



Subquery & aggregate function

د. روان قرعوني

محتوى مجاني غير مخصص للبيع التجاري

قواعد معطيات 1

25/5/2023

RB Informatics;

مقدمة:

تحدثنا في المحاضرة السابقة عن الاستعلام بلغة SQL وعن المتحولات الحدودية وانتهينا بالحديث عن العمليات على المجموعات وفي هذه المحاضرة سنتحدث عن ال Subqueries وال Aggregate functions.

التوابع التجميعية Aggregate Functions

تطبق التوابع التجميعية على مجموعة من القيم وتعيد قيمة واحدة.

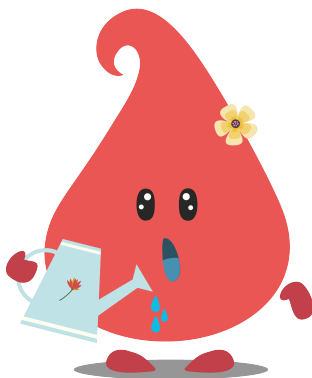
مناقشة:

في الكثير من الأحيان قد نحتاج إلى معرفة ال max للرواتب أو قيمة أصغر راتب للموظفين أو مجموع الرواتب sum أو عدد الموظفين count أو وسطي أعمار الموظفين وغيرها من الاستعلامات التي قد ترد. وهنا نحتاج إلى ما يسمى بالتوابع التجميعية.

أهم هذه التوابع:

التابع	دلالة التابع
Avg	المتوسط Average
Min	الأصغر Minimum
Max	الأكبر Maximum
Sum	المجموع Total
Count	العدد Count

يتم وضع تلك التوابع ضمن تعليمة Select



Success In No Accident.

It Is Hard Work, Perseverance, Learning, Studying, Sacrifice, And **MOST** Of All Love Of What You're Doing.

■ بالاعتماد على Employee-Company Database سنطبق بعض الأمثلة للتوضيح:

```
Select Max (salary)
from Employee
or using Alias
Select Max (e.salary)
from Employee e
```

يعيد لنا هذه الاستعلام قيمة أكبر راتب موجود في جدول الموظفين Employee.

```
Select Count (*)
from Employee
```

يعيد لنا هذا الاستعلام عدد أسطر جدول ال Employee.

```
Select Count (distinct name)
from Employee
```

يعيد لنا هذا الاستعلام عدد أسماء الموظفين بدون تكرار.

```
Select Count (*) from (
Select distinct name
from Employee)
```

يعيد لنا هذا الاستعلام نفس نتيجة الاستعلام السابق ولكن الفرق يكمن في أنه في هذا الاستعلام قمنا بجلب الأسماء الغير مكررة من جدول ال Employee ووضعهم في جدول جديد في ال view ومن ثم قمنا بحساب عددهم. على عكس الاستعلام السابق الذي قمنا فيه بعد أسماء الموظفين غير المكررة مباشرة من جدول ال Employee الموجود في ال Database.

إن هذا الشكل من الاستعلام هو أبسط أنواع ال (Subqueries).

■ ملاحظة:

كل التوابع التجميعية تتجاهل ال null ما عدا تابع ال count.

مثال على تابع يتجاهل ال null:

نريد حساب ال Avg للرواتب. عندها يقوم بجمع جميع الرواتب باستثناء ال null ومن ثم يقوم بتقسيمها على عددها.

لدينا 1000 موظف منهم 900 موظف رواتبهم غير معدومة و 100 موظف رواتبهم معدومة. لحساب الـ Avg للرواتب:

$$\frac{\text{مجموع قيم الرواتب}}{900 (\text{عدد الموظفين})} = avg \text{ salary}$$

أما تابع الـ count فلا يحذف القيمة null أي لو فرضاً قمنا بالاستعلام عن

Select count (*)
from Employee

أي أنه سيُردّ لنا 1000 موظف وليس 900 لأنه يقوم بحساب الـ row وإن كانت الـ salary مساوية للقيمة null.

Employee

ID	Name	Age	Address	Salary	CDd
1	Ahmad	20	Al-Mazeh	1000	1
2	Husam	25	..	2000	1
3	Khaled	30	..	3000	1
4	Wael	35	Maliki	4000	2
5	Loai	40	..	5000	2

Company

ID	Name
1	Syr
2	MTN
...	...
...	...
...	...

Employee - Company Database

Group by

لنناقش معاً:

نريد الحصول على عدد موظفي شركة MTN وعدد موظفي شركة Syriatel بحيث نحصل كنتيجة على الجدول التالي:

Syr	3
MTN	2

في المحاضرة السابقة لاحظنا أن الاستعلام التالي خاطئ:

Select *
from Employee e, Company c



لأنه سيُرَد لنا الجداء الديكارتي لجدولي (Employee) و (Company) وهذا خاطئ كنتيجة للاستعلام الذي نريد فيه الحصول على معلومات موظفي كل شركة (سنحصل على جدول يحوي 10 أسطر أي لن نحصل على كل موظف مع الشركة التي يعمل فيها وإنما كل موظف مع كل شركة سواء كان يعمل فيها أم لا وهذا هو ما قصدناه بالجداء الديكارتي)

ولحل هذه المشكلة

قمنا بوضع شرط يحدد كل موظف مع الشركة التي يعمل فيها فأصبح الاستعلام كالتالي:

```
Select *
from Employee e, Company c
where e.cid=c.id
```

إن الاستعلام السابق يعطينا كنتيجة جدول يحوي معلومات كل موظف مع الشركة التي يعمل فيها ولكن الاستعلام الذي نريده هو فقط عدد موظفين كل شركة.

؟ إذا ما هو التعديل الذي يجب القيام به لنحصل على ذلك؟! ؟

بما أنه ذكر في الاستعلام كلمة (عدد) إذا سيخطر لنا مباشرة استخدام الدالة التجميعية التي تحدثنا عنها قبل قليل **count** لنجرب معاً:

```
Select count (*)
from Employee e, Company c
where e.cid=c.id
and c.name=Syr
```

```
Select count (*)
from Employee e, Company c
where e.cid=c.id
and c.name=MTN
```

إن كل استعلام سيعيد لنا عدد موظفي شركة (شركة واحدة المحددة بـ c.name) بجدول منفصل أي سنحصل على ما يلي:

Syr	3
-----	---

MTN	2
-----	---

إن السؤال الذي سنطرحه أنه لماذا لم نقوم بوضع وكتابة الاستعلام كالتالي:

```
Select count (*)
from Employee e, Company c
where e.cid=c.id
and c.name=MTN
and c.name=Syr
```

لأنه هنا بشرط الـ and نقوم بجمع شرطين معاً هو أن يقوم بعدد الموظفين في كل من الشركتين Syr و MTN معاً وهذا ليس ما نريده

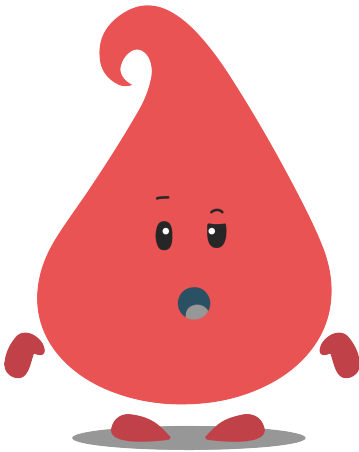
	5
--	---



ولا كالاستعلامين السابقين حتى

ماذا لو قمنا بوضع UNION؟؟

```
Select count (*)
from Employee e, Company c
where e.cid=c.id
and c.name=Syr
UNION
Select count (*)
from Employee e, Company c
where e.cid=c.id
and c.name=MTN
```



إن UNION قدمت لنا الاستعلام السابق كنتيجة بجدول واحد ولكن كجدول مُبهم القيم:

1	3
2	2

ماذا لو كان لدينا عدد كبير من القيم، كيف سنعرف كل قيمة إلى أي شركة؟؟

وهنا ظهرت لدينا Group by التي تقدم لنا إمكانية تجزئة العلاقة وفق قيمة أحد الواصفات ومن ثم إجراء عملية تجميع على واحد أو أكثر من الواصفات الباقية، إن استخدام Group by مُجدد و خاصة عندما لا تكون لدينا فكرة سابقة عن محتوى الـ attribute كالمثال السابق، من الممكن جداً أن لا تكون لدينا خلفية عن أسماء جميع الشركات الموجودة في الداتايز إلا بعد العودة و إلقاء النظر عليها.

فيصبح شكل الاستعلام كالتالي:

```
Select count (*)
from Employee e, Company c
where e.cid=c.id
group by c.name
```

ولكنه لا يزال غير صحيحاً لأن الذي تقوم بعملية Group by عليها يجب أن يكون لدينا ضمن أحد خيارات الـ Select

```
Select c.name, count (*)
from Employee e, Company c
where e.cid=c.id
group by c.name
```

■ ملاحظة:

لا يمكننا وضع c.address مثلاً بجانب الـ c.name

```
Select c.name, c.address, count (*)
from Employee e, Company c
where e.cid=c.id
group by c.name
```

لأنه كيف سنضع address بجانب Syriatel وقد يكون لـ Syriatel أكثر من address وأكثر من فرع؟؟
إذاً الحقول التي نقوم بـ Select عليها هي حصراً التي نريد القيام بـ Group by عليها والتوابع التجميعية.

لا يمكننا وضع count قبل Group by (يعني الدالة التي نقوم بعمل Group by عليها)

لأننا على أساس ماذا سنقوم بالعد؟!

إذا نضع بعد Select العمود الذي نريد القيام بـ Group by عليه ومن ثم التابع التجميعي الذي نريد تطبيقه.

يمكننا القيام بـ Group by على أكثر من attribute ولكن ليس أكثر من 2 غالباً

```
Select c.name, c.address, count (*)
from Employee e, Company c
where e.cid=c.id
group by c.name, c.address
```

سؤال:

ماذا لو أننا بعدما قمنا بـ Group by حسب اسم الشركة أردنا معرفة الشركات التي عدد موظفيها أكثر من 2 (أكبر من 2)

هل يمكننا استخدام where ولماذا؟؟

لا، لا يمكننا استخدام where لأن where تستعمل على مستوى الـ row في الـ Tables في الـ database ولكننا هنا قمنا بـ Group by أي أننا لدينا هو النتيجة فقط ولم تعد الجداول هي ما يظهر في الـ view وإنما نتيجة Group by

ومن هنا ظهر لدينا مصطلح having

تستعمل لوضع شرط على نتيجة استعلام ناتج عن القيام بعملية تجميعية.

فيصبح الاستعلام بالشكل التالي:

```
Select c.name, count (*) as x
from Employee e, Company c
where e.cid=c.id
group by c.name
having x>2
```

إذا where و having كلاهما يعملان عمل فلترة ولكن where على مستوى row أما having على نتيجة عملية تجميعية.

```
Select c.name, count (*)
from Employee e, Company c
where e.cid=c.id
group by c.name
X having salary>1000
```

لأن count قامت بجمع قيم الـ salary لكل شركة أي أن القيمة الظاهرة لنا هي قيمة الجمع وليس كل salary على حدى أي لم يعد لدينا access للـ salary الواحد.

Sub Query الاستفسارات الجزئية المضمنة:

الاستفسار الجزئي هو تعبير من الشكل Select - from- where ويكون مضمناً في استفسار آخر.

■ أمثلة على استخدامات الـ Sub Query:

In:

```
select *
from school
where parent.id in (select id
from Employee)
```

استعلام يجلب لنا معلومات الطلاب الذين يعمل والدهم موظف.

نلاحظ أن الشرط كان على الـ id إذا يجب أن تعيد Select المضمنة قيم id {أي من نفس النوع الذي وضع عليه الشرط}.

```
select *
from Employee e
where c.name=syr
And e.salary > some (
select e.salary
from company c
where c.name= MTN)
```

:Some

استعلام يجلب لنا الموظفين في Syriatel والتي رواتبهم أكبر من راتب واحد على الأقل من رواتب الموظفين في MTN.

All:

```
select *
from Employee e
where c.name = Syr
And e.salary > all (
  select e.salary
  from company c
  where c.name = MTN )
```

يجلب لنا نتيجة مشابهة لنتيجة الاستعلام السابق (يقوم بجلب موظفي Syriatel الذين رواتبهم أكبر من كل رواتب موظفي MTN)

```
where exist (
  select *
  from Employee e
  where e.salary > 1000)
```

:Exist

أي أنه لو كان لدينا موظف واحد على الأقل راتبه أكبر من 1000 سيتحقق لدينا الشرط.

Select:

```
select *
from (select ....)
```

هو أبسط أنواع
Sub Queries

أي فرضاً لو كان لدينا مليون attribute ولكننا نريد attributes 2 أو 3 أو ... فنقوم بإحاطتهم ب select ثم الاختيار منهم.

👉 توضيح:

```
select x.A, x.B
from (select ...) as x
```

سَتَرُدُّ لنا جدول نقوم بعمل select منه

A	B	C	D	E	F
...
...
...
...

X

ستعيد لنا قيمة واحدة

```
Select c.id, c.name, (
  Select count (*)
  from Employee e
  where e.cid=c.id)
from company c
```

Select داخل الـ columns

سيرد لنا هذه الاستعلام id الشركات مع أسمائهم مع عدد موظفيهم.

c.id c.name count

1	Syr	3
2	MTN	2

DB Modifications

وهي تعني بأننا نستطيع القيام بعمليات delete, insert, update على database ضمن استعلام.

```
insert into Employee (name, salary)
Values (Ahmed, 1000)
```

هنا نستطيع وضع select عوضاً عن values
insert into Employee (name, salary)
select x, y From A

```
Update Employee e
set name=(select
  x from a
  where a.id=e.id)
```

```
delete From Employee
where id in (select....)
```

SQL Case Expression

تشبه if then Else

case syntax:

```
case
  when condition 1 then Result 1
  when condition 2 then Result 2
  Else Result
End
```

Select name, age
From Employee

يعيد الاستعلام السابق اسم وعمر كل موظف

Name	Age
Ahmed	20
Hasan	23
Khaled	25

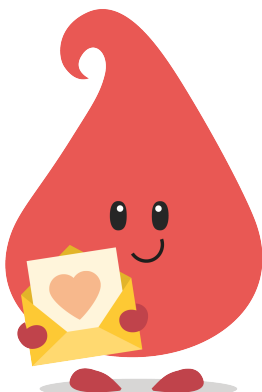
نريد استعلام حسب العمر (Age) يقوم بطباعة جملة ما وهنا نستخدم **case Expression**

```
Select name, age;
case
when age >20 then 'old'
when age <=20 then 'young'
Else ' _ '
End as age_description
From Employee
```

Name	Age	age_description
Ahmed	20	Young
Hasan	23	Old
Khaled	25	Old

يقوم بجلب الاسم والعمر وبطباعة جملة حسب العمر ووضعها ضمن attribute جديدة تدعى age_description ؛
فإذا كان age أكبر من 20 يطبع old وإذا كان أصغر أو يساوي 20 يطبع young .

The End.



Eventually, you'll end up where you need to be,
with who you're meant to be with, and doing
what you should be doing.