

# **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](mailto:elmouelhi.achref@gmail.com)



## Une transaction

- Un ensemble de requêtes SQL
- Exécutée en bloc (soit tout, soit rien)

# Les transactions

## Une transaction

- Un ensemble de requêtes SQL
- Exécutée en bloc (soit tout, soit rien)

## Exemple : un virement bancaire

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

# Les transactions

## 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.

# Les transactions

## 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.

# Les transactions

## Autocommit

- Par défaut, activé.

## Désactivons-le

```
SET autocommit=0;
```

# Les transactions

## Autocommit

- Par défaut, activé.

## Désactivons-le

```
SET autocommit=0;
```

## Vérifions la modification

```
SELECT @@autocommit;
```

# Les transactions

Testons cela

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

# Les transactions

Testons cela

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

Vérifions l'insertion

```
SELECT * FROM personne;
```

# Les transactions

Testons cela

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

Vérifions l'insertion

```
SELECT * FROM personne;
```

Annulons la transaction

```
ROLLBACK;
```

# Les transactions

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;
```

# Les transactions

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.

# Les transactions

Comment valider alors ?

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

# Les transactions

Comment valider alors ?

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

Vérifions l'insertion

```
SELECT * FROM personne;
```

# Les transactions

Comment valider alors ?

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

Vérifions l'insertion

```
SELECT * FROM personne;
```

Annulons la transaction

```
COMMIT;
```

# Les transactions

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;
```

# Les transactions

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.

# Les transactions

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

# Les transactions

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

```
SET autocommit=1;
```

# Les transactions

**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;**

# Les transactions

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;
```

# Les transactions

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;
```

Annulons la transaction

```
ROLLBACK;
```

# Les transactions

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;
```

Annulons la transaction

```
ROLLBACK;
```

Vérifions l'annulation

```
SELECT * FROM personne;
```

# Les transactions

**Ressaisissons la transaction précédente et finissons la par un commit**

**Ressaisissons la transaction précédente et finissons la par un commit**

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