

ALKALMAZOTTAK ÉLETPÁLYA MODELLJE – JAVA ÉS SQL ESETTANULMÁNY AZ ORACLE HR SÉMÁRA ÉPÍTVE

CAREER MODEL FOR EMPLOYEES – JAVA AND SQL CASE STUDY BASED ON ORACLE HR SCHEMA

Kaczur Sándor^{0009-0009-0118-2575 1*}, Kiss Balázs^{0009-0004-4422-0731 1}

¹ it-tanfolyam.hu, Magyarország

<https://doi.org/10.47833/2023.2.CSC.007>

Kulcsszavak:

oktatás
it-tanfolyam.hu
adatbázis-kezelés
Java
objektumorientált programozás
SQL
Oracle HR séma

Keywords:

education
it-tanfolyam.hu
database management
Java
object-oriented programming
SQL
Oracle HR schema

Cikk történet:

Beérkezett 2023. augusztus 27.
Átdolgozva 2023. október 25.
Elfogadva 2023. október 30.

Összefoglalás

Az objektumorientált programozás oktatásának része olyan kliensprogramok tervezése, kódolása és tesztelése, amelyek képesek adatbázishoz csatlakozni. A belépő szint csupán lekérdezés, adatok megjelenítése űrlapokon, táblázatos komponensekben, vagy grafikonokon. Haladó szinten már szükséges az adatok karbantartása is, illetve konzolos, asztali alkalmazásokról át lehet térni webes és mobil platformokra is. A szerzők ebből a tanulási útból emel ki egy esettanulmányt, amely Java és SQL nyelveken készült és MVC architektúris tervezési mintát használ. A felhasznált mintaadatok az Oracle HR sémából származnak. Az elemzés betekintést nyújt a modellalkotás lehetőségeibe – többféle megközelítést alkalmazva, elvi és konkrét szinten is.

Abstract

Part of teaching object-oriented programming is designing, coding and testing client programs that can connect to a database. The entry level is just querying, displaying data on forms, table display components, or graphs. At an advanced level, data maintenance is also required, and it is possible to change from console and desktop applications to web and mobile platforms. The authors' article highlights a case study from this learning path, written in Java and SQL and using the MVC architectural design pattern. The sample data are derived from the Oracle HR schema. The analysis provides insight into the possibilities of model creation – using several approaches, both on a principled and practical level.

1. Bevezetés

Az algoritmusok, programozási tételek és a Java programozás oktatása során többször áll rendelkezésre olyan kritikus tömegű ismeretanyag, amely lehetővé teszi nagyobb léptékű projektmunkák, komplex esettanulmányok megtervezését, elkészítését, tesztelését. Az Alkalmazottak

* Kapcsolattartó szerző.
E-mail cím: kaczursandor@gmail.com

A projektet három, egymást követő fázisra osztjuk az alábbiak szerint:

- Ellenőrzéseket végzünk arra vonatkozóan, hogy illeszkednek-e a meglévő adatok egy tipikus életpálya modellbe. Lekérdezzük a rendelkezésre álló adatokat.
- Megalkotunk egy fiktív életpálya modellt és „mi lenne, ha” típusú kérdéseket teszünk fel. Származtatott mezők segítségével előállítunk olyan adatokat, amelyek illeszkednek a fiktív életpálya modellbe.
- Két másféle életpálya modellt definiálva, nyújtunk – házi feladatként tanfolyamaink hallgatói számára – továbbfejlesztési, gyakorlási lehetőséget.

3. Megvalósítás

Ebben az alfejezetben a fokozatosság elve szerint haladunk, ahogyan érdemes kontakt foglalkozásokon és ahogyan tesszük azt blended learning tananyagaink felépítése során egyaránt. Négy részből áll egy-egy részfeladat az alábbiak szerint:

- Szövegesen specifikáljuk megoldandó részfeladatot.
- Ismertetjük a megoldáshoz szükséges SQL parancsot.
- Bemutatjuk a kapott eredménytáblát.
- Elemezzük az eredményt.

3.1. Lehetséges életpálya modell ellenőrzése

Lehetséges életpálya-modellnek tekinthető a munkakörhöz (JOB_ID és JOB_TITLE) tartozó, adható legkisebb és legnagyobb fizetés. Másképpen: fizetési modell. Például: pályakezdőt a legkisebb fizetéssel veszünk fel és a legnagyobb fizetéssel megy nyugdíjba. Vajon minden alkalmazottra teljesül, hogy a fizetése a megadott határok között található (zárt intervallumként kezelve)? Ennek ellenőrzésére használható az alábbi SQL parancs (2. ábra):

```

1 SELECT J.JOB_ID, JOB_TITLE, MIN_SALARY, MAX_SALARY,
2    MIN(SALARY), MAX(SALARY), AVG(SALARY), COUNT(EMPLOYEE_ID) AS COUNT_EMPLOYEE
3 FROM JOBS J JOIN EMPLOYEES E ON J.JOB_ID=E.JOB_ID
4 GROUP BY J.JOB_ID, JOB_TITLE, MIN_SALARY, MAX_SALARY
5 ORDER BY JOB_TITLE

```

2. ábra. Lehetséges életpálya modell ellenőrzése (SQL parancs)

Ez az egyszerű, két táblát összekapcsoló lekérdezés, mindössze néhány aggregáló függvényt, csoportosítást és sorrendiséget tartalmaz. Az alábbi eredménytáblát adja (3. ábra):

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY	MIN(SALARY)	MAX(SALARY)	AVG(SALARY)	COUNT_EMPLOYEE
FI_ACCOUNT	Accountant	4200	9000	6900	9000	7920	5
AC_MGR	Accounting Manager	8200	16000	12000	12000	12000	1
AD_ASST	Administration Assistant	3000	6000	4400	4400	4400	1
AD_VP	Administration Vice President	15000	30000	17000	17000	17000	2
FI_MGR	Finance Manager	8200	16000	12000	12000	12000	1
HR_REP	Human Resources Representative	4000	9000	6500	6500	6500	1
MK_MAN	Marketing Manager	9000	15000	13000	13000	13000	1
MK_REP	Marketing Representative	4000	9000	6000	6000	6000	1
AD_PRES	President	20000	40000	24000	24000	24000	1
IT_PROG	Programmer	4000	10000	4200	9000	5760	5
AC_ACCOUNT	Public Accountant	4200	9000	8300	8300	8300	1
PR_REP	Public Relations Representative	4500	10500	10000	10000	10000	1
PU_CLERK	Purchasing Clerk	2500	5500	2500	3100	2780	5
PU_MAN	Purchasing Manager	8000	15000	11000	11000	11000	1
SA_MAN	Sales Manager	10000	20000	10500	14000	12200	5
SA_REP	Sales Representative	6000	12000	6100	11500	8350	30
SH_CLERK	Shipping Clerk	2500	5500	2500	4200	3215	20
ST_CLERK	Stock Clerk	2000	5000	2100	3600	2785	20
ST_MAN	Stock Manager	5500	8500	5800	8200	7280	5

3. ábra. Lehetséges életpálya modell ellenőrzése (eredménytábla)

Az eredménytábla első négy oszlopában az adatok közvetlenül az adatbázisból származnak. Az utolsó négy oszlopban lévő adatok aggregáltak. Könnyen ellenőrizhető, hogy minden rekordra igaz, hogy a munkakörhöz tartozó lehetséges legkisebb fizetés kisebb vagy egyenlő az előforduló legkisebb fizetéshez viszonyítva, másképpen $\text{MIN_SALARY} \leq \text{MIN}(\text{SALARY})$. Hasonlóan így van a legnagyobb fizetés esetében is a munkakört betöltő összes alkalmazottra (COUNT_EMPLOYEE).

A válasz a kérdésre: a lehetséges életpálya modell bemeneti feltételei teljesülnek.

Másképpen is lehetne: SQL paranccsal számoljuk meg, hány olyan rekord van, amire teljesül a feltétel? Ha ez a munkakörök számát adja, akkor mindegyikre igaz, vagyis a fenti bemeneti feltételek teljesülnek.

3.2. Munkakörök esetén lehetséges jutalék

Adódik néhány tipikus kérdés. Milyen munkakörök esetén lehetséges jutalék? Ha van jutalék, akkor mekkora mértékű? Hány alkalmazottnak lehet/van jutaléka? Az alábbi SQL parancs (4. ábra) megadja a válaszokat:

```

1 SELECT J.JOB_ID, JOB_TITLE, COMMISSION_PCT, COUNT(EMPLOYEE_ID) AS COUNT_EMPLOYEE
2 FROM JOBS J JOIN EMPLOYEES E ON J.JOB_ID=E.JOB_ID
3 WHERE COMMISSION_PCT IS NOT NULL
4 GROUP BY J.JOB_ID, JOB_TITLE, COMMISSION_PCT
5 ORDER BY JOB_TITLE, COMMISSION_PCT

```

4. ábra. Munkakörök esetén lehetséges jutalék (SQL parancs)

A lekérdezés feltétele kihagyja azokat a munkaköröket, amikhez nem tartozik jutalék. Az alábbi eredménytáblát adja (5. ábra):

JOB_ID	JOB_TITLE	COMMISSION_PCT	COUNT_EMPLOYEE
SA_MAN	Sales Manager	.2	1
SA_MAN	Sales Manager	.3	3
SA_MAN	Sales Manager	.4	1
SA_REP	Sales Representative	.1	6
SA_REP	Sales Representative	.15	5
SA_REP	Sales Representative	.2	6
SA_REP	Sales Representative	.25	6
SA_REP	Sales Representative	.3	4
SA_REP	Sales Representative	.35	3

5. ábra. Munkakörök esetén lehetséges jutalék (eredménytábla)

35 fő dolgozik a Sales részlegben, kétféle munkakört betöltve. Az 5 fő Sales Manager (értékesítési vezető) jutaléka a fizetés 20%-ától 40%-áig terjedhet 10%-os lépésközzel (3-féle lehet). A 30 fő Sales Representative (üzletkötő) jutaléka a fizetés 10%-ától 35%-áig terjedhet 5%-os lépésközzel (6-féle lehet). A Sales részlegben mindenkinek van jutaléka: nem csak, hogy nem NOT NULL, hanem pozitív tört. A létszámok (35, 5, 30) külön-külön SQL parancsokkal is előállíthatók, mintegy ellenőrzésként.

3.3. Első életpálya modell, összesített

Az előző két pontban láttuk, hogy az értékesítési vezetők adható havi fizetése 10000 és 20000 között van, átlagfizetésük 12200. Az üzletkötők paraméterei hasonlóan: 6000, 12000, 8350.

Az első életpálya modellben tegyük fel, hogy a jutalék negyedévente kerül kifizetésre és a havi fizetés megadott százaléka. Például: Elizabeth Bates üzletkötő havi fizetése 7300, jutaléka 15%, azaz minden 3. hónapban a fizetése 8395 helyett 10585. Adjuk meg a cég bérköltségét a negyedévek első hónapjában és utolsó hónapjában! Ez összesített adat azon alkalmazottakra, akik részleghez tartoznak (106 fő a 107 főből). A 6. ábrán látható SQL parancs előállítja a szükséges adatokat.

Az utolsó két aggregálásnál beépített elágazás megoldja, hogy csak értelmes esetben érdemes figyelembe venni a jutalékot. NULL adatot nem helyettesítene be a kifejezésbe akkor sem, ha a 13. sorból kihagynánk a részleg nevére vonatkozó feltételt (ami helyett részlegazonosító is

használható). A 7. ábrán látható, hogy a negyedévek első két hónapjában a cég bérköltése 691400, az utolsó hónapjában pedig 765090.

```

1 SELECT DEPARTMENT_NAME AS "Részleg",
2        SUM(SALARY) AS "Fizetés",
3        (SELECT SUM(SALARY) FROM EMPLOYEES) AS "Összes fizetés",
4        SUM(SALARY*(1+CASE WHEN COMMISSION_PCT IS NULL
5             THEN 0
6             ELSE COMMISSION_PCT
7             END)) AS "Jutalékos fizetés",
8        (SELECT SUM(SALARY*(1+CASE WHEN COMMISSION_PCT IS NULL
9             THEN 0
10            ELSE COMMISSION_PCT
11            END)) FROM EMPLOYEES) AS "Összes jutalékos fizetés"
12 FROM EMPLOYEES E, DEPARTMENTS D
13 WHERE E.DEPARTMENT_ID=D.DEPARTMENT_ID AND DEPARTMENT_NAME='Sales'
14 GROUP BY DEPARTMENT_NAME
15 ORDER BY DEPARTMENT_NAME

```

6. ábra. Első életpálya modell, összesített (SQL parancs)

Részleg	Fizetés	Összes Fizetés	Jutalékos Fizetés	Összes Jutalékos Fizetés
Sales	304500	691400	377140	765090

7. ábra. Első életpálya modell, összesített (eredménytábla)

Mindez arra a 34 fő alkalmazottra vonatkozik, akik a részlegük alapján a Sales részleghez tartoznak. Ez az SQL parancs 13. sorából derül ki, ahol a két táblát összeköti a részleg azonosító, ami természetesen NOT NULL. 1 fő alkalmazott hiányzik, aki munkaköre alapján Sales-es.

3.4. Első életpálya modell, részlegenként, 1. megoldás

Finomítsuk az első életpálya modell összesített előző, cég szintjén összesített változatát! Bontsuk le a részlegekre összesítve a fizetéseket és a jutalékokat! Fejezzük ki százalékosan (egy tizedesjegyre kerekítve) a részlegekre jutó bérköltés arányát a tényleges fizetésre és a jutalékos fizetésre vonatkoztatva egyaránt! Vegyük figyelembe azt a 106 fő alkalmazottat, akik részleghez tartoznak (a 107 fő közül)! Az alábbi SQL parancs (8. ábra) megadja a szükséges adatokat:

```

1 SELECT DEPARTMENT_NAME AS "Részleg",
2        COUNT(EMPLOYEE_ID) AS "Létszám",
3        SUM(SALARY) AS "Fizetés",
4        ROUND(SUM(SALARY) / (SELECT SUM(SALARY) FROM EMPLOYEES) * 100, 1) AS "%",
5        SUM(SALARY*(1+CASE WHEN COMMISSION_PCT IS NULL
6             THEN 0
7             ELSE COMMISSION_PCT
8             END)) AS "Jutalékos fizetés",
9        ROUND(SUM(SALARY*(1+CASE WHEN COMMISSION_PCT IS NULL
10            THEN 0
11            ELSE COMMISSION_PCT
12            END)) / (SELECT SUM(SALARY*(1+CASE WHEN COMMISSION_PCT IS NULL
13            THEN 0
14            ELSE COMMISSION_PCT
15            END)) FROM EMPLOYEES) * 100, 1) AS "%"
16 FROM EMPLOYEES E, DEPARTMENTS D
17 WHERE E.DEPARTMENT_ID=D.DEPARTMENT_ID
18 GROUP BY DEPARTMENT_NAME
19 ORDER BY DEPARTMENT_NAME

```

8. ábra. Első életpálya modell, részlegenként, 1. megoldás (SQL parancs)

A megoldás egyszerűbb az előzőnél, két ok miatt. Egyrészt a 17. sorból csupán elhagyni kell a részleg nevére vonatkozó előző feltételt (6. ábra, 13. sor). Másrészt itt már mindenképp szükséges az utolsó két aggregáló kifejezésben az elágazás, hiszen NULL jutalékkal nem végezhető aritmetikai művelet (és ez az előző SQL parancsból áttemelhető). Az eredménytábla a 9. ábrán látható:

Részleg	Létszám	Fizetés	%	Jutalékos Fizetés	%
Accounting	2	20300	2,9	20300	2,7
Administration	1	4400	,6	4400	,6
Executive	3	58000	8,4	58000	7,6
Finance	6	51600	7,5	51600	6,7
Human Resources	1	6500	,9	6500	,8
IT	5	28800	4,2	28800	3,8
Marketing	2	19000	2,7	19000	2,5
Public Relations	1	10000	1,4	10000	1,3
Purchasing	6	24900	3,6	24900	3,3
Sales	34	304500	44	377140	49,3
Shipping	45	156400	22,6	156400	20,4

9. ábra. Első életpálya modell, részlegenként, 1. megoldás (eredménytábla)

Látható, hogy minden körbeér: a fizetés és a jutalékos fizetés a Sales részlegen kívül mindenhol azonos, azaz csak a Sales részleghez tartozó munkakörökhöz (és így az ottani alkalmazottakhoz) tartozik jutalék. Mindez már 106 fő alkalmazottra vonatkozik. Vegyük észre, hogy a Sales részleg létszámára 34 főt kapunk.

3.5. Első életpálya modell, részlegenként, 2. megoldás

Láttuk, hogy a Sales részlegben 35-en dolgoznak. Ez akkor helytálló, ha a munkakörök alapján kérdezzük le és láttuk, hogy a 35 főből értékesítési vezetőként 5 fő, üzletkötőként 30 fő dolgozik. Igen ám, de van egy olyan alkalmazott, aki nem tartozik egy részleghez sem (DEPARTMENT_ID IS NULL), ezért kapjuk 9. ábra szerint a Sales részlegben a 34 főt. Ugyanis az azt előállító lekérdező parancs a DEPARTMENT_ID részlegazonosító alapján kapcsolja össze a két táblát. Ha az ő fizetését is figyelembe kell venni, akkor ez lehetséges az alábbi SQL lekérdező paranccsal (10. ábra):

```

1  SELECT DEPARTMENT_NAME AS "Részleg",
2  COUNT(EMPLOYEE_ID) AS "Létszám",
3  SUM(SALARY) AS "Fizetés",
4  ROUND(SUM(SALARY) /
5  (SELECT SUM(SALARY) FROM EMPLOYEES)*100, 1) AS "%",
6  SUM(SALARY*(1+CASE WHEN COMMISSION_PCT IS NOT NULL
7  THEN COMMISSION_PCT ELSE 0 END)) AS "Jutalékos fizetés",
8  ROUND(SUM(SALARY*(1+CASE WHEN COMMISSION_PCT IS NOT NULL
9  THEN COMMISSION_PCT ELSE 0 END)) /
10 (SELECT SUM(SALARY*(1+CASE WHEN COMMISSION_PCT IS NOT NULL
11 THEN COMMISSION_PCT ELSE 0 END)) FROM EMPLOYEES)*100, 1) AS "%"
12 FROM EMPLOYEES E, DEPARTMENTS D
13 WHERE E.DEPARTMENT_ID=D.DEPARTMENT_ID
14 GROUP BY DEPARTMENT_NAME
15
16 UNION ALL
17
18 SELECT '-', COUNT(EMPLOYEE_ID), SUM(SALARY),
19 ROUND(SUM(SALARY) /
20 (SELECT SUM(SALARY) FROM EMPLOYEES)*100, 1),
21 SUM(SALARY*(1+CASE WHEN COMMISSION_PCT IS NOT NULL
22 THEN COMMISSION_PCT ELSE 0 END)),
23 ROUND(SUM(SALARY*(1+CASE WHEN COMMISSION_PCT IS NOT NULL
24 THEN COMMISSION_PCT ELSE 0 END)) /
25 (SELECT SUM(SALARY*(1+CASE WHEN COMMISSION_PCT IS NOT NULL
26 THEN COMMISSION_PCT ELSE 0 END)) FROM EMPLOYEES)*100, 1)
27 FROM EMPLOYEES
28 WHERE DEPARTMENT_ID IS NULL
29 GROUP BY '-'

```

10. ábra. Első életpálya modell, részlegenként, 2. megoldás (SQL parancs)

A UNION alparancs egyesíti a két azonos szerkezetű (rész)eredménytáblát. Az alkalmazottak két diszjunkt részhalmaza (van részlege, nincs részlege) az SQL parancs 13. és 28. sora alapján válik szét. Az eredménytáblában az utolsó, 12. rekord tartalmazza az eddig hiányzó 1 fő alkalmazott adatait (11. ábra):

Részleg	Létszám	Fizetés	%	Jutalékos Fizetés	%
Administration	1	4400	,6	4400	,6
Accounting	2	20300	2,9	20300	2,7
Executive	3	58000	8,4	58000	7,6
IT	5	28800	4,2	28800	3,8
Purchasing	6	24900	3,6	24900	3,3
Human Resources	1	6500	,9	6500	,8
Public Relations	1	10000	1,4	10000	1,3
Shipping	45	156400	22,6	156400	20,4
Finance	6	51600	7,5	51600	6,7
Sales	34	304500	44	377140	49,3
Marketing	2	19000	2,7	19000	2,5
-	1	7000	1	8050	1,1

11. ábra. Első életpálya modell, részlegenként, 2. megoldás (eredménytábla)

Összesítve a 107. alkalmazott, másképpen a Sales részleg 35. – eddig kimaradt – alkalmazottjának a fizetése 7000 és jutalékos fizetése 8050. Az ő fizetése az első életpálya modell alapján (kerekítve) egyszázad arányban (az egytizedes különbség miatt: 1 vs. 1,1) „árnyalja” cég szinten részlegek között szétsztható fizetések arányát. A különbség a negyedévek első két hónapja és harmadik hónapja között adódik a modell szerint.

Az eredménytábla – az utolsó rekord kivételével – majdnem megegyezik az előzővel. A fizetési modell szerint a negyedévek első két hónapjában a cégre vonatkozó bérköltség 7000-rel növekszik és a negyedévek harmadik hónapjában pedig 8050-nel. A fizetések arányát százalékban egy tizedesjeggyel ábrázolva szinte nem vehető észre a különbség. A rekordok különböző sorrendjét a UNION és az ORDER BY alparancsok együttes alkalmazása okozza.

3.6. Második életpálya modell

Módosítsuk az előző életpálya modellt! Vajon hogyan változna az előző életpálya modell, ha a negyedévente kifizetendő jutalék számítási alapja a havi fizetés helyett a háromhavi – időszakra vonatkozó – fizetés megadott százaléka lenne? Hogyan alakulna ez alapján a cég bérköltsége havi és negyedéves szinten?

3.7. Harmadik életpálya modell

Módosítsuk valamelyik korábbi – akár mindkét – életpálya modellt! Hogyan változna a részlegek alkalmazottainak fizetése, ha valamelyik korábbi jutalék által meghatározott bértömeg cég szintjén kerülne szétszthatásra az alkalmazottak fizetésének arányában? Másképpen: azok is kapnának jutalékot, akik eddig nem kaptak, és akik eddig kaptak, kevesebbet fognak kapni („arányosan”).

4. Összegzés

Tanfolyamaink hallgatói az első életpálya modell gondolatmenete alapján, a példák feldolgozását követően tudnak becslést és konkrét válasz(oka)t is adni a második és harmadik életpálya modell felvetett módosításai alapján a problémákra. Mindezek kellő kihívást jelentő gyakorlati, egyben továbbfejlesztési lehetőségek. Gyakorlatorientált problémák, kicsiben átgondolt döntéstámogató modellek („mi lenne, ha”) is egyben. A megfogalmazott tapasztalatok öt korábbi tanfolyami csoportból (2018-2022), valamint két Erasmus program keretében megvalósult két kurzus [6, 7] visszajelzéseiből származnak (2016 és 2017). A programozás és az adatbázis-kezelés témakörök egyesítése, összekapcsolása fontos mérföldkő a szakmajegyzékes szoftverfejlesztő és –tesztelő technikus képzésen, valamint a felsőoktatás mérnökinformatikus szakján egyaránt.

A projekt tervezése során törekedni kell a következetes objektumorientált tervezésre, az MVC architekturális tervezési minta egyes rétegeinek megfelelő, egészséges arányaira. Nem szabad túlburjánzó, minden (is) megoldani képes modell réteget létrehozni. A rétegek közötti interfészek is kiemelt figyelmet igényelnek [4, 6, 11]. Az esetlegesen előforduló kivételobjektumot következetes végiggörgetését a modell rétegen át a vezérlőn át a nézetig (hogy meg tudjon jelenni a GUI-n egy testre szabott hibaüzenet) is körültekintően meg kell oldani [5, 8, 12]. Ez a rétegek közötti adatátvitelt, metódushívást, kommunikációt jelenti. Érdemes még kiemelni az SQL parancsok futtatását lehetővé tevő JDBC réteg verziói közötti különbségeket, valamint a szintén verzióspecifikus eltéréseket az ORACLE SQL-ben [10].

Köszönetnyilvánítás

Az it-tanfolyam.hu oktatói csapata köszönetet mond a Java adatbázis-kezelő tanfolyam [2] aktív és alumni hallgatóinak, akik hosszú évek óta folyamatosan hasznos észrevételeket tesznek a felhasznált projektjeink, esettanulmányaink, oktatóprogramjaink kapcsán és követik szakmai blogunkat a tanfolyam tematikájához kötődő kategóriában [3] is.

Irodalomjegyzék

- [1] Iványi, A.: Informatikai algoritmusok I., II. ELTE, 2004., 2005., ISBN 9634637752
- [2] Java adatbázis-kezelő tanfolyam, it-tanfolyam.hu landing page, <https://it-tanfolyam.hu/java-adatbazis-kezelelo-tanfolyam/>, 2023.08.21.
- [3] Java adatbázis-kezelő tanfolyam, it-tanfolyam.hu szakmai blog bejegyzések kategóriája, <https://it-tanfolyam.hu/kategoria/javaab/>, 2023.08.21.
- [4] Kaczur, S.: Haladó lekérdezések az Oracle HR sémában, Digitális Témahét 2020, Budapest, it-tanfolyam.hu, 2020.03.27., előadás műhely napon
- [5] Kaczur, S.: Kik vettek részt projektmunkában? - Java esettanulmány az Oracle HR sémára építve, XXIX. Nemzetközi Számítástechnika és Oktatás Konferencia – SzámOkt 2019 kiadvány, Erdélyi Magyar Műszaki Tudományos Társaság, Temesvár, Románia, 2019, ISSN 1842-4546, p. 179-184
- [6] S. Kaczur: Java and programming; Distributed web applications in Java Server Pages platform; Database handling with Oracle, Erasmus+ oktatói mobilitás, Panevėžys, Litvánia, Panevėžys University of Applied Sciences, Faculty of Technological Sciences, 2017.09.19-24., kurzus Erasmus oktatói mobilitás keretében
- [7] S. Kaczur: Creating of the distributed web application OracleHR, Erasmus+ oktatói mobilitás, Santarém, Portugália, Santarém Polytechnic Institute, Management and Technology School, Department of Computer Science Quantitative Methods, 2016.11.4-12., kurzus Erasmus oktatói mobilitás keretében
- [8] Kaczur, S.: Lekérdezések és adatfeldolgozási algoritmusok optimalizálási lehetőségei Java SE és Oracle HR környezetben, 6. Báthory-Brassai Nemzetközi Konferencia Tanulmánykötet 1. kötet, Budapest, 2015, ISBN 978-615-5460-38-5, p. 753-765
- [9] Kaczur, S.: Az OracleHRJSP webalkalmazás működése, Matematikát, fizikát és informatikát oktatók (MAFIOK) XXXVIII. országos konferenciája, Pécs, Pécsi Tudományegyetem Pollack Mihály Műszaki és Informatikai Kar, 2014, ISBN 978-963-7298-55-4, p. 121-126
- [10] Oracle Help Center: 4.3 HR Sample Schema Table Descriptions, <https://docs.oracle.com/en/database/oracle/oracle-database/19/comsc/HR-sample-schema-table-descriptions.html#GUID-506C25CE-FA5D-472A-9C4C-F9EF200823EE>, 2023.05.28.
- [11] M. Parsian: JDBC Metadata, MySQL, and Oracle Recipes - A Problem-Solution Approach, Apress Berkeley, CA, eBook ISBN 978-1-4302-0134-2, 2006, <https://doi.org/10.1007/978-1-4302-0134-2>, Softcover ISBN 978-1-4842-2095-5, 2016
- [12] E. Sciore: Database Design and Implementation, Springer Nature Switzerland AG 2020, eBook ISBN 978-3-030-33836-7, <https://doi.org/10.1007/978-3-030-33836-7>
- [13] X. Qu: Application of Java Technology in Dynamic Web Database Technology, Journal of Physics: Conference Series, Volume 1744, 2020 International Conference on Mechanical Automation and Computer Engineering (MACE 2020) 28-30 October 2020, Xi'an, ShaanXi, China, <https://doi.org/10.1088/1742-6596/1744/4/042029>
- [14] C. Zhou, P. Frankl: Empirical Studies on Test Effectiveness for Database Applications, Empirical Studies on Test Effectiveness for Database Applications, Print ISSN: 2159-4848, <https://doi.org/10.1109/ICST.2012.86>