**SQL Server مش بس SELECT و JOIN… 3 Features هتخليك تبص للداتا بيز بعين تانية!**

معظم الناس لما تسمع "SQL Server"، أول حاجة بتيجي في بالهم:  
SELECT \* FROM Table أو INNER JOIN Customers ON Orders.CustomerID = Customers.ID

بس الحقيقة؟  
SQL Server فيه Features قوية ممكن تفرق تمامًا في الأداء، الأمان، والتحليل… خصوصًا لو بتتعامل مع قواعد بيانات ضخمة أو سيستم إنتاجي.

النهارده هشارك معاك 3 Features متقدمة غالبًا مش بتستخدمها – أو مش واخد بالك من قوتها👇

**– Indexed Views1️⃣ لما الـ View يبقى أسرع من الجدول نفسه!**

لو عندك View بيتحسب دايمًا وفيه Aggregations تقيلة (زي SUM أو COUNT)،  
ممكن تحوّله لـ **Indexed View** يخزن النتيجة فعليًا زي جدول… وتكسب أداء رهيب!

CREATE VIEW SalesSummary

WITH SCHEMABINDING

AS

SELECT CustomerID, SUM(TotalAmount) AS Total

FROM dbo.Sales

GROUP BY CustomerID;

CREATE UNIQUE CLUSTERED INDEX idx\_sales\_summary

ON SalesSummary(CustomerID);

📌 أسرع بكتير في الـ Reports وQueries الثقيلة!

**–Query Store 2️⃣ جهاز تسجيل الأداء عندك**

لو حصل Drop في الأداء بعد نشر جديد ومش عارف السبب؟  
Query Store هو الـ **Black Box** بتاعتك!  
بيخزن Execution Plans لكل Query وبيساعدك ترجع للـ Plan القديم لو حصل Regression.

ALTER DATABASE MyDB

SET QUERY\_STORE = ON;

🎯 مثالي في Debug الأداء بعد أي تغييرات.

**– Temporal Tables 3️⃣ التاريخ محفوظ تلقائيًا**

عايز ترجع للداتا قبل ما تتعدل؟ أو تعمل Audit بدون ما تبني System معقد؟  
SQL Server بيسهّلك ده بـ **Temporal Tables**:

CREATE TABLE Employees

(

ID INT PRIMARY KEY,

Name NVARCHAR(100),

Position NVARCHAR(100),

SysStartTime DATETIME2 GENERATED ALWAYS AS ROW START,

SysEndTime DATETIME2 GENERATED ALWAYS AS ROW END,

PERIOD FOR SYSTEM\_TIME (SysStartTime, SysEndTime)

)

WITH (SYSTEM\_VERSIONING = ON);

⏳ كل تغيير محفوظ تلقائيًا… تقدر ترجع لأي لحظة في التاريخ بدون مجهود!

**💡 الخلاصة**

SQL Server مش بس أداة لتخزين البيانات…  
هو منصة كاملة لو استغلتها صح، هتوفر وقت، ترفع الأداء، وتحسّن تجربة المستخدم.

✅ جرّب الحاجات دي في مشروعك الجاي  
✅ ولو عندك Feature مفضّلة في SQL Server… شاركني في التعليقات!

#SQLServer #Database #Performance #QueryOptimization #DataEngineering #MicrosoftSQL