



جامعة المنارة
كلية الهندسة
قسم الهندسة المعلوماتية

مقرر قواعد بيانات 1 استخدام توابع التجميع

د.م. كندة أبو قاسم
م. جمال محمود – م. جهاد عيسى

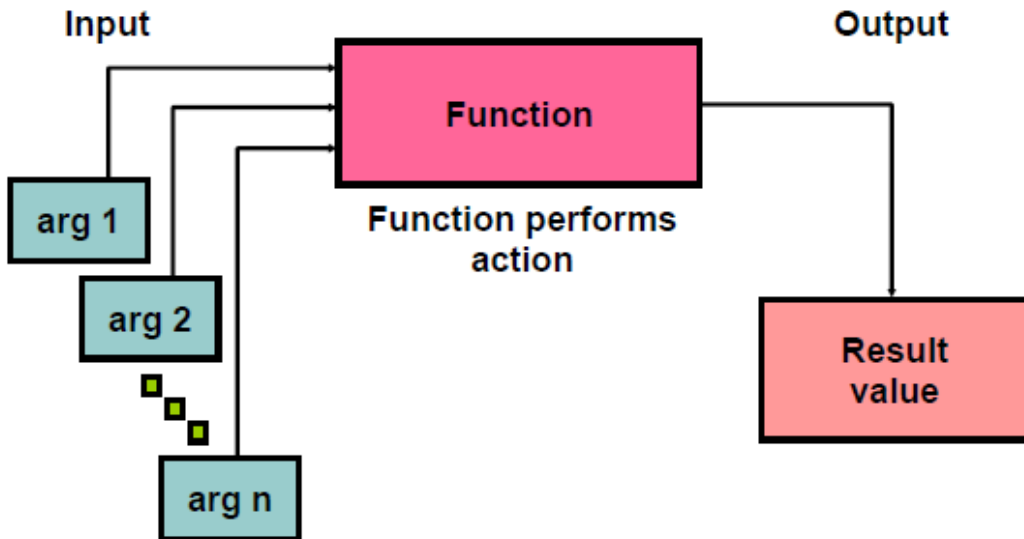
جلسة الأسبوع الثامن
(الفصل الثاني 2023 | 2024)

الغاية من الجلسة

- ✓ التعريف بمفهوم التابع.
- ✓ التعريف بأنواع التوابع في SQL SERVER.
- ✓ تجميع البيانات باستخدام توابع التجميع واللاحقة GROUP BY.

مفهوم التابع في SQL

التابع هو عبارة عن تعبير رياضي يأخذ مجموعة من قيم الدخل التي ندعوها مُعاملات، ويعيد قيمة خرج وحيدة ندعوها قيمة التابع. تتعلق قيمة التابع (أي الخرج) بمُعاملاته (أي بالدخل)، كحال التابع الذي يقوم بحساب مجموع قيم عددية.



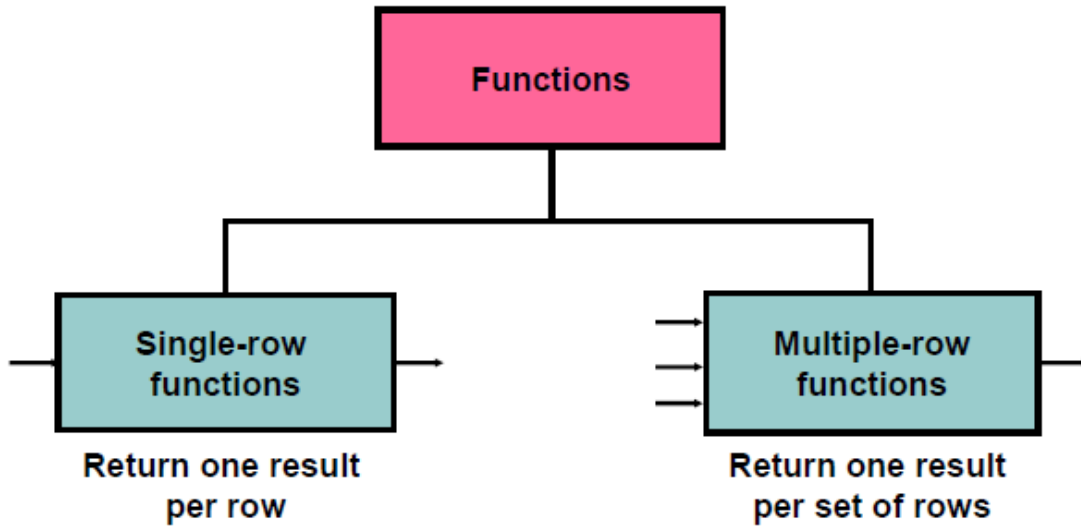
إن التوابع هي من المزايا المميزة للغة SQL، ويمكن أن يتم استخدامها للأهداف التالية:

- إجراء العمليات الحسابية على البيانات.
- تعديل بعض عناصر البيانات المنفردة.
- معالجة خرج مجموعات من الصفوف.
- تنسيق عرض التواريخ والأرقام .
- تحويل أنواع بيانات الأعمدة.

تنقسم التوابع في لغة SQL SERVER إلى نوعين:

- ✓ التوابع المعرفة من قبل النظام System Defined Functions (وهي موضوع درسنا الحالي).
- ✓ التوابع المعرفة من قبل المستخدم User Defined Functions (وهي خارج إهتمامنا في هذا المقرر).

فيما يتعلق بالتوابع المعرفة من قبل النظام فهي تنقسم إلى نوعين:



1. توابع الصف الواحد Single-Row Functions أو التوابع السلمية Scalar Functions وهي توابع تؤثر في صف واحد من البيانات وتعيد نتيجة واحد لكل صف، وهي على عدة أشكال نذكر منها:

- a. توابع التعامل مع المحارف.
- b. توابع التعامل مع الأرقام.
- c. توابع التعامل مع التاريخ.
- d. توابع التحويل.

2. توابع الصفوف المتعددة Multiple-Row Functions أو توابع التجميع Group Functions وهي توابع تعمل على مجموعات من الصفوف وتعيد نتيجة واحدة لكل مجموعة من الصفوف.

توابع الصفوف المتعددة أو توابع التجميع

هي مجموعة من التوابع التي تستخدم لتوليد خلاصات إجمالية عن مجموعات الصفوف المخزنة في جدول ما من قاعدة البيانات.

EMPLOYEES

DEPARTMENT_ID	SALARY
90	24000
90	17000
90	17000
60	9000
60	6000
60	4200
60	5800
60	3500
50	3100
60	2800
60	2500
80	10500
80	11000
80	8600
	7000
10	4400

Maximum salary in
EMPLOYEES table

MAX(SALARY)
24000



تابع العدد COUNT

يحسب التابع COUNT عدد البيانات الموجودة في الجدول من أجل حقل معين أي يقوم بحساب عدد الأسطر الموجودة.

الصيغة العامة لهذا التابع:

```
SELECT COUNT ([* |ALL|DISTINCT] column_name)
FROM table_name;
```

- ✓ يستخدم الخيار ALL عندما نريد الحصول على عدد البيانات الموجودة في الجدول، بالنسبة لحقل معين، مع استبعاد القيم التي تساوي NULL. يعتبر هذا الخيار هو الخيار التلقائي في حال عدم تحديد أي من الخيارين DISTINCT أو ALL.
- ✓ يستخدم الخيار DISTINCT عندما نريد الحصول على عدد البيانات الموجودة في الجدول، بالنسبة لحقل معين، مع استبعاد القيم التي تساوي NULL واستبعاد القيم المكررة.
- ✓ يستخدم الخيار * عندما نريد الحصول على عدد البيانات الموجودة في الجدول، بالنسبة لحقل معين، بما فيها البيانات ذات القيمة NULL. ولكن بدون كتابة اسم الحقل.

تمرين 1 : نفذ الاستعلام التالي في قاعدة البيانات Northwind ولاحظ النتائج

```
SELECT COUNT(*) , COUNT (DISTINCT CustomerID) , COUNT (ALL CustomerID)
FROM Orders
```

تمرين 2 : من قاعدة معطيات Northwind احسب عدد المناطق التي يوجد فيها زبائن بدون تكرارات

```
SELECT COUNT (DISTINCT Region) AS 'Region Count'
FROM Customers
```



تمرين 3 : من قاعدة معطيات Northwind احسب عدد المناطق التي يوجد فيها زبائن مع الحفاظ على التكرارات

```
SELECT COUNT (ALL Region) AS 'Region Count'  
FROM Customers
```

تمرين 4 : من قاعدة معطيات Northwind احسب عدد المناطق التي يوجد فيها زبائن بغض النظر عن وجود قيم صفرية NULL.

```
SELECT COUNT (*) AS 'Region Count'  
FROM Customers
```

تابع المجموع SUM

يحسب التابع SUM مجموع القيم في تعبير حسابي يتضمن عموداً أو أكثر.

الصيغة العامة لهذا التابع:

```
SELECT SUM (ALL|DISTINCT] column_name)  
FROM table_name;
```

تمرين 5 : أكتب استعلاماً لحساب مجموع المبيعات من جميع الكتب في قاعدة البيانات Pubs.

```
SELECT SUM(ytd_sales) 'YTD SALES'  
FROM titles;
```

تمرين 6 : جرب الاستعلام السابق بالصيغة التالية وقارن النتائج.

```
SELECT SUM( DISTINCT ytd_sales) 'YTD SALES'  
FROM titles;
```



تابع المتوسط الحسابي AVG

يحسب التابع AVG المتوسط الحسابي لتعبير حسابي يتضمن عموداً أو أكثر.

الصيغة العامة لهذا التابع:

```
SELECT AVG (ALL|DISTINCT] column_name)
FROM table_name;
```

تمرين 7 : أكتب استعلاماً لحساب متوسط أسعار الكتب في قاعدة البيانات Pubs فيما لو تم زيادة سعر كل كتاب بمقدار \$10.

```
SELECT AVG(price+10) 'AVG PRICE'
FROM titles;
```

تمرين 8 : لا تدخل القيم الصفرية في حساب المتوسط الحسابي، للتحقق من ذلك، نفذ الاستعلام التالي ولاحظ النتائج.

```
SELECT SUM(price)/COUNT(*) AS 'AVG WITH NULL',
       SUM(price)/COUNT(price) AS 'AVG WITHOUT NULL',
       AVG(price) AS 'AVG FUNC'
FROM titles;
```

تابع القيمة العظمى MAX، تابع القيمة الصغرى MIN

يحسب التابع MAX القيمة العظمى والتابع MIN القيمة الصغرى لتعبير حسابي يتضمن عموداً أو أكثر.

الصيغة العامة لهذه التوابع:

```
SELECT MAX ( column_name) , MIN(column_name)
FROM table_name;
```



تجدر الإشارة إلى عدم وجود أي تأثير للخيارين ALL و DISTINCT على التوابع MIN و MAX رغم أنه بالإمكان استخدامها. فالقيمة العظمى أو القيمة الصغرى لقيم حقل، تبقى نفسها، حتى ولو كان هناك تكرار في قيم الحقل وحتى ولو كان هناك قيم غير محددة (أي تساوي NULL).

تمرين 9: أكتب استعلاماً لإيجاد أعلى نسبة مبيعات، وأدنى قيمة مبيعات لكتاب في قاعدة البيانات Pubs.

```
SELECT MAX(ytd_sales) 'BEST BOOK', MIN(ytd_sales) 'LOWEST BOOK'  
FROM titles;
```

تجميع السجلات باستخدام توابع التجميع واللاحقة GROUP BY

عندما نتكلم عن توابع التجميع فلا بد لنا أن نتساءل: هل نستطيع أن نطبق هذه التوابع على مجموعات جزئية من السجلات بدلاً من تطبيقها على كامل السجلات؟

فإذا كان لدينا جدول منتجات، وأردنا حساب مجموع أسعار المنتجات التي نحصل عليها من المورد الأول، وحساب مجموع أسعار المنتجات التي نحصل عليها من المورد الثاني، وحساب مجموع أسعار المنتجات التي نحصل عليها من المورد الثالث،

فإننا سنحتاج لكتابة ثلاث تعليمات منفصلة تعتمد على التابع التجميعي SUM،

```
SELECT SUM(UnitPrice) FROM products WHERE supplierID = 1;
```

```
SELECT SUM(UnitPrice) FROM products WHERE supplierID = 2;
```

```
SELECT SUM(UnitPrice) FROM products WHERE supplierID = 3;
```

لكن هل يمكننا أن نصل إلى نفس النتيجة بتعليمة واحدة فقط؟



في الحقيقة تقدم لغة SQL إمكانية القيام بهذا الأمر عن طريق تحديد العمود (أو الأعمدة) التي نرغب بأن يتم التجميع وفقها وبذلك باستخدام اللاحقة GROUP BY وفق الصيغة العامة التالية:

```
SELECT column_name1[,column_name2,.....] , AGR_FUNC1(column_name1)[,
AGR_FUNC1(column_name2)....]
FROM table_name
WHERE condition
GROUP BY column_name1[,column_name2,.....]
HAVING condition
ORDER BY column_name1,.....;
```

✓ عند وجود تابع تجميع في تعليمة الاختيار SELECT فإن كل عمود موجود في تعليمة الاختيار إلى جانب تابع

التجميع يجب وضعه في فقرة الـ GROUP BY .

✓ لا يمكن أن يحوي تعبير التجميع GROUP BY على توابع تجميع.

✓ تفيد كلمة ALL في إعادة جميع المجموعات الناتجة بما في ذلك المجموعات الفارغة.

✓ غالباً ما تستخدم فقرة HAVING مع فقرة GROUP BY لتحديد شرط معين على المجموعات المختارة وهي

تشبه فقرة WHERE بالنسبة لتعليمة SELECT.

✓ الأسطر العائدة من التعليمة لا تكون في ترتيب محدد لذلك يفضل دوماً استخدام الترتيب ORDER BY

لتحديد الترتيب المرغوب..

تمرين 10 : أكتب استعلاماً يعرض مجموع أسعار المنتجات بحسب المزود من الجدول products.

```
SELECT supplierID, SUM (UnitPrice)
FROM products
WHERE supplierID IN (1,2,3)
GROUP BY supplierID ;
```



تمرين 11 : أكتب استعلاماً يعرض أعداد الموظفين اليذين تم تعيينهم بحسب العام من الجدول Employees في قاعدة البيانات Northwind.

```
SELECT DATEPART(yy,hiredate) AS 'YEAR', COUNT (*) AS 'HIRED COUNT'  
FROM Employees  
GROUP BY DATEPART(yy,hiredate) ;
```

تمرين 12 : أكتب استعلاماً لجلب الناشرين الذين تجاوزت مبيعاتهم 23000 من قاعدة البيانات Pubs.

```
SELECT pub_id,SUM(ytd_sales)  
FROM titles  
GROUP BY pub_id  
HAVING SUM(ytd_sales)>25000  
ORDER BY pub_id;
```

نهاية الجلسة