

مفاهيم قواعد البيانات
**Database Concepts and
Design**

المستوى : الرابع

رمز المقرر : ٢٢٣ حسب

المتطلبات السابقة : ١٢١ حسب

طبيعة المقرر : ساعتين نظري + ساعتين عملي

المرجع : أصول نظم قواعد البيانات - الجزء الأول

تأليف : أ.د. رامي المصري / أ.د. شامكانت نافاث

ترجمة د.م. خالد ناصر السيد

استاذة المادة

م/ ليندا البديري

التعامل مع البيانات

Data manipulation languages (DML)

وتتكون لغة التعامل مع البيانات (DML) من عدة جمل وهي كالتالي :

■ جملة إضافة بيانات إلى الجدول INSERT INTO .

■ جملة التعديل في بيانات الجدول UPDATE .

■ جملة حذف بيانات من الجدول DELETE FROM .

إضافة سجل أو عدد من السجلات إلى جدول معين (INSERT INTO).

إضافة سجل أو سجلات جديدة إلى جدول هي عملية إضافة بيانات جديدة إلى جدول معين عن

طريق استخدام جملة الإضافة (INSERT INTO).

الصيغة العامة لإضافة سجلات جديدة إلى جدول .

```
SQL > INSERT INTO جدول (عمود١ , عمود٢ , ..... عمود٣ )  
VALUES ( ..... قيمة٣ , قيمة٢ , قيمة١ ) ;
```

القواعد التي يجب التقيد بها عند الإضافة:

- يجب أن يكون عدد القيم التي سيتم إدخالها مساوياً لعدد الأعمدة المذكورة في جملة INSERT .
- يجب أن تكون القيم مرتبة بنفس ترتيب الأعمدة المراد إدخال القيم إليها ، حيث إن (القيمة ١) سوف تسجل في (العمود ١) وهكذا . كما يجب أن تكون القيم أيضاً من نفس نوع بيانات الأعمدة .
- عند إدخال قيم التاريخ والنصوص لا بد من وضعها داخل علامتي تنصيص ' ' .
- يجب إدخال قيمة للأعمدة التي لا تقبل قيمة فارغة NULL مثل أعمدة المفتاح الأساسي (Primary key) مثلاً (empno) في جدول الموظفين .
- يجوز عدم ذكر أسماء الأعمدة في جملة INSERT وفي هذه الحالة لا بد من إدخال جميع قيم الأعمدة الموجودة في الجدول حسب ترتيب الأعمدة داخل الجدول مع مراعاة نوع البيانات لكل عمود .

مثال (1) : إضافة سجل جديد إلى جدول الإدارات أي إضافة إدارة جديدة .

```
SQL> INSERT INTO dept (deptno , dname , loc )  
2 VALUES ( 50 , 'DEVELOPMENT' , 'DETROIT' ) ;
```

جدول الإدارات (DEPT) قبل الإضافة

DEPTNO	DNAME	LOC
10	ACCOUNT	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPRATIONS	BOSTON

جدول الإدارات (DEPT) بعد الإضافة

DEPTNO	DNAME	LOC
10	ACCOUNT	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPRATIONS	BOSTON
50	DEVELOPMENT	DETROIT

السجل (الصف) الذي تم إضافته

إضافة قيمة فارغة (NULL) إلى عمود :

يتم إضافة القيمة NULL إلى الأعمدة بطريقتين :

الأولى : عدم كتابة الأعمدة المراد تسجيل القيمة NULL بها في الجزء INSERT .

الثانية : أن نكتب الأعمدة ولكن نكتب قيمتها NULL داخل الجزء VALUES .

بشرط أن تقبل الأعمدة هذه القيمة أي إنها ليست عليها قيود مثل (primary key) كما في المثال

التالي :

مثال (٢) : إضافة سجل جديد إلى جدول الإدارات يحتوي هذا السجل على رقم الإدارة واسمها فقط .

```
SQL> INSERT INTO dept (deptno , dname )  
2 VALUES ( 60 , 'MIS' ) ;
```

جدول الإدارات (DEPT) بعد الإضافة

DEPTNO	DNAME	LOC
10	ACCOUNT	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPRATIONS	BOSTON
50	DEVELOPMENT	DETROIT
60	MIS	

قيمة فارغة NULL

```
SQL> INSERT INTO dept (deptno , dname , loc )  
2 VALUES ( 60 , 'MIS' , NULL ) ;
```

إضافة قيم خاصة داخل الأعمدة:

إذا أردنا إضافة تاريخ اليوم الحالي (SYSDATE) إلى العمود (HIREDATE) الموجود داخل جدول الموظفين وذلك عند إضافة سجل جديد أي إضافة بيانات موظف جديد فيمكن عمل ذلك كما هو واضح من المثال التالي :

مثال (٢) : إضافة بيانات موظف جديد إلى جدول الموظفين .

```
SQL> INSERT INTO emp
2      (empno , ename , job, mgr , hiredate , sal , comm , deptno)
3  VALUES
4  (7196 , 'AHMED' , 'SALESMAN' , 7782 , SYSDATE , 2000 , NULL , 10) ;
```

المثال السابق يمكن كتابته بحيث لا نذكر أسماء الأعمدة في الجزء INSERT ، وهنا لا بد من

كتابة كل القيم حسب ترتيب الأعمدة داخل الجدول كالآتي :

```
SQL> INSERT INTO emp
VALUES
(7196 , 'AHMED' , 'SALESMAN' , 7782 , SYSDATE , 2000 , NULL , 10) ;
```

أضافة سجلات عن طريق المتغيرات البديلة (Substitution variables)

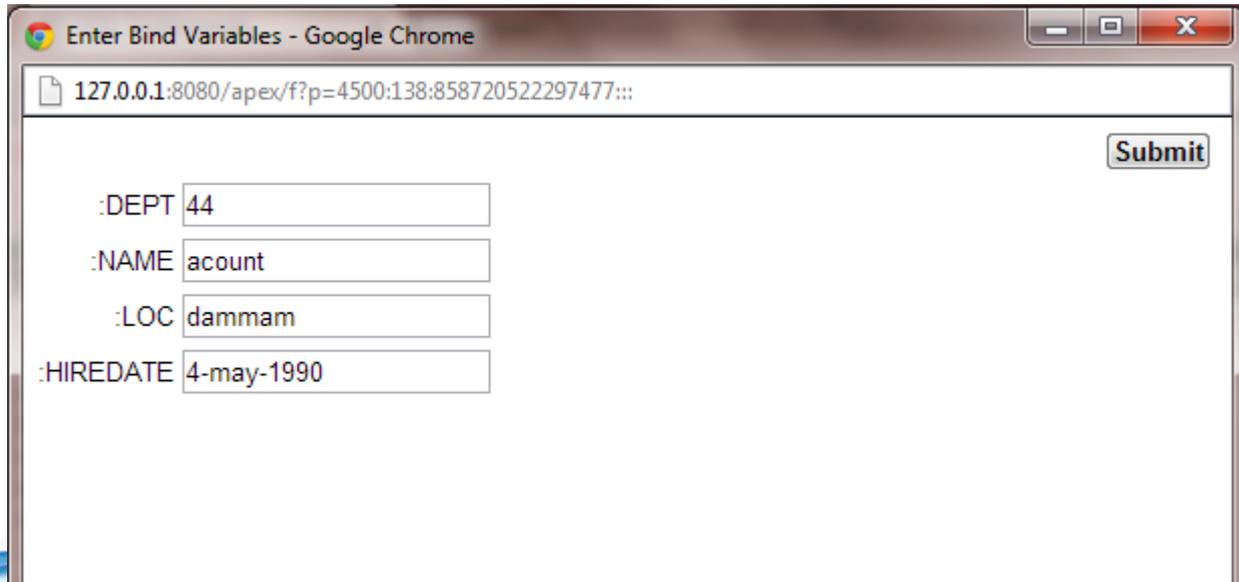
مثال (٤) : إضافة إدارة جديدة الى جدول الإدارات باستخدام المتغيرات البديلة .

```
insert into dept(deptno,dname,loc,hire)
values(:dept,:name,:loc,:hiredate);
```

المتغيرات البديلة

DEPTNO	DNAME	LOC	HIRE
44	acount	dammam	04-MAY-90
30	computer	dammam	15-MAR-13
30	computer	-	15-MAR-13
50	-	-	15-MAR-13

يتم الإدخال بهذه الطريقة عند استخدام المتغيرات البديلة



Enter Bind Variables - Google Chrome

127.0.0.1:8080/apex/f?p=4500:138:858720522297477:::

Submit

:DEPT 44

:NAME acount

:LOC dammam

:HIREDATE 4-may-1990

إضافة سجلات جديدة عن طريق نسخها من جدول آخر

```
SQL> INSERT INTO managers (id , name , salary , hiredate)
2       select empno , ename , sal , hiredate
3       from emp
4       where job='MANAGER' ;
```

في المثال السابق تم استخدام الاستعلام الجزئي التالي :

```
2       select empno , ename , sal , hiredate
3       from emp
4       where job='MANAGER' ;
```

التعديل في بيانات سجل أو سجلات معينة داخل جدول (UPDATE).

التعديل في سجل أو سجلات معينة داخل جدول هي عملية تعديل بيانات عمود أو عدة أعمدة عن طريق استخدام جملة التعديل (UPDATE).

الصيغة العامة لتعديل البيانات داخل الجدول .

```
SQL > UPDATE جدول  
      SET عمود٢ = قيمة٢ , عمود١ = قيمة١ .....  
      WHERE شرط ;
```

القواعد التي يجب التمسك بها عند التعديل :

- يجب أن يكون نوع البيانات الجديدة من نفس نوع بيانات الأعمدة المطلوب التعديل فيها.
- عند تعديل قيم التاريخ أو النصوص يجب وضع القيم الجديدة بين علامتي التنصيص الفردية (' ').
- يجب أخذ الحذر عند كتابة الجزء WHERE في جملة التعديل لتحديد أي الصفوف التي سوف يتم التعديل فيها .

مثال (٧) : تعديل اسم الإدارة رقم (30) ليصبح (EDUCATION) بدلاً من (SALES)

```
SQL> UPDATE dept
2 SET dname='EDUCATION' ← القيمة الجديدة لاسم الإدارة رقم (30)
3 WHERE deptno=30 ;
```

1 row updated ← رسالة تدل على أن التعديل تم في الإدارة رقم (30) فقط

مثال (٨) : تعديل اسم الإدارة رقم (30) ليصبح (EDUCATION) بدلاً من (SALES)

```
SQL> UPDATE dept  
2 SET dname='EDUCATION' ;
```

رسالة تدل على أن التعديل تم في جميع الصفوف ← 4 row updated

في المثال السابق تم تعديل اسم جميع الإدارات إلى (EDUCATION) ، ولذلك لا بد أن من الحذر عند التعديل في صف معين فلا بد من تحديد الصف المراد التعديل فيه باستخدام الجزء (WHERE) .

التعديل في أكثر من عمود :

في الأمثلة السابق تم التعديل في عمود واحد فقط وهو عمود أسم الإدارات ، فإذا أردنا التعديل في أكثر من عمود ، فمثلاً لتعديل رقم الإدارة والوظيفة للموظف (BLAKE) ليصبح مثل الموظف (WARD) ، فماذا نفعّل ؟ انظر المثال التالي :

مثال (٩) :

```
SQL> UPDATE emp
  2 SET (job , deptno) = (select job , deptno from emp where ename='WARD')
  3 WHERE ename='BLAKE' ;
```

نتيجة الاستعلام الفرعي (30) (MANAGER

1 row updated

حذف سجل أو سجلات معينة داخل جدول (DELETE FROM).

حذف سجل أو سجلات معينة داخل جدول هي عملية إلغاء بيانات عمود أو عدة أعمدة عن طريق

استخدام جملة الحذف (DELETE FROM).

الصيغة العامة لحذف البيانات من الجدول .

```
SQL > DELETE FROM جدول
```

```
WHERE شرط ;
```

القواعد التي يجب التقيدها عند الحذف :

● يجب الحذر عند كتابة الجزء WHERE في جملة الحذف لتحديد أي الصفوف التي سوف

يتم حذفها ، عندما لا نكتب جزء الشرط في جملة الحذف فإنه يتم حذف جميع صفوف

الجدول كاملة .

مثال (١٠) : حذف الإدارة رقم (40) من جدول الإدارات .

```
SQL> DELETE FROM dept  
2 WHERE deptno = 40 ;
```

1 row deleted.

مثال (١١) : حذف جميع الموظفين من جدول الموظفين .

```
SQL> DELETE FROM emp ;
```

14 row deleted.

مثال (١٢) : حذف جميع الموظفين المسجلين في الإدارة التي لها اسم (SALES).

```
SQL> DELETE FROM emp
```

```
2 WHERE deptno=( select deptno from dept where dname='SALES' ) ;
```

```
6 row deleted.
```

نتيجة الاستعلام الفرعي هو الإدارة رقم (30)

ماذا يحدث لو أردت حذف الإدارة رقم (10) من جدول الإدارات ؟ للإجابة على هذا السؤال دعنا

نرى هذا المثال .

مثال (١٢) :

```
SQL> DELETE FROM dept  
2  WHERE deptno= 10 ;
```

ERROR at line 1 :

ORA-02292 : integrity constraint (USR.EMP_DEPTNO_FK)

Vaiolated -child record found

عمليات قواعد البيانات (Database Transactions)

يوجد عملية تبدأ عند قيامنا بالإضافة أو التعديل أو الحذف في قاعدة البيانات هذه العملية تسمى Database transactions وهي عملية انتقال البيانات من مرحلة إلى أخرى ، فمثلاً عندما نقوم بعمل إضافة سجل إلى جدول ما فإن هذه الإضافة لا تتم بشكل نهائي إلا إذا قمنا بإصدار أمر من أوامر Database transactions وهو الأمر (COMMIT) وهذا الأمر يعني تثبيت أو حفظ البيانات بشكل نهائي سواء كنا نقوم بعملية إضافة أو تعديل أو حذف بيانات ، فهو يعادل تماماً أمر حفظ (SAVE) . وإليك بعض أوامر Database transactions وهي كالتالي :

- COMMIT يقوم هذا الأمر بحفظ البيانات التي تم إجراء عمليات الإضافة أو التعديل أو الحذف عليها .
- ROLLBACK يقوم هذا الأمر بالتراجع عن عملية الإضافة أو التعديل أو الحذف .

أمر حفظ البيانات بشكل نهائي COMMIT .

هذا الأمر ينفذ بطريقتين :

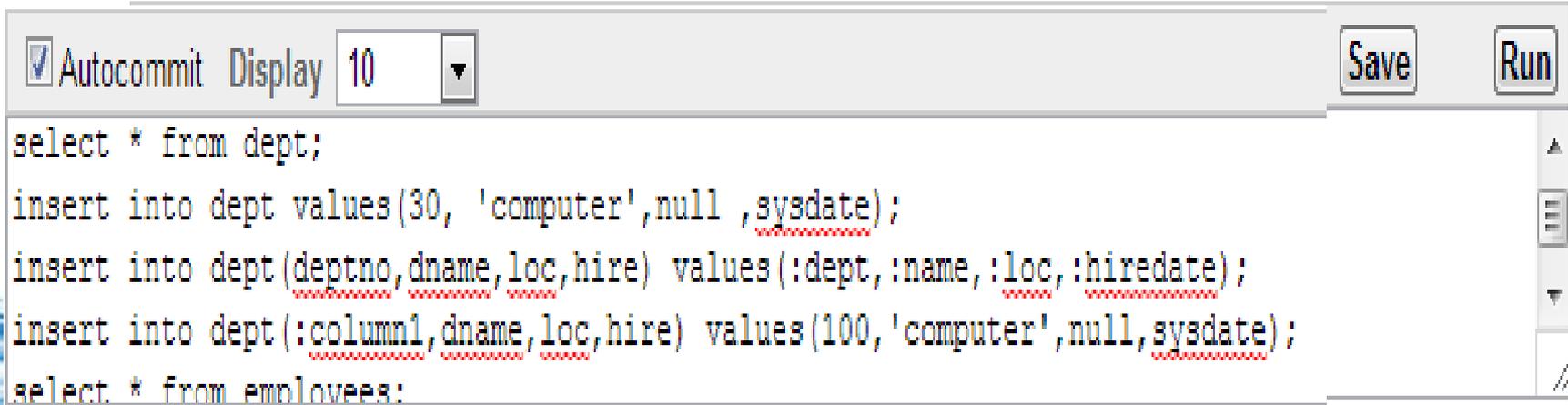
الأولى : ينفذ هذا الأمر بمجرد كتابته مباشرةً بعد عمليات الإضافة والتعديل والحذف كالتالي :

```
SQL> DELETE FROM emp  
2 WHERE deptno = 30 ;
```

6 row deleted.

```
SQL> COMMIT ;
```

الثانية : عن طريق وضع علامة «check» لتفعيل الحفظ التلقائي كما في الشكل التالي :



```
Autocommit Display 10 Save Run  
select * from dept;  
insert into dept values(30, 'computer',null ,sysdate);  
insert into dept(deptno,dname,loc,hire) values (:dept,:name,:loc,:hiredate);  
insert into dept(:column1,dname,loc,hire) values(100,'computer',null,sysdate);  
select * from employees;
```

الدوال التجميعية

Grouping functions

الدوال التجميعية لأكثر من صف والتي تسمى (GROUP FUNCTIONS) هي دوال تتعامل مع بيانات مجموعة من الصفوف لإخراج قيمة واحدة فقط ، فمثلاً إذا أردنا إيجاد مجموع ما يأخذه الموظفون من رواتب فبدلاً من جمع كل الرواتب معاً ، نستخدم دالة من الدوال التجميعية تسمى SUM وهذه الدالة تقوم بعملية جمع للرواتب وتكون نتيجتها قيمة واحدة فقط وهي المجموع .

والشكل التالي يوضح كيفية عمل الدالة التجميعية SUM .

ENAME	SAL
SMITH	800
ALLEN	1600
WARD	1250
JONES	2975
MARTIN	1250
BLAKE	2850
CLARK	2450
SCOTT	3000
KING	5000
TURNER	1500
ADAMS	1100
JAMES	950
FORD	3000
MILLER	1300

SQL > SELECT SUM(SAL)
FROM EMP ;

29025

م/ ليندا البديري

أنواع الدوال التجميعية : Type of Group Functions

الجدول التالي يبين أنواع الدوال التجميعية ووظيفة كل دالة :

FUNCTION	الدالة	وظيفتها
SUM		دالة تستخدم لإيجاد المجموع لعدد من القيم
MAX		دالة تستخدم لإيجاد أكبر قيمة من بين مجموعة من القيم
MIN		دالة تستخدم لإيجاد أقل قيمة من بين مجموعة من القيم
AVG		دالة تستخدم لإيجاد المتوسط الحسابي لمجموعة من القيم
COUNT		دالة تستخدم لإيجاد عدد القيم أو عدد الصفوف وهذه الدالة تتجاهل القيم الفارغة NULL عند عملية العد .
STDDEV DEVIATION		دالة تستخدم لإيجاد الانحراف المعياري لمجموعة من القيم
VARIANCE		دالة تستخدم لإيجاد مقدار التباين (التشتت) لمجموعة من القيم

أمثلة على الدوال التجميعية .

مثال (1) :

```
SQL> SELECT SUM(sal) , MAX(sal) , MIN(sal) , AVG(sal)
2 FROM emp ;
```

SUM(SAL)	MAX(SAL)	MIN(SAL)	AVG(SAL)
29025	5000	800	2073.21429

مثال (٢) :

```
SQL> SELECT MAX(ename) , MIN(ename)
2 FROM emp ;
```

MAX(ENAME)	MIN(ENAME)
WARD	ADAMS

مثال (٢) :

```
SQL> SELECT AVG( NVL(comm , 0) )  
2 FROM emp ;
```

AVG(NVL(comm , 0))
157.14286

```
SQL> SELECT AVG( comm )  
2 FROM emp ;
```

AVG(comm)
550

المتوسط الحسابي هنا خطأ لأنه تم القسمة على أربعة بينما عدد الموظفين هو ١٤ موظفاً .

التعامل مع الدالة COUNT .

يوجد حالتان للدالة count هما :

- COUNT(*) .
- COUNT(column) .

• تستخدم الدالة count(*) لعد جميع الصفوف داخل الجدول بما فيها الصفوف المكررة والصفوف التي تحتوي على القيم الفارغة NULL ، وإذا كانت جملة الاستفسار تحتوي على شرط where فإنها تقوم بعد الصفوف حسب جملة الشرط .

- تستخدم الدالة count(column) لعد قيم أو بيانات عمود معين مع تجاهل القيم NULL .
والأمثلة التالية توضح ذلك .

مثال (٤) :

```
SQL> SELECT COUNT(*),COUNT(comm), COUNT(deptno)
2 FROM emp ;
```

COUNT(*)	COUNT(COMM)	COUNT(DEPTNO)
14	4	14

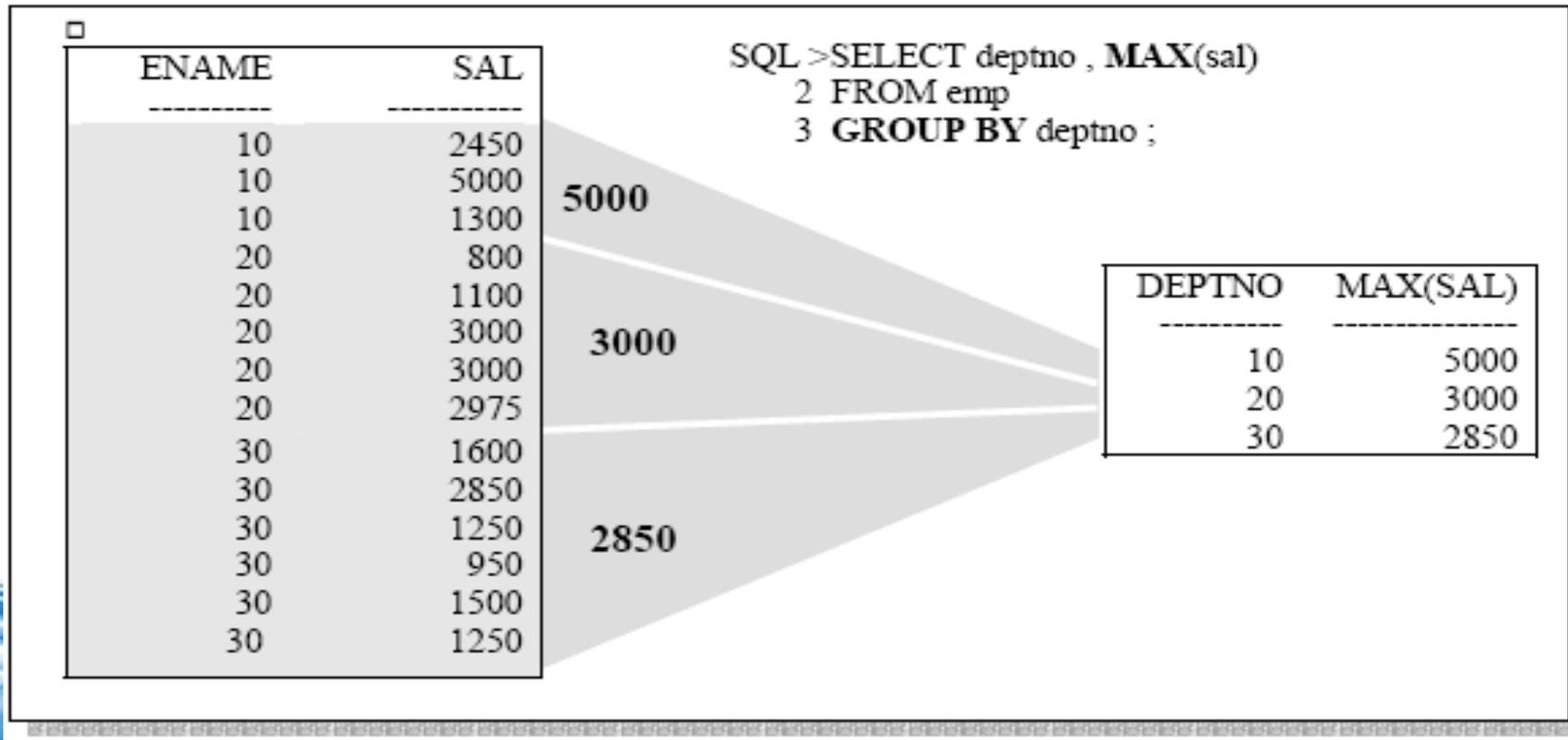
مثال (5) :

```
SQL> SELECT COUNT(comm), COUNT(*)  
2 FROM emp  
3 WHERE deptno=30 ;
```

COUNT(COMM) -----	COUNT(*) -----
4	6

إنشاء مجموعات من البيانات باستخدام الجزء GROUP BY :

يفرض أننا نريد إيجاد أكبر راتب يأخذه موظف في كل إدارة معنى ذلك أننا نقسم الموظفين داخل الجدول إلى مجموعات حسب الإدارات ثم نقوم بإيجاد أكبر راتب في كل إدارة ، ولعمل ذلك نستخدم الجزء GROUP BY والذي يقوم بتقسيم البيانات إلى مجموعات على حسب عمود معين أو أكثر .



مثال (٦) :

```
SQL> SELECT deptno , AVG(sal)
2 FROM emp
3 GROUP BY deptno
4 ORDER BY AVG(sal) ;
```

DEPTNO	AVG(SAL)
30	1566.66667
20	2175
10	2916.66667

ملاحظات على استخدام الدوال التجميعية :

- عند كتابة أي عمود داخل قائمة SELECT لابد من كتابته مع الجزء GROUP BY وذلك لأن الدوال التجميعية تتعامل مع عدة صفوف.
- يمكن استخدام الجزء ORDER BY لترتيب الصفوف مع الدوال التجميعية كما هو مبين في المثال السابق.
- لا يمكن استخدام الدوال التجميعية في الجزء WHERE ولكن نستخدم الجزء HAVING بدلاً منها.

مثال (٧) :

```
SQL> SELECT deptno , AVG(sal)
2 FROM emp
3 ORDER BY AVG(sal) ;
```

عند تنفيذ هذا المثال تم إعطاء رسالة خطأ وذلك لأننا كتبنا اسم العمود (deptno) ضمن قائمة select ولم نكتبه ضمن الجزء group by والتصحيح هذا الخطأ انظر المثال رقم (٦) .

ERROR at line 1 :

```
ORA-00937: not a single-group group function
```

رسالة خطأ

مثال (٩) :

```
SQL> SELECT deptno , AVG(sal)
2 FROM emp
3 GROUP BY deptno
4 HAVING AVG(sal) > 2000 ;
```

DEPTNO	AVG(SAL)
10	2916.66667
20	2175

مثال (١٠) :

```
SQL> SELECT job , SUM(sal)
2 FROM emp
3 WHERE job not like 'SALES%'
4 GROUP BY job
5 HAVING SUM(sal) >5000
6 ORDER BY SUM(sal) ;
```

JOB	SUM(SAL)
ANALYST	6000
MANAGER	8275

THE END
