

Langage SQL

1

Cours pour les Master I

Mr. A. KHELAIFA

Définition

2

- C'est une norme de l'ANSI (SQL-92) et de l'ISO (SQL2) pour les SGBD relationnels
- Il comporte a la fois des énoncés de définitions des données (**LDD**) et de manipulation des données (**LMD**).
- il existe 3 niveaux de conformité dans la norme SQL-92: basique , Intermédiaire, complète;

La Table dans SQL

3

- une table est un ensemble de **tuples**;
- on utilise aussi **relation** comme synonyme de table, et **ligne** ou **enregistrement** comme synonymes de tuple;
- tous les tuples d'une table ont le même format; ce format est défini par un ensemble d'**attributs**;

La Table dans SQL

4

la définition en SQL d'une table comporte les éléments suivants:

- son nom
- ses attributs
- ses contraintes d'intégrité; il y a trois types de contraintes d'intégrité:
 - clé primaire;
 - clé unique;
 - clé étrangère.

Types de données (ANSI SQL)

5

- **INTEGER/INT:** permet de définir un nombre entier.
- **NUMERIC(p, s):**
p : le nombre total de chiffres stockés pour un nombre
s : nombre total de chiffres après la virgule;
- **REAL:** permet de stocker un nombre en virgule flottant.
- **CHAR(n) :** représente une chaîne de caractères de longueur fixe n (La chaîne 'abc12' est une valeur du type *CHAR(5)*).
- **VARCHAR(n):** stocker une chaîne de caractères de longueur maximale n; par rapport a *CHAR(n)*, permet de mieux gérer l'espace disque si les chaînes de caractères ne sont pas toujours de longueur n;

Types de données (ANSI SQL)

6

- **DATE:** comprend une date en format YYYY-MM-DD.
- **TIME:** comprend une heure en format HH:MM:SS.
- **TIMESTAMP:** comprend la date et l'heure en format (YYYY-MM-DD HH:MM:SS.), ou représente une fraction de seconde;

Définition des tables

7

Notation utilisée pour décrire la syntaxe du langage SQL

- **MOT CLE :** mot réservé du langage SQL;
- « **symbole terminal** » : peut être remplacé par un identificateur ou une constante (nombre, chaîne de caractère, etc.);
- « **symbole non terminal** » : doit être remplacé par sa définition;
- **::= :** définition d'un symbole non terminal;
- " {" et " } " : équivalent des parenthèses en mathématiques;
- **+** : une ou plusieurs occurrences;
- ***** : zéro ou plusieurs occurrences;
- **[élément optionnel]**
- **| :** choix entre plusieurs options

Définition des tables

8

- **Syntaxe générale**
« creation-table » ::=
CREATE TABLE « nom-table » (
« liste-attributs »
[, « liste-contraintes »]
)
- **Définition des attributs**
« liste-attributs » ::=
« attributs » { , « attributs » } *
« attribut » ::=
« nom-attribut » « type » [DEFAULT « expression »]
[NOT NULL] [CHECK (« condition »)]

Définition des tables

9

- Définition des contraintes
 - « liste-contraintes » ::=
 - « contrainte » {, « contrainte » }
 - « contrainte » ::=
 - « cle-primaire » | « cle-unique » | « cle-etrangere »
- Clé primaire
 - « cle-primaire » ::=
 - CONSTRAINT « nom-contrainte » PRIMARY KEY (« liste-noms-attribut »)
 - il ne peut y avoir deux tuples avec les mêmes valeurs pour les attributs de la clé primaire;
 - on peut définir une seule clé primaire pour une table;
 - la valeur d'un attribut d'une clé primaire ne peut être NULL dans un tuple.

Définition des tables

10

- Clé unique
 - « cle-unique » ::=
 - CONSTRAINT « nom-contrainte » UNIQUE (« liste-noms-attribut »)
- on peut définir plusieurs clés uniques pour une table;
un attribut d'une clé unique peut être NULL, toutefois, la combinaison de tous les attributs non NULL doit être unique.

Définition des tables

11

- **Clé étrangère:** On dénote deux cas possibles:
 - ❑ faire référence à la clé primaire d'une autre table
 - « cle-etrangere » ::=
 - CONSTRAINT « nom-contrainte »
 - FOREIGN KEY (« liste-attributs »)
 - REFERENCES « nom-table-referencee »
 - [ON DELETE CASCADE]
 - ❑ faire référence à une clé unique d'une autre table
 - « cle-unique » ::=
 - CONSTRAINT « nom-contrainte »
 - FOREIGN KEY (« liste-attributs »)
 - REFERENCES « nom-table-referencee »
 - [(« liste-attributs-cle-unique »)]
 - [ON DELETE CASCADE]

Définition des tables

12

- les types de « liste-attributs » doivent être les mêmes que les types « liste-attributs-cle-unique »;
- pour chaque tuple de la table dont les attributs de clé étrangère sont tous différents de NULL, il doit exister un tuple dans « nom-table-referencee » avec la même valeur pour « liste-attributs »;
- ON DELETE CASCADE : si un tuple dans « nom-table-referencee » est supprimé, tous les tuples de la table qui le référencent sont aussi supprimés.

Modification des tables

13

- ALTER TABLE « relation » {
« ajout-attribut » |
« modification-attribut » |
« suppression-attribut » |
« ajout-contrainte » |
« suppression-contrainte » }
 - « ajout-attribut » ::= ADD (« liste-attributs »)
 - « modification-attribut » ::= MODIFY (« liste-attributs »)
 - « suppression-attribut » ::= DROP (« liste-noms-attribut »)
 - « ajout-contrainte » ::= ADD (« liste-contrainte »)
 - « suppression-contrainte » ::= DROP « nom-contrainte »
[CASCADE]

Modification des tables

14

- On peut modifier le type, la valeur par défaut ou l'option NULL or NOT NULL des attributs de la liste en spécifiant seulement les parties à modifier;
- pour modifier le type, la valeur de chaque attribut doit être NULL pour tous les tuples de la table; et pour spécifier NOT NULL, il faut que l'attribut satisfasse déjà cette condition.
- CASCADE : supprime aussi toutes les contraintes qui dépendent de la contrainte supprimée.

Suppression des tables

15

- DROP TABLE « nom-table » [CASCADE CONSTRAINTS]

Langage de manipulation des données

16

- **Insert**
INSERT INTO « nom-table »
[(« liste-noms-attribut »)]
{ VALUES (« liste-expressions ») | « select » }
- **Update**
UPDATE « nom-table »
SET { « liste-affectation » | « affectation-select » }
[WHERE « condition »]
« liste-affectation » ::=
« affectation » [, « affectation »*]
« affectation » ::=
« nom-attribut » = « expression »
« affectation-select » ::=
(« liste-noms-attribut ») = « select »
« expression » peut être un énoncé select.

Langage de manipulation des données

17

- **Delete**

DELETE FROM « nom-table »

[WHERE « condition »]

si WHERE n'est pas spécifié, l'énoncé DELETE supprime tous les tuples.

- **Select**

Syntaxe générale

« enonce-select-base » ::=

SELECT [DISTINCT] « liste-expressions-colonne »

FROM « liste-expressions-table »

[WHERE « condition-tuple »]

[GROUP BY « liste-expressions-colonne »]

[HAVING « condition-groupe »]

[ORDER BY « liste-expressions-colonne »]

Langage de manipulation des données

18

« enonce-select-compose » ::=

« enonce-select-base »

{ UNION [ALL] | INTERSECT | MINUS }

« enonce-select-compose »

Langage de manipulation des données

19

- Chaque SGBD utilise un algorithme propre pour exécuter un énoncé SELECT. Toutefois, le résultat est le même que celui donné par la procédure

1. évaluer le produit cartésien des relations du FROM;
2. sélectionner les tuples satisfaisant la clause WHERE;
3. regrouper les tuples selon la clause GROUP BY;
4. sélectionner les groupes selon la condition HAVING;
5. évaluer les expressions du SELECT;
6. éliminer les doublons si clause DISTINCT;
7. évaluer l'union, l'intersection, ou la différence des selects (si nécessaire);
8. trier les tuples selon la clause ORDER BY.

Langage de manipulation des données

20

- **Fonctions d'agrégation:** appliquée à l'ensemble des tuples d'un select;
 - ❑ count(« expr »), sum(« expr »), avg(« expr »), min(« expr »), max(« expr »), etc;
 - ❑ count(*) : compte aussi les valeurs NULL;
 - ❑ count(attribut) : compte seulement les valeurs non NULL;
 - ❑ count(distinct attribut) : une valeur est comptée une seule fois, même si plusieurs tuples ont cette valeur;
 - ❑ GROUP BY : fonction appliquée aux groupes, plutôt qu'à l'ensemble du select.

Langage de manipulation des données

21

- **Operations ensemblistes**
- ❑ UNION : union de tous les tuples des subselects avec élimination des doublons;
- ❑ UNION ALL : union de tous les tuples des subselects sans élimination des doublons;
- ❑ INTERSECT : intersection avec élimination doublon;
- ❑ MINUS : différence, avec élimination doublon.

exemple de la bibliothèque

22

- le schéma relationnel
- 1. editeur : idediteur, nom, pays.
- 2. auteur : idauteur, nom.
- 3. livre: idlivre, titre, idauteur, idediteur, dateAcquisition, prix
- 4. membre: idmembre, nom, telephone, limitePret.
- 5. pret: idmembre, idlivre, datePret.
- 6. reservation: idreservation, idmembre, idlivre, dateReservation