

# JEGYZŐKÖNYV

Adatbázis rendszerek II.

Féléves feladat

Készítette: **Mihály Gergő**

Neptunkód: **OY6ICJ**

Dátum: **2025. 05. 12.**

## **A feladat leírása**

Egy Java nyelven írt alkalmazás, és egy PL/SQL nyelven írt program elkészítése egy relációs adatbázishoz. A relációs adatbázis egy egyetemi nyilvántartó rendszert modellez. Három darab táblából áll, az elsőben (Student tábla) a hallgatók vannak, neptun kóddal, névvel, a már teljesített kredit számukkal, a születési idejükkel és a email címükkel, a másodikban (Course tábla) a felvehető tantárgyak vannak névvel, a kredit számukkal, a vizsga időpontjával, és azzal, hogy kötelező tárgy-e. A harmadik tábla (Has tábla) egy kapcsoló tábla, azt írja le, hogy melyik hallgató melyik tárgyakat vette fel a félévre.

## **A táblákat létrehozó SQL parancsok**

### **Course tábla:**

```
CREATE TABLE Course(  
    id INTEGER(2) PRIMARY KEY,  
    name CHAR(30) NOT NULL,  
    credit INTEGER(2) NOT NULL,  
    exam_date DATE NOT NULL,  
    required INTEGER(1) NOT NULL,  
    CHECK(required = 1 OR required = 0)  
);
```

### **Student tábla:**

```
CREATE TABLE Student(  
    neptun CHAR(6) PRIMARY KEY,  
    name CHAR(30) NOT NULL,  
    credit_sum INTEGER(2) NOT NULL,  
    date_of_birth DATE NOT NULL,  
    email CHAR(30) NOT NULL  
);
```

**Has tábla:**

```
CREATE TABLE Has(  
    neptun CHAR(6),  
    id INTEGER(2),  
    FOREIGN KEY(neptun) REFERENCES Student(neptun),  
    FOREIGN KEY(id) REFERENCES Course(id)  
);
```

**A táblákat feltöltő SQL parancsok****Course tábla:**

```
INSERT INTO Course VALUES(1, 'Analízis 1.', 5, '2025-06-08', 1);  
INSERT INTO Course VALUES(2, 'Lineáris Algebra', 6, '2025-06-18', 1);  
INSERT INTO Course VALUES(3, 'Villámvédelem', 2, '2025-06-20', 0);  
INSERT INTO Course VALUES(4, 'Programozás', 7, '2025-06-24', 1);  
INSERT INTO Course VALUES(5, 'Fizika', 3, '2025-06-27', 1);
```

**Student tábla:**

```
INSERT INTO Student VALUES('m2fq5', 'Kis Pista', 30, '1985-01-20',  
'kispista@gmail.com');  
INSERT INTO Student VALUES('8jq3qn', 'Nagy Lajos', 83, '1985-04-25',  
'nagylajos@gmail.com');  
INSERT INTO Student VALUES('nhsbeg', 'Kovacs Imre', 15, '1989-11-  
03', 'kovacsimre@gmail.com');  
INSERT INTO Student VALUES('xs0tfa', 'Orosz Vince', 38, '1991-01-21',  
'oroszvince@gmail.com');  
INSERT INTO Student VALUES('2bk1k6', 'Gáspár Ödön', 54, '1998-09-  
23', 'gasparodon@gmail.com');
```

**Has tábla:**

```
INSERT INTO Has VALUES('m2fq5', 4);  
INSERT INTO Has VALUES('m2fq5', 5);
```

```
INSERT INTO Has VALUES('8jq3qn', 5);
INSERT INTO Has VALUES('8jq3qn', 4);
INSERT INTO Has VALUES('nhsbeg', 2);
INSERT INTO Has VALUES('nhsbeg', 1);
INSERT INTO Has VALUES('nhsbeg', 5);
INSERT INTO Has VALUES('xs0tfa', 1);
INSERT INTO Has VALUES('2bk1k6', 3);
INSERT INTO Has VALUES('2bk1k6', 2);
```

### A JDBC program

A program egy konzolos applikáció, egy menürendszerben lehet navigálni a funkciók között.

### A program főmenüje:

```
=====
Main menu:
(pick one by typing it's number)
0. Exit
1. Display table
2. Insert record
3. Modify record
4. Delete record
5. Get
6. Insert records from file (csv file, without headers)
=====
```

### “1. Display table“:

A táblák teljes tartalmát lehet lekérdezni vele.

### A menüje:

```
=====
Select the table:
(pick one by typing it's number)
0. Student table
1. Course table
2. Has table
=====
```

### A Course tábla lekérdezése:

id(number)	name(text)	credit(number)	exam_date(date)	required(boolean)
1	Analízis 1.	5	2025-06-08	1
2	Lineáris Algebra	6	2025-06-18	1
3	Villámvédelem	2	2025-06-20	0
4	Programozás	7	2025-06-24	1
5	Fizika	3	2025-06-27	1

## “2. Insert record”:

Egy rekordot lehet felvinni vele a táblákba.

**A Student táblára az adatok megadása:**

```
Student
Neptun:
SOAH38
Name:
Horvat Bela
Credit Sum:
67
Date Of Birth:
1999-10-03
Email:
horvatbela@gmail.com
Insert successful!
```

**A felvitel után a Student tábla:**

neptun(text)	name(text)	credit_sum(number)	date_of_birth(date)	email(text)
m2fqc5	Kis Pista	30	1985-01-20	kispista@gmail.com
8jq3qn	Nagy Lajos	83	1985-04-25	nagylajos@gmail.com
nhsbeg	Kovacs Imre	15	1989-11-03	kovacsimre@gmail.com
xs0tfa	Orosz Vince	38	1991-01-21	oroszvince@gmail.com
2bk1k6	Gáspár Ödön	54	1998-09-23	gasparodon@gmail.com
SOAH38	Horvat Bela	67	1999-10-03	horvatbela@gmail.com

## “3. Modify record”:

A táblák rekordjait lehet vele módosítani.

**A Student táblára:**

```

Neptun:
SOAH38

=====
Select the field to update:
(pick one by typing it's number)
 0. name
 1. credit_sum
 2. date_of_birth
 3. email
=====
1
credit_sum:
89
Update succesful!

```

**A Student tábla a módosítás után:**

Student				
neptun(text)	name(text)	credit_sum(number)	date_of_birth(date)	email(text)
m2fqc5	Kis Pista	30	1985-01-20	kispista@gmail.com
8jq3qn	Nagy Lajos	83	1985-04-25	nagylajos@gmail.com
nhsbeg	Kovacs Imre	15	1989-11-03	kovacsimre@gmail.com
xs0tfa	Orosz Vince	38	1991-01-21	oroszvince@gmail.com
2bk1k6	Gáspár Ödön	54	1998-09-23	gasparodon@gmail.com
SOAH38	Horvat Bela	89	1999-10-03	horvatbela@gmail.com

#### “4. Delete record”:

A táblák rekordjait lehet törölni vele.

**A Has táblára:**

```

Neptun:
2bk1k6
Id:
2
Delete successful!

```

### A Has tábla a törlés után:

Has	
neptun(text)	id(number)
m2fqc5	4
m2fqc5	5
8jq3qn	5
8jq3qn	4
nhsbeg	2
nhsbeg	1
nhsbeg	5
xs0tfa	1
2bk1k6	3

### “5. Get”:

5 darab előre definiált lekérdezést lehet vele megvalósítani.

#### A lekérdezések:

```
=====
Select one of the options:
(pick one by typing it's number)
0. Student name and email by neptun
1. Student names by date of birth (born after)
2. Courses by neptun and if it is required
3. Exam date and credit by neptun
4. Credit by course name
=====
```

### “2. Courses by neptun and if it is required”:

```
Neptun:
nhsbeg
Required:
1
Lineáris Algebra
Analízis 1.
Fizika
```

## A JDBC program extra funkciói

### “6. Insert records from file”:

Több rekord felvitelét valósítja meg. A bemeneti fájl egy fejléc nélküli csv fájl, ennek az útvonalát meg kell adni a programnak. A program mellett található egy testInputs nevű mappa, amiben mindhárom táblára találhatóak bemeneti fájlok.

#### A course.csv fájl tartalma:

```
6;Testnevelés;1;2025-06-09;0
7;Szigorlat;10;2025-06-29;1
```

#### A Course tábla a felvitel után:

course				
id(number)	name(text)	credit(number)	exam_date(date)	required(boolean)
1	Analízis 1.	5	2025-06-08	1
2	Lineáris Algebra	6	2025-06-18	1
3	Villámvédelem	2	2025-06-20	0
4	Programozás	7	2025-06-24	1
5	Fizika	3	2025-06-27	1
6	Testnevelés	1	2025-06-09	0
7	Szigorlat	10	2025-06-29	1

### Anomália kezelés:

A törlési műveletnél csak akkor enged a program törölni egy rekordot, ha nincs rá hivatkozás a Has kapcsolótáblában.

#### Programüzenet, a nem megengedett törlési műveletnél:

```
Neptun:
m2fqc5
Has table has atleast one reference to neptun m2fqc5!
```

### Logolás:

A programnak indításakor meg lehet adni, hogy debug, vagy felhasználói módban fusson (ha paraméterként meg van adva, hogy “debug”, akkor debug módban indul, egyébként felhasználói módban).

#### Felhasználói mód:

Ebben a módban csak a hibákat logolja a program a képernyőre, pl.: ha a program nem találja meg a JDBC drivert, vagy egy SQL parancs nem tudott lefutni.

#### Log, ha nem találja a program a JDBC drivert:



```
máj. 12, 2025 4:35:32 DU application.Application main  
SEVERE: JDBC driver could not be found, aborting!
```

### Debug mód:

Ebben a módban minden SQL műveletről logot készít a program, akár sikeres, akár nem.

### Log, egy sikeres lekérdezésről:

```
máj. 12, 2025 4:47:57 DU db.DbUtils selectAllStudent  
INFO: Select successful: SELECT * FROM Student
```

## A PL/SQL program

A program 3 csomagból épül fel, minden táblához tartozik egy, ebben vannak megvalósítva a funkciók.

### Adatok felvitele

Mindhárom csomagban implementált, a New nevű függvénnyel érhető el.

#### Adatfelvitel hívása a Student táblára:

```
BEGIN  
  StudentPackage.New('SOAH38', 'Horvát Béla', 67, '10/03/1999', 'horvatbela@gmail.com');  
END;
```

#### A Student tábla adat felvitel után:

NEPTUN	NAME	CREDIT_SUM	DATE_OF_BIRTH	EMAIL
m2lqc5	Kis Pista	30	1/20/1985	kispista@gmail.com
8jq3qn	Nagy Lajos	83	4/25/1985	nagylojos@gmail.com
nhsbeg	Kovacs Imre	15	3/11/1989	kovacsimre@gmail.com
xs0rla	Orosz Vince	38	1/21/1991	oroszvince@gmail.com
zbtkl6	Gáspár Odón	54	9/23/1998	gasparodon@gmail.com
SOAH38	Horvát Béla	67	10/3/1999	horvatbela@gmail.com

### Adatok módosítása

Csak a fő táblákra implementált, a Mod függvénnyel érhető el.

#### Adatmódosítás hívása a Course táblára:

```
BEGIN
    CoursePackage.Mod(1, '06/11/2025');
END;
```

**A Course tábla a módosítás után:**

ID	NAME	CREDIT	EXAM_DATE	REQUIRED
1	Analízis 1.	5	6/11/2025	1
2	Lineáris Algebra	6	6/18/2025	1
3	Villámvédelem	2	6/20/2025	0
4	Programozás	7	6/24/2025	1
5	Fizika	3	6/27/2025	1

## Adatok törlése

Mind három táblára implementálva, a Del függvénnyel érhető el.

**Adattörlés hívása a Has táblára:**

```
BEGIN
    HasPackage.Del('2bk1k6', 2);
END;
```

**A Has tábla a törlés után:**

NEPTUN	ID
m2lq5	4
m2lq5	5
8jq5qn	5
8jq5qn	4
nhsbeg	2
nhsbeg	1
nhsbeg	5
xs0tfa	1
zbk1k6	3

## Rekord lekérdezése

Csak a fő táblákra implementált, a GetStudent és a GetCourse függvényekkel érhető el.

**Rekord lekérdezés eredménye a Student táblára:**

```
Neptun: SOAH38
Name: Horvát Béla
Credit sum: 67
Date of birth: 10/3/1999
Email: horvatbela@gmail.com
```

**Rekord lekérdezés eredménye a Course táblára:**

```
Id: 4
Name: Programozás
Credit: 7
Exam date: 6/24/2025
Required: 1
```

## Aggregált lekérdezés

Csak a fő táblákra implementált

### Student tábla

A GetFailingStudents függvénnyel érhető el, kiírja mennyi hallgató áll bukásra ( $\text{credit\_sum} < 50$ ) és kik azok a hallgatók.

#### A lekérdezés eredménye:

```
3 students are failing:
Name: Orosz Vince, Credit sum: 38
Name: Kis Pista, Credit sum: 30
Name: Kovacs Imre, Credit sum: 15
```

### Course tábla:

A GetAvgCreditOfRequiredCourses függvénnyel érhető el, kiírja a kötelező tárgyak kredit átlagát, és a kötelező tárgyakat.

#### A lekérdezés eredménye:

```
The average credit is: 5.25
Name: Programozás, Credit: 7
Name: Lineáris Algebra, Credit: 6
Name: Analízis 1., Credit: 5
Name: Fizika, Credit: 3
```

## Trigger módosítás naplózására:

A két fő táblára van implementálva, INSERT, UPDATE és DELETE műveleteket naplóznak egy Logs nevű táblába. A Student táblában a credit\_sum mező van naplózva, a Course táblában pedig az exam\_date.

### A Student tábla 3 módosítása után a Logs tábla:

TABLE_NAME	EVENT	CH	T
Student	Insert	fültj U 67	5/12/2025
Student	Update	fültj U 67->90	5/12/2025
Student	Delete	fültj U 90	5/12/2025

## A Course tábla 3 módosítása után a Logs tábla:

TABLE_NAME	EVENT	CH	T
Course	Insert	7 U 6/29/2025	5/12/2025
Student	Insert	fültj U 67	5/12/2025
Course	Update	7 U 6/29/2025->6/30/2025	5/12/2025
Student	Update	fültj U 67->90	5/12/2025
Student	Delete	fültj U 90	5/12/2025
Course	Delete	7 U 6/30/2025	5/12/2025

## Trigger automatikus kulcsérték megadásra

A Course táblára van implementálva.

### Adatfelvitel hívása a Course táblára:

```
BEGIN
    CoursePackage.New(0, 'Szigorlat', 10, '06/29/2025', 1);
END;
```

### A Course tábla az adatfelvitel után:

ID	NAME	CREDIT	EXAM_DATE	REQUIRED
6	Testnevelés	1	6/9/2025	0
1	Analízis I.	5	6/11/2025	1
2	Lineáris Algebra	6	6/18/2025	1
3	Villámvédelem	2	6/20/2025	0
4	Programozás	7	6/24/2025	1
5	Fizika	3	6/27/2025	1
7	Szigorlat	10	6/29/2025	1

## Trigger a módosítás kontrollálására

A Course táblában van implementálva az exam\_date mezőre, nem engedi az aznapitól korábbi időpont felvitelét.

### Adatfelvitel hívása a Course táblára:

```
BEGIN
    CoursePackage.New(0, 'Szigorlat', 10, '06/29/2024', 1);
END;
```

### A hibaüzenet:

ORA-20010: The exam date cannot be earlier than today!