SQL

السلام عليكم ... النهاردة هنتكلم عن ال SQL

اول حاجة ال SQL ينقسم لحاجتين DDL,DML

DDL : select , insert , update , delete

DML : create , alter , drop ,truncate

نبدء باول حاجة الDML

1- Create

وديه بتستخدم لحاجتين create database , create table يعني لو عايزين نعمل قاعدة بيانات جديدة او نعمل جدول جوه قاعدة بيانات موجودة حاليا ..
طب هننشي قاعدة بيانات جديدة

```
Create database database_name
EX : create database  school
```

كده انا انشئت قاعدة بيانات جديدة اسمها school ....

طب بعد ما انشئت قاعدة البيانات عايز انشئ جدول بقي هشتغل عليه ...

```
Create table table_name
(
        Column1   dataType ,
        Column2   dataType ,
);
EX : Create table students
        (
                Name    varchar( 50 ),
                ID  int ,
                Age  int ,
);
```

كده انا انشئت جدول جديد اسمه students وعملت جواه 3 اعمدة كل عمود ليه اسم خاص بيه ....

|| طب انا ليه استخدمت varchar(50) وكان ممكن استخدم char(50)
لان ال char (50)هتحجزلي 50 مكان حتي لو انا خدت عدد قليل مثلا كتبت اسم
ahmedكده 5 حروف بردوا هيفضل محجوزلي 45 مكان فاضيين ..
انما لو استخدمت varchar(50)فهيا تعتبر زي ال string لو حاجز 50 مكان
ودخلت اسم ف 5 اماكن بس هيحجزلي ال5 اماكن بس ويلغي الباقي ||

2- Drop

ديه بنستخدمها لو عايزين نمسح حاجة مثلا قاعدة بيانات او جدول مش عايزينه
فبنمسحه ...
هنمسح ال tableالي انشاءناه فوق ..

```
Drop table table_name
EX : Drop table students
```

كده مسحنا الجدول الي اسمه students

ناو هنمسح قاعدة البيانات الي انشاءناها فوق ..

```
Drop database database_name
EX : Drop database school
```

كده نبقي مسحنا قاعدة البيانات الي اسمها school

3- Alter

ديه بنستخدمها عشان نعدل ع ال table سواء نضيف عمود جديد او نحذفه ..
حاليا عايز اضيف column جديد وليكن اسمه Address

```
Alter table table_name add column_name datatype
EX : alter table students add Address varchar(30);
```

طب لو عايز احذفه تاني بقي فهنستخدم ايه !!
- هنستخدم alter
لو فكرت تستخدم Dropتبقي غلطت..قلنا drop لقاعدة بيانات او جدول كامل
بس..

Alter table table_name drop column column_name
EX : Alter table table_name drop column Address ;

كده انا حذفت العمود الي اسمه Address

4- Truncateديه بتسخدمها عشان افضي ال table بتاعي لو في اي بيانات...
بس من غير ما تمسحه بيبقي موجود ك جدول فاضي خالص ...
طب ناو انا عندي جدول students فيه بيانات عايز اشيلها ..

Truncate table table_name ;
EX : truncate table students ;

كده انا فضيت الجدول خالص مفهوش اي بيانات قديمة ..

نبدء بقي ف الجزء التاني DDL

1- Insert
ديه بنستخدمها عشان نملي بيانات ف الجدول الي احنا انشأناه .....
عايزين ندخل بيانات ف الجدول بقي ناو بندخل صف كامل

Insert into table_name values (value1 , value 2 , value3)
EX :
Insert into students values(‘mohamed ‘ , 1 , 18);

كده انا ضيفت اول صف كامل ... الاسم وال idوالسن ... ع حسب الاعمدة الي
انا معرفها كلها بقي بزود ...

2- Update
ديه بستخدمها لو عايز اعدل اي بيانات انا دخلتها قبل كده ....

Update table_name set column_name=value where
column_name = some_value ;
EX :
update students set age = 22 where id = 2 ;

كده انا غيرت السن ل 22 للشخص الي ال idبتاعه 2 .

3- delete
ديه بستخدمها لو عايز امسح صف كامل من الجدول بتاعي ....

Delete from table_name where column_name =some_value;
EX :
Delete from students where id =2 ;

كده احنا مسحنا الصف الي فيه ال id بيساوي 2 ...
طب احنا عايزين نشيل كل البيانات الي ف الجدول باستخدام ال delete

Delete from table_name ;
EX : delete from students ;

كده فضيت الجدول خالص نفس استخدام ال truncate

---

select – 4
ديه بنستخدمها عشان نستخرج معلومات معينة من الجدول بتاعنا ..
مثلا عايز اجيب البيانات الي ف الجدول كلها ...

Select * from table_ name
ال * = all يعني كل حاجة ...

EX : select * from students ;
طب لو عايز اجيب عمود معين عندي مثلا اشوف كل الاسماء

Select column_name from table_name ;
EX : select name from students ;
EX : select id,name from students ;

في عندي نوع تاني من ال selectوهو ال select distinct

ده بيعمل ايه بيشيل التكرار مثلا عندي ال idمتكرر 3 مرات هيشيل مرتين ويطبعه مرة واحدة ..

Select distinct column_name from table_name;
EX : select distinct id from students ;

احنا كنا استخدمنا where قبل كده بس معرفناش بتاعت ايه ولا بتعمل ايه ....

ديه بنستخدمها عشان نحدد الشرط بتاعنا مثلا لما بكون عايز امسح صف بقوله

Delete from students where id =2 ;
حددت شرط انه يمسح الصف الي فيه ال idبيساوي 2 ...

Where column_name operator value

ده السنتكس الاساسي بتاعها ال operatorهما 3 عمليات = ، < ، >

نشوف مثال بال deleteومثال select

EX 1 : Delete from students where id =2 ;

EX 2 : Select name from students where age > 20 ;

ديه حاجة جديدة ف ال selectاسمها like و ديه بنستخدمها عشان نجيب بيانات صف معين او حاجة معينة من الجدول ....

مثلا عايز اجيب بيانات كل الي اسمهم بيبدأ بحرف m

Select * from table_name where column_name like '%' ;

EX :

Select * from students where name like 'm%' ;

ال % ديه بتدل ع ان في حروف بعدها يعني mahamed يبقي % ='ahamed'

ونكتبها m%كده ....

كده انا قلتله هاتلي بيانات كل الاسماء الي بتبدأ بحرف ال m

في حاجتين كمان "%m" كده بقوله هاتلي الاسماء الي اخرها حرف m بس .

"%m%" كده بقوله هاتلي كل الاسماء الي فيها حرف mسواء ف الاول،النص،الاخر.

هنشوف ناو حاجة جديدة وهيا الترتيب order by

ديه بنستخدمها عشان نرتب بمزاجنا واحنا بنظهرهم يعني عايز اظهر كل حاجة بس يكون ال ageمن الكبير للصغير ....

احنا عندنا نوعين من الترتيب

= من الصغير للكبير وديه بنستخدم فيها ASC

= من الكبير للصغير وديه بنستخدم فيها DESC

Select * from table_name order by column_name ASC ;

EX :

Select * from students order by age ASC ;

كده بقوله اظهرلي كل البيانات الي ف الجدول بس يكون السن مترتب من الكبير للصغير

.. وباقي الاعمدة بتترتب ع حسب العمود ده ....

ال AND _OR

هنا بستخدم ال AND لو عايز اجيب بيانات معينة فبحط شوية شروط مش شرط واحد فلو اتحقق كل الشرط ديه هيجبلي النتيجة .... لو شرط واحد متحققش مش هيجبلي حاجة...

مثلا عندي صفين

Id 1 = 1 name 1 = Mohamed age 1= 20

Id 2 = 2 name 2 = Ahmed age 2= 25

```
Select * from table_name where column_name = value AND
column_name = value ;
```

بنستخدم اسماء اعمدة مختلفة زي مهنشوف ف المثال ...

فجيت قلتله كده <<

```
Select  *  from students where id =2  AND name = 'Ahmed' ;
```

هيطبعلي البيانات بتاعت احمد كلها .....

طب لو قلتله <<

```
Select  *  from students where id =1  AND name = 'Ahmed' ;
```

كده مش هيطبعلي حاجة لان كل الشروط متحققتش يعني ال id ب1 بس الاسم مختلف فمش هيلاقي نتيجة يجيبها ...

انما ممكن نستخدم ال OR لو شرط واحد اتحقق هيجبلي ناتج ليه ..

```
Select * from table_name where column_name = value OR
column_name = value ;
```

بنستخدم اسماء اعمدة مختلفة زي مهنشوف ف المثال ...

EX :

```
Select * from students where id =1  OR name = 'Ahmed' ;
```

هنا هيلاقي اول حاجة الشرط الاول id ب1 بعدين OR هيخش يلاقي اسم مختلف عن ال ID اللي ف الاول ..... فهينفذ الشرطين و هيطبع بيانات الشخص اللي ال id بتاعه 1 و هيطبع بيانات الشخص اللي اسمه Ahmed

---

ال IN

ديه بنستخدمها لو عايز اجيب بيانات صفوف كتير وانا عارف حاجة واحدة من البيانات ديه مثلا عندي جدول فيه الاسم وال id والعنوان والسن وانا عارف ال id بس هقدر اجيب باقي البيانات ...

```
Select * from table_name where column_name IN
(value1,value2 , ……..) ;
EX :
select * from students where id IN (1,2,3) ;
```

كده هيجبلي بيانات الناس الي ال idبتاعها 1و2و3 بس ...

ال Between….AND

ديه بستخدمها لو عايز احدد نسبة معينة مثلا عايز اجيب بيانات الناس الي بين سن ال 15 وال 30 بستخدم Between….AND

```
select column_name from table_name where column_name
between value1 AND value2  ;
```

نشوف مثلا عايز اجيب بيانات الناس الي بين idال 1و10 ...

```
select * from students where id  between 1 AND 10 ;
```

ال Alias

ديه بنستخدمها عشان نغير اسم الجدول او اسم عمود ف الجدول اثناء العرض ...

```
Select column_name as new_name from table_name ;
EX :
select  name as [ Full Name ] from students ;
```

كده انا غيرت اسم العمود من name وخليته [Full Name]

- لازم نستخدم الاقواس عشان حاطين مسافة بين الكلمتين ...

ال foreign key ……primary key

ال primary key ده المفتاح الاساسي بخليه حاجة اساسية ..

ال foreign key ده المفتاح الفرعي بستخدمه عشان اربطه بال primary عشان استخدمه ف جدول تاني ...

نشوف اول حاجة ال primary key

```
Constraint PK_name PRIMARY KEY ( column_name) ;
```

فمثلا هعمل جدول فيه name ,id ,age واخلي ال id هو ال primary key

```
create table employee
(
id int ,
age int ,
```

```
name varchar(50),
constraint PK_ID PRIMARY KEY (id)
);
```

كده عملت ال idهو ال primary key

او في طريقة تاني اني اعرف الprimary key جنبه وانا بعرفه كده

Id int PRIMARY KEY ,

الطريقتين صح ....

بعدين عايزين نعمل جدول تاني عشان نستخدم ال foreign key

```
create table customer
(
C_id int ,
C_age int ,
C_name varchar(50),
E_id int ,
constraint FK_E_C foreign key (E_id) references employee (id)
);
```

كده عملنا ال foreign key بيشاور ع ال primary key الي ف الجدول الاول employee

---

عايزين بقي نربط الجدولين ببعض ..... احنا ربطنا عن طريق ال id فاحنا هنقول ..

```
Select column_name1 , column_name2
from  table_name1 , table_name 2
Where table_name1.column_name1 =
tablename2.column_name2 ;
EX :
Select id , C_id from  employee , customer
Where  employee.id = customer.C_id ;
```

كده ربطنا الجدولين ببعض هيطبعلي الid و C_id لما يكونوا بيساوا بعض ...

هنتكلم ناو عن ال union .....union all

انا عندي جدولين الاول فيه id,name والتاني فيه id ,name نفس اسماء الصفوف بس اسم الجدول مختلف .....

انا عايز اطلعهم كلهم ف جدول واحد تحت بعض ...

```
Select column_name1  from table_name1
Union
```

```
Select column_name2  from table_name2 ;
```

كده هيجبلي الي ف الجدول الاول والي ف الجدول التاني تحت بعض ولو في حاجة متكررة مش بيعرضها ....

طب فرضا انا عايز اعرض المتكرر ده بستخدم Union all

```
Select column_name1  from table_name1
Union all
Select column_name2  from table_name2 ;
```

يعني الفرق بينهم ان واحدة بتعرض المتكرر والتاني بتلغي التكرار وتعرضه مرة واحدة ....

ناو وصلنا لل SQL function ...Group by .. having

اول حاجة عندنا ال functions....

Sum , max,min , count , avg ده اغلب المستخدم ....

```
Select function_name(table_name.column_name )  from
tabe_name ;
 /*max min count avg sum */
select sum(person.salary) from person ;
```

ممكن نغير sum ديه براحتنا حسب نوع الفنكشن الي احنا عايزينها ...

نستخدم بقي ال having هيا للشروط زي where بس في فرق بينهم

ال where بنستخدمها لو شرط ع صف واحد بس ....

انما ال having تستخدم ع الجدول كله حتي لو 100 صف ....

```
SELECT name,SUM(salary) FROM person
GROUP BY name HAVING SUM(salary)>400
```

اخر حاجة ال join

ده بيتقسم ل شوية اجزاء >>

1- inner join

ده بستخدمه عشان اجيب المشترك بين عمودين معينين ف جدولين مثلا الجدول الاول عمود الid والجدول التاني عمود ال id هيطلعلي الي شبه بعض ف الجدولين ....

```
SELECT column_name(s)  FROM table1 INNER JOIN
table2 ON table1.column_name=table2.column_name;
```

2-left outer join

ده بستخدمه عشان اجيب العمود الي ف الجدول الي ع الشمال واجيب الي مربوط بيه ف الجدول الي ع اليمين لو ملقاش حاجة مربوطة بيجيب NULL

```
SELECT column_name(s)  FROM table1 LEFT OUTER JOIN table2  ON table1.column_name=table2.column_name;
```

3- right outer join

ده بستخدمه عشان اجيب العمود الي ف الجدول الي ع اليمين واجيب الي مربوط بيه ف الجدول الي ع الشمال لو ملقاش حاجة مربوطة بيجيب NULL

```
SELECT column_name(s)  FROM table1 RIGHT OUTER JOIN table2 ON table1.column_name=table2.column_name;
```

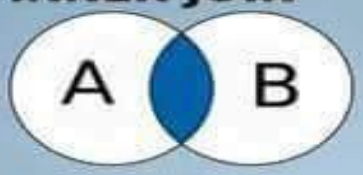
4-full outer join

ده بستخدمه عشان اجيب العمود الي ف الجدول الي ع الشمال والعمود الي ف الجدول الي ع اليمين سواء مربوطين ببعض او لا الاتنين هيتطبعوا ....

```
SELECT * FROM table1 FULL JOIN table2 ON table1.column_name=table2.column_name;
```

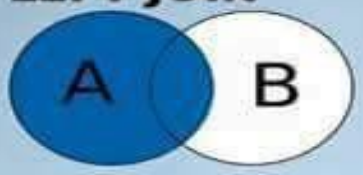
SQL JOINS

INNER JOIN



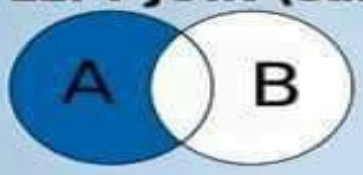

```
SELECT *
FROM A
INNER JOIN B ON A.key = B.key
```

LEFT JOIN



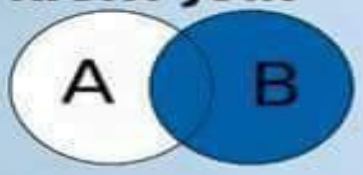

```
SELECT *
FROM A
LEFT JOIN B ON A.key = B.key
```

LEFT JOIN (sans l'intersection de B)



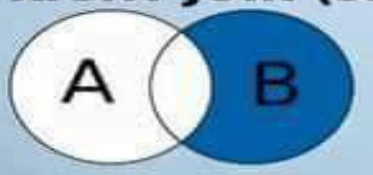

```
SELECT *
FROM A
LEFT JOIN B ON A.key = B.key
WHERE B.key IS NULL
```

RIGHT JOIN



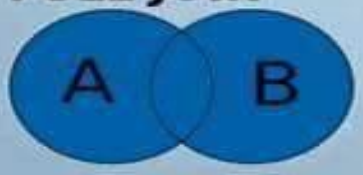

```
SELECT *
FROM A
RIGHT JOIN B ON A.key = B.key
```

RIGHT JOIN (sans l'intersection de A)




```
SELECT *
FROM A
RIGHT JOIN B ON A.key = B.key
WHERE B.key IS NULL
```

FULL JOIN




```
SELECT *
FROM A
FULL JOIN B ON A.key = B.key
```

FULL JOIN (sans intersection)

```
SELECT *
FROM A
FULL JOIN B ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL
```

sql.sh