

SQL : Langage de Contrôle de Transactions

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille
Chercheur en programmation par contrainte (IA)
Ingénieur en génie logiciel

`elmouelhi.achref@gmail.com`



Transaction

- Ensemble de requêtes **SQL** (insertion, suppression ou modification)
- Exécutée en bloc (soit tout, soit rien)

© Achref EL M

Transaction

- Ensemble de requêtes **SQL** (insertion, suppression ou modification)
- Exécutée en bloc (soit tout, soit rien)

Exemple : un virement bancaire

- Un retrait d'un premier compte
- Un versement vers un deuxième

Avec MySQL

- À la fin de chaque transaction, on peut soit valider, soit annuler, même en cas de réussite de toutes les requêtes.
- Si une requête échoue, la transaction sera annulée et les changements ne seront pas validés.
- Par défaut, chaque requête est considérée comme une transaction : validé si la requête n'échoue pas.

© Achref EL

Avec MySQL

- À la fin de chaque transaction, on peut soit valider, soit annuler, même en cas de réussite de toutes les requêtes.
- Si une requête échoue, la transaction sera annulée et les changements ne seront pas validés.
- Par défaut, chaque requête est considérée comme une transaction : validé si la requête n'échoue pas.

Vérifions cela

```
SELECT @@autocommit;
```

MySQL

Avec MySQL

- À la fin de chaque transaction, on peut soit valider, soit annuler, même en cas de réussite de toutes les requêtes.
- Si une requête échoue, la transaction sera annulée et les changements ne seront pas validés.
- Par défaut, chaque requête est considérée comme une transaction : validé si la requête n'échoue pas.

Vérifions cela

```
SELECT @@autocommit;
```

Ou

```
SHOW variables WHERE variable_name='autocommit';
```

Autocommit

Par défaut, activé.

© Achref EL MOUELHI

Autocommit

Par défaut, activé.

Désactivons-le

```
SET autocommit=0;
```

Autocommit

Par défaut, activé.

Désactivons-le

```
SET autocommit=0;
```

Vérifions la modification

```
SELECT @@autocommit;
```

Testons cela

```
INSERT INTO personne SET nom='wick', prenom = 'john', num = 5;
```

© Achref EL MOUELHI ©

Testons cela

```
INSERT INTO personne SET nom='wick', prenom = 'john', num = 5;
```

Vérifions l'insertion

```
SELECT * FROM personne;
```

© Achref EL MOUELHI

Testons cela

```
INSERT INTO personne SET nom='wick', prenom = 'john', num = 5;
```

Vérifions l'insertion

```
SELECT * FROM personne;
```

Annulons la transaction

```
ROLLBACK;
```

Testons cela

```
INSERT INTO personne SET nom='wick', prenom = 'john', num = 5;
```

Vérifions l'insertion

```
SELECT * FROM personne;
```

Annulons la transaction

```
ROLLBACK;
```

Vérifions l'insertion

```
SELECT * FROM personne;
```

Testons cela

```
INSERT INTO personne SET nom='wick', prenom = 'john', num = 5;
```

Vérifions l'insertion

```
SELECT * FROM personne;
```

Annulons la transaction

```
ROLLBACK;
```

Vérifions l'insertion

```
SELECT * FROM personne;
```

le tuple n'existe plus.

Comment valider alors ?

```
INSERT INTO  personne set nom='wick', prenom = 'john', num = 5;
```

© Achref EL MOUELHI ©

Comment valider alors ?

```
INSERT INTO  personne set nom='wick', prenom = 'john', num = 5;
```

Vérifions l'insertion

```
SELECT * FROM  personne;
```

© Achref EL MOUELHI

Comment valider alors ?

```
INSERT INTO  personne set nom='wick', prenom = 'john', num = 5;
```

Vérifions l'insertion

```
SELECT * FROM  personne;
```

Annulons la transaction

```
COMMIT;
```

Comment valider alors ?

```
INSERT INTO  personne set nom='wick', prenom = 'john', num = 5;
```

Vérifions l'insertion

```
SELECT * FROM  personne;
```

Annulons la transaction

```
COMMIT;
```

Vérifions l'insertion

```
SELECT * FROM  personne;
```

Comment valider alors ?

```
INSERT INTO personne set nom='wick', prenom = 'john', num = 5;
```

Vérifions l'insertion

```
SELECT * FROM personne;
```

Annulons la transaction

```
COMMIT;
```

Vérifions l'insertion

```
SELECT * FROM personne;
```

le tuple a bien été ajouté.

Comment déclarer le début d'une transaction si l'`autocommit=1` ?

© Achref EL MOUELHI ©

Comment déclarer le début d'une transaction si l'autocommit=1 ?

Réactivons l'autocommit

```
SET autocommit=1;
```

Comment déclarer le début d'une transaction si l'autocommit=1 ?

Réactivons l'autocommit

```
SET autocommit=1;
```

Déclarons le début d'une transaction

```
START TRANSACTION;
```

Insérons un tuple

```
INSERT INTO personne SET nom='wick', prenom = 'john', num = 5;
```

© Achref EL MOUELHI ©

Insérons un tuple

```
INSERT INTO personne SET nom='wick', prenom = 'john', num = 5;
```

Annulons la transaction

```
ROLLBACK;
```

Insérons un tuple

```
INSERT INTO  personne SET nom='wick', prenom = 'john', num = 5;
```

Annulons la transaction

```
ROLLBACK;
```

Vérifions l'annulation

```
SELECT * FROM  personne;
```

Recommençons la transaction précédente et finissons la par un commit.

© Achref EL MOUELHI

Recommençons la transaction précédente et finissons la par un commit.

`commit` **et** `rollback` **signalent la fin de la transaction.**