الجمهورية العربية السورية
وزارة التعليم العالي
جامعة حمص
الكلية التطبيقية
قسم تقنيات حاسوب

قواعد معطيات 1

القسم العملي

المحاضرة الثالثة

Structure Query Language

(SQL)

اعداد المدرسين:

م . زينب مراد

م . بتول الليوس

لغات قواعد المعطيات

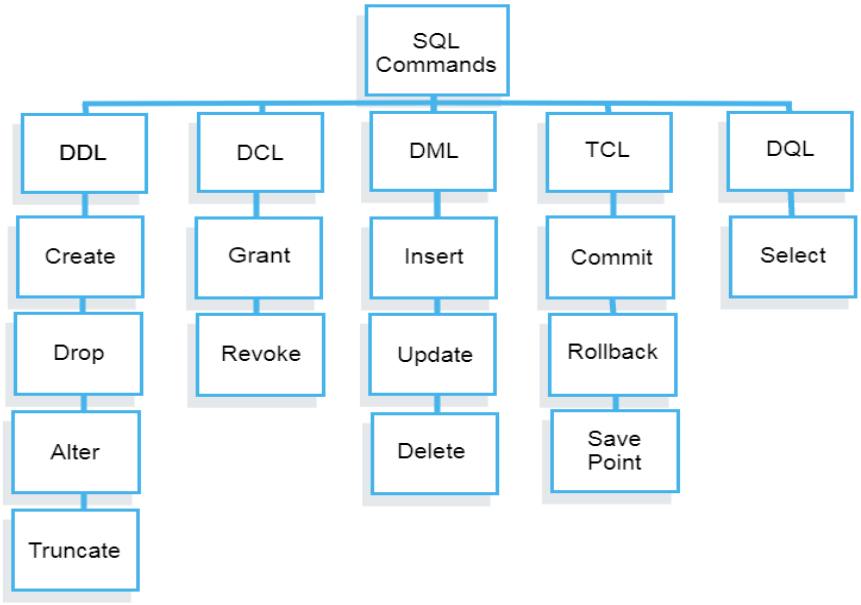
يوفر كل نظام إدارة قواعد معطيات على الأقل لغة واحدة تتيح لمستخدميه القيام بالعديد من المهام كتعريف بنية قاعدة المعطيات ، منح صالحيات الوصول إلى المعطيات .....وغيرها.

لغة الاستفسارات البنيوية

Structure Query Language (SQL)

تعتبر لغة قياسية للتعامل مع قواعد المعطيات العلائقية,وهي لغة تصريحية وليست لغة إجرائية ويمكن تقسيمها الى:

- **DQL (Data Query Language) لغة استعلام المعطيات**
  تستخدم للاستفسار عن المعطيات و استعادتها من قاعدة المعطيات .
- **DDL (Data Definition Language) لغة تعريف المعطيات**
  تستخدم للتعامل مع كائنات قواعد المعطيات مثل انشائها أو تعديلها أو ...
- **DML (Data Manipulation Language) لغة التعامل مع المعطيات**
  تستخدم للتعامل مع المعطيات مثل تعديلها أو اضافتها أو حذفها .
- **DCL (Data Control Language) لغة التحكم بالمعطيات**
  تستخدم لتعريف المستخدمين وتحديد صلاحياتهم.



مميزات لغة SQL :

- الوصول الى قاعدة المعطيات والتعامل معها.
- اضافة/حذف سجلات من/الى قاعدة المعطيات.
- تحديث السجلات في قاعدة المعطيات .
- استخراج البيانات من قاعدة المعطيات .

لغة تعريف المعطيات

DATA DEFINTION LANGUAGE (DDL)

قبل انشاء الجداول لابد من انشاء قاعدة معطيات لتمكين حفظ الجداول والمعلومات داخلها .
تسمية قاعدة المعطيات , الجداول والأعمدة تكون من اختيار المبرمج , لكن لابد من مراعاة الشروط التالية :

١) ان يكون الاسم متصل ويبدأ بحرف
٢) الحد المسموح به للاسم يكون من حرف واحد الى 30 حرف
٣) ان يكون الاسم غير مستخدم مسبقا" (يمنع تكرار الأسماء في نفس المكان)
٤) يجب الا يكون الاسم من الكلمات المحجوزة في لغة SQL

- **صيغة انشاء قاعدة معطيات جديدة :**

```
CREATE  DATABASE  DB_Name ;
```

- **لحذف قاعدة معطيات كاملة :**

```
DROP DATABASE DB_Name;
```

- **صيغة انشاء جدول جديد:**

```
CREATE TABLE Table_Name;
```

- **صيغة تعديل هيكل الجدول (إضافة أو حذف أعمدة):**

```
ALTER TABLE Table_Name  (ADD/DROP)  column_name ;
```

- **لحذف الجدول كامل :**

```
DROP TABLE Table_Name;
```

- **CREATE :** كلمة محجوزة تعني انشاء
- **DATABASE :** كلمة محجوزة تعني قاعدة معطيات
- **DB_Name :** اسم قاعدة المعطيات (اختياري) , يجب ان يكون مستوفي الشروط الأربعة المذكورة سابقا"
- **TABLE:** كلمة محجوزة تعني جدول
- **Table_Name:** اسم جدول (اختياري) وهو عادة نفس اسم الكيان الموضح في المخطط
- **DROP :** كلمة محجوزة تعني حذف
- **ALTER:** كلمة محجوزة تستخدم لتعديل هيكل جدول
- **ADD:** كلمة محجوزة تعني إضافة

- في لغة SQL يجب تسمية القيود ، تنطبق عليها نفس شروط تسمية الجداول والأعمدة

أنواعها

١) قيود المفاتيح الأساسية (Primary Key)
٢) قيود المفاتيح الفرعية (Foreign Key)
٣) قيود التحقق من الصحة (Check)
٤) قيود فريدة (Unique)
٥) قيود افتراضية (Default)
٦) قيود مطلوبة (Not NULL)

قيود المفاتيح الأساسية أو الرئيسية (Primary Key Constraints):

- يؤكد قيد المفتاح الأساسي عدم إدخال قيم مكررة في أعمدة معينة. ويمكنك استخدام قيود المفاتيح الأساسية لفرض التفرّد بالإضافة إلى التكامل المرجعي. على سبيل المثال، يعرّف عمود SID(الرقم الجامعي للطالبة) بشكل فريد لكل طالبة في جدول (Student)

- لا يمكن إدخال القيم الخالية (Null)
- يمكن أن يوجد مفتاح أساسي واحد فقط للجدول ويكون نوعه إما بسيط أو متفرع

```
CREATE  TABLE  College.Student ( SID  INT (7) ,
    Name VARCHAR (60) ,
    CONSTRAINT PK PRIMARY KEY (SID ) ) ;
```

Student

Name	SID
Nawal Hasan Al Ahmadi	3153363
Norah Saleh Al Malki	3152364
Sara Yousf Shaker	3122303

قبل إدخال أي سجل جديد يتم التأكد من عدم تكرار البيان الموجود في عامود المفتاح الأساسي

مثال:

```
create table project
(pnumber int not null primary key,
pname nvarchar(25),
dnum int)
```

ملاحظة:

إذا كان الجدول موجود و أردنا أن نحدد له مفتاح أساسي:

```
alter table Employees
add constraint pk_ssn
primary key(ssn)
```

Multi – Column Primary Key

مثال:

```
create table my4 ( id1 int ,id2 int ,name varchar(30), constraint pk1 primary key (id1,id2))
```

ملاحظة:

إذا كان الجدول موجود و أردنا أن نحدد له مفتاح أساسي:

```
alter table my4
add constraint pppk primary key(id1,id2)
```

مثال:

```
create table works_on
(essn int,
 pno int,
 constraint pk primary key clustered(essn,pno))
```

قيود المفاتيح الفرعية أو الثانوية (Foreign Key Constraints) :

- يعمل قيد المفتاح الفرعي بالاقتران مع المفتاح لفرض التكامل المرجعي ضمن جداول محددة. على سبيل المثال، يمكنك وضع قيد مفتاح خارجي في عمود SID (الرقم الجامعي للطالبة) في جدول (Course) لضمان أن القيمة المدخلة في هذا العمود تطابق القيمة الموجودة في عمود SID في جدول (Student)
- يمكن أن تكون القيم مكررة ،
- يمكن إدخال القيم الخالية (Null)
- يمكن أن يوجد مفتاح خارجي واحد أو أكثر في نفس الجدول

```
CREATE TABLE College.Course ( CID VARCHAR(7) PRIMARY KEY,
  SID INT (7) ,
  CONSTRAINT FK FOREIGN KEY (SID) REFERENCES Student(SID)) ;
```

Course

C_ID	SID
CCP241	3153003
CCP262	3153003
CCN222	3122111
CCP232	3153003

قبل إدخال أي سجل جديد يتم التأكد من وجود البيان في عامود المفتاح الأساسي في الجدول المصدر

Student

Name	SID
Nawal Hasan Al Ahmadi	3153003
Norah Saleh Al Malki	3152004
Sara Yousf Shaker	3122111

يقوم بمهمة إنشاء ارتباط بين جدولين , نسمي الجدول الذي عرفنا المفتاح الثانوي ضمنه ب (referencing Table) أما الجدول الذي يرتبط به referenced Table.

الارتباط يعني أي سطر نريد لإضافته Referencing table يجب أن يكون له سطر مقابل في Reference Table أو أن تكون قيمة المفتاح الثانوي NULL.

مثال:

```
create table dep_location
(dnumber int foreign key references department(dnumber)
on update cascade,
dlocation varchar(20))
```

ملاحظة1:

يمكن أن نحدد آلية تعامل المفتاح الثانوي في حال تمت عملية حذف أو تعديل في referenced table. أي ما هو تأثير عملية حذف أو تعديل سطر في Referenced Table على الأسطر المرتبطة به في Referencing Table(الذي يحتوي المفتاح الثانوي).

FOREIGN KEY REFERENCES <*table name*>(<*column name*>)
[ON DELETE {CASCADE|NO ACTION|SET NULL|SET DEFAULT}]
[ON UPDATE {CASCADE|NO ACTION|SET NULL|SET DEFAULT}]

On delete : تحدد تأثير عملية الحذف.
On Upate: تحدد تأثير عملية التعديل.
Cascade: سوف يؤدي الحذف(التعديل) إلى حذف(تعديل) الأسطر المقابلة
No action: لن يحدث أي تعديل أو حذف
Set Null: سوف تعطى القيمة الفارغة بدل من قيمة المفتاح الثانوي
Set Default: سوف تعطى القيمة الافتراضية(في حال وجودها) الخاصة بالحقل الذي يعبر عن المفتاح الثانوي

ملاحظة2:

الحقل المشار إليه Referenced Column يجب أن يكون مفتاح أساسي أو Unique.

ملاحظة3:

إضافة شرط المفتاح الثانوي لجدول موجود مسبقا:

```
alter table Employees
add constraint fk_d
foreign key(dno) references department(dnumber)
```

ملاحظة4:

أن يرتبط المفتاح الثانوي بحقل موجود في نفس الجدول. **Self-referencing**

```
alter table Employees
add constraint fk_s
foreign key(supperssn) references Employees(ssn)
```

القيود الفريدة (Unique) :

- يؤكد القيد الفريد عدم إدخال أية قيم مكررة في الأعمدة المعينة التي لا تكون مفتاح أساسي لجدول. على سبيل المثال، في جدول (Contact الاتصال) حيث يعد عمود (POBox صندوق البريد) هو عامود له قيمة فريدة (غير مكررة)
- يمكن إدخال القيم الخالية (Null)

```
CREATE  TABLE  College.Contact (  ID  INT (7)
                PRIMARY KEY , phone INT (10) ,
   PObox INT (5) UNIQUE ) ;
```

Contact

ID	Phone	PObox
49192	0531192302	**20011**
21419	0503340311	
29192	0502554391	**21900**

قبل إدخال أي سجل جديد يتم التأكد من عدم تكرار البيان الموجود في العامود الفريد

يمكن أن يحوي على قيم فارغة NULL

يتميز عن المفتاح الأساسي بأن الجدول الواحد يمكن أن يضم أكثر من حقل فريد.

مثال:

```
create table department
 (dnumber int identity not null,
  dname nvarchar(25),
  mgssn int unique,
  mgrstartdate date)
```

ملاحظة:
إضافة شرط القيم الفريدة على جدول موجود سابقاً

```
alter table department
add constraint u
unique(mgssn)
```

قيود التحقق من الصحة (Check) :

- **يحدد قيد التحقق من الصحة قيم البيانات أو التنسيقات المقبولة في عمود واحد أو أكثر في جدول. فمثلاً، يمكن أن تحتاج إلى عمود Salary (الراتب) في جدول (Employee) للسماح بإدخالات بيانات رقمية تحت شرط معين**
- **يمكنك إنشاء تعبير قيد بسيط للتحقق من البيانات بحثاً عن شرط بسيط ؛ أو يمكنك إنشاء تعبير معقد، باستخدام عوامل التشغيل (Booleanالمنطقية)**

مثال:

```
alter table Employees
add constraint ch
check
(brithday <getdate())
```

ملاحظة1:
كيف يمكننا أن نطبق Checkعلى قاعدة تحوي بيانات لا تراعي هذه الشروط؟

```
alter table Employees
with nocheck
add constraint ch1
check
(phone like '([0-9][0-9][0-9])[0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
```

ملاحظة2:
كيف يمكننا أن نعطل الشرط مؤقتا ثم نعيده:

- ```
  alter table Employees
    nocheck
    constraint ch1
  ```

- ```
  alter table Employees
    check
    constraint ch1
  ```

ملاحظة3:
حتى نعرف حالة الشرط إذا كان معطل أو لا يمكن استخدام الإجرائية التالية:

```
exec sp_helpconstraint Employees
```

القيود الافتراضية (Default) :

- تمكّنك القيود الافتراضية من تعريف القيمة التي سيتم تزويدها لعمود متى فشل مستخدم في إدخال إحدى القيم. على سبيل المثال، في جدول Student ضفنا عامود يحوي على معلومات ما إذا كانت الطالبة متخرجة أم لا
- يمكنك إرشاد قاعدة البيانات الخاص بك بإدخال القيمة "NO" كقيمة إفتراضية
- يجب أن تُحاط القيمة الافتراضية بعلامات تنصيص " اقتباس"

```
ALTER TABLE  College.Student ADD Graduated  VARCHAR (3)
   DEFAULT " No" ) ;
```

Student

Name	SID	Graduated
Nawal Hasan Al Ahmadi	3153363	No
Norah Saleh Al Malki	3152364	No
Sara Yousf Shaker	3122303	No

القيمة الافتراضية للعامود Graduated No

مثال:

```
create table dependent
(essn int foreign key references Employees(ssn),
 dependent_name varchar(20),bdate date default getdate())
```

ملاحظة:
إضافة شرط القيم الافتراضية على جدول موجود سابقاً

```
alter table Employees
add constraint df
default 'unknown' for address
```

قيود مطلوبة (Not NULL)

- تمكّنك القيود المطلوبة من تعريف القيمة التي سيتم تزويدها لعمود بأي قيمة غير فارغة NULL ، ويجب أن يكون الحقل ليس مفتاح أساسي للجدول
- على سبيل المثال، يعرّف عمود GovernmentID (رقم الهوية) بأنه حقل مطلوب في جدول (Student)

```
ALTER TABLE  College.Student ADD GovermentID INT (10)
   NOT NULL ) ;
```

القيمة عامود GovermentID مطلوبة عند أدخال الحقول

Student

Name	SID	Graduated	GovermentID
Nawal Hasan Al Ahmadi	3153363	No	0
Norah Saleh Al Malki	3152364	No	0
Sara Yousf Shaker	3122303	No	0


Any
questions