Manipuler des données

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Objectifs

A la fin de ce chapitre, vous pourrez :

- décrire chaque instruction LMD
- insérer des lignes dans une table
- modifier des lignes dans une table
- supprimer des lignes d'une table
- fusionner des lignes dans une table
- contrôler les transactions

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Langage de manipulation de données

- Une instruction LMD est exécutée lorsque vous :
  - ajoutez des lignes à une table,
  - modifiez des lignes existantes dans une table,
  - supprimez des lignes d'une table.
- Une *transaction* est un ensemble d'instructions LMD formant une unité de travail logique.

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Ajouter une nouvelle ligne dans une table

70	Public Relations	100	1700

Nouvelle ligne

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

... insérer une nouvelle ligne dans la table DEPARMENTS ...

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

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Syntaxe de l'instruction INSERT

- L'instruction INSERT permet d'ajouter de nouvelles lignes dans une table.

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

- Cette syntaxe n'insère qu'une seule ligne à la fois.

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Insérer de nouvelles lignes

- Insérez une nouvelle ligne en précisant une valeur pour chaque colonne.
- Indiquez les valeurs dans l'ordre par défaut des colonnes dans la table.
- Indiquez éventuellement les colonnes dans la clause `INSERT`.

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

- Placez les valeurs de type caractère et date entre apostrophes.

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Insérer des lignes contenant des valeurs NULL

- Méthode implicite : n'indiquez pas 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.
```


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Insérer des valeurs spéciales

La fonction SYSDATE enregistre la date et l'heure en cours.

```
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.
```


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Insérer des dates dans un format spécifique

- Ajoutez 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.

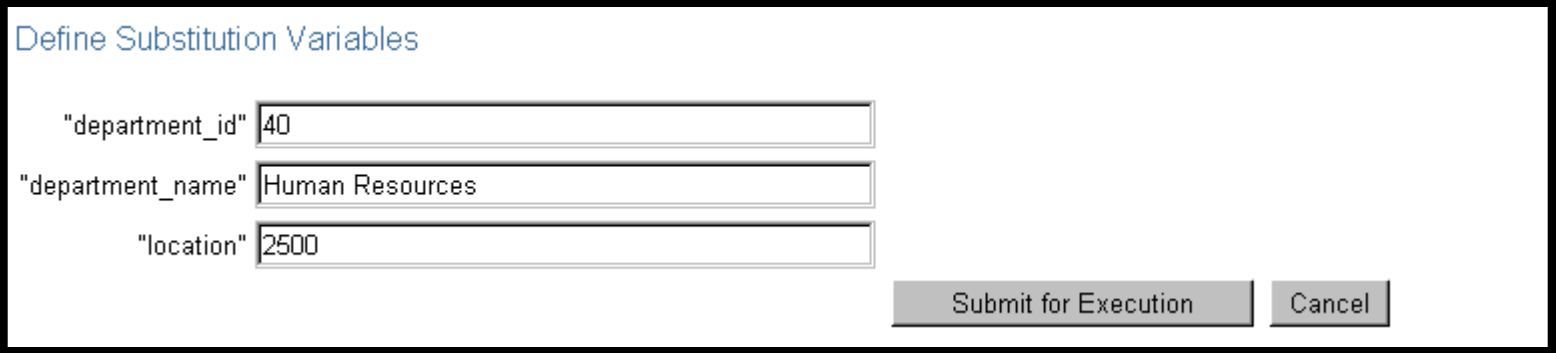
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_P
114	Den	Raphealy	DRAPHEAL	515.127.4561	03-FEB-99	AC_ACCOUNT	11000	

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Créer un script

- Utilisez le caractère de substitution & dans une instruction SQL de saisie de valeurs.
- & est une marque de réservation pour les variables.

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




```
1 row created.
```


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Copier des lignes d'une autre table

- Ecrivez votre instruction `INSERT` en précisant une sous-interrogation.

```
INSERT INTO sales_reps(id, name, salary, commission_pct)
  SELECT employee_id, last_name, salary, commission_pct
  FROM   employees
  WHERE  job_id LIKE '%REP%';

4 rows created.
```

- N'utilisez pas la clause `VALUES`.
- Le nombre de colonnes de la clause `INSERT` doit correspondre à celui de la sous-interrogation.

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Modifier les données d'une table

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID	COMMISSION_P
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	

Mettez à jour les lignes de 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	

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Syntaxe de l'instruction UPDATE

- Utilisez l'instruction UPDATE pour modifier des lignes existantes.

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

- Si nécessaire, vous pouvez modifier plusieurs lignes à la fois.

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Modifier des lignes d'une table

- La clause `WHERE` permet de modifier une ou plusieurs lignes spécifiques.

```
UPDATE employees
SET    department_id = 70
WHERE  employee_id = 113;
1 row updated.
```

- En cas d'absence de la clause `WHERE`, toutes les lignes sont modifiées.

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


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Modifier deux colonnes à l'aide d'une sous-interrogation

Modifiez le poste et le salaire de l'employé 114 pour 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.
```


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Modifier des lignes en fonction d'une autre table

Utilisez des sous-interrogations dans l'instruction UPDATE pour modifier des lignes d'une table à l'aide de 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.
```


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Erreur de contrainte d'intégrité lors de la modification de lignes

```
UPDATE employees
SET    department_id = 55
WHERE  department_id = 110;
```

```
UPDATE employees
       *
ERROR at line 1:
ORA-02291: integrity constraint (HR.EMP_DEPT_FK)
violated - parent key not found
```

Le numéro de service 55 n'existe pas.

Copyright © Oracle Corporation, 2001. Tous droits réservés.

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

Supprimez une ligne de la table DEPARTMENTS.

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

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Instruction DELETE

Vous pouvez supprimer des lignes d'une table au moyen de l'instruction DELETE.

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


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Supprimer des lignes d'une table

- La clause `WHERE` permet de supprimer des lignes spécifiques.

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

- En cas d'absence de la clause `WHERE`, toutes les lignes sont supprimées.

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


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Supprimer des lignes associées à des valeurs d'une autre table

Utilisez des sous-interrogations dans l'instruction DELETE pour supprimer des lignes dont certaines valeurs correspondent à celles d'une autre table.

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

1 row deleted.
```


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Erreur de contrainte d'intégrité lors de la suppression de lignes

```
DELETE FROM departments
WHERE       department_id = 60;
```

```
DELETE FROM departments
            *
ERROR at line 1:
ORA-02292: integrity constraint (HR.EMP_DEPT_FK)
violated - child record found
```

Vous ne pouvez pas supprimer une ligne qui contient une clé primaire utilisée comme clé étrangère dans une autre table.

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Utiliser une sous-interrogation dans une instruction INSERT

```
INSERT INTO
        (SELECT employee_id, last_name,
                email, hire_date, job_id, salary,
                department_id
         FROM   employees
         WHERE  department_id = 50)
VALUES (99999, 'Taylor', 'DTAYLOR',
        TO_DATE('07-JUN-99', 'DD-MON-RR'),
        'ST_CLERK', 5000, 50);

1 row created.
```


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Utiliser une sous-interrogation dans une instruction INSERT

```
SELECT employee_id, last_name, email, hire_date,
       job_id, salary, department_id
FROM   employees
WHERE  department_id = 50;
```

EMPLOYEE_ID	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID
124	Mourgos	KMOURGOS	16-NOV-99	ST_MAN	5800	50
141	Rajs	TRAJS	17-OCT-95	ST_CLERK	3500	50
142	Davies	CDAVIES	29-JAN-97	ST_CLERK	3100	50
143	Matos	RMATOS	15-MAR-98	ST_CLERK	2600	50
144	Vargas	PVARGAS	09-JUL-98	ST_CLERK	2500	50
99999	Taylor	DTAYLOR	07-JUN-99	ST_CLERK	5000	50

6 rows selected.

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Utiliser le mot-clé WITH CHECK OPTION avec les instructions LMD

- Une sous-interrogation permet d'identifier la table et les colonnes des instructions LMD.
- Le mot-clé WITH CHECK OPTION vous empêche de modifier les lignes qui ne sont pas présentes dans la sous-interrogation.

```
INSERT INTO  (SELECT employee_id, last_name, email,
                     hire_date, job_id, salary
              FROM   employees
              WHERE  department_id = 50 WITH CHECK OPTION)
VALUES (99998, 'Smith', 'JSMITH',
        TO_DATE('07-JUN-99', 'DD-MON-RR'),
        'ST_CLERK', 5000);
INSERT INTO
         *
ERROR at line 1:
ORA-01402: view WITH CHECK OPTION where-clause violation
```


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Valeur par défaut explicite : présentation

- La fonction de définition de valeur par défaut explicite vous permet d'utiliser le mot-clé `DEFAULT` en tant que valeur de colonne lorsque vous avez besoin d'une valeur de colonne par défaut.
- L'intégration de cette fonction permet la conformité à la norme SQL: 1999.
- L'utilisateur peut ainsi contrôler quand et où la valeur par défaut doit être appliquée aux données.
- Les valeurs par défaut explicites peuvent être utilisées dans les instructions `INSERT` et `UPDATE`.

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Utiliser des valeurs explicites par défaut

- **DEFAULT avec INSERT :**

```
INSERT INTO departments
  (department_id, department_name, manager_id)
VALUES (300, 'Engineering', DEFAULT);
```

- **DEFAULT avec UPDATE :**

```
UPDATE departments
SET manager_id = DEFAULT WHERE department_id = 10;
```


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Instruction MERGE

- Permet de mettre à jour ou d'insérer des données dans une table, de façon conditionnelle.
- Exécute une instruction `UPDATE` si la ligne existe et une instruction `INSERT` s'il s'agit d'une nouvelle ligne :
  - Evite des mises à jour distinctes
  - Améliore les performances et facilite l'utilisation
  - S'avère particulièrement utile dans les applications de data warehouse

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Syntaxe de l'instruction MERGE

L'instruction MERGE vous permet de mettre à jour ou d'insérer des lignes dans une table de façon conditionnelle.

```
MERGE INTO table_name table_alias
  USING (table|view|sub_query) alias
  ON (join condition)
  WHEN MATCHED THEN
    UPDATE SET
    col1 = col_val1,
    col2 = col2_val
  WHEN NOT MATCHED THEN
    INSERT (column_list)
    VALUES (column_values);
```


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Fusionner des lignes

Insérez ou mettez à jour des lignes dans la table `COPY_EMP` pour qu'elle corresponde à la table `EMPLOYEES`.

```
MERGE INTO copy_emp  c
  USING employees e
  ON (c.employee_id = e.employee_id)
WHEN MATCHED THEN
  UPDATE SET
     c.first_name     = e.first_name,
     c.last_name      = e.last_name,
     ...
     c.department_id  = e.department_id
WHEN NOT MATCHED THEN
 INSERT VALUES(e.employee_id, e.first_name, e.last_name,
          e.email, e.phone_number, e.hire_date, e.job_id,
          e.salary, e.commission_pct, e.manager_id,
          e.department_id);
```


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Fusionner des lignes

```
SELECT *
FROM COPY_EMP;

no rows selected
```

```
MERGE INTO copy_emp c
  USING employees e
  ON (c.employee_id = e.employee_id)
WHEN MATCHED THEN
  UPDATE SET
     ...
WHEN NOT MATCHED THEN
 INSERT VALUES...;
```

```
SELECT *
FROM COPY_EMP;

20 rows selected.
```


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Transactions de la base de données

Une transaction de base de données est constituée de l'un des éléments suivants :

- des instructions LMD effectuant une modification cohérente des données,
- une instruction LDD,
- une instruction LCD.

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Transactions de la base de données

- Commence à l'exécution de la première instruction LMD SQL.
- Se termine par l'un des événements suivants :
  - Une instruction `COMMIT` ou `ROLLBACK` est lancée.
  - Une instruction LDD ou LCD (validation automatique) est exécutée.
  - L'utilisateur quitte *i*SQL*Plus.
  - Le système tombe en panne.

Copyright © Oracle Corporation, 2001. Tous droits réservés.

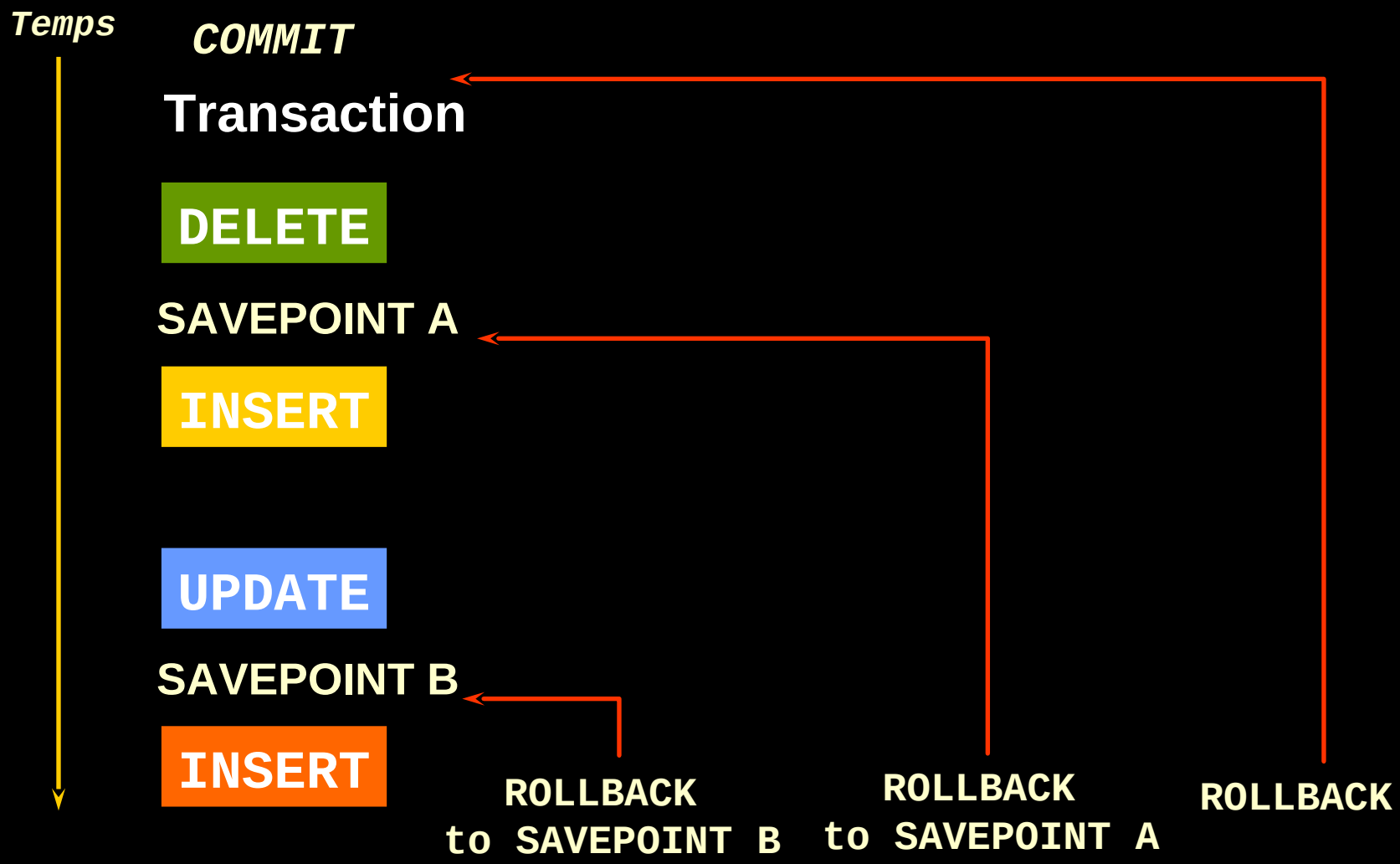
Avantages des instructions COMMIT et ROLLBACK

Les instructions COMMIT et ROLLBACK vous permettent :

- de garantir la cohérence des données,
- d'afficher le résultat des modifications de données avant qu'elles ne soient définitives,
- de regrouper de manière logique des opérations associées.

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Contrôler les transactions



Copyright © Oracle Corporation, 2001. Tous droits réservés.

Annuler des modifications jusqu'à une étiquette

- Créez une étiquette dans la transaction courante à l'aide de l'instruction `SAVEPOINT`.
- Annulez la transaction jusqu'à cette étiquette en utilisant l'instruction `ROLLBACK TO SAVEPOINT`.

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


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Traitement implicite des transactions

- Une validation automatique a lieu dans les situations suivantes :
  - exécution d'une instruction LDD,
  - exécution d'une instruction LCD,
  - sortie normale d'*i*SQL*Plus, sans instruction `COMMIT` ou `ROLLBACK` explicite.
- Il se produit une annulation automatique en cas de sortie anormale d'*i*SQL*Plus ou d'une panne du système.

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Etat des données avant COMMIT ou ROLLBACK

- Il est possible de restaurer l'état précédent des données.
- L'utilisateur en cours peut afficher le résultat des opérations LMD au moyen de l'instruction SELECT.
- Les résultats des instructions LMD exécutées par l'utilisateur courant *ne peuvent pas* être affichés par d'autres utilisateurs.
- Les lignes concernées sont *verrouillées*. Aucun autre utilisateur ne peut les modifier.

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Etat des données après COMMIT

- Les modifications des données dans la base sont définitives.
- L'état précédent des données est irrémédiablement perdu.
- Tous les utilisateurs peuvent voir le résultat des modifications.
- Les lignes verrouillées sont libérées et peuvent de nouveau être manipulées par d'autres utilisateurs.
- Tous les savepoints sont effacés.

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Valider des données

- Effectuez les modifications.

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

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

- Validez les modifications.

```
COMMIT;
Commit complete.
```


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Etat des données après ROLLBACK

L'instruction ROLLBACK permet de rejeter toutes les modifications de données en cours :

- Les modifications sont annulées.
- Les données retrouvent leur état précédent.
- Les lignes verrouillées sont libérées.

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


Copyright © Oracle Corporation, 2001. Tous droits réservés.

Annulation au niveau instruction

- Si une instruction LMD échoue pendant l'exécution, seule cette instruction est annulée.
- Le serveur Oracle met en oeuvre un savepoint implicite.
- Toutes les autres modifications sont conservées.
- L'utilisateur doit terminer explicitement les transactions en exécutant une instruction `COMMIT` ou `ROLLBACK`.

Copyright © Oracle Corporation, 2001. Tous droits réservés.

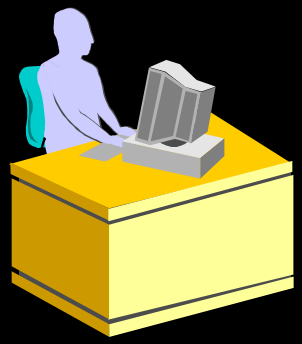
Cohérence en lecture

- La cohérence en lecture garantit à tout moment une vue homogène des données.
- Les modifications apportées par un utilisateur n'entrent pas en conflit avec celles d'un autre utilisateur.
- La cohérence en lecture garantit sur les mêmes données que :
  - la lecture ignore les écritures en cours,
  - l'écriture ne perturbe pas la lecture.

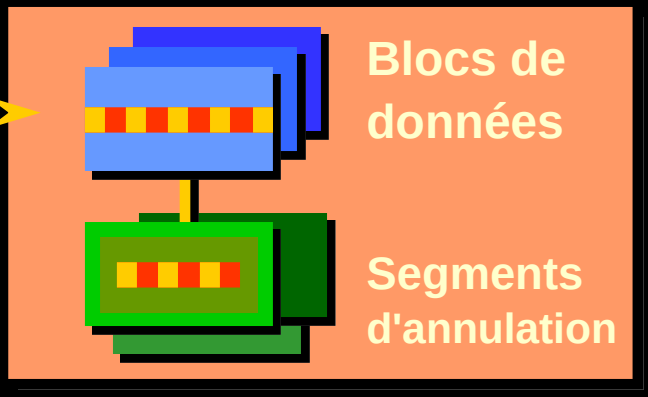
Copyright © Oracle Corporation, 2001. Tous droits réservés.

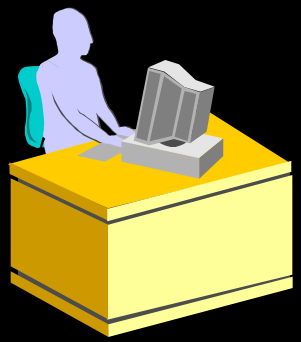
Implémenter la cohérence en lecture

Utilisateur A

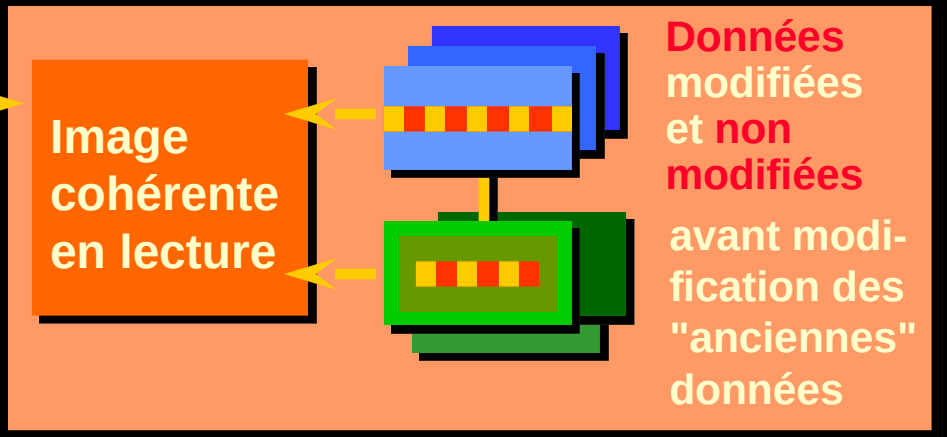


```
UPDATE employees
SET    salary = 7000
WHERE  last_name = 'Goyal';
```





```
SELECT  *
FROM userA.employees;
```



Utilisateur B

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Verrouillage

Les verrous d'une base de données Oracle :

- évitent les risques de destruction des données en cas de transactions simultanées,
- n'exigent aucune intervention de l'utilisateur,
- s'appliquent au niveau de restriction le plus bas,
- sont actifs durant toute la transaction,
- sont de deux types : explicite et implicite.

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Verrouillage implicite

- Deux modes de verrouillage :
  - Verrou de type Exclusive : Verrouille l'accès à tous les autres utilisateurs
  - Verrou de type Share : Permet aux autres utilisateurs d'accéder également aux données
- Haut niveau de simultanéité d'accès aux données
  - LMD : Partage des tables, row exclusive
  - Interrogations : Aucun verrou requis
  - LDD : Protège les définitions d'objet
- Verrouillage maintenu jusqu'à validation ou annulation

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Synthèse

Ce chapitre vous a permis d'apprendre à utiliser des instructions LMD et à contrôler les transactions.

Instruction	Description
`INSERT`	Ajoute une nouvelle ligne dans une table
`UPDATE`	Modifie des lignes dans une table
`DELETE`	Supprime des lignes d'une table
`MERGE`	Insère ou met à jour des données dans une table de façon conditionnelle
`COMMIT`	Valide toutes les modifications de données en cours jusqu'à l'étiquette du savepoint
`SAVEPOINT`	Permet une annulation jusqu'à un savepoint
`ROLLBACK`	Annule toutes les modifications de données en instance

Copyright © Oracle Corporation, 2001. Tous droits réservés.

Présentation de l'exercice 8

Dans cet exercice, vous allez :

- **insérer des lignes dans une table**
- **mettre à jour et supprimer des lignes dans une table**
- **contrôler des transactions**

Copyright © Oracle Corporation, 2001. Tous droits réservés.