

الفرق بين Eager و Deferred Execution

Execution: الفهم العميق واستخدامات عملية في الـ Business

في عالم البرمجة وخصوصاً مع الـ LINQ في .NET، يظهرون مفهومين مهمين:

• Deferred Execution

• Eager Execution

الآن لنرى تأثير مباشر على أداء البرامج وعلى الـ business use cases التي الشركات تعتمد عليها.

1. ما هو Deferred Execution؟

- الكود يكتب، لكن مش بيتنفذ غير وقت استدعاء النتيجة.
 - الاستعلام (Query) بيكون جاهز، لكن التنفيذ بيحصل عند الطلب (On-demand).
- مثال: زي ما تكتب طلب شراء في شركة، لكنه مش يتنفذ غير لما العميل يطلب المنتج فعلاً. 📌

2. ما هو Eager Execution؟

- الكود بيتنفذ مباشرة أول ما يكتب.
 - النتيجة بتتخزن وتتخزن في الذاكرة فوراً.
- مثال: زي ما تحضر مخزون منتج وتخزنه قبل ما العميل يطلبه. 📌
-

3. مقارنة سريعة

| العنصر | Deferred Execution | Eager Execution |
|-----------------|--|------------------------------------|
| وقت التنفيذ | عند الطلب | فوراً |
| الأداء | أفضل مع البيانات الكبيرة أو queries المتغيرة | أسرع مع النتائج الصغيرة أو الثابتة |
| استهلاك الذاكرة | أقل | أعلى |
| المرونة | عالي | محدود |

4. أمثلة عملية من الـ Business

أ. E-commerce

- **Deferred Execution**: فلترة المنتجات (سعر، ماركة، تقييم) → الاستعلام يتنفذ فقط عند ضغط "عرض النتائج".
- **Eager Execution**: قائمة "الأكثر مبيعاً" → تتخزن مسبقاً (Cache) لسرعة العرض.

ب. Banking Systems

- **Deferred Execution**: تقارير يومية عن حركة الحسابات → تتولد عند الطلب.
- **Eager Execution**: الرصيد المتاح يتحدث فوراً بعد أي عملية.

ج. Healthcare Systems

- **Deferred Execution**: استعلام معقد عن بيانات مريض → يتنفذ فقط وقت الحاجة.
- **Eager Execution**: نتائج فحوصات حرجة تظهر فور توفرها.

5. إزاي تختار بينهم؟

استخدم **Deferred Execution** لما: 

- البيانات كبيرة.
- الشروط ممكن تتغير قبل التنفيذ.
- عايز تقلل استهلاك الذاكرة.

استخدم **Eager Execution** لما: 

- النتيجة صغيرة أو ثابتة.
- محتاج سرعة استجابة فورية.
- النتيجة هتستخدم أكثر من مرة.

الخلاصة

الفرق بين **Deferred Execution** و **Eager Execution** مش مجرد فرق تقني، لكنه قرار استراتيجي بيأثر على:

- أداء النظام
- تجربة المستخدم
- تكلفة البنية التحتية