



Approche de SQL

Introduction à SQL

Fabien Coelho, Claire Medrala

Mines Paris – PSL, MESR

Janvier 2025



[Intro SQL](#)
[FC/CM](#)
[What/How](#)
[SQL](#)
[Syntaxe](#)
[Histoire](#)
[Standards](#)
[Dialectes](#)

1 / 9

2 / 9

Quoi vs Comment

- langage **déclaratif**, description du résultat
 - description basée sur l'algèbre relationnelle
 - Turing complet avec requêtes *récursives*
- **pas de programmation !** (au sens classique)
 - pas de contrôle explicite : `if else while for...`
 - pas d'interaction : `print read...`
 - pas de fonctions ? CTE, PL/pgSQL...
- administration des performances
 - *comment* déterminé par l'optimiseur
 - indexation des tables...



Principes syntaxiques

SQL

Structured Query Language

[Intro SQL](#)
[FC/CM](#)
[What/How](#)
[SQL](#)
[Syntaxe](#)
[Histoire](#)
[Standards](#)
[Dialectes](#)

nombreux dialectes, variante PostgreSQL présentée

- casse** indifférente, **mais par convention :**
 - MAJUSCULES** mots clefs et fonctions de SQL
 - minuscules** identificateurs, longueur max 63, double quotes "
- verbe** désignant l'opération à effectuer
 - `SELECT DELETE CREATE ALTER SET COMMIT...`
- clauses** ordonnées précisant l'opération
 - `FROM WHERE GROUP HAVING LIMIT OFFSET`
- expressions** types, constantes, opérateurs, fonctions...
 - `INTEGER BOOL 123 'hello' AND + SQRT() SIN()`
- commentaire** `-- ceci est un commentaire`
- terminaison** par un point-virgule ;

3 / 9

4 / 9

Fonctions

DDL Data Definition Language	<code>CREATE ALTER DROP</code>
DML Data Manipulation Language	<code>SELECT INSERT UPDATE DELETE</code>
DCL Data Control Language	<code>GRANT REVOKE COMMIT ROLLBACK SET</code>

Objets

utilisateurs	<code>ROLE USER GROUP</code>
catalogues	<code>DATABASE SCHEMA</code>
données	<code>TABLE SEQUENCE VIEW</code>
divers	<code>TRIGGER RULE CONSTRAINT INDEX</code>
extensions	<code>CONVERSION AGGREGATE CAST FUNCTION</code>



Évolution

[Intro SQL](#)
[FC/CM](#)
[What/How](#)
[SQL](#)
[Syntaxe](#)
[Histoire](#)
[Standards](#)
[Dialectes](#)



des Standards...

[Intro SQL](#)
[FC/CM](#)
[What/How](#)
[SQL](#)
[Syntaxe](#)
[Histoire](#)
[Standards](#)
[Dialectes](#)

Historique

- 1970 modèle relationnel par Codd, langage Alpha
- 1974 QUEL langage pour Ingres, basé sur Alpha
- 1974 Sequel (*Structured English Query Language – IBM*)
- 1976 Sequel 2 = SQL, toujours prononcé *sequel*
- 1982 début de standardisation ANSI

Standardisation

- ANSI American National Standards Institute
X3.135, versions de 1986, 1989, 1992, 1999
- NIST (US) National Institute of Standards and Technology
- ISO International Standards Organization ISO/IEC 9075:1992...

5 / 9

1986 ANSI – bases

- 1989 ANSI/ISO – intégrité référentielle
- 1992 ANSI/ISO – SQL2 extensions
 - 3 niveaux *Entry, Intermediate, Full*
 - puis *Transitional* (NIST)
- 1999 SQL3 réorganisation avec 9 extensions *Core, Enhanced*
- 2003 XML, séquences...
- 2006 XML (suite)...
- 2008 MERGE TRUNCATE...
- 2011 gestion du temps (*temporal databases*)
- 2016 JSON, ...
- 2023 JSON amélioré, graph ...

6 / 9



ISO/IEC 9075:2023

SQL:2023



... et des dialectes !

[Intro SQL](#)
[FC/CM](#)
[What/How](#)
[SQL](#)
[Syntaxe](#)
[Histoire](#)
[Standards](#)
[Dialectes](#)

[Intro SQL](#)
[FC/CM](#)
[What/How](#)
[SQL](#)
[Syntaxe](#)
[Histoire](#)
[Standards](#)
[Dialectes](#)

- Part 1 Framework
- Part 2 Foundation
- Part 3 Call-Level Interface
- Part 4 Persistent Stored Modules
- Part 5 Row Pattern Recognition
- Part 6 Support for JSON
- Part 7 Polymorphic table functions
- Part 9 Management of External Data
- Part 10 Object Language Bindings
- Part 11 Information and Definition Schemas
- Part 13 Routines and types using Java
- Part 14 XML-Related Specifications
- Part 15 Multi-dimensional arrays
- Part 16 Property graph queries

- SQL/Framework
- SQL/Foundation
- SQL/CLI
- SQL/PSM
- SQL/JSON
- SQL/MED
- SQL/OLB
- SQL/Schemata
- SQL/JRT
- SQL/XML
- SQL/MDA
- SQL/PGQ

7 / 9

Respect très élastique de la norme

- | | |
|--------|--|
| MySQL | <ul style="list-style-type: none"> ■ pas de subquery avant v4 ■ pas de transactions avant v5 (et encore) ■ <code>information_schema</code> non standard... ■ pas d'opérateur différence ni intersection... |
| Oracle | <ul style="list-style-type: none"> ■ <code>NULL</code> et '' équivalents ■ pas de <code>AS</code> ■ <code>MINUS</code> au lieu de <code>EXCEPT</code> ■ syntaxe jointure (<code>WHERE, UNION, (+)</code>) |
| DB2 | <ul style="list-style-type: none"> ■ clef primaires <code>NULL</code> |
- (v8)

8 / 9



SQL – langage universel d'accès aux données ?

Intro SQL

FC/CM

What/How

SQL

Syntaxe

Histoire

Standards

Dialectes

Distributed DB K-V, document stores, ...

- Google BigTable/GFS
- Hadoop HBase/HDFS
- Cassandra...

Query avec *map-reduce*

- fonctions d'extraction et de combinaisons
- complexe, performances souvent médiocres

Pseudo SQL surcouche Google Spanner, Apache Hive et Spark

- simple, sémantique bien définie
- compilation/optimisation vers *map-reduce*
- performance pas forcément très bonnes...