تمهيد

إن أهمية المعلومات في وقتنا الحاضر زاد من أهمية إدارة قواعد البيانات التي تستخدم في تنظيم البيانات والمعلومات التي تحتاجها في كثير من التطبيقات العملية ، وهكذا فقد استمر تطور علم إدارة نظم قواعد البيانات (DMBS) منذ بداية الستينات من موضوع فرعي في تطبيقات الحاسوب إلى موضوع أساسي ومكون رئيسي من مكونات علم الحاسب الحديث وتطبيقاته إذ صارت قواعد البيانات منهاجاً رئيسياً من مناهج علم الحاسوب ، وبغض النظر عن البرامج العديدة والتطبيقات المستخدمة في إدارة قواعد البيانات حيث تجد أنه لا يختلف أي نظام أو تطبيق يختص بقواعد البيانات عن غيره من حيث المفاهيم الاساسية وطريقة بناء النظام حيث ستجد إن كل الأنظمة والتطبيقات مشتركة في أساس موحد من حيث التركيب وتختلف فقط من حيث البيانات والمعلومات المدخلة للنظام .

ومن خلال هذا المنهج المبسط سواء للطلاب الدارسين أو الاشخاص المهتمين في بعلم قواعد البيانات وتطبيقاته سنتطرق إلى شرح المفاهيم الأساسية لقواعد البيانات لبناء أي نظام وفهم ماهية قاعدة البيانات ومكوناتها والتعرف على أنواع العلاقات بين جداول قاعدة البيانات ومن ثم ندرس تطبيق أوراكل الذي من خلاله سنتعلم كيف يتم بناء التطبيقات والبرامج الارشفيه التي نراها ونسمع عنها مثل البرامج المحاسبية والإدارية وبرامج شئون الطلاب وبرامج شئون الموظفين والبرامج المخزنية وكل البرامج الأخرى التي تعتمد على أرشفة وحفظ البيانات والمعلومات وللعلم هناك الكثير من اللغات والبرامج التطبيقية المتخصصة في إدارة قواعد البيانات منها على سبيل الذكر لا الحصر (أكسس – فوكس برو برادوكس – أوراكل - .... الخ) ومعظم نظم إدارة قواعد البيانات تشترك في مجموعة من الوظائف منها :

١- إضافة معلومات أو بيانات إلى الجدول .
٢- حذف البيانات القديمة .
٣- التعديل في البيانات الموجودة .
٤- ترتيب وتنظيم البيانات داخل الملف .
٥- عرض البيانات على شكل تقارير أو نماذج .

لماذا أوراكل

كما ذكرنا سابقاً سنتعلم لتطبيق مفاهيم قاعدة البيانات وقد تم اختيارها لما تمتلك من مميزات تميزها عن غيرها من التطبيقات منها :

- قاعدة قوية وآمنة ، إذ تتمتع بأمان عالي جداً وهو سبب أساسي لانتشارها الهائل رغم تكاليفها الباهظة .
- قاعدة بيانات ضخمة مقارنة بغيرها .
- الميزة العالية في التعامل مع الشبكات .
- مقرر دراسي أساسي في جميع المعاهد والجامعات .

وأريد التوضيح هنا أن أوراكل ليست لغة برمجية وإنما لغة قواعد بيانات مبرمجة لقواعد البيانات فيجب علينا أن نعرف هذا الفرق الأساسي بين لغات قواعد البيانات المتخصصة ولغات البرمجة الأخرى ، فأوراكل شبيه بـ Microsoft Office Access وأقرب إلى Microsoft SQL Server

فهذا يعني أنها قاعدة بيانات وليست لغة برمجة عامة فيجوال بيسك لغة برمجة يمكن من خلالها عمل برامج متعددة الأغراض مثل برامج الرسم والألعاب كما يمكن ربطها بقاعدة بيانات ، بينما لغات أو تطبيقات قواعد البيانات تكون متخصصة بحفظ وفرز وأرشفة البيانات .

وسنبدأ في دراسة أوراكل من الصفر إن شاء الله حيث يوجد فرعين لمن أراد تعلم أوراكل هما :

الفرع الاول :- مدير قاعدة بيانات أوراكل (ADMINISTRATOR)

وينقسم إلى :

- لغة الاستعلام البنيوية (SQL *PLUS)
- لالغة برمجة قاعدة البيانات (PL/SQL)

الفرع الثاني : مطور قاعدة بيانات أوراكل (DEVELOPER)

وينقسم إلى :

- باني الشاشات (FORM)
- باني التقارير (REPORT)

ونحن من خلال دراستنا بصدد الوصول إلى مطور قاعدة البيانات ولكن قبل ذلك يجب أن تتقن لغة الاستعلام SQL ولغة البرمجة PL/SQL ومن خلال هذا المنهج سوف ندرس كورس كامل على لغة الاستعلام البنيوية SQL .

SQL

لغة الاستعلام البنيوية

STRUCTURE QUERY LANGUAGE

❖ **تعريف لغة الإستعلام البنيوية Structure Query Language (SQL Plus)**

هي مجموعة من الأوامر التي تحتاجها البرامج وكذلك المستخدمين للوصول إلى البيانات الموجودة ضمن قاعدة بيانات أوراكل .

تم تطوير هذه اللغة من قبل شركة IBM وكانت تسمى SYSTEM R حيث كانت تعتبر نموذج لنظام ادارة قاعدة بيانات وفي عام ١٩٧٩م طرحت شركة أوراكل أول نسخة تجارية من لغة SQL

وتعتبر لغة SQL لغة غير اجرائية لانها تتعامل مع مجموعة من السجلات في نفس الوقت وليس مع سجل وحيد ، وتحتوي لغة SQL على تعليميات تفيد المستخدمين وتمكنهم من إدارة النظام وقواعد البيانات والتطبيقات عليها ، كما أنها تحتوى على أوامر لإنجاز مهام مختلفة ومتعددة كالبحث عن بيانات والتعديل والتعامل مع السجلات وتضمن تناسق وتكامل البيانات ومن خلال SQL سيتم التعامل مع الجداول من حيث إنشاء جدول وتعديل مواصفات جدول وحذف جدول وكذلك إنشاء وحذف القيود على الجدول ، وكيفية التعامل مع البيانات من اضافة وتعديل وحذف بيانات بشكل عام ووظيفة الاستعلام بشكل خاص والتي بدورها تعتبر من أهم ما يميز لغة SQL لما تمتلكه من قدرة عالية في جلب البيانات من الجداول .

❖ **الفرق بين لغة SQL و SQL* PLUS**

SQL هي عبارة عن تعليميات للاتصال بمخدم أوراكل من خلال أي أداة أو تطبيق ، فعند كتابة أي تعليمة من تعليميات اللغة يتم تخزين التعليمية في الذاكرة المؤقتة للغة SQL وتظل التعليمية في الذاكرة حتى يتم كتابة تعليمة جديدة ، أما SQL PLUS* فهي عبارة عن أداة من أدوات أوراكل يمكنها التعرف على تعليمات SQL وإرسالها الى مخدم أوراكل لتنفيذها وكذلك تمتلك تعليمات خاصة بها .

❖ تقسم أوامر لغة (SQL Plus) إلى مجموعات كل مجموعة منها تمثل لغة فرعية وهي كتالي :

أ- أوامر لغة التحكم بالبيانات Data Control Language (DCL)

اللغة المسؤولة عن إعطاء الصلاحيات بين المستخدمين وإلغاءها .

وتحتوي على أمرين

١- GRANT : منح الصلاحيات
٢- REVOKE : منع أو سحب الصلاحيات

ب- أوامر لغة تعريف البيانات Data Definition Language(DDL)

تمتاز بأنها تتعامل مع قاموس البيانات ومع هيكل الجداول ولا تتعامل مع البيانات

وتحتوي على ثلاثة أوامر

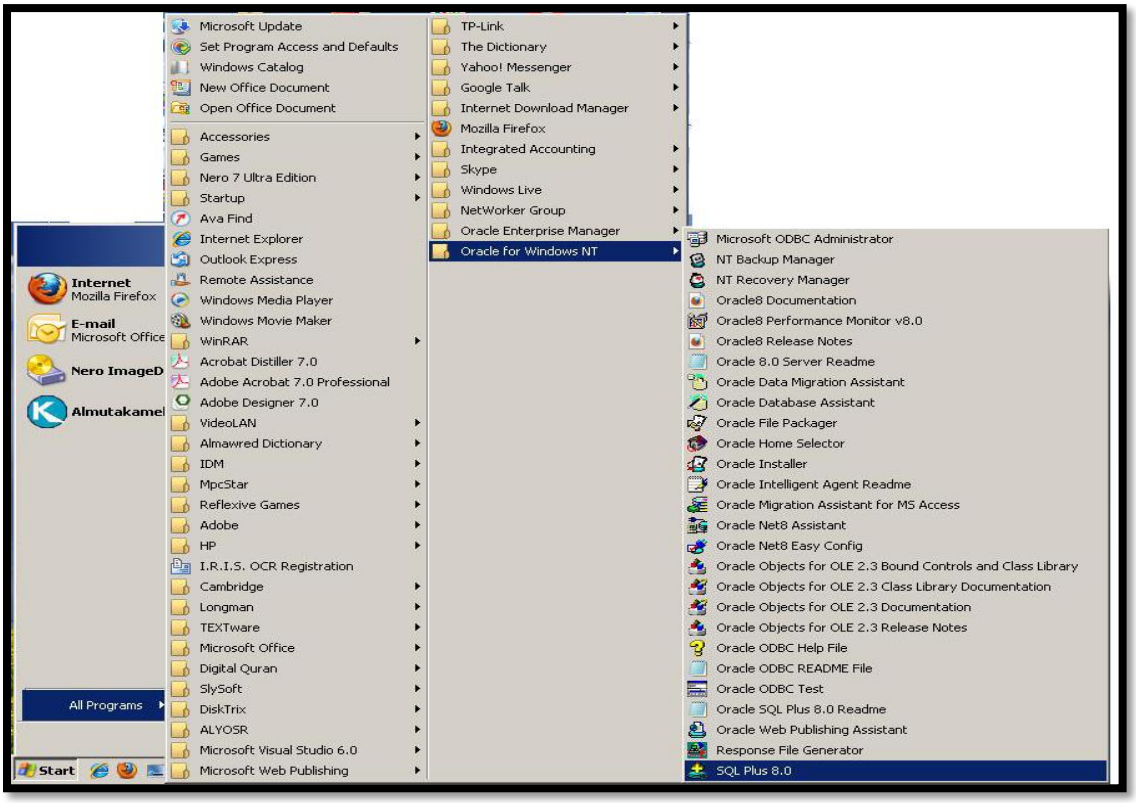
١- إنشاء جداول Create Table
٢- تعديل مواصفات جدول Alter Table
٣- حذف جدول Drop Table

ج- لغة التعامل مع البيانات Data Manipulation Language (DML)

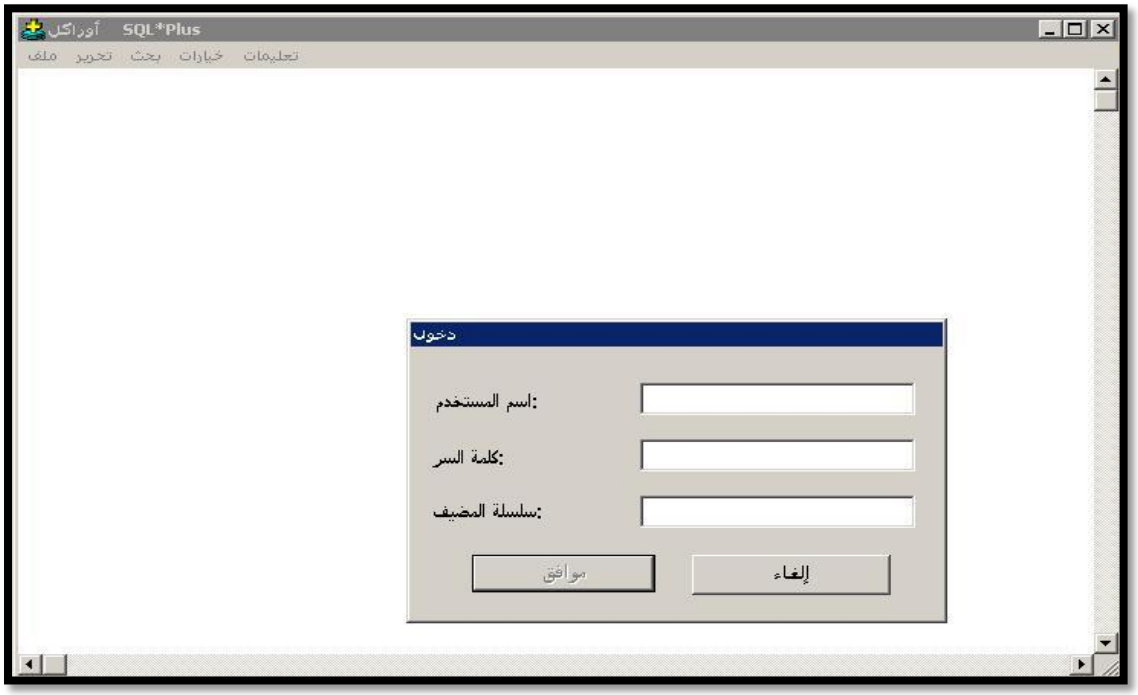
تتعامل مع البيانات مباشرة وتحتوي على أربعة أوامر :

١- إدخال بيانات إلى جدول Insert Into
٢- تعديل بيانات من جدول Update
٣- حذف بيانات من جدول Delete From
٤- الإستعلام (جلب البيانات) Select

الدخول إلى لغة أوراكل(ORACLE)←——(SQL Plus)الشكل التالي يوضح ذلك:



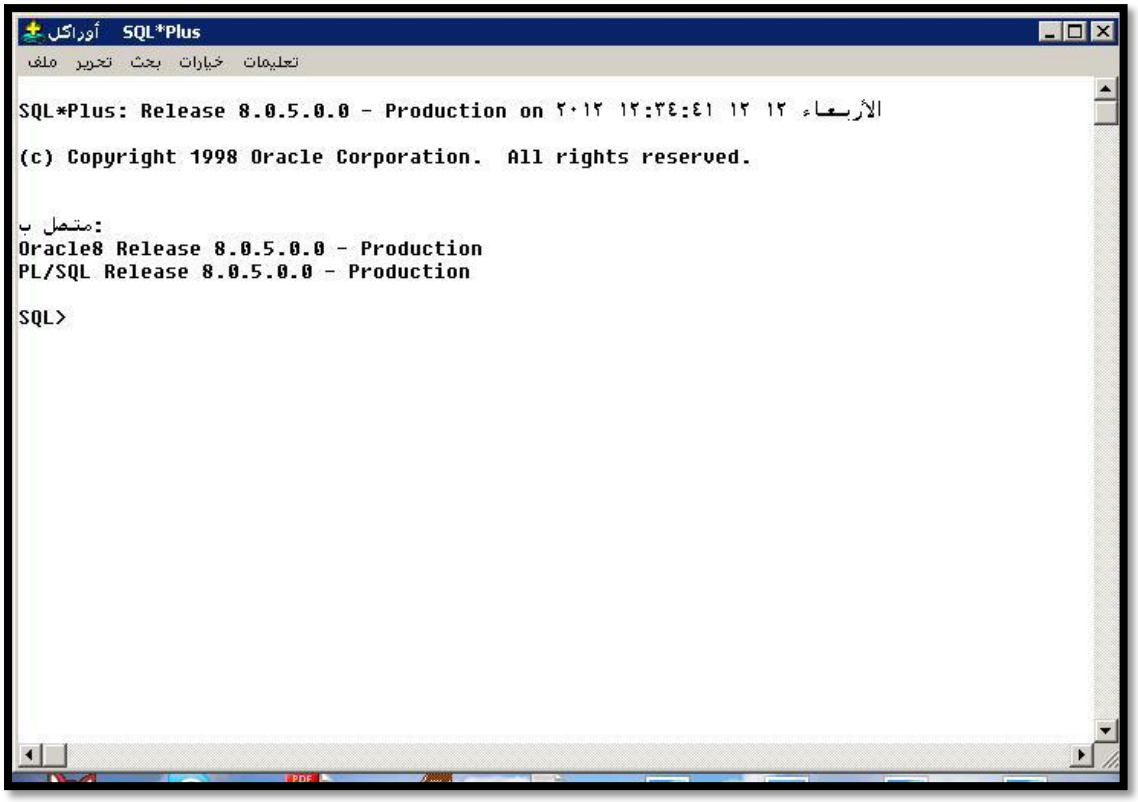
بعد ذلك تظهر النافذة التالية :



اسم المستخدم :Scott

كلمة السر : Tiger

بعد كتابة اسم المستخدم وكلمة المرور تظهر النافذة التالية تخبرك بانه متصل



أولاً:- لغة تعريف البيانات Data Definition Language(DDL)

لغة تعريف البيانات :- جزء من أجزاء لغةٍ وتعتبر أصل بناء ، حيث سيتم من خلال هذا الجزء التعرف على كيفية إنشاء الجداول والتعديل في مواصفات الجدول وكذلك حذف جدول ، وانشاء وتعديل وحذف القيود على الجداول ، وعندما تقوم بتعريف هذه البيانات (الجداول) يتم وضع إدخالات لها في قاموس البيانات الخاص بـ ORACLE

وللغة أوامر أساسية هي :-

١- إنشاء جداول Create Table
٢- تعديل مواصفات جدول Alter Table
٣- حذف جدول Drop Table

وقبل أن نتعلم كيفية استخدام الأوامر السابقة يجب أن نتعرف على أنواع البيانات التي تخزن في الجدول وهذه البيانات إما حرفية أو رقمية أو تأريخ أو بيانات أخرى والجدول التالي يوضح أنواع البيانات :

نوع الحقل	الإستخدام
Char(size)	يستخدم لتخزين البيانات الحرفية ثابتة الطول والحد الأقصى لعدد الأحرف لهذا النوع ٢٠٠٠ بايت
Var Char(Size)	ويستخدم هذا النوع لتخزين بيانات حرفية متنوعة ، والحد الأقصى لعدد الأحرف هو ٤٠٠٠ بايت
Var Char2(size)	يستخدم هذا النوع لتخزين بيانات حرفية متنوعة ، والحد الأقصى لعدد الأحرف ٤٠٠٠ بايت والفرق بين Var Char(Size) و Var Char2(size) هو أن Var Char2(size) يسمى المتغير المطاطي أي إذا حجزنا ١٠ خانات وكان المدخل يتكون من ٦ خانات سوف يقصر إلى ٦ خانات تلقائياً بعكس Var Char(Size) فسوف يحجز جميع الخانات حتى لو لم تستعمل .
Number(p,s)	يستخدم هذا النوع مع البيانات الرقمية . يمثل الحرف P طول العدد العشري كاملاً العدد الصحيح وما على يمين الفاصلة ايضاً أما الحرف s فيمثل طول الأعداد العشرية يمين الفاصلة
Date	يستخدم لتخزين البيانات ذو القيمة التأريخية والوقت
Long	يستخدم لتخزين البيانات كبيرة الحجم التي تصل إلى ٢جيجابايت
RAW	ويستخدم لخزن البيانات الثنائية ، وأقصى طول ٢٠٠٠ بايت
Clob-BLop-Bfile	تستخدم لخزن بيانات الملفات الخارجية مثل الصور والرسومات ويمكن أن يخزن هذا النوع بيانات يصل حجمها إلى ٤ جيجابايت
وهناك أنوع أخرى في الإصدارات الجديدة	

إنشاء جدول جديد

يوجد طريقتين لإنشاء جدول

١- إنشاء جدول جديد .
٢- إنشاء جدول جديد بمواصفات جدول قديم .

أولاً :- إنشاء جدول جديد

الصيغة العامة لإنشاء جدول جديد

```
SQL> create table اسم الجدول (الحقل١ نوع البيانات قيد, الحقل٢ نوع البيانات قيد,.....);
```

❖ **شرح الصيغة العامة :**

حقل ١ : اسم الحقل مثل رقم الطالب (ST_NO)

نوع البيانات :- نوع البيانات التي ستخزن في هذا الحقل (حرفي – رقمي - ....)

قيد : قيد أو شرط للبيانات التي سيتم إدخالها لهذا الحقل .

❖ **القواعد التي يجب التقيد بها عند تسمية الجدول والحقول**

١- لا يتجاوز طول اسم الجدول واسم الحقل عن ٣٠ حرفاً .
٢- يمكن أن يكون اسم الجدول أو اسم الحقل خليط من الأرقام والحروف والرموز الخاصة ولكن لا بد أن يبدأ بحرف .
٣- أن لا يكون اسم الجدول أو اسم الحقل كلمة محجوزة في اللغة .
٤- أن لا يتكرر اسم الجدول داخل قاعدة البيانات ولا يتكرر اسم الحقل داخل الجدول .
٥- يفضل أن يكون اسم الحقل واسم الجدول له معنى بحيث يعبر عن المدخلات

تكليف :-

قم بانشاء جدول باسم (Student) يحتوي على الحقول التالية :

اسم الحقل	St_no	St_name	St_address	St_mobile
نوع البيانات	Number(5)	Char(30)	Char(30)	Number(20)

```
SQL> create table student (st_no  number(5),st_name  char(30), St_address
       char(30),st_mobile number(20));
```

تم تكوين الجدول؟

الآن بعد أن تم تكوين الجدول في قاعدة بيانات أوراكل

نريد عرض البناء الداخلي للجدول نستخدم التعليمة

```
SQL> desc  اسم الجدول ;
```

```
SQL> desc student;
NAME                NULL       TYPE
---- -------- ------------------------------
ST_NO                          NUMBER(5)
ST_NAME                         CHAR(30)
ST_ADRRESS                      CHAR(30)
ST_MOBILE                       NUMBER(20)
```

ثانياً :- إنشاء جدول جديد بمواصفات جدول قديم

الصيغة العامة :-

```
SQL> create table اسم الجدول الجديد (الحقل١ ,الحقل٢ ,الحقل٣ )

2  AS

3  SELECT حقل٣,حقل٢,حقل٣ FROM  اسم الجدول القديم
```

تكليف:-

قم بإنشاء جدول بإسم tats يحتوي على حقل الرقم وحقل الإسم بنفس مواصفات حقلي الرقم والإسم لجدول بيانات الطلاب (Student)

```
SQL> Create Table Tats(T_no,T_name)
  2  AS
  3  Select  St_no, St_name From Student;
```

تم تكوين الجدول?

تم استخدام الجملة Select لجلب حقول الجدول Student حيث سيتم شرح جملة Select في محاضرات قادمة .

الآن سنقوم بعرض مواصفات الجدول للتأكد من ذلك .

```
SQL> desc Tats;
NAME                NULL       TYPE
---- -------- ------------------------------
T_NO                           NUMBER(5)
T_NAME                          CHAR(30)
```

تعديل مواصفات جدول

توفر لنا لغة الإستعلام أوراكل إمكانية مهمة هي إمكانية التعديل على مواصفات الجدول والمقصود هنا هو إمكانية إضافة حقل إلى الجدول وحذف حقل من الجدول والتعديل في نوع وحجم الحقل في الجدول وكذلك تعديل وحذف وإضافة القيود على الجدول ويتم ذلك عن طريق ALTER TABLE والأمر ALTER يأخذ ثلاثة ملحقات هي :

١- ADD تستخدم لإضافة حقل جديد .
٢- MODIFY تستخدم لتعديل نوع أو حجم الحقل .
٣- DROP تستخدم لحذف حقل من الجدول .

والصيغة العامة هي :

```
SQL> ALTER TABLE  اسم الجدول  ADD OR MODIFY OR DROP

 (اسم الحقل  نوع الحقل  الطول  القيد);
```

القواعد التي يجب مراعاتها عند استخدام الأمر ALTER TABLE :-

أ. عند استخدام الملحق

١- يمكنك زيادة حجم الحقل .

٢- يمكن تغير نوع البيانات من نوع إلى نوع بحيث لا يؤثر ذلك على بيانات الحقل إذ كانت موجودة

٣- لا يمكن تغيير طول الحقل إذا كان يحتوي على بيانات .

ب. عن استخدام الملحق

١- يجب أن يكون الحقل المراد حذفه فارغ من البيانات .

٢- لا يمكن حذف أكثر من حقل في الأمر الواحد .

٣- يجب أن يتبقى حقل واحد على الأقل بعد عملية حذف الحقول .

٤- لا يمكن حذف حقل (مفتاح رئيسي) تم ربطه بجدول آخر .

تكليف :-

١- قم بتعديل حجم حقل اسم الطالب (st_name) من ٣٠ إلى ٢٠ في جدول الطلاب (Student) :

```
SQL> Alter Table student Modify(st_name Char(20));
```

تم تعديل الجدول?

الآن سنقوم بعرض مواصفات الجدول للتأكد من ذلك .

```
SQL> desc student;
NAME                NULL       TYPE
---- -------- ------------------------------
ST_NO                          NUMBER(9)
ST_NAME                        CHAR(20)
ST_ADRRESS                     CHAR(30)
ST_MOBILE                      NUMBER(20)
```

٢- قم بإضافة حقل تأريخ ميلاد الطالب(st_date) لجدول الطلاب (Student) .

```
SQL> Alter Table student ADD (st_date Date);
```

تم إضافة الحقل إلى الجدول?

الآن سنقوم بعرض مواصفات الجدول للتأكد من ذلك .

```
SQL> desc student;
NAME              NULL       TYPE
---- -------- ------------------------------
ST_NO                        NUMBER(9)
ST_NAME                       CHAR(20)
ST_ADRRESS                    CHAR(30)
ST_MOBILE                     NUMBER(20)
ST_ DATE                        DATE
```

٣- قم بحذف حقل رقم الهاتف(st_Mobile) من جدول الطلاب (Student)

```
SQL> Alter Table Student DROP COLUMN  st_Mobile;
```

تم حذف الحقل من الجدول?

الآن سنقوم بعرض مواصفات الجدول للتأكد من ذلك .

```
SQL> desc student;
NAME              NULL       TYPE
---- -------- ------------------------------
ST_NO                        NUMBER(9)
ST_NAME                       CHAR(20)
ST_ADRRESS                    CHAR(30)
ST_ DATE                        DATE
```

حذف جدول

يمكن حذف أي جدول غير مرغوب فيه وذلك عن طريق الأمر DROP TABLE والصيغة العامة توضح استخدام الأمر :

```
SQL> DROP TABLE  اسم الجدول ;
```

عند حذف أي جدول يجب مراعاة القواعد التالية :

١- لا يمكن حذف جدول يحتوي على بيانات .
٢- لا يمكن حذف جدول يحتوي على مفتاح رئيسي وتم تعريفه في جدول آخر كمفتاح ربط (مفتاح أجنبي) إلا إذا تم استخدام الأمر الموجود بين الحاصرتين حيث يعتبر هذا الأمر اختياري ، وبدون هذا الأمر يجب حذف الجدول المرتبط أولاً .
٣- لا يمكن استعادة الجدول بعد عملية الحذف .

تكليف :

قم بحذف الجدول (Tats).

```
SQL> Drop Table Tats;
```

تم حذف الجدول?

تغيير اسم جدول

يمكن تغيير اسم جدول عن طريق الصيغة التالية :

```
SQL> RENAME  اسم الجدول القديم  TO  اسم الجدول الجديد
```

مع العلم أنه لا يمكن تغيير اسم جدول إذا كان يحتوي على مفتاح رئيسي وتم تعريفه في جدول آخر كمفتاح ربط .

تكليف :-

قم بتغير اسم الجدول (Student) إلى (base) .

```
SQL> rename student to base;
تم تغيير اسم الجدول?

SQL> desc base;
NAME              NULL       TYPE
---- -------- ------------------------------
ST_NO                        NUMBER(9)
ST_NAME                       CHAR(20)
ST_ADRRESS                    CHAR(30)
ST_ DATE                        DATE
```

سنقوم بتنفيذ جميع الأوامر التى تم ذكرها سابقاً

```
SQL> create table student(st_no number(5),st_name
char(30),st_address   char(30),st_mobile number(20));

SQL> desc student;
 NAME                      NULL    TYPE
 ------------------------------ -------- ----
 ST_NO                        NUMBER(5)
 ST_NAME                       CHAR(30)
 ST_ADDRESS                     CHAR(30)
 ST_MOBILE                     NUMBER(20)

SQL> create table tats(t_no,t_name)
  2  as
  3  select st_no,st_name from student;

SQL> desc tats;
NAME                      NULL    TYPE
 ------------------------------ -------- ----
 T_NO                         NUMBER(5)
 T_NAME                        CHAR(30)
SQL> alter table student modify(st_name char(20));
```

```
SQL> desc student;
NAME                      NULL    TYPE
 ----------------------------- -------- ----
 ST_NO                            NUMBER(5)
 ST_NAME                          CHAR(20)
 ST_ADDRESS                        CHAR(30)
 ST_MOBILE                        NUMBER(20)
SQL> alter table student add(st_date date);
SQL> desc student;
NAME                      NULL    TYPE
 ----------------------------- -------- ----
 ST_NO                            NUMBER(5)
 ST_NAME                          CHAR(20)
 ST_ADDRESS                        CHAR(30)
 ST_MOBILE                        NUMBER(20)
 ST_DATE                          DATE

SQL> drop table tats;
SQL> rename student to base;

SQL>  desc base;
NAME                      NULL    TYPE
 ----------------------------- -------- ----
 ST_NO                            NUMBER(5)
 ST_NAME                          CHAR(20)
 ST_ADDRESS                        CHAR(30)
 ST_MOBILE                        NUMBER(20)
 ST_DATE                          DATE
```

تمرين ١ :-

- قم بإنشاء جدول التخصص (SPEC) يحتوي على الحقول التالية :
  يحتوي على حقل الرقم(Sp_no) وحقل الإسم(sp_name) بنفس مواصفات حقلي الرقم والإسم لجدول بيانات الطلاب (Student) .
- قم بعرض البناء الداخلي لجدول (SPEC) للتأكد من تكوين الجدول
- قم بتعديل حجم حقل اسم التخصص (sp_name) من ٣٠ إلى ٢٠ في جدول التخصص (SPEC)
- قم بإضافة حقل الرسوم(sp_amount) لجدول التخصص (SPEC) .
- قم بعرض البناء الداخلي لجدول (SPEC) للتأكد من إضافة حقل الرسوم إلى الجدول.
- قم بتغير اسم الجدول إلى (dept).
- قم بحذف حقل الرسوم من جدول (dept) .
- قم بحذف جدول (dept) .

إنشاء القيود على الحقول

المقصود بالقيود وضع شروط معينة على الجدول لتنظيم العمليات المختلفة التي تتم على الجدول من إدخال وتعديل وحذف ، ولتوضيح ذلك نفترض أننا قمنا بإدخال بيانات لجدول التخصصات كتالي :-

رقم التخصص	اسم التخصص	الرسوم
١	محاسبة	٤٠٠٠٠
٢	حاسوب	
٣	سكرتارية	٤٠٠٠٠
٤	محاسبة	٤٠٠٠٠

لاحظ البيانات السابقة فتخصص المحاسبة تكرر ورسوم الحاسوب لم تدخل ولو قمنا بإنشاء قيود لهذا الجدول بحيث نعمل قيد لحقل الرقم يمنع تكرار أي قيمة بداخلة وكذلك حقل اسم التخصص وعمل قيد آخر لحقل الرسوم يمنع قبول القيمة الفارغة لهذا الحقل . لتجاوزنا الأخطاء والجدول التالي يوضح أنواع ووظيفة القيود التي يمكن استخدامها على حقل الجدول .

أنواع القيود

القيود	معنى القيد
NOT NULL	يمنع هذا القيد أن يترك هذا الحقل فارغاً أي يجب أن يحتوي الحقل على قيمة
Unique	يمنع هذا القيد تكرار القيمة في الحقل
Default	يعطي هذا القيد قيمة افتراضية للحقل
Check	يجري فحص معين على القيمة المدخلة للحقل بمعنى أن الحقل لن يقبل إلا القيمة المطابقة لعملية الفحص
Primary Key	يستخدم لعمل مفتاح أساسي داخل الجدول والمفتاح الأساسي يتميز بعدم تكرار القيمة وعدم ترك القيمة فارغة فهذا القيد هو عبارة عن القيدين الأول والثاني معاً كما أن هذا القيد يعتبر معرف للجداول المرتبطة بهذا الجدول
Foreign Key	يستخدم لعمل المفتاح الفرعي أي عمل ربط الجدول مع جدول آخر من خلال الحقل الذي يحتوي على هذا القيد ولكي تنشئ هذا القيد أو المفتاح الفرعي يجب أن يكون هذا الحقل مفتاح رئيسي في جدول آخر ويحمل نفس المواصفات

طريقة إنشاء القيود :-

توجد طريقتين لإنشاء القيود السابقة وهي :

- ❖ الطريقة الأولى In Line Constraint : وتعني إنشاء القيد على مستوى تعريف الحقل أي يكتب القيد في نفس سطر تعريف الحقل أو في السطر التالي مباشرة (بعد اسم ونوع الحقل).

```
SQL> create table اسم الجدول (الحقل١ نوع البيانات قيد, الحقل٢ نوع البيانات قيد,.....);
```

- ❖ الطريقة الثانية Out Line Constraint : وتعني إنشاء القيود على مستوى تعريف الجدول أي تكتب جميع القيود بعد الإنتهاء من تعريف الحقول ، وهذه الطريقة المعتمدة من الشركة وهي الأفضل حسب رأي الكثيرين ..

```
SQL> create table اسم الجدول (الحقل١ نوع البيانات, الحقل٢ نوع البيانات, .....
                              اسم القيد Constraint نوع القيد (الحقل١)
                              اسم القيد Constraint نوع القيد (الحقل٢) .....);
```

ملاحظة :-

ننبه إن هاتين الطريقتين تسريان على جميع القيود الستة التي ذكرناها ،ما عدا قيد القيمة الافتراضية للحقل (Default) فتسري عليه الطريقة الأولى فقط .

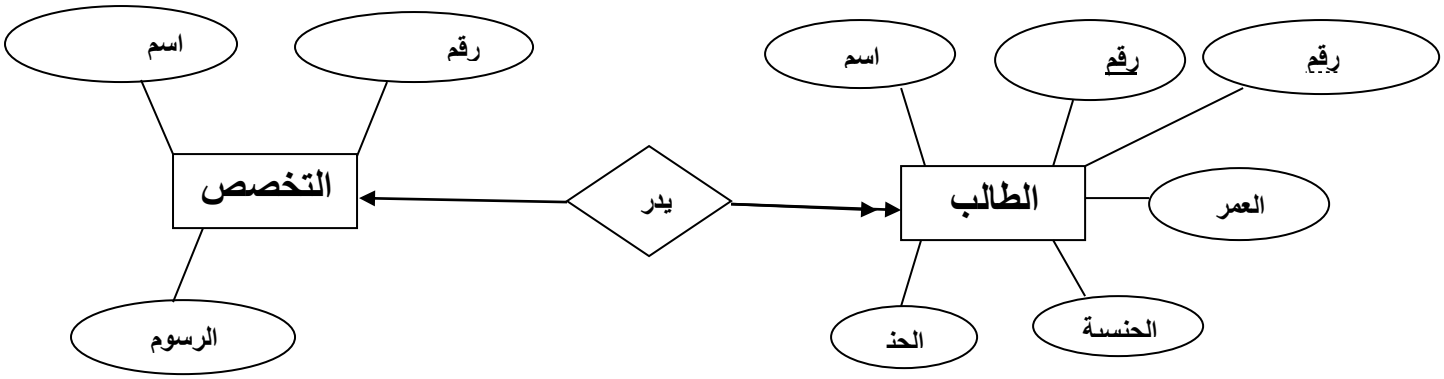
ما الفرق بين الطريقتين ؟

الفرق بين الطريقتين هو أن الطريقة الثانية سيكون للقيد اسم ويجب أن لا يتكرر اسم القيد داخل حساب واحد وكذلك لا يزيد عن ٣٠ حرف أما الطريقة الأولى فلن يكون للقيد اسم ويفضل استخدام الطريقة الثانية لأنه يمكن حذف هذا القيد بسهولة عند الحاجة كما سنرى لاحقاً ، ولكي تضمن أن لا يتكرر اسم القيد ولسهولة تذكر اسم القيد يفضل أن يكون كتالي :

```
اسم القـــــــيد

اسم الحقل_ حرفين أو أكثر من اسم الجدول _حرفين من نوع القيد Constraint
```

قم بإنشاء الجدولين الخاصة بالكينونات التالية :



نقوم أولاً : بتحويل نموذج الكينونات إلى النموذج العلائقي (الجداول)

جدول الطلاب student					
رقم الطالب St_no	اسم الطالب St_name	العمر St_age	الجنس St_sex	الجنسية st_flag	رقم التخصص sp_no

جدول التخصصات spec		
رقم التخصص sp_no	اسم التخصص Sp_name	الرسوم sp_amount

الأن سنقوم بإنشاء الجداول داخل قاعدة البيانات أوراكل

أولاً سنقوم بانشاء جدول التخصصات(Spec) وسنقوم باضافة القيود التالية :
حقل الرقم التخصص سيكون مفتاح رئيسي وحقل اسم التخصص لا يتكرر ولا يقبل القيمة الفارغة وحقل الرسوم لا يقبل القيمة الفارغة بأستخدام الطريقتين على مستوى الحقل وعلى مستوى الجدول ؟
أولاً:- باستخدام الطريقة الأولى على مستوى الحقل

```
SQL> Create Table Spec(sp_no number(2) primary Key,
  2  SP_name varchar2(50) Not Null Unique ,
  3  sp_amount number(5) Not Null);
```

?تم تكوين الجدول

```
SQL> desc spec;
NAME                 NULL        TYPE
 ------------------------------ -------- ----
 SP_NO               NOT NULL  NUMBER(2)
 SP_NAME             NOT NULL  VARCHAR2(50)
 SP_AMOUNT           NOT NULL  NUMBER(5)
```

الآن سنقوم بإنشاء الجدول بالطريقة الثانية ولكن قبل ذلك سنقوم أولاً بحذف جدول التخصصات ولأننا سنقوم بإنشائه مرة أخرى باستخدام الطريقة الثانية على مستوى الجدول

السبب في ذلك أنه لا يمكن أن يكون هناك جدولان بنفس الإسم لذا يجب عليك أولاً حذف الجدول باستخدام التعليمة drop كالتالي :

```
SQL> drop table Spec;
تم حذف الجدول ?
```

الآن بعد حذف الجدول سنقوم بإنشاء نفس الجدول باستخدام الطريقة الثانية

ثانياً :- باستخدام الطريقة الثانية على مستوى الجدول

```
SQL> create table spec(sp_no number(2),
  2                    sp_name varchar2(50),
  3                    sp_amount number(5),
  4  constraint pk_sp_no primary key(sp_no),
  5  constraint nt_sp_name check(sp_name is not null),
  6  constraint un_sp_name unique(sp_name),
  7  constraint nt_sp_amo check(sp_amount is not null));
```

تم تكوين الجدول?

لاحظ طريق كتابة القيد Not Null لحقل الإسم في المثال السابق فقد تم استخدام القيد Check لفحص الحقل بشرط أن لا يكون فارغ ،حيث لا يمكن كتابة القيد Not Null عند استخدام الطريقة الثانية إلا بهذه الطريقة .

```
SQL> desc spec;
NAME                 NULL        TYPE
 ------------------------------ -------- ----
 SP_NO               NOT NULL  NUMBER(2)
 SP_NAME                       VARCHAR2(50)
 SP_AMOUNT                     NUMBER(5)
```

ثانياً : سنقوم بانشاء جدول الطلاب(student) وسنقوم باضافة القيود التالية :

حقل رقم الطالب مفتاح رئيسي واسم الطالب لا يتكرر ولا يترك فراغ وحقل الجنس نصي إما ذكر أو أنثى والقيمة الافتراضية لحقل الجنسية يمني إذا تركت فارغة وحقل العمر محصور بين ٢٠و٢٥ فقط ورقم التخصص مفتاح ربط مع جدول التخصصات ؟

أولاً:- باستخدام الطريقة الأولى على مستوى الحقل :

```
SQL>  create table student (st_no number(6) primary key,
  2   st_name varchar2(50) not null unique ,
  3   st_sex varchar2(4) check(st_six in('انثى','ذكر')),
  4 st_flag    varchar2(10) default'يمني',
  5 st_age     number(2) check(st_age between 20 and 25),
  7 sp_no number(2) references spec(sp_no));
```

تم تكوين الجدول?

```
SQL> desc student;
NAME                 NULL        TYPE
 ------------------------------ -------- ----
 ST_NO               NOT NULL   NUMBER(6)
 ST_NAME             NOT NULL   VARCHAR2(50)
 ST_SEX                         VARCHAR2(4)
 ST_FLAG                        VARCHAR2(10)
 ST_AGE                         NUMBER(2)
SP_NO                           NUMBER(2)
```

ثانياً:- الطريقة الثانية على مستوى الجدول

سنقوم أولاً بحذف الجدول

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

تم حذف الجدول ?

```
SQL> create table student(st_no number(6),
  2 st_name varchar2(50),
  3 st_sex varchar2(4),
  4 st_flag varchar2(4) default'يمني',
  5 st_age number(2),
  6 sp_no number(2),
  7 constraint pk_st_no primary key(st_no),
  8 constraint not_st_name check(st_name is not null),
```

```
 9  constraint un_st_name unique(st_name),
 10  constraint ch_s_sex check(st_sex in('انثى','ذكر')) ,
 11  constraint ch_s_age check(st_age between 20 and 25) ,
 12  constraint fk_sp_no foreign key(sp_no)
                 references spec(sp_no));
```

ملاحظة :-

في المثال السابق وباستخدام الطريقة الثانية لم يتم تعريف القيد Defualt هذا على مستوى الجدول لان هذا القيد الوحيد الذي يعرف على مستوى الحقل فقط.

```
SQL> desc student;
NAME                 NULL        TYPE
 ------------------------------ -------- ----
 ST_NO          NOT NULL       NUMBER(6)
 ST_NAME                       VARCHAR2(50)
 ST_SEX                        VARCHAR2(4)
 ST_FLAG                       VARCHAR2(4)
 ST_AGE                        NUMBER(2)
ST_NO                          NUMBER(2)
```

- **قم بإضافة حقل تأريخ الميلاد لجدول الطلاب Student ؟**

```
SQL> alter table student add(st_date date);
تم إضافة حقل?
```

```
SQL> desc student;
NAME                 NULL        TYPE
 ------------------------------ -------- ----
 ST_NO              NOT NULL   NUMBER(6)
 ST_NAME                       VARCHAR2(50)
 ST_SEX                        VARCHAR2(4)
 ST_ADDRESS                    VARCHAR2(50)
 ST_TYPE                       VARCHAR2(6)
 ST_AVG                        NUMBER(5,2)
 SP_NO                         NUMBER(2)
 ST_DATE                       DATE
```

إضافة وإلغاء القيود للجدول :-

بالنسبة لإضافة قيد على حقل في الجدول فيمكن استخدام الطريقة (على مستوى الجدول) على كل القيود السابقة الذكر والأمثلة التالية توضح الطريقتين :-

الصيغ العامة لإضافة قيد باستخدام الطريقة الأولى (على مستوى الحقل)

SQL>Alter Table اسم الجدول ADD(Unique (اسم الحقل))

SQL>Alter Table اسم الجدول ADD(check(اسم الحقل الشرط))

SQL> Alter Table اسم الجدول ADD(Primary Key (اسم الحقل))

SQL> Alter Table اسم الجدول ADD(Foreign key (اسم الحقل)
References اسم الجدول المرتبط (اسم الحقل)

أما القيد NOT NULL فيأخذ الصيغة التالية :-

SQL> Alter Table اسم الجدول Modify(اسم الحقل NOT NULL)

الصيغ العامة لإضافة قيد باستخدام الطريقة الثانية(على مستوى الجدول):

SQL> Alter Table اسم الجدول ADD(Constraint اسم القيد
Foreign Key (اسم الحقل) References اسم الجدول المرتبط (اسم حقل الربط)

SQL> Alter Table اسم الجدول ADD(Constraint اسم القيد
Primary Key(اسم الحقل))

SQL>> Alter Table اسم الجدول ADD(Constraint اسم القيد
Check(اسم الحقل الشرط))

- **أضف القيد لحقل تأريخ الميلاد في جدول الطلاب؟**

باستخدام الطريقة الثانية على مستوى الجدول

```
SQL> alter table student ADD (constraint not_st_date
     check(st_date is not null));
```

باستخدام الطريقة الأولى على مستوى الحقول

```
SQL> alter table student add(check(st_date is not null));
```

إذا أردت استخدام القيد(NOT NULL) على حقل تأريخ التسجيل في جدول الطلاب فيكون الأمر كالتالي:-

```
SQL> alter table student modify(st_date not null);
```

```
SQL> desc student;
NAME                NULL        TYPE
 ------------------------------ -------- ----
 ST_NO              NOT NULL    NUMBER(6)
 ST_NAME            NOT NULL    VARCHAR2(5
 ST_SEX                         VARCHAR2(4
 ST_FLAG                        VARCHAR2(1
 ST_AGE                         NUMBER(2)
 SP_NO                          NUMBER(2)
 ST_DATE            NOT NULL    DATE
```

أما بالنسبة لإلغاء أو حذف قيد من جدول فلابد أن تكون قد أنشات القيد باستخدام الطريقة الثانية أي أن يكون للقيد المراد حذفه ، والمثال التالي يوضح عملية حذف قيد من جدول .

- قم بحذف القيد الخاص بحقل تأريخ الميلاد في جدول الطلاب Student

```
SQL> alter table student drop constraint  not_st_name;
تم حذف القيد للحقل?
```

تفعيل وإلغاء تفعيل القيود :-

المقصود بعملية تفعيل القيود هو أن القيد يقوم بوظيفته المعروفة من تحقق شرط الإدخال إلا إذا تم إلغاء التفعيل أي توقيف عمل القيد ومن البديهي أن لن تتم عملية تفعيل القيد إلا إذا تم إلغاء تفعيل القيد ويتم ذلك بالصيغة التالية :-

الصيغة العامة لتفعيل قيد

```
SQL> Alter Table   اسم الجدول   Enable Constraint   اسم القيد
```

الصيغة العامة لإلغاء تفعيل قيد

```
SQL> Alter Table  اسم الجدول  Disable Constraint  اسم القيد
```

- قم بإلغاء تفعيل قيد إسم الطالب(st_name) في جدول الطلاب Student ؟

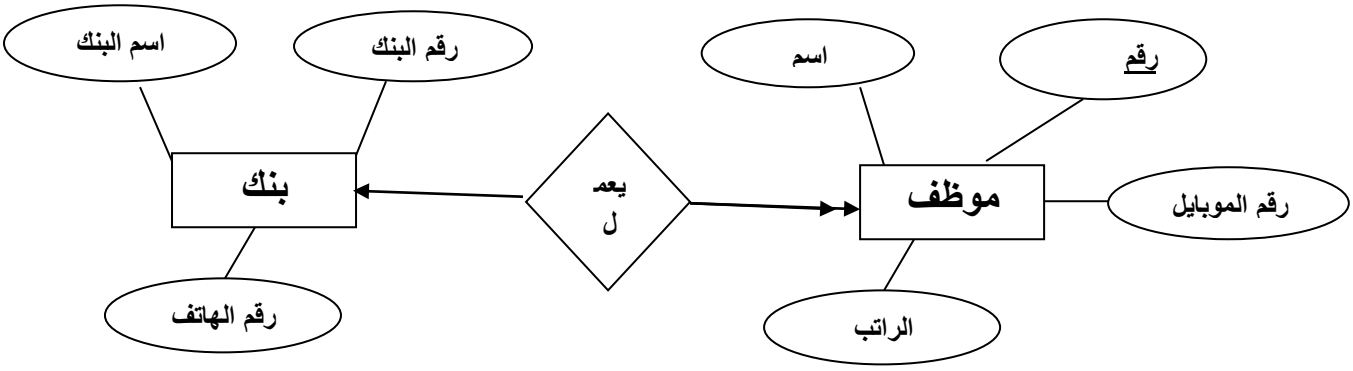
```
SQL>  alter table student enable constraint not_st_name ;
تم إلغاء تفعيل القيد?
```

- قم بتفعيل قيد إسم الطالب(st_name) في جدول الطلاب Student ؟

```
SQL> alter table student disable constraint not_st_name ;
تم تفعيل القيد?
```

أنشطة للتطبيق

قم بإنشاء الجداول الخاصة بالكينونات التالية :



جدول بنك BANK

الاسم		النوع	الحجم	قيود	المفاتيح
رمز البنك	BANK_ID	رقم	٤	NOT NULL	مفتاح رئيسي
اسم البنك	BANK_NAME	نص	٣٠	NOT NULL	ـــــ
رقم الهاتف	BANK_TEL	نص	٦	NOT NULL	ـــــ

الشفرة

```
SQL> Create Table BANK(
     BANK_ID NUMBER(4) NOT NULL ,
    BANK_NAME VARCHAR2(30) NOT NULL ,
     BANK_TEL VARCHAR2(6)  NOT NULL,
      PRIMARY KEY(BANK_ID));
      Table Created.
```

جدول موظف EMPLOYEE :

الاسم		النوع	الحجم	قيود	المفاتيح
رقم الموظف	EMP_NO	رقم	٤	NOT NULL	مفتاح رئيسي
اسم الموظف	EMP_NAME	نص	٣٠	NOT NULL	ـــــ
رقم الموبايل	EMP_MOBILE	نص	٩	NULL	ـــــ
الراتب	EMP_SAL	رقم	٦	NOT NULL	ـــــ
رمز البنك	BANK_ID_FK	رقم	٤	NOT NULL	مفتاح أجنبي

الشفرة

```
SQL> Create Table EMPLOYEE(
     EMP_NO NUMBER(4) NOT NULL ,
```

```
EMP_NAME VARCHAR2(30) NOT NULL ,
EMP_MOBILE VARCHAR2(9)  NOT NULL,
EMP_SAL VARCHAR2(6)  NOT NULL,
BANK_ID_FK NUMBER(4) NOT NULL ,
PRIMARY KEY(EMP_NO),
FOREIGN KEY(BANK_ID_FK) REFERENCES BANK(BANK_ID));
Table Created.
```

قم بإضافة الحقل التالي لجدول بنك

الاسم		النوع	الحجم	قيود
العنوان	BANK_ADDRESS	نص	50	NOT NULL

```
SQL> ALTER TABLE BANK ADD(BANK_ADDRESS VARCHAR2(50) NOT NULL);
```

تم تبديل جدول

قم بتعديل مواصفات الحقول التالية للجدول بنك

الاسم		النوع	الحجم
اسم الموظف	BANK_TEL	نص	7

```
SQL> ALTER TABLE BANK MODIFY(BANK_TEL VARCHAR2(7));
```

تم تبديل جدول

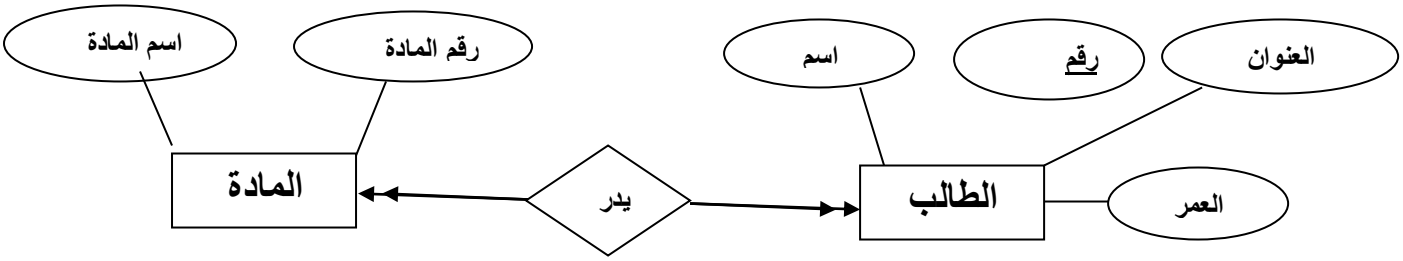
قم بتعديل مواصفات الحقول التالية للجدول موظف

الاسم		النوع	الحجم
اسم الموظف	EMP_NAME	نص	40

```
SQL> ALTER TABLE EMPLOYEE MODIFY(EMP_NAME VARCHAR2(40));
```

تم تبديل جدول

تمرين١ :-



قم بإنشاء الجداول في قاعدة بيانات أوراكل بالطريقتين على النحو التالي :
- إنشىء جدول الطالب مع تعريف القيود التالية :
رقم الطالب St_no مفتاح أساسي و حقل اسم الطالب St_name لا يتكرر ولا يترك فراغ وحقل العمر St_age محصور بين ١٠و٢٠ فقط ؟

- إنشىء جدول المادة Subject مع تعريف القيود التالية :
رقم المادة Sub_no مفتاح رئيسي وحقل اسم المادة Sub_name لا يتكرر ولا يترك فراغ ؟

- إنشىء جدول ثالث (ضعيف) جدول ربط سمه مثلاً باسم StudSub يحتوي على حقل رقم الطالب St_no وحقل رقم المادة Sub_no .

- قم باضافة حقل الدرجة deg كحقل ثالث في الجدول الضعيف(StudSub) ؟
- قم بإضافة قيد لحقل الدرجة في الجدول الضعيف بشرط تكون القيمة فيه محصورة بين ٠ و ١٠٠ فقط ؟ باستخدام الطريقتين .

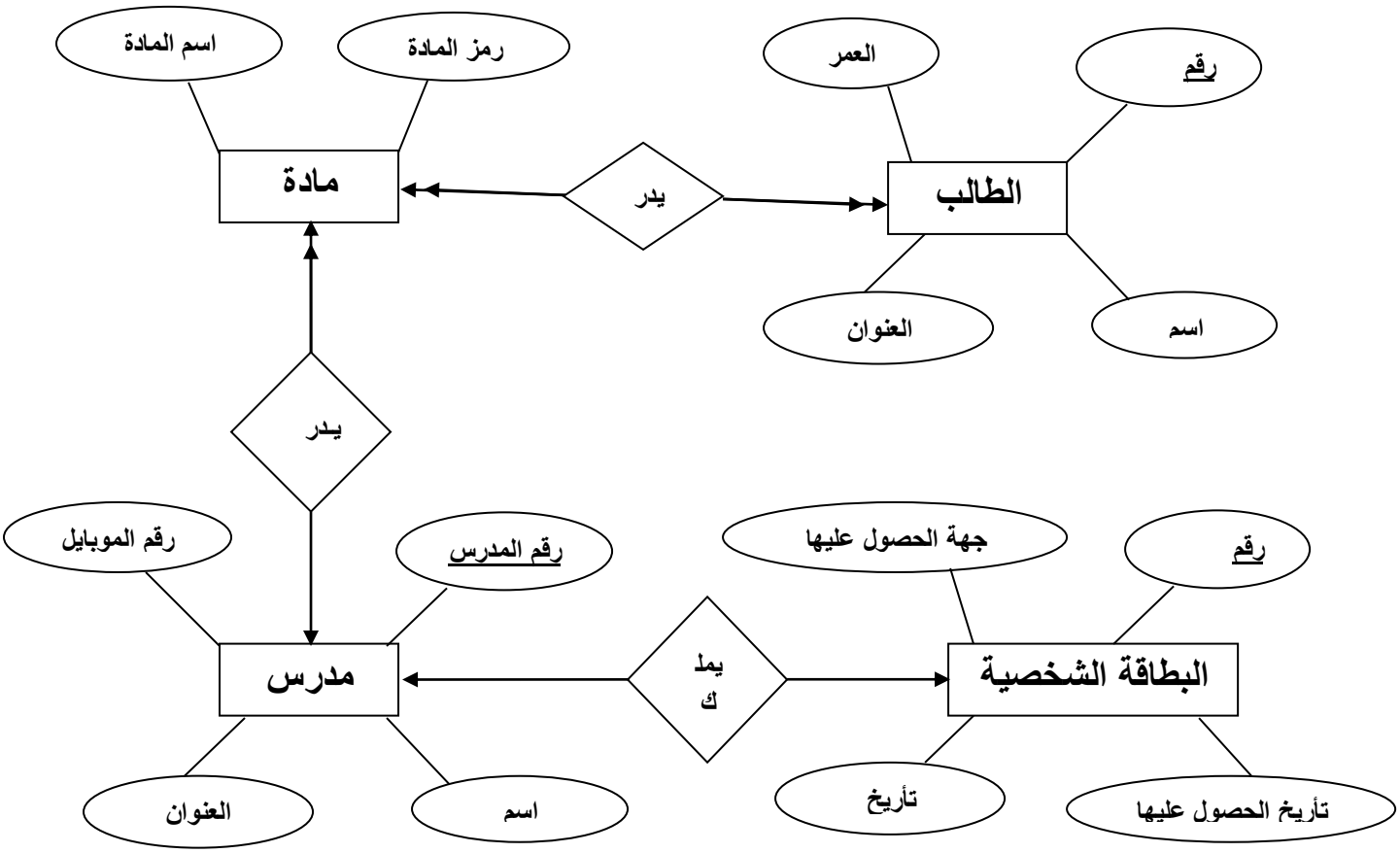
- قم باضافة حقل العنوان St_address لجدول الطلاب (Student) ؟

- قم بإضافة قيد لحقل العنوان في جدول الطلاب بشرط لا يترك فراغ؟بالطريقتين

- قم بحذف القيد لحقل اسم الطالبSt_name من جدول الطلابStudent ؟

تمرين ٢ :

قم بإنشاء الجداول الخاصة بالكينونات التالية :



ثانياً:- لغة التعامل مع البيانات Data Manipulation Language (DML)

عندما تريد إضافة ، تعديل أو حذف بيانات من قاعدة بيانات فإنك ستحتاج لمعرفة الجمل الخاصة بمعالجة البيانات والموجودة ضمن لغة الاستعلام SQL لتقوم بتنفيذها وهذه الجمل تندرج تحت مسمى Data Manipulation Language (DML)

يطلق على مجموعة من جمل الـ DML والمسئولة عن القيام بعمل محدد بـ "حركة" ففي حالة نظام بنكي وأراد أحد الزبائن تحويل مبلغ من حسابه إلى حساب أحد الاشخاص فإننا بحاجة للقيام بحركة مكونة من عدة جمل وهي :

إنقاص المبلغ من حساب الزبون ، زيادة المبلغ في حساب المحول له ، اضافة بيانات خاصة بعملية التحويل هذه ، وفي حالة عدم نجاح أحد الجمل السابقة يجب عدم اتمام الحركة نهائياً ، وتتواجد جمل خاصة للتحكم بالحركات تندرج تحت مسمى Transaction Control

تبدأ الحركة بعد تنفيذ أول جملة DML وتنتهي عند حدوث واحدة من هذه الاحداث :

- تنفيذ جملة COMMIT أو ROLLBACK
- تنفيذ جملة من جمل DDL أو DCLيؤدي إلى عمل تأكيد للحركة .
- الخروج من بيئة SQL* plus بشكل طبيعي يؤدي إلى عمل تأكيد للحركة .
- انهيار النظام أو فشل جهاز الحاسب عن العمل يؤدي إلى عمل تراجع للحركة .

سوف نتعرف من خلال هذا الجزء على كيفية التعامل مع البيانات في الجداول من حيث الإضافة والتعديل والحذف والإستعلام من خلال جمل SQL/DML التالية :

١- إدخال بيانات إلى جدول Insert Into
٢- تعديل بيانات من جدول Update
٣- حذف بيانات من جدول Delete From
٤- الإستعلام (جلب البيانات) Select

سوف نتعلم في هذه المحاضرة كيفية :-

- اضافة سجلات
- تعديل سجلات موجودة في جدول
- حذف سجلات موجودة في جدول

- التحكم بالحركات باستخدام الجمل , COMMIT , SAVEPOINT ROLLBACK

أولاً:- إدخال بيانات إلى جدول

يتم إدخال البيانات إلى الجدول عن طريق استخدام جملة الإضافة Insert Into والصيغة العامة لإضافة سجل جديد إلى جدول :

```
SQL> insert into اسم الجدول(الحقل١ ,الحقل٢ ,الحقل٣)
Values(قيمة حقل١ ،قيمة حقل٢ ،قيمة حقل٣ ،....);
```

شرح الصيغة العامة :-

اسم الجدول : المطلوب اضافة سجلات فيه

(حقل١ , حقل ٢ , حقل٣) اسماء الحقول المطلوب إدخال البيانات إليها .

(قيمة حقل١ , قيمة حقل ٢ , قيمة حقل٣) القيم المطلوب اضافتها .

القواعد التي يجب التقيد بها عند الإضافة:-

- يجب أن يكون عدد القيم التي سيتم إدخالها مساوية لعدد الحقول المذكورة مع الجملة insert
- يجب أن تكون القيم مرتبة بنفس ترتيب الحقول المراد إدخال البيانات إليها حيث أن (القيمة١) سوف تسجل في (الحقل١).
- يجب أن تكون نوع القيمة المدخلة من نفس الحقل الذي يقابلها .
- عند إدخال قيم التأريخ والنصوص يجب أن توضع بين علامتي تنصيص فردية مثل ('محمد').
- لا يشترط أن تكون عدد القيم مساوية لعدد الحقول الموجودة في الجدول أي أنه يمكن إهمال بعض الحقول بشرط أن تكون هذه الحقول تقبل القيم NULL
- يجب إدخال قيم للأعمدة التي لا تقبل القيمة NULL مثل حقل المفتاح الأساسي Primary Key شرط أن لا تتكرر هذه القيم .
- يجوز عدم ذكر اسماء الحقول في جملة insert وفي هذه الحالة يجب إدخال كل قيم الحقول وحسب ترتيبها داخل الجدول .
- في الجدول الذي يحتوي على مفتاح ربط Foreign Key يجب أن تكون القيمة المدخلة لهذا الحقل موجودة في الحقل الأساسي للجدول المرتبط .

- مراعاة القيود المدرجة مع كل حقل عند عملية الإدخال .
- التأكد من ظهور الرسالة التي تفيد الإضافة ("تم تكوين صف واحد") .
- كتابة أمر الحفظ COMMIT بعد الإضافة أو قبل الخروج من SQL

ملاحظة : إذا نسيت أسماء وأنواع الحقول يمكنك استخدام الأمر

```
SQL> Desc اسم الجدول ;
```

مثال(١) :- قم بإنشاء جدول التخصصات(Spec) مع إضافة القيود :
حقل الرقم التخصص سيكون مفتاح رئيسي وحقل اسم التخصص لا يتكرر ولا يقبل القيمة الفارغة وحقل الرسوم لا يقبل القيمة الفارغة ؟

```
SQL> Create Table Spec(sp_no number(2) primary Key,
       SP_name varchar2(50) Not Null Unique ,
        sp_amount number(5) Not Null);
```

تم تكوين الجدول؟

بعد تكوين الجدول قم بإضافة البيانات التالية إلى الجدول

sp_no	sp_name	sp_amount
1	حاسوب	20000
2	محاسبة	15000
3	سكرتارية	12000
4	تسويق	12000

الطريقة الأولى :-

```
SQL> insert into spec (sp_no,SP_name,sp_amount)
  values(1,'حاسوب',20000);
تم تكوين صف واحد
SQL> insert into spec(sp_no,SP_name,sp_amount)
     2  values(2,'محاسبة',15000);
تم تكوين صف واحد
SQL> insert into spec(sp_no,SP_name,sp_amount)
   2    values(3,'سكرتارية',12000);
تم تكوين صف واحد
SQL> insert into spec(sp_no,SP_name,sp_amount)
   2  values(4,'تسويق',12000);
تم تكوين صف واحد
SQL> Rollback;
```

تم الغاء تعديلات

الطريقة الثانية :- إضافة سجلات عن طريق المتغيرات البديلة

تمكنك لغة SQL من عمل متغيرات تمسى المتغيرات البديلة وهي مخزن مؤقت للبيانات حيث يتم تخزين قيمة داخل هذه المتغيرات وعند تنفيذ جملة SQL يتم استبدال هذه المتغيرات بقيمها ويتم تعريفها مع كتابة جملة SQL وذلك بوضع علامة (&) قبل اسم المتغير وعند تنفيذ الجملة تظهر رسالة تطلب منك إدخال قيمة المتغير

ملاحظة:-

- عند إدخال قيم المتغيرات لا تنسى عمل علامتي تنصيص فردي مع القيم النصية والتأريخ أو عمل علامتي التنصيص أثناء كتابة المتغيرات في جملة SQL مثل ('&x2',&x1) .
- الحرف (R) يستخدم لتكرار تنفيذ الأمر من أجل إضافة سجل جديد .

```
SQL> insert into spec(sp_no,SP_name,sp_amount)
  2  values(&x1,'&x2',&x3);
x1: 1 أدخل قيمة من أجل
x2: حاسوب أدخل قيمة من أجل
x3: 20000 أدخل قيمة من أجل
تم تكوين صف واحد
SQL> r
  1  insert into spec(sp_no,SP_name,sp_amount)
  2* values(&x1,'&x2',&x3)
x1: 2 أدخل قيمة من أجل
x2: محاسبة أدخل قيمة من أجل
x3: 15000 أدخل قيمة من أجل
تم تكوين صف واحد
SQL> r
  1  insert into spec(sp_no,SP_name,sp_amount)
  2* values(&x1,'&x2',&x3)
x1: 3 أدخل قيمة من أجل
x2: سكرتارية أدخل قيمة من أجل
```

```
x3: 12000 أدخل قيمة من أجل
تم تكوين صف واحد
SQL> r
  1  insert into spec(sp_no,SP_name,sp_amount)
  2* values(&x1,'&x2',&x3)
x1: 4 أدخل قيمة من أجل
تسويق :x2 أدخل قيمة من أجل
x3: 12000 أدخل قيمة من أجل
تم تكوين صف واحد
SQL> Commit;
تم التثبيت
```

مثال(٢) :- كون جدول يحتوي على رقم الطالب واسم الطالب والجنس والجنسية والعمر ورقم التخصص مع إضافة القيود التالية:-
حقل رقم الطالب مفتاح رئيسي واسم الطالب لا يتكرر ولا يترك فراغ وحقل الجنس نصي إما ذكر أو أنثى والقيمة الافتراضية لحقل الجنسية يمني إذا تركت فارغة وحقل العمر محصور بين ٢٠و٢٥ فقط ورقم التخصص مفتاح ربط مع جدول التخصصات ؟

```
SQL> create table student(st_no number(6) primary key,
2      st_name varchar2(50) not null unique ,
3    st_sex varchar2(5) check(st_sex in('ذكر'،'انثى')),
4    st_flag     varchar2(10) default'يمني',
5    st_age      number(2) check(st_age between 20 and 25),
 6   sp_no number(2) references spec(sp_no));
تم تكوين جدول
```

بعد تكوين الجدول قم بإضافة البيانات التالية عليه :

ST_NO	ST_NAME	ST_SEX	ST_FLAG	ST_AGE	SP_NO
101	محمد	ذكر	يمني	20	1
102	سوسن	انثى	يمنية	25	2
103	صالح	ذكر	يمني	20	3
104	عبدالله	ذكر	قطري	22	4
105	وفاء	انثى	يمنية	24	1
106	معتز	ذكر	يمني	23	1

```
SQL> insert into  student(st_no,st_name,st_sex,st_flag,st_age,sp_no)
  2  values(&x1,'&x2','&x3','&x4',&x5,&x6);
```

```
x1: 101 أدخل قيمة من أجل
x2: محمد أدخل قيمة من أجل
x3: ذكر أدخل قيمة من أجل
x4: يمني أدخل قيمة من أجل
x5: 20 أدخل قيمة من أجل
x6: 1 أدخل قيمة من أجل
تم تكوين صف واحد
SQL> r
  1  insert into student(st_no,st_name,st_sex,st_flag,st_age,sp_no)
  2* values(&x1,'&x2','&x3','&x4',&x5,&x6)
x1: 102 أدخل قيمة من أجل
x2: سوسن أدخل قيمة من أجل
x3: انثى أدخل قيمة من أجل
x4: يمنية أدخل قيمة من أجل
x5: 25 أدخل قيمة من أجل
x6: 2 أدخل قيمة من أجل
تم تكوين صف واحد
SQL> r
  1  insert into student(st_no,st_name,st_sex,st_flag,st_age,sp_no)
  2* values(&x1,'&x2','&x3','&x4',&x5,&x6)
x1: 103 أدخل قيمة من أجل
x2: صالح أدخل قيمة من أجل
x3: ذكر أدخل قيمة من أجل
x4: يمني أدخل قيمة من أجل
x5: 20 أدخل قيمة من أجل
x6: 3 أدخل قيمة من أجل
تم تكوين صف واحد
SQL> r
  1  insert into student(st_no,st_name,st_sex,st_flag,st_age,sp_no)
  2* values(&x1,'&x2','&x3','&x4',&x5,&x6)
x1: 104 أدخل قيمة من أجل
x2: عبدالله أدخل قيمة من أجل
x3: ذكر أدخل قيمة من أجل
x4: قطري أدخل قيمة من أجل
x5: 22 أدخل قيمة من أجل
x6: 4 أدخل قيمة من أجل
تم تكوين صف واحد
SQL> r
```

```
  1  insert into student(st_no,st_name,st_sex,st_flag,st_age,sp_no)
  2* values(&x1,'&x2','&x3','&x4',&x5,&x6)
x1: 105 أدخل قيمة من أجل
x2: وفاء أدخل قيمة من أجل
x3: انثى أدخل قيمة من أجل
x4: يمنية أدخل قيمة من أجل
x5: 24 أدخل قيمة من أجل
x6: 1 أدخل قيمة من أجل
تم تكوين صف واحد
SQL> r
  1  insert into student(st_no,st_name,st_sex,st_flag,st_age,sp_no)
  2* values(&x1,'&x2','&x3','&x4',&x5,&x6)
x1: 106 أدخل قيمة من أجل
x2: معتز أدخل قيمة من أجل
x3: ذكر أدخل قيمة من أجل
x4: يمني أدخل قيمة من أجل
x5: 23 أدخل قيمة من أجل
x6: 1 أدخل قيمة من أجل
تم تكوين صف واحد
SQL> Commit;
تم التثبيت
```

إضافة سجلات بدون ذكر أسماء الحقول :-

يمكنك عدم كتابة أسماء الحقول مع الجملة Insert Into ولكن شرط أن يتم كتابة كل القيم لكل الحقول وحسب ترتيب الحقول في الجدول .

مثال١ : قم بإضافة السجل التالي إلى جدول التخصصات(Spec) ؟

SP_NO	SP_NAME	SP_AMOUNT
5	مختبرات	50000

```
SQL> insert into spec values(5,'مختبرات' ,50000);
تم تكوين صف واحد
```

مثال١ : قم بإضافة السجل التالي إلى جدول الطلاب(Student) ؟

ST_NO	ST_NAME	ST_SEX	ST_FLAG	ST_AGE	SP_NO
107	علي	ذكر	سعودي	25	3

```
SQL> insert into Student values(107,'علي' ,
'سعودي','ذكر',25,3);
تم تكوين صف واحد
```

ثانياً:- التعديل في بيانات الجدول

عند إدخال البيانات عادة ما يتم الوقوع في الخطأ أو قد تحتاج إلى تعديل بيانات مستحدثة وعملية التعديل قد تكون على حقل معين أو مجموعة حقول وقد يكون التعديل على سجل أو مجموعة سجلات حسب الحاجة للتعديل ، حيث تتم عملية التعديل عن طريق استخدام جملة التعديل(Update) .

الصيغة العامة لتعديل بيانات داخل جدول

```
SQL>Update جدول  Set حقل٢ =قيمة٢ ,حقل١ = قيمة١.....
Where شـــــرط;
```

شرح الصيغة العامة :

- **جدول :- اسم الجدول المطلوب تعديل سجلاته.**
- **(حقل١،حقل٢) :- أسماء الحقول المطلوب تعديل بياناتها .**
- **(قيمة١،قيمة٢) :- القيم الجديدة المراد وضعها بدلاً عن القيم القديمة**
- **شرط :- شرط لاختيار السجلات المطلوب تعديلها وبدون هذا الشرط سيتم التعديل على كل السجلات .**

القواعد التي يجب التقيد بها عن التعديل

- **يجب أن تكون نوع البيانات الجديدة نفس نوع بيانات الحقول المطلوب تعديلها .**
- **عند تعديل قيم التأريخ والنصوص يجب وضع علامتي تنصيص فردية للقيم الجديدة .**
- **مراعاة القيود المدرجة مع كل حقل (القيمة الفارغة – التكرار ...)**
- **التأكد من ظهور الرسالة التي تفيد التعديل ("تم تحديث صف")**
- **كتابة أمر الحفظ Commit بعد التعديل أو قبل الخروج من بيئة SQL**

مثال : قم بتعديل بيانات الطالب رقم 102 بحيث يكون رقم التخصص رقم 1 (حاسوب) والعمر 23 ؟

```
SQL> update student set sp_no=1,st_age=23
 2       where st_no=102;
```

تم تحديث صف واحد

ماذا سوف يحدث إذا كتبنا الأمر بالشكل التالي :

```
SQL> update student set sp_no=2;
```

تم تحديث ٧ صفوف

سنلاحظ أنه تم تعديل أرقام التخصصات لكل الطلاب لذا يجب أن تكون حذر أثناء كتابة الشرط .

مثال :- قم بإضافة حقل جديد لجدول الطلاب باسم رقم الهاتف (st_mobile)

```
SQL> alter table student add(st_mobile number(9));
```

تم تبديل جدول

<u>قم بإضافة بيانات إلى الحقل الجديد</u>

```
SQL> update student set st_mobile=&f1 where st_no=&f2;
f1: 777777أدخل قيمة من أجل
f2: 101أدخل قيمة من أجل
تم تحديث صف واحد
SQL> r
 *١  update student set st_mobile=&f1 where st_no=&f2
f1: 888888أدخل قيمة من أجل
f2: 102أدخل قيمة من أجل
تم تحديث صف واحد
SQL> r
 *١  update student set st_mobile=&f1 where st_no=&f2
f1: 999999أدخل قيمة من أجل
f2: 103أدخل قيمة من أجل
تم تحديث صف واحد
SQL> r
 *١  update student set st_mobile=&f1 where st_no=&f2
f1: 666666أدخل قيمة من أجل
f2: 104أدخل قيمة من أجل
تم تحديث صف واحد
SQL> r
 *١  update student set st_mobile=&f1 where st_no=&f2
```

```
f1: 555555أدخل قيمة من أجل
f2: 105أدخل قيمة من أجل
تم تحديث صف واحد
SQL> r
  ١* update student set st_mobile=&f1 where st_no=&f2
f1: 44444أدخل قيمة من أجل
f2: 106أدخل قيمة من أجل
تم تحديث صف واحد
```

ثالثاً:- حذف سجل أو مجموعة سجلات من جدول

نحتاج عادة إلى حذف سجل أو مجموعة سجلات من أي جدول لأي سبب كان حيث يتم ذلك عن طريق استخدام جملة الحذف Delete From

الصيغة العامة للجملة Delete From

```
SQL>Delete From جدول
Where شرط ;
```

شرح الصيغة العامة :-

جدول :- اسم الجدول المراد الحذف منه .

شرط :- كتابة شرط لتحديد السجلات أو السجل الذي سوف يحذف.

القواعد التي يجب التقيد بها عند الحذف

- الحذر عند كتابة الشرط Where في جملة الحذف لتحديد السجلات التي سوف تحذف فإذا لم نقم بكتابة الشرط سيتم حذف جميع السجلات .

- في الجداول ذات العلاقة يجب أولاً من حذف السجلات من الجداول الفرعية المرتبطة ومن ثم حذف السجلات في الجدول الرئيسي .

- التأكد من ظهور الرسالة التي تفيد الحذف ("تم حذف صف واحد") .

- كتابة أمر الحفظ COMMIT بعد الحذف أو قبل الخروج من بيئة SQL .

مثال :- قم بحذف سجل الطالب الذي يحمل الرقم (107) .

```
SQL> Delete from Student
  2  where st_no= 106;
تم حذف صف واحد
```

مثال :- حذف جميع الطلاب من جدول (Student) .

```
SQL> Delete from Student;
```

لا تقم بتنفيذ المثال السابق للحفاظ على بيانات الجدول وإذا كنت قد فعلت قم بإستخدام الأمر لإسترجاع البيانات ROLLBACK .

حذف سجلات من جداول مترابطة

مثال :- قم بحذف التخصص حاسوب من جدول التخصصات (Spec) .

```
SQL> Delete from spec
  2  where sp_name='حاسوب';
ERROR AT LINE1:
```

نلاحظ أننا لن نستطيع حذف التخصص حاسوب والدليل على ذلك ظهور هذه الرسالة("تم انتهاك قيد التكامل")

ومعنى الرسالة أنه تم انتهاك قيد أو شرط ربط جدول التخصصات بجدول الطلاب ومعنى ذلك أنه لا يمكن حذف التخصص بسبب وجود طلاب مسجلين في هذا التخصص لجدول(Student) وذلك لوجود ربط بين الجدولين عن طريق الحقل(SP_NO) فهو بالنسبة لجدول التخصصات يعتبر مفتاح أساسي(Primary Key) وبالنسبة لجدول الطلاب يعتبر مفتاح أجنبي "ربط" (Foreign Key).

في هذه الحالة يجب حذف الطلاب الذين تخصصاتهم حاسوب أولاً ومن ثم حذف التخصص .

لذلك أولاً سنقوم بحذف جميع الطلاب الذين أرقام تخصصاتهم 1 (حاسوب) .

```
SQL> Delete from Student
    2  where sp_no= 1;
تم حذف ٣صف
```

بعد أن تم حذف جميع الطلاب الذين أرقام تخصصاتهم ١(حاسوب) الآن نستطيع حذف التخصص حاسوب من جدول التخصصات

```
SQL> Delete from spec
  2  where sp_name='حاسوب';
تم حذف صف واحد
```

مثال : قم بحذف التخصص مختبرات من جدول التخصصات .

```
SQL> Delete from spec
  2  where sp_name='مختبرات';
تم حذف صف واحد
```

نلاحظ ظهور رسالة تأكيد الحذف مباشرة (" تم حذف صف واحد ") والسبب في ذلك أنه لا يوجد أي طالب مسجل في هذا التخصص .

هام :-

لاتنسى استخدام الأمر COMMIT بعد الجملة DELETE أو قبل الخروج من بيئة SQL ليتم حذف السجلات .

أنشطة للتطبيق

- إدخال سجلات لجدولي بنك BANK وموظف EMPLOYEE مع استخدام الجملتين Commit و Rollback .

```
SQL> insert into bank values(1,'yemeni Bank',3222111,'Sana a
Street');
```

تم تكوين صف واحد

```
SQL> Commit;
```

تم التثبيت

```
SQL> insert into employee (emp_no , emp_name , emp_mobile
,emp_sal,
        bank_id_fk) values(100,'waleed ahmed
',734588935,50000,1);
```

تم تكوين صف واحد

```
SQL> Commit;
```

تم التثبيت

```
SQL> insert into employee (emp_no , emp_name , emp_mobile
,emp_sal,
        bank_id_fk) values(1001,'Saleh
Omer',734153011,30000,1);
```

تم تكوين صف واحد

```
SQL> Rollback;
```

تم الغاء تعديلات

```
SQL> insert into employee (emp_no , emp_name , emp_mobile
,emp_sal,
        bank_id_fk) values(1002,'Mohmmed
Ali',733154014,55000,1);
```

تم تكوين صف واحد

- قم بزيادة رواتب الموظفين بنسبة ١٠ % مع استخدام الجملتين Commit و Savepoint

```
SQL> update  employee Set emp_sal =emp_sal+emp_sal*.1 ;
        تم تحديث ٢ صف
 SQL> Commit;
تم التثبيت
SQL> Savepoint test;
تم تكوين نقطة حفظ
```

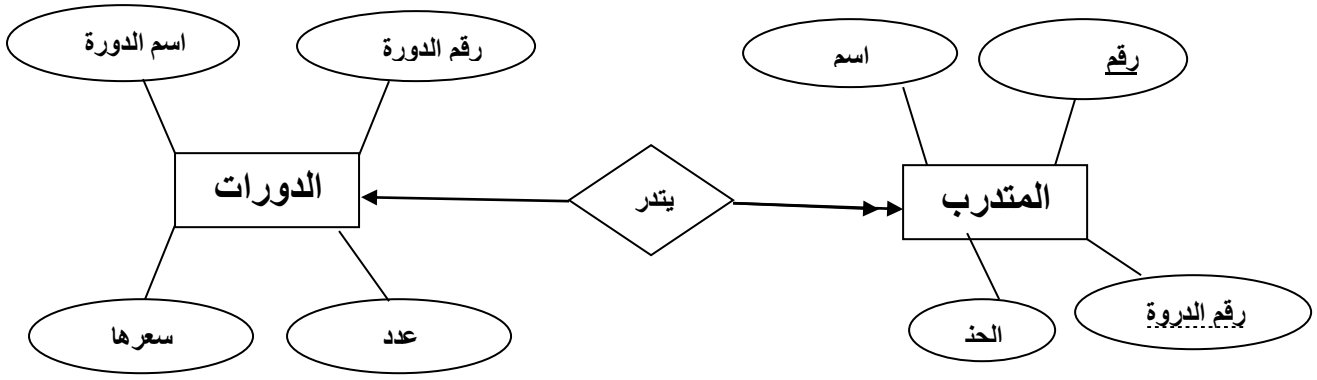
- قم بزيادة راتب الموظف رقم 100 للضعف

```
SQL> update  employee Set emp_sal =emp_sal*2
        Where emp_no=100;
        تم تحديث  صف واحد
```

- قم بحذف سجل موظف رقم 1002 ومن ثم التراجع إلى نقطة الحفظ السابقة

```
SQL> Delete employee  Where emp_no=1002;
         تم حذف  صف واحد
 SQL> Rollback To test;
تم الغاء تعديلات
```

تمرين١ :-



- إنشى جدول الدورات(course) يحتوي على الحقول التالية:

رقم الدورة(c_no) مفتاح أساسي – اسم الدورة(c_name) – السعر(price)

عدد الساعات(h_number).

- إنشى جدول المتدربين() يحتوي على البيانات التالية :

رقم المتدرب(t_no) مفتاح أساسي – اسم المتدرب(t_name) – الجنس(sex) - رقم الدورة(c_no) كمفتاح ربط أجنبي ؟

- قم بإضافة البيانات التالية لجدول الدورات(course):

c_no	c_name	price	h_number
1	C++	5000	12
2	SQL	9000	15
3	JAVA	15000	10
4	VB.NET	15000	30
5	PHP-html	12000	15

- قم بإضافة البيانات التالية لجدول المتدربين (trained)

t_no	t_name	sex	c_no
101	هلال	ذكر	1
102	عبدالرحمن	ذكر	1
103	علي يوسف	ذكر	2
104	فوزية	انثى	2
105	يونس	ذكر	3
106	طارق	ذكر	3
107	عبداللطيف	ذكر	4
108	فاطمة	انثى	4
109	عبدالرحيم	ذكر	5
110	زايد	ذكر	5

- قم بإضافة السجل التالي لجدول الدورات(course):

c_no	c_name	price	h_number
6	prolog	10000	20

- قم بإضافة السجل التالية لجدول المتدربين (trained):

t_no	t_name	sex	c_no
111	حنان	انثى	5

- قم بعرض جميع بيانات جدول المتدربين(trained):
- قم بتعديل بيانات المتدربين رقم 102 بحيث يكون رقم الدورة رقم 2 (SQL) ؟
- قم بإضافة حقل جديد لجدول المتدربين باسم رقم الهاتف (mobile)
- قم بإضافة البيانات التالية إلى الحقل الجديد(mobile)

t_no	mobile
101	7333333
102	7444444
103	75555555
104	76666666

105	7777777
106	78888888
107	79999999
108	71111111
109	72222222
110	70000000
111	71010101

- قم بحذف سجل المتدرب الذي يحمل الرقم (111) .
- قم بحذف الدورة (prolog) من جدول الدورات (course) .
- قم بحذف الدورة(PHP-html) من جدول الدورات(course) .

تمرين٢ :

❖ قم بإضافة سجلات للجداول التي قمت بإنشائها في تمرين المحاضرة (٢) كما يلي :

- جدول المدرس

رقم المدرس	اسم المدرس	رقم الموبايل	العنوان
1	وضاح وهيب حسن	771846901	شارع صنعاء
2	نائف عبدالوهاب الدعيس	770672876	
3	محمد حمود جبريل		شارع غليل

- جدول البطاقة الشخصية

رقم البطاقة	جهة الحصول عليها	تأريخ الحصول عليها	تأريخ انتهاؤها	رقم المدرس
05011122277	الحديدة	1/1/2006	31/12/2016	1
05022211188	الحديدة	15/11/2005	14/11/2015	3
05033377799	إب	19/3/2006	18/3/2016	2

- جدول المادة

رمز المادة	اسم المادة	رقم المدرس
31001	نظم قواعد البيانات	1
43001	شبكات عصبية	3
43002	نظم تشغيل	3
32001	تحكم آلي	2

- جدول طالب

رقم الطالب	اسم الطالب	العمر	العنوان
20061500	محمد احمد محمد	25	شارع صنعاء
20061501	فاطمة محمد سعيد	24	شارع الميناء
20051500	صالح قاسم عبدالله	22	شارع المطار
20052500	خالد عبدالرحمن حيدر	23	شارع الميناء

- جدول درجات الطالب

رقم الطالب	رقم المادة	العام الجامعي	الدرجة
20061500	43001	2009/2010	25
20061501	43002	2009/2010	29
20061500	43002	2009/2010	28
20051500	31001	2009/2010	30
20052500	32001	2009/2010	27

- ❖ قم بزيادة درجات الطلاب بمقدار درجة واحدة بشرط ألا تصل إلى درجة أكبر من 30
- ❖ قم بتعديل عنوان الطالب رقم 20061501الى ش/ الحمدي
- ❖ قم بحذف سجل درجة الطالب رقم 20052500 ثم تراجع عن العملية باستخدام :
  - RollBack
  - Savepoint
- ❖ هل بإمكانك حذف سجل الطالب رقم 20052500 من جدول الطالب ؟ مع ذكر السبب ؟

جملة الإستعلام SELECT

تعتبر جملة الإستعلام SELECT من أهم جمل SQL على الإطلاق والذي بواسطتها يتم استرجاع البيانات(اختيار البيانات لغرض العرض) من الجداول أو من مجموعة جداول مترابطة ويعتبر هذا الموضوع من أهم المواضيع التي تم شرحها حيث لا يمكن الحصول على مخرجات أي نظام إلا من خلال جملة الإستعلام SELECT ومن خلال هذا الجزء سيتم شرح هذه الجملة بالتدرج حتى نصل إلى شرح كل ما يتعلق بالإستعلام .

الصيغة العامة لجملة SELECT :-

```
SELECT [* , COLUMN,.....]
FROM TABLE
WHERE CONDITION
```

شرح الصيغة العامة

SELECT :- تستخدم في بداية الأمر لإسترجاع البيانات من جدول .

* :- هذا الأمر يستخدم في استرجاع كل الحقول من الجدول .

COLUMN :- أسماء الحقول المراد استرجاع البيانات منها ونفصل بين حقل وحقل آخر بعلامة (،) .

FROM :- تستخدم للإعلان عن اسم جدول .

TABLE :- اسم الجدول أو الجداول التي سيتم استرجاع البيانات منها وإذا كان هناك أكثر من جدول نفصل بينهما بعلامة(،)

WHERE :- تستخدم للإعلان عن الشرط أو الشروط (اختياري) .

CONDITION :- الشرط أو الشروط لحصر البيانات المراد استرجاعها .

اليك الجداول التالية :

department			
DEPT_NO رقم القسم	DEPT_NAME اسم القسم	DEPT_AMOUNT الرسوم	DEPT_LOC الموقع
1	حاسوب	50000	كلية علوم وهندسة الحاسوب
2	محاسبة	45000	كلية التجارة والاقتصاد
3	رياضيات	40000	كلية التربية

student				
ST_NO رقم الطالب	ST_NAME اسم الطالب	ST_ADDRESS العنوان	ST_AGE العمر	DEPT_NO رقم القسم
101	محمد شيبه	باجل	25	1
102	محمد صالح	الحديدة	20	3
103	أحمد محمد	صنعاء	22	3
104	خالد عبده	الحديدة	30	2
105	ايهاب البكاري	باجل	23	1
106	صادق محمد	القطيع	23	2
107	صدام علي	القطيع	23	3
108	شمسان عامر	برع	25	1
109	وليد أحمد	الحديدة	26	2
110	عبدالكريم	باجل	22	1

سنقوم بإنشاء هذه الجداول داخل قاعدة البيانات أوراكل :

أولاً :- سنقوم بإنشاء جدول الاقسام (department)

```
SQL> create table department(dept_no number(9),dept_name varchar2(20),dept_amount number(20),  dept_
loc varchar2(30),
  2   constraint pk_dept_no primary key(dept_no));

تم تكوين جدول.
```

ثانياً:- سنقوم بإنشاء جدول الطلاب (student)

```
SQL> create table student(st_no number(9),st_name varchar2(30),st_address varchar2(30),st_age number

  2  (9),dept_no number(9),
  3     constraint fk_dept_no foreign key(dept_no) references department(dept_no));

تم تكوين جدول.
```

سنقوم بإدخال البيانات أولا الى جدول الاقسام (department)

```
SQL> insert into department values(&x1,'&x2',&x3,'&x4');
أدخل قيمة من أجل x1: 1
أدخل قيمة من أجل x2: حاسوب
أدخل قيمة من أجل x3: 50000
أدخل قيمة من أجل x4: كلية علوم وهندسة الحاسوب
١    قديم: insert into department values(&x1,'&x2',&x3,'&x4')
١    جديد: insert into department values(1,'حاسوب',٥٠٠٠٠,'كلية علوم وهندسة الحاسوب')

تم تكوين صف واحد
```

ثانياً سنقوم بإدخال البيانات الى جدول الطلاب (student)

```
SQL> insert into student values(&x1,'&x2','&x3',&x4,&x5);
أدخل قيمة من أجل x1: 101
أدخل قيمة من أجل x2: محمد شيبه
أدخل قيمة من أجل x3: باجل
أدخل قيمة من أجل x4: 25
أدخل قيمة من أجل x5: 1
١    قديم: insert into student values(&x1,'&x2','&x3',&x4,&x5)
١    جديد: insert into student values(101,'محمد شيبه','باجل',٢٥,١)

تم تكوين صف واحد
```

قم بعرض جميع البيانات من جدول الأقسام (department) .

```
SQL> select * from department;

   DEPT_NO DEPT_NAME           DEPT_AMOUNT DEPT_LOC
---------- -------------------- ----------- ----------------------------
         1 حاسوب                      50000 كلية علوم وهندسة الحاسوب
         2 محاسبة                    45000 كلية التجارة والاقتصاد
         3 رياضيات                   40000 كلية التربية
```

قم بعرض جميع البيانات من جدول الطلاب (student)

SQL> Select * from student;

ST_NO	ST_NAME	ST_ADDRESS	ST_AGE	DEPT_NO
101	محمد شيبه	باجل	25	1
102	محمد صالح	الحديدة	20	3
103	أحمد محمد	صنعاء	22	3
104	خالد عبده	الحديدة	30	2
105	ايهاب البكاري	باجل	23	1
106	صادق محمد	القطيع	23	2
107	صدام علي	القطيع	23	3

1	25	برع	شمسان عامر	108
2	26	الحديدة	وليد أحمد	109
1	22	باجل	عبدالكريم	110

عرض بيانات الحقول التالية st_no,st_name,st_age لكل الطلاب .

```
SQL> select st_no,st_name,st_age from student;

     ST_NO ST_NAME                          ST_AGE
---------- ------------------------------ ----------
       101 محمد شيبه                              25
       102 محمد صالح                              20
       103 أحمد محمد                              22
       104 خالد عبده                              30
       105 ايهاب البكاري                          23
       106 صادق محمد                              23
       107 صدام علي                               27
       108 شمسان عامر                             25
       109 وليد أحمد                              26
       110 عبد الكريم                             22
```

عرض بيانات الحقول dept_no,dept_name مع عرض بيانات الحقل dept_amount مضافا اليه ١٠٠٠٠ لكل قسم .

```
SQL> select dept_no,dept_name,dept_amount+10000 as Total
  2     from department;

   DEPT_NO DEPT_NAME                      TOTAL
---------- ------------------- ----------
         1 حاسوب                       60000
         2 محاسبة                      55000
         3 رياضيات                     50000
```

عرض بيانات الحقل st_name مدموجا مع الحقل st_no لكل الطلاب وتخصيص noname ليعرض كبديل عن عنوان العمود الناتج

```
SQL> select st_no||st_name as "noname" from student;

noname
--------------------------------------------------
101محمد شيبه
102محمد صالح
103أحمد محمد
104خالد عبده
105ايهاب البكاري
106صادق محمد
107صدام علي
108شمسان عامر
109وليد أحمد
110عبد الكريم
```

ملاحظة:

لعرض محتويات حقلين كحقل واحد

إعد

نستخدم أداة الربط (| |)

عرض بيانات الحقل dept_no مدموجاً مع النص ' is a ' والحقل dept_name لكل الاقسام ، وتخصيص الاسم depart ليعرض كبديل عن عنوان "

```
SQL> select dept_no || 'is a' || dept_name as "depart"
  2  from department;

depart
---------------
1is aحاسوب
2is aمحاسبة
3is aرياضيات
```

عرض أرقام الأقسام deptno المنتمي إليها كل الموظفين بدون تكرار باستخدام العبارة nct

```
SQL> select distinct dept_no from student;

   DEPT_NO
----------
         1
         2
         3
```

جملة الشرط Where

تكتب هذه الجملة بعد جملة FROM وتستخدم لحصر البيانات حسب شرط أو الشروط الملحقة مع الجملة ويتكون الشرط من جزيين يفصل بينهم عامل مقارنة مثل (= - <> => -=<) وعند تحقق الشرط تعرض السجلات المطابقة للشرط وعند عدم تحقق الشرط تظهر رسالة (NO ROW SELECTED) ومعناها لا يوجد أي سجل مطابق للشرط ، ولا يخلوا الشرط من أسماء حقول ويمكن أن يحتوي على قيم ثابتة نصية أو رقمية وكذلك تعبيرات حسابية ، كم يمكن أن يحتوي الشرط على جملة SELECT أخرى تسمى الجملة الفرعية .

<u>ويجب مراعاة التالي عند كتابة الشرط :-</u>

١- عند استخدام قيمة نصية ثابتة أو قيم من نوع تأريخ يجب وضعها بين علامة تنصيص فردية مثل("ذكر") .
٢- في حالة استخدام قيمة نصية باللغة الإنجليزية يجب مراعاة الأحرف الصغيرة أو الكبيرة كذلك المسافات .
٣- في حالة استخدام قيمة من نوع تأريخ يجب مراعاة تنسيق التأريخ عند ادخاله إلى الجدول حيث أن التنسيق الإفتراضي للتأريخ هو (DD- MON- YY) سنة – شهر- يوم .

معاملات المقارنة المستخدمة مع جملة Where

المعامل	المعنى
=	يساوي
<> أو =!	لا يساوي
>	أكبر من
<	أقل من
>=	أكبر من أو يساوي
<=	أقل من أو يساوي
معاملات أخرى	
قيمة AND قيمة Between	حصر البيانات بين قيمتين
قيمة AND قيمة Not Between	حصر البيانات التي ليست بين القيمتين
IN(قيمة١ ، قيمة ٢، قيمة٣)	حصر البيانات ضمن مجموعة قيم
NOT IN(قيمة١ ، قيمة ٢، قيمة٣)	حصر البيانات التي ليست بين القيمتين
LIKE(-_%)	حصر البيانات حسب مطابقة النص أو الحرف
Not LIKE(-_%)	حصر البيانات التي لا تطابق النص أو الحرف
IS NULL	حصر البيانات ذات القيمة الخالية
IS NOT NULL	حصر البيانات التي ليست خالية

ملاحظة :- NOT تستخدم للنفي .

عرض بيانات الحقول st_no,st_name,st_age للطلاب الذين ينتمون للقسم رقم ١

```
SQL> select st_no,st_name,st_age from student
  2  where dept_no=1;

     ST_NO ST_NAME                          ST_AGE
---------- ------------------------------ ----------
       101 محمد شيبه                              25
       105 ايهاب البكاري                          23
       108 شمسان عامر                             25
       110 عبد الكريم                             22
```

عرض بيانات الحقول st_name,st_age للطلاب الذين أعمارهم أقل من أو تساوي ٢٥

```
SQL> select st_name,st_age from student
  2  where st_age>=25;

ST_NAME                          ST_AGE
------------------------------ ----------
محمد شيبه                            25
خالد عبده                            30
صدام علي                             27
شمسان عامر                           25
وليد أحمد                            26
```

عرض بيانات الحقول st_name,st_age للطلاب الذين أعمارهم ما بين ٢٠ و ٢٥

```
SQL> select st_name,st_age from student
  2  where st_age between 20 and 25;

ST_NAME                          ST_AGE
------------------------------ ----------
محمد شيبه                            25
محمد صالح                            20
أحمد محمد                            22
ايهاب البكاري                        23
صادق محمد                            23
شمسان عامر                           25
عبد الكريم                           22
```

عرض بيانات الحقول st_name,st_address للطلاب الذين يسكنون في باجل

```
SQL> select st_name,st_address from student
  2  where st_address in('باجل');

ST_NAME                        ST_ADDRESS
------------------------------ ----------
محمد شيبه                      باجل
ايهاب البكاري                  باجل
عبد الكريم                     باجل
```

عرض بيانات st_no,st_name لطلاب المحاسبة والحاسوب بشرط أن لا يقل عمرعن 25

```
SQL> select st_no,st_name from student
  2  where (dept_no=1 or dept_no=2) and st_age>=25;

     ST_NO ST_NAME
---------- ------------------------------
       101 محمد شيبه
       104 خالد عبدي
       108 شمسان عامر
       109 وليد أحمد
```

عرض بيانات الحقول st_name,st_address للطلاب الذين يسكنون في باجل والقطيع

```
SQL> select st_name from student
  2  where st_address in('القطيع','باجل');

ST_NAME
------------------------------
محمد شيبه
ايهاب البكاري
صادق محمد
صدام علي
عبد الكريم
```

المعامل LIKE() :-

يستخدم هذا المعامل للبحث عن نص معين داخل متغير أو حقل نصي وفي أي جزء من النص .

والرموز المستخدمة مع المعامل هي :

(%) يستخدم للتعبير عن مجموعة حروف مجهولة وهو نفس استخدام الرمز (*) عند البحث في نظام التشغيل .

(_) يستخدم لتعبير عن حرف واحد مجهول وهو نفس استخدام الرمز (؟) عند البحث في نظام التشغيل .

عرض أسماء الطلاب الذين الحرف الاول من أسمائهم هو الحرف (م) .

```
SQL> select st_name from student
  2  where st_name like 'م%';

ST_NAME
-------------
محمد شيبه
محمد صالح
```

عرض أسماء الطلاب الذين الحرف الثاني من أسمائهم هو الحرف (ح) .

```
SQL> select st_name from student
  2  where st_name like '_ح%';

ST_NAME
----------
محمد شيبه
محمد صالح
أحمد محمد
```

عرض أسماء الطلاب الذين يتكون حروف أسمائهم من 5 أحرف وينتهي بحرف (د).

```
SQL> select st_name from student
  2  where st_name like '____د%';

ST_NAME
------------------------------
محمد شيبه
محمد صالح
أحمد محمد
خالد عبده
وليد أحمد
```

عرض كل اسم يحتوي على حرف (د) .

```
SQL>  select st_name from student
  2   where st_name like '%د%';

ST_NAME
------------------------------
محمد شيبه
محمد صالح
أحمد محمد
خالد عبده
صادق محمد
صدام علي
وليد أحمد
عبد الكريم
```

عرض أسماء الطلاب الذين لا يبدأ الاسم الاول من اسمائهم بالحرف (م) .

```
SQL> select st_name from student
  2  where st_name not like 'م%';

ST_NAME
------------------------------
أحمد محمد
خالد عبده
ايهاب البكاري
صادق محمد
صدام علي
شمسان عامر
وليد أحمد
عبد الكريم
```

عرض أسماء الطلاب الذين لا عمر لهم .

```
SQL> select st_name from student
  2  where st_age is null;

لم يتم تحديد صفوف
```

وللحصول على عكس النتيجة نستخدم معامل النفي (NOT)

```
SQL> select st_name from student
  2  where st_age is not null;
```

عرض بيانات الحقول st_no,st_name,st_age للطلاب الذين اعمارهم اكبر أو يساوي 25 والحرف الاول من أسمائهم هو (م) .

```
SQL> select st_no,st_name,st_age from student
  2  where st_age>=25 and st_name like 'م%';

     ST_NO ST_NAME                          ST_AGE
---------- ------------------------------ ----------
       101 محمد شيبه                              25
```

عرض بيانات الحقول st_name,st_address للطلاب الذين لا ينتمون للقسم 2 ولا للقسم 3

```
SQL> select st_name,st_address from student
  2  where dept_no not in(2,3);

ST_NAME                        ST_ADDRESS
------------------------------ ----------
محمد شيبه                      باجل
ايهاب البكاري                  باجل
شمسان عامر                     برع
عبد الكريم                     باجل
```

جملة الترتيب ORDER BY

تعتبر عملية الترتيب أو فرز البيانات في قواعد البيانات من العمليات الهامة والمطلوبة دائماً وعادة ما يكون إخراج البيانات بشكل مفهرس بحسب أي حقل وذلك ليسهل عملية الوصول إلى أي معلومات بشكل سريع

بالنظر إلى ناتج جميع الامثلة السابقة سوف تجد أن البيانات تم عرضها حسب ترتيب إدخالها وعن طريق الجملة ORDER BY تستطيع ترتيب الصفوف الناتجة ترتيباً تصاعدياً أو تنازلياً وتكتب في نهاية جملة SELECT .

ويجب مراعاة التالي عند كتابة جملة الترتيب :-

- **يكتب اسم الحقل المراد الترتيب حسبه بعد عبارة ORDER BY مباشرة .**
- **يمكن استخدام الاسم المستعار للحقل بدل اسم الحقل .**
- **يمكن أن يكون الترتيب على أكثر من حقل بحيث تفصل بين أسماء الحقول بفاصلة.**
- **لترتيب البيانات تصاعدياً استخدم العبارة ASC بعد اسم الحقل وهي اختصار لـ ASCENDING**
- **لترتيب البيانات تنازلياً استخدم العبارة DESC بعد اسم الحقل وهي اختصار لـ DESCENDING**

عرض كل حقول ملف القسم شرط أن تكون البيانات مرتبة ترتيباً ابجدياً تصاعديا حسب [illegible]

```
SQL> select * from department
  2  order by dept_name ASC;

   DEPT_NO DEPT_NAME            DEPT_AMOUNT DEPT_LOC
---------- -------------------- ----------- --------------------------
         1 حاسوب                      50000 كلية علوم وهندسة الحاسوب
         3 رياضيات                    40000 كلية التربية
         2 محاسبة                     45000 كلية التجارة و الاقتصاد
```

عرض طلاب الحاسوب فقط بحيث تكون البيانات مرتبة ترتيباً تصاعدياً بحسب حقل الاسم

```
SQL> select st_name from student
  2  where dept_no=1
  3  order by st_name;

ST_NAME
------------------------------
ايهاب البكاري
شمسان عامر
عبد الكريم
محمد شيبه
```

ملاحظة: الترتيب الافتراضي للجملة ORDER BY هو ترتيب تصاعدي فعند عدم كتابة العبارة ASC أو DESC فإن الترتيب سيكون ترتيب تصاعدياً .

عرض بيانات الحقول dept_name,dept_amount+10000 وتخصيص الاسم sal ليعرض كبديل عن عنوان العمود الناتج ، وترتيب العرض بحسب الاسم البديل sal تنازليا

```
SQL> select dept_name,dept_amount+10000 sal from department
  2   order by sal;

DEPT_NAME                       SAL
-------------------- ----------
رياضيات                      50000
محاسبة                       55000
حاسوب                        60000
```

عرض بيانات الحقول st_name,st_age,dept_no لكل الطلاب مرتبين بحسب رقم القسم تصاعدياً والعمر تنازلياً .

```
SQL> select st_name,st_age,dept_no from student
  2  order by dept_no,st_age desc;

ST_NAME                          ST_AGE    DEPT_NO
------------------------------ ---------- ----------
محمد شيبه                            25          1
شمسان عامر                           25          1
ايهاب البكاري                        23          1
عبد الكريم                           22          1
خالد عبده                            30          2
وليد أحمد                            26          2
صدام علي                             27          3
صادق محمد                            23          3
أحمد محمد                            22          3
محمد صالح                            20          3
```

تمارين المحاضرة الرابعة :-

الجداول التالية موجودة ضمن بيئة قاعدة البيانات الخاصة بأوراكل خاصة بالمستخدم SCOTT وكلمة السر Tiger:-

الجدول الاول : DEPT ويحتوي على بيانات الاقسام

رقم القسم	Depno
اسم القسم	Dname
الموقع	Loc

الجدول الثاني : EMP ويحتوي على بيانات الموظفين

رقم الموظف	Empno
اسم الموظف	Ename
الوظيفة	Job
رقم المدير	Mgr
تأريخ بدء الخدمة	Hiredate
الراتب	Sal
العمولة	Comm
رقم القسم المنتمي إليه الموظف	Deptno

باستخدام جدول الموظفين EMP الخاص بالمستخدم SCOTT :-

١- نفذ استعلام يعرض جميع بيانات الموظفين (السجلات + الحقول)

٢- نفذ استعلام يعرض بيانات الحقول Sal , ename, empno لكل الموظفين

٣- نفذ اسعلام يعرض بيانات الحقول empno,ename مع عرض بيانات الحقل Sal مضاف إليه ٣٠٠ لكل الموظفين .

٤- نفذ استعلام يعرض بيانات الحقل ename مدموجاً مع الحقل job لكل الموظفين وتخصيص الاسم Employees ليعرض كبديل عن عنوان العمود الناتج .

٥- نفذ استعلام يعرض بيانات الحقل ename مدموجاً مع النص ' is a ' والحقل job لكل الموظفين ، وتخصيص الاسم Employee Details ليعرض كبديل عن عنوان العمود الناتج .

٦- نفذ استعلام يعرض أرقام الأقسام deptno المنتمي إليها كل الموظفين بدون تكرار باستخدام العبارة Distinct

٧- نفذ استعلام يعرض بيانات الحقول empno,ename,job,deptno للموظفين الذين ينتمون للقسم رقم 20 .

٨- نفذ استعلام يعرض بيانات الحقول ename,sal للموظفين الذين واتبهم أقل من أو تساوي ٣٠٠٠

٩- نفذ استعلام يعرض بيانات الحقول ename,sal للموظفين الذين رواتبهم ما بين ٢٥٠٠و٣٥٠٠

١٠- نفذ استعلام يعرض بيانات الحقول ename,sal,mgr للموظفين الذين أرقام مدرائهم تساوي ٧٥٦٦ أو ٧٧٨٨ أو ٧٩٠٢

١١- نفذ استعلام يعرض أسماء الموظفين الذين الحرف الأول من أسمائهم هو الحرف S

١٢- نفذ استعلام يعرض أسماء الموظفين الذين الحرف الثاني من أسمائهم هو الحرف O

١٣- نفذ استعلام يعرض أسماء الموظفين الذين لا مدير لهم

١٤- نفذ استعلام يعرض بيانات الحقول empno,ename,job,sal للموظفين الذين

رواتبهم أكبر من أو تساوي ٢٠٠٠ ووظيفتهم تحتوي على السلسلة MAN

١٥- نفذ استعلام يعرض بيانات الحقول ename,job,deptno للموظفين الذين

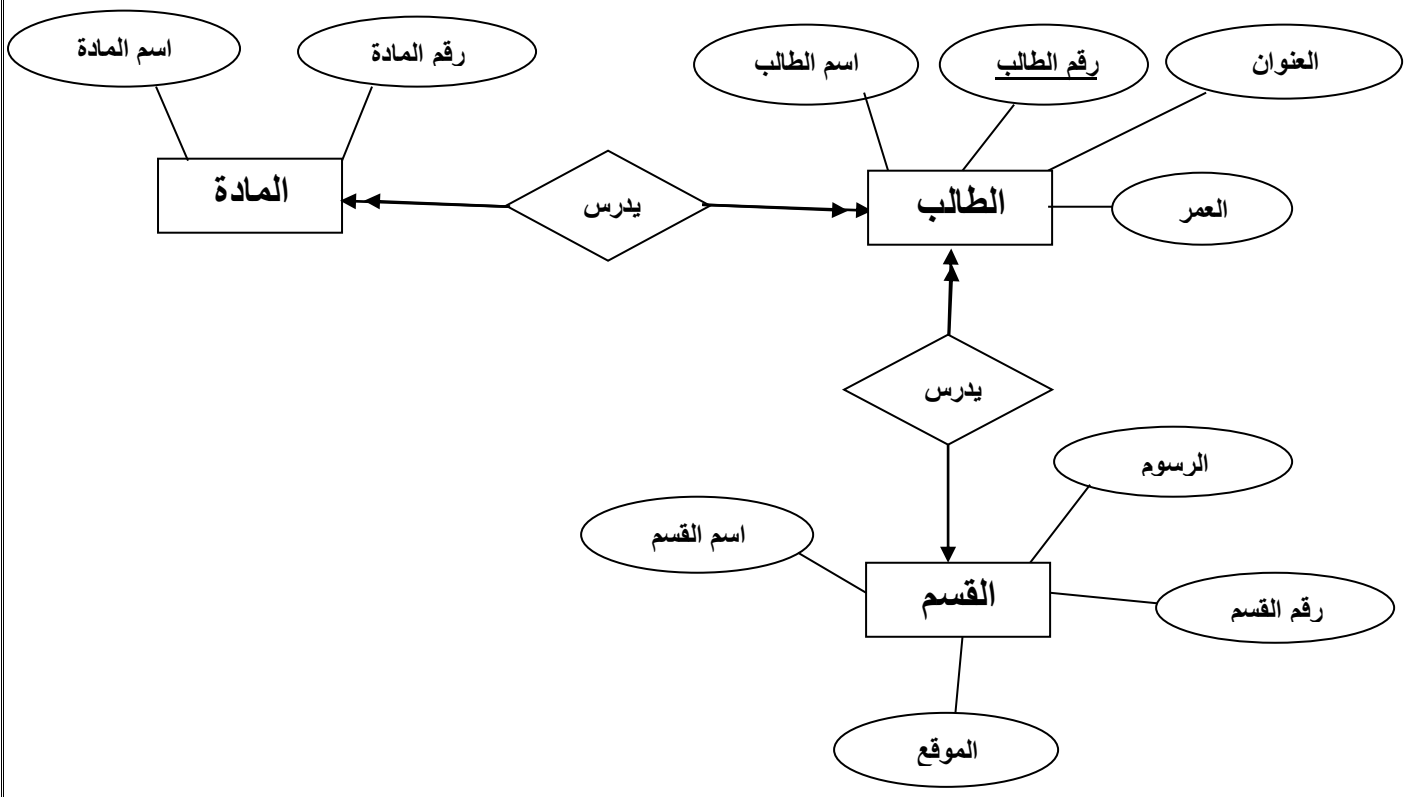
لاينتمون للقسم 10 ولا للقسم 20

١٦- نفذ استعلام يعرض بيانات الحقول ename,job,depno,hiredate مرتبين بحسب تاريخ بدء الخدمة HIREDATE

استرجاع بيانات من أكثر من جدول

تعلمنا في المحاضرة السابقة كيفية استرجاع بيانات من جدول واحد فقط وفي هذه المحاضرة سنتعلم كيفية استرجاع بيانات من أكثر من جدول باستخدام طرق الربط.

ولتنفيذ ذلك اليك النموذج التالي :-



أولاً :- سنقوم بتحويل نموذج الكينونات الى النموذج العلائقي (الجداول)

جدول الاقسام (Department)			
DEPT_NO رقم القسم	DEPT_NAME اسم القسم	DEPT_AMOUNT الرسوم	DEPT_LOC الموقع

جدول الطلاب(Student)				
ST_NO رقم الطالب	ST_NAME اسم الطالب	ST_ADDRESS العنوان	ST_AGE العمر	DEPT_NO رقم القسم

جدول المواد(Subject)	
Sub_NO رقم المادة	Sub_NAME اسم المادة

الجدول الضعيف بين جدول الطلاب والمواد(Stsub)		
ST_NO رقم المادة	Sub_No اسم المادة	Deg الدرجة

سنقوم بإنشاء الجداول أولاً :- جدول الاقسام

```
SQL> create table department(dept_no number(9) primary key,dept_name varchar2(20),dept_amount number
(20),  dept_loc varchar2(30));

تم تكوين جدول.
```

ثانياً:- جدول الطلاب

```
SQL> create table student(st_no number(9) primary key ,st_name varchar2(30),st_address varchar2(30),
st_age number(9),dept_no number(9) references department(dept_no));

تم تكوين جدول.
```

ثالثاً:- جدول المواد

```
SQL> create table subject(sub_no number(9) primary key,
  2  sub_name varchar2(20));

تم تكوين جدول.
```

رابعاً :- الجدول الضعيف

```
SQL> create table stsub(st_no number(9) references student(st_no),
  2  sub_no number(9) references subject(sub_no),
  3  deg number(6));

تم تكوين جدول.
```

سنقوم بادخال البيانات التالية الى الجداول :

department			
DEPT_NO رقم القسم	DEPT_NAME اسم القسم	DEPT_AMOUNT الرسوم	DEPT_LOC الموقع
1	حاسوب	50000	كلية علوم وهندسة الحاسوب
2	محاسبة	45000	كلية التجارة والاقتصاد
3	رياضيات	40000	كلية التربية

```
SQL> insert into department values(&x1,'&x2',&x3,'&x4');
أدخل قيمة من أجل x1: 1
أدخل قيمة من أجل x2: حاسوب
أدخل قيمة من أجل x3: 50000
أدخل قيمة من أجل x4: كلية علوم وهندسة الحاسوب
قديم 1: insert into department values(&x1,'&x2',&x3,'&x4')
جديد 1: insert into department values(1,'حاسوب',50000,'كلية علوم وهندسة الحاسوب')

تم تكوين صف واحد
```

Student				
ST_NO رقم الطالب	ST_NAME اسم الطالب	ST_ADDRESS العنوان	ST_AGE العمر	DEPT_NO رقم القسم
101	محمد شيبه	باجل	25	1
102	محمد صالح	الحديدة	20	3
103	أحمد محمد	صنعاء	22	3
104	خالد عبده	الحديدة	30	2
105	ايهاب البكاري	باجل	23	1
106	صادق محمد	القطيع	23	2

```
SQL> insert into student values(&x1,'&x2','&x3',&x4,&x5);
أدخل قيمة من أجل x1: 101
أدخل قيمة من أجل x2: محمد شيبه
أدخل قيمة من أجل x3: باجل
أدخل قيمة من أجل x4: 25
أدخل قيمة من أجل x5: 1
١    قديم : insert into student values(&x1,'&x2','&x3',&x4,&x5)
١    جديد: insert into student values(101,'محمد شيبه','باجل',٢٥,١)

تم تكوين صف واحد
```

جدول المواد (subject)	
SUB_NO رقم المادة	SUB_NAME اسم المادة
1	SQL
2	C++
3	JAVA
4	PHP

```
SQL> insert into subject(sub_no,sub_name)
  2  values(&x1,'&x2');
أدخل قيمة من أجل x1: 1
أدخل قيمة من أجل x2: sql
٢    قديم : values(&x1,'&x2')
٢    جديد: values(1,'sql')

تم تكوين صف واحد
```

جدول ربط بين جدول الطلاب والمواد(stsub)		
ST_NO رقم الطالب	SUB_NAME رقم المادة	Deg الدرجة
101	1	100
101	2	50
102	1	60
102	3	70
103	4	100
104	3	50
104	4	85
105	2	90
106	1	40

```
SQL> insert into stsub(st_no,sub_no,deg)
  2  values(&x1,&x2,&x3);
أدخل قيمة من أجل x1: 101
أدخل قيمة من أجل x2: 1
أدخل قيمة من أجل x3: 100
٢    قديم : values(&x1,&x2,&x3)
٢    جديد: values(101,1,100)

تم تكوين صف واحد
```

- **طرق استرجاع بيانات من أكثر من جدول**

١- طريقة الربط باستخدام الضرب الكارتيزي

ويتم استخدامها عند استرجاع بيانات من أكثر من جدول بدون استخدام شرط للربط والشكل المجاور يوضح ذلك حيث يتم ربط كل سجل في الجدول الاول بكل سجلات الجدول التالي وهكذا .

6 صفوف	
ST_NO	ST_NAME
101	محمد شيبه
102	محمد صالح
103	أحمد محمد
104	خالد عبده
105	ايهاب البكاري
106	صادق محمد

3 صفوف	
DEPT_NO	DEPT_NAME
1	حاسوب
2	محاسبة
3	رياضيات

الجدول الناتج من عملية الاستعلام ٣×٦=١٨ صفوف

عرض بيانات الحقول st_name,dept_name من الجدولين الطلاب والاقسام.

```
SQL> select dept_name,st_name from department,student;

DEPT_NAME            ST_NAME
-------------------- ------------------------------
حاسوب                محمد شيبه
حاسوب                محمد صالح
حاسوب                أحمد محمد
حاسوب                خالد عبده
حاسوب                ايهاب البكاري
حاسوب                صادق محمد
محاسبة               محمد شيبه
محاسبة               محمد صالح
محاسبة               أحمد محمد
محاسبة               خالد عبده
محاسبة               ايهاب البكاري

DEPT_NAME            ST_NAME
-------------------- ------------------------------
محاسبة               صادق محمد
رياضيات              محمد شيبه
رياضيات              محمد صالح
رياضيات              أحمد محمد
رياضيات              خالد عبده
رياضيات              ايهاب البكاري
رياضيات              صادق محمد

تم اختيار ١٨ صف
```

عرض بيانات الحقول st_name,sub_name من الجدولين الطلاب والمواد .

```
SQL> select st_name,sub_name from student,subject;

ST_NAME                        SUB_NAME
------------------------------ --------------------
محمد شيبه                      sql
محمد شيبه                      c++
محمد شيبه                      java
محمد شيبه                      php
محمد صالح                      sql
محمد صالح                      c++
محمد صالح                      java
محمد صالح                      php
أحمد محمد                      sql
أحمد محمد                      c++
أحمد محمد                      java

ST_NAME                        SUB_NAME
------------------------------ --------------------
أحمد محمد                      php
خالد عبده                      sql
خالد عبده                      c++
خالد عبده                      java
خالد عبده                      php
ايهاب البكاري                  sql
ايهاب البكاري                  c++
ايهاب البكاري                  java
ايهاب البكاري                  php
صادق محمد                      sql
صادق محمد                      c++

ST_NAME                        SUB_NAME
------------------------------ --------------------
صادق محمد                      java
صادق محمد                      php

تم اختيار ٢٤ صف
```

٢- طريقة الربط باستخدام الربط المتساوي = الربط الداخلي

ويتم استخدامها عند استرجاع بيانات من أكثر من جدول باستخدام الربط بين الحقول الموجودة في كلا الجدولين وهي المفاتيح الاساسية والمفاتيح الاجنبية .

6 صفوف		
ST_NO	ST_NAME	DEPT_NO
101	محمد شيبه	1
102	محمد صالح	3
103	أحمد محمد	3
104	خالد عبده	2
105	ايهاب البكاري	1
106	صادق محمد	2

Foreign key

3 صفوف	
DEPT_NO	DEPT_NAME
1	حاسوب
2	محاسبة
3	رياضيات

Primary key

عرض اسماء جميع الطلاب مع تخصصاتهم ؟

```
SQL> select st_name,dept_name from student,department
  2  where student.dept_no=department.dept_no;

ST_NAME                        DEPT_NAME
------------------------------ --------------------
محمد شيبه                      حاسوب
محمد صالح                      رياضيات
أحمد محمد                      رياضيات
خالد عبده                      محاسبة
ايهاب البكاري                  حاسوب
صادق محمد                      محاسبة

تم اختيار ٦ صف
```

عرض أسماء الطلاب وأسماء الاقسام الذين أرقام تخصصاتهم تساوي ١

```
SQL> select st_name,dept_name from student,department
  2  where student.dept_no=1 and department.dept_no=1;

ST_NAME                        DEPT_NAME
------------------------------ --------------------
محمد شيبه                      حاسوب
ايهاب البكاري                  حاسوب
```

عرض اسم الطالب واسم مادة

```
SQL> select st_name,sub_name,deg from student,subject,stsub
  2  where student.st_no=stsub.st_no and subject.sub_no=stsub.sub_no;
```

إن من أهم المميزات التي تمتاز بها أوراكل عن غيرها من اللغات المتخصصة في برمجة قواعد البيانات هي حماية وسرية البيانات حيث توفر صلاحيات معينة لكل جزء من أجزاء قاعدة البيانات مثل صلاحية إنشاء الجداول وتعديلها ،ومنح صلاحيات إنشاء المستخدمين وامتيازات الإتصال بالنظام وصلاحيات التعامل مع البيانات من اضافة وتعديل وحذف .....الخ ويتم منح هذه الصلاحيات والأمتيازات حسب الحاجة والأهمية ، ومن خلال هذا الجزء من لغة SQL سنتعرف على أمرين هما :-

٣- GRANT : منح الصلاحيات
٤- REVOKE : منع أو سحب الصلاحيات

حيث يتم من قبل المشرف العام على قاعدة البيانات منح ومنع الصلاحيات للمستخدمين ولكن قبل أن نتعرف على الصلاحيات التي تمنح يجب أولاً أن نتعرف على كيفية إنشاء مستخدمين ومن خلال التالي سوف نتعرف على :

١- إنشاء المستخدمين .
٢- تعديل بيانات المستخدمين .
٣- حذف مستخدم .

إنشاء مستخدم جديد

إن أي مستخدم لقاعدة بيانات أوراكل لا بد أن يكون له حساب خاص به وكل حساب له اسم وكلمة مرور خاصة به ولكي تنشئ حساب خاص بك قم بتشغيل لغة الإستعلام SQL ستظهر لك النافذة التالية :

حيث تطلب هذه النافذة بإدخال اسم المستخدم وكلمة المرور الخاصة بك فإذا كنت قد قمت بإنشاء مستخدم من قبل فاكتب اسم المستخدم وكلمة المرور الخاصة بك

أما إذا كنت مستخدم جديد فادخل بإسم SYSTEM كلمة مرور MANAGER حيث يعتبر هذا الحساب خاص بمدير قاعدة البيانات وهو الذي يملك أعلى الإمتيازات في قاعدة البيانات حيث لا يتم إنشاء حسابات المستخدمين إلا من خلال مدير قاعدة البيانات أو حساب يمتلك صلاحيات الإنشاء .

الصيغة العامة لإنشاء مستخدم جديد :

```
SQL> CREATE USER  اسم المستخدم  IDENTIFIED BY  كلمة المرور
```

مثال : قم بإنشاء مستخدم جديد باسم test وكلمة مرور tt .

```
SQL> create user test identified by tt;

تم تكوين مستخدم
```

بعد كتابة التعليمة السابقة والضغط على ENTER يجب أن تظهر الرسالة التالية User created ومفادها أنه تم انشاء المستخدم أما إذا ظهرت الرسالة التالية :

"الامتيازات غير كافية" ومفاد هذه الرسالة أنك ليس لديك أي صلاحية كي تنشئ مستخدم جديد والسبب هو عدم اتصالك بأوراكل وبمعنى أصح عدم اتصالك بمدير قاعدة البيانات فقد ذكرت سابقاً انه لا يمكن إنشاء مستخدم إلا من خلال الإتصال بـ MANAGER /SYSTEM

ولكي تتصل بمدير النظام أو أي مستخدم آخر استخدم الأمر Connect كالتالي :

```
SQL> Connect System /Manager
```

ومن ثم قم بإنشاء المستخدم المطلوب .

بعد أن قمت بإنشاء المستخدم test يجب أن تعرف أن هذا المستخدم لا يملك أي صلاحيات حتى ابسط الصلاحيات وهي حق الإتصال بالنظام فإذا خرجت من بيئة SQL ودخلت بهذا الحساب سوف يرفض الدخول بسب عدم امتلاكه صلاحية الإتصال حيث يجب أن تعرف أن أي مستخدم يجب أن يعطى امتيازات وصلاحيات ويتم ذلك من خلال مدير قاعدة البيانات وعن طريق الأمر grant .

الصيغة العامة لمنح أي صلاحية :

```
SQL> grant  ١ اسم الصلاحية ,٢ اسم الصلاحية TO اسم المستخدم;
```

- **امنح المستخدم test صلاحية الإتصال بقاعدة البيانات عبر الصلاحية Connect**

```
SQL> grant connect to test;
تم منح بنجاح
```

<u>ومن أهم الصلاحيات التى يمكن منحها للمستخدمين</u>

القدرة على الإتصال بقاعدة البيانات CREATE SESSION

إنشاء جداول CREATE TABLE .

إنشاء عرض CREATE VIEW

إنشاء روتين CREATE PROCEDURE

إنشاء تتابع CREATE SEQUENCE

ويمكن استخدام الصيغة السابقة لمنح الصلاحيات المذكورة .

- **امنح جميع الصلاحيات للمستخدم test.**

```
SQL> grant  DBA TO test;
تم منح بنجاح
```

ملاحظة :

يمكن لأي مستخدم إعطاء صلاحية خاصة به إلى مستخدم آخر ، كإعطاء صلاحية إضافة وتعديل وحذف ....ألخ في الجداول التي يمتلكها

- **امنح صلاحية استعراض البيانات select من جدول EMP الموجودة لدى المستخدم SCOTT للمستخدم test .**

```
SQL> grant select on scott.emp to test;
تم منح بنجاح
```

- امنح صلاحية استعراض البيانات select وصلاحية اضافة سجلات insert من جدول dept الموجود لدى المستخدم Scott للمستخدم test

```
SQL> grant select,insert on scott.dept to test;
تم منح بنجاح
```

منع(سحب) الصلاحيات أو الإمتيازات :-

منع أو سحب الصلاحيات هي من اختصاصات الأمر REVOKE حيث يقوم بإلغاء أي صلاحية تم منحها من قبل الأمر GRANT سواء الصلاحيات التي تم منحها للمستخدمين من قبل مدير قاعدة البيانات أو صلاحيات تم منحها من قبل مستخدم لمستخدم آخر .

الصيغة العامة للأمر REVOKE

```
SQL> REVOKE  اسم الصلاحية ١ , اسم الصلاحية٢ FROM اسم المستخدم ;
                سحب أو منع كافة الصلاحيات
SQL> REVOKE  ALL FROM اسم المستخدم ;
```

- اسحب صلاحية استعراض البيانات select وصلاحية اضافة سجلات insert من جدول dept الموجود لدى المستخدم scott من المستخدم test.

```
SQL> revoke select,insert on scott.dept from test;
تم سحب بنجاح
```

تعديل كلمة مرور مستخدم

بعد أن تعلمت كيفية إنشاء مستخدم جديد ، ولأي سبب كان أردت تغيير كلمة المرور الخاصة بك سوف تستخدم الأمر ALTTER USER والصيغة التالية توضح ذلك

```
SQL> ALTER USER اسم المستخدم IDENTIFIED BY كلمة المرور الجديدة
```

مثال : قم بتغيير كلمة المرور الخاصة بالمستخدم test من tt إلى rr .

```
SQL> ALTER USER test IDENTIFIED BY rr;

تم تبديل مستخدم
```

حذف مستخدم

```
SQL> drop USER اسم المستخدم ;
```

يستخدم الأمر drop لحذف مستخدم مع العلم انه لا يمكن حذف مستخدم إلا من خلال مدير قاعدة البيانات System أو من خلال مستخدم يمتلك صلاحيات الـ DBA .

```
SQL> drop user test;

تم إسقاط مستخدم
```