

Bases de données

SQL

LMD – partie 1

Modification (Insert, Delete, Update)

BD104
v302a

2020-09-17

Département d'informatique
Faculté des sciences



Christina.Khnaisser@USherbrooke.ca
<http://info.USherbrooke.ca/ckhnaisser>
Luc.Lavoie@USherbrooke.ca
<http://info.USherbrooke.ca/llavoie>

PLAN

- INSERT
- DELETE
- UPDATE
- Exercices
- Incidences sur les clés référentielles
- Références
- Les colles du prof



LE LANGAGE SQL

INSERT (SYNTAXE)

insertion ::=

```
INSERT INTO nom_table
[ ( listeNomsColonne ) ]
{
  DEFAULT VALUES
| VALUES listeExpressionInit
| requête
}
[ suiteInsertion ]
```

listeExpressionInit ::=

```
{ tuple-val ... , }
```

tuple-val ::=

```
( { { expression | DEFAULT } ... , } )
```

LE LANGAGE SQL

DELETE (SYNTAXE)

retrait ::=

contexte

DELETE FROM [ONLY] *nomTable* [*]

[[AS] *alias*]

[USING *liste_using*]

[WHERE *condition* | *curseur*]

[*suiteRetrait*]

LE LANGAGE SQL

UPDATE

miseAJour ::=

contexte

UPDATE [ONLY] *nom_table* [*] [[AS] *alias*]

SET { *paire* , ... }

[FROM *liste_from*]

[WHERE *condition* | *curseur*]

suiteMiseAJour

paire ::=

nomCol = *val*

| ({ *nomCol* ... , }) = ({ *val* ... , })

val ::=

{ *expression* | DEFAULT }

EXEMPLES

- Évaluation
- Gaspard et Madeleine (à venir)

ÉVALUATION – RAPPELS

EXEMPLE DE DONNÉES

Activité		TypeÉvaluation	
sigle	titre	code	description
IFT 159	Analyse et programmation	IN	Examen intra
IFT 187	Éléments de bases de données	FI	Examen final
IMN 117	Acquisition des médias numériques	TP	Travail pratique
IGE 401	Gestion de projets	PR	Projet
GMQ 103	Géopositionnement		

Résultat					Étudiant		
matricule	TE	activité	trimestre	note	matricule	nom	adresse
15113150	TP	IFT 187	20133	80	15113150	Paul	>Δ ⁵ σ ^{5b}
15112354	FI	IFT 187	20123	78	15112354	Éliane	Blanc-Sablon
15113150	TP	IFT 159	20133	75	15113870	Mohamed	Tadoussac
15112354	FI	GMQ 103	20123	85	15110132	Sergeï	Chandler
15110132	IN	IMN 117	20123	90			
15110132	IN	IFT 187	20133	45			
15112354	FI	IFT 159	20123	52			

ÉVALUATION

INSERTION

- Ajouter un nouveau type d'évaluation

```
INSERT INTO TypeEvaluation (code, description)
VALUES ('TD', 'Travail dirigé');
```

- Ajouter une nouvelle activité au programme

```
INSERT INTO Activite (sigle, titre)
VALUES ('IFT339', 'Structure de données');
```

- Ajouter le résultat de l'examen final de Paul

```
INSERT INTO Resultat
VALUES('15113150', 'FI', 'IFT187', '20133', 75);
-- L'omission de la liste des colonnes n'est pas
-- une bonne idée, pourquoi ?
```


LE LANGAGE SQL

INSERT (EXEMPLES)

- INSERT INTO Resultat VALUES – êtes-vous certain de l'ordre ?
('15113150', '20133', 'IFT187', 'TP', 80);
- INSERT INTO Resultat
(matricule, activite, trimestre, TE, note)
VALUES
('15113150', 'IFT187', '20133', 'TP', 80);
- INSERT INTO Resultat
(matricule, activité, trimestre, TE, note)
VALUES
('15112354', 'IFT187', '20123', 'FI', 78),
('15113150', 'IFT159', '20133', 'TP', 75),
('15112354', 'GMQ103', '20123', 'FI', 85),
('15110132', 'IMN117', '20123', 'IN', 90),
('15110132', 'IFT187', '20133', 'IN', 45),
('15112354', 'IFT159', '20123', 'FI', 52);

ÉVALUATION RETRAIT

- Retirer les activités IMN du catalogue

```
DELETE FROM Activite  
WHERE SUBTR(sigle,1,3) = 'IMN';
```

- Retirer le type d'évaluation TP

```
DELETE FROM TypeEvaluation  
WHERE code = 'TP';
```

- Retirer les notes des TP

```
DELETE FROM Resultat  
WHERE TE = 'TP';
```

LE LANGAGE SQL

DELETE (EXEMPLES)

- DELETE FROM Resultat ;
- DELETE FROM Resultat WHERE trimestre = '20123' ;

ÉVALUATION MISE À JOUR

- Ajouter 10% pour tous les étudiants

```
UPDATE Resultat SET note = note + (note * 0,1);
```

ou encore

```
UPDATE Resultat SET note = note * 1,1;
```

- Retirer 5 points à la note de TP de tous les étudiants de l'activité 'IFT187'

```
UPDATE Resultat SET note = note - 5
```

```
WHERE TE = 'TP' AND activite = 'IFT187';
```

LE LANGAGE SQL

UPDATE (EXEMPLES)

```
○ UPDATE Resultats
  SET note = 90
  WHERE matricule = '15113150'
     AND activite = 'IFT187'
     AND trimestre = '20133'
     AND TE = 'TP'
  ;
```

ÉVALUATION

MISE À JOUR AVEC JOINTURE

- Retirer 5 points à l'examen final d'Éliane pour son cours 'IFT187'

```
UPDATE Resultat
  SET note = note - 5
FROM Etudiant
WHERE Etudiant.matricule=Resultat.matricule
  AND nom = 'Éliane'
  AND activite = 'IFT187'
  AND TE = 'FI';
```

INCIDENCE SUR LES CLÉS RÉFÉRENTIELLES

- Que peut-il se passer ?

QUE PEUT-IL SE PASSER ?

- Que se passe-t-il quand
 - un tuple référencé est détruit ?
 - un tuple référencé est modifié ?
 - un tuple référent est détruit ?
 - un tuple référent est modifié ?

INCIDENCES SUR LES CLÉS RÉFÉRENTIELLES

CREATE TABLE — LES ACTIONS

defContrainte ::=

```
[ CONSTRAINT nomContrainte ]  
{  
  CHECK (condition) - - voir module BD102-SQL-LDD-02  
| PRIMARY KEY ( listeNomsColonne )  
| UNIQUE ( listeNomsColonne )  
| foreignKey  
}
```

foreignKey ::=

```
FOREIGN KEY ( listeNomsColonne )  
REFERENCES nomTable [ ( listeNomsColonne ) ]  
[ MATCH { SIMPLE | PARTIAL | FULL } ]  
[ ON UPDATE action ]  
[ ON DELETE action ]
```

action ::=

```
CASCADE | SET NULL | SET DEFAULT | NO ACTION
```

INCIDENCES SUR LES CLÉS ÉTRANGÈRES

NOTE (1/2)

- Les termes

 - [**ON UPDATE** action]

 - [**ON DELETE** action]

permettent de spécifier une action lorsqu'une clé de la contrainte référentielle est modifiée. Les actions possibles sont

CASCADE | SET NULL | SET DEFAULT | NO ACTION

- L'action ainsi spécifiée est appliquée aux tuples dépendants consécutivement à une modification du tuple référencé.

INCIDENCES SUR LES CLÉS RÉFÉRENTIELLES

NOTE (2/2)

Le terme

[MATCH { SIMPLE | PARTIAL | FULL }]

permet de contrôler le traitement des NULL lors de la comparaison des clés :

- **si** vous avez suivi nos conseils, tous vos attributs (a fortiori ceux participant à une clé) sont NOT NULL... c'est donc inutile;
- **sinon devez** prendre en compte cet extrait de la norme ISO :
 - A referential constraint is satisfied if one of the following conditions is true, depending on the <match type> specified in the <referential constraint definition>:
 - If no <match type> was specified then, for each row *R1* of the *referencing table*, either at least one of the values of the *referencing columns* in *R1* shall be a null value, or the value of each referencing column in *R1* shall be equal to the value of the corresponding *referenced column* in some row of the *referenced table*.
 - If MATCH FULL was specified then, for each row *R1* of the *referencing table*, either the value of every *referencing column* in *R1* shall be a null value, or the value of every *referencing column* in *R1* shall not be null and there shall be some row *R2* of the *referenced table* such that the value of each *referencing column* in *R1* is equal to the value of the corresponding *referenced column* in *R2*.
 - If MATCH PARTIAL was specified then, for each row *R1* of the *referencing table*, there shall be some row *R2* of the *referenced table* such that the value of each *referencing column* in *R1* is either null or is equal to the value of the corresponding *referenced column* in *R2*.
 - NOTE 30 — If MATCH FULL or MATCH PARTIAL is specified for a referential constraint and if the referencing table has only one column specified in <referential constraint definition> for that referential constraint, or if the referencing table has more than one specified column for that <referential constraint definition>, but none of those columns is nullable, then the effect is the same as if no <match type> were specified.

RÉFÉRENCES

- Elmasri et Navathe (4^e ed.), chapitre 7
- Elmasri et Navathe (6^e ed.), chapitre 4
- [Date2012]
Date, Chris J. ;
SQL and Relational Theory: How to Write Accurate SQL Code.
2nd edition, O'Reilly, 2012.
ISBN 978-1-449-31640-2.
- Le site d'Oracle (en anglais)
 - http://docs.oracle.com/cd/E11882_01/index.htm
- Le site de PostgreSQL (en français)
 - <http://docs.postgresqlfr.org>

