

## 2<sup>ο</sup> Σύνολο Ασκήσεων

Μέλη ομάδας:

- |                          |      |
|--------------------------|------|
| • Βλάρα Δήμητρα          | 3190 |
| • Καρακίτσος Βασίλης     | 3241 |
| • Ράπτης Λεωνίδα         | 3322 |
| • Τομπουλίδης Καλλίνικος | 3344 |

Τα στοιχεία υποβολής στο ecouse:

- kaltompoulides@gmail.com

Άσκηση 1 (σχεσιακή άλγεβρα)

α)  $R_1 = \text{professor} \bowtie_{p\_id=u\_id} \text{taughtby}$   
 $R_2 = \text{student} \bowtie_{s\_id=u\_id} \text{taughtby}$

$$\pi_{course\_id}(R_1 \bowtie_{R_1.course\_id \neq R_2.course\_id} R_2)$$

β)  $\pi_{t_1, t_2}(\rho_{s_1(id_1, ip_1, t_1)}(\text{student}) \bowtie_{t_1 < t_2} \rho_{s_2(id_2, ip_2, t_2)}(\text{student}))$

γ)  $\pi_{course\_id}(\text{taughtby} \bowtie_{u\_id=p\_id \text{ AND } u\_id=s\_id} \text{advisedby})$

δ)  $\pi_{course\_id}(\sigma_{u\_id=NULL}(\text{taughtby}))$

ε)  $Q = \pi_{p\_id}(\sigma_{s\_id=81}(\text{advisedby}))$

$$\pi_{s\_id}(\sigma_{p\_id=Q}(\text{advisedby}))$$

## Άσκηση 2 (σχεσιακός λογισμός)

α)  $\{t1.yearsinProgram, t2.yearsinProgram \mid student(t1,t2) \text{ and } ((\forall t3)(student(t3) \text{ and } t1.yearsinProgram < t3.yearsinProgram) \text{ and } ((\forall t4)(student(t4) \text{ and } t2.yearsinProgram \geq t4.yearsinProgram)))\}$

β) Με τον ποσοδείκτη «υπάρχει»:

$\{t.s\_id \mid student(t) \text{ and } ((\exists d) (student(d)) \text{ and } d.s\_id = 81 \text{ and } d.s\_id = t.s\_id))\}$

Με τον ποσοδείκτη «για κάθε»:

$\{t.s\_id \mid student(t) \text{ and not}((\forall d) (not student(d)) \text{ or } d.s\_id \neq 81 \text{ or } d.s\_id \neq t.s\_id))\}$

## Άσκηση 3 (SQL)

Για την υλοποίηση της τρίτης άσκησης χρησιμοποιήσαμε το Workbench σε λειτουργικό σύστημα Ubuntu 18.04. Έχει δημιουργήσαμε μία βάση school και φορτώσαμε τους πίνακες με την επιλογή import data. Παρακάτω παραθέτουμε και την φόρτωση των πινάκων με την εντολή load όπως ζητείται στην εκφώνηση (το αρχείο το προσθέσαμε και στο ecourse ως initialization.sql).

```
-- MySQL dump 10.13 Distrib 5.7.28, for Linux (x86_64)
--
-- Host: 127.0.0.1 Database: school

DROP TABLE IF EXISTS `advisedby`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `advisedby` (
  `s_id` int(11) NOT NULL,
  `p_id` int(11) NOT NULL,
  PRIMARY KEY (`p_id`,`s_id`),
  KEY `fk_advisedby_2_idx` (`p_id`)
) ENGINE=InnoDB DEFAULT CHARSET=big5;
/*!40101 SET character_set_client = @saved_cs_client */;

-- Dumping data for table `advisedby`

LOCK TABLES `advisedby` WRITE;
LOAD DATA INFILE 'advisedby.csv'
INTO TABLE discounts
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
UNLOCK TABLES;

-- Table structure for table `course`

DROP TABLE IF EXISTS `course`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `course` (
  `course_id` int(11) NOT NULL,
  `courseLevel` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`course_id`),
  UNIQUE KEY `course_id_UNIQUE` (`course_id`)
```

```

) ENGINE=InnoDB DEFAULT CHARSET=big5;
/*!40101 SET character_set_client = @saved_cs_client */;

-- Dumping data for table `course`

LOCK TABLES `course` WRITE;
LOAD DATA INFILE 'course.csv'
INTO TABLE discounts
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
UNLOCK TABLES;

-- Table structure for table `professor`

DROP TABLE IF EXISTS `professor`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `professor` (
  `p_id` int(11) NOT NULL,
  `hasPosition` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`p_id`),
  UNIQUE KEY `p_id_UNIQUE` (`p_id`)
) ENGINE=InnoDB DEFAULT CHARSET=big5;
/*!40101 SET character_set_client = @saved_cs_client */;

-- Dumping data for table `professor`

LOCK TABLES `professor` WRITE;
LOAD DATA INFILE 'professor.csv'
INTO TABLE discounts
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
UNLOCK TABLES;

-- Temporary table structure for view `professorCount`

DROP TABLE IF EXISTS `professorCount`;
/*!50001 DROP VIEW IF EXISTS `professorCount` */;
SET @saved_cs_client = @@character_set_client;
SET character_set_client = utf8;
/*!50001 CREATE VIEW `professorCount` AS SELECT
1 AS `p_id`,
1 AS `students` */;
SET character_set_client = @saved_cs_client;

-- Table structure for table `student`

DROP TABLE IF EXISTS `student`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `student` (
  `s_id` int(11) NOT NULL,
  `inPhase` varchar(45) DEFAULT NULL,
  `yearsInProgram` int(11) DEFAULT NULL,
  PRIMARY KEY (`s_id`),
  UNIQUE KEY `s_id_UNIQUE` (`s_id`)
) ENGINE=InnoDB DEFAULT CHARSET=big5;
/*!40101 SET character_set_client = @saved_cs_client */;

-- Dumping data for table `student`

LOCK TABLES `student` WRITE;

```

```

LOAD DATA INFILE 'student.csv'
INTO TABLE discounts
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
UNLOCK TABLES;

-- Table structure for table `taughtby`

DROP TABLE IF EXISTS `taughtby`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `taughtby` (
  `course_id` int(11) NOT NULL,
  `u_id` int(11) DEFAULT NULL,
  PRIMARY KEY (`course_id`)
) ENGINE=InnoDB DEFAULT CHARSET=big5;
/*!40101 SET character_set_client = @saved_cs_client */;

-- Dumping data for table `taughtby`

LOCK TABLES `taughtby` WRITE;
LOAD DATA INFILE 'taughtby.csv'
INTO TABLE discounts
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
UNLOCK TABLES;

```

- α)i)     select s1.yearsinProgram as t1, s2.yearsinProgram as t2  
           from student s1, student s2  
           where s1.yearsinProgram < ALL (select max(s2.yearsinProgram) from student  
           s2 where s1.yearsinProgram < s2.yearsinProgram)  
           order by s1.yearsinProgram asc, s2.yearsinProgram desc  
           limit 1
- ii)     select distinct course\_id  
           from taughtby  
           where u\_id in(select s\_id from advisedby where u\_id = p\_id)
- iii)    select distinct s\_id  
           from advisedby  
           where p\_id in  
           (select p\_id FROM advisedby where s\_id = 81);
- iv)     select hasPosition, count(s\_id) as students  
           from professor inner join advisedby on professor.p\_id = advisedby.p\_id  
           group by professor.p\_id  
           order by students desc  
           limit 1;

- v) `(select taughtby.course_id as courses, count(professor.p_id) as professors, 0  
as students  
from professor  
inner join taughtby on professor.p_id = taughtby.u_id  
group by taughtby.course_id)  
UNION  
(select taughtby.course_id as courses, 0 as professors, count(student.s_id) as  
students  
from student  
inner join taughtby on student.s_id = taughtby.u_id  
group by taughtby.course_id)  
order by courses asc`
- vi) `select s.yearsinProgram, count(s.s_id)/tmp.students as percentage  
from taughtby tb, student s, (select count(stu.s_id) as students,  
stu.yearsinProgram as years  
from student stu  
group by stu.yearsinProgram) tmp  
where tb.u_id = s.s_id and s.yearsinProgram = tmp.years  
group by s.yearsinProgram`
- γ) i) `create or replace VIEW professorCount  
AS  
select p_id , count(*) as students  
from advisedby  
group by p_id;  
select * from professorCount`
- ii) `DELETE FROM advisedby WHERE s_id = 18;  
DELETE FROM advisedby WHERE s_id = 37;  
select * from professorCount`

Είναι updatable διότι διαγράφοντας τους φοιτητές με s\_id 18 & s\_id 37 μειώνονται οι μετρητές των αντίστοιχων καθηγητών συμβούλων p\_id 335 & 79 αντίστοιχα.

δ) i) Για την εκτέλεση του προγράμματος απαιτούνται οι εξής εντολές στο τερματικό:

```
javac di.java
java -cp ./mysql-connector-java-8.0.18.jar di
```

Ο κώδικας του αρχείου java:

```
import java.sql.*;
import java.util.Scanner;

class di
{
    public static void main(String ... args)
    {
        try
        {
            Scanner input = new Scanner(System.in);
            System.out.println("Enter the path to the database (ours was
            \"jdbc:mysql://localhost:3306/school\"): ");
            String dbUrl = input.nextLine();
            System.out.println("Enter username (i.e. \"root\"): ");
            String username = input.nextLine();
            System.out.println("Enter password: ");
            String password = input.nextLine();
            Connection myConnection = DriverManager.getConnection(dbUrl, username,
            password);

            Statement myStatement = myConnection.createStatement();
            System.out.println("Enter table name: ");
            String tableName = input.nextLine();
            System.out.println("Enter column name: ");
            String columnName = input.nextLine();
            System.out.println("Enter value: ");
            String value = input.nextLine();
            System.out.println("SELECT * FROM " + tableName + " WHERE " +
            columnName + " = " + value + ";");
            ResultSet myResultSet = myStatement.executeQuery("SELECT * FROM " +
            tableName + " WHERE " + columnName + " = " + "\"" + value + "\"" + ";");
            if(tableName.equals("advisedby"))
            {
                System.out.println("s_id    p_id");
            }else if(tableName.equals("course"))
            {
                System.out.println("course_id    courseLevel");
            }else if(tableName.equals("professor"))
            {
                System.out.println("p_id    hasPosition");
            }else if(tableName.equals("student"))
            {
                System.out.println("s_id    inPhase    yearsinProgram");
            }else if(tableName.equals("taughtby"))
            {
                System.out.println("course_id    u_id");
            }while(myResultSet.next())
            {
                if(tableName.equals("advisedby"))
                {
                    System.out.println(myResultSet.getString("s_id") + "    " +
                    myResultSet.getString("p_id"));
                }else if(tableName.equals("course"))
                {
                    System.out.println(myResultSet.getString("course_id") + "
                    " + myResultSet.getString("courseLevel"));
                }else if(tableName.equals("professor"))
```

```

        {
            System.out.println(myResultSet.getString("p_id") + " " +
myResultSet.getString("hasPosition"));
        }else if(tableName.equals("student"))
        {
            System.out.println(myResultSet.getString("s_id") + " " +
myResultSet.getString("inPhase") + " " + myResultSet.getString("yearsInProgram"));
        }else if(tableName.equals("taughtby"))
        {
            System.out.println(myResultSet.getString("course_id") + "
" + myResultSet.getString("u_id"));
        }
    }
}catch(Exception e)
{
    System.out.println(e.getMessage());
}
}
}

```

ii) Για την εκτέλεση του προγράμματος απαιτούνται οι εξής εντολές στο τερματικό:

```
javac dii.java
java -cp ./mysql-connector-java-8.0.18.jar dii
```

Ο κώδικας του αρχείου java:

```
import java.sql.*;
import java.util.Scanner;

class dii
{
    public static void main(String ... args)
    {
        try
        {
            Scanner input = new Scanner(System.in);
            System.out.println("Enter the path to the database (ours was
            \"jdbc:mysql://localhost:3306/school\"): ");
            String dbUrl = input.nextLine();
            System.out.println("Enter username (i.e. \"root\"): ");
            String username = input.nextLine();
            System.out.println("Enter password: ");
            String password = input.nextLine();
            Connection myConnection = DriverManager.getConnection(dbUrl, username,
            password);

            Statement myStatement = myConnection.createStatement();
            ResultSet myResultSet = myStatement.executeQuery("(select
            taughtby.course_id as courses, professor.p_id as u_id, professor.hasPosition as pos from professor inner
            join taughtby on professor.p_id = taughtby.u_id) UNION (select taughtby.course_id as courses, student.s_id
            as u_id, student.inPhase as pos from student inner join taughtby on student.s_id = taughtby.u_id) order by
            courses asc");

            System.out.println("course_id      id      pos\n");
            while(myResultSet.next())
            {
                String pos = myResultSet.getString("pos");
                if(pos.equals("Faculty") || pos.equals("Faculty_eme") ||
                pos.equals("Faculty_aff") || pos.equals("Faculty_adj"))
                {
                    pos = pos.replace(pos, "prof");
                }else if(pos.equals("Post_Generals") || pos.equals("Pre_Quals") ||
                pos.equals("Post_Quals"))
                {
                    pos = pos.replace(pos, "stud");
                }
                System.out.println(myResultSet.getString("courses") + "
                " + myResultSet.getString("u_id") + "      " + pos);
            }
        }catch(Exception e)
        {
            System.out.println(e.getMessage());
        }
    }
}
```