

لغة الاستعلامات الهيكلية

Structured Query Language

المحاضر: حسام يونسو

كلية العلوم - تخصص تقنية معلومات IT

- المخرجات المتوقعة من الدرس
 - مقدمة
 - الدوال في SQL
 - دوال التجميع (Aggregate Functions)
 - دوال النصوص (String Functions)
 - دوال التاريخ (Date Functions)
 - استخدام الدوال مع GROUP BY
 - استخدامات الدوال العملية.
- § الربط بين الجداول (JOINS)
 - § الخاتمة والتوصيات
 - § الخاتمة

المخرجات المتوقعة من الدرس

- فهم مفهوم الدوال في SQL، وأهميتها في تحليل البيانات ومعالجتها داخل قواعد البيانات.
 - تمييز أنواع الدوال المختلفة (التجميعية، النصية، التاريخية، الرياضية) ومعرفة متى يُستخدم كل نوع منها.
 - تطبيق دوال SQL على بيانات حقيقية ضمن استعلامات SELECT، بما يشمل:
 - استخدام GROUP BY مع الدوال التجميعية لتقسيم وتحليل البيانات حسب مجموعات منطقية مثل الدولة أو الموظف.
 - كتابة استعلامات SQL مركبة تجمع بين الدوال وجملَة JOIN لربط بيانات من عدة جداول داخل قاعدة بيانات
- علائقية.

في عالم تُحركه البيانات، أصبحت إدارة هذه البيانات بفعالية ضرورة لا غنى عنها لكل مؤسسة، سواء كانت حكومية أو خاصة.

وهنا تأتي أهمية لغة الاستعلامات الهيكلية (SQL)، فهي الأداة الأساسية للتفاعل مع قواعد البيانات العلائقية، تُمكننا من تخزين البيانات، استرجاعها، تحليلها، وتنظيمها بطريقة آمنة وفعالة.

خلال هذا العرض، سنغوص أكثر في أدوات SQL المتقدمة مثل الدوال النصية والرياضية والتجميعية، وكيف يمكننا ربط الجداول واستخدام GROUP BY، مما يساعد في بناء أنظمة معلومات قوية تخدم الأعمال وتُحسن اتخاذ القرار.



مقدمة إلى الدوال في SQL

الدوال في SQL – ما هي ولماذا نستخدمها؟

تعريف الدوال (Functions):

الدالة في SQL هي عملية مدمجة في لغة SQL (أي تم برمجتها مسبقًا من قبل نظام إدارة قاعدة البيانات)، تُستخدم لتحويل أو تحليل البيانات الموجودة في قاعدة البيانات. الدالة تأخذ مدخلًا (أو أكثر)، وتُعيد ناتجًا (Output) بناءً على هذا المدخل.



أنواع الدوال الرئيسية

❖ دوال التجميع (Aggregate Functions): تعالج عدة صفوف وتُعيد نتيجة واحدة.

❖ دوال النصوص (String Functions): تعالج النصوص مثل تحويل الحروف والدمج.

❖ دوال التاريخ (Date Functions): تعالج التواريخ مثل استخراج السنة أو الفارق الزمني.

❖ دوال رياضية (Mathematical Functions): تستخدم للعمليات الحسابية.



استخدامات الدوال في SQL

🔗 1. تحليل البيانات واستخلاص معلومات منها

الدوال تساعدنا في تلخيص وفهم البيانات. مثلاً، إذا أردت أن تعرف عدد الطلبات التي تم تنفيذها، تستخدم COUNT().

```
SELECT COUNT(*) FROM Orders;
```

🔗 2. تنسيق عرض النتائج

تُمكننا الدوال من تعديل طريقة عرض البيانات. مثلاً، لعرض الأسماء في تنسيق موحد أو تنسيق التواريخ.

```
SELECT CONCAT(FirstName, ' ', LastName) AS FullName FROM Employees;
```

استخدامات الدوال في SQL

④ 3. توفير الوقت وتقليل الحاجة إلى منطق برمجي خارجي

بدلاً من استخدام كود برمجي معقد خارج SQL (مثل في Java أو Python) لتحليل البيانات، يمكنك القيام بكل شيء داخل SQL باستخدام الدوال.

④ 4. تسهيل الفلتر والبحث ضمن شروط WHERE

بعض الدوال مثل LEN() أو YEAR() تُستخدم لتطبيق شروط بحث مرنة:

```
SELECT * FROM Orders WHERE YEAR(OrderDate) = 1998;
```


كيف تعمل الدوال؟

□ بعض الدوال تعمل على مستوى صف واحد فقط (وتسمى *scalar functions*)

مثل: LEN(), NOW(), UPPER().

□ بعضها تعمل على مجموعة صفوف دفعة واحدة (وتسمى *aggregate functions*)

مثل: AVG(), SUM(), COUNT().



أماكن تستخدم فيها الدوال

الدوال يمكن استخدامها تقريبًا في كل أجزاء الاستعلام:

```
SELECT
    EmployeeID,
    COUNT(OrderID) AS TotalOrders,
    AVG(Freight) AS AverageFreight
FROM Orders
WHERE YEAR(OrderDate) = 1997
GROUP BY EmployeeID
HAVING AVG(Freight) > 50
ORDER BY TotalOrders DESC;
```

• داخل SELECT

• داخل WHERE

• مع GROUP BY, HAVING

• حتى داخل ORDER BY

🔍 الشرح:

- حسبنا عدد الطلبات ومتوسط الشحن لكل موظف.
- فلترنا فقط الطلبات من سنة 1997.
- أظهرنا الموظفين الذين كان متوسط الشحن لديهم مرتفعًا.

دوال التجميع (Aggregate Functions)

ما هي دوال التجميع؟

دوال التجميع تُستخدم لمعالجة عدة صفوف من البيانات (مثل أعمدة كاملة)،
وتُعيد نتيجة واحدة تلخص تلك الصفوف.

مثال: لو لديك 100 طلب في جدول Orders، يمكنك استخدام دالة
SUM(Freight) لحساب إجمالي تكاليف الشحن لجميع هذه الطلبات.

أهم دوال التجميع في SQL

الدالة	الوظيفة	مثال توضيحي
COUNT()	تحسب عدد الصفوف أو القيم غير الفارغة	عدد الطلبات
SUM()	تجمع القيم الرقمية	مجموع الشحن
AVG()	تحسب المتوسط	متوسط سعر منتج
MIN()	أصغر قيمة	أرخص منتج
MAX()	أكبر قيمة	أغلى منتج

📌 1. عدد الطلبات في الجدول Orders: تعيد عدد كل عمليات الشراء في جدول الطلبات.

```
SELECT COUNT(*) AS TotalOrders  
FROM Orders;
```

📌 2. عدد العملاء الذين لديهم رقم فاكس: هنا نحسب عدد العملاء الذين لديهم قيمة (غير NULL) في عمود Fax.

```
SELECT COUNT(Fax) AS ClientsWithFax  
FROM Customers;
```

📌 3. أرخص منتج وأعلى منتج: تستخدم لمقارنة الأسعار أو تحليل السوق داخليًا.

```
SELECT MIN(UnitPrice) AS Cheapest, MAX(UnitPrice) AS MostExpensive  
FROM Products;
```

📌 4. متوسط سعر المنتجات: مفيد في معرفة تسعير المنتجات لدى الشركة.

```
SELECT AVG(UnitPrice) AS AveragePrice  
FROM Products;
```

دوال النصوص (String Functions)

ما المقصود بدوال النصوص؟

دوال النصوص (String Functions) هي أدوات في SQL تُستخدم لمعالجة الأعمدة النصية (مثل أسماء العملاء، العناوين، أسماء المنتجات).

تساعد هذه الدوال في:

- تنسيق النصوص (حروف كبيرة/صغيرة).
- قياس طول النص.
- دمج عدة أعمدة نصية في عمود واحد.
- استخراج أجزاء من النص.

أهم دوال النصوص في SQL

الدالة	الوظيفة
UPPER(str)	تحويل النص إلى حروف كبيرة
LOWER(str)	تحويل النص إلى حروف صغيرة
LEN(str) أو LENGTH(str)	عدد الأحرف في النص
CONCAT(str1, str2, ...)	دمج أكثر من نص
SUBSTRING(str, start, length)	استخراج جزء من النص

📌 1. تحويل أسماء الشركات إلى حروف كبيرة: مفيد لتوحيد التنسيق عند عرض أو مقارنة النصوص.

```
SELECT CompanyName, UPPER(CompanyName) AS Company_Upper  
FROM Customers;
```

📌 2. عرض طول أسماء المنتجات: يساعد في تحديد أسماء طويلة جدًا أو تحديد الحقول التي تحتاج إلى تنسيق.

```
SELECT ProductName, LENGTH(ProductName) AS NameLength  
FROM Products;
```

دوال التاريخ (Date Functions)

ما هي دوال التاريخ في SQL؟

دوال التاريخ (Date Functions) تُستخدم لمعالجة الأعمدة التي

تحتوي على تواريخ أو أوقات، مثل:

- استخراج أجزاء من التاريخ (سنة، شهر، يوم).
- حساب الفارق بين تاريخين.
- الحصول على التاريخ الحالي.

هذه الوظائف مفيدة جدًا في تتبع الطلبات، فترات التوصيل، التحليل الزمني، تقارير الأداء... إلخ.

الدالة	الوظيفة
GETDATE() أو NOW()	تُعيد التاريخ والوقت الحالي
YEAR(date)	تُعيد السنة من تاريخ معين
MONTH(date)	تُعيد رقم الشهر
DAY(date)	تُعيد اليوم من التاريخ
DATEDIFF(unit, date1, date2)	تحسب الفرق بين تاريخين
CURDATE()	MySQL (فقط) التاريخ الحالي بدون وقت)

📌 1. استخراج سنة الطلب من OrderDate: يساعدنا في تصنيف الطلبات حسب السنوات.

```
SELECT OrderID, OrderDate, YEAR(OrderDate) AS OrderYear  
FROM Orders;
```

📌 2. عرض الطلبات التي تم تنفيذها في السنة الحالية (مثال عام): يُستخدم لتقارير الأداء السنوية.

```
SELECT * FROM Orders  
WHERE YEAR(OrderDate) = YEAR(NOW());
```

استخدام الدوال مع GROUP BY

ما هي GROUP BY؟

GROUP BY هي عبارة تُستخدم في SQL لتجميع البيانات حسب قيمة واحدة أو أكثر، ثم تطبيق دوال التجميع مثل AVG(), SUM(), COUNT() على كل مجموعة بشكل منفصل.

```
SELECT عمود_التجميع, دالة_تجميع  
FROM اسم_الجدول  
GROUP BY عمود_التجميع;
```

✈ تخيل أن لدينا جدول طلبات يحتوي على 1000 طلب من 10 دول. إذا أردنا معرفة عدد الطلبات في كل دولة، نستخدم GROUP BY ShipCountry.

أمثلة على استخدام الدوال مع GROUP BY

📌 1. عدد الطلبات لكل دولة: تُعيد قائمة بالدول وعدد الطلبات في كل واحدة.

```
SELECT ShipCountry, COUNT(*) AS OrdersPerCountry  
FROM Orders  
GROUP BY ShipCountry;
```

📌 2. متوسط سعر المنتجات حسب الفئة: مفيد في معرفة الفروقات بين الفئات.

```
SELECT CategoryID, AVG(UnitPrice) AS AvgPrice  
FROM Products  
GROUP BY CategoryID;
```

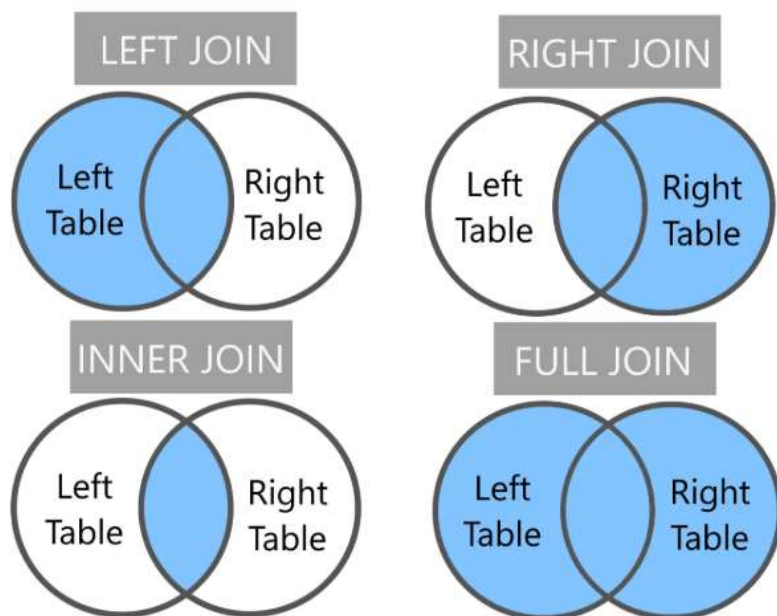
استخدام JOIN في SQL

ما هو JOIN؟

كلمة JOIN في SQL تُستخدم لربط جداول متعددة بناءً على علاقة منطقية بين الأعمدة المشتركة، غالبًا عبر مفتاح أساسي في جدول ومفتاح أجنبي في جدول آخر.

الشكل العام لاستخدام JOIN:

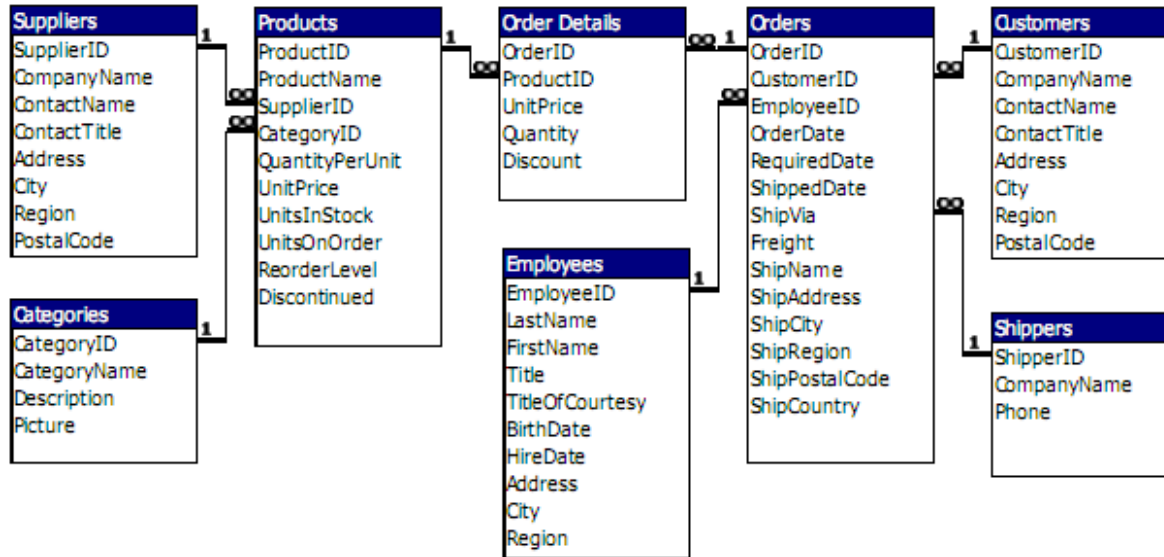
```
SELECT table1.column, table2.column  
FROM table1  
JOIN table2 ON table1.key = table2.key;
```



استخدام SELECT مع JOIN في SQL

في قاعدة Northwind لدينا علاقات كثيرة بين الجداول، مثل:

- Orders ↔ Customers: العميل له عدة طلبات.
- OrderDetails ↔ Orders: كل طلب يحتوي على تفاصيل منتجات.
- OrderDetails ↔ Products: كل منتج يمكن أن يوجد في عدة طلبات.

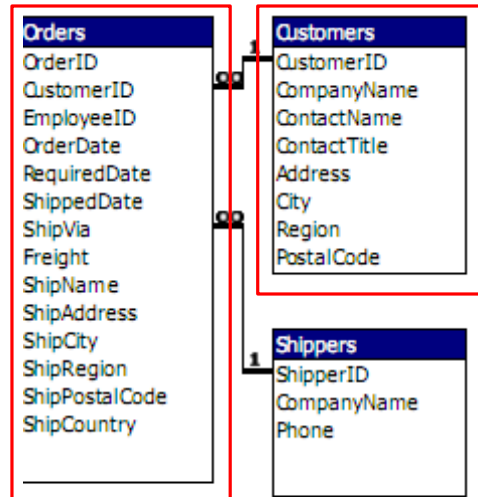


أنواع JOIN في SQL

النوع	ماذا يفعل؟
INNER JOIN	يعرض فقط الصفوف التي لها تطابق في كلا الجدولين.
LEFT JOIN	يعرض كل صفوف الجدول الأيسر، ويكمل من الجدول الأيمن إذا وُجد تطابق.
RIGHT JOIN	يعرض كل صفوف الجدول الأيمن، ويكمل من الجدول الأيسر إذا وُجد تطابق.

لماذا JOIN مهم جدًا؟

- لأنه يتيح لك العمل مع بيانات مترابطة من جداول مختلفة.
- من المستحيل عمل أنظمة حقيقية (مثل متجر أو نظام تسجيل) بدون ربط الجداول.



مثال عملي: ربط جدول Orders مع Customers

الهدف:

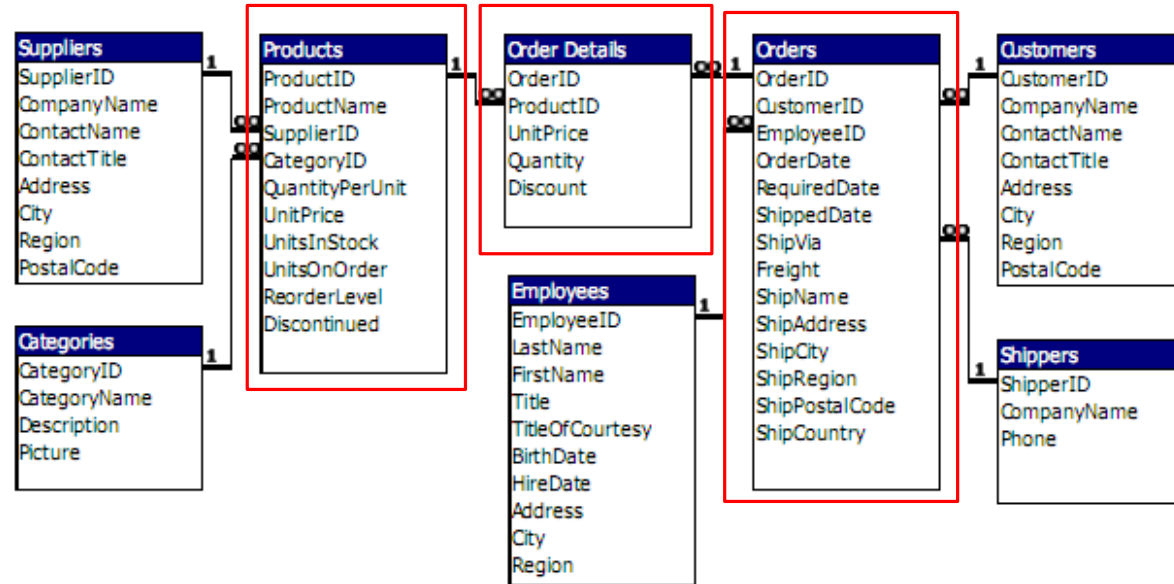
عرض رقم الطلب، اسم العميل، وتاريخ الطلب.

الشرح:

- نربط جدول Orders مع Customers باستخدام العمود المشترك CustomerID.
- نعرض رقم الطلب، اسم الشركة، وتاريخ الطلب.

```

SELECT Orders.OrderID, Customers.CompanyName, Orders.OrderDate
FROM Orders
JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
  
```



مثال ثاني: ربط 3 جداول معًا

الهدف:

عرض اسم المنتج، رقم الطلب، وكمية المنتج في الطلب.

الشرح:

- بدأنا من OrderDetails لأنه جدول العلاقة.
- ربطنا بـ Products لجلب اسم المنتج.
- ربطنا بـ Orders لجلب رقم الطلب.

```
SELECT Products.ProductName, Orders.OrderID, OrderDetails.Quantity
FROM OrderDetails
JOIN Products ON OrderDetails.ProductID = Products.ProductID
JOIN Orders ON OrderDetails.OrderID = Orders.OrderID;
```

مثال باستخدام LEFT JOIN:

الهدف:

عرض كل العملاء حتى لو لم يقدموا أي طلب.

🚩 **النتيجة:** العملاء الذين لا يملكون طلبات ستظهر لديهم OrderID ك NULL.

```
SELECT Customers.CompanyName, Orders.OrderID  
FROM Customers  
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

في ختام هذا الدرس، نؤكد أن إتقان الدوال في SQL يُعد خطوة أساسية لكل من يعمل أو يطمح للعمل في مجالات تحليل البيانات، تطوير قواعد البيانات، أو بناء الأنظمة المعلوماتية.

لقد تعرفنا على:

- دور الدوال في تحويل وتحليل البيانات.
- كيفية استخدام الدوال النصية والتجميعية والتاريخية.
- كيفية كتابة استعلامات فعّالة باستخدام GROUP BY و JOIN.

✈ توصيات للاستفادة القصوى من الدرس:

- طبق كل ما تعلمته عملياً باستخدام قاعدة بيانات مثل Northwind أو قاعدة بيانات خاصة بك.
- استخدم SQL Playground أو أدوات مثل phpMyAdmin أو SQLite Browser لاختبار استعلاماتك وتطوير مهاراتك.
- تمرّن على استعلامات واقعية: احسب تقارير، قارن أسعار، احلّ الطلبات... كما لو كنت تعمل على نظام حقيقي.
- احرص على كتابة استعلامات واضحة ومنظمة واستخدم ALIAS والتنسيق الجيد لتسهيل القراءة والفهم.
- تابع التعلم المستمر، فـ SQL مجال غني ويتكامل مع لغات وتقنيات كثيرة مثل Python و Power BI و Excel و R.

مؤلف الكتاب: المهندس حسام يونسو

روابط قد تفيد الطالب

• <https://www.w3schools.com>

شكراً لكم