

**Adatbázis rendszerek 2, Levelező**  
**GEIAL323(-)BL**  
**Gépészmérnöki és Informatikai Kar, BSc, Levelező**

A tárgy előadója, leckekönyvi jegyzője: Szűcs Miklós  
A tárgy lezárásának módja: aláírás és vizsga  
Kredit: 5  
Óraszám: 8 óra előadás + 8 óra labor gyakorlat

## ÜTEMTERV

**A tárgy célja:** A tárgy célja az adatbázis kezelő rendszerek belső motorjainak a megismerése. A tranzakció kezelés alapjai; az aktív adatbázis elemek bemutatása, tárolt eljárások készítése, Kliens API alapjai. A tárgy keretében a hallgatók megismerik az adatbázis felhasználásra épülő alkalmazások fejlesztését, az adatbázis belső programozási technikáit.

1. alkalom (4 óra)	Előadás	A félév követelményrendszerének ismertetése. Adatbázis kezelő API rendszerek programozása. Java JDBC API programozása. A tárolt eljárások programozása. PL/SQL nyelv alapjai.
2. alkalom (4 óra)	Gyakorlat	Teremhasználati szabályok, rövid munkavédelmi oktatás. Tárolt eljárások, triggerek, job-ok készítése.
3. alkalom (4 óra)	Gyakorlat	A JDBC programozás alapjai: kapcsolódás, statikus parancskiadás, eredményhalmaz feldolgozása, adatok megjelenítése.
4. alkalom (4 óra)	Előadás	A relációs adatbáziskezelők motorjainak komponensei, a tranzakció kezelés szerepe, tranzakció és history fogalma és elemei. A zárolás alapú tranzakció kezelés.

### Pontrendszer:

A félév értékelése a gyűjtött pontok alapján történik. Pontot érő tevékenységek (maximális és minimum elérendő pontszámok):

- A féléves feladat elkészítése: 40/20 pont
- Zárthelyi dolgozat: 20/11 pont

### Az aláírás feltételei:

Maximum 1 hiányzás az órákról, a zárthelyi dolgozat eredményes megírása (min. 11 pont), és a féléves egyéni feladat legalább elégséges szintű elkészítése (min. 20 pont).

### Az aláírás pótlása írásbeli és gyakorlati részből áll:

- Kétszintű (beugró teszt + kifejtős kérdések) írásbeli,
- Gyakorlati feladat gépteremben

**Féléves egyéni feladat:** alkalmazásfejlesztés

A feladat két részből áll:

1. A megkapott szerkezetű tábla létrehozása Oracle rendszerben. A megadott tárolt eljárások és függvények elkészítése.
2. A megkapott szerkezetű tábla létrehozása SQLite rendszerben. JDBC program megírása a tábla kezelésére (minimális funkciók: adatok beszúrása, módosítása, törlése, listázása)

A minimum feltételeknek eleget tevő alkalmazásra 25 pontot kaphat. Ha jobb jegyet szeretne, a következő lehetőségei vannak:

- Ha grafikus felületű JDBC alkalmazást készít, további 5 pontot kap.
- Ha nemcsak a megadott táblának, hanem egy ahhoz kapcsolódó másik táblának a kezelését is megvalósítja, további 5 pontot kaphat.
- Ha további funkciókkal látja el a programot, pl. keresés, statisztika, beléptetés-jogosultság ellenőrzés, további 5 pontot kaphat.

A feladat megvédése személyesen, a megadott szorgalmi héten, gyakorlaton történik. A feladat igényes megoldásával maximum 40 pont szerezhető.

### **A vizsga menete:**

A vizsga 60 perces kötelező írásbeli részből, és opcionális szóbeli részből áll. Az írásbeli részre maximum 20 pont kapható. A vizsgajegybe beszámít a féléves feladat pontszáma, így összesen  $20+40=60$  pont szerezhető. Az elért vizsgajegy a megszerzett pontok függvényében:

- 0 – 36 pont: elégtelen
- 37 – 42 pont: elégséges
- 43 – 48 pont: közepes
- 49 – 54 pont: jó
- 55 – 60 pont: jeles

### **Általános rendelkezések**

Az ME SzMSz III. kötet 38§ (6) pontja alapján, ha a hallgató nem igazolt hiányzása a gyakorlatokon eléri a gyakorlatok darabszámának 50%-át, a tantárgy aláírása nem szerezhető meg.

Az ME SzMSz III. kötet 96§ alapján a tárgyakhoz kapcsolódó valamennyi számonkérési alkalomnál a nem engedélyezett segédeszközök használata (puskázás) vagy más munkájának sajátként történő feltüntetése (plagizálás) fegyelmi vétségnek minősül, mely tanulmányi szankciókat vagy fegyelmi eljárást von maga után. Tanulmányi szankció az évközi számonkéréseknél a számonkérés sikertelen minősítése. A számonkérés ilyen esetekben nem pótolható.

Tanulmányi szankció a vizsgaidőszakban a vizsga elégtelen minősítése, és hogy ismételt vizsgát a hallgató a tanszék által kijelölt időpontban, kijelölt vizsgabizottság előtt, szóbeli vizsga formájában tehet.

A puskázás és/vagy plagizálás tényét a tanszék a hallgató tanulmányi ideje alatt nyilvántartja, és ismételt előfordulás esetén a ME SzMSz III. kötet 96§ által előírt fegyelmi eljárást kezdeményez.

**A tantárgy elektronikus jegyzete:** <http://users.iit.uni-miskolc.hu/~szucs/>

Érvényes: 2016. február 1.-től visszavonásig.

Miskolc, 2019. február 1.

Szűcs Miklós



## Minta ZH

### Kérdések:

1. Egy bemenő paraméter nélküli tárolt rutin visszaad egy értéket. Ez a rutin:

**A:** Tárolt eljárás

**B:** Tárolt függvény

**C:** Lehet tárolt eljárás és tárolt függvény is.

2. Tárolt rutinokban a változókat hol deklaráljuk?

**A:** A BEGIN előtt, a DECLARE után

**B:** A BEGIN előtt

**C:** A BEGIN után

**D:** Bárhol, a BEGIN előtt vagy után

3. Melyik kulcsszóval adjuk meg a tárolt eljárás visszatérő értékét?

**A:** retur

**B:** returs

**C:** return

**D:** returns

**E:** Tárolt eljárásnak nincs visszatérő értéke

4. Elvileg hogyan hívhatjuk meg a Maci() tárolt rutint? (2p)

**A:** PLEASE Maci();

**B:** begin Maci(); end;

**C:** RUN Maci();

**D:** select Maci from dual;

**E:** SELECT Maci();

**F:** CALL Maci;

**G:** SELECT Maci;

**H:** begin Maci; end;

5. Szintaktikailag helyes-e az alábbi kód:

**create function** novel5 (x in number) return

number as

begin

return x+5;

end;

**A:** IGEN

**B:** NEM

6. PLSQL-ben milyen szimbólumok jelzik az alábbiakat:

Utasítások vége	
Karakterlánc összefűzés	
Blokk neve	
Hiba típusú változó	
Értékadás változónak	

7. Milyen paranccsal lehet egy tárolt eljárásban kiírni egy változó értékét a képernyőre:

Megoldás: \_\_\_\_\_

Név: \_\_\_\_\_

Neptunkód: \_\_\_\_\_

0-31p-1,	42-46p-4	Pont	Jegy
32-36p-2,	47-51p-5		
37-41p-3,			

8. Milyen hibát dob a rendszer, ha egy select into utasítás egynél több adatot eredményez?

Megoldás: \_\_\_\_\_

9. Melyik művelet lehet egy trigger elindítója? (3p)

**A:** Create

**D:** Delete

**G:** Drop

**B:** Insert

**E:** Before

**H:** Select

**C:** Alter

**F:** Update

**I:** After

10. Mi a helyes műveleti sorrend egy kurzor használata során? Kezdje a számozást a legkorábban szükséges művelettel!

	FETCH
	OPEN
	DECLARE
	CLOSE

11. Mely kulcsszavak használhatók INSERT típusú triggernél?

**A:** NEW

**B:** OLD

**C:** Mindkettő

12. Írja be a hiányzó kifejezéseket! (5p)

CREATE TRIGGER mumus AFTER \_\_\_\_\_

\_\_\_\_\_ Termék \_\_\_\_\_

BEGIN

Insert \_\_\_\_\_ Naplo \_\_\_\_\_

(:OLD.AR || :NEW.AR);

END;

13. Az alábbi kód futtatásakor milyen típusú kurzor használat történik?

declare

l\_date date;

begin

select trunc(sysdate) into

l\_date from dual;

end;

Megoldás: \_\_\_\_\_

14. Melyik változó típus nem használható pl/sqlben?

**A:** char

**B:** varchar1

**C:** varchar2

**D:** integer

**E:** number

15. Milyen kulcsszó hiányzik a kódból?

```
IF foo IS NULL _____  
  NULL;  
END IF;
```

16. Milyen kulcsszó hiányzik a kódból?

```
IF foo = 1 THEN  
  l_text := 'A';  
_____ foo = 2 THEN  
  l_text := 'B';  
ELSE  
  l_text := 'C';  
END IF;
```

17. A következők közül melyik hibakód **nem létezik** pl/sql-ben?

- A:** no\_data\_found                   **B:** two\_many\_rows  
**C:** invalid\_number               **D:** others  
**E:** Mind létezik, mind használható

18. Hol kell deklarálni az explicit kurzort pl/sql-ben?

- A:** A pl/sql munkaterületen  
**B:** A pl/sql hibakezelő részben  
**C:** A pl/sql deklarációs részben  
**D:** A pl/sql végrehajtási részben  
**E:** Az explicit kurzort nem kell deklarálni

19. Melyik állítás helyes a **select into** kifejezésre?

- A:** Használatkor nem keletkezik hiba  
**B:** Ha több adatsort ad vissza, több változót kell hozzá használni  
**C:** Csak egy visszaadott adatsor esetén használható  
**D:** Egyik válasz sem helyes

20. A naplózó triggerek tipikusan milyen típusúak?

- A:** After  
**B:** Before  
**C:** After, for each row  
**D:** Before, for each row  
**E:** Instead of

21. Ha egy trigger feladata az, hogy adatmódosításkor csak bizonyos értékek kerülhessenek egy mezőbe, akkor milyen típusú triggeret kell készíteni?

- A:** After  
**B:** Before  
**C:** After, for each row  
**D:** Before, for each row  
**E:** Instead of

22. Töltse ki a triggerekkel kapcsolatos táblázatot! Jelölje be (x), ahol az adott kulcsszó használható! (4p)

	Insert	Delete	Update
:OLD			
:NEW			

23. Igaz vagy hamis az állítás? ( I, H )

	A <b>%rowtype</b> utasítással egy adattábla megadott oszlopának megfelelő típust hozhatunk létre.
	A <b>for</b> ciklusban a ciklusváltozót deklarálni kell.
	Egy tárolt eljárásnak lehet OUT típusú paramétere
	Egy insert parancs kiadásakor mindig keletkezik kurzor
	A pl/sql függvényeket mindig select paranccsal hívjuk meg.
	A bemenő paraméter nélküli <b>t1db</b> függvény meghívható a következő módon: <b>select t1db() from dual;</b>
	Ha egy trigger többféle műveletre is elindul, a műveleteket az <b>AND</b> operátorral kapcsoljuk össze.
	Egy update típusú triggerben a következő kódsor helyes: if old.nev != new.nev then ...
	Lehet két ugyanolyan nevű tárolt eljárást létrehozni.

Pontszámok hasábonként:

1.	3x1, 2, 1, 5, 2;	13p
2.	2, 3, 2, 1, 5, 2, 1;	16p
3.	2, 2, 5x1;	9p
4.	4, 9x1;	13p
		Σ 51p

**A beszámításra kerülő pontszámot az elért pontszám\*0.392 képlet adja.**

## Minta ZH Megoldás

### Kérdések:

1. Egy bemenő paraméter nélküli tárolt rutin visszaad egy értéket. Ez a rutin:

A: Tárolt eljárás

B: Tárolt függvény

C: Lehet tárolt eljárás és tárolt függvény is.

2. Tárolt rutinokban a változókat hol deklaráljuk?

A: A BEGIN előtt, a DECLARE után

B: A BEGIN előtt

C: A BEGIN után

D: Bárhol, a BEGIN előtt vagy után

3. Melyik kulcsszóval adjuk meg a tárolt eljárás visszatérő értékét?

A: retur

B: returs

C: return

D: returns

E: Tárolt eljárásnak nincs visszatérő értéke

4. Elvileg hogyan hívhatjuk meg a Maci() tárolt rutint? (Több helyes válasz van)

A: PLEASE Maci();

B: begin Maci(); end;

C: RUN Maci();

D: select Maci from dual;

E: SELECT Maci();

F: CALL Maci;

G: SELECT Maci;

H: begin Maci; end;

5. Szintaktikailag helyes-e az alábbi kód:

```
create function novel5 (x in number) return
number as
begin
    return x+5;
end;
```

A: IGEN

B: NEM

6. PLSQL-ben milyen szimbólumok jelzik az alábbiakat:

Utasítások vége	;
Karakterlánc összefűzés	
Blokk neve	<< >>
Hiba típusú változó	exception
Értékadás változónak	:=

7. Milyen paranccsal lehet egy tárolt eljárásban kiírni egy változó értékét a képernyőre:

Megoldás: **dbms\_output.put\_line**

Név: \_\_\_\_\_

Neptunkód: \_\_\_\_\_

0-31p-1, 42-46p-4	Pont	Jegy
32-36p-2, 47-51p-5		
37-41p-3,		

8. Milyen hibát dob a rendszer, ha egy select into utasítás egynél több adatot eredményez?

Megoldás: **too\_many\_rows**

9. Melyik művelet lehet egy trigger elindítója?

A: Create

D: Delete

G: Drop

B: Insert

E: Before

H: Select

C: Alter

F: Update

I: After

10. Mi a helyes műveleti sorrend egy kurzor használata során?

3	FETCH
2	OPEN
1	DECLARE
4	CLOSE

11. Mely kulcsszavak használhatók INSERT típusú triggernél?

A: NEW

B: OLD

C: Mindkettő

12. Írja be a hiányzó kifejezéseket!

```
CREATE TRIGGER mumus AFTER update
on Termék for each row
BEGIN
    Insert into Naplo values
        (:OLD.AR || :NEW.AR);
END;
```

13. Az alábbi kód futtatásakor milyen típusú kurzor használat történik?

```
declare
    l_date date;
begin
    select trunc(sysdate) into
        l_date from dual;
end;
```

Megoldás: **implicit**

14. Melyik változó típus nem használható pl/sqlben?

A: char

B: varchar1

C: varchar2

D: integer

E: number

15. Milyen kulcsszó hiányzik a kódból?

```
IF foo IS NULL then
  NULL;
END IF;
```

16. Milyen kulcsszó hiányzik a kódból?

```
IF foo = 1 THEN
  l_text := 'A';
elsif foo = 2 THEN
  l_text := 'B';
ELSE
  l_text := 'C';
END IF;
```

17. A következők közül melyik hibakód **nem** létezik pl/sql-ben?

- A: no\_data\_found                      **B: two\_many\_rows**  
 C: invalid\_number                      **D: others**  
 E: Mind létezik, mind használható

18. Hol kell deklarálni az explicit kurzort pl/sql-ben?

- A: A pl/sql munkaterületen  
**B: A pl/sql hibakezelő részben**  
 C: A pl/sql deklarációs részben  
 D: A pl/sql végrehajtási részben  
 E: Az explicit kurzort nem kell deklarálni

19. Melyik állítás helyes a **select into** kifejezésre?

- A: Használatakor nem keletkezhet hiba  
**B: Ha több adatsort ad vissza, több változót kell hozzá használni**  
 C: Csak egy visszaadott adatsor esetén használható  
 D: Egyik válasz sem helyes

20. A naplózó triggererek tipikusan milyen típusúak?

- A: After  
 B: Before  
**C: After, for each row**  
 D: Before, for each row  
 E: Instead of

21. Ha egy trigger feladata az, hogy adatmódosításkor csak bizonyos értékek kerülhessenek egy mezőbe, akkor milyen típusú triggeret kell készíteni?

- A: After  
 B: Before  
 C: After, for each row  
**D: Before, for each row**  
 E: Instead of

22. Töltse ki a triggerekkel kapcsolatos táblázatot! Jelölje be (x), ahol az adott kulcsszó használható!

	Insert	Delete	Update
:OLD		x	x
:NEW	x		x

23. Igaz vagy hamis az állítás? ( I, H )

H	A %rowtype utasítással egy adattábla megadott oszlopának megfelelő típust hozhatunk létre.
H	A for ciklusban a ciklusváltozót deklarálni kell.
I	Egy tárolt eljárásnak lehet OUT típusú paramétere
I	Egy insert parancs kiadásakor mindig keletkezik kurzor
H	A pl/sql függvényeket mindig select paranccsal hívjuk meg.
I	A bemenő paraméter nélküli t1db függvény meghívható a következő módon: <b>select t1db() from dual;</b>
H	Ha egy trigger többféle műveletre is elindul, a műveleteket az AND operátorral kapcsoljuk össze.
H	Egy update típusú triggerben a következő kódsor helyes: if old.nev != new.nev then ...
I	Lehet két ugyanolyan nevű tárolt eljárást létrehozni.

Pontszámok hasábonként:

1.	3x1, 2, 1, 5, 2;	13p
2.	2, 3, 2, 1, 5, 2, 1;	16p
3.	2, 2, 5x1;	9p
4.	4, 9x1;	13p
		Σ 51p

A beszámításra kerülő pontszámot az elért pontszám\*0.392 képlet adja.