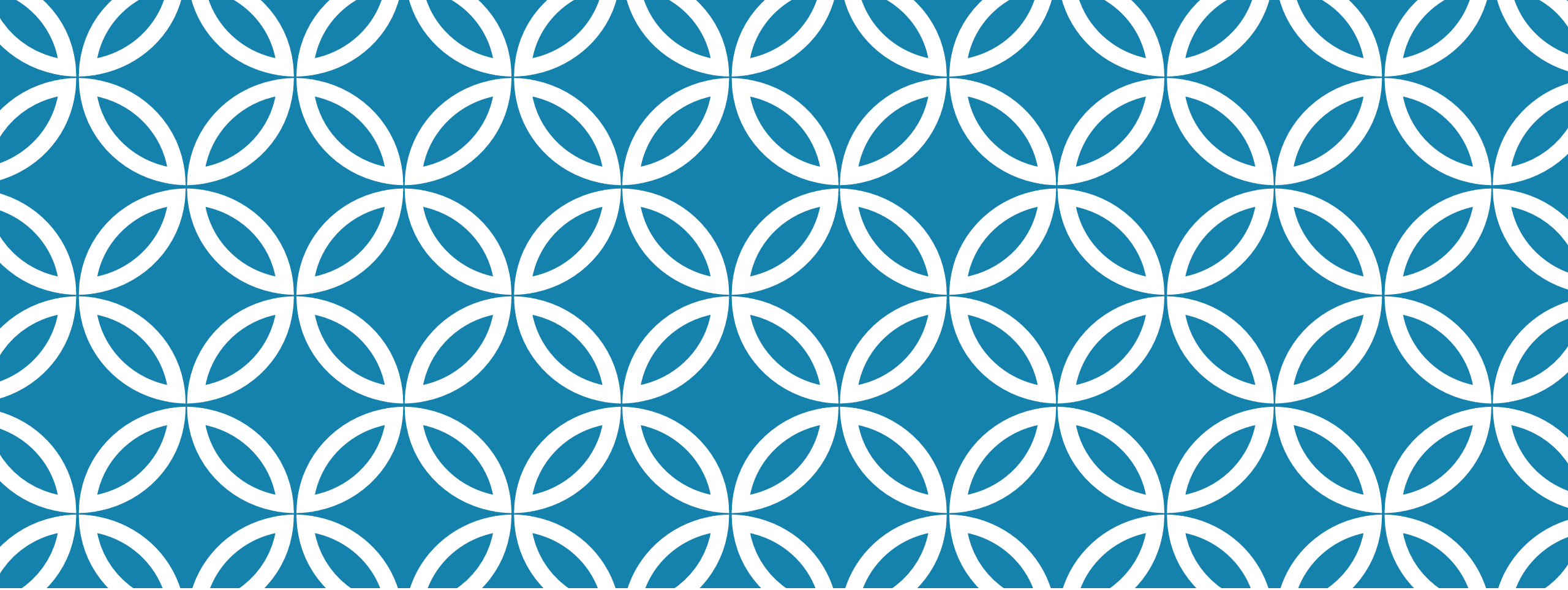

ENTITY FRAMEWORK
CODE FIRST & DATABASE FIRST

Arctech Info Private Limited

ENTITY FRAMEWORK CORE IMPLEMENTATION

EF Core can be implemented using one of the 3 approaches

- Database First
  - Create the database manually.
  - Then create entity classes and map to database tables
- Code First
  - Create the entity classes
  - Execute couple of commands to automatically generate the database and tables within.

CODE FIRST APPROACH

Create the Entity classes with properties defined in it.

- Entity classes represent tables and properties represent fields.
- Use Data Annotations to specify database specific settings

Create a class inheriting from DbContext and add DbSet properties for all entity classess.

- DbContext class represents the database

Use the following .NET commands in the Package manager console

- Add-migration <MigrationName>
  - Create a new Migration based on the current entity classes.
  - You can add multiple migrations over the development cycle of your project
- Remove-migration
  - Delete any newly created migrations.
- Update-database [-TargetMigration:<migrationName>]
  - To update all migrations into the database
  - Specify the –TargetMigration argument to rollback a migration up to a specific point

CODE FIRST APPROACH

Advantages

Auto create database and tables from business objects

Your code is always talking to a database schema, therefore there is rarely a “mismatch” in column types, table definitions etc.

Database version control

Out of the box way to manage database migrations. (Scripts not needed)

Good for apps which are not data-intensive. Most apps are not data-intensive

Preferred for new apps

You do not need SQL programmers and DBA in your development team

Disadvantages

You have to write everything related to the database in C#

For stored procedures you have to write the script inside a C# string and map it using Fluent API

To change anything in a database tables, you have to edit entity class and execute update-database

Not preferred for data-intensive apps

DATABASE FIRST APPROACH

Create the database and related tables using the database client like MS Sql Management Studio

Create the Entity classes with properties defined in it.

- Entity classes represent tables and properties represent fields.

Create a class inheriting from DbContext and add DbSet properties for all entity classess.

- DbContext class represents the database

3rd Party tools like EF Core Power tools

- This can be used to auto generate Entity Classes and DbContext class from an existing database

DATABASE FIRST APPROACH

Advantages

Simple to create the data model

Use the GUI provided by the database vendor. E.g. MS SQL Management Service

Mapping and creation of keys and relationships are easy using GUI

Preferred for data-intensive apps

Preferred for apps where a legacy database already exists

Disadvantages

If there is any change in database, model class needs to be extended with the same properties.

Creating and managing of keys and relationships requires more code changes.

So, changes need to be done in two places, for any database related changes

Database and C# entity classes could go out of sync.

You need SQL programmers and DBA in your development team