement influencée par le dialecte d'IBM. SQL2 a été normalisée en 1992. Elle définit quatre niveaux de conformité : le niveau d'entrée (*entry level*), les niveaux intermédiaires (*transitional* et *intermediate levels*) et le niveau supérieur (*full level*). Les langages SQL des principaux éditeurs sont tous conformes au premier niveau et ont beaucoup de caractéristiques relevant des niveaux supérieurs. Depuis 1999, la norme est appelée SQL3. Elle comporte de nombreuses parties (concepts objets, entrepôts de données, séries temporelles, accès à des sources non SQL, réplication des données, etc.).

Le succès que connaissent les éditeurs de SGBD relationnels a plusieurs origines et repose notamment sur SQL :

- Le langage est une norme depuis 1986, qui s'enrichit au fil du temps.
- SQL peut s'interfacer avec des langages de troisième génération comme C ou Cobol, mais aussi avec des langages plus évolués comme C++, Java ou C#. Certains considèrent ainsi que le langage SQL n'est pas assez complet (le dialogue entre la base et l'interface n'est pas direct), et la littérature parle de « défaut d'impédance » (*impedance mismatch*).
- Les SGBD rendent indépendants programmes et données (la modification d'une structure de données n'entraîne pas forcément une importante refonte des programmes d'application).
- Ces systèmes sont bien adaptés aux grandes applications informatiques de gestion (architectures type client-serveur et Internet) et ont acquis une maturité sur le plan de la fiabilité et des performances.
- Ils intègrent des outils de développement comme les précompilateurs, les générateurs de code, d'états, de formulaires.

- Ils offrent la possibilité de stocker des informations non structurées (comme le texte, l'image, etc.) dans des champs appelés LOB (*Large Object Binary*).

Nous étudierons les principales instructions SQL de MySQL qui sont classifiées dans le tableau suivant :

Tableau I-1 Classification des ordres SQL

Ordres SQL	Aspect du langage
CREATE - ALTER - DROP - COMMENT - RENAME - TRUNCATE - GRANT - REVOKE	Définition des données (DDL : <i>Data Definition Language</i>)
SELECT - INSERT - UPDATE - DELETE - LOCK TABLE	Manipulation des données (DML : <i>Data Manipulation Language</i>)
COMMIT - ROLLBACK - SAVEPOINT - SET TRANSACTION	Contrôle des transactions (TCL : <i>Transaction Control Statements</i>)

Modèle de données

Le modèle de données relationnelles repose sur une théorie rigoureuse bien qu'adoptant des principes simples. La table relationnelle (*relational table*) est la structure de données de base qui contient des enregistrements appelés aussi « lignes » (*rows*). Une table est composée de colonnes (*columns*) qui décrivent les enregistrements.

Tables et données

Considérons la figure suivante qui présente deux tables relationnelles permettant de stocker des compagnies, des pilotes et le fait qu'un pilote soit embauché par une compagnie :

Figure I-1 Deux tables

Compagnie

comp	nrue	rue	ville	nomComp
AF	124	Port Royal	Paris	Air France
SING	7	Camparols	Singapour	Singapore AL

Pilote

brevet	nom_pil	pseudo	nbHVol	compa
PL-1	Ferrage	Shark	450	AF
PL-2	Tisseyre	Ninja	900	AF
PL-3	Guilbaud	Thai	1000	SING

Les clés



La clé primaire (*primary key*) d'une table est l'ensemble minimal de colonnes qui permet d'identifier de manière unique chaque enregistrement.

Dans la figure précédente, les colonnes « clés primaires » sont notées en gras. La colonne comp représente le code de la compagnie et la colonne brevet décrit le numéro du brevet.



Une clé est dite « candidate » (*candidate key*) si elle peut se substituer à la clé primaire à tout instant. Une table peut contenir plusieurs clés candidates ou aucune.

Dans notre exemple, les colonnes nomComp et pseudo peuvent être des clés candidates si on suppose qu'aucun homonyme n'est permis.



Une clé étrangère (*foreign key*) référence dans la majorité des cas une clé primaire d'une autre table (sinon une clé candidate sur laquelle un index unique aura été défini). Une clé étrangère est composée d'une ou de plusieurs colonnes. Une table peut contenir plusieurs clés étrangères ou aucune.

Dans notre exemple, la colonne compa (notée en italique dans la figure) est une clé étrangère, car elle permet de référencer un enregistrement unique de la table Compagnie via la clé primaire comp.

Le modèle relationnel est ainsi fondamentalement basé sur les valeurs. Les associations entre tables sont toujours binaires et assurées par les clés étrangères. Les théoriciens considèrent celles-ci comme des pointeurs logiques. Les clés primaires et étrangères seront définies dans les tables en SQL à l'aide de contraintes.

MySQL

Le SGBD MySQL a été développé en C et C++ par l'équipe suédoise TcX, dans l'objectif d'améliorer le logiciel mSQL. La première version est apparue en mai 1995, elle fut distribuée par la société MySQL AB (Uppsala, Suède) fondée par David Axmark, Allan Larsson et Michael Widenius. Voici le modeste slogan affiché sur le site <http://www.mysql.com> : *The world's most popular open source database* (« La base de données open source la plus populaire au monde »).

Depuis 1999, par analogie avec les systèmes d'exploitation, MySQL connaît le succès de Linux. Téléchargée plus d'un million de fois par mois, la version *production* doit sa popularité à son caractère *open source*, ses fonctionnalités de plus en plus riches, ses performances, son

ouverture à tous les principaux langages du marché, son fonctionnement sur les systèmes les plus courants et sa facilité d'utilisation pour des applications Web de taille moyenne.

Les produits

Les produits de la société MySQL sont les suivants :

- *MySQL Enterprise*, qui inclut *MySQL Enterprise Server* (le SGBD version complète), *MySQL Enterprise Monitor* (la console d'administration) et *MySQL Production Support* (le service de support) ;
- *MySQL Cluster*, qui implémente la solution de haute disponibilité (architecture en *cluster*) ;
- *MySQL Embedded Server* (SGBD seul), supporté par plus de 20 plates-formes, de Microsoft Windows à Mac OS X en passant par Linux, Sun Solaris, HP-UX, IBM AIX, Novell Netware, etc. ;
- *MySQL Connectors*, pilotes (*drivers*) permettant un accès à tout programme. Citons principalement ADO.NET (*Connector/NET*), ODBC (*Connector/ODBC*), JDBC (*Connector/J*), C++ (*Connector/C++*), C (*Connector/C*), *MySQL Connector for OpenOffice.org*, pilotes pour PHP, Perl, Python et Ruby ;
- *MySQL Workbench*, outil graphique pour les DBA, développeurs et concepteurs.
- *MySQL Fabric*, qui permet d'administrer des serveurs en cluster en basculant par exemple, un serveur esclave pour devenir primaire suite à un incident sur le serveur principal.

Licences

Deux types de licences sont proposés par MySQL : commerciale et GPL (depuis la version 3.23.19 en juin 2000). Dans le cadre d'un développement d'application entièrement sous licence GPL, MySQL est gratuit. Il en va de même s'il n'est pas copié, modifié, distribué ou employé pour une utilisation en combinaison avec un serveur Web (si vous développez l'application Web vous-même).

Dans tous les autres cas, il est nécessaire d'obtenir une licence commerciale. Par exemple, si vous incluez un serveur MySQL ou des pilotes MySQL dans une application qui n'est pas *open source*.

Les versions

Le tableau suivant présente l'historique des versions de MySQL. Le rythme des mises à jour majeures (*Generally Available*, aussi appelées GA) est d'environ une tous les 18 à 24 mois. Des versions intermédiaires (*Milestone* puis *Release Candidate*) apparaissent régulièrement entre deux versions de production.

Tableau I-2 Dates importantes pour MySQL

Année – version	Caractéristiques principales
1999 – 3.23.x	Réplication – Recherches textuelles – Transactions – Intégrité référentielle pour le moteur InnoDB (2002 – 3.23.44)
2001 – 4.0.x	Cache de requêtes – Sécurisation SSL
2003 – 4.1.x	Support de Unicode – Données géographiques – SQL dynamique
2005 – 5.0.x	Vues – Curseurs – Procédures cataloguées – Déclencheurs – Dictionnaire des données – Transactions distribuées (XA)
2008 – 5.1.x	Jointures externes – Planificateur d'événements – Partitionnement – Réplication au niveau ligne – Tables <i>server log</i>
2010 – 5.5	Nouveaux algorithmes de requêtes – Optimisations des sous-requêtes – Montée en charge accrue
2012 – 5.6	Évolutions de l'optimiseur et de la réplication – NoSQL pour InnoDB – Évolutions SIG - Conformité IPv6
2015 – 5.7	Évolutions de l'optimiseur, de la sécurité – Prise en compte de JSON – Mode Document (NoSQL) – X Protocol – Évolutions SIG
2017 – 8.0	Dictionnaire des données transactionnel – Rôles – Index invisibles – Abandon de MySAM



Vous trouverez toutes les nouvelles fonctionnalités à l'adresse suivante : <http://dev.mysql.com/doc/relnotes/mysql/x.y/en/> (*x.y* étant le numéro de version, par exemple 5.7).

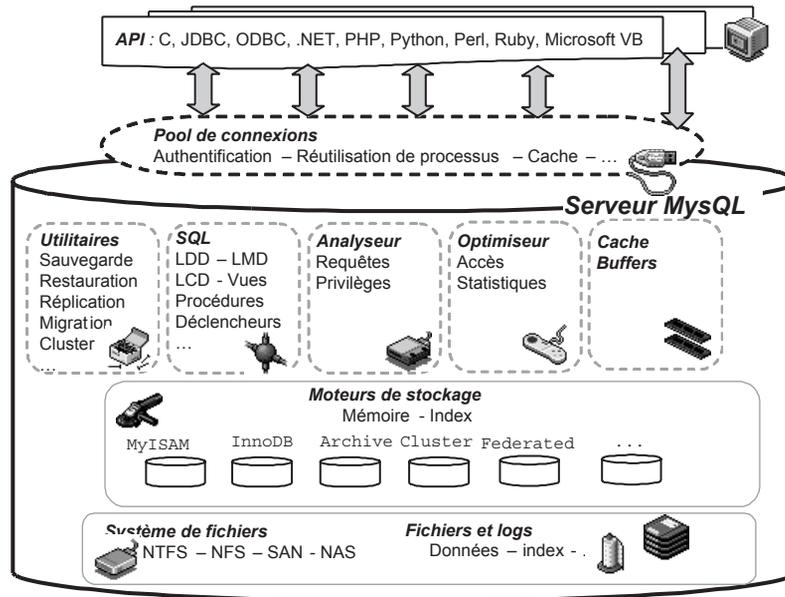
Architecture

La figure suivante (je remercie au passage le site <http://www.iconarchive.com>) présente la majeure partie des fonctionnalités du serveur de données (SGBD).

Les API permettent d'intégrer SQL dans des programmes de différents langages. Le langage SQL sera utilisé par tous ceux (manuellement ou par un outil) travaillant sur la base de données (administrateur, développeur, utilisateur). Le langage procédural de MySQL permet d'incorporer nativement tout ordre SQL dans un programme.

Concrètement, une fois téléchargé et installé, vous avez accès à un SGBD, un client en mode texte (interface de commande). Les pilotes ODBC, JDBC, API pour les langages C et C++, et les outils d'administration seront à installer par la suite. Si vous désirez utiliser MySQL dans le cadre d'un site Web, il existe des paquetages incluant notamment Apache et PHP (WAMP ou LAMP respectivement pour Windows et Linux). Le chapitre 9 décrit une procédure manuelle pour interconnecter ces systèmes.

Figure I-2 Offre MySQL



Moteurs de stockage

La particularité du SGBD MySQL est de pouvoir proposer différents moteurs de stockage. Le choix de tel ou tel moteur dépendra de la façon dont on veut stocker et traiter les données de chaque table (voir le chapitre 1).

Les principaux paramètres à prendre en considération concernent les capacités de stockage, les transactions, les modes de verrouillage, la gestion des index, des caches, des données textuelles, géographiques, compressées ou cryptées, etc. Plusieurs moteurs sont proposés par la société MySQL AB, d'autres sont issus de communautés ou d'éditeurs indépendants. Parmi les moteurs natifs les plus utilisés, citons :

- **MyISAM** : moteur par défaut. Ne supporte pas les transactions mais possède des fonctionnalités de recherche de texte.
- **InnoDB** : sans doute le plus utilisé de nos jours. Supporte le mode transactionnel (verrouillage niveau ligne, *commit* et *rollback*) et les contraintes référentielles (clés étrangères).
- **MEMORY** (anciennement **HEAP**) : stockage des données et index en RAM. Convient à des données non persistantes.
- **ARCHIVE** : stockage des données sous une forme compressée (seuls les `SELECT` et `INSERT` sont possibles). Modèle idéal pour archiver des données.

- CSV (*Comma Separated Value*) : stockage des données sous forme de fichiers texte dans lesquels les valeurs sont séparées par une virgule.
- FEDERATED : convient pour les architectures réparties (plusieurs serveurs).
- NDB (*Network DataBase*) : convient pour les architectures en cluster.

Oracle devient propriétaire

Contrairement à la rumeur qui courait début 2007, MySQL n'est pas entré en Bourse. Il a été racheté pour un milliard de dollars en janvier 2008 par Sun Microsystems, déjà propriétaire de Java. Sun se positionne ainsi sur un segment où il était encore absent jusque là, aux côtés d'Oracle, d'IBM et de Microsoft.

Craignant l'achat de Sun par IBM et redoutant HP dans le haut de gamme Unix, Oracle se repositionne dans le hardware et sur le marché des services pour *datacenters* en avril 2009, en achetant Sun. Ce sont aussi les langages Java et le système d'exploitation Solaris qui ont pesé dans la balance. En effet, c'est sur Solaris, et non sur Linux, que sont déployés le plus grand nombre de serveurs Oracle.

Il faudra attendre novembre 2009 pour que la Commission européenne confirme son refus de la fusion entre Oracle et Sun, suspectant que le rachat de MySQL aboutisse à une situation de quasi monopole sur le marché des SGBD. En décembre 2009, avec le soutien de quelques cinquante-neuf sénateurs américains, Oracle publie dix engagements concernant toutes les zones géographiques et pour une durée de cinq ans :

1. Assurer aux utilisateurs le choix de leur moteur (*MySQL's Pluggable Storage Engine Architecture*).
2. Ne pas changer les clauses d'utilisation d'une manière préjudiciable à un fournisseur tiers.
3. Poursuivre les accords commerciaux contractés par Sun.
4. Garder MySQL sous licence GPL.
5. Ne pas imposer un support des services d'Oracle aux clients du SGBD.
6. Augmenter les ressources allouées à la R&D de MySQL.
7. Créer un comité d'utilisateurs pour, dans les six mois, étudier les retours et priorités de développement de MySQL.
8. Créer ce même comité pour les fournisseurs de solutions incluant MySQL.
9. Continuer d'éditer, mettre à jour et distribuer gratuitement le manuel d'utilisation du SGBD.
10. Laisser aux utilisateurs le choix de la société qui assurera le support de MySQL.

Considérant d'une part ces engagements, et d'autre part l'existence de concurrents (notamment IBM, Microsoft et PostgreSQL dans le monde de l'open source), la Commission européenne avalise la fusion fin janvier 2010 pour un montant de 7,4 milliards de dollars.

Notion de database (schéma)

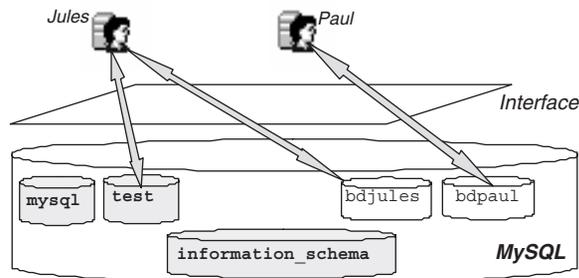
MySQL appelle *database* un regroupement logique d'objets (tables, index, vues, déclencheurs, procédures cataloguées, etc.) pouvant être stockés à différents endroits de l'espace disque. Je ferai donc souvent référence au terme « base de données » pour parler de cette notion.

On peut aussi assimiler ce concept à la notion de schéma, pour ceux qui connaissent Oracle. Là où MySQL et d'autres SGBD diffèrent, c'est sur la notion d'utilisateur (*user*) :

- Pour tous, un utilisateur sera associé à un mot de passe pour pouvoir se connecter et manipuler des tables (s'il en a le droit, bien sûr).
- Pour MySQL, il n'y a pas de notion d'appartenance d'un objet (table, index, etc.) à un utilisateur. Un objet appartient à son schéma (*database*). Ainsi, deux utilisateurs distincts (Jules et Paul) se connectant sur la même base (*database*) ne pourront pas créer chacun une table ayant pour nom *Compagnie*. S'ils doivent le faire, ce sera dans deux bases différentes (*bdjules* et *bdpaul*).
- Pour Oracle ou d'autres SGBD, chaque objet appartient à un schéma (*user*). Ainsi, deux utilisateurs distincts (Jules et Paul) se connectant à la base (qui est un ensemble de schémas) pourront créer chacun une table ayant pour nom *Compagnie* (la première sera référencée *Jules.Compagnie*, la seconde *Paul.Compagnie*).

La figure suivante illustre deux utilisateurs travaillant sur différentes bases par une interface qui peut être la fenêtre de commande en ligne ou un langage de programmation via une API C, Java ou PHP. Notez l'existence de trois bases présentes initialement (*mysql*, *test* et *information_schema*) que nous détaillerons au chapitre 5.

Figure I-3 Bases et utilisateurs MySQL



Notion d'hôte

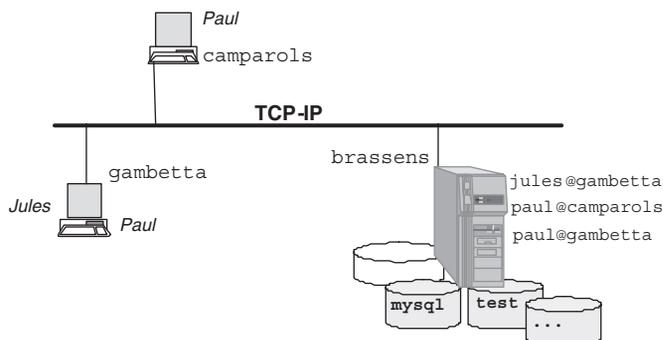
MySQL dénomme *host* la machine hébergeant le SGBD. MySQL diffère aussi à ce niveau des autres SGBD, car il est possible de distinguer des accès d'un même utilisateur suivant qu'il se

connecte à partir d'une machine ou d'une autre. La notion d'identité est basée sur le couple nom d'utilisateur MySQL (*user*) côté serveur, machine cliente.

Identities

Ainsi, l'utilisateur Paul, se connectant depuis la machine *camparols*, peut ne pas être le même que celui se connectant depuis la machine *gambetta*. S'il s'agit du même, il faudra, au niveau du serveur, éventuellement composer un ensemble de prérogatives équivalent pour les deux accès (voir le chapitre 5). S'il s'agit de deux personnes différentes, il faudra distinguer les attributions des différents droits. La figure suivante illustre le fait que deux utilisateurs peuvent se connecter par deux accès différents. Trois identités seront donc à créer et à administrer côté serveur.

Figure I-4 Notion d'hôte MySQL



Le chapitre 5 décrit la configuration d'un serveur et de clients. Reportez-vous également au site d'accompagnement de l'ouvrage qui présente quelques outils graphiques d'aide à l'administration.

Accès à MySQL

Une fois que vous aurez installé MySQL sur votre ordinateur, vous serez libre de choisir l'accès qui vous convient. Ce livre utilise essentiellement l'interface en ligne de commandes fournie avec le SGBD, mais aussi Java via JDBC, et le navigateur Web au travers de PHP.

Aspects étudiés

Nous n'étudierons pas tous les concepts d'un serveur MySQL, car certains sont assez spécifiques. Le tableau suivant indique le chapitre du livre dans lequel vous trouverez le descriptif du concept étudié.

Tableau I-3 Éléments d'une base MySQL

Éléments étudiés – Chapitre	Aspects non étudiés
Tables et index – Chapitre 1	
Séquences – Chapitres 2 et 5	Haute disponibilité
Vues (<i>views</i>) et utilisateurs – Chapitre 5	Réplication
Déclencheurs (<i>triggers</i>) – Chapitre 7	Sauvegardes et restaurations
Fonctions et procédures (<i>stored programs</i>) – Chapitre 7	Données spatiales (GIS)
Mode document (NoSQL) – Chapitre 7	
Moteurs de stockage (<i>storage engines</i>) – Chapitre 10	
Partitionnement – Chapitre 10	

Décrivons à présent l'installation du SGBD sous Windows. À noter que les options d'installation sont identiques sous Linux.

Mise en œuvre de MySQL sous Windows

Si la procédure se déroule sans problème, vous n'aurez besoin que de quelques minutes pour installer MySQL. Je vous conseille toutefois de créer un point de restauration pour pouvoir revenir si besoin à la dernière bonne configuration connue.

Pour obtenir les dernières versions en cours de développement et celles de production, rendez-vous sur le site <http://www.mysql.com/downloads>, puis sélectionnez MySQL Community Server.

La première étape consiste à sélectionner votre plate-forme (qui va de Windows à Mac OS en passant par différentes éditions de Linux). Pour pouvoir télécharger le logiciel et poster sur les forums officiels, vous devez vous enregistrer (si vous disposez déjà d'un compte Oracle, c'est le même). Dans le cas de Windows, le logiciel MySQL est fourni avec un fichier d'installation (*MSI Installer*) qu'il convient d'utiliser.

Installation

Après avoir accepté les termes de la licence, vous êtes invité à choisir le répertoire d'installation (par défaut `Program Files\MySQL`) et le type d'installation (par défaut *Developer*). Ensuite, une étape de mise à jour de composants et de vérification des prérequis s'opère, au cours de laquelle vous pouvez agir avant le récapitulatif des produits qui seront installés.

La configuration de base inclut :

- le mot de passe de l'utilisateur `root` qui permet d'ajouter des nouveaux utilisateurs en leur accordant des rôles prédéfinis (administrateur, concepteur, importateur, sauvegarde, etc.) ;
- un service Windows dédié à MySQL (nommé par défaut `MySQLxx`, qu'il sera possible d'arrêter via le panneau de configuration) ;

- le port UDP d'écoute (par défaut 3306), vous pouvez ainsi ajouter une exception à votre pare-feu (option cochée par défaut).

Suivant la version choisie, la configuration avancée vous permettra d'agir sur :

- le type de votre serveur (machine de développement, serveur ou machine dédiée) ;
- les répertoires qui contiendront les fichiers de trace ;
- L'inclusion du chemin de l'exécutable `mysql` dans la variable d'environnement (`path`) ;
- le mode comportemental du serveur par rapport à la syntaxe des instructions SQL ;
- le type de base de données (multifonction, mode transactionnel ou pas) ;
- nombre de connexions (15 par défaut) ;
- le jeu de caractères (West European pour nous).

Il est possible d'ajouter un utilisateur MySQL (autre que `root`) lors de l'installation mais ce n'est pas la peine de le faire car il vaut mieux le faire ajouter une fois l'installation terminée.

Une fois MySQL installé, et suivant la version choisie et celle de votre Windows, c'est peut-être la fête des répertoires car des arborescences de MySQL risquent d'apparaître dans les trois répertoires suivants : `C:\Program Files`, `C:\ProgramData\MySQL` (par défaut invisible dans l'explorateur) et `C:\Program Files (x86)`. Le sous-répertoire `MySQL Server x.x` indique la version de votre serveur.

Le fichier de configuration `my.ini` (qui se trouve en principe dans `C:\ProgramData\MySQL`) est stratégique : il contient un grand nombre de paramètres relatifs au serveur. Si vous devez modifier un paramètre, vous devrez redémarrer le service. Un conseil toutefois : copiez ce fichier initial pour pouvoir annuler vos tentatives de modifications en cas de refus de démarrage du service MySQL.

Vos bases de données se trouveront dans le répertoire indiqué par la variable `datadir` contenue dans le fichier `my.ini`. Il s'agit de `C:\ProgramData\MySQL\MySQL Server x.x\Data` par défaut pour Windows 7 et de `C:\Documents and Settings\All Users\Application Data\MySQL\MySQL Server x.x` pour Windows XP.

Si cela n'a pas été fait, ajoutez le chemin `C:\Program Files\MySQL\MySQL Server x.x\bin` à votre variable d'environnement `path` (système ou utilisateur). Pour XP, Poste de travail/Propriétés/Avancé, puis Variables d'environnement en choisissant la variable système `path`. Pour Vista ou Seven, le poste de travail s'appelle *Ordinateur*, ensuite vous trouverez le même processus via Propriétés/Paramètres système avancés. N'oubliez pas d'ajouter le point-virgule pour séparer votre nouveau chemin du dernier des différents chemins existants.

Premiers pas

Cette section va guider vos premiers pas pour travailler avec l'interface de commandes en ligne. Il s'agira de stocker les fichiers qui vous serviront à effectuer différentes actions (créations de tables, de vues ou d'utilisateurs, insertions, modifications ou suppressions d'enregistrements, élaboration de requêtes, de procédures cataloguées, etc.).

L'interface de commande

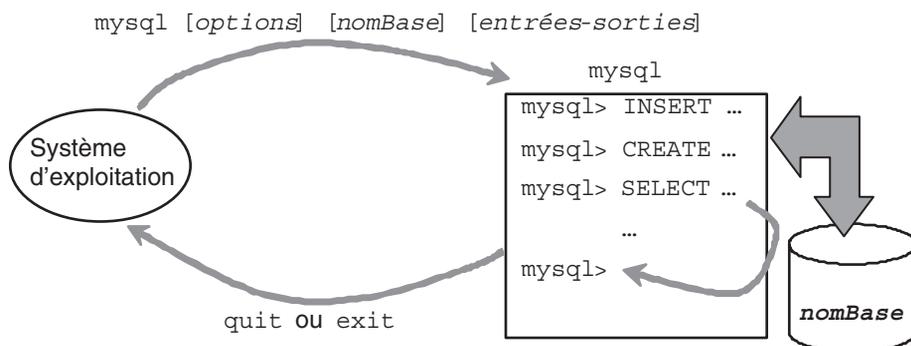
L'interface en ligne de commande se lance grâce à l'exécutable `mysql`. Cette interface ressemble à une fenêtre DOS ou Telnet et permet de dialoguer très simplement avec la base de données. L'utilisation peut être interactive ou en mode *batch*. Dans le premier cas (c'est le mode le plus courant), le résultat des extractions est présenté sous une forme tabulaire au format ASCII.

Vous verrez qu'il est notamment possible :

- d'exécuter des instructions SQL (créer des tables, manipuler des données, extraire des informations, etc.) ;
- de compiler des procédures cataloguées et des déclencheurs ;
- de réaliser des tâches d'administration (création d'utilisateurs, attribution de privilèges, etc.).

Le principe général de l'interface est le suivant : après une connexion locale ou distante, des instructions sont saisies et envoyées à la base qui retourne des résultats affichés dans la même fenêtre de commande.

Figure I-5 Principe général de l'interface en ligne de commande





N'ayez pas honte de bien maîtriser cette interface au lieu de connaître toutes les options d'un outil graphique (comme *PhpMyAdmin*, *MySQL Administrator* ou autre). Il vous sera toujours plus facile de vous adapter aux différents boutons et menus, tout en connaissant les instructions SQL, que l'inverse.

Imaginez-vous un jour à Singapour sur une machine ne disposant d'aucun outil graphique et qu'un client vous demande la réduction que vous pouvez lui faire sur la piscine intérieure d'un Airbus A380. Vous devez interroger une table sur le serveur du siège social à Blagnac et vous ne savez pas vous servir de l'interface en ligne de commande : vous n'êtes pas un informaticien !

Création d'un utilisateur

Vous allez maintenant créer un utilisateur MySQL. Pour cela, ouvrez le fichier `premierPas.sql` qui se trouve dans le répertoire `Introduction`, à l'aide du bloc-notes (ou d'un éditeur de texte de votre choix). Remplacez « util » par le nom de l'utilisateur à créer (modifiez aussi le nom de la base). Vous pouvez également changer le mot de passe si vous le souhaitez. Enregistrez ce fichier dans l'un de vos répertoires.

Connexion au serveur

Dans une fenêtre de commande Windows, Linux (ou autre), lancez l'interface en ligne de commandes en connectant l'utilisateur `root` avec le mot de passe spécifié lors de l'installation :

```
mysql --user=root -p
```

Figure I-6 Interface en mode ligne de commande

```

C:\Users\soutou>mysql --user=root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.7.13-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

```

Une fois connecté, par copier-coller (en effectuant un clic droit dans la fenêtre de commande MySQL), exécutez une à une les différentes instructions (création de la base, de l'utilisateur, des

privilèges et déconnexion de l'utilisateur `root`). Nous étudierons au chapitre 5 les notions élémentaires de droits et de sécurité. Les lignes encadrées par les signes `/*` et `*/` sont des commentaires. Votre utilisateur (*util*) est désormais créé, il peut se connecter et il possède toutes les prérogatives sur la base (*bdutil*) pour exécuter les instructions décrites dans cet ouvrage. Pour tester votre connexion, lancez la commande suivante qui se connecte au serveur sur la base *bdutil*, sous l'utilisateur *util*.

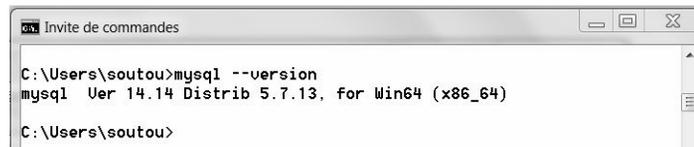
```
mysql --user=util --host=localhost -p --database=bdutil
```

Vérification de la version

Pour contrôler la version de votre serveur, exécutez la connexion-déconnexion suivante dans une fenêtre de commande (Windows ou Linux) :

```
mysql --version
```

Figure I-7 Version du serveur MySQL



Si vous êtes déjà connecté, la commande `SELECT VERSION()` ; vous renseignera de la même manière à ce propos.

Options de base

Les principales options au lancement de `mysql` sont résumées dans le tableau suivant :

Tableau I-4 Principales options de la commande `mysql`

Option	Commentaire
<code>--help</code> ou <code>-?</code>	Affiche les options disponibles, l'état des variables d'environnement et rend la main.
<code>--batch</code> ou <code>-B</code>	Toute commande SQL peut être lancée dans la fenêtre de commandes système sans pour autant afficher l'invite. Les résultats (colonnes) sont séparés par des tabulations.
<code>--database=nomBD</code> ou <code>-D nomBD</code>	Sélection de la base de données à utiliser après la connexion.
<code>--host=nomServeur</code> ou <code>-h nomServeur</code>	Désignation du serveur.

Tableau I-4 Principales options de la commande `mysql` (suite)

Option	Commentaire
<code>--html</code> ou <code>-H</code>	Formate le résultat des extractions en HTML.
<code>--one-database</code> ou <code>-O</code>	Restreint les instructions à la base de données spécifiée initialement.
<code>-p</code>	Demande le mot de passe sans l'employer en tant que paramètre.
<code>--password=motdePasse</code>	Transmission du mot de passe de l'utilisateur à connecter. Évitez le plus possible cette option et préférez-lui la précédente.
<code>--prompt=parametre</code>	Personnalise l'invite de commande (par défaut <code>mysql></code>).
<code>--silent</code> ou <code>-s</code>	Configure le mode silence pour réduire les messages de MySQL.
<code>--skip-column-names</code> ou <code>-N</code>	N'écrit aucun en-tête de colonne pour les résultats d'extraction.
<code>--table</code> ou <code>-t</code>	Formate le résultat des extractions en tables à en-tête de colonne (par défaut dans le mode interactif).
<code>--tee=cheminNomFichier</code>	Copie la trace de toute la session dans le fichier spécifié.
<code>--user=utilisateur</code> ou <code>-u utilisateur</code>	Désigne l'utilisateur devant se connecter.
<code>--verbose</code> ou <code>-v</code>	Mode verbeux pour afficher davantage de messages du serveur.
<code>--version</code> ou <code>-V</code>	Affiche la version du serveur et rend la main.
<code>--vertical</code> ou <code>-E</code>	Affiche les résultats des extractions verticalement (non plus en lignes horizontales).
<code>--xml</code> ou <code>-X</code>	Formate le résultat des extractions en XML. Les noms de balises générées sont <code><resultset></code> pour la table résultat, <code><row></code> pour chaque ligne et <code><field></code> pour les colonnes.

Ces options peuvent se combiner en les séparant simplement par un espace (exemple : `mysql --tee=D:\dev\sortiemysql.txt --database=bdsoutou` va se connecter anonymement à la base `bdsoutou` en inscrivant le contenu de la trace de la session dans le fichier `sortiemysql.txt` situé dans le répertoire `D:\dev`).

Batch

Pour lancer plusieurs commandes regroupées dans un fichier ayant pour extension `.sql`, il faut préciser le chemin du fichier et celui qui contiendra les éventuels résultats. Ainsi, l'instruction suivante exécute dans la base `bdsoutou`, sous l'autorité de l'utilisateur `soutou`, les commandes contenues dans le fichier `Testbatch.sql` situé dans le répertoire `D:\dev` (notez l'utilisation du double *back-slash* pour désigner une arborescence Windows). Le résultat sera consigné dans le fichier `sortie.txt` du même répertoire.

```
mysql --user=soutou --password=iut bdsoutou
<D:\\dev\\Testbatch.sql >D:\\dev\\sortie.txt
```

Ici, la saisie du mot de passe n'est pas possible et il convient de le passer en clair dans la commande.

Votre prompt, et vite !

L'exécution de l'instruction `mysql --prompt="(\\u@\\h) [\\d]> " --user=root -p` dans une fenêtre de commande *shell* ou DOS connectera l'utilisateur `root` en lui demandant son mot de passe. L'invite de commande à l'affichage sera de la forme suivante : `(root@localhost) [bdsoutou]>` une fois que `root` aura sélectionné la base `bdsoutou` (par la commande `use nombase;`).

Le tableau suivant résume les principaux paramètres pour afficher les invites de commande (relatives à l'option `prompt`).

Tableau I-5 Principales options de la commande `mysql`

Option	Commentaire
<code>\\v</code>	Version du serveur
<code>\\d</code>	Base de données en cours d'utilisation
<code>\\h</code>	Nom du serveur
<code>\\u</code>	Nom d'utilisateur
<code>\\U</code>	Nom d'utilisateur long (au format <i>nom@serveur</i>)
<code>_</code>	Un espace
<code>\\R</code>	Heure (0 à 23)
<code>\\m</code>	Minutes
<code>\\s</code>	Secondes
<code>\\Y</code>	Année sur quatre chiffres
<code>\\D</code>	Date en cours
<code>\\c</code>	Compteur d'instructions



Pour personnaliser l'invite de commandes, deux possibilités s'offrent à vous. La première consiste à créer la variable d'environnement de nom `MYSQL_PS1` et de valeur `(\\u@\\h) [\\d] mysql>`. La seconde solution nécessite d'ajouter une entrée au fichier de configuration `my.ini`, les lignes suivantes sous la section `[mysql]` :

```
#mon prompt
prompt=(\\u@\\h) [\\d]\\_mysql>\\_
```

Par ailleurs, si `my.ini` se trouve dans `C:\ProgramData\MySQL`, vous aurez peut-être besoin de dupliquer ce fichier dans le répertoire `C:\Programmes\MySQL\MySQL Server x.x` afin de bénéficier de votre prompt personnalisé.

Une fois le serveur redémarré, en considérant que votre utilisateur se nomme `util` et la base `bdutil`, le prompt précédant chacune de vos commandes SQL devrait être le suivant : `(util@localhost) [bdutil] mysql>`.

Commandes de base

Une fois connecté, vous pouvez utiliser des commandes ou effectuer des copier-coller d'un éditeur de texte dans l'interface `mysql` (ce moyen de faire correspond plus à un environnement de test qui conviendra à l'apprentissage). Le tableau suivant résume les principales instructions pour manipuler le *buffer* d'entrée de l'interface.

Tableau I-6 Commandes de base du buffer d'entrée

Commande	Commentaire
<code>?</code>	Affichage des commandes disponibles.
<code>delimiter chaîne</code>	Modifie le délimiteur (par défaut <code>;</code>).
<code>use nomBase</code>	Rend une base de données courante.
<code>prompt chaîne</code>	Modifie l'invite de commande avec les paramètres vus précédemment.
<code>quit</code> ou <code>exit</code>	Quitte l'interface.
<code>source cheminNomFichier.sql</code>	Charge et exécute dans le buffer le contenu du fichier <code>cheminNomFichier.sql</code> (par exemple, <code>source D:\dev\Testbatch.sql</code> exécutera le script <code>Testbatch.sql</code> situé dans <code>D:\dev</code>).
<code>tee nomFichierSortie</code>	Création du fichier <code>nomFichierSortie</code> dans le répertoire <code>C:\Program Files\MySQL\MySQL Server n.n\bin</code> qui contiendra la trace de la session.

La commande `source` est très utile afin d'éviter les copier-coller de trop nombreuses instructions.