

Cours base de données

CHAPITRE 5 : SQL

Dr Coulibaly Tiekoura

PLAN DU CHAPITRE

1. Extraire des données à l'aide de l'instruction SQL SELECT
2. Restreindre et trier les données
3. Utiliser des fonctions monolignes afin de personnaliser la sortie
4. Afficher des données agrégées à l'aide des fonctions de groupe
5. Afficher des données de plusieurs tables
6. Utiliser des sous -interrogations
7. Utiliser des opérateurs ensemblistes

PLAN DU CHAPITRE (SUITE)

8. Manipuler des données
9. Utiliser des instructions LDD pour créer et gérer des tables

Cours base de données

CHAPITRE 5.8:
**MANIPULER DES
DONNÉES**

OBJECTIFS

- ◉ Décrire chaque instruction du langage de manipulation des données (LMD).
- ◉ Insérer des lignes dans une table.
- ◉ Mettre à jour des lignes dans une table.
- ◉ Supprimer des lignes d'une table.
- ◉ Gérer les transactions.

LANGAGE DE MANIPULATION DE DONNÉES

- ◉ Une instruction LMD??
- ◉ Une instruction LMD est exécutée pour :
 - Ajouter de nouvelles lignes dans une table.
 - Modifier des lignes existantes d'une table.
 - Supprimer des lignes existantes d'une table.
- ◉ Une transaction??
- ◉ Une transaction est constituée d'un ensemble d'instructions LMD.

AJOUTER UNE NOUVELLE LIGNE À UNE TABLE

DEPARTMENTS

70	Public Relations	100	1700
----	------------------	-----	------

Nouvelle
ligne

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

Insérer la nouvelle ligne
dans la table

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700
70	Public Relations	100	1700

SYNTAXE DE L'INSTRUCTION INSERT

- ⦿ Ajouter de nouvelles lignes à une table à l'aide de l'instruction INSERT.

```
INSERT INTO  table [(column [, column...])]  
VALUES      (value [, value...]);
```

- ⦿ Avec cette syntaxe, une seule ligne est insérée à la fois.

INSÉRER DE NOUVELLES LIGNES

- ◉ Insérez une nouvelle ligne contenant des valeurs pour chaque colonne.
- ◉ Répertoriez les valeurs dans l'ordre par défaut des colonnes de la table.
- ◉ Vous pouvez également indiquer les colonnes de la clause INSERT.

```
INSERT INTO departments(department_id,  
                        department_name, manager_id, location_id)  
VALUES (70, 'Public Relations', 100, 1700);  
1 row created.
```

- ◉ Incluez les valeurs de type caractère et de type date entre apostrophes.

INSÉRER DES LIGNES AVEC DES VALEURS NULL

- Méthode implicite : omettez la colonne dans la liste.

```
INSERT INTO departments (department_id,  
                          department_name   )  
VALUES (30, 'Purchasing');  
1 row created.
```

- Méthode explicite : indiquez le mot-clé NULL dans la clause VALUES.

```
INSERT INTO departments  
VALUES (100, 'Finance', NULL, NULL);  
1 row created.
```

INSÉRER DES VALEURS SPÉCIALES

- La fonction qui enregistre la date et l'heure??
- La fonction SYSDATE

```
INSERT INTO employees (employee_id,  
                        first_name, last_name,  
                        email, phone_number,  
                        hire_date, job_id, salary,  
                        commission_pct, manager_id,  
                        department_id)  
VALUES  
      (113,  
       'Louis', 'Popp',  
       'LPOPP', '515.124.4567',  
       SYSDATE, 'AC_ACCOUNT', 6900,  
       NULL, 205, 100);  
  
1 row created.
```

INSÉRER DES VALEURS DE DATE SPÉCIFIQUES

- Ajouter un nouvel employé.

```
INSERT INTO employees
VALUES      (114,
             'Den', 'Raphealy',
             'DRAPHEAL', '515.127.4561',
             TO_DATE('FEB 3, 1999', 'MON DD, YYYY'),
             'AC_ACCOUNT', 11000, NULL, 100, 30);
1 row created.
```

- Vérifiez l'ajout.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_P
114	Den	Raphealy	DRAPHEAL	515.127.4561	03-FEB-99	AC_ACCOUNT	11000	

CRÉER UN SCRIPT

- Utilisez l'esperluette d'interprétation dans une instruction SQL pour inviter l'utilisateur à saisir des valeurs.
- & est une marque de réservation pour la valeur de la variable.

```
INSERT INTO departments
      (department_id, department_name, location_id)
VALUES (&department_id, '&department_name', &location);
```

Define Substitution Variables

"department_id"	<input type="text" value="40"/>	<input type="button" value="Cancel"/>	<input type="button" value="Continue"/>
"department_name"	<input type="text" value="Human Resources"/>	<input type="button" value="Cancel"/>	<input type="button" value="Continue"/>
"location"	<input type="text" value="2500"/>	<input type="button" value="Cancel"/>	<input type="button" value="Continue"/>

```
1 row created.
```

MODIFIER DES DONNÉES DANS UNE TABLE

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID	COMMISSION_F
100	Steven	King	SKING	17-JUN-87	AD_PRES	24000	90	
101	Neena	Kochhar	NKOCHHAR	21-SEP-89	AD_VP	17000	90	
102	Lex	De Haan	LDEHAAN	13-JAN-93	AD_VP	17000	90	
103	Alexander	Hunold	AHUNOLD	03-JAN-90	IT_PROG	9000	60	
104	Bruce	Ernst	BERNST	21-MAY-91	IT_PROG	6000	60	
107	Diana	Lorentz	DLORENTZ	07-FEB-99	IT_PROG	4200	60	
124	Kevin	Mourgos	KMOURGOS	16-NOV-99	ST_MAN	5800	50	

Mise à jour des lignes dans la table EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID	COMMISSIO
100	Steven	King	SKING	17-JUN-87	AD_PRES	24000	90	
101	Neena	Kochhar	NKOCHHAR	21-SEP-89	AD_VP	17000	90	
102	Lex	De Haan	LDEHAAN	13-JAN-93	AD_VP	17000	90	
103	Alexander	Hunold	AHUNOLD	03-JAN-90	IT_PROG	9000	30	
104	Bruce	Ernst	BERNST	21-MAY-91	IT_PROG	6000	30	
107	Diana	Lorentz	DLORENTZ	07-FEB-99	IT_PROG	4200	30	
124	Kevin	Mourgos	KMOURGOS	16-NOV-99	ST_MAN	5800	50	

SYNTAXE DE L'INSTRUCTION UPDATE

- Modifier des lignes existantes à l'aide de l'instruction UPDATE:

```
UPDATE      table  
SET         column = value [, column = value, ...]  
[WHERE      condition];
```

- Mettre à jour plusieurs lignes à la fois (si nécessaire).

METTRE À JOUR DES LIGNES DANS UNE TABLE

- Des lignes spécifiques sont modifiées si vous indiquez la clause WHERE :

```
UPDATE employees  
SET    department id = 70  
WHERE  employee id = 113;  
1 row updated.
```

- Toutes les lignes de la table sont modifiées si vous omettez la clause WHERE:

```
UPDATE    copy_emp  
SET       department_id = 110;  
22 rows updated.
```


METTRE À JOUR DEUX COLONNES AVEC UNE SOUS-INTERROGATION

- Mettez à jour le poste et le salaire de l'employé 114 afin qu'ils correspondent à ceux de l'employé 205.

```
UPDATE    employees
SET       job_id  = (SELECT  job_id
                     FROM    employees
                     WHERE    employee_id = 205),
          salary  = (SELECT  salary
                     FROM    employees
                     WHERE    employee_id = 205)
WHERE     employee_id = 114;
1 row updated.
```

METTRE À JOUR DES LIGNES EN FONCTION D'UNE AUTRE TABLE

- Utilisez des sous-interrogations dans les instructions UPDATE pour mettre à jour des lignes d'une table en fonction des valeurs d'une autre table.

```
UPDATE copy_emp
SET    department_id = (SELECT department_id
                        FROM employees
                        WHERE employee_id = 100)
WHERE  job_id         = (SELECT job_id
                        FROM employees
                        WHERE employee_id = 200);

1 row updated.
```

SUPPRIMER UNE LIGNE D'UNE TABLE

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing		
100	Finance		
50	Shipping	124	1500
60	IT	103	1400

Delete a row from the DEPARTMENTS table:

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing		
50	Shipping	124	1500
60	IT	103	1400

INSTRUCTION DELETE

- Vous pouvez supprimer des lignes existantes d'une table à l'aide de l'instruction DELETE:

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

SUPPRIMER DES LIGNES D'UNE TABLE

- Des lignes spécifiques sont supprimées si vous indiquez la clause WHERE:

```
DELETE FROM departments  
WHERE department_name = 'Finance';  
1 row deleted.
```

- Toutes les lignes de la table sont supprimées si vous omettez la clause WHERE:

```
DELETE FROM copy_emp;  
22 rows deleted.
```

SUPPRIMER DES LIGNES EN FONCTION D'UNE AUTRE TABLE

- Utilisez des sous-interrogation dans les instructions DELETE afin de supprimer des lignes d'une table en fonction des valeurs d'une autre table :

```
DELETE FROM employees
WHERE department_id =
    (SELECT department_id
     FROM departments
     WHERE department_name
       LIKE '%Public%');

1 row deleted.
```

TRANSACTIONS DE BASE DE DONNÉES

- ◉ Une transaction de base de données est constituée de l'un des éléments suivants:
 - Des instructions LMD.
 - Une instruction LDD.
 - Une instruction de langage de contrôle de données(LCD)

TRANSACTIONS DE BASE DE DONNÉES

- ◉ Commencent lorsque la première instruction SQL LMD est exécutée
- ◉ Se terminent lorsque l'un des événements suivants se produit:
 - Une instruction COMMIT ou ROLLBACK est exécutée.
 - Une instruction LDD ou LCD est exécutée (validation automatique)
- ◉ S'annulent lorsque l'un des événements suivants se produit:
 - L'utilisateur quitte la fenêtre SQL
 - Le système connaît une défaillance

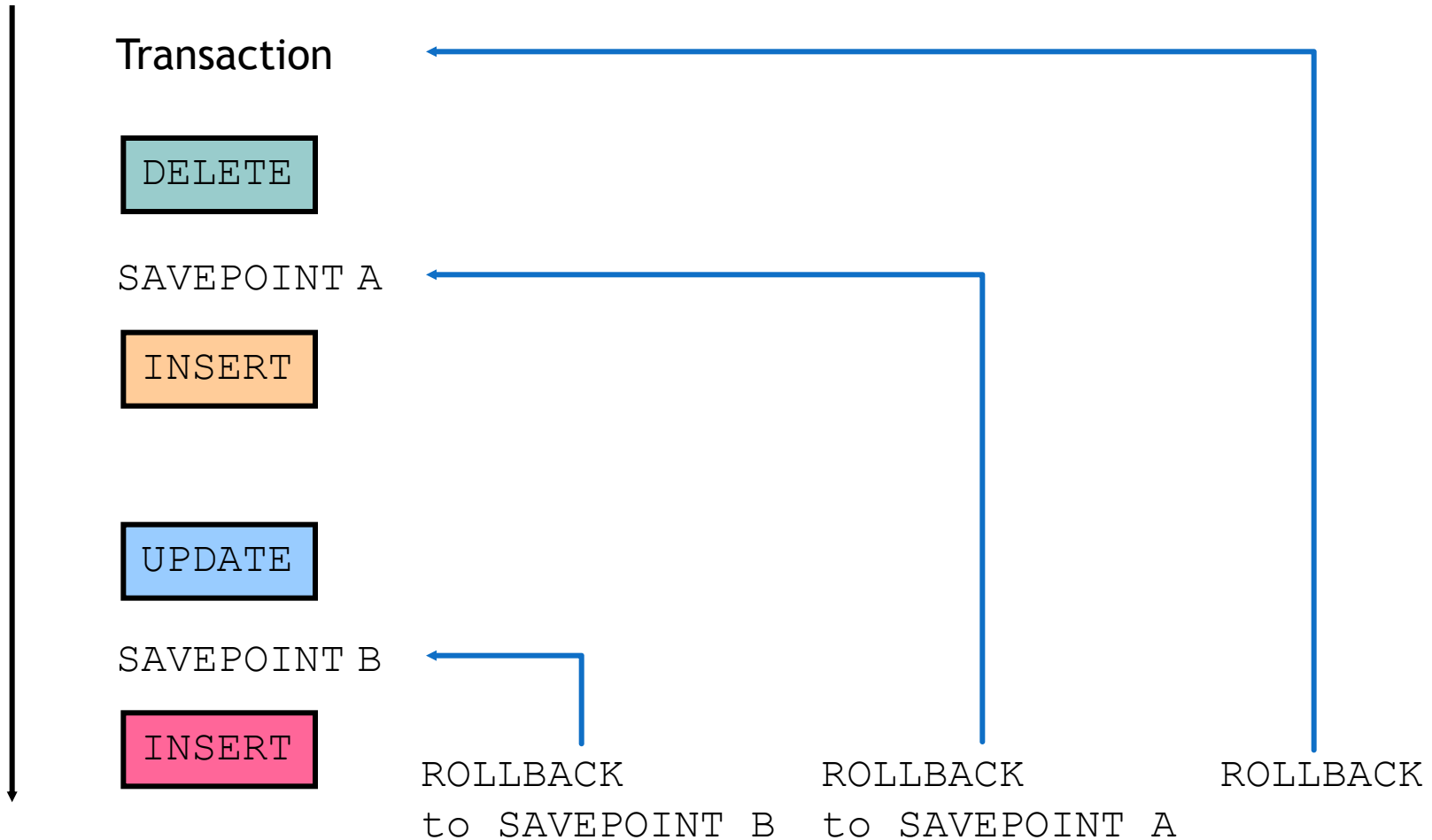
AVANTAGES DES INSTRUCTIONS COMMIT ET ROLLBACK

- ◉ Avec les instructions COMMIT et ROLLBACK, vous pouvez :
 - Garantir la cohérence des données.
 - Empêcher les modifications de données avant que les changements ne soient définitifs.
(verrous)
 - Regrouper des opérations liées logiquement.

GÉRER LES TRANSACTIONS

Time

COMMIT



ANNULER LES MODIFICATIONS APPORTÉES JUSQU'À UN MARQUEUR

- Créez un marqueur dans une transaction en cours à l'aide de l'instruction **SAVEPOINT**.
- Procédez à une annulation jusqu'à ce marqueur à l'aide de l'instruction **ROLLBACK TO SAVEPOINT**.

```
UPDATE...  
SAVEPOINT update_done;  
Savepoint created.  
INSERT...  
ROLLBACK TO update_done;  
Rollback complete.
```

ÉTAT DES DONNÉES AVANT L'OPÉRATION COMMIT OU ROLLBACK

- ◉ L'état antérieur des données peut être récupéré.
- ◉ L'utilisateur actuel peut examiner les résultats des opérations LMD à l'aide de l'instruction SELECT.
- ◉ Les autres utilisateurs ne peuvent pas afficher les résultats des instructions LMD exécutées par l'utilisateur actuel.
- ◉ Les lignes affectées sont verrouillées ; les autres utilisateurs ne peuvent donc pas modifier les données affectées.

ÉTAT DES DONNÉES APRÈS L'OPÉRATION COMMIT

- ⦿ Les modifications apportées aux données de la base sont rendues définitives.
- ⦿ L'état antérieur des données est définitivement perdu.
- ⦿ Tous les utilisateurs peuvent voir les résultats.
- ⦿ Les verrous externes sur les lignes affectées sont libérées ; ces lignes peuvent alors être manipulées par d'autres utilisateurs.
- ⦿ Tous les savepoints sont supprimés.

VALIDER DES INSTRUCTIONS LMD

● Apporter les modifications

```
DELETE FROM employees  
WHERE employee_id = 99999;  
1 row deleted.
```

```
INSERT INTO departments  
VALUES (290, 'Corporate Tax', NULL, 1700);  
1 row created.
```

```
COMMIT;  
Commit complete.
```

ÉTAT DES DONNÉES APRÈS L'OPÉRATION ROLLBACK

- Annulez toutes les modifications en cours à l'aide de l'instruction ROLLBACK:
 - Les modifications apportées aux données sont annulées.
 - L'état antérieur des données est restauré.
 - Les verrous externes sur les lignes affectées sont libérés.

```
DELETE FROM copy_emp;  
22 rows deleted.  
ROLLBACK ;  
Rollback complete.
```

ANNULER DES INSTRUCTIONS LMD

```
DELETE FROM test;  
25,000 rows deleted.
```

```
SELECT * FROM test;  
No rows selected.
```

```
ROLLBACK;  
Rollback complete.
```

```
SELECT * FROM test;  
25,000 rows selected.
```


SYNTHÈSE

- Ce chapitre vous a permis d'apprendre à utiliser les instructions suivantes:

Function	Description
INSERT	Ajoute une nouvelle ligne à la table
UPDATE	Modifie les lignes existantes dans la table
DELETE	Supprime des lignes existantes de la table
COMMIT	Rend définitive toutes les modifications en cours
SAVEPOINT	Est utilisé pour annuler une transaction jusqu'au marqueur de savepoint
ROLLBACK	Annule toutes les modifications de données en cours