

# JEGYZŐKÖNYV

Adatbázis rendszerek MSc

2021. tavasz féléves feladat

Készítette: **Fekete Antal Kristóf**

Neptun kód: **B5T3ZN**

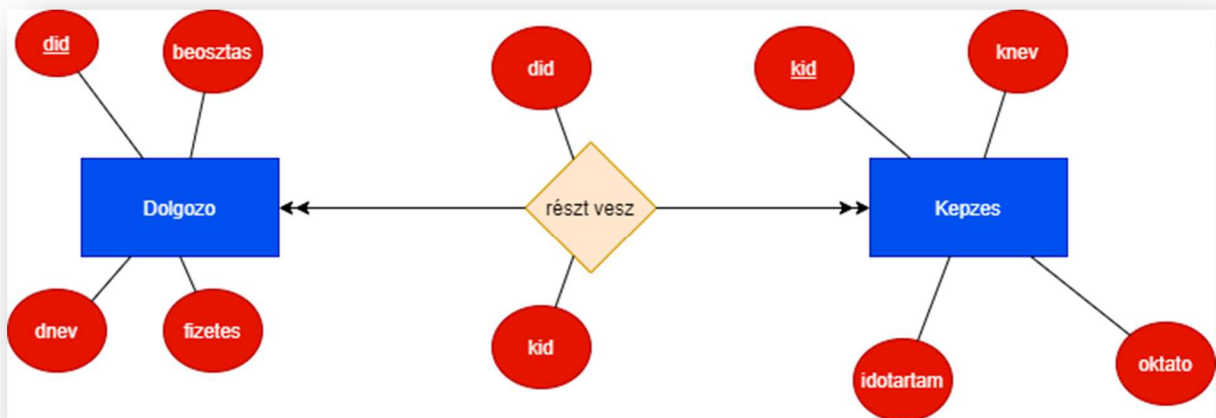
## A feladatok leírása

1. feladat: Adott az alábbi séma: **Book[isbn, title, price]**. Hozza létre a táblát!

MyBatis keretrendszer használatával valósítsa meg az alábbi funkciókat:

1. Új rekord felvitele
2. Felvitt rekord módosítása (pl. ár módosítása megadott isbn-re vonatkozóan)
3. Rekord törlése
4. Összes könyv lekérdezése
5. 5000-nél drágább könyvek lekérdezése

2. feladat: Tekintsük az alábbi ER modellt



Hozza létre a *dolgozo*, *kepzes* táblát, valamint az összerendelésükhöz szükséges táblát!

MyBatis keretrendszer használatával valósítsa meg az alábbi funkciókat:

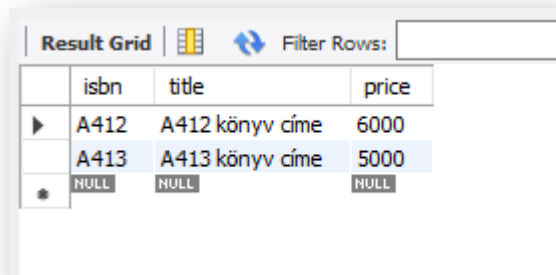
1. adatfelvitel a *dolgozo* és a *kepzes* táblákba
2. dolgozók és képzések összerendelése
3. dolgozó fizetésének módosítása
4. dolgozó beosztásának módosítása
5. dolgozó törlése
6. képzés időtartamának módosítása
7. képzés oktatójának módosítása
8. képzés törlése
9. összes dolgozó és képzés lekérdezése
10. paraméterként megadott fizetésnél többet kereső dolgozók lekérdezése
11. paraméterként megadott időtartamnál rövidebb idejű képzések lekérdezése
12. paraméterként megadott oktatóhoz tartozó képzések lekérdezése
13. paraméterként megadott fizetésnél magasabb fizetésű dolgozók által teljesített képzések lekérdezése
14. paraméterként megadott beosztású dolgozók közül ki milyen képzésen vett részt

## Az 1. feladat elkészítésének lépései

Az 1. feladat vonatkozásában először MySQL Workbench segítségével létrehoztam az adatbázist, majd a sémának megfelelő *books* táblát. Ezután 2 db rekordot felvittem:

1. **CREATE DATABASE** book;
2. **USE** book;
- 3.
4. **CREATE TABLE** books (isbn **VARCHAR**(14) **PRIMARY KEY**, title **VARCHAR**(50), price **INT**);
- 5.
6. **INSERT INTO** books **VALUES** ('A412', 'A412 könyv címe', 6000);
7. **INSERT INTO** books **VALUES** ('A413', 'A413 könyv címe', 5000);

A felvitt rekordokat lekérdezve MySQL Workbench-ben az alábbi eredmény látható:



	isbn	title	price
▶	A412	A412 könyv címe	6000
	A413	A413 könyv címe	5000
*	NULL	NULL	NULL

Ezt követően az Eclipse integrált fejlesztői környezet használatával létrehoztam a feladatnak megfelelő JAVA projektet. A feladathoz 4 db osztályt definiáltam:

1. MyBatisUtil: a myBatis SQL session kiépítését végzi el
2. Book: a feladatkirásnak megfelelő Book táblát reprezentálja, továbbá az attribútumokhoz getter és setter metódusokat tartalmaz
3. BookDAO: megteremti a kapcsolatot az adatbázissal
4. App: futtatható osztály main függvénnnyel, a különböző részfeladatok érdemben itt kerültek megoldásra.

A fenti osztályok forráskódjai a kurzus repositoryja alatt található alábbi jegyzékben érhetők el és tekinthetők meg: *B5T3ZN\_210302\myBatis\_book\src\main\java\mybatis*

Az alkalmazás működéséhez szükséges a myBatis keretrendszert is megfelelően konfigurálni. Ezt először a *B5T3ZN\_210302\myBatis\_book\src\main\resources* jegyzék alatti *mybatis-config.xml* állományban kell megtenni:

1. `<?xml version="1.0" encoding="UTF-8"?>`
2. `<!DOCTYPE configuration`
3. `PUBLIC "-//mybatis.org//DTD Config 3.0//EN"`
4. `"http://mybatis.org/dtd/mybatis-3-config.dtd">`
5. `<configuration>`
6. `<typeAliases>`
7. `<typeAlias type="mybatis.Book" alias="book"/>`

```

8.         </typeAliases>
9.         <environments default="development">
10.             <environment id="development">
11.                 <transactionManager type="JDBC"/>
12.                 <dataSource type="POOLED">
13.                     <property name="driver"
value="com.mysql.cj.jdbc.Driver"/>
14.                     <property name="url"
value="jdbc:mysql://localhost:3306/book"/>
15.                     <property name="username"
value="root"/>
16.                     <property name="password"
value="root"/>
17.                 </dataSource>
18.             </environment>
19.         </environments>
20.         <mappers>
21.             <mapper resource="mybatis/BookMapper.xml"/>
22.         </mappers>
23.     </configuration>

```

Ebben a fájlban a 13-16, továbbá a 21. sor szorul kisebb magyarázatra.

A 13. sorban a drivert kell megadni, a 14. sorban pedig azt az erőforrás azonosítót (linket), ahol az adatbázisszerver fut. A 15-16. sorban az adatbázishoz való kapcsolódáshoz szükséges felhasználónevet és jelszót kell megadni, míg a 21. sorban a feladathoz tartozó mapper fájl elérési útvonala került megadásra.

A *BookMapper.xml*-ben előbb a feladatban kikötött sémának megfelelő attribútumokat definiáltam a *<resultMap>* tag-en belül (6-10. sor), majd a különböző adatbázisműveletek SQL utasításait írtam meg a megfelelő tag-ek közé a szükséges attribútumok megadása mellett (12-30. sor)

```

1. <?xml version="1.0" encoding="UTF-8" ?>
2. <!DOCTYPE mapper
3.     PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
4.     "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
5. <mapper namespace="mybatis.BookMapper">
6.     <resultMap type="book" id="bookResult" >
7.         <id property="isbn" column="isbn" />
8.         <result property="title" column="title" />
9.         <result property="price" column="price" />
10.    </resultMap>
11.
12.    <insert id="insertBook" parameterType="book"
keyProperty="isbn">
13.        INSERT INTO books(isbn, title, price)
VALUES(#{isbn}, #{title}, #{price})
14.    </insert>

```

```

15.
16.         <update id="updateBook" parameterType="book">
17.             UPDATE books SET price = #{price} WHERE isbn =
               #{isbn}
18.         </update>
19.
20.         <delete id="deleteBook" parameterType="string">
21.             DELETE FROM books WHERE isbn = #{isbn}
22.         </delete>
23.
24.         <select id="selectBooks" resultType="book"
               resultMap="bookResult">
25.             SELECT isbn, title, price FROM books WHERE
               price > 5000
26.         </select>
27.
28.         <select id="findAllBooks" resultType="book"
               resultMap="bookResult">
29.             SELECT isbn, title, price FROM books
30.         </select>
31.
32.     </mapper>

```

## Az 1. feladat futtatásának eredménye

A feladatkírást a program a futtatáskor automatikusan teljesíti. Ekkor az alábbi kimenet áll elő:

```

Tárolva: ISBN: 759, TITLE: Könyv 8809, PRICE: 2096

Módosítva: ISBN: A412, PRICE: 7700

Deleted.

5000 ár feletti könyvek:
ISBN: A412, TITLE: A412 könyv címe, PRICE: 7700

Összes könyv:
ISBN: 759, TITLE: Könyv 8809, PRICE: 2096
ISBN: A412, TITLE: A412 könyv címe, PRICE: 7700

```

## A 2. feladat elkészítésének lépései

Az előző feladathoz hasonló lépésekben végeztem el a 2. feladat megoldását. Először MySQL Workbench segítségével létrehoztam az adatbázist, majd a sémának megfelelő táblákat, illetve felvittem néhány rekordot, valamint dolgozókat és képzéseket rendelttem össze:

```
1. CREATE DATABASE dolgozodb;
2. USE dolgozodb;
3.
4. CREATE TABLE dolgozo(did INT PRIMARY KEY, dnev VARCHAR(255),
    beosztas VARCHAR(255), fizetes INT);
5. CREATE TABLE kepzes(kid INT PRIMARY KEY, knev VARCHAR(255),
    oktato VARCHAR(255), idotartam INT);
6. CREATE TABLE dolgkepzes(id INT PRIMARY KEY, did INT REFERENCES
    dolgozo(did), kid INT REFERENCES kepzes(kid));
7.
8. -- dolgozo(did, dnev, beosztas, fizetes)
9. INSERT INTO dolgozo VALUES(100, 'Dolgozó 1', 'beosztás 1',
    620000);
10.    INSERT INTO dolgozo VALUES(101, 'Dolgozó 2', 'beosztás
    2', 320000);
11.    INSERT INTO dolgozo VALUES(102, 'Dolgozó 3', 'beosztás
    3', 460000);
12.    INSERT INTO dolgozo VALUES(103, 'Dolgozó 4', 'beosztás
    2', 200000);
13.    INSERT INTO dolgozo VALUES(104, 'Dolgozó 5', 'beosztás
    2', 770000);
14.    INSERT INTO dolgozo VALUES(105, 'Dolgozó 6', 'beosztás
    3', 180000);
15.    INSERT INTO dolgozo VALUES(106, 'Dolgozó 7', 'beosztás
    1', 930000);
16.
17.    -- kepzes(kid, knev, oktato, idotartam)
18.    INSERT INTO kepzes VALUES(100, 'képzés 1', 'oktató 1',
    60);
19.    INSERT INTO kepzes VALUES(101, 'képzés 2', 'oktató 2',
    180);
20.    INSERT INTO kepzes VALUES(102, 'képzés 2', 'oktató 3',
    300);
21.
22.    INSERT INTO dolgkepzes VALUES(1, 100, 100); -- dolgozó1
    részt vesz képzés1-en
23.    INSERT INTO dolgkepzes VALUES(2, 100, 101); -- dolgozó1
    részt vesz képzés2-n
24.    INSERT INTO dolgkepzes VALUES(3, 102, 100); -- dolgozó3
    részt vesz képzés1-en
25.    INSERT INTO dolgkepzes VALUES(4, 103, 100); -- dolgozó4
    részt vesz képzés1-en
```

```

26.      INSERT INTO dolgkepzes VALUES(5, 106, 102); -- dolgozó7
        részt vesz képzés3-on
27.      INSERT INTO dolgkepzes VALUES(6, 103, 101); -- dolgozó4
        részt vesz képzés2-n

```

Ezután az Eclipse integrált fejlesztői környezet használatával létrehoztam a feladatnak megfelelő JAVA projektet. A feladathoz 7 db osztályt definiáltam:

1. MyBatisUtil: a myBatis SQL session kiépítését végzi el.
2. DAO: megteremti a kapcsolatot az adatbázissal.
3. Dolgozo: a feladatkiírásnak megfelelő dolgozo táblát reprezentálja, továbbá az attribútumokhoz getter és setter metódusokat tartalmaz.
4. Kepzes: a feladatkiírásnak megfelelő kepzes táblát reprezentálja, továbbá az attribútumokhoz getter és setter metódusokat tartalmaz.
5. Kozos: együttesen reprezentálja a két táblát, azon műveleteknél használatos, amelyeknél a két tábla együttes kezelése szükséges.
6. DolgKepzOsszerendeles: a dolgozók és képzések összerendelését reprezentálja.
7. App: futtatható osztály main függvénnnyel, a különböző részfeladatok érdemben itt kerültek megoldásra.

A fenti osztályok forráskódjai a kurzus repositoryja alatt található alábbi jegyzékben érhetők el és tekinthetők meg:

*B5T3ZN\_210302\myBatis\_Dolgozo\src\main\java\myBatis*

Az alkalmazás működéséhez szükséges myBatis keretrendszer megfelelő konfigurálása túlnyomórészt hasonló az 1. feladatban már ismertetettekhez. A lényegi különbség az, hogy nem 1 db mapper XML fájlt használtam, hanem 4 db-ot:

1. DolgozoMapper.xml
2. KepzesMapper.xml
3. KozosMapper.xml
4. Osszerendeles.xml

A *mybatis-config.xml*-ben az érdemi eltérés a `<typeAliases>` és a `<mappers>` tagek tartalmában áll fenn, amelyek a 4 db mapper fájl és típus okán az alábbiak szerint épülnek fel:

```

1. <typeAliases>
2.      <typeAlias type="myBatis.Dolgozo" alias="dolgozo"/>
3.      <typeAlias type="myBatis.Kepzes" alias="kepzes"/>
4.      <typeAlias type="myBatis.Kozos" alias="Kozos" />
5.      <typeAlias type="myBatis.DolgKepzOsszerendeles"
        alias="Osszerendeles" />
6. </typeAliases>
7. ...
8. <mappers>
9.      <mapper resource="mybatis/DolgozoMapper.xml"/>
10.     <mapper resource="mybatis/KepzesMapper.xml" />
11.     <mapper resource="mybatis/KozosMapper.xml" />
12.     <mapper resource="mybatis/Osszerendeles.xml" />

```

13.       </mappers>

A 4 db mapper XML a kurzus repositoryja alatti *B5T3ZN\_210302\myBatis\_Dolgozo\src\main\resources\myBatis* jegyzékben található meg. Ezek felépítése hasonló az 1. feladathoz, eltérés lényegében az SQL utasításokban van.

## A 2. feladat futtatásának eredménye

Az alkalmazás elindítását követően egy menürendszer kerül megjelenítésre, ahol a feladatkiírásban megkövetelt különböző funkciók egy számjegy kiválasztásával hozhatók működésbe. Ezek a menüpontok nem feltétlenül egyeznek meg a feladatkiírás vonatkozó alpontjaival:

```
Válasszon műveletet!
1      dolgozó felvitele
2      dolgozó fizetésének módosítása
3      dolgozó beosztásának módosítása
4      dolgozó törlése
5      képzés felvitele
6      képzés hosszának módosítása
7      képzés oktatójának módosítása
8      képzés törlése
9      összes dolgozó lekérdezése
10     összes képzés lekérdezése
11     paraméternél magasabb fizetésű dolgozók lekérdezése
12     rövidebb idejű képzések lekérdezése
13     paraméterként megadott oktatóhoz tartozó képzések
14     paraméterként megadott fizetésnél magasabb fizetésű dolgozók milyen képzésben vettek részt
15     paraméterként megadott beosztásúak közül ki milyen képzésben vett részt
16     dolgozó és képzés összerendelése
0      kilépés
```

### 1. Dolgozó felvitele:

```
1
Dolgozó ID-ja:
900
Dolgozó neve:
xy
Dolgozó beosztása:
vezető
Dolgozó fizetése:
600000
Dolgozó tárolva a következő adatokkal: ID: 900, Név: xy, Beosztás: vezető, Fizetés: 600000
```

### 2. A dolgozó fizetésének módosítása:

```
0      kilépés
2
Dolgozó ID-ja:
900
Dolgozó új fizetése:
550000
Dolgozó fizetése módosítva: ID: 900, új fizetés: 550000
Válasszon műveletet!
```



### 3. A dolgozó beosztásának módosítása:

```
3
Dolgozó ID-ja:
900
Dolgozó új beosztása:
csoportvezető
Dolgozó beosztása módosítva: ID: 900, új beosztás: csoportvezető
Válasszon műveletet!
```

### 4. Dolgozó törlése:

```
0 kilepes
4
Törlendő dolgozó ID-ja:
102
Törölve.
Válasszon műveletet!
```

### 5. Képzés felvitele:

```
0 kilepes
5
Képzés ID-ja:
900
Képzés neve:
képzés900
Képzés oktatója:
oktató
Képzés időtartama:
120
Képzés tárolva a következő adatokkal: ID: 900, Név: képzés900, Oktató: oktató, Időtartam: 120
Válasszon műveletet!
```

### 6. Képzés hosszának (időtartamának módosítása):

```
0 kilepes
6
Képzés ID-ja:
900
Képzés új időtartama:
180
Képzés időtartama módosítva: ID: 900, új időtartam: 180
Válasszon műveletet!
```

### 7. Képzés oktatójának módosítása:

```
Képzés ID-ja:
900
Képzés új oktatója:
Kiss_István
Képzés oktatója módosítva: ID: 900, új oktató: Kiss_István
Válasszon műveletet!
```

## 8. Képzés törlése:

```
8
Törlendő képzés ID-ja:
104
Törölve.
```

**9. Összes dolgozó lekérdezése:** ennek során látható a korábban felvitt, majd a fizetés és beosztás tekintetében is módosított 900 ID-val rendelkező dolgozó is, továbbá a 102 ID dolgozó nem található, hiszen törlésre került a 4. pontban.

```
9
Összes dolgozó:
ID: 100, Név: Dolgozó 1, Beosztás:beosztás 1, Fizetés: 620000
ID: 101, Név: Dolgozó 2, Beosztás:beosztás 2, Fizetés: 320000
ID: 103, Név: Dolgozó 4, Beosztás:beosztás 2, Fizetés: 200000
ID: 104, Név: Dolgozó 5, Beosztás:beosztás 2, Fizetés: 770000
ID: 105, Név: Dolgozó 6, Beosztás:beosztás 3, Fizetés: 180000
ID: 106, Név: Dolgozó 7, Beosztás:beosztás 1, Fizetés: 930000
ID: 900, Név: xy, Beosztás:csoportvezető, Fizetés: 550000
Válasszon műveletet!
```

**10. Összes képzés lekérdezése:** a 9. ponthoz hasonlóan láthatók az oktatók.

```
10
Összes képzés:
ID: 100, Név: képzés 1, Oktató: oktató 1, Időtartam: 60
ID: 101, Név: képzés 2, Oktató: oktató 2, Időtartam: 180
ID: 102, Név: képzés 2, Oktató: oktató 3, Időtartam: 300
ID: 900, Név: képzés900, Oktató: Kiss_István, Időtartam: 180
Válasszon műveletet!
```

**11. Paraméternél (jelenleg megadott: 500000) magasabb fizetésű dolgozók lekérdezése:**

```
11
Adja meg azt az értéket, amelynél magasabb fizetésű dolgozók nevét kéri listázni:
500000
Név: Dolgozó 1, Fizetés: 620000
Név: Dolgozó 5, Fizetés: 770000
Név: Dolgozó 7, Fizetés: 930000
Név: xy, Fizetés: 550000
Válasszon műveletet!
```

## 12. Paraméternél (jelenleg megadott: 120) rövidebb idejű képzések lekérdezése:

```
12
0      kilépés
12
Adja meg azt az értéket, amelynél rövidebb idejű képzések nevét kéri listázni:
120
Név: képzés 1, Időtartam: 60
Válasszon műveletet!
```

## 13. Paraméterként megadott oktatóhoz tartozó képzések lekérdezése:

```
13
Adja meg azt az oktatót, akihez tartozó képzések nevét kéri listázni:
Kiss_István
Név: képzés900, Oktató: Kiss_István
Válasszon műveletet!
```

## 14. Paraméternél magasabb fizetésű dolgozók lekérdezése:

```
14
Adja meg azt a fizetésértéket, amelynél magasabb fizetésűek által elvégzett képzések listáját kéri.
500000
Dolgozó neve: xy, képzés neve: képzés 1, dolgozó fizetése: 550000
Dolgozó neve: Dolgozó 7, képzés neve: képzés 1, dolgozó fizetése: 930000
Dolgozó neve: Dolgozó 5, képzés neve: képzés 1, dolgozó fizetése: 770000
Dolgozó neve: Dolgozó 1, képzés neve: képzés 1, dolgozó fizetése: 620000
```

## 15. Paraméterben megadott beosztású dolgozók által elvégzett képzések:

```
15
Adja meg azt a beosztást, amelyhez kapcsolódóan a képzések listázását kéri
csoportvezető
Dolgozó neve: xy, képzés neve: képzés 1
Dolgozó neve: xy, képzés neve: képzés 2
Dolgozó neve: xy, képzés neve: képzés 2
Dolgozó neve: xy, képzés neve: képzés900
```

## 16. Dolgozók és képzések összerendelése: összerendelem a korábban felvitt 900 ID dolgozót a 900 ID képzéssel.

```
16
Adja meg a dolgozó ID-ját, akit képzéshez szeretne rendelni:
900
Adja meg a képzés ID-ját, amelyhez 900 ID-jű dolgozót szeretné rendelni.
900
Összerendelve.
```