

Subtask 2; Flyway használata kutatómunka /quickStart

2/1; Mi a Flyway és milyen lehetőségeket kínál?

A Flyway egy adatbázis migrációs eszköz, részben nyílt forráskódú, segít sokféle adatbázis rendszerbeli adatbázis-séma módosítások kezelésében, nyomon követésében és telepítésében. Java-orientált, és Java-alapú migrációt biztosít, Java alapú, és széles körű támogatást nyújtanak az olyan Java keretrendszerekhez, mint a Spring Boot és a Vert.x. Támogatja az olyan builderekkel való integrációt, mint a Maven és a Gradle.

- Verzióalapú migrációs megközelítést nyújt az adatbázis-séma módosításához.
- A megadott .sql szkripteken keresztül, parancssorból függetlenül futtatható, tehát **SQL-t használ a változás meghatározásához.**
- A Flyway egy lineáris adatbázis-verziórendszerre épül, amely minden egyes verzióváltásnál növekszik. Ez néha konfliktusokat okozhat a párhuzamos fejlesztéssel. Az .sql szkript fájlneve határozza meg a migráció típusát általánosságban így:
 - V01__Add_New_Column.sql
 - V (a versionált), U (a visszavonás) és R (a megismételhető), utána verziószám
- A Flyway a migrációkat az adatbázis sémájában tárolja alapértelmezetten a **flyway_schema_history** nevű táblában. Postgres adatbázisban a public schema-ban.
- A változás sorrendjének kezelése viszonylag nehéz a Flywayben. A Flyway esetében a sorrend a fájlnevben szereplő verziószámtól és a migráció típusától függ.
- A visszaállításra (rollback), ha egy rossz módosítás katasztrofális problémát okozott az alkalmazásban: a Flyway rendelkezik egy visszavonó migrációval is, amelyet egy U-val kezdődő fájlnévvel lehet telepíteni, amelyet a visszavonandó verzió követ, ingyenes verziója megfelelően jó, de természetesen a fizetős verziója még összetettebb visszavonási funkcionalitást is kínál.
- A Flyway nehézkes környezetként mást és mást telepíteni (vagy nem telepíteni), mert akkor minden környezet/adatbázishoz más konfigurációs fájl kellene beállítani.
- Java-alapú migrációt biztosít.
- A Flyway nem támogat előfeltételeket, de az egyes sql-procedúrákon keresztül a legtöbb SQL-alapú adatbázisban írhatunk.

2/2; Hogyan köthető be olyan alkalmazásba, ami már rendelkezik adatokkal?

- 1; pom.xml-be flyway dependencia hozzáadása


```
<dependency>
  <groupId>org.flywaydb</groupId>
  <artifactId>flyway-core</artifactId>
</dependency>
```
- 2; app.yaml flyway konfigurációs beállításai:


```
spring:
  flyway:
    baseline-on-migrate: true
```

```
locations: classpath:db/migration
out-of-order: true
```

- 3; Migrációs fájlok létrehozása a src/main/resources/db/migration könyvtárban:

```
-- V1__init.sql
CREATE TABLE application_user (
  id BIGSERIAL PRIMARY KEY,
  active BOOLEAN,
  default_course_id BIGINT,
  email VARCHAR(255),
  first_name VARCHAR(255),
  last_name VARCHAR(255),
  . . .
);
```

- 4; adatbázis konfiguráció check in app.yaml fájlban:

```
spring:
  datasource:
    url: jdbc:postgresql://localhost:5433/database_name
    username: postgres
    password: root
    driver-class-name: org.postgresql.Driver
  jpa:
    hibernate:
      ddl-auto: validate
```

- 5; alkalmazás futtat, és a Flyway automatikusan végrehajtja a migrációkat az adatbázisban.

- **baseline-on-migrate: true** ha egy üres adatbázisra futtatjuk a migrációkat, a Flyway létrehoz egy alapvonalat (baseline) a migrációk előtt. Ez hasznos lehet, ha egy már létező adatbázist szeretnénk migrációs eszközzel kezelni, és el akarjuk kerülni a már meglévő sémák újraalkalmazását.
- **locations: classpath:db/migration** meghatározza, hogy hol találhatóak a migrációs fájlok. A classpath:db/migration azt jelenti, hogy a migrációs fájlokat a projekt classpath útvonalán belül a db/migration könyvtárban (ez az src/main/resources/db/migration könyvtár lesz) kell elhelyezni. A Flyway innen fogja betölteni és végrehajtani a migrációs fájlokat.
- **out-of-order: true** lehetővé teszi, hogy a migrációk nem sorrendben kerüljenek végrehajtásra. Ha egy korábbi verziójú migrációs fájl később kerül hozzáadásra, mint a már végrehajtott migrációk, a Flyway végrehajtja ezt a migrációt is. Ez hasznos lehet, ha több fejlesztő dolgozik párhuzamosan és különböző migrációkat hoznak létre.

classpath:

A projekt osztályútvonala (classpath) az a hely, ahol a Java futtatókörnyezet keresi az osztályfájlokat és egyéb erőforrásokat. Maven alapú Java projektekben ezen könyvtárak tartoznak az classpath-hoz:

- **src/main/java** - Az alkalmazás forráskódja.
- **src/main/resources** - Az alkalmazás erőforrásai (pl. konfigurációs fájlok, statikus erőforrások).
- **target/classes** - A lefordított osztály fájlok.

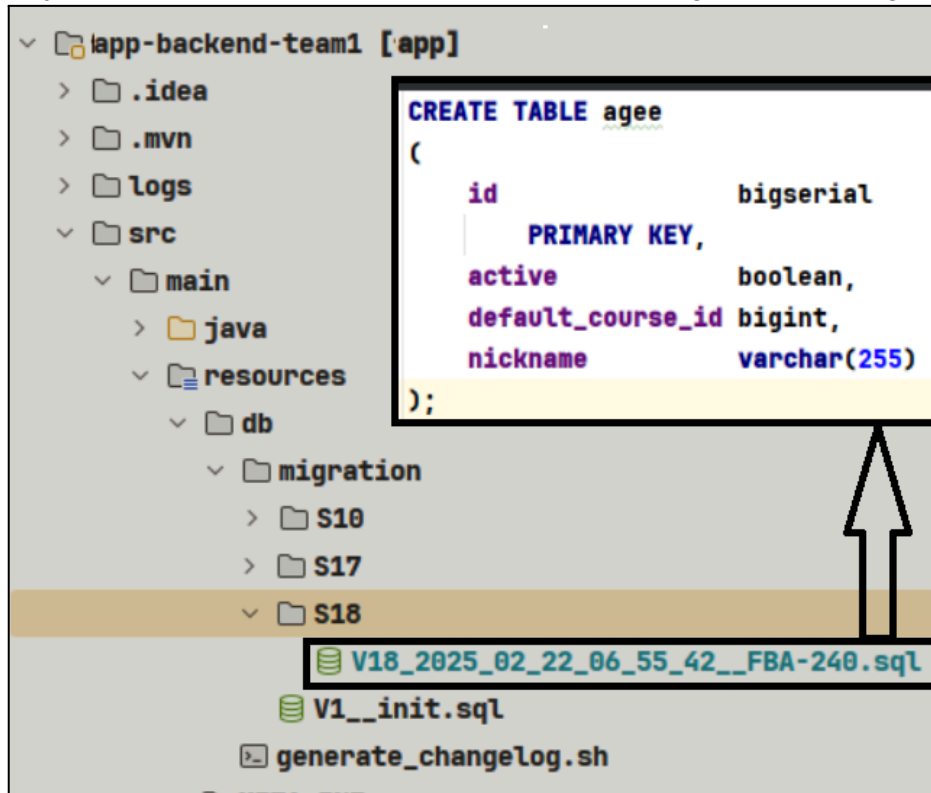
A Flyway migrációs fájlok esetében a classpath:db/migration azt jelenti, hogy a migrációs fájlokat a src/main/resources/db/migration könyvtárban kell elhelyezni.

generate_changelog.sh fájl:

- ő egy futtatható bash script
- ezzel a kis bash szkripttel hozzuk létre az új üres tartalmú .sql scripteket
- **futtatása**
 - például:(ajánlott a resources/db> mappából) **generate_changelog.sh 8 34**

- ahol a 8 a sprint száma, a 34 pedig a ticket száma
- lép be a cmd-édbbe (ajánlott a Run As admin)
- `./generate_changelog.sh 18 240`
- `app-backend-team1\src\main\resources\db> generate_changelog.sh 18 240`

C:\Windows\system32>cd C:\Users\HP\IdeaProjects\app-backend-team1\src\main\resources\db
 C:\Users\HP\IdeaProjects\app-backend-team1\src\main\resources\db>generate_changelog.sh 18 240



Azért, hogy ne legyenek ütközések, és egységesen tudjunk kezelni migrációs
 # szkripteket, ezzel a kis szkripttel tudunk létrehozni újat, ha szükséges
 #
 # Egyszerűen csak adjuk át paraméterként a **sprint** és a **ticket** számát
 # Kiadható többször egymás után, hiszen a másodperc is a fájlnev része,
 # és így lesz folytatódólagos
 #

pl.:
 # `./generate_changelog.sh 8 34`

```
SPRINT_NUMBER=$1
TICKET_NUMBER=$2
DATE_TIME=$(date '+%Y_%m_%d_%H_%M_%S')
```

```
if [ "$SPRINT_NUMBER" != "" ] || [ "$TICKET_NUMBER" != "" ]; then
```

```
    MIGRATION_SCRIPT_IDENTIFIER=V"$SPRINT_NUMBER"_"$(echo $DATE_TIME)__FBA-$2
```

```
    NEW_SCRIPT_FOLDER=./migration/"$S1"
```

```
    mkdir -p $NEW_SCRIPT_FOLDER
    touch $NEW_SCRIPT_FOLDER/$MIGRATION_SCRIPT_IDENTIFIER.sql
    echo "Created: $NEW_SCRIPT_FOLDER/$MIGRATION_SCRIPT_IDENTIFIER.sql"
```

```
else
```

```
echo "Missing parameters!"
fi
```

Ez a bash script a következő lépéseket hajtja végre:

- 1. Beállítja a `SPRINT_NUMBER` változót `8`-ra és a `TICKET_NUMBER` változót `34`-re.
- 2. Létrehozza a `DATE_TIME` változót az aktuális dátum és idő formátumával: `YYYY_MM_DD_HH_MM_SS`.
- 3. Összeállítja a `MIGRATION_SCRIPT_IDENTIFIER` változót a következő formátumban: `V8_YYYY_MM_DD_HH_MM_SS_FBA-34`.
- 4. Létrehozza a `NEW_SCRIPT_FOLDER` változót a következő formátumban: `./migration/S8`.
- 5. Létrehozza a `NEW_SCRIPT_FOLDER` könyvtárat, ha még nem létezik.
- 6. Létrehoz egy üres SQL fájlt a `NEW_SCRIPT_FOLDER` könyvtárban a `MIGRATION_SCRIPT_IDENTIFIER` névvel.
- 7. Kiírja a létrehozott fájl nevét a konzolra.

Például, ha a script 2023. október 10-én, 15:30:45-kor fut, a következő fájl jön létre:

```
./migration/S8/V8_2023_10_10_15_30_45__FBA-34.sql
```

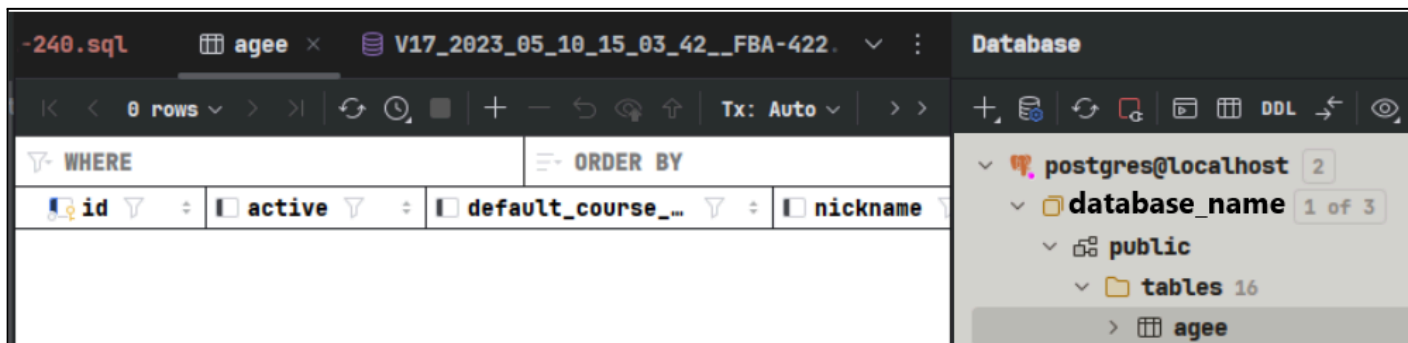
Feltöltöd a futtatandó sql parancsokat, ami bármi adatbázis építő lehet: pl. CREATE TABLE, INDEX, vagy INSERT, UPDATE... stb.

Ezek után elindított alkalmazásodat, a Spring Boot automatikusan lefuttatja az eddig még nem lefutott, tehát az új .sql scripteket.

```
Run Application
Console
C:\Users\HP\jdk\corretto-17.0.14\bin\java.exe ...

Spring Boot (v3.0.3)

2025-02-22 07:17:46.493 +01:00 INFO 58932 --- [main] h.p.feedbackapp.FeedbackApplication : Starting Application using Java 17.0.14 with PID 58932
(C:\Users\HP\IdeaProjects\app-backend-team1\target\classes started by HP in C:\Users\HP\IdeaProjects\app-backend-team1)
2025-02-22 07:17:46.500 +01:00 DEBUG 58932 --- [main] h.p.feedbackapp.FeedbackApplication : Running with Spring Boot v3.0.3, Spring v6.0.5
2025-02-22 07:17:46.501 +01:00 INFO 58932 --- [main] h.p.feedbackapp.FeedbackApplication : No active profile set, falling back to 1 default profile: "default"
2025-02-22 07:17:48.223 +01:00 INFO 58932 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2025-02-22 07:17:48.383 +01:00 INFO 58932 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 152 ms. Found 13 JPA repository interfaces.
2025-02-22 07:17:49.498 +01:00 INFO 58932 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2025-02-22 07:17:49.521 +01:00 INFO 58932 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-02-22 07:17:49.521 +01:00 INFO 58932 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.5]
2025-02-22 07:17:49.705 +01:00 INFO 58932 --- [main] o.a.c.c.[] [Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-02-22 07:17:49.786 +01:00 INFO 58932 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 3135 ms
2025-02-22 07:17:50.143 +01:00 INFO 58932 --- [main] o.f.c.internal.license.VersionPrinter : Flyway Community Edition 9.5.1 by Redgate
2025-02-22 07:17:50.143 +01:00 INFO 58932 --- [main] o.f.c.internal.license.VersionPrinter : See what's new here: https://flywaydb.org/documentation/learnmore/releaseNotes#9.5.1
2025-02-22 07:17:50.144 +01:00 INFO 58932 --- [main] o.f.c.internal.license.VersionPrinter :
2025-02-22 07:17:50.173 +01:00 INFO 58932 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2025-02-22 07:17:50.397 +01:00 INFO 58932 --- [main] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection org.postgresql.jdbc.PgConnection@3534651a
2025-02-22 07:17:50.399 +01:00 INFO 58932 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2025-02-22 07:17:50.429 +01:00 INFO 58932 --- [main] o.f.c.i.database.base.BaseDatabaseType : Database: jdbc:postgresql://localhost:5433/database_name (PostgreSQL 12.22)
2025-02-22 07:17:50.550 +01:00 INFO 58932 --- [main] o.f.core.internal.command.DbValidate : Successfully validated 21 migrations (execution time 00:00.065s)
2025-02-22 07:17:50.574 +01:00 INFO 58932 --- [main] o.f.core.internal.command.DbMigrate : Current version of schema "public": 17.2023.05.10.15.03.42
2025-02-22 07:17:50.574 +01:00 WARN 58932 --- [main] o.f.core.internal.command.DbMigrate : outOfOrder mode is active. Migration of schema "public" may not be reproducible
2025-02-22 07:17:50.592 +01:00 INFO 58932 --- [main] o.f.core.internal.command.DbMigrate : Migrating schema "public" to version "18.2025.02.22.06.55.42 - FBA-240"
2025-02-22 07:17:50.776 +01:00 INFO 58932 --- [main] o.f.core.internal.command.DbMigrate : Successfully applied 1 migration to schema "public", now at version 18.2025.02.22.06.55.42 (execution time 00:00.212s)
2025-02-22 07:17:50.911 +01:00 INFO 58932 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2025-02-22 07:17:50.995 +01:00 INFO 58932 --- [main] org.hibernate.Version : HHH000412: Hibernate ORM core version 6.1.7.Final
2025-02-22 07:17:51.403 +01:00 INFO 58932 --- [main] SQL dialect : HHH000400: Using dialect: org.hibernate.dialect.PostgreSQLDialect
```



2/3; Mit jelent a baseline script?

A baseline script a Flyway adatbázis-migrációs eszközben egy olyan SQL vagy Java fájl, amely az adatbázis aktuális állapotát rögzíti egy adott ponton. Ez a script lehetővé teszi, hogy egy már létező adatbázis sémát és adatokat kezelni lehessen a Flyway segítségével anélkül, hogy újra kellene alkalmazni a már meglévő változtatásokat.

Példa egy baseline script-re:

```
-- V1__baseline.sql
CREATE TABLE application_user (
  id BIGSERIAL PRIMARY KEY,
  active BOOLEAN,
  default_course_id BIGINT,
  email VARCHAR(255),
  first_name VARCHAR(255),
  last_name VARCHAR(255),
  exercise_status VARCHAR(255),
  password VARCHAR(255),
  role VARCHAR(255) NOT NULL,
  username VARCHAR(255)
);

CREATE TABLE course (
  id BIGSERIAL PRIMARY KEY,
  name VARCHAR(255),
  description TEXT,
  zoom_id BIGINT
);

-- További táblák és sémák létrehozása...
```

Flyway konfiguráció a baseline script használatához:

spring:

flyway:

baseline-on-migrate: true

baseline-version: 1

locations: classpath:db/migration

- **baseline-on-migrate: true** beállítás azt jelenti, hogy a Flyway létrehoz egy alapvonalat (baseline) a migrációk előtt, ha egy üres adatbázisra futtatjuk a migrációkat.
- **baseline-version: 1** beállítás meghatározza, hogy melyik verziót tekintse alapvonalnak.
 - A Flyway által nyilvántartott alapvonalak (baseline-ok) megtekintéséhez az **flyway_schema_history** táblát kell lekérdezni az adatbázisban. Ez a tábla tartalmazza az összes migrációs információt, beleértve az alapvonalakat is. Az alábbi SQL lekérdezéssel megtekintheted az összes alapvonalat, ez a lekérdezés visszaadja az összes olyan rekordot,

amely alapvonal típusú (type = 'BASELINE').:

```
SELECT * FROM flyway_schema_history WHERE type = 'BASELINE';
```

installed_rank	version	description	type	script	checksum	installed_by	installed_on	execution_time	success
1	1.1	<< Flyway Baseline >>	BASELINE	<< Flyway Baseline >>	<null> null		2023-01-09 13:37:06.748617		0 - true
2	2.10.2022.11.28.14.56.45	FBA-168	SQL	S10/V10_2022_11_28_14_56_45_FBA-168.sql	1524958457 postgres		2023-01-09 13:37:06.847695	24 -	true
3	3.13.2023.01.10.09.16.57	FBA-267	SQL	S13/V13_2023_01_10_09_16_57_FBA-267.sql	1535978429 postgres		2023-01-20 13:14:45.854070	44 -	true
4	4.13.2023.01.11.11.33.08	FBA-237	SQL	S13/V13_2023_01_11_11_33_08_FBA-237.sql	1932377148 postgres		2023-01-20 13:14:45.996805	26 -	true
5	5.13.2023.01.12.22.27.37	FBA-236	SQL	S13/V13_2023_01_12_22_27_37_FBA-236.sql	1910678470 postgres		2023-01-20 11:52:32.520539	35 -	true
6	6.13.2023.01.12.22.29.50	FBA-238	SQL	S13/V13_2023_01_12_22_29_50_FBA-238.sql	288025686 postgres		2023-01-20 11:52:32.664065	18 -	true
7	7.14.2023.01.24.15.21.14	FBA-214	SQL	S14/V14_2023_01_24_15_21_14_FBA-214.sql	-834261344 postgres		2023-01-30 11:52:32.778023	22 -	true
8	8.14.2023.02.06.14.00.56	FBA-295	SQL	S14/V14_2023_02_06_14_00_56_FBA-295.sql	3599977575 postgres		2023-02-07 09:10:27.251562	17 -	true
9	9.13.2023.01.13.15.06.23	FBA-235	SQL	S13/V13_2023_01_13_15_06_23_FBA-235.sql	-1745391573 postgres		2023-02-14 13:13:01.761987	74 -	true
10	10.14.2023.02.03.13.14.35	FBA-310	SQL	S14/V14_2023_02_03_13_14_35_FBA-310.sql	900422973 postgres		2023-02-14 13:17:35.297076	40 -	true
11	11.14.2023.02.08.13.30.29	FBA-335	SQL	S14/V14_2023_02_08_13_30_29_FBA-335.sql	-1035444521 postgres		2023-02-14 13:17:35.416986	340 -	true
12	12.15.2023.02.17.09.36.19	FBA-353	SQL	S15/V15_2023_02_17_09_36_19_FBA-353.sql	-518865298 postgres		2023-03-09 15:11:52.728231	43 -	true
13	13.15.2023.02.20.11.11.21	FBA-353	SQL	S15/V15_2023_02_20_11_11_21_FBA-353.sql	-1814585031 postgres		2023-03-09 15:11:52.965493	21 -	true
14	14.15.2023.03.23.10.48.58	FBA-399	SQL	S15/V15_2023_03_23_10_48_58_FBA-399.sql	5313399 postgres		2023-03-23 11:52:12.404410	26 -	true
15	15.15.2023.02.27.13.15.36	FBA-329	SQL	S15/V15_2023_02_27_13_15_36_FBA-329.sql	1135135174 postgres		2023-03-27 10:08:20.873540	38 -	true
16	16.15.2023.03.20.15.07.40	FBA-329	SQL	S15/V15_2023_03_20_15_07_40_FBA-329.sql	643554687 postgres		2023-03-27 10:08:21.020475	21 -	true
17	17.17.2023.04.18.13.30.45	FBA-413	SQL	S17/V17_2023_04_18_13_30_45_FBA-413.sql	-644888107 postgres		2023-04-19 11:00:36.604947	34 -	true
18	18.17.2023.04.18.17.20.49	FBA-413	SQL	S17/V17_2023_04_18_17_20_49_FBA-413.sql	256059083 postgres		2023-04-19 11:00:36.725653	10 -	true
19	19.17.2023.05.10.15.03.42	FBA-422	SQL	S17/V17_2023_05_10_15_03_42_FBA-422.sql	-29339961 postgres		2023-06-01 12:00:45.754034	50 -	true
20	20.16.2023.02.22.06.55.42	FBA-240	SQL	S18/V18_2023_02_22_06_55_42_FBA-240.sql	-205120193 postgres		2023-02-22 07:17:50.569924	102 -	true

2/4; Hogyan változik meg egy entity módosításának folyamata Flyway használatával?

Ez a kérdés azt jelenti, hogy a domain/ -be tett JPAs'@Entity-edden változtatsz, pl. hozzáadsz a .java fájl forráskódban egy új mezőt, akkor a Flyway hogyan szedi ezt fel?

1; Entitás Módosítása

Módosítsd a Java entitás osztályt a szükséges változtatásokkal. Például, ha az ApplicationUser entitáshoz hozzáadunk egy új mezőt:

@Entity

```
public class ApplicationUser {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private Boolean active;
    private Long defaultCourseId;
    private String email;
    private String firstName;
    private String lastName;
    .
    private String newField; // Új mező hozzáadása

    // Getters és Setters
}
```

2; Flyway Migrációs Fájl Létrehozása

Hozz létre egy új Flyway .sql migrációs fájlt a src/main/resources/db/migration könyvtárban. Például

2/1; generate changelog.sh val hozd létre az üres sql script fájlot

- CMD-ét ha lehet Admin userként nyisd meg
- C:\Windows\system32>cd C:\Users\HP\IdeaProjects\app...\src\main\resources\db
- C:\Users\HP\IdeaProjects\app...\src\main\resources\db>generate_changelog.sh 18 241

2/2; Töltsd fel a megfelelő, az imént sql utasítással

```
-- V2__add_new_field_to_application_user.sql
```

```
ALTER TABLE application_user ADD COLUMN new_field VARCHAR(255);
```

3. Flyway Konfiguráció Ellenőrzése

A Flyway megfelelően legyen konfigurálva az application.yaml fájlban:

```
spring:  
  flyway:  
    baseline-on-migrate: true  
    locations: classpath:db/migration
```

4. Alkalmazás Futtatása

Indítsd el az alkalmazást, és a Flyway automatikusan végrehajtja az új migrációs fájlt, amely hozzáadja az új mezőt az adatbázis táblához. Ezekkel a lépésekkel sikeresen módosíthatod az entitást és az adatbázis sémát Flyway használatával.