

# سلسلة تعلم

ORACLE

## بسهولة

### تابع إنشاء القيود على حقول الجداول

### الدرس السادس

قمنا في درسنا السابق بشرح الأربعة قيود الأولى ، وفي درسنا هذا اليوم القصير في مادته العلمية .. المهم في مضمونه ؛ سنكمل إن شاء الله القيدين الأخيرين ، وهما الأهم من جميع القيود لأنهما المسؤولان عن ربط الجداول بعضها ببعض من حيث تكوين العلاقات بينها .. وهما : قيد المفتاح الرئيسي وقيد المفتاح الثانوي (الأجنبي) ..

## ٥- القيد الخامس ( Primary Key ) :

وظيفة هذا القيد إعطاء حقل معين من عدة حقول في جدول ما ؛ صفة المفتاح الرئيسي في هذا الجدول ..

**والفئة الرئيسية** كما عرفناه في الدرس الثالث في دورة مفاهيم قواعد البيانات أنه : المفتاح الذي يُحدد بشكل وحيد ومتفرد بحيث يتميز عن غيره ، فلا تتكرر قيمته في أكثر من حقل واحد ، ولا يقبل قيمة Null (أي لا يمكننا أن نترك الحقل فارغاً بدون قيمة) .

• مثال بطريقة In line :

```
SQL> create table student
2 ( std_no number(10)
3 primary key,
4 std_name varchar2(50)
5 );
```

**نلاحظ هنا :**

أننا أنشأنا جدول الطلاب بمفتاح رئيسي هو حقل رقم الطالب (std\_no) ، فن تعامل مع بيانات الطلاب ، بواسطة هذا الحقل والذي هو مفتاح رئيسي ؛ ونستطيع من خلاله البحث عن طالب معين

وكذلك عند حذف طالب معين ، بمجرد أن ندخل رقم الطالب ، ويكون هذا المفتاح رئيسياً في جدول الطلاب ، وأجنبياً في الجدول الأخرى ..

مثال بطريقة Out line :

```
SQL> create table student
2 ( std_no number(10) ,
3 std_name varchar2(50),
4 constraint pk_std_01
5 primary key (std_no)
6 );
```

ملاحظة هامة جداً :

في بعض الحالات نضطر إلى إنشاء أكثر من حقل رئيسي في جدول واحد كما في ال Access وفي

ORACLE نستطيع ذلك ولكن بالطريقة الثانية الغير خطية ( Out line ) .

## ٦- القيد السادس ( Foreign Key ) :

وظيفة هذا القيد إعطاء حقل معين من عدة حقول في جدول ما ؛ صفة المفتاح الأجنبي في هذا الجدول .

**والمفتاح الأجنبي** كما عرفناه في الدرس الثالث في دورة مفاهيم قواعد البيانات أنه :

عبارة عن حقل (صفة) أو أكثر يستخدم للربط بين جدولين ، وسُمي المفتاح الأجنبي بهذا الاسم لأنه ليس من الحقول الموجودة أصلاً في الجدول ، أي أنه عبارة عن حقل أو أكثر تُضاف إلى جدول لربطه مع جدول آخر .

ويستخدم المفتاح الأجنبي كمؤشر مقابل للمفتاح الرئيسي ، بمعنى آخر فإن المفتاح الأجنبي هو عبارة عن حقل (صفة) أو أكثر تُضاف لجدول لربطه مع جدول آخر ، مع الالتزام بوجود مفتاح رئيسي مقابل مع ملاحظة أن المفتاح الأجنبي يجب أن يكون من نفس نوع بيانات ( Data Type ) المفتاح الرئيسي ، فلو كان المفتاح الرئيسي من النوع رقم مثلاً ، يجب أن يكون المفتاح الأجنبي من النوع رقم ، لذلك يعتبر الأجنبي مؤشر للرئيسي .

• مثال بطريقة In line :

حتى ننشئ مفتاحاً أجنبياً في جدول ما ، يجب أن يكون هذا المفتاح رئيسياً في جدول آخر أنشأناه سابقاً ، وفي هذا المثال سنقوم بإنشاء مفتاح رئيسي في جدول الأقسام وهو رقم القسم ، ثم نقوم بإنشاء حقل أجنبي في جدول الطلاب يعود لهذا الرئيسي ، وبالمثال يتضح المقال !

- إنشاء جدول الأقسام ، ويسمى هذا الجدول بالرئيسي (Mister) أو الأب ..

```
SQL> create table section
2 ( sec_no number(2)
3 primary key ,
4 sec_name varchar2(20) not null
5 );
```

نلاحظ أننا أنشأنا جدول الأقسام ؛ وفيه رقم القسم (sec\_no) كمفتاح رئيسي ، واسم القسم (sec\_name) بقيد ألا يكون فارغاً ..

- إنشاء جدول الطلاب ، ويسمى هذا الجدول بالتفصيل (Detail) أو الابن ..

```
SQL> create table student (
2 std_no number(7)
3 primary key,
4 std_name varchar2 (30) not null,
5 sec_no number(2) references
6 section (sec_no)
7 );
```

أنشأنا جدول الطلاب ؛ وفيه رقم الطالب كمفتاح رئيسي .. واسم الطالب بقيد ألا يكون فارغاً .. لكن الأهم :

- أننا عرفنا حقل واسمه رقم القسم (sec\_no) الذي باللون الأخضر) وهذا أمر ضروري جداً أن نعرف الأجنبي أولاً في الجدول التفصيل ، ثم نبين أنه يُوْشِرُ إلى مفتاح رئيسي في جدول آخر بحجم هو نفس حجم رقم القسم في جدول الأقسام .. ، لأننا نعلم أن المفتاح الأجنبي يجب أن يكون تعريفه من نفس نوع بيانات وحجم المفتاح الرئيسي الذي يُوْشِرُ إليه .. ولكن ليس شرطاً أن يكون بنفس الاسم ..

- كتبنا كلمة references والتي تعني أنه يُوْشِرُ إلى المفتاح الرئيسي في جدول الأقسام Section وهو المفتاح (sec\_no باللون الرمادي) ولا ننسى أن ماسبق كله هو بطريقة In line .

- مثال بطريقة Out line :

نقوم بإنشاء الجدول الثاني وهو التفصيل بهذه الطريقة ، لأن الجدول الأول لايهمنا إنشائه بأي طريقة ..

لا يمكننا حذف جدول الأقسام ، إلا بعد أن نحذف جدول الطلاب لوجود علاقة بينهما ، لنجرب ونكتب ..

```
SQL> drop table section;
```

ستظهر لنا رسالة خطأ ، مفادها أنه لا يمكننا حذف الجدول ، لارتباطه بحقل رئيسي بجدول الطلاب ..

فنقوم أولاً بحذف جدول الطلاب ، لإنشائه مرة أخرى بطريقة **Out line** ..

```
SQL> drop table student;
```

Table dropped.

ولا نحذف جدول الأقسام بل نبقىه كما هو ..

وحتى نعرف الجدوال التي لدينا الآن نكتب :

```
SQL> select *from tab;
```

فتظهر لنا النتيجة التالية ..

TNAME	TABTYPE	CLUSTERID
SECTION	TABLE	

الآن ننشئ جدول الطلاب كما أنشأناه بطريقة **In line** أما المفتاح الأجنبي فسننشئه إن شاء الله بطريقة **Out line** ..

```
SQL> create table student (  
2 std_no number(7)primary key,  
3 std_name varchar2 (30) not null,  
4 sec_no number(2),  
5 std_age number(2),  
6 constraint fk_std_01  
7 foreign key (sec_no)  
8 references section (sec_no)  
9 );
```

نلاحظ أنه بالإضافة إلى وجود تعريف للمفتاح الرئيسي في هذا الجدول وهو رقم الطالب (**std\_no**) ، احتوى هذا الجدول أيضاً مفتاحاً ثانوياً وهو رقم القسم (**sec\_no**) ، فيسمح تعريف المفتاح الثانوي بتحديد الحقل المرجع في الجدول المرتبط به ، والذي يمكن أن يأخذ اسماً آخر ..

ولكي يكون تعريف المفتاح الثانوي سليماً ، يجب أن يكون كلا الحقلين من نفس نوع البيانات ونفس الحجم كما ذكرنا سابقاً .. (جرب تعريف الحقل الثانوي بنوع آخر أو بحجم يختلف عن الحقل المؤشر إليه) .

وعند تعريف المفتاح الثانوي فإننا نخبر أوراكل بأننا نريد إنشاء تكامل مرجعي بين الحقل رقم القسم (sec\_no) في الجدول student والجدول section .  
وهذا الأمر يمنع الحقل في جدول الإبن : student ، من احتواء قيمة غير موجودة في العمود المرتبط به في الجدول الأب : section ..

### ملاحظة هامة جداً :

عندما يكون لدينا جدولين أحدهما رئيسي (Mister) أي أب ، والذي تعتمد عليه الجداول الأخرى في البيانات .. والثاني تفصيلي (Detail) أي ابن ..  
وكان هناك علاقة بين الجدولين بواسطة مفتاح أجنبي فإنه لا نستطيع أن نحذف البيانات الموجودة في الجدول الأب الـ (Mister) ..  
وتوجد لهذه المشكلة طريقتان لحلها :

- الطريقة الأولى الأسهل والأضعف : حذف البيانات في الجدول الابن الـ (Detail) ثم حذف البيانات في الجدول الأب الـ (Mister) ..

```
SQL> delete from student ;
```

```
SQL> delete from section ;
```

- الطريقة الثانية الأقوى والأمثل : وهي أنه عند تصميم الجدول الابن الـ (Detail) فإننا نضيف بعد تعريف المفتاح الأجنبي العبارة التالية ..

### On delete cascade

إذا نكتب الجدول الابن الـ (Detail) وهو جدول الطلاب لدينا كما يلي ..

```
SQL> create table student (  
2 std_no number(7)primary key,  
3 std_name varchar2 (30) not null,  
4 sec_no number(2),  
5 std_age number(2),  
6 constraint fk_std_01  
7 foreign key (sec_no)  
8 references section (sec_no)  
on delete cascade  
9 );
```

وعند كتابتنا لهذه العبارة بعد تعريف الحقل الثانوي ، فإن أوراكل لن يسمح فقط بحذف السجل المسار إليه من الجدول الأب وهو الأقسام (section) ، وإنما سيتتالى الحذف في الجدول الابن أيضاً وهو جدول الطلاب (student) .

# وأخيراً .. رجاء حار إليكم ..

تطبيق كل مثال أورده هنا مع كل قيد وبكلا الطريقتين ، وبدون نسخ الجدول ولصقه ، قم بكتابة كل تعليمة بنفسك حتى تستفيد وتتعلم ..

## ملخص الدرس :

## ملخص هذا الدرس هو زيادة الدرس ، وبالتفصيل

## الممل فاحرص على فهمه جيداً .

\* **القيد ( Primary Key )** أي المفتاح الرئيسي :

- وظيفته إعطاء حقل معين من عدة حقول في جدول ماصفة المفتاح الرئيسي في هذا الجدول ، ويكون المفتاح الرئيسي في جدول ما ، وأجنبياً في الجداول الأخرى ..
- في بعض الحالات نضطر إلى إنشاء أكثر من حقل رئيسي في جدول واحد ، ونستطيع عمل ذلك بالطريقة الغير خطية ( Out line ) فقط .

\* **القيد ( Foreign Key )** :

- وظيفته إعطاء حقل معين من عدة حقول في جدول ما ؛ صفة المفتاح الأجنبي في هذا الجدول ، وهو عبارة عن حقل (صفة) أو أكثر تضاف لجدول لربطه مع جدول آخر ، مع الالتزام بوجود مفتاح رئيسي مقابل .
- يجب أن المفتاح الأجنبي يجب أن يكون من نفس نوع بيانات (Data Type) المفتاح الرئيسي ..
- الصيغة العامة لتعريف حقل أجنبي في جدول بالطريقة الغير خطية هي :

اسم القيد constraint

اسم الحقل كما عرفناه في الجدول الحالي وهو جدول الأبن foreign key

(الحقل المرجع إليه وهو الرئيسي في الأب) الجدول الأب references

on delete cascade

وهذا شرح الصيغة بالتفصيل الممل ...

- **constraint** كلمة تعني أن ما بعدها سيكون اسماً لقيد معين على حقل معين ، وهي محجوزة كما عرفناها سابقاً وتكتب في الطريقة الغير خطية فقط ..
- **foreign key** كلمة تعني أن ما بعدها سيكون اسماً لقيد المفتاح الثانوي (الأجنبي) .
- اسم القيد وهنا نكتب اسم القيد ، وكلما كان اسم القيد معبراً وسهلاً كان أفضل ، فمثلاً أعلاه في مثال الطريقة الثانية ، كان اسم القيد هو fk\_std\_01 فالجزء الأول fk

**الجواب منطقي ... أنه عندما تنشئ مستقبلاً بإذن الله عشرات وقد تصل إلى مئات الجد وال ،  
وتريد أن تغير في قيد حقل معين أو أن تلغي قيد حقل معين ..فإنك ستمكث وقتاً ليس  
بالييسر من أجل ذلك ..أما عند تسميته بهذه الطريقة فستصل إليه بسرعة تامة إن شاء الله  
وتعد له في أسرع وقت ..**

- \*\*\*\*\*

**Email: [amaar1422@hotmail.com](mailto:amaar1422@hotmail.com)**