

Subtask 1; Adatbázis migráció kutatómunka

/intro

1/1; Mit jelent az adatbázis migráció?

Minden egyes szoftverfejlesztés általában magába foglal

- kód módosítások végrehajtását
- adatbázis módosításokat végrehajtását,
- rendszer üzemeltetési változtatásokat.

A Flyway egy adatbázis migrációs eszköz ezért most bevezetésképpen közelítsük meg magát az adatbázis migráció témakörét.

A kódváltoztatásokat a Git verziókezelők egyikének segítségével rendszerezetten változtatásokként, általában egyszerre több funkció kiélesztésekor pedig egy verziószámmal ellátva tárolják. A funkciók vs fejlesztések nyilvántartása nagyon fontos a változások követéért, tárolásáért, másrészt szükség esetén a könnyen vissza állíthatóságért a kód egy korábbi még jól működő állapotára.

[Atomic:](<https://github.com/MazinIsmael/JavaLearnings/tree/master/src/main/concurrency/com/atomic>)

Az adatbázis változtatásokat adatbázis migrációs eszközök segítségével összefogottan, rendszerezetten lehet kivitelezni, adatbázis scriptek automatikus futtatásával.

Egybentartás:

Erősen ajánlott a szoftverkód, azon belül mind a backend kód verziók, és mind a frontend kód verziók, és a az adatbázis verziók együttes nyilvántartása, a kompatibilitás fenntartásért, és a "mi mivel működik együtt" átlátásáért. Ezt a backend kód git felületén egy pl. projektmappa/doc mappában is lehet vezetni tehát jól látható, mappa struktúrában projekt mappához közeli helyen, amire például a git felületén megnyíló README.md fájlban egy az alábbi általános példa szerint könnyen oda navigálhatunk:

[a relative link](markdown_file.md)

például egyrészt egy:

- CHANGELOG.md fájlra, amelyben előre meghatározott struktúrában adják meg időrendben a kiélesztett fejlesztéseket, benne a megvalósított funkciókkal (JIRA, vagy REDMINE fejlesztési jegyek), funkciók megnevezése, kitett backend verzió, kiélesztett adatbázis verzió, mindezek szükséges üzemeltetési összetevői.

Ezenkívül ajánlott a kiélesztett frontend felületén is a verzió számokat egy footer részben, vagy esetleg egy /config (akár jelszóval védett) url felületén megjeleníteni.

What is a changelog?

Forrás:

<https://www.freecodecamp.org/news/a-beginners-guide-to-git-what-is-a-changelog-and-how-to-generate-it/>

A changelog is a file that shares a chronologically ordered list of the changes you've made on your project. It's often organized by the version with the date followed by a list of added, improved, and removed features. Globally, there are two ways to write a changelog:

- the usual way: create a text file and start to enumerate all your changes with a specific date
- the developer choice (alias the lazy option): auto-generate your changelog from your commit messages. I have good news for you – this is what you're going to learn in this article!

Adatbázis migráció:

Forrás: Copilot

Az adatbázis migráció egy olyan folyamat, amely során az adatbázis sémáját vagy adatait egy adott állapotból egy másik állapotba változtatjuk. Ez magában foglalhatja az új táblák létrehozását, meglévő táblák módosítását, adatok áthelyezését vagy törlését. Az adatbázis migrációk célja, hogy az adatbázis struktúráját és tartalmát szinkronban tartsák az alkalmazás változásaival. Az adatbázis migrációk kezelése érdekében gyakran használnak eszközöket, mint például a Flyway vagy a Liquibase, amelyek lehetővé teszik a migrációk verziókezelését és automatizálását. Ezek az eszközök segítenek abban, hogy a fejlesztők könnyen alkalmazhassák a szükséges változtatásokat az adatbázisban, és biztosítsák, hogy minden környezetben (fejlesztői, teszt, éles) ugyanazok a változtatások legyenek alkalmazva. Például egy Flyway migrációs fájl így nézhet ki:

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

Ez a migrációs fájl létrehoz egy `table_name` nevű táblát az adatbázisban a megadott oszlopokkal.

Forrás:

<https://www.acronis.com/en-us/blog/posts/what-is-database-migration/>

Database migration refers to migrating data from a source database to a target database via database migration tools. When the migration process is complete, the dataset from the source database resides entirely on the target database, often in a restructured form.

Az adat-migráció az adatok migrálását jelenti egy forrásadatbázisból egy céladatbázisba adatbázis-migrációs eszközök segítségével. A migrációs folyamat befejeztével a forrásadatbázis adatállománya teljes egészében a céladatbázisban található, gyakran átstrukturált formában.

A forrásadatbázishoz hozzáférő felhasználók átkerülnek a céladatbázisra; ha az átállás sikeres, a forrásadatbázist általában leállítják.

Adatmigráció:

Forrás:

<https://euromacc.hu/adatmigracio/>

Az adatmigráció különböző szoftverrendszerek vagy rendszer verziók közötti adatmozgatás, költöztetés. Gyakorlatban ez azt jelenti, hogy az egyik rendszer adatszerkezetéből átalakítjuk az adatokat úgy, hogy a másik adatbázisrendszer tudja használni.” Típusai:

- BigBang során egy lépésben kerül át minden adat az új rendszerbe. Ennek leginkább az egyszerűségében rejlik az előnye. Ez folyamatos rendszerhozzáférés esetén hátrány, hogy az élő rendszer leáll a migrációs folyamat során.
- Iteratív migráció: párhuzamosan futtatja a régi és az új rendszert, miközben az adatok szakaszonként mozognak. Ez kiküszöböli az állásidőt vagy a működési zavarokat, és megkönnyíti az adatmigrációs csapat számára az egyes fázisok sikerességének megállapítását.

1/2; Mikor lehet szükség a db migráció használatára?

A vállalatok különböző okok miatt végeznek adatbázis-migrációt:

- Az informatikai költségek csökkentése azáltal, hogy az adatokat egy régebbi adatbázisból egy jobb erőforrás-felhasználású adatbázisba migrálják.
- Más típusú adatbázis átvétele a késleltetés csökkentése és a mindennapi adatkezelési folyamatok fellendítése érdekében.
- Az adatok integritásának védelme egy régebbi rendszer nyugdíjazásával az adatbázis korszerűsítése érdekében.
- Az adatbázis legfrissebb verziójára való frissítés a biztonság javítása és a megfelelés teljesítése érdekében.
- Több adatbázis meglévő adatainak egyetlen tárhelyre történő összevonása az egységes fájlnézet lehetővé tétele érdekében.
- Adatreplikáció végrehajtása a felhőalapú adatbázis-tárolás diverzifikálása (kockázat csökkentése) érdekében.

1/3; Milyen eszközök közül választhatunk és ezeknek mik az előnyei/hátrányai?

Adatbázis migrációs eszközök: számos adatbázis-kezelő rendelkezik beépített migrációs eszközökkel, amelyek lehetővé teszik az adatbázisok közötti átmenetet. Ezek az eszközök általában parancssori vagy grafikus felületen keresztül érhetők el, és különböző funkciókat kínálnak, például adatok exportálását, importálását, adatbázisok összevonását stb.

Forrás: <https://www.baeldung.com/liquibase-vs-flyway>

A folyamatos integráció és az adatbázis-fejlesztés automatizált refaktorálásának korában élünk, melyet egy ún. evolúciós adatbázis-tervezés technikai megvalósító az adatbázis migrációs eszközök is megvalósíthatnak többek között (cégek, és migrációs-felhős szolgáltatások is). Az olyan eszközök, mint a **Liquibase** és a **Flyway** ezeket a technikákat követik, és iteratív fejlesztési megközelítéssel dolgoznak.

Mindkét eszköz: Bizonyos mértékig nyílt forráskódúak, és segítenek az adatbázis-séma módosítások kezelésében, nyomon követésében és telepítésében. Verzióalapú migrációs megközelítést alkalmaznak az adatbázis-séma módosításához. Java alapúak, és széles körű támogatást nyújtanak az olyan Java keretrendszerekhez, mint a Spring Boot és a Vert.x. Támogatják az olyan építőeszközökkel való integrációt, mint a Maven és a Gradle. A megadott szkripteken keresztül parancssorból függetlenül futtathatók. Az adatbázisok széles skáláját támogatják.

A Flyway SQL-t használ a változás meghatározásához. Másrészt a **Liquibase** rugalmasságot biztosít a változás különböző formátumokban történő meghatározásához, beleértve az SQL-t is, mint például XML, YAML és JSON.

A Flyway egy lineáris adatbázis-verziórendszerre épül, amely minden egyes verzióváltásnál növekszik. Ez néha konfliktusokat okozhat a párhuzamos fejlesztéssel. A Flyway szkript fájlneve határozza meg a migráció típusát. Például egy migrációnak a következő előtagokkal kell rendelkeznie:

- V (a versionált), U (a visszavonás) és R (a megismételhető). Ezt követi egy verziószám és egy elválasztó __ (két aláhúzás), majd egy leírás és egy
- .sql utótag, például V01__Add_New_Column.sql.

A Liquibase migrációnak nem kell követnie a fájlnev konvenciókat. A Liquibase-ben a változásokat egy főkönyv, az úgynevezett master changelog kezeli, amely úgy lesz definiálva, hogy tartalmazza az összes migrációt.

Mindkét eszköz egy táblázatban tárolja a telepített változásokat. A **Flyway** a migrációkat az adatbázis sémájában tárolja alapértelmezetten a **flyway_schema_history** nevű táblában. Hasonlóképpen, a Liquibase a telepített migrációkat az databasechangelog nevű táblában tárolja. Mindkét eszköz támogatja az alapértelmezett konfiguráció felülbírlását a tábla nevének megváltoztatásához.

A változás sorrendjének kezelése viszonylag nehéz a **Flywayben**. A **Flyway** esetében a sorrend a fájlnevben szereplő verziószámtól és a migráció típusától függ. Ezzel szemben a Liquibase egy külön **master_changelog** nevű fájlt használ, amelyben a módosítások a meghatározott sorrendben kerülnek telepítésre.

A visszaállításra (rollback) mindig akkor van szükség, ha egy rossz módosítás katasztrofális problémát okozott az alkalmazásban. A Liquibase lehetőséget biztosít a teljes visszaállításra vagy bizonyos migrációk visszavonására (csak a fizetős verziókban érhető el).

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.

Vannak olyan felhasználási esetek, amikor egy változást csak az egyik környezetbe kell telepítenünk, ekkor a Liquibase jobb. A **Flyway** is képes erre, de nehézkes, mert akkor minden környezethez vagy adatbázishoz más konfigurációs fájlt kellene beállítani. A Liquibase segítségével könnyen hozzáadhatunk címkéket és kontextusokat, hogy biztosítsuk a telepítést bizonyos helyeken.

Mindkét eszköz erősen Java-orientált, és Java-alapú migrációt biztosít. A **Flyway** és a Liquibase lehetővé teszi a migráció meghatározását egy Java-fájlon belül.

A Liquibase lehetővé teszi a felhasználók számára, hogy pillanatfelvételt (snapshot) készítsenek az adatbázis aktuális állapotáról. Ezt az állapotot használhatjuk egy másik adatbázissal való összehasonlításra. Ez nagyon hasznos lenne olyan forgatókönyvekben, mint a failover és az adatbázis-replikáció. A **Flyway** viszont nem támogatja a pillanatkép-funkciókat.

A Liquibase egy további funkciót kínál, az úgynevezett előfeltételeket (**pre-conditions**). Az előfeltételek lehetővé teszik a felhasználók számára, hogy az adatbázis aktuális állapota alapján alkalmazzanak módosításokat. Egy módosításkészlet csak akkor hajtódik végre, ha megfelel ezeknek az előfeltételeknek. A **Flyway** viszont nem támogatja ezt. A procedúrákon keresztül azonban a legtöbb SQL-alapú adatbázisban alkalmazhatunk ilyen előfeltételeket.