Introduction to Object-Relational Mapping for DBAs

Chris Cumming
Saturday Morning Productions

What is Object-Relational Mapping (ORM)?

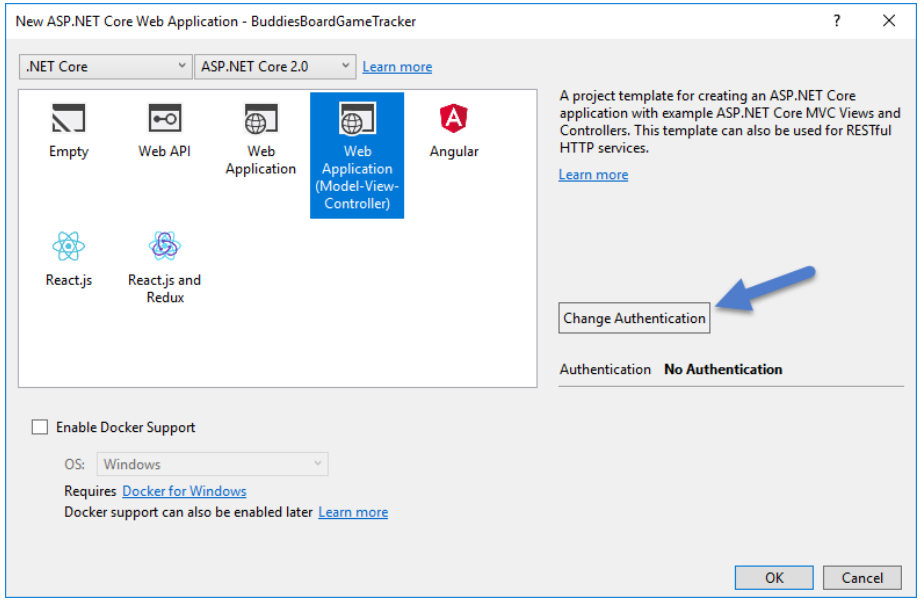
Wikipedia Definition:

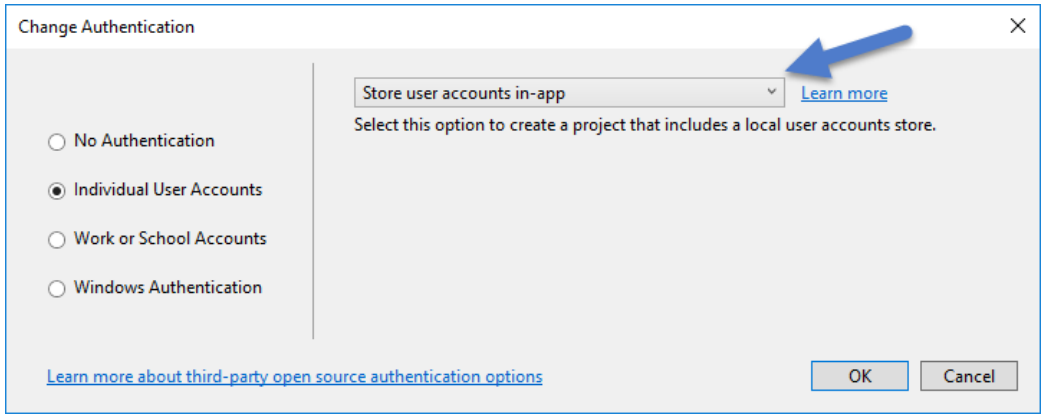
Object-relational mapping (ORM, O/RM, and O/R mapping tool) in computer science is a programming technique for converting data between incompatible type systems using object-oriented programming languages. This creates, in effect, a "virtual object database" that can be used from within the programming language. There are both free and commercial packages available that perform object-relational mapping, although some programmers opt to construct their own ORM tools.

My Definition:

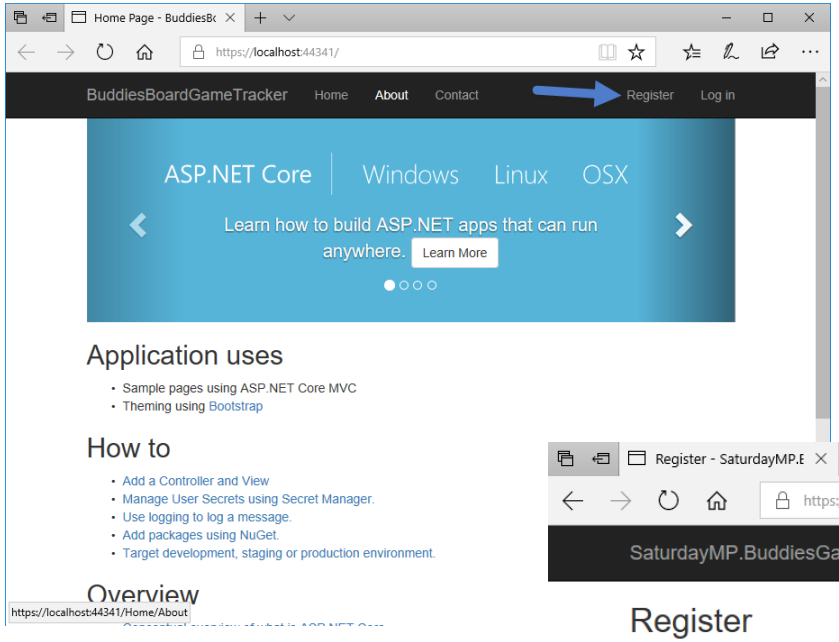
Maps database tables to classes.

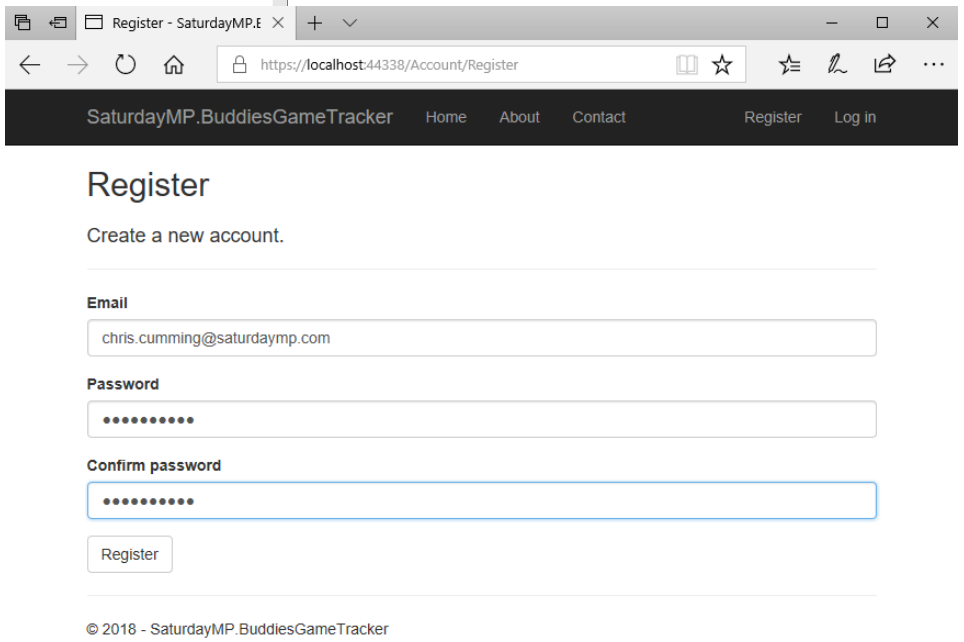
Bud the developer creates a new application with authentication



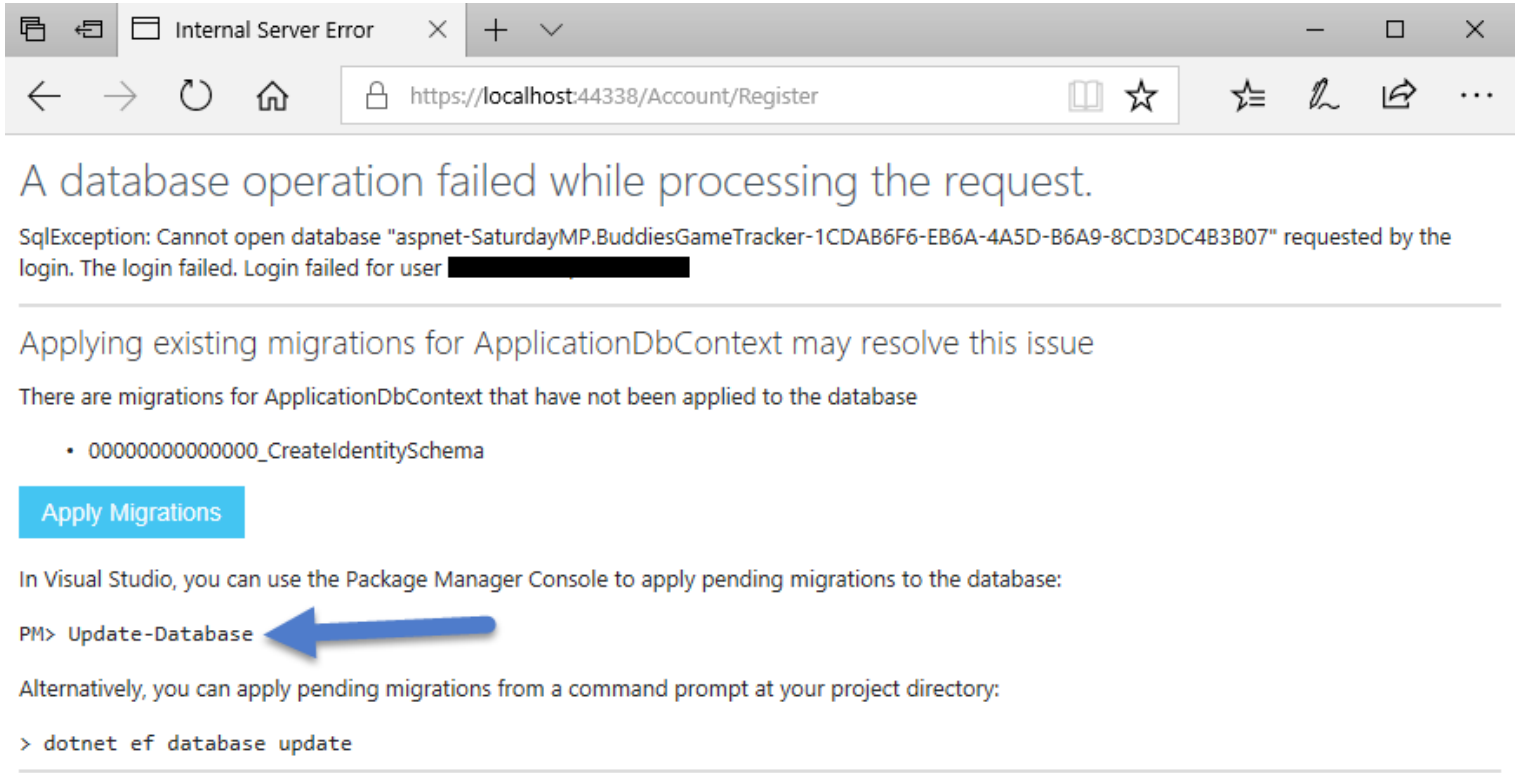


Bud runs the application



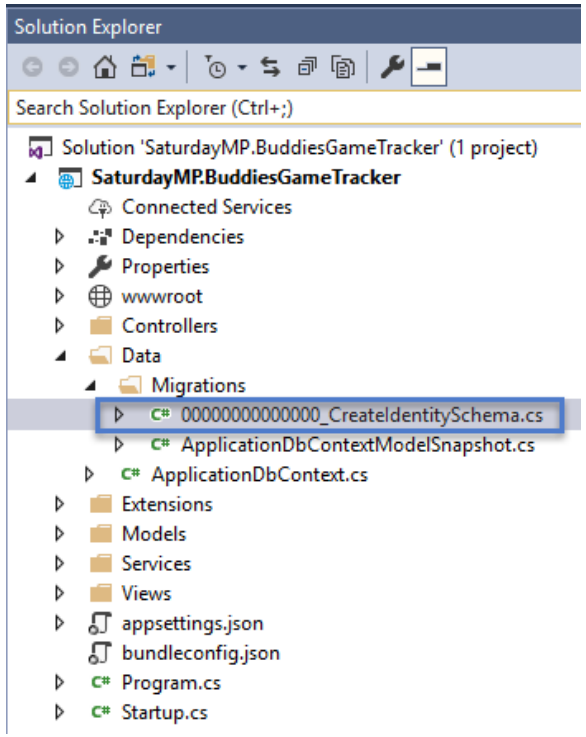


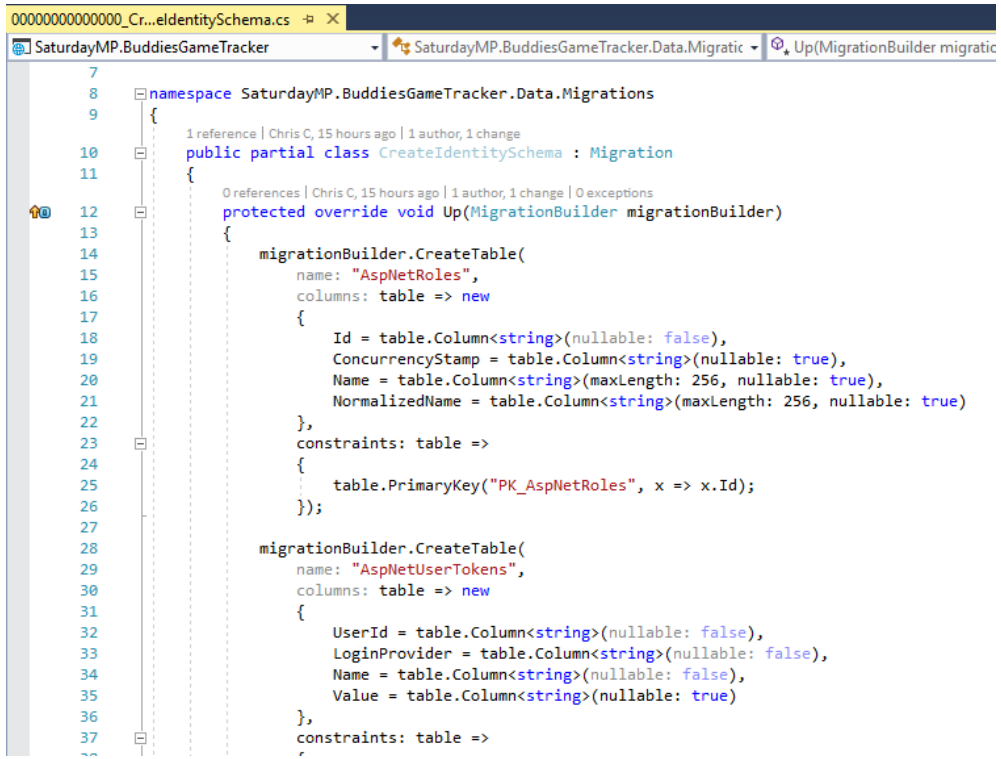
But the application raises an error



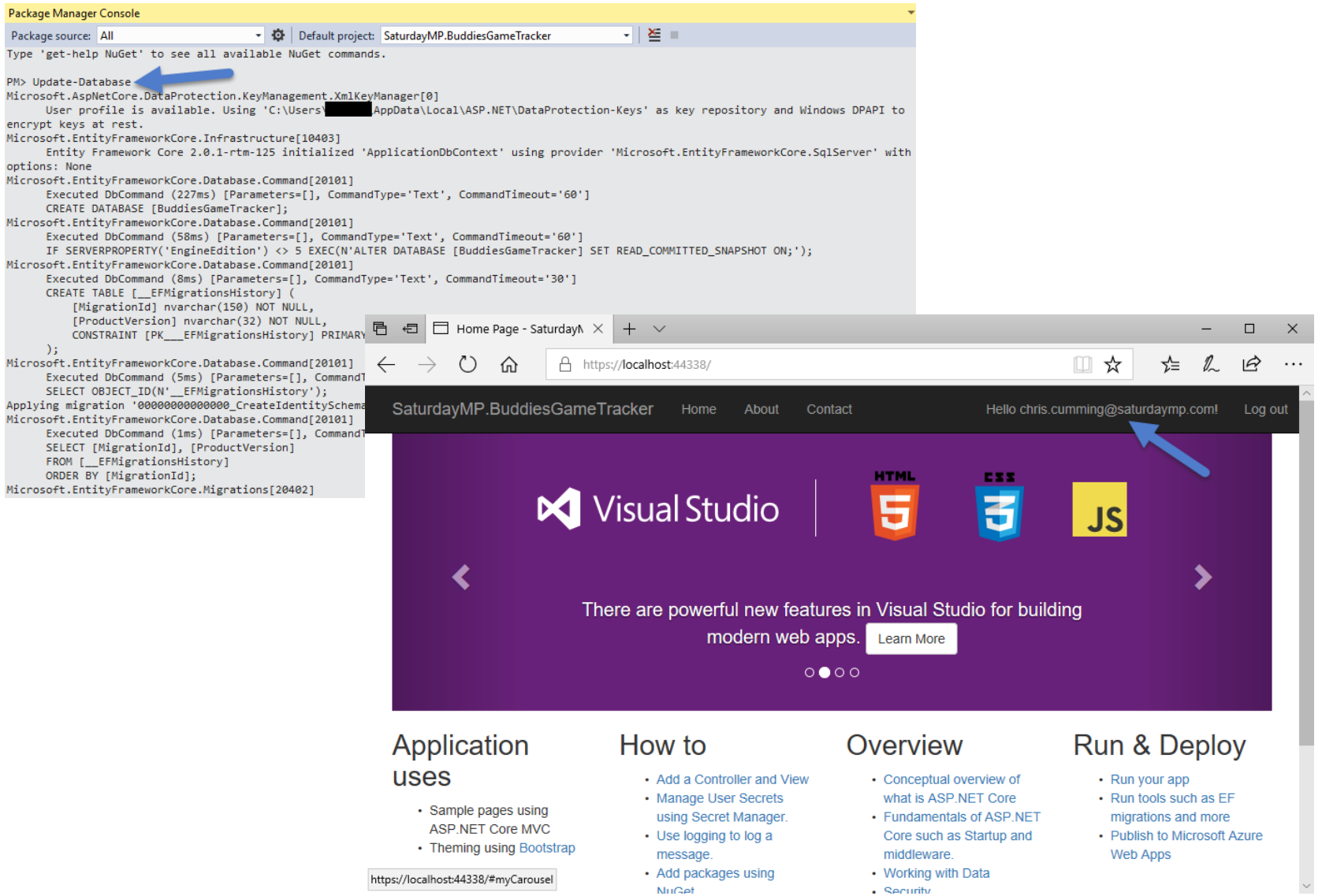
What is a migration?

- A file that describes changes to make to the database
- Can be auto generated or manually created
- Usually not written in SQL
- ORM tracks which migrations need to be run



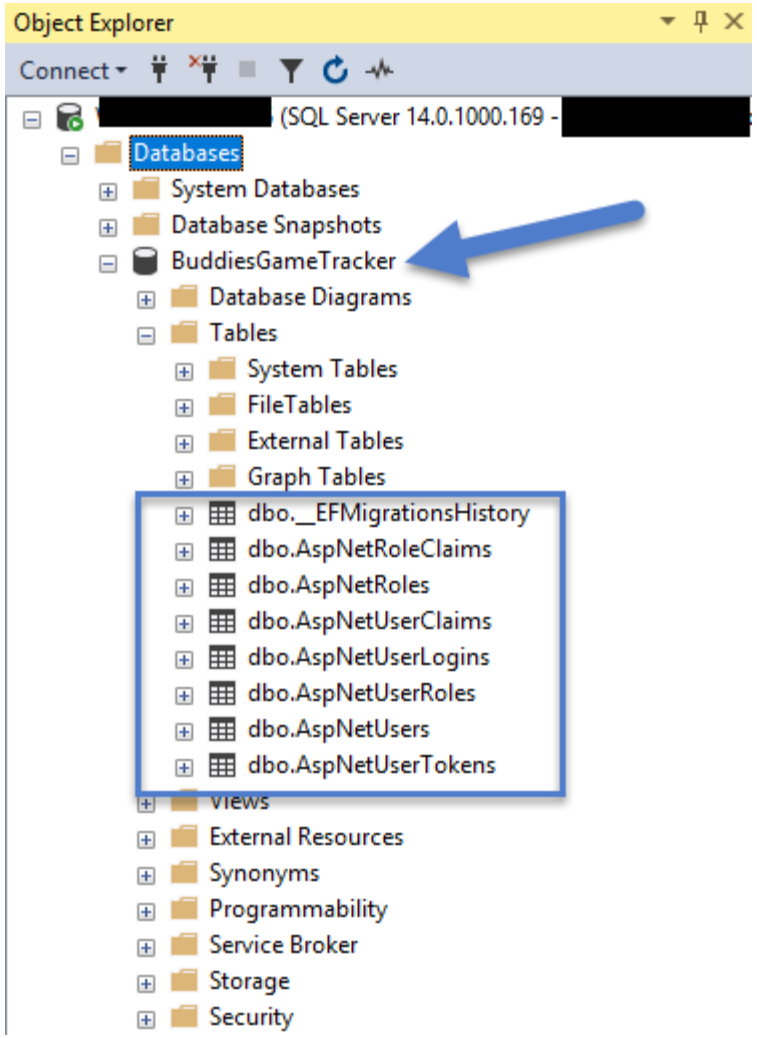


Bud runs the migration and can now create users



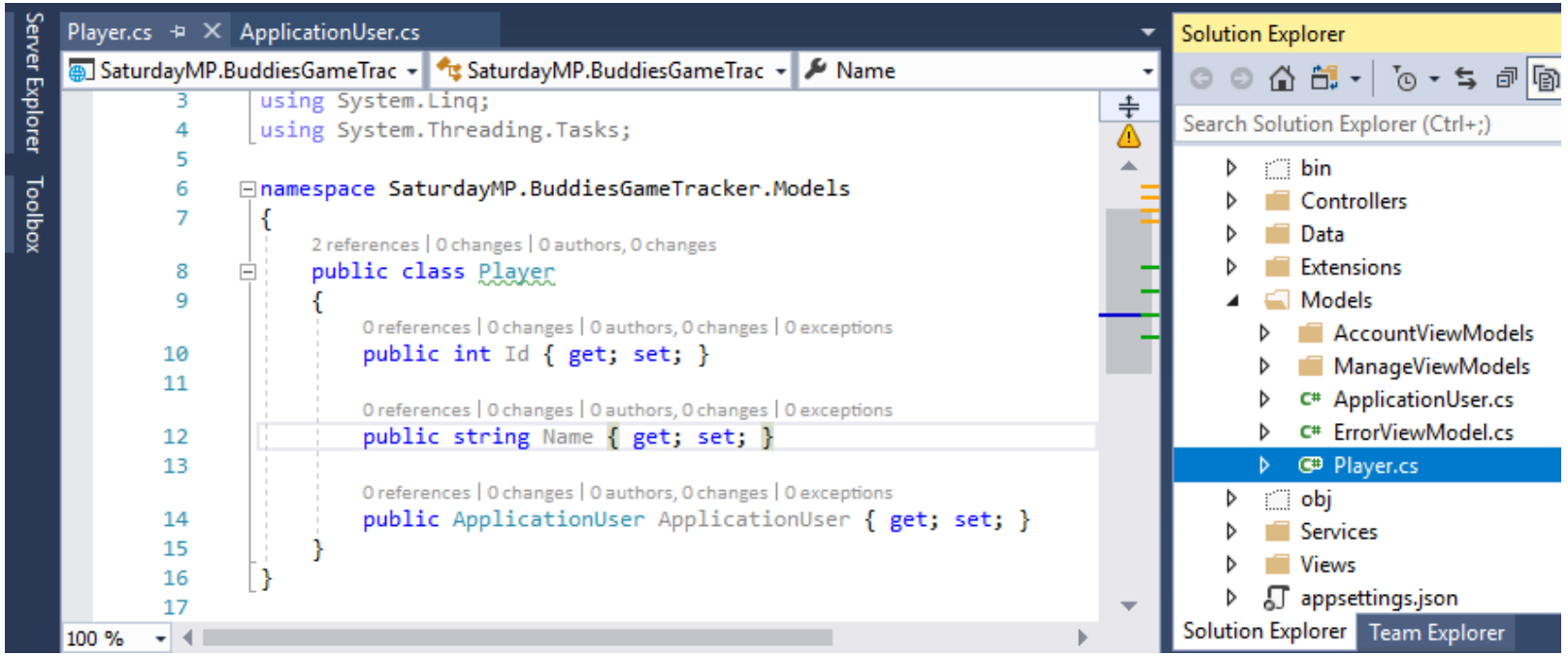
Bud doesn't look at the database but omnipotent DBAs do

- The migration created the database.
- It also created the authentication tables.
- Bud didn't have to open SQL Server or run a script.

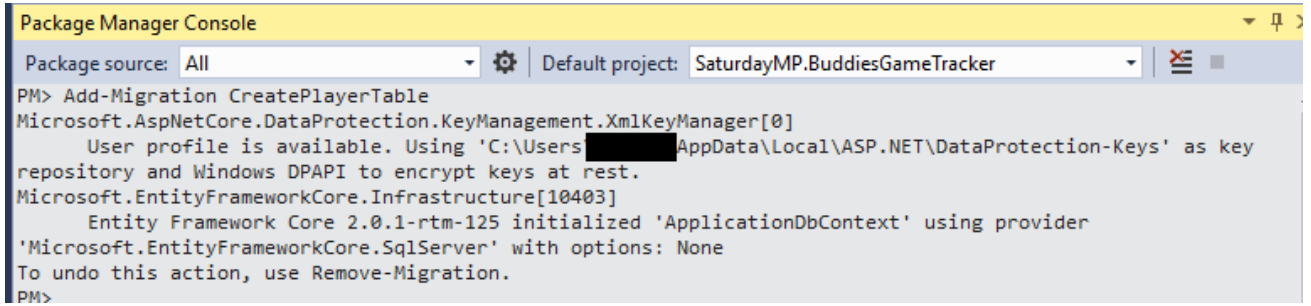


Don’t panic, migrations can create indexes

Bud creates the player class

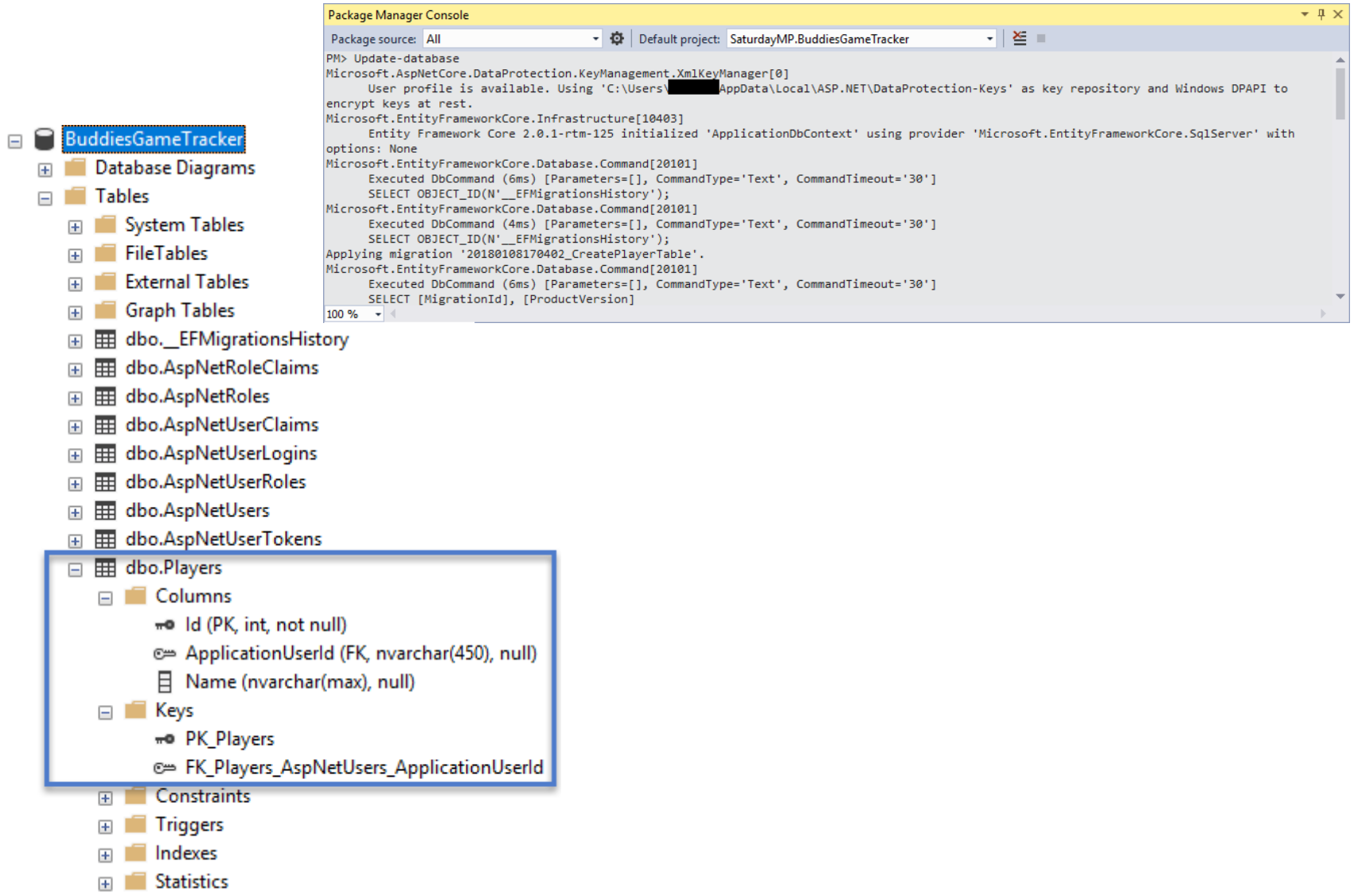


He creates the migration




```
protected override void Up(MigrationBuilder migrationBuilder)
{
    migrationBuilder.DropIndex(
        name: "UserNameIndex",
        table: "AspNetUsers");

    migrationBuilder.DropIndex(
        name: "IX_AspNetUserRoles_UserId",
        table: "AspNetUserRoles");

    migrationBuilder.DropIndex(
        name: "RoleNameIndex",
        table: "AspNetRoles");

    migrationBuilder.CreateTable(
        name: "Players",
        columns: table => new
        {
            Id = table.Column<int>(nullable: false)
                .Annotation("SqlServer:ValueGenerationStrategy", SqlServerValueGenerationStrategy.IdentityColumn),
            ApplicationUserId = table.Column<string>(nullable: true),
            Name = table.Column<string>(nullable: true)
        },
        constraints: table =>
        {
            table.PrimaryKey("PK_Players", x => x.Id);
            table.ForeignKey(
                name: "FK_Players_AspNetUsers_ApplicationUserId",
                column: x => x.ApplicationUserId,
                principalTable: "AspNetUsers",
                principalColumn: "Id",
                onDelete: ReferentialAction.Restrict);
        });

    migrationBuilder.CreateIndex(
        name: "UserNameIndex",
        table: "AspNetUsers"
```

Then runs the migration



Why Developers Use ORMs (i.e. their biggest strength)

~~[Shelock] His~~ The developer's ignorance was remarkable as his knowledge. Of contemporary literature, philosophy and politics he [or she] appeared to know next to nothing. My surprise reached a climax, however, when I found incidentally that he [or she] was ignorant of the ~~Copernican Theory~~ database design.

"You appear to be astonished" he said, smiling at my expression of surprise. "Now that I do know it I shall do my best to forget it."

"To forget it!"

… <Developer gives long speech about how a developer's brain is like an attic and only hold so much information>…

"But the ~~Solar System~~ Database!" [Watson] I protested.

"What the deuce is it to me?" he interrupted impatiently: "you say that we ~~go around the sun~~ store data in a database. If we ~~went round the moon~~ stored the data on a stone tablet it would not make a pennyworth of difference to me or to my work".

Why DBAs dislike ORMs (i.e. their biggest weaknesses)

Biggest weakness is also the abstraction.

"With great ~~power~~ abstractions comes great responsibility."

Common ORM problems that affect DBAs:

- N+1 Select
- Select unneeded columns
- Complex queries
- Don't play well with Views and Stored Procedures

How DBAs and ORMs can work together

- Let the ORM generate schema changes but review them
- Let the ORM access tables directly
- Adapt the ORM's naming schema
- Don't worry about being made obsolete
- Educate bad developers, if you can