

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

إنشاء الجداول

Creating tables



في هذه المحاضرة سوف نتعرف على قسم مهم جداً من أقسام لغة SQL وهو لغة تعريف البيانات (Data Definition Language) والتي عادةً ما يرمز لها بـ (DDL) ، وهذه اللغة هي التي تمكننا من إنشاء وتعديل وإلغاء أي كائن داخل قاعدة البيانات ، وكما هو معروف أن قاعدة البيانات تتكون من كائنات مختلفة وأهم هذه الكائنات هي الجداول (Tables) والتي سوف نركز في هذا الفصل على كيفية إنشائها والتعديل فيها وإلغائها . وإليك بعض الكائنات التي تتكون منها قاعدة البيانات :

الكائن	وصف الكائن
Table	هو الوحدة الأساسية لمكونات قاعدة البيانات والتي نستخدمها في حفظ البيانات ويتكون من عدة صفوف وأعمدة .
View	المنظير : عبارة عن جزء مؤقت من جدول معين يتكون من عدة صفوف وأعمدة ويستخدم لغرض معين بشكل مؤقت .
Sequence	سلسلة : عبارة عن سلسلة تستخدم لتوليد أرقام متتالية بشكل معين دون تكرار لذلك يفضل استخدامها لتسجيل بيانات المفتاح الأساسي داخل جدول
Index	فهرس : ويستخدم في عملية فهرست بعض الأعمدة لتسهيل عملية البحث فيها عن معلومة معينة ، وأيضاً لتقليل وقت الاستفسارات من الجداول .
Synonym	مرادفات : تستخدم لإعطاء أكثر من اسم على كائن معين .

انواع البيانات (Data types)

نوع البيانات	الوصف
Varchar2(الحجم)	تستخدم مع البيانات الحرفية المتغيرة الطول
Char(الحجم)	تستخدم مع البيانات الحرفية الثابتة الطول لا بد من تحديد طول البيانات الحرفية
Number(p,s)	تستخدم مع البيانات الرقمية ويمثل الحرف (p) الجزء الصحيح قبل العلامة العشرية ، والحرف (s) يمثل الجزء العشري بعد العلامة العشرية .
Date	تستخدم مع بيانات التاريخ والوقت
Long	تستخدم لتمثيل البيانات الكبيرة الحجم التي تصل إلى (2) جيجا بايت .
CLOB – BLOB	تستخدم لتمثيل البيانات الكبيرة مثل الصور والرسومات والتي تصل حجمها إلى أكثر من (4) جيجا بايت
Bfile	تستخدم لتخزين الملفات الكبيرة والخارجية والتي يصل حجمها إلى أكثر من (4) جيجا بايت .

الشروط الواجب توافرها عند اختيار اسم الجدول أو أسماء الأعمدة :

- يجب أن يبدأ اسم الجدول أو اسم العمود بحرف .
- يجب أن لا يزيد طول الاسم عن (30) حرفاً .
- من الممكن أن يتكون من حروف كبيرة وصغيرة وأرقام ورموز خاصة مثل (# , \$, _).
- يجب أن لا يتكرر اسم الجدول أكثر من مرة داخل قاعدة البيانات الواحدة .
- يجب أن لا يتكرر اسم عمود أكثر من مرة داخل الجدول الواحد .
- يجب أن لا يكون من الأسماء المحجوزة لأوراق مثل (FROM , SELECT ...).
- يفضل أن يكون اسم الجدول له معنى بحيث يُعبر عن نوع بيانات الجدول .

إنشاء الجداول CREATE TABLES .

الصيغة العامة لإنشاء الجداول .

```
SQL > CREATE table ( اسم الجدول  
    , نوع البيانات العمود ١  
    , نوع البيانات العمود ٢  
    ) ;
```


مثال (٢) : إنشاء جدول للإدارة رقم (30) باستخدام بيانات جدول الموظفين (EMP).

```
SQL > CREATE TABLE dept30
```

```
2 AS
```

```
3 SELECT empno , ename , sal*12 annsal , hiredate
```

```
4 FROM emp
```

```
5 WHERE deptno = 30 ;
```

الاستعلام الفرعي

```
Table created .
```

مثال (٢) : إنشاء جدول للإدارة رقم (20) باستخدام جدول الموظفين (EMP) يحتوي على أسماء
أعمدة مختلفة عن الأعمدة في جدول الموظفين (EMP) .

```
SQL > CREATE TABLE dept20
```

```
6         (emp_id , emp_name , salary , start_date )
```

```
7 AS
```

```
8 SELECT empno , ename , sal , hiredate
```

```
9 FROM emp
```

```
10 WHERE deptno = 20 ;
```

} الاستعلام الفرعي

```
Table created .
```

التعديل في الجداول باستخدام ALTER TABLE .

توفر لنا لغة الاستعلام أوراكل (SQL) إمكانية مهمة جداً وهي إمكانية التعديل في هيكل (البناء الداخلي) لجدول قد تم أنشاؤه مسبقاً باستخدام الأمر ALTER TABLE ، وعملية التعديل في الجداول تشتمل على ثلاث إمكانيات وهي إما إضافة أعمدة جديدة على الجدول أو التعديل في نوع البيانات عمود معين أو إلغاء عمود معين . كما هو موضح بالجدول التالي :

أوجه التعديل في الجدول باستخدام الأمر ALTER TABLE	
تستخدم لإضافة أعمدة جديدة إلى الجدول	ADD
تستخدم للتعديل في نوع البيانات للجدول	MODIFY
تستخدم لإلغاء عمود معين من الجدول	DROP

مثال (٤) : إضافة عمود جديد يسمى (REGION) إلى جدول الإدارات DEPT2 .

```
SQL> ALTER TABLE dept2  
2 ADD ( region VARCHAR2(20) );  
Table altered .
```

Desc dept2;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>DEPT2</u>	<u>DEPTNO</u>	Number	-	2	0	-	✓	-	-
	<u>DENAME</u>	Varchar2	14	-	-	-	✓	-	-
	<u>LOC</u>	Varchar2	13	-	-	-	✓	-	-
	<u>REGOIN</u>	Varchar2	20	-	-	-	✓	-	-
									1-4

مثال (0) : التعديل في طول بيانات العمود (DNAME) ليصبح بطول 20 بدلاً من 14 .

```
SQL> ALTER TABLE dept2  
2 MODIFY (dname VARCHAR2(20));  
Table altered .
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>DEPT2</u>	<u>DEPTNO</u>	Number	-	2	0	-	✓	-	-
	<u>DENAME</u>	Varchar2	20	-	-	-	✓	-	-
	<u>LOC</u>	Varchar2	13	-	-	-	✓	-	-
	<u>REGOIN</u>	Varchar2	20	-	-	-	✓	-	-
									1 - 4

مثال (٧) : إلغاء الجدول المسمى بـ (DEPT30) .

```
SQL> DROP TABLE dept 30 ;  
Table dropped .
```

مثال (٨) : تغيير اسم جدول الإدارات (DEPT2) ليصبح باسم (DEPARTMENT) .

```
SQL> RENAME dept2 TO department  
Table renamed.
```

القيود على الجداول

Constraints



أنواع القيود Constraints .

الجدول التالي يبين الأنواع المختلفة من القيود ومعنى كل منها :

القيود (constraint)	معنى القيد (Description)
NOT NULL	يمنع هذا القيد ترك عمود معين فارغ (لا بد أن يدخل قيمة للعمود) يطبق على مستوى العمود فقط
UNIQUE	يمنع هذا القيد تكرار القيم داخل العمود (القيم داخل العمود وحيدة) يطبق على مستوى العمود أو الجدول .
PRIMARY KEY	يستخدم لعمل مفتاح أساسي داخل الجدول ، والمفتاح الأساسي يتميز بعدم تكرار القيم ، وعدم ترك القيم فارغة أي إنه عبارة عن القيدين السابقين . يطبق على مستوى العمود أو الجدول .
FOREIGN KEY	يستخدم لعمل مفتاح ربط بين جدولين . يطبق على مستوى العمود أو الجدول .
CHECK	يستخدم لاختبار قيمة عمود بحيث لا يقبل هذا العمود إلا قيم حسب شرط معين. يطبق على مستوى العمود أو الجدول .

إنشاء القيود **Create a Constraint**

تُنشأ القيود بطريقتين هما:

- عمل القيود أثناء إنشاء الجدول .
- عمل القيود بعد إنشاء الجدول .

القيود PRIMARY KEY .

هذا القيد يتم إنشاؤه على مستوى العمود أو على مستوى الجدول ، ومعنى هذا القيد هو إنشاء مفتاح أساسي (primary key) داخل الجدول وذلك لتمييز عمود معين بحيث إن هذا العمود يكون له خاصيتان هما :

١ - عدم قبول تكرار القيم داخله .

٢ - عدم السماح بتكرار قيمته فارغة (NULL) .

فمثلاً إذا أردنا تمييز المدرسين فإننا نميزهم عن طريق الرقم الأكاديمي فهذا الرقم يجب أن يكون رقماً وحيداً لا يتكرر فكل مدرس يحمل رقماً أكاديمياً وحيداً خاصاً به وأيضاً يجب أن يكون لكل مدرس رقم أكاديمي فليس من المعقول أن نسجل بيانات مدرس دون تسجيل رقمه ، ولتحقيق ذلك نقوم بعمل عمود داخل جدول المدرسين ونطبق عليه القيد (primary key) .

مثال (1) : إنشاء القيود أثناء إنشاء جدول الإدارات وتطبيقها على مستوى الأعمدة.

```
SQL> CREATE TABLE dept (  
5      deptno NUMBER(2) PRIMARY KEY,  
6      {  dname VARCHAR2(14) NOT NULL,  
7      {  loc   VARCHAR2(13)  
8      } ;  
9
```

أسماء الأعمدة

القيود على مستوى الأعمدة

Table created .

مثال (٢) : إنشاء القيود أثناء إنشاء جدول الإدارات وتطبيقها على مستوى الجدول.

```
SQL > CREATE TABLE dept (  
2         deptno NUMBER(2),  
3         dname VARCHAR2(14) NOT NULL,  
4         loc   VARCHAR2(13) ,  
5     CONSTRAINT dept_deptno_pk PRIMARY KEY(deptno) );
```

Table created .

القيود على مستوى الجدول



القيود UNIQUE KEY

هذا القيد يتم إنشاؤه على مستوى العمود أو على مستوى الجدول ، ومعنى هذا القيد هو عدم السماح بتكرار القيم داخل العمود .

مثال (٢) : إنشاء القيد (UNIQUE) أثناء إنشاء جدول الإدارات وتطبيقه على مستوى الجدول .

```
SQL > CREATE TABLE dept (  
2         deptno NUMBER(2) ,  
3         dname VARCHAR2(14) ,  
4         loc   VARCHAR2(13) ,  
5         CONSTRAINT dept_deptno_uk UNIQUE(dname) ) ;
```

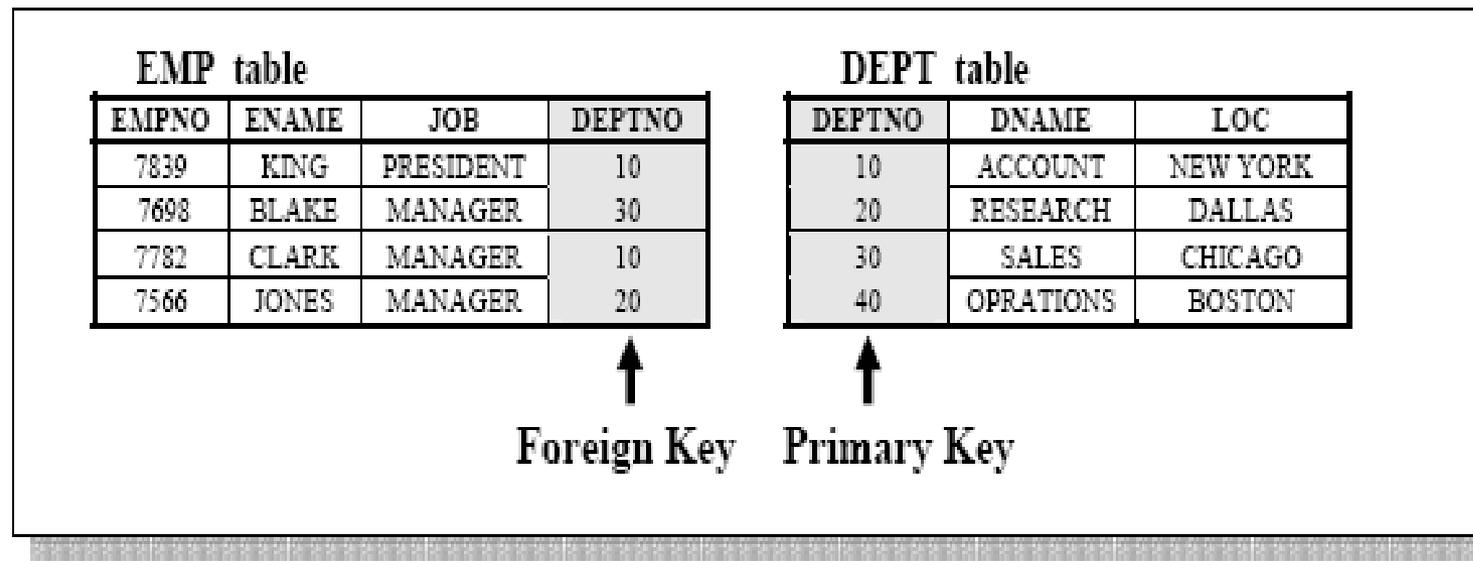
Table created .

القيود على مستوى الجدول



القيد FOREIGN KEY

هذا القيد يتم إنشاؤه على مستوى العمود أو على مستوى الجدول ، ويستخدم هذا القيد عندما نريد ربط جدولين ببعض ، فمثلاً لربط جدول الموظفين بجدول الإدارات يفرض معرفة موظفي إدارة معينة ، فإنه لا بد من وجود عمود (primary key) داخل جدول الإدارات ونفس هذا العمود يوجد في جدول الموظفين ويسمى (foreign key) ، كما في الشكل التالي :



مثال (٤) : إنشاء القيد (foreign key) على العمود deptno في جدول الموظفين وتطبيقه على مستوى العمود .

```
SQL > CREATE TABLE emp (  
2     empno  NUMBER(4) ,  
3     ename  VARCHAR2(10) NOT NULL ,  
4     job    VARCHAR2(9) ,  
5     mgr    NUMBER(4) ,  
6     hiredate DATE ,  
7     sal    NUMBER(7,2) ,  
8     comm   NUMBER(7,2) ,  
9     deptno NUMBER(2) REFERENCES dept(deptno) ) ;
```

Table created .

القيد على مستوى العمود



مثال (0) : إنشاء القيد (foreign key) على العمود deptno في جدول الموظفين وتطبيقه على مستوى الجدول .

```
SQL > CREATE TABLE emp (  
2     empno  NUMBER(4) ,  
3     ename  VARCHAR2(10) NOT NULL ,  
4     job    VARCHAR2(9) ,  
5     mgr    NUMBER(4) ,  
6     hiredate DATE ,  
7     sal    NUMBER(7,2) ,  
8     comm   NUMBER(7,2) ,  
9     deptno NUMBER(2) ,  
10    CONSTRAINT emp_deptno_fk FOREIGN KEY (deptno)  
11           REFERENCES dept(deptno) ) ;
```

Table created .

القيد على مستوى الجدول



القيود CHECK

مثال (٦) : إنشاء القيد (CHECK) على العمود DEPTNO في جدول الإدارات وتطبيقه على مستوى الجدول .

```
SQL > CREATE TABLE dept (  
2         deptno NUMBER(2) ,  
3         dname VARCHAR2(14) ,  
4         loc   VARCHAR2(13) ,  
5         CONSTRAINT dept_deptno_ck CHECK(deptno BETWEEN 10 AND 99) ) ;
```

Table created .

القيود على مستوى الجدول

إضافة قيود على الجداول Adding Constraint

يتم إضافة القيود على الجدول بعد إنشائه باستخدام الأمر Alter Table كما في الصيغة العامة

التالية :

```
SQL > ALTER TABLE اسم الجدول  
      ADD CONSTRAINT اسم القيد نوع القيد ;
```

مثال (٧) : إضافة القيد (FOREIGN KEY) على العمود MGR في جدول الموظفين .

```
SQL > ALTER TABLE emp  
2     ADD CONSTRAINT emp mgr fk  
3     FOREIGN KEY(mgr) REFERENCES emp(empno) ;
```

Table altered .

مثال (٨) : إضافة القيد (PRIMARY KEY) على العمود DEPTNO في جدول الإدارات .

```
SQL > ALTER TABLE DEPT  
2 ADD CONSTRAINT dept_deptno_pk PRIMARY KEY(deptno) ;
```

Table altered .

إزالة القيود من الجداول DROP.

يتم إزالة القيود من الجدول عن طريق استخدامنا لأمر Alter Table ، كما في المثال التالي :

مثال (٩) : إزالة القيد المسمى بـ (emp_mgr_fk) من جدول الموظفين .

```
SQL > ALTER TABLE emp  
2 DROP CONSTRAINT emp_mgr_fk ;
```

Table altered .

مثال (١٠) : إزالة العلاقة بين جدولي الموظفين والإدارات عن طريق إزالة المفتاح الأساسي من جدول الإدارات وإزالة توابعه .

```
SQL> ALTER TABLE dept  
2 DROP primary key CASCADE ;
```

Table altered .

استعراض القيود المطبقة علي الجداول

مثال :

إستعراض القيود المطبقة على الجداول

```
select constraint_name , constraint_type  
from user_constraints;
```

CONSTRAINT_NAME	CONSTRAINT_TYPE
SYS_C003108	O
JHIST_DEPT_FK	R
JHIST_EMP_FK	R
JHIST_JOB_FK	R
JHIST_DATE_INTERVAL	C
JHIST_JOB_NN	C
JHIST_END_DATE_NN	C
JHIST_START_DATE_NN	C
JHIST_EMPLOYEE_NN	C
DEPT_MGR_FK	R
More than 10 rows available. Increase rows selector to view more rows.	

- الحرف C يعني أن نوع القيد هو CHECK .
- الحرف P يعني أن نوع القيد هو Primary key .
- الحرف R يعني أن نوع القيد هو Foreign key .
- الحرف U يعني أن نوع القيد هو UNIQUE .
- أما نوع القيد NOT NULL فيظهر مثل القيد CHECK .

استعراض القيود المطبقة علي الجدول

مثال :

إستعراض القيود المطبقة على جدول JOBS

```
select constraint_name , constraint_type  
from user_constraints  
where table_name = 'JOBS';
```

CONSTRAINT_NAME	CONSTRAINT_TYPE
JOB_TITLE_NN	C
JOB_ID_PK	P

THE END
