BASES DE DONNÉES

SQL – LMD
Introduction à la gestion des transactions

BD130
v110a

2016-11-15

Christina KHNAISSER et Luc LAVOIE
Département d'informatique
Faculté des sciences

UNIVERSITÉ DE SHERBROOKE

Christina.Khnaisser@usherbrooke.ca
Luc.Lavoie@usherbrooke.ca
http://info.usherbrooke.ca/llavoie

Plan

- Motivation
- Syntaxe
- Exemples
- Synthèse

Motivation

- Regrouper des instructions en imposant
  - soit leur exécution complète sans erreur
  - soit leur abandon
- Lever temporairement l'application de certaines contraintes

La forme typique la plus simple

START TRANSACTION

SET CONSTRAINTS { ALL | nom [, ...] } DEFERRED

...

... liste de commandes SQL ...

...

SET CONSTRAINTS { ALL | nom [, ...] } IMMEDIATE

COMMIT

NIVEAUX DE TRANSACTION

Niveau	Possibilité de lecture		
	transitoire	non répétable	fantôme
READ UNCOMMITTED	oui	oui	oui
READ COMMITTED	non	oui	oui
REPEATABLE READ	non	non	oui
SERIALIZABLE	non	non	non

Exemple

Elmasri, pp. 637-636

```
1.  EXEC SQL
2.    WHENEVER SQLERROR GO TO UNDO;
3.  EXEC SQL
4.    SET TRANSACTION
5.          READ WRITE
6.          DIAGNOSTICS SIZE 5
7.          ISOLATION LEVEL SERIALIZABLE;
8.  EXEC SQL
9.    INSERT
10.         INTO EMPLOYEE (FNAME, LNAME, SSN, DNO, SALARY)
11.         VALUES ('Robert','Smith','991004321',2,35000);
12. EXEC SQL
13.   UPDATE EMPLOYEE
14.         SET SALARY = SALARY * 1.1
15.         WHERE DNO = 2;
16. EXEC SQL
17.   COMMIT;
18.   GOTO  THE_END;
19.   UNDO: EXEC SQL ROLLBACK;
20.   THE_END:  ...
```

16.1 <START TRANSACTION STATEMENT>

Rôle

Start an SQL-transaction and set its characteristics.

Syntaxe

<start transaction statement> ::=

START <transaction characteristics>

<TRANSACTION CHARACTERISTICS>

<transaction characteristics> ::=
    TRANSACTION <transaction mode> [{ <comma> <transaction mode> }...]
<transaction mode> ::=
      <isolation level>
    | <transaction access mode>
    | <diagnostics size>
<transaction access mode> ::=
      READ ONLY
    | READ WRITE
<isolation level> ::=
    ISOLATION LEVEL <level of isolation>
<level of isolation> ::=
      READ UNCOMMITTED
    | READ COMMITTED
    | REPEATABLE READ
    | SERIALIZABLE
<diagnostics size> ::=
    DIAGNOSTICS SIZE <number of conditions>
<number of conditions> ::=
    <simple value specification>

16.2 <SET TRANSACTION STATEMENT>

Rôle

Set the characteristics of the next SQL-transaction for the SQL-agent.

Note 412

This statement has no effect on any SQL-transactions subsequent to the next SQL-transaction.

Syntaxe

<set transaction statement> ::=

SET [LOCAL] <transaction characteristics>

16.3 <SET CONSTRAINTS MODE STATEMENT>

Rôle

If an SQL-transaction is currently active, then set the constraint mode for that SQL-transaction in the current SQL-session. If no SQL-transaction is currently active, then set the constraint mode for the next SQL-transaction in the current SQL-session for the SQL-agent.

Note 414

This statement has no effect on any SQL-transactions subsequent to this SQL-transaction.

Syntaxe

<set constraints mode statement> ::=

SET CONSTRAINTS <constraint name list>
{ DEFERRED | IMMEDIATE }

<constraint name list> ::=

ALL

| <constraint name> [{ <comma> <constraint name> }...]

16.4 <SAVEPOINT STATEMENT>

Rôle

Establish a savepoint.

Syntaxe

<savepoint statement> ::= SAVEPOINT <savepoint specifier>

<savepoint specifier> ::= <savepoint name>

16.5 <RELEASE SAVEPOINT STATEMENT>

Rôle

Destroy a savepoint.

Syntaxe

<release savepoint statement> ::=

RELEASE SAVEPOINT <savepoint specifier>

16.6 <COMMIT STATEMENT>

Rôle

Terminate the current SQL-transaction with commit.

Syntaxe

<commit statement> ::=
COMMIT [WORK] [AND [NO] CHAIN]

16.7 <ROLLBACK STATEMENT>

Rôle

Terminate the current SQL-transaction with rollback, or rollback all actions affecting SQL-data and/or schemas since the establishment of a savepoint.

Syntaxe

<rollback statement> ::=
ROLLBACK [WORK]
[AND [NO] CHAIN] [<savepoint clause>]

<savepoint clause> ::=
TO SAVEPOINT <savepoint specifier>

SET CONSTRAINTS
{ ALL | nom [, ...] } { DEFERRED | IMMEDIATE }

Portée

les contraintes

par défaut : DEFERRABLE INITIALLY IMMEDIATE

exclusion PostgreSQL : NULL, NOT NULL, CHECK

les triggers

avec la mention DEFERRABLE

voir aussi

BEGIN

END

ABORT

WHENEVER

START TRANSACTION
[mode_transaction [, ...]]

où mode_transaction peut être :

ISOLATION LEVEL
{ SERIALIZABLE | REPEATABLE READ | READ COMMITTED | READ UNCOMMITTED }

READ WRITE | READ ONLY

[NOT] DEFERRABLE

http://docs.postgresql.fr/9.6/sql-set-constraints.html

SET CONSTRAINTS

http://docs.postgresql.fr/9.6/sql-set-transaction.html

SET TRANSACTION

http://docs.postgresql.fr/9.6/sql-commit.html

COMMIT

http://docs.postgresql.fr/9.6/sql-rollback.html

ROLLBACK

20.2 <EMBEDDED EXCEPTION DECLARATION>

Rôle

Specify the action to be taken when an SQL-statement causes a specific class of condition to be raised.

Syntaxe

<embedded exception declaration> ::=
WHENEVER <SQL condition> <condition action>
<SQL condition> ::=
<major category>
| SQLSTATE (<SQLSTATE class value> [, <SQLSTATE subclass value>])
| CONSTRAINT <constraint name>
<major category> ::=
SQLEXCEPTION | SQLWARNING | NOT FOUND
<SQLSTATE class value> ::=
!! See the Syntax Rules.
<SQLSTATE subclass value> ::=
!! See the Syntax Rules.
<condition action> ::=
CONTINUE | <go to>
<go to> ::=
{ GOTO | GO TO } <goto target>
<goto target> ::=
<host label identifier>
| <unsigned integer>

Références

- Elmasri et Navathe (6e ed.), section 21.6
- Ullman section 6.6
- [Loney2008]
  Loney, Kevin ;
  Oracle Database 11g: The Complete Reference.
  Oracle Press/McGraw-Hill/Osborne, 2008.
  ISBN 978-0071598750.
- [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://www.oracle.com/pls/db10g/portal.portal_demo3?selected=5
  - http://docs.oracle.com/cd/B19306_01/server.102/b14200/toc.htm
- Le site de PostgreSQL (en français)
  - http://docs.postgresqlfr.org