

LE LANGAGE DE REQUETES SQL

Origines et Evolutions

SQL1 86: la base

SQL1 89: l'intégrité

SQL2 92: la nouvelle norme

SQL3 98: les évolutions objets

1. Origines et Evolutions

- SQL est dérivé de l'algèbre relationnelle et de SEQUEL
- Il a été intégré à SQL/DS, DB2, puis ORACLE, INGRES, ...
- Il existe trois versions normalisées, du simple au complexe :
 - SQL1 86 version minimale
 - SQL1 89 addendum (intégrité)
 - SQL2 (92) langage complet à 3 niveaux
- Une version 3 étendue (objets, règles) est en préparation.
- La plupart des systèmes supportent SQL1 complet

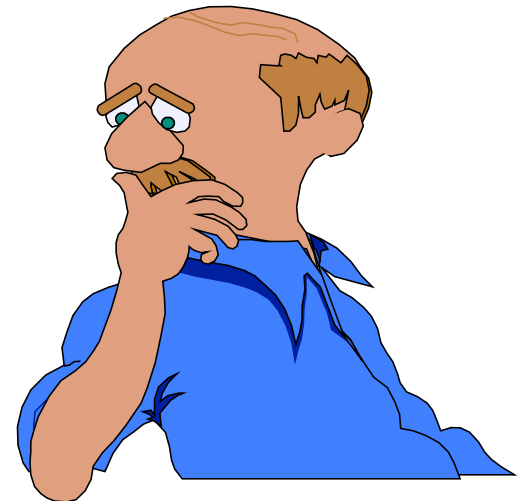
OPERATIONS

➤ Opérations de base

- select, insert, update, delete

➤ Opérations additionnelles

- définition et modification de schémas
- définition de contraintes d'intégrité
- définition de vues
- accord des autorisations
- gestion de transactions



ORGANISATION DU LANGAGE

- SQL comprend quatre parties :
- Le langage de définition de schéma (Tables, Vues, Droits)
- Le langage de manipulation (Sélection et mises à jour)
- La spécification de modules appelables (Procédures)
- L'intégration aux langages de programmation (Curseurs)

2. SQL1 - 86

➤ LANGAGE DE DEFINITIONS DE DONNEES

- CREATE TABLE
- CREATE VIEW

➤ LANGAGE DE MANIPULATION DE DONNEES

- SELECT OPEN
- INSERT FETCH
- UPDATE CLOSE
- DELETE

➤ LANGAGE DE CONTROLE DE DONNEES

- GRANT et REVOKE
- BEGIN et END TRANSACTION
- COMMIT et ROLLBACK

BASE DE DONNEES

Collection de tables et de vues dans un schéma

VITICULTEURS (NVT, NOM, PRENOM, VILLE, REGION)

VINS (NV, CRU, MILLESIME, DEGRE, NVT, PRIX)

BUVEURS (NB, NOM, PRENOM, VILLE)

ABUS (NV, NB, DATE, QTE)

GROS_BUVEURS (NB, NOM, PRENOM)



Exemples de Questions (1)

➤ Q1: Crus des vins sans doubles.

- SELECT DISTINCT CRU
- FROM VINS

➤ Q2: Noms des buveurs ayant bus des Beaujolais 87 ou 88.

- SELECT DISTINCT NOM
- FROM BUVEURS B, VINS V, ABUS
- WHERE B.NB = ABUS.NB
- AND ABUS.NV = V.NV
- AND CRU LIKE '%BEAUJOLAIS%'
- AND MILLESIME IN (1987, 1988)

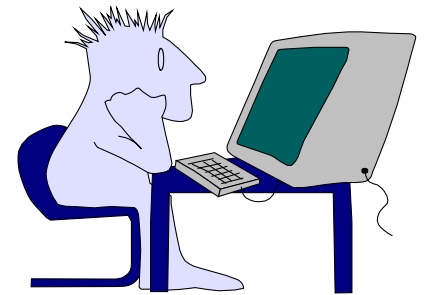
Exemples de Questions (2)

➤ Q3 : Noms et prénoms des buveurs de vins dont le cru commence par B, de degré inconnu ou compris entre 11 et 13.

- SELECT NOM, PRENOM
- FROM BUVEURS B, VINS V, ABUS A
- WHERE B.NB = A.NB AND A.NV = V.NV
- AND CRU LIKE "B%"
- AND (DEGRE BETWEEN 11 AND 13 OR DEGRE IS NULL)

➤ Q4 : Noms des crus bus par au moins un buveurs.

- SELECT DISTINCT CRU
- FROM VINS V
- WHERE EXISTS (SELECT *
- FROM BUVEURS B, ABUS A
- WHERE B.NB = A.NB AND A.NV = V.NV)



Exemples de Questions (3)

- Q5: Calculer le degré moyen pour chaque cru.
 - SELECT CRU, AVG(DEGRE)
 - FROM VINS
 - GROUP BY CRU

- Q6 : Calculer le degré moyen et le degré minimum pour tous les crus de 94 dont le degré minimum est supérieur à 12.
 - SELECT CRU, AVG(DEGRE), MIN(DEGRE)
 - FROM VINS
 - WHERE MILLESIME = 1994
 - GROUP BY CRU
 - HAVING MIN(DEGRE) > 12

Select: Forme Générale

- SELECT <liste de projection>
- FROM <liste de tables>
- [WHERE <critère de jointure> AND <critère de restriction>]
- [GROUP BY <attributs de partitionnement>]
- [HAVING <critère de restriction>]

➤ Restriction :

- arithmétique (=, <, >, ≠, ≥, ≤)
- textuelle (LIKE)
- sur intervalle (BETWEEN)
- sur liste (IN)

➤ Possibilité de blocs imbriqués par :

- IN, EXISTS, NOT EXISTS, ALL, SOME, ANY

COMMANDE INSERT

INSERT INTO <relation name>

[(attribute [,attribute] ...)]

{VALUES <value spec.> [, <value spec.>] ...

| <query specification>}

➤ Exemples

- INSERT INTO VINS (NV, CRU, MILLESIME)
- VALUES 112, "JULIENAS", NULL

- INSERT INTO BUVEURS (NB,NOM,PRENOM)
- SELECT NVT, NOM, PRENOM
- FROM VITICULTEURS
- WHERE VILLE LIKE '%DIJON%'

COMMANDE UPDATE

➤ UPDATE <relation name>
SET <attribute = {value expression | NULL}>
[<attribute> = {value expression | NULL}] ...
[WHERE <search condition>]

➤ EXEMPLE

- UPDATE ABUS
- SET QTE = QTE * 1.1
- WHERE ABUS.NV IN
- SELECT NV
- FROM VINS
- WHERE CRU = 'VOLNAY' AND MILLESIME =
1990

COMMANDE DELETE

➤ DELETE FROM <relation name>
[WHERE <search condition>]

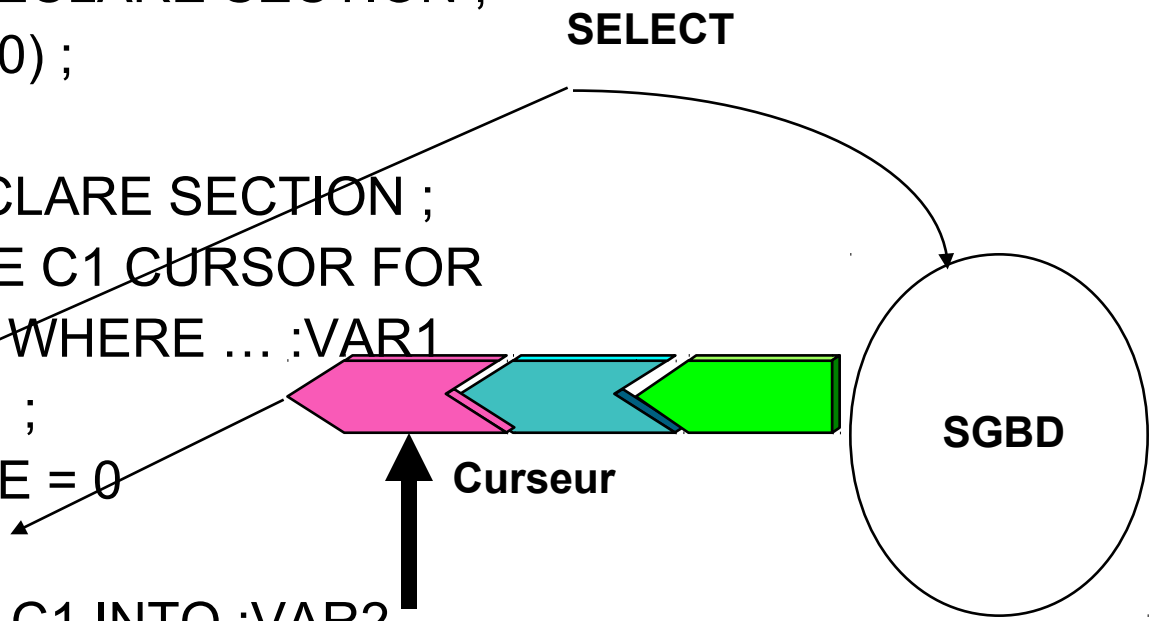
➤ EXEMPLE

- DELETE FROM ABUS
- WHERE NV IN
- SELECT NV
- FROM VINS
- WHERE DEGRE IS NULL

LP

- **Intégration de deux systèmes de types**
 - utilisation d'un pré-compilateur et d'une librairie
- **Passage de l'enssembliste au tuple à tuple**
 - utilisation de curseurs et Fetch
- **Exemple Program PL/1-SQL**

- EXEC SQL BEGIN DECLARE SECTION ;
- DCL VAR1 CHAR(20) ;
- DCL VAR2 INT ;
- EXEC SQL END DECLARE SECTION ;
- EXEC SQL DECLARE C1 CURSOR FOR
- SELECT ...FROM ... WHERE ... :VAR1
- EXEC SQL OPEN C1 ;
- DO WHILE SQLCODE = 0
- BEGIN
- EXEC SQL FETCH C1 INTO :VAR2



SQL1 - 89 : INTEGRITE

➤ VALEURS PAR DEFAULT

- CREATE TABLE VINS
- (NV INT UNIQUE,
- CRU CHAR(10),
- ANNEE INT,
- DEGRE FIXED (5,2) ,
- NVT INT,
- PRIX FIXED(7,2) DEFAULT 40)

➤ CONTRAINTES DE DOMAINES

- SALAIRE INT CHECK BETWEEN 6000 AND 100000

REFERENTIELLE

➤ CLE PRIMAIRE ET CONTRAINTE REFERENTIELLE

- CREATE TABLE VINS
- (NV INT PRIMARY KEY,
- CRU CHAR(10),
- ANNEE INT,
- DEGRE FIXED (5,2) ,
- NVT INT REFERENCES VITICULTEURS,
- PRIX DEFAULT 40)

➤ REFERENCE EN PRINCIPE LA CLE PRIMAIRE

- celle de VITICULTEURS

LA NORMALISATION DE SQL

➤ GROUPE DE TRAVAIL ANSI/X3/H2 et ISO/IEC JTC1/SC2

➤ DOCUMENTS ISO :

- SQL1 - 86 : Database Language SQL X3.135 ISO-9075-1987)
- SQL1 - 89 : Database Language SQL with Integrity Enhancement X3.168 ISO-9075-1989
- SQL2 - 92 : Database Language SQL2 X3.135 ISO-9075-1992

➤ ARGUMENTS POUR :

- Réduction des coûts d'apprentissage
- Portabilité des applications
- Longévité des applications
- Langage de communication inter-systèmes

➤ ARGUMENTS CONTRE :

- Manque de rigueur théorique
- Affaiblit la créativité

3. SQL2

➤ Trois niveaux distingués :

- Entry SQL2 = SQL89 + manques
- Intermediate SQL2 = Compléments relationnels
- Full SQL2 = Gadgets en plus

SQL2 Entry

- Codes réponses SQLSTATE
- Renommage des colonnes résultats
- Mots clés utilisables entre " "
- Métabase normalisée (schémas)

SQL2 Intermediate

➤ TYPES DE DONNEES DATE AVEC OPERATIONS

- DATE, TIME et TIMESTAMP
- Intervalles de temps

➤ CASCADE DES MISES A JOUR

- Suppression en cas d' intégrité référentielle avec options
 - Cascader les suppressions (CASCADE)
 - Rendre nul l'attribut référençant (NULLIFY)

➤ DIFFERENTS ALPHABETS ET ORDRES DE LETTRES

SQL2 Intermediate

➤ POSSIBILITE DE CREER DES DOMAINES

- CREATE DOMAINE MONEY IS DECIMAL (5,2)
- DEFAULT (-1)
- CHECK (VALUE = -1 OR VALUE > 0)
- NOT NULL

➤ JOINTURE EXTERNE (OUTER-JOIN)

- SELECT ...
- FROM R1 [NATURAL] [{LEFT | RIGHT}] JOIN R2 [ON (A=B)], ...
- WHERE ...

➤ EXPRESSIONS DE SELECT

- [OUTER] UNION
- INTERSECT
- EXCEPT

SQL2 Full

- Extension des dates et temps
- Expressions étendues avec correspondances de colonnes
- Possibilité de SELECT en argument d'un FROM
- Vues concrètes
- Contraintes d'intégrité multi-tables
- Contrôles d'intégrité différés

5. CONCLUSION

- Un standard de plus en plus complet et de plus en plus suivi
 - Attention aux approximations et imitations incomplètes
 - Tout existe dans les propositions SQL2 ou SQL3
 - Une référence pour implémenter et utiliser chaque aspect des BD
- Le langage de communication inter-système
 - RDA, TP, SQL ACCESS GROUP (CLI, FAP)
- Le langage universel sur lequel s'appuie les progiciels
- SQL réussira-t-il à bien intégrer l'objet ?