



ASP .NET Core 3.0 {پیشرفته}

 Visual Studio 2019

مدرس: علیرضا ارومند
nikamooz;
آموزش برنامه نویسی و اجرای پروژه

www.NikAmooz.com

جلسه سوم

معرفی علیرضا ارومند

۱. مدرس و مشاور ASP.NET Core و معماری‌های نرم‌افزاری (نیک آموز)
۲. مدیر فنی خبرگزاری نسیم
۳. کارشناس ارشد توسعه نرم افزار داتین (فناپ)
۴. کارشناس ارشد توسعه نرم افزار ارتباط فردا (بانک آینده)
۵. متخصص انجام پروژه‌های وب و .NET
۶. و ...



Introducing DAL

مدرس: علیرضا ارومند

nikamooz;
آموزش برنامه نویسی و اجرای پروژه

چه خواهیم آموخت؟

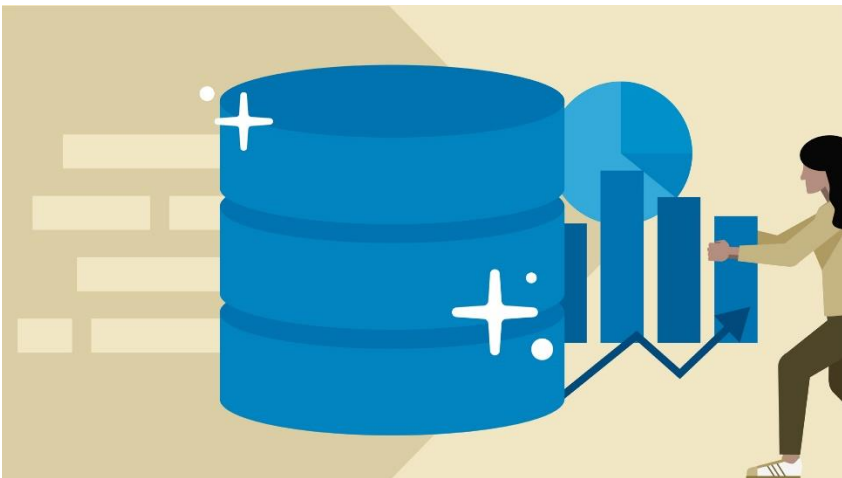
۱. چرا داده‌ها اهمیت دارند؟
۲. آشنایی با Enterprise App
۳. ابزارهای مورد نیاز چیست؟
۴. ...



اهمیت داده‌ها و مدیریت آن‌ها

۱. استفاده برای تصمیمات مهم
۲. استفاده برای تعیین اهداف سازمان
۳. نیاز به سرعت و دقت در دنیای رقابتی





چراهای همیشگی!

۱. چند مشتری داریم؟!
۲. آخرین بار مشتری X چه کالایی خریداری کرده؟
۳. پربازدید ترین گروه خبری چیست؟
۴. موجود حساب‌های من چقدر است؟

دسته‌بندی نرم افزارها

۱. هر گروه برنامه ویژگی‌هایی دارند
۲. مدیریت سوخت هواپیما!
۳. کارکرد پرینتر با سیستم عامل!



Enterprise Application ها

۱. مسئولیت اصلی کار با داده‌ها

۲. حقوق و دستمزد، حسابداری، فروشگاه و ...



دهه شصتی‌ها

۱. حداکثر حجم داده‌ها چند گیگ
۲. تعداد کاربران محدود
۳. همزمانی بسیار کم و محدود

There's usually a lot of data. a moderate system will have over 1 GB of data organized in tens of millions of records.so much that managing it is a major part of the system.

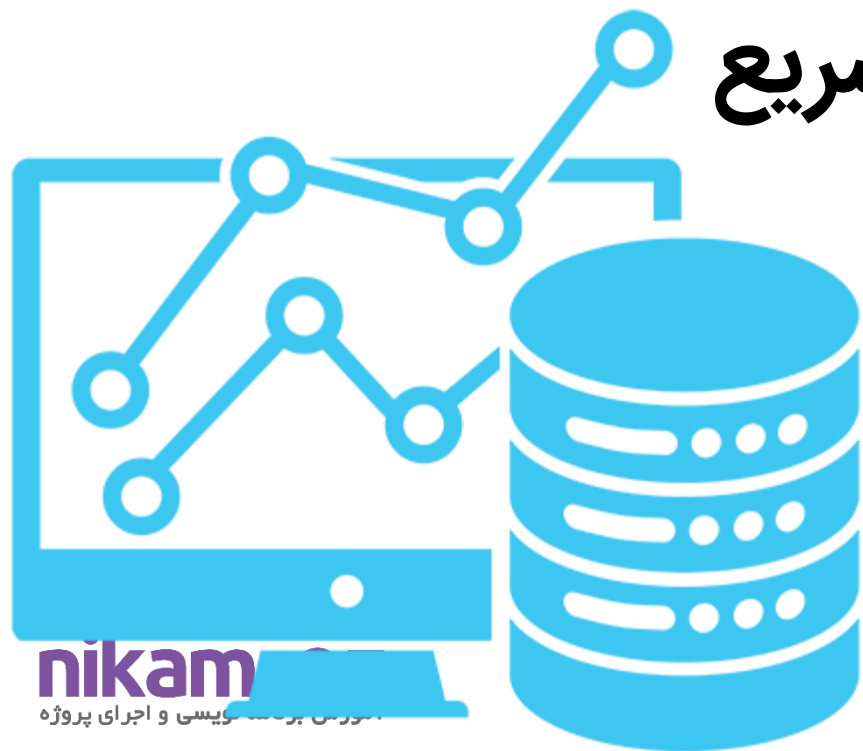
Martin fowler

گودزیلاهای دهه نودی

۱. انفجادهای حجم داده‌ها

۲. تعداد بسیار زیاد کاربران

۳. سرعت رشد به شدت سریع



انواع برنامه‌های Enterprise



۱. اطلاعات کوچک با حجم زیاد یا برعکس
۲. تعداد تراکنش زیاد
۳. تعداد کاربر همزمان بسیار زیاد

لایه‌های تشکیل دهنده

۱. خروجی‌های برنامه (تعامل با سایرین)

۲. منطق برنامه

۳. ذخیره و بازیابی اطلاعات



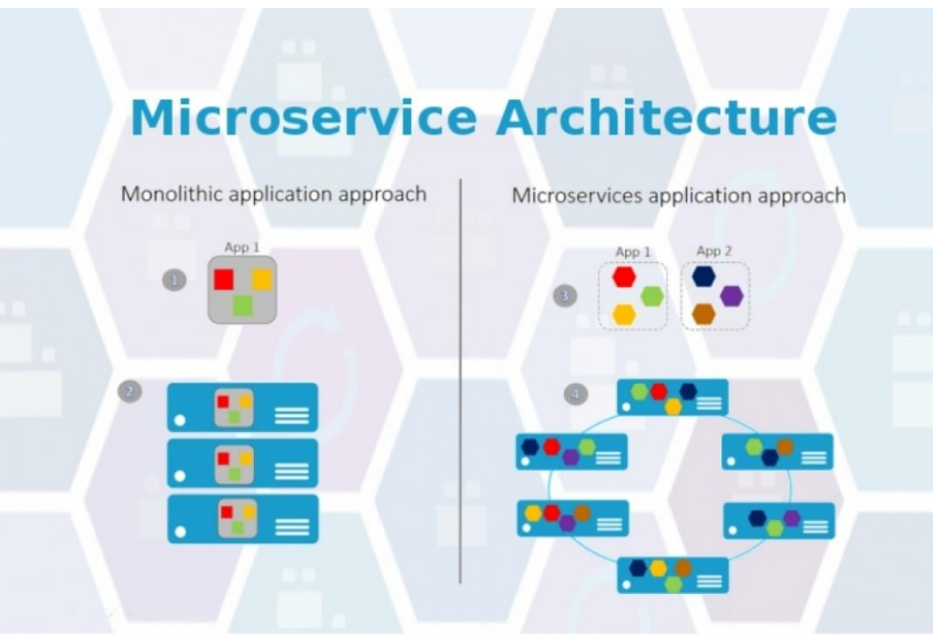
بازخوانی یک پرونده

۱. روش سنتی و استفاده از Database First
۲. استفاده از روش‌های سنتی دسترسی به داده
۳. نهایت پیشرفت استفاده از ORM



توسعه مدرن نرم افزار

۱. توسعه میکروسرویس ها
۲. معماری و ساختار دهی اختصاصی سرویس ها
۳. انتخاب ابزار مناسب برای هر کار



Polyglot Programming

۱. استفاده از چندین زبان برنامه نویسی در توسعه
۲. نیازمندی تعیین کننده زبان و ابزارهای توسعه



Polyglot Persistence

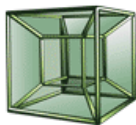
۱. استفاده از چندین Db engine

APACHE
HBASE

 **Cassandra**


CouchDB
relax

 **riak**



HYPERTABLE INC



mongoDB



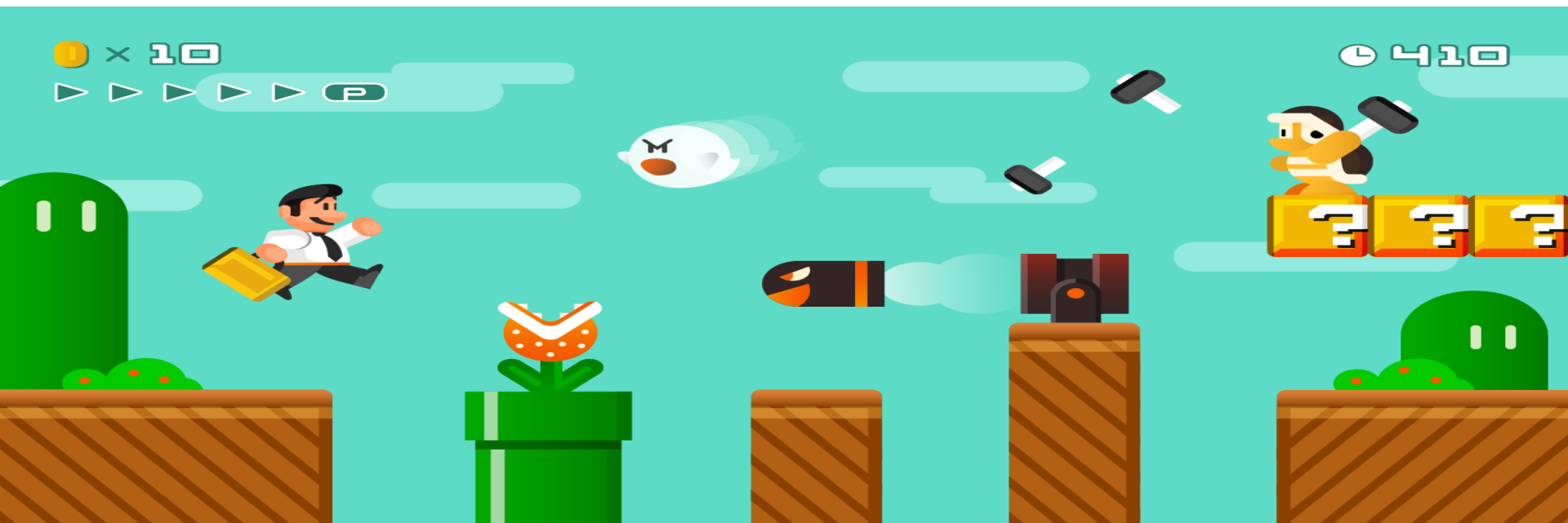
Neo4j



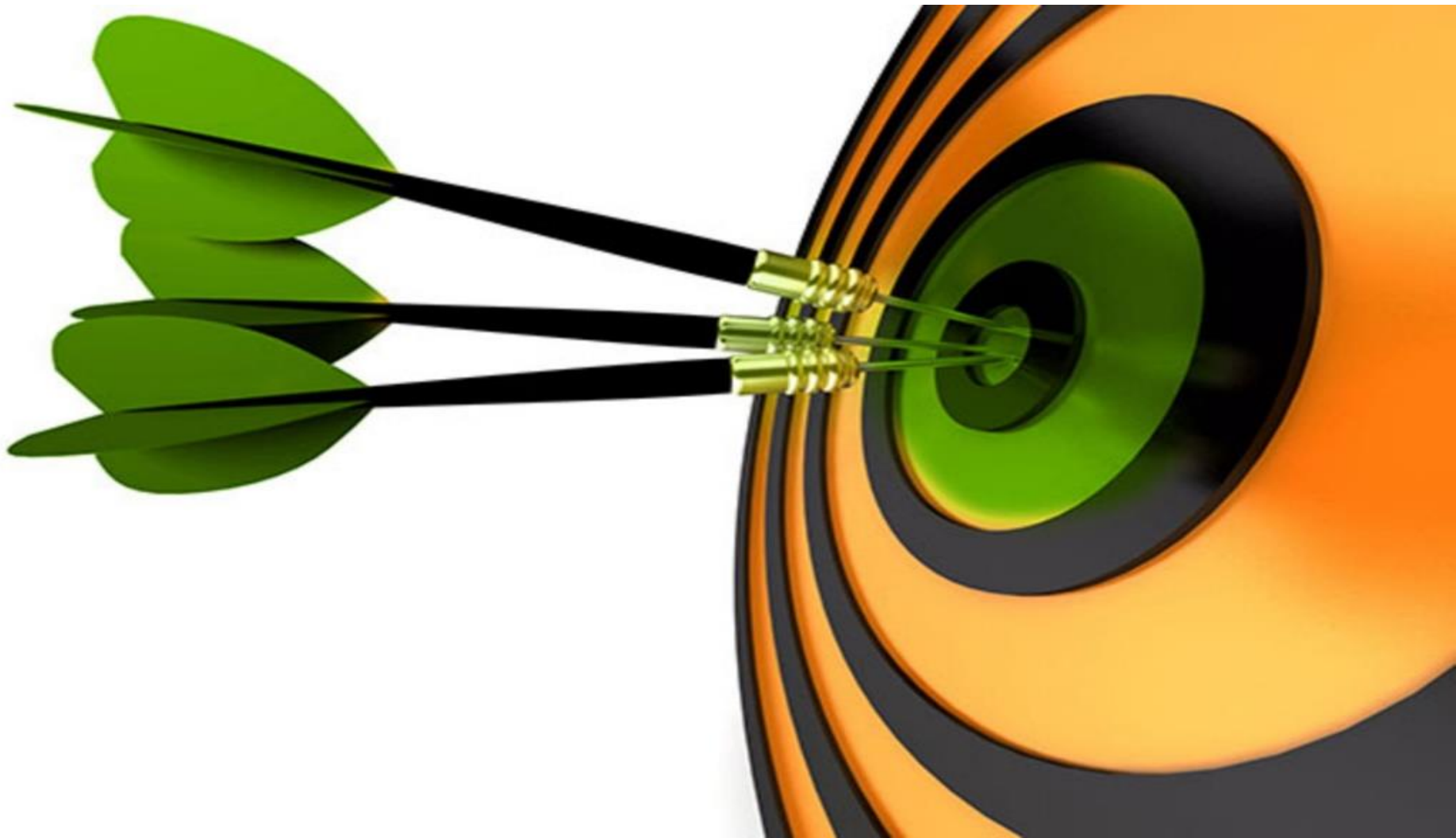
redis

چالش‌های دسترسی به داده‌ها

۱. چه ابزاری چه زمانی مناسب است؟
۲. پیچیدگی‌های وارد به برنامه چیست؟
۳. چگونگی مدیریت پیچیدگی‌ها؟



پله پله تا هدف



آشنایی با Entity Framework Core 3

۱. آشنایی با EF Core

۲. بررسی ساختار و اجزا تشکیل دهنده EF

۳. معرفی الگوهای پرکاربرد

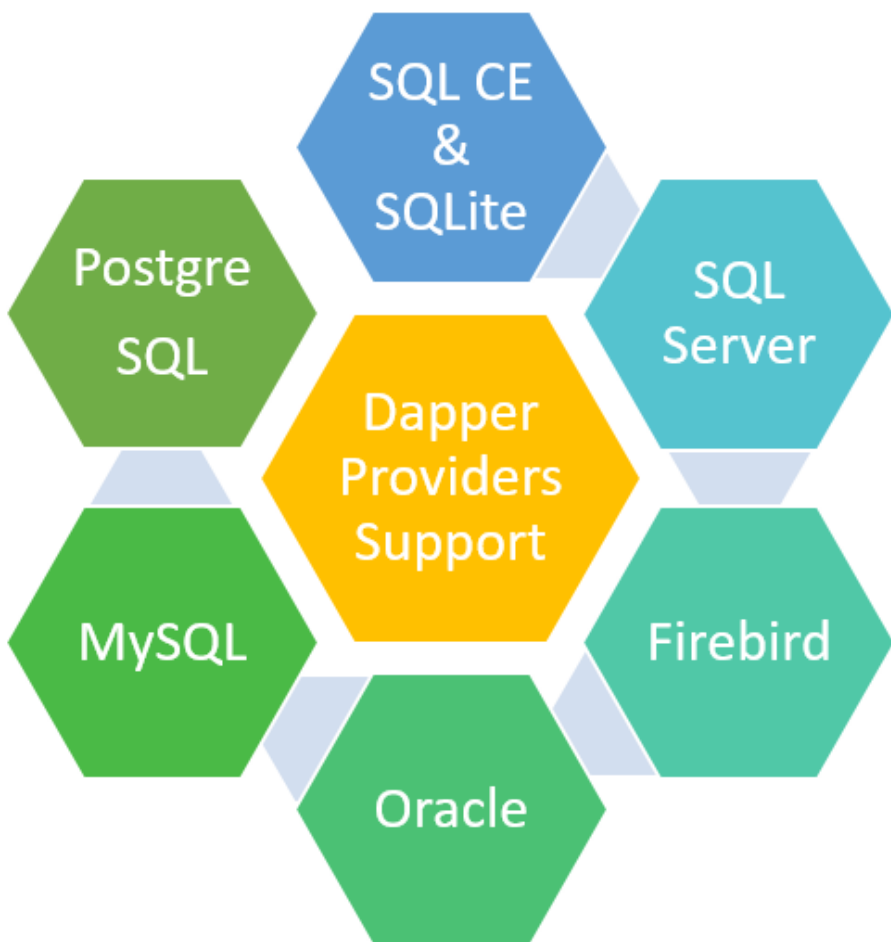
۴. آشنایی با بهینه سازی DAL

۵. تست لایه DAL

Entity Framework



Dapper



۱. Micro ORM چیست؟

۲. آشنایی با Dapper

خارج از سرفصل

۱. NoSQL چیست؟

۲. معرفی خانواده‌های مختلف NoSQL ها

۳. Redis چیست؟



redis

Introduction To Entity Framework Core

مدرس: علیرضا ارومند

nikamooz;
آموزش برنامه نویسی
آموزش برنامه نویسی و اجرای پروژه

چه خواهیم آموخت؟

۱. Entity Framework چیست؟

۲. ساختار Entity Framework

۳. از EF Core استفاده کنیم یا نه؟

۴. ...



اهمیت داده‌ها و مدیریت آن‌ها

۱. ORM اختصاصی مایکروسافت
۲. از نظر ساختاری ادامه نسخه 6.X
۳. تغییرات ساختاری و عملگری



EF Core اتصال دنیاها

دیتابیس	برنامه نویسی
جدول	.NET Class
ستون های جدول	Properties/fields
ردیف های جدول	لیستی از کلاس
Primary Key/Unique Key	وهله ای از کلاس
Foreign Key	ارتباط بین کلاس ها
دستورات SQL	دستورات LINQ

نیمه خالی لیوان



Which half are you?

۱. کار با ORM ها ساده است

۲. جادوگری همیشه به نفع ما نیست

۳. تفاوت دو دنیا کار توسعه را سخت می کند

۴. مشکلات Performance

جایگاه NoSql ها

۱. در دنیای واقعی در کنار هم هستند
۲. در نسخه ۶.۰ پشتیبانی نمی شد
۳. طراحی برای پشتیبانی از هر دو نوع



Cassandra

mongoDB

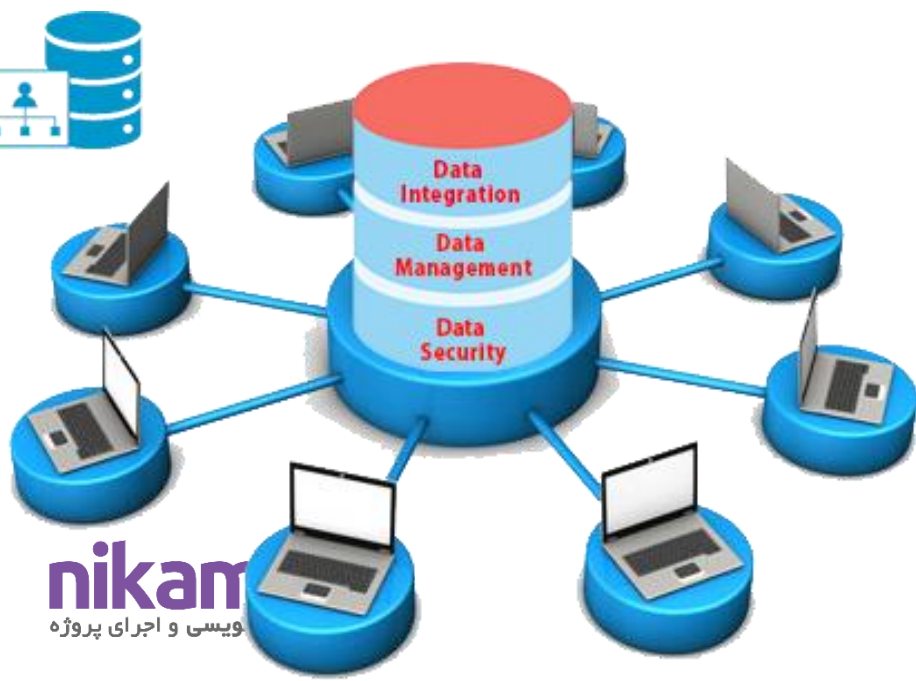


membase



دیتابیس من کجاست؟

۱. ایجاد دیتابیس توسط EF
۲. استفاده از دیتابیس موجود
۳. خداحافظ EDMX



آشنایی با DbContext

۱. نماینده دیتابیس

۲. ارث بری از کلاس DbContext

۳. معرفی جداول به کمک DbSet ها



Convention ها

۱. تنظیمات جداول از کجا می آیند؟

۲. اصل Convention Over Configuration

۳. تنظیمات به کمک Fluent API

۴. تنظیمات به کمک Attribute ها



بررسی روند اجرا

۱. عدم نیاز هنگام کاربرد عادی

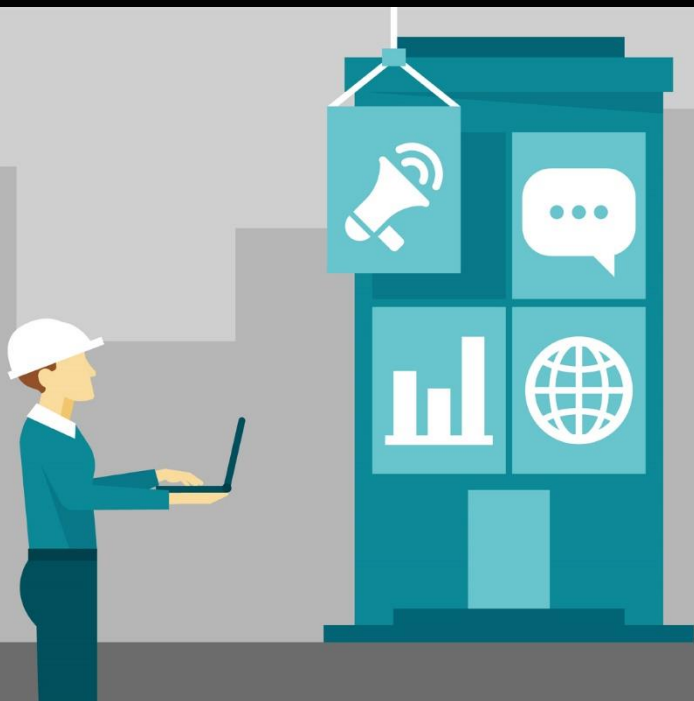
۲. کمک به فهم بهتر عملکرد

۳. یادگیری بهتر

۴. استفاده دقیق تر



ساخت مدل

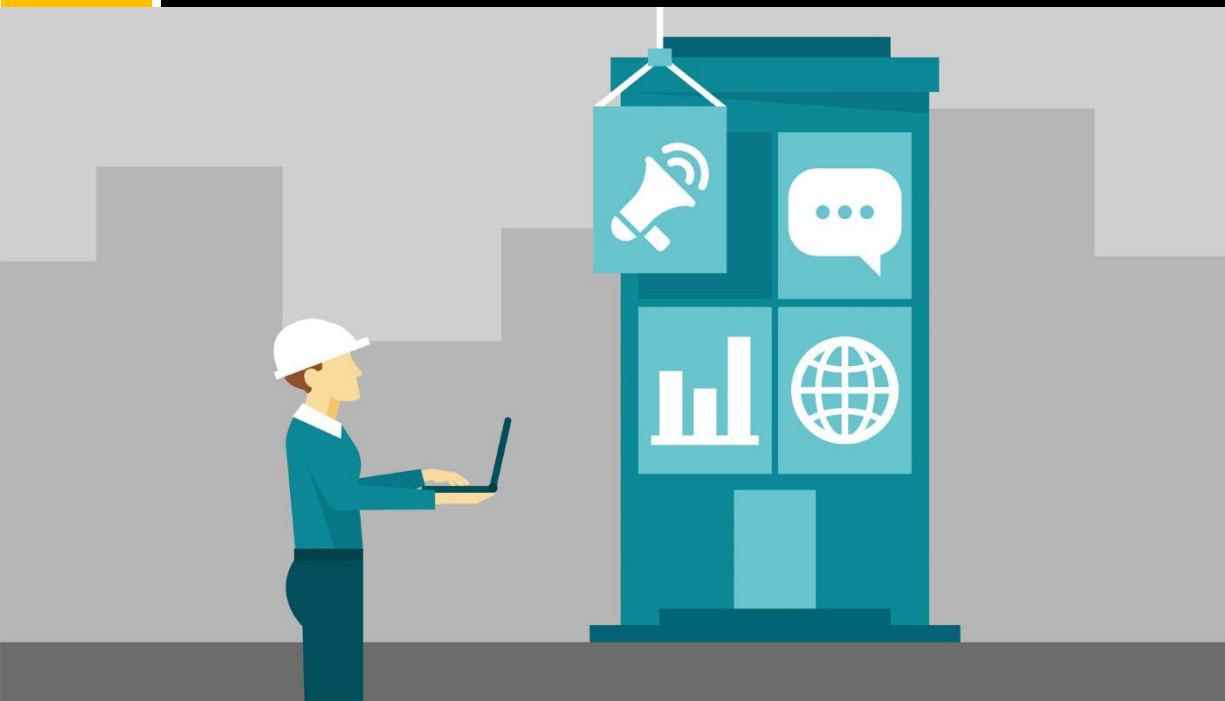


۱. شروع با نمونه سازی DbContext

۲. شناسایی تمامی DbSet ها

۳. شناسایی روابط داخلی

ساخت مدل



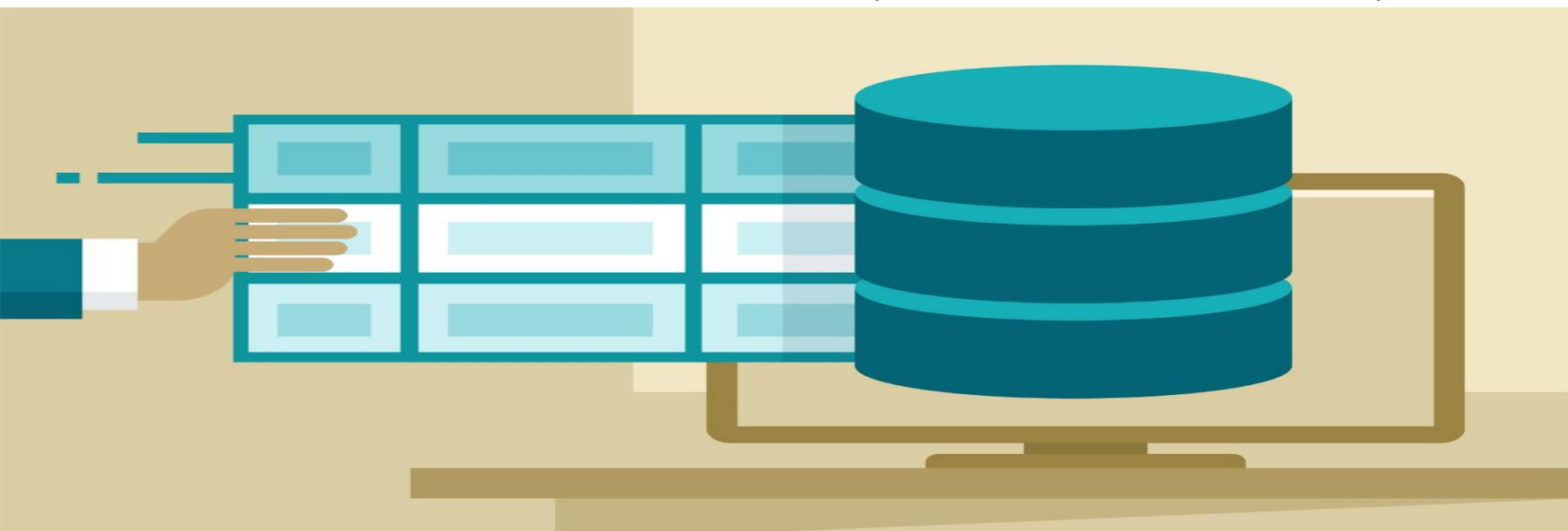
۴. اجرای OnModelCreating

۵. ایجاد مدل دیتابیس

۶. Cache کردن مدل برای استفاده‌های آینده

واکشی داده‌ها

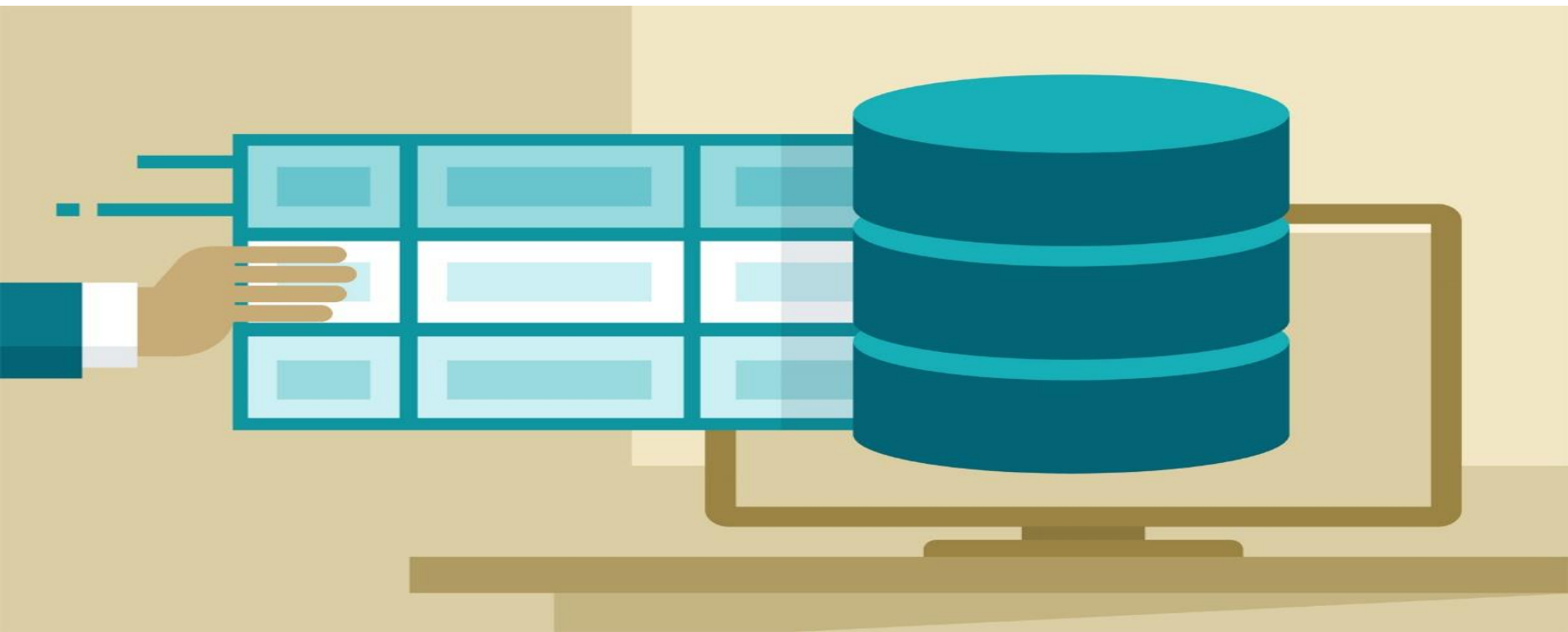
۱. استفاده از LINQ برای خواندن
۲. ترجمه دستور و Cache کردن دستور
۳. واکشی اطلاعات و تبدیل به کلاس



واکشی داده‌ها

۴. انجام عملیات Relationship fixup

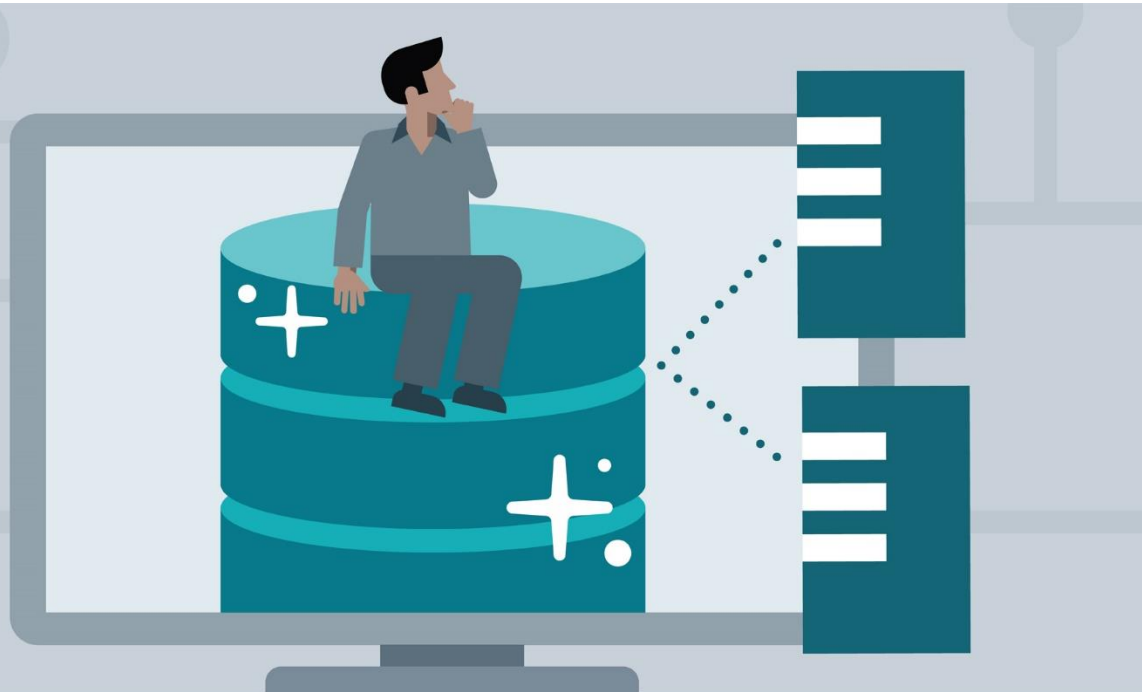
۵. ایجاد Tracking Snapshot



به روز رسانی اطلاعات

۱. یافتن موارد برای بروز رسانی

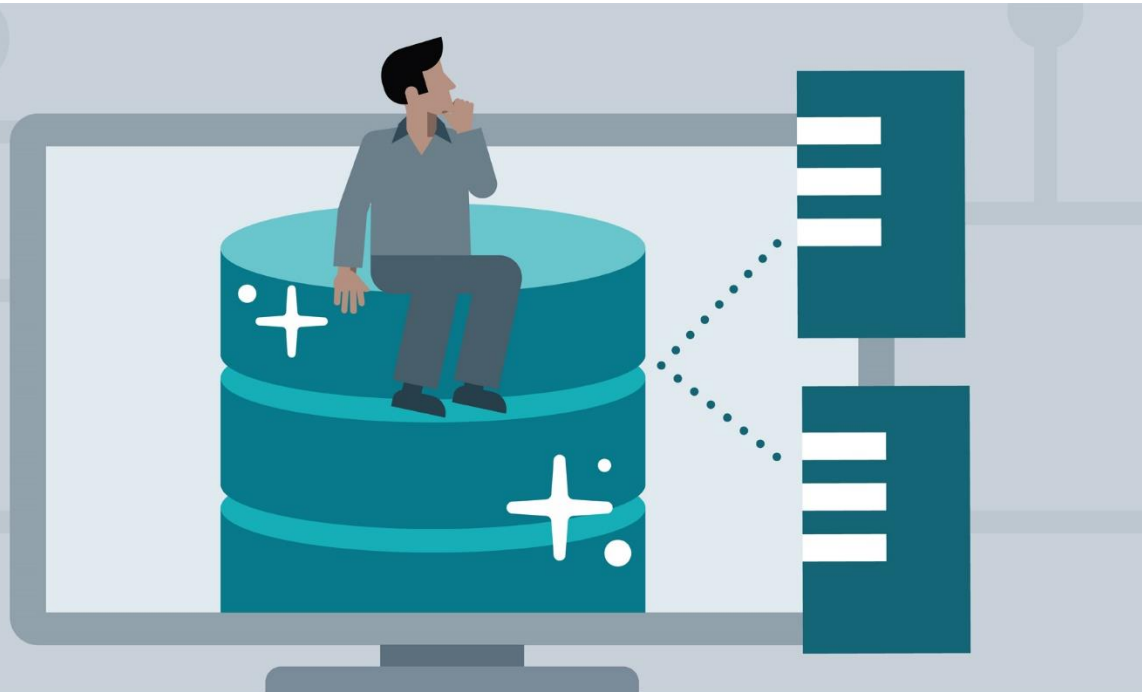
۲. ایجاد Tracking snapshot



به روز رسانی اطلاعات

۳. ایجاد تغییرات مورد نظر

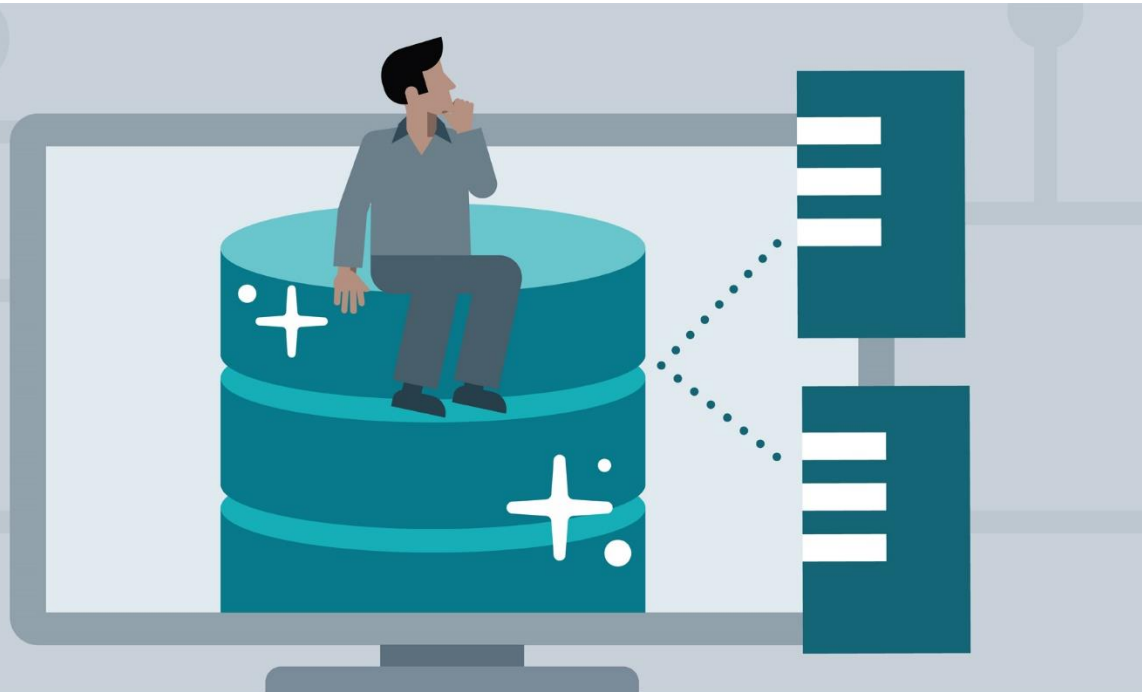
۴. ذخیره اطلاعات با کمک `SaveChanges`



به روزرسانی اطلاعات

۵. اجرای متد Detect changes

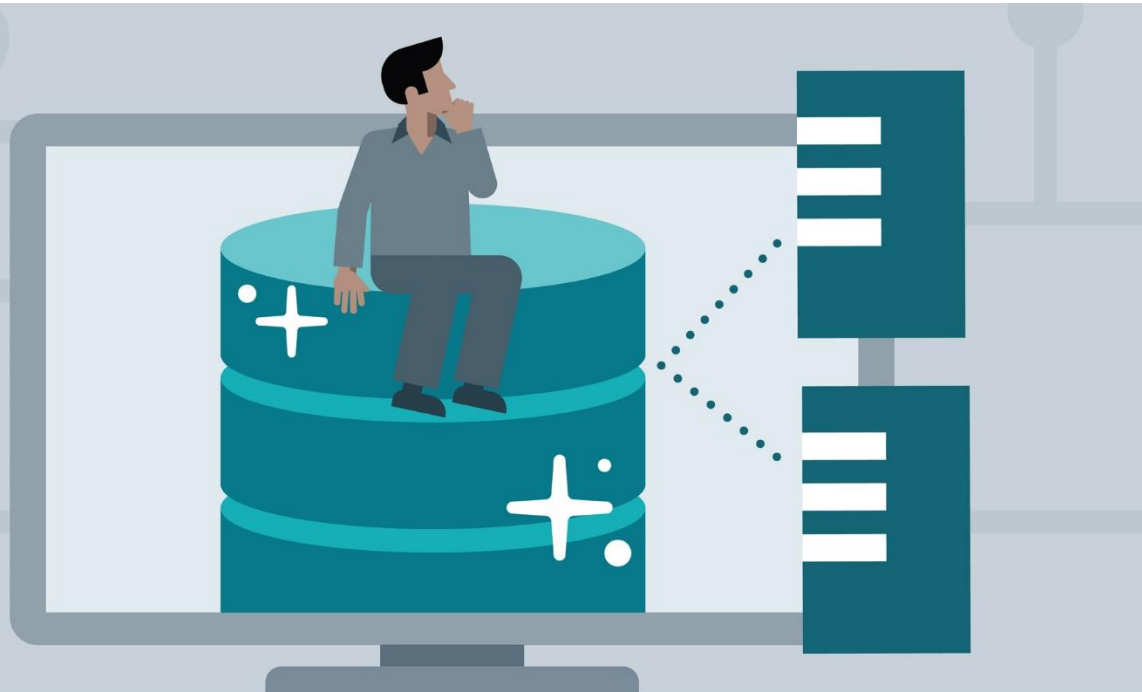
۶. شروع یک تراکنش (همه یا هیچ)



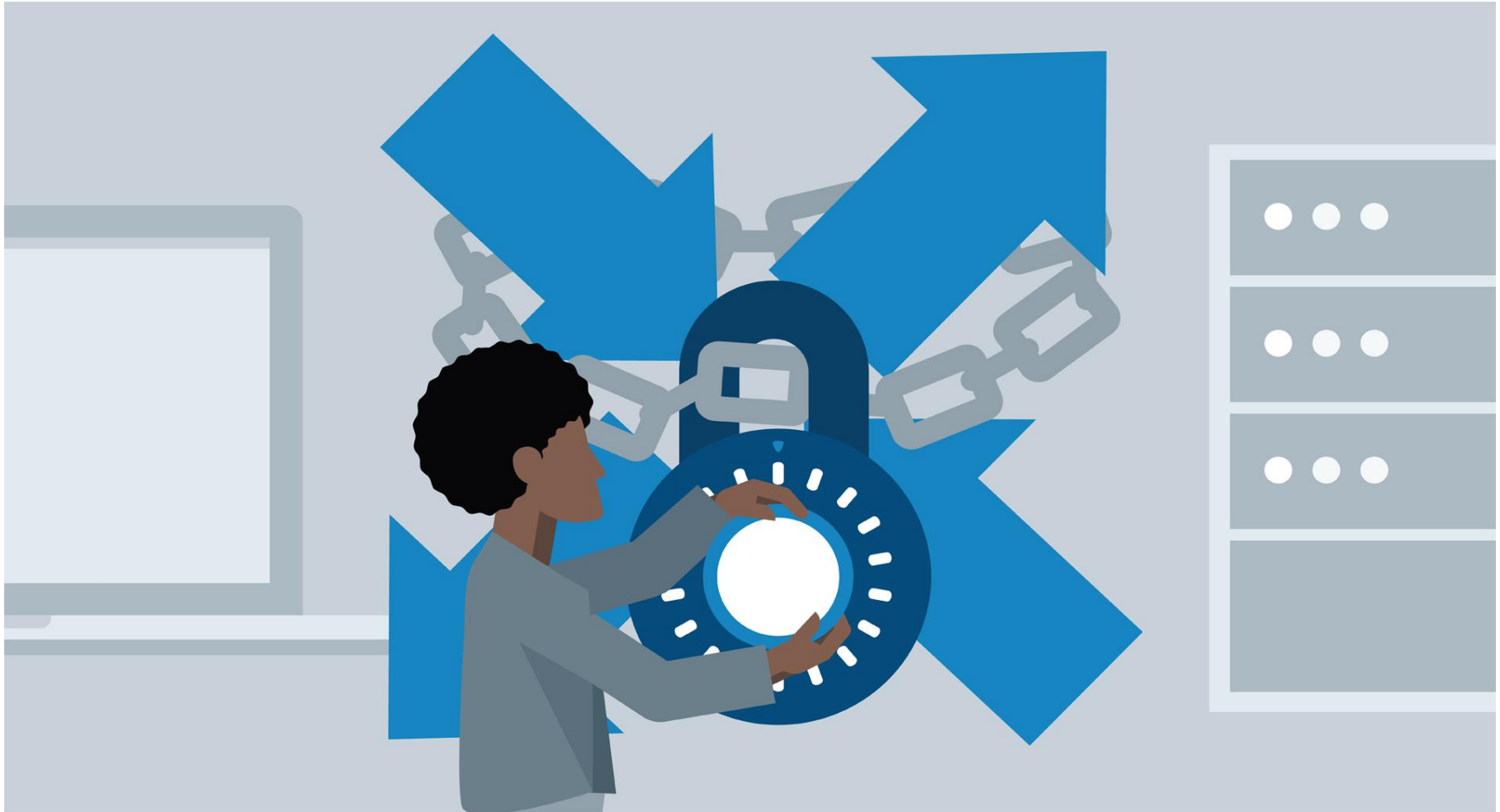
به روز رسانی اطلاعات

۷. تبدیل دستور Update به فرمت صحیح

۸. Commit در صورت صحت انجام کار



آیا از EF Core استفاده کنم؟



مزایای EF Core

۱. استفاده از آخرین تکنولوژی
۲. پشتیبانی از پلت فرم‌های مختلف
۳. توسعه سریع و ساده
۴. مستندات و آموزش‌های خوب



مزایای EF Core

۵. پشتیبانی از اکثر بسته های Nuget

۶. یک ORM کامل

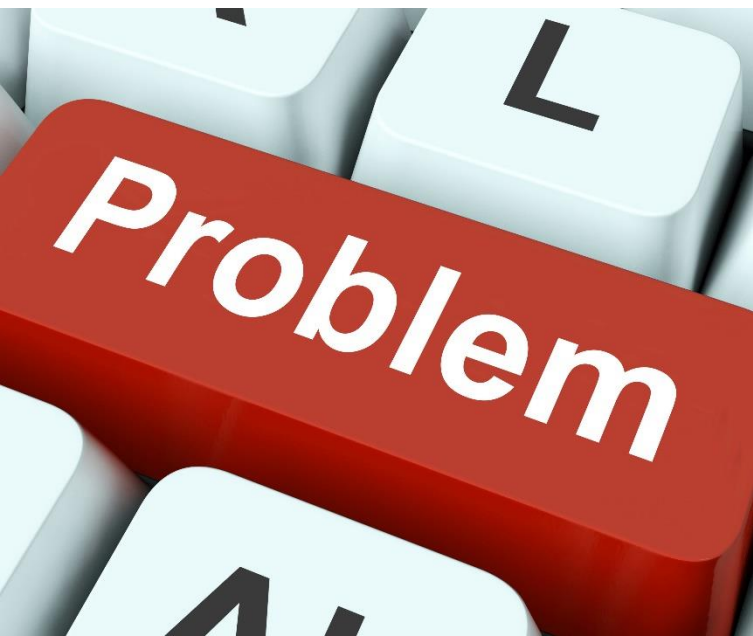
۷. کتابخانه با ثبات و قابل اطمینان

۸. پرفورمنس بسیار بالا



چرا از EF Core استفاده نمی کنم؟

۱. عدم پشتیبانی از DB Engine
۲. نیاز به پرفورمنس فوق العاده بالا
۳. نیاز ترکیبی و هیبرید Data Access



شبکه‌های اجتماعی نیک آموز

اطلاع رسانی سریع کارگاه‌های نسبتاً رایگان،

کوپن‌های تخفیف، مقالات، فیلم و دوره‌های نیک آموز



Instagram



Telegram

