

الفصل الثامن

Manipulating Data

الأهداف

- بعد إكمال هذا الدرس أنت يجب إن شاء الله أن تكون قادر علي عمل التالي:
- وصف كل تعبيرات DML (Data Manipulating Language) .
- إدخال صفوف جديدة للجدول.
- عمل تحديث وتغيير لبيانات الجدول.
- حذف صفوف معينة من الجدول .
- دمج صفوف من جدولين في جدول واحد.
- امكانية التعامل مع كلا من (Rollback, Commit and Savepoint).

تعبير DML ينفذ عند :

- اضافة صفوف جديدة للجدول.
- تعديل صفوف موجودة في الجدول.
- حذف صفوف موجودة في الجدول.

Adding a New Row to a Table

إضافة صف جديد للجدول

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	140	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

New row

70 Public Relations 100 1700

...insert a new row into the DEPARTMENTS table...

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	140	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700
70	Public Relations	100	1700

الصف الجديد المراد أضافته رقم قسمة 70 و أسم القسم هو علاقات عامة ورقم المدير هو 100 ورقم الموقع 1700
و يوضح لنا الرسم الفرق بين الجدول قبل إدخال الصف وبعد إدخال الصف. يمكنك إضافة صف جديد للجدول باستخدام امر (INSERT) والصيغة العامة لتلك الامر كما يلي:

**INSERT INTO table_Name [(column [, column...])]
VALUES (value1 [, value2,.....]);**

أمر INSERT يبدأ بعبارة INSERT INTO يليها اسم الجدول المراد الإضافة فيه مثلاً DEPT ثم بعد ذلك أسماء الأعمدة المراد إضافة البيانات فيها (وإذا لم تقم بكتابة الأعمدة فهذا يعني أنك سوف تدخل بيانات لكل الأعمدة ولا يمكنك تجاهل أي عمود بالجدول في هذه الحالة) ثم يتم كتابة Values ويتم كتابة القيم الجديدة بعدها في قوسين ولا بد أن يكون ترتيب تلك القيم الجديدة متوافق مع ترتيب الأعمدة السابق إدراجها.

ملاحظة:

إذا كنت تريد إضافة بيانات جديدة لكل الأعمدة في الجدول فلا داعي الى ذكر أسماء تلك الأعمدة ولكن يراعى ترتيب القيم الجديدة وفقاً لترتيب الأعمدة بالجدول حتى لا يحدث Error . وتجد في الناتج ظهور (1 ROW Created).
وعند استعراض بيانات الجدول يظهر لنا الصف الجديد الذي تم اضافته .

مثال:

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

لاحظ عند ادخال قيم بحروف او تاريخ لابد من وضع تلك القيم بين Single Quotations كما في المثال السابق فى اسم الادارة الجديد. وهذا يعنى ان اى قيمة تشتمل علي حرف أو تاريخ لابد من وضعها بين علامة ' ' .

ولاحظ ان البيانات الجديدة لابد ان تتوافق مع نوع بيانات العمود. فلا يمكن ادراج اسم الادارة السابق فى عمود رقم الادارة والسبب ان عمود رقم الادارة يسمح ببيانات أرقام فقط .

ولذلك لابد من معرفة نوع البيانات لكل عمود تريد اضافة بيانات جديدة به وذلك باستخدام الامر DESCRIBE او DESC كما فى المثال التالى:

```
DESCRIBE departments
```

Name	Null?	Type
DEPARTMENT_ID	NOT NULL	NUMBER(4)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
MANAGER_ID		NUMBER(6)
LOCATION_ID		NUMBER(4)

إضافة صفوف تحتوي علي قيم فارغة او NULL .

من الممكن إضافة صفوف تحتوي علي قيم فارغة ونقوم بعمل ذلك بطريقتين:-

- الطريقة الاولى : حذف العمود من قائمة الأعمدة.

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

الطريقة الأولى وهي إغفال ذكر اسم العمود المراد جعل قيمته فارغة (NULL). وتسمى هذه الطريقة بالطريقة الضمنية.

- الطريقة الثانية: ادراج العمود و تحديد كلمة NULL في فقرة VALUES

```
Insert into dept (deptno,dname,loc)
```

```
Values (50,'EDUCATION',NULL);
```

وهذه هي الطريقة الثانية وهى وضع كلمة NULL و تسمى بالطريقة الواضحة أو الصريحة .

الطرق لإدخال قيم فارغة

Implicit	حذف العمود من قائمة الأعمدة المراد ادخال بيانات بها
Explicit	تحديد NULL للعمود او تحديد مجموعة فارغة (' ') في قائمة VALUES

* أخطاء شائعة تحدث عند استخدام INSERT :-

- عدم ادخال قيمة فى عمود لابد من ادخال قيمة به مثل عمود (Primary Key).
- ادخال قيمة سبق ادخالها فى عمود (Primary Key) .
- ادخال قيم عدد حروفها اكبر من المتاح لتلك العمود.

```
Insert into EMP (ename, job)
```

```
Values ('ZAKI','MANAGER') ;
```

```
ERROR at line 1:
ORA-01400: cannot insert NULL into ("SCOTT"."EMP"."EMPNO")
```

وسبب هذا Error :

ان عمود EMPNO يعتبر Primary Key ولا يمكن ان يترك بدون قيمة.

استخدام SYSDATE و USER في جملة Insert

عند إضافة صفوف جديدة يمكن استخدام SYSDATE للتعبير عن التاريخ الحالي بدون كتابة يدويا. ويمكن استخدام كلمة User كقيمة لإضافة اسم المستخدم الحالي كما في المثال التالي:

Insert into login values (user, sysdate);

فلنفترض وجود جدول يحتوي على عمودين الاول لاسم المستخدم والآخر لتسجيل الوقت الحالي.

مثال:

Insert into employees

(employee_id, last_name, job_id, hire_date, commission_pct)

Values (113,user,'ac_account',sysdate,null);

```
SELECT employee_id, last_name, job_id, hire_date, commission_pct
FROM employees
WHERE employee_id = 113;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	HIRE_DATE	COMMISSION_PCT
113	Popp	AC_ACCOUNT	27-SEP-01	

هنا نجد أن **SYSDATE** أدت إلي إدخال تاريخ النظام وقت تنفيذ الأمر وأدت كلمة **user** إلي إدخال اسم المستخدم الحالي وهو **Popp**.
وأدى استخدام كلمة **NULL** الى ادراج **NULL** بداخل العمود.
إدخال قيم التاريخ

```
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.
```

- للتحقق من الإضافة

Select * from employees

Where employee_id=114;

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	

عند إدخال تاريخ يجب مراعاة كتابة بطريقة مبدئية وهي اليوم مكون من رقمين والشهر مكون من ثلاثة حروف ثم السنة مكونة من أربعة أرقام.
(DD-MON-YY) وعند إدخال التاريخ بطريقة مختلفة نستخدم الدالة **TO_DATE** وذلك لتحويل التاريخ إلي الطريقة التي يقبلها الأوراكل والنتائج يوضح شكل التاريخ بعد إدخالها كما هو موضح أعلي.
حيث يكون الناتج كالتالي 03-FEB-99

استخدام العلامة &

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

Define Substitution Variables

"department_id"	40
"department_name"	Human Resources
"location"	2500

Submit for Execution Cancel

1 row created.

في حالة الرغبة في إدخال أكثر من صف بقيم متغيرة يمكنك استخدام أمر **INSERT** بطريقة سهلة تجعلك من الممكن إدخال أكثر من قيمة وذلك عن طريق الرمز (&) للقيمة المراد إدخالها .
وعند تنفيذ الأمر يطلب من المستخدم إدخال القيمة فيقوم المستخدم بإدخالها كما في الرخصة السابقة.
وعند إدخال قيم تاريخية أو حرفية يجب وضعهم بين (' ') علامة التنصيص المفردة

ملحوظة :

داخل ال Item لا نضع علامات تنصيص ولا فصلة منقوطة (;) أثناء تنفيذ الامر .
كما يجب عدم إدخال قيم متكررة في عمود المفتاح الرئيسي.
وهذا يسمح لك أن تشغل نفس الامر مرارا وتكرارا ولكن بمجموعة مختلفة من القيم.

Coping a Rows from Another Table INSERT إمكانية استعمال SubQuery مع جملة

```
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.
```

ملاحظة:

* لا تستخدم فقرة VALUES مع SubQuery.
* لابد من مطابقة عدد الأعمدة في فقرة INSERT مع عدد الأعمدة في الاستعلام الفرعي SubQuery.
* بدلا من فقرة VALUES يتم استخدام الاستعلام الفرعي.
ويكون الشكل الرئيسي لها:
Table أسم الجدول.
Column أسماء الأعمدة في الجدول المراد اضافة بيانات به.
Subquery الاستعلام الفرعي الذي يسترجع صفوف من جدول معين.

- يجب تتطابق أنواع البيانات في الاستعلام الفرعي مع الأعمدة التي سوف يضاف بها البيانات.
- كي ننشئ نسخة من الصفوف في الجدول استخدم *SELECT في الاستعلام الفرعي.

Changing Data in a Table تغيير البيانات في الجدول

الموظفين

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID	COMMISSION_PCT
100	Steven	King	SKING	17-JUN-07	AD_PRES	24000	90	
101	Neena	Kochhar	NKOCHHAR	21-SEP-08	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	KMORGOS	16-NOV-99	ST_MAN	5800	50	

تعديل الصفوف في جدول الموظفين.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID	COMMISSION_PCT
100	Steven	King	SKING	17-JUN-07	AD_PRES	24000	90	
101	Neena	Kochhar	NKOCHHAR	21-SEP-08	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	30	
107	Diana	Lorentz	DLORENTZ	07-FEB-99	IT_PROG	4200	30	
124	Kevin	Mourgos	KMORGOS	16-NOV-99	ST_MAN	5800	50	

المثال المعروض أمامنا يقوم بتغيير رقم قسم الموظفين بقسم 60 إلي القسم رقم 30.

الصيغة الرئيسية لتعبير UPDATE تحديث أكثر من صف واحد في نفس الوقت ،إذا تطلب ذلك يمكن أن تعدل صفوف موجودة باستخدام تعبیر UPDATE.على النحو التالي:

Table أسم الجدول
Column أسم العمود المراد تغييره.
Value القيمة الجديدة.
Condition تحديد الصفوف التي ينطبق عليها الشرط المطلوب.
وباغفال الشرط يحدث تغيير بكل الأعمدة.

ملحوظة :

بشكل عام يمكن استخدام المفتاح الرئيسي (Primary Key) لتحديد صف وحيد. حيث استخدام الأعمدة الأخرى يمكن أن يؤدي الى نتائج غير متوقعة والسبب أن صفوف عديدة ينطبق عليها نفس الشرط.

كمثال: تحديد صف واحد من جدول الموظفين عن طريق استخدام عمود الاسماء كشرط لاجراء التغير المطلوب وهذا قد يؤدي الى حدوث خطأ لأنه من الممكن ان يكون أكثر من موظف بنفس الاسم. وفي تلك الحالة من الافضل استخدام عمود Primary Key لكي تتمكن من اجراء التغير بدقة حيث ان البيانات بتلك العمود تكون غير متكررة.

تعديل بيانات في الجدول

تحديد صف أو مجموعة صفوف في فقرة WHERE .

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

يمكن لصفوف الجدول كلها ان تعدل عن طريق حذف فقرة WHERE.

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

عندما نريد تعديل بيانات نستخدم الأمر UPDATE ويكتب علي النحو التالي:
الأمر UPDATE يليه أسم الجدول مثلا Employees المراد تعديل بياناته ثم كلمة SET يليها أسم العمود المراد تغيير قيمته ثم علامة يساوي ثم القيمة الجديدة. و يلي ذلك جملة الشرط (Condition) التي سوف تحدد الصفوف المراد تعديلها في الجدول.

ملاحظة: اذا لم تقم بتحديد الصفوف المراد تغييرها في جملة الشرط (Condition) فإن التعديل سيكون علي جميع صفوف الجدول.

كمثال نريد نقل الموظف رقم 7788 (SCOTT) للقسم رقم 30 .

Update EMP set deptno=30

Where empno=7788;

قبل اجراء التعديل كان رقم القسم 20 للموظف SCOTT .

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20

بعد اجراء التعديل اصبح رقم القسم 30 للموظف SCOTT .

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7788	SCOTT	ANALYST	7566	19-APR-87	3000		30

*إذا حذفت فقرة WHERE فإن كل صفوف الجدول سوف تعدل ويصبح الموظفين كلهم في الادارة رقم 30.

Update Two Column With SubQuery تعديل البيانات باستخدام Subquery

المثال التالي يريد تعديل الموظف رقم 114 بوظيفة ومرتب الموظف رقم 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.
```

هنا نجد إمكانية تعديل البيانات باستخدام أكثر من استعلام فرعي كما في المثال السابق.

ملحوظة :

إذ لم ينطبق الشرط بمعنى عدم وجود أي موظف بالرقم 114 تظهر لنا تلك الرسالة 0 Rows Updated بمعنى أن الشرط لم يتحقق وبالتالي لم يتم بآجراء أي تغيير بالجدول.

*تعديل صفوف معتمدا علي جدول آخر

```
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.
```

يمكن أن تستخدم الاستعلام الفرعي كي تعدل صفوف بجدول عن طريق استخدام بيانات من جدول آخر كما في المثال السابق .
المثال المعروض لتعديلات بجدول COPY_EMP معتمد علي قيم من جدول الموظفين.

Updating Rows: Integrity Constraint Error

```
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
```

عند محاولة تعديل المفتاح الأجنبي (Foreign Key) برقم غير موجود في عمود (Primary Key) المرتبط به بالجدول الآخر فينتج عن ذلك Error.
ففي **المثال** السابق حاول تغيير رقم القسم 110 بالرقم 55 مع العلم أن الرقم 55 غير موجود بعمود الأقسام (Primary Key) بجدول الأقسام ولهذا حدث Error .
وهذا يعني أن رقم الإدارة الجديد لابد أن يكون موجود في عمود (Primary Key) بالجدول الرئيسى أولا.

Removing a Row from Table حذف الصف من الجدول

جدول الأقسام

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

تم حذف صف من جدول الأقسام

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

لحذف صف أو مجموعة من الصفوف لجدول ما نستخدم الأمر Delete ويتم كتابته كما يلي:

1-Delete 2- From 3- Table ثم

الجدول الذي يتم الحذف منه

ثم جملة الشرط WHERE التي ستحدد الصف المراد حذفه. ويجب ملاحظة أنه في حالة إغفال جملة الشرط فإنه يقوم بحذف كل صفوف الجدول. والمثال السابق يحذف قسم المالية من جدول الأقسام.

ملحوظة :

إذا لم يكن هناك صف ينطبق عليه الشرط فتظهر تلك الرسالة 0 Rows Deleted وتعني عدم القيام بحذف أي صف.
حذف صفوف معتمدة علي جدول آخر ذلك عن طريق استخدام الاستعلام الفرعي (SubQuery) في فقرة Where .

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

1 row deleted.
```

يمكنك استخدام الاستعلام الفرعي كي تحذف صفوف من جدول معتمدا علي قيم من جدول آخر في جملة الشرط.
والمثال السابق يقوم بحذف كل الموظفين بالقسم الذي يحتوي اسمة علي مجموعة الحروف "Public" ، والاستعلام الفرعي يبحث في جدول الأقسام عن رقم القسم الذي يحتوي اسمة علي مجموعة الحروف "Public" .

Deleting Rows: Integrity Constraint Error

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

You cannot delete a row that contains a primary key that is used as a foreign key in another table.

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

*ملاحظة :

عند إجراء عملية حذف في جدول فلا يمكن حذف عمود يتم الاعتماد عليه في جدول آخر أي أن تلك العمود Primary Key و لعلاج ذلك لابد من حذف كل الصفوف الموجودة في الجدول الآخر بعمود (Foreign Key) أولا .
*يمكنك الاستغناء عن كلمة From كما في المثال التالي:

Delete employees
Where employees_id=7788;

استخدام الاستعلام الفرعي في تعبير 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.
```

يمكن أن تستخدم الاستعلام الفرعي (Subquery) بدلا من اسم الجدول في فقرة INTO في فقرة INSERT .
لاحظ أن قائمة الاعتماد في الاستعلام الفرعي (SubQuery) يجب أن يكون بنفس الترتيب للقيم في فقرة VALUES .

Using With Check Option on DML Statement

تستخدم فقرة With Check Option عندما تريد تحديد بعض الشروط عند إضافة صفوف جديدة .

```
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
```

فقرة WITH CHECK OPTION تمنعك من اضافة صفوف ليست في الاستعلام الفرعي.
فالمثال السابق كان سبب Error انة لم يقم باضافة رقم القسم المحدد فى جملة الشرط وهو (50) فى عمود رقم الاقسام بالنسبة للصف الجديد.
ولمعالجة ذلك نقم باضافة القسم 50 للموظف الجديد حيث انة تم تحديد فى جملة Where. اى اننا لايمكننا من اضافة موظفين الا فى القسم رقم 50.

```
Insert into (select employee_id, last_name, email,
hire_date,job_id,salary,
department_id from employees
where department_id=50 with check option)
values (9999, 'SMITH', 'JSMITH',
to_date ('12-oct-81','dd-mon-rr'), 'ST_CLERK' ,50000 ,50);
1 row created
```

اما اذا قمنا بتغيير رقم القسم بحيث يكون مختلف عن رقم القسم المحدد فى جملة WHERE ففى تلك الحالة لايسمح بذلك كما فى المثال التالى ولاحظ رقم القسم الجديد:

```
Insert into (select employee_id, last_name, email,
hire_date,job_id,salary,
department_id from employees
where department_id=50 with check option)
values (9999, 'SMITH', 'JSMITH',
to_date ('12-oct-81','dd-mon-rr'), 'ST_CLERK' , 50000 ,30);
```

```
Insert into (select employee_id, last_name, email,
*)
ERROR at line 1:
ORA-01402: view WITH CHECK OPTION where-clause violation
```

استخدام القيم المحددة سابقا (DEFAULT)

• DEFAULT with INSERT:

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

• DEFAULT with UPDATE:

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

Default هو قيمة افتراضية تم وضعها لعمود معين وذلك عند انشاء الجدول. بحيث اذا لم يتم ادخال بيانات محددة فى تلك العمود يمكنك استخدام Default المحدد مسبقا كما فى المثال السابق.
وهذا سوف يناقش إن شاء الله تعالى في الفصل التالى.

The MERGE Statement (الدمج) MERGE

- 1- إمكانية التعديل في بيانات بشكل شرطي أو إدخال بيانات في جداول قواعد البيانات.
- 2- زيادة الأداء والسهولة في الاستخدام و مفيد في تخزين البيانات.
- 3- أمر MERGE يجمع أوامر INSERT, UPDATE حيث أنك تحتاج كلاهما.
- 4- أمر MERGE يقوم بدمج جدولين معا ولكي يتم ذلك فلا بد من التأكد ان كلا من الجدولين يتضمن نفس عدد الاعمدة وبنفس نوع البيانات.

الصيغة الرئيسية لجملَة MERGE

```
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);
```

يمكنك تعديل الصفوف الموجودة و إدخال صفوف شرطية جديدة باستخدام تعبير
MERGE ويعنى ادمج.

فقرَة INTO تحديد الجدول المستهدف الذي يتم تعديله او إدخال
بيانات عليه.

فقرَة USING يعنى مصدر البيانات كي تعدل أو تدخل.

فقرَة ON الشرط الذي يعتمد عليه في MERGE .

WHEN MATCHED و يعنى ماذا يفعل عند تحقق الشرط .

WHEN NOT MATCHED و يعنى ماذا يفعل عند عدم تحقق الشرط .

مثال علي دمج صفوف كلا من جدول Employees وجدول Copy_emp

```
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.email = e.email,
    c.phone_number = e.phone_number,
    c.hire_date = e.hire_date,
    c.job_id = e.job_id,
    c.salary = e.salary,
    c.commission_pct = e.commission_pct,
    c.manager_id = e.manager_id,
    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);
```

الشرط الموجود في **المثال** السابق هو تساوي رقم الموظف في كلا من الجدولين
وعند تحقق تلك الشرط يقوم بتعديل البيانات في جدول Copy_emp بالبيانات الموجودة
في جدول Employees .

وعند عدم تحقق تلك الشرط فهذا يعنى ان الموظف غير موجود في جدول Copy_emp
ولهذا سوف نقوم باضافة تلك الموظف الى جدول Copy_emp .
والهدف من MERGE ان يكون بيانات الجدولين متماثلة.

(Database Transactions)

تحتوي علي واحد من التالي:-

- تعبير DML (Data Manipulating Language) وتحتوى على:

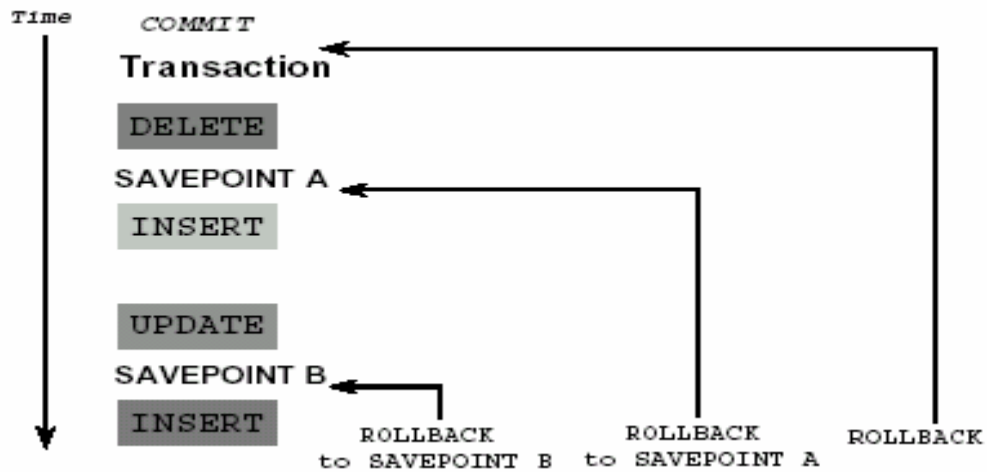
(INSERT, UPDATE, DELETE, MERGE)

- تعبير DDL (Data Definition Language) وتحتوى على:

(CREATE, MODIFY, DROP, RENAME, TRUNCATE)

- تعبير DCL (Data Control Language) و تحتوى على: (GRANT, REVOKE)

معاملات التحكم



يمكنك التحكم في المعاملات باستخدام أوامر COMMIT,SAVEPOINT,ROLLBACK.

الأوامر	الوصف
COMMIT	يقوم بحفظ أى تعديلات على قاعدة البيانات من Insert,Update,Delete,MERGE.
SAVEPOINT name	مؤشرات لحفظ البيانات الى حد او مكان معين
ROLLBACK	يقوم بإلغاء كل التغيرات التي تمت
ROLLBACK TO SAVEPOINT name	يقوم بالرجوع حتى نقطة معينة محددة فى فقرة (savepoint). حتى يتراجع عن جزئية معينة من العمليات .

*يستخدم الامر Commit لحفظ أى تعديلات فى قاعدة البيانات فإذا قمت بعمل DML و اردت حفظ تلك التغيرات فلا بد من استخدام الامر Commit.
مثال:

```
Update emp set ename ='ASAS'
Where empno=7782;
```

ثم بعد ذلك

```
Commit;
```

فهذا يعنى انك تريد حفظ تلك التغير.

*يستخدم الامر Savepoint لامكانية انشاء علامة تذكرك بأخر التعديلات التى قمت بها و يمكنك الرجوع اليها .
مثال:

```
Insert into emp (empno,ename)
Values(10,'MOMO');
```

ثم اردت حفظ هذا التعديل بعلامة لكى يمكنك بعد ذلك الرجوع لها.

```
Savepoint a;
```

ثم قمت بعد ذلك بتعديل مرتب موظف معين مثلاً

```
Update emp set sal=2500
Where empno=4444;
```

واردت بعد ذلك الغاء تلك التغير الاخير فى المرتب .بدون الغاء جملة Insert الاولى.
وذلك عن طريق استخدام Rollback.

*يستخدم الامر Rollback لامكانية الرجوع عن كل التعديلات التى تمت او لتحديد المكان الذى ترغب فى الرجوع اليه.

مثال:

إذا اردنا الرجوع الى اخر تعديل حدث فنقوم بالآتى:

```
Rollback to a;
```