



# Base de données

## Le langage de manipulation de données (LMD)

# Objectif

- Quelques éléments utiles du MySQL
- L'insertion de données (insert)
- La mise à jour (update)
- La suppression d'informations (delete)



# Quelques éléments utiles du MySQL

- MySQL :
  - Langage procédural associé à la base de données MySQL.
  - N'appartient pas à la norme.
  - Utilisable au sein des instructions SQL.
- Pourquoi l'aborder ici ?
  - Expressions

Ne pas confondre MySQL avec SQL (qui est une Norme)  
Diff entre MySQL, Postgres,....

Type	constantes	valeurs
Caractère	'AZERTY' 'Ecole d"Informatique'	
Numérique	10 -15.26	
Date	'2007-12-31 23:59:59.99'	31 déc. 2007 23:59:59:99

Caractère d'échappement '  
pour insérer une autre ' (dépend du SGBD)



# Quelques éléments utiles du MySQL

- MySQL :
  - Langage procédural associé à la base de données MySQL.
  - N'appartient pas à la norme.
  - Utilisable au sein des instructions SQL.
- Pourquoi l'aborder ici ?
  - Opérateurs

IN: permet de renvoyer true or false si on trouve la chaîne de caractère dans l'autre

LIKE 'masque' : recherche chaîne de caractères 'masque'

Type	
Arithmétiques	+, -, *, /, %, ()
Comparaisons	=, >, <, >=, <=, <>, != IN (exp1, exp2, ...) LIKE 'masque' BETWEEN min AND max EXISTS (sous-requête)
Logiques	OR, AND, NOT
Concaténation	+



# Quelques éléments utiles du MySQL

- MySQL :
  - Langage procédural associé à la base de données MySQL.
  - N'appartient pas à la norme.
  - Utilisable au sein des instructions SQL.
- Pourquoi l'aborder ici ?
  - Fonctions

Attention: a priori getdate() semble ne pas fonctionner sur phpMyadmin

LTRIM: Left trim: retire les espaces à gauche

Types	
Agrégation	COUNT(*), COUNT(expr), SUM(expr), AVG(expr), MIN(expr), MAX(expr), ...
Dates	GETDATE(), DATEPART(format, expr), DATEDIFF(format, expr1, expr2), DATEADD(format,n,expr), DAY(expr), MONTH(expr), YEAR(expr), ...
Chaines de caractères	LTRIM(expr), RTRIM(expr), LOWER(expr), UPPER(expr), SUBSTRING(expr,id,lg), LENGTH(expr), REPLACE(expr1, expr2, expr3), ...
Conversion	CAST(expr AS data_type), CONVERT(data_type, expr [,style]), ...
Diverses	RAND([exprn]), ROUND(exprn,n[,f])



# Quelques éléments utiles du MySQL

- MySQL :
  - Langage procédural associé à la base de données MySQL.
  - N'appartient pas à la norme.
  - Utilisable au sein des instructions SQL.
- Pourquoi l'aborder ici ?
  - Fonctions : comment tester ?

```
SELECT RAND() * 100;  
SELECT TOUPPER('Bonjour');  
SELECT LTRIM('test');
```

- Liste complète : <https://sql.sh>



# Quelques éléments utiles du MySQL

*A VOUS  
DE JOUER !*

Dans une BDD, pour afficher qqch, on effectue tjrs un select.

```
-- Afficher 33.33 arrondi au centieme
SELECT round(100.0/3, 2);

-- Ajouter 30 jours à la date du jour
SELECT DATE_ADD(CURRENT_DATE, INTERVAL 30 DAY);

-- 'MySQL - requêtes SQL et MySQL'
SELECT SUBSTRING('MySQL - requêtes SQL et MySQL',
    LOCATE ('requêtes', 'MySQL - requêtes SQL et MySQL'),
    LENGTH ('requêtes'));
```

- Testez les fonctions vous permettant :
  - D'afficher 33.33, arrondi au centième, suite au calcul  $100.0/3$  (fonctions ROUND et CONVERT)
  - D'ajouter 30 jours à la date du jour (fonctions DATEADD et GETDATE)
  - D'extraire la chaîne 'requêtes' de la chaîne 'MySQL – requêtes SQL et MySQL' (fonctions CHARINDEX et SUBSTRING)



# Insertion des données

- 2 possibilités :
  - Valoriser toutes les colonnes d'une ligne

```
INSERT [INTO] table_name VALUES (val, ...);
```

insérer une ligne dans une table;

val1, val2, dans l'ordre.

INTO facultatif mais on le met systématiquement

- Valoriser quelques colonnes d'une ligne

```
INSERT [INTO] table_name (col, ...) VALUES (val, ...);
```

Essayer cette syntaxe

+ intéressante , car effet de bords important dans l'insertion de ligne





# Modification des données

- 2 possibilités :
  - Modifier toutes les lignes d'une table

```
UPDATE table_name SET column_name = value [,column_name = value];
```

set: affecte une nouvelle val

- Restreindre les lignes à modifier

```
UPDATE table_name SET column_name = value [,column_name = value]  
[WHERE condition];
```

where id=5; par exemple

[ ] pour dire champ optionnel



# Suppression des données

- 2 possibilités :
  - Supprimer toutes les lignes d'une table

```
DELETE [FROM] table_name;
```

- Restreindre les lignes à supprimer

```
DELETE [FROM] table_name [WHERE condition];
```

Erreur s'il y a des liens qui font référence à cet élément



Tenir compte de l'intégrité référentielle



# Suppression des données

AVOUS  
DE JOUER !

- Supprimez les employés à la retraite (+ de 65 ans)

Pas la table employe => on ne fait pas



Constatez que les congés et congés mensuels associés ont été supprimés également.  
Comportement défini par l'option ON DELETE CASCADE sur les clé étrangères.



# Gestion des transactions

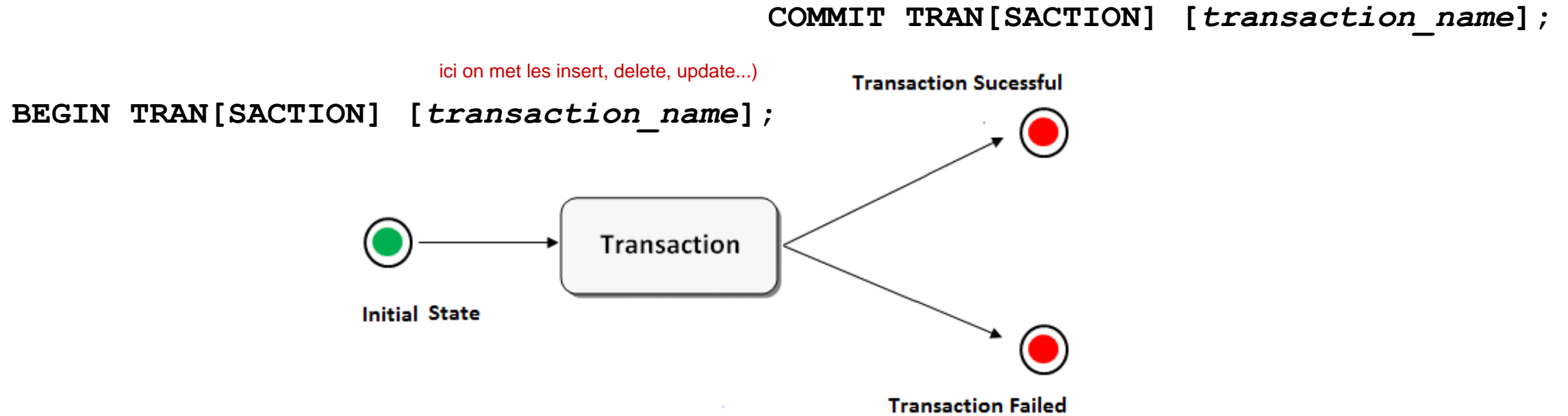
- Une transaction, c'est quoi ?
  - Ensemble indivisible d'instructions SQL (DML)
    - Soit toutes les instructions réussissent,
    - Soit aucune n'est validée.
- L'utiliser c'est :
  - Assurer l'intégrité et la cohérence des données,
  - Améliorer les performances,
  - Tester les effets de mises à jour massives,
  - Gérer les verrous sur les lignes. pas concerné.
- Principes ACID : rare
  - Atomicity on ne peut pas subdiviser un enregistrement (???)
  - Consistency lien et règle logique restent intactes
  - Isolation isoler la table un court instant pour une modif par exemple
  - Durability

Tout réussit ou tout plante:  
on ne veut pas ajouter une catégorie si les articles concernés ne sont pas MAJ

Verrou: lors d'une MAJ (ex sur Ameli), on verrouille la base pour qu'une seule modif soit effectuée et non 2 en mm temps



# Gestion des transactions



**ROLLBACK TRAN[SACTION]** Annule la transaction  
**[*transaction\_name* | *point\_name*];**

**SAVE TRAN[SACTION] [*point\_name*];** Fait des mini ROLLBACK



# Les verrous

- SQL Server pose automatiquement les verrous lors de l'exécution des transactions.
- Le type de verrou est choisi par SQL Server pour minimiser les coûts.
- Il est toutefois possible de définir le verrouillage des instructions SELECT pour toutes les transactions de la session.

**SET TRANSACTION ISOLATION LEVEL *option***

Option des verrous

Ex: user A maj, pdt qu'on update

- READ COMMITTED
- READ UNCOMMITTED    rare
- REPEATABLE READ
- SERIALIZABLE    pas touche a la table, tant qu'on l'update

Ex2: UserA et UserB mettent à jours 2 bases croisées, 1 verrou peuvent apparaître pour l'un et l'autre => timeout



# DML – La mise à jour des données

AVOUS  
DE JOUER !

- Insérez les données dans la base Gestions\_Employes en exécutant le script *Insert\_Data\_Gestion\_Employes.sql*

Zappe



Pensez à terminer la transaction démarrée en début de script





# Base de données

## Le langage de manipulation de données (LMD)