

Modern Data Management & Business Intelligence
Assignment #1 – Due Date: November 20th, 2019 – Groups of two students

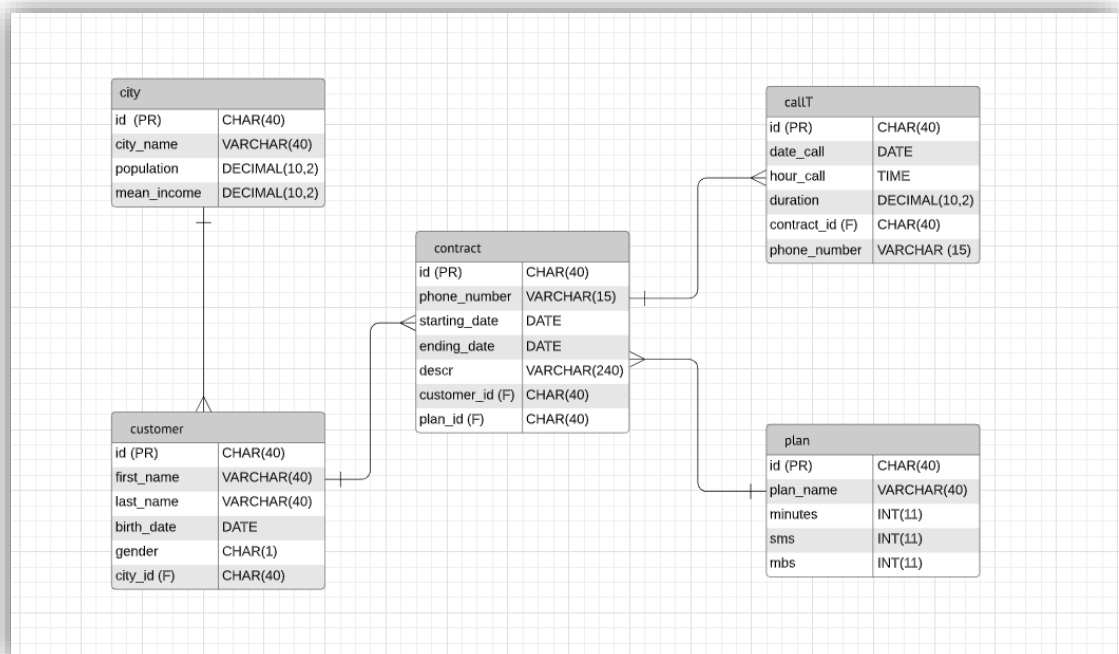
Ονοματεπώνυμο: Ερασμία Κορνελάτου
AM: f2821907

Description of the Case:

A telecom provider (TP) wants to develop a relational database to monitor customers, calls and plans. Customers of TP are described through a unique identifier, first and last name, date of birth, gender ('male' or 'female') and live in a city. Cities are described by a unique identifier, name, population and mean income. A customer has one or more contracts with TP. A contract has a unique identifier, phone number, starting date, ending date and a description. A contract is also associated to a plan offered by TP (e.g. Red1 of Vodafone). A plan is described by a unique identifier, name, free-minutes, free-sms and free-MB attributes. Finally, calls made by a phone number have to be stored, along with a unique identifier, the date/time of the call (hour, minute, day, month, year), the called phone number and the duration of the call (in seconds).

Deliverables (in one word document):

1. (10%) Use the Entity-Relationship Diagram (ERD) to model entities, relationships, attributes, cardinalities, and all necessary constraints. Use any tool you like to draw the ERD.



—+— πολλά προς ένα, (PR): Primary Key, (F): Foreign Key

2. (10%) Create the relational schema in MySQL/SQLServer and insert a few records into the tables to test your queries below. You will have to hand in the CREATE TABLE statements.

```

1  DROP DATABASE IF EXISTS telecom;
2
3  CREATE DATABASE telecom;
4
5  USE telecom;
6
7  CREATE TABLE city
8  (
9      id          CHAR(40) NOT NULL,
10     city_name   VARCHAR(40) NOT NULL ,
11     population  DECIMAL(10, 2) NOT NULL,
12     mean_income DECIMAL(10, 2) NOT NULL,
13     PRIMARY KEY (id)
14 );
15 INSERT INTO city (id,city_name,population,mean_income) VALUES
16 ('1','Athens','27000','9010'),
17 ('2','Thessaloniki','22000','8500'),
18 ('3','Patra','19000','8300');
19 CREATE TABLE customer
20 (
21     id          CHAR(40) NOT NULL,
22     first_name   VARCHAR(40) NOT NULL ,
23     last_name   VARCHAR(40) NOT NULL,
24     birth_date  DATE NOT NULL,
25     gender      CHAR(1) NOT NULL,
26     city_id     CHAR(40) NOT NULL,
27     PRIMARY KEY (id),
28     FOREIGN KEY (city_id) REFERENCES city(id)
29 );
30 INSERT INTO customer (id,first_name,last_name,birth_date,gender,city_id) VALUES
31 ('1','Rita','Sakellariou','1964-03-07','f','1'),
32 ('2','Manthos','Foustanos','1992-01-01','m','2'),
33 ('3','Maria','Antouaneta','1932-03-02','f','3'),
34 ('4','Sakis','Rouvas','1975-01-07','m','1'),
35 ('5','Freddy','Mercury','1958-09-06','m','1'),
36 ('6','Pamela','Anderson','1968-02-01','f','2'),
37 ('7','Anna','Frank','1932-02-09','f','2'),
38 ('8','Zoi','Konstantopoulou','1970-08-06','f','3'),
39 ('9','Miriam','Andrikopoulou','1965-05-05','f','3'),
40 ('10','Josoua','Andrikopoulos','1963-05-04','m','3'),
41 ('11','Oliver','Twist','1935-09-01','m','3');
42
43 CREATE TABLE plan
44 (
45     id          CHAR(40) NOT NULL,
46     plan_name   VARCHAR(40) NOT NULL ,
47     minutes     INT,
48     sms         INT,
49     mbs         INT,
50     PRIMARY KEY (id)
51 );
52 INSERT INTO plan (id,plan_name,minutes,sms,mbs) VALUES
53 ('1','student_mobile','1200','1200','1200'),
54 ('2','combo_mobile','600','600','600'),
55 ('3','Freedom','600','0','0');
56
57 CREATE TABLE contract
58 (
59     id          CHAR(40) NOT NULL,
60     phone_number VARCHAR(15) NOT NULL ,
61     starting_date DATE NOT NULL,
62     ending_date  DATE NOT NULL,
63     descr        VARCHAR(240) NOT NULL,
64     customer_id  CHAR(40) NOT NULL,
65     plan_id      CHAR(40) NOT NULL,
66     PRIMARY KEY (id),
67     FOREIGN KEY (customer_id) REFERENCES customer(id),
68     FOREIGN KEY (plan_id) REFERENCES plan(id)
69 );

```

```

70
71 INSERT INTO contract (id,phone_number,starting_date,ending_date,descr,customer_id,plan_id) VALUES
72 ('1','6912345187','2016-01-01','2020-01-01','4 years','1','1'),
73 ('2','6956192310','2016-01-09','2019-01-09','3 year','1','3'),
74 ('3','2104449104','2016-01-10','2019-01-10','1 year','3','3'),
75 ('4','2151237890','2016-01-10','2020-01-10','4 year','3','2'),
76 ('5','6912039182','2016-02-13','2020-02-13','4 year','1','3'),
77 ('6','6933501958','2016-02-13','2020-02-13','4 year','4','3'),
78 ('7','6933991958','2016-02-13','2020-02-13','4 year','2','3'),
79 ('8','6933501954','2016-02-14','2020-02-14','4 year','5','1'),
80 ('9','6933881954','2016-02-14','2020-02-14','4 year','6','1'),
81 ('10','6933881780','2016-03-14','2020-03-14','4 year','7','2'),
82 ('11','6933881430','2016-03-14','2020-03-14','4 year','8','2'),
83 ('12','6932681430','2016-03-14','2020-03-14','4 year','9','1'),
84 ('13','6932680980','2016-03-15','2020-03-15','4 year','10','1'),
85 ('14','6932681180','2016-03-15','2020-03-15','4 year','11','2');
86
87
88 CREATE TABLE callT
89 (
90     id          CHAR(40) NOT NULL,
91     date_call   DATE NOT NULL,
92     hour_call   TIME(0) NOT NULL,
93     duration    DECIMAL(10, 2) NOT NULL,
94     contract_id CHAR(40) NOT NULL,
95     phone_number VARCHAR(15) NOT NULL,
96     PRIMARY KEY (id),
97     FOREIGN KEY (contract_id) REFERENCES contract(id)
98 );
99

```

```

100 INSERT INTO callT (id,date_call,hour_call,duration,contract_id,phone_number) VALUES
101 ('1','2017-06-30','09:00:00','20','1','6956192310'),
102 ('2','2017-07-23','10:00:00','29','1','6912959310'),
103 ('3','2018-06-25','09:30:00','31','1','6956192310'),
104 ('4','2018-06-27','09:35:00','22','1','6976123465'),
105 ('5','2018-06-30','09:40:00','25','2','6976123465'),
106 ('6','2018-07-28','09:34:00','28','3','6976123465'),
107 ('7','2018-07-28','22:00:00','35','1','6912530129'),
108 ('8','2018-07-28','22:01:00','150','8','6912530129'),
109 ('9','2018-07-28','22:02:00','33','9','6912530779'),
110 ('10','2018-07-28','22:03:00','100','7','6912530779'),
111 ('11','2018-07-28','22:04:00','110','12','6912530779'),
112 ('12','2018-07-28','22:04:00','10','13','6912530779'),
113 ('13','2018-07-28','22:03:00','120','10','6920530779'),
114 ('14','2018-07-29','22:01:00','121','11','6920530779'),
115 ('15','2018-07-28','23:00:00','120','1','6912530129'),
116 ('16','2018-07-29','23:00:00','23','2','6976123465'),
117 ('17','2018-07-29','23:25:00','29','1','6956192310'),
118 ('18','2018-07-30','23:29:00','99','8','6957869869'),
119 ('19','2018-07-30','23:27:00','23','6','6912345187'),
120 ('20','2018-07-30','23:27:00','200','7','6912345187'),
121 ('21','2018-07-30','23:25:00','23','2','6912345187'),
122 ('22','2019-10-25','12:00:00','22','7','6976123465'),
123 ('23','2019-10-25','13:00:00','300','7','6976014594'),
124 ('24','2019-10-26','13:00:00','300','7','6976014594'),
125 ('25','2019-10-27','13:25:00','350','8','6976395106'),
126 ('26','2019-10-27','13:35:00','340','8','6976395106');

```

3. (60%) Write SQL code and test it to your data for the following queries

a. Show the call id of all calls that were made between 8am and 10am on June 2018 having duration < 30

```

1  -- a) Show the call id of all calls that were made between 8am and 10am on June 2018 having duration < 30
2  • SELECT id AS call_id
3      FROM callT
4      WHERE date_call BETWEEN '2018-06-01' AND '2018-06-30' AND
5             hour_call BETWEEN '08:00:00' AND '10:00:00' AND duration < 30;
6

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
call_id			
4			
5			

b. Show the first and last name of customers that live in a city with population greater than 20000

```

6
7 -- b) Show the first and last name of customers that live in a city with population greater than 20000
8 • SELECT first_name, last_name
9     FROM customer,city
10    WHERE customer.city_id = city.id AND population > 20000;
11

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
first_name	last_name		
Rita	Sakellariou		
Sakis	Rouvas		
Freddy	Mercury		
Manthos	Foustanos		
Pamela	Anderson		
Anna	Frank		

c. Show the customer id that have a contract in the plan with name LIKE 'Freedom' (use nested queries).

```

12 -- c) Show the customer id that have a contract in the plan with name LIKE 'Freedom' (use nested queries).
13 • SELECT DISTINCT contract.customer_id
14     FROM contract
15    WHERE contract.plan_id IN
16          (SELECT plan.id FROM plan WHERE plan_name LIKE 'Freedom' )
17    ORDER BY customer_id;

```

customer_id
1
2
3
4

d. For each contract that ends in less than sixty days from today, show the contract id, the phone number, the customer's id, his/her first name and his/her last name

```

18
19 /*d) For each contract that ends in less than sixty days from today, show the contract id,
20    the phone number, the customer's id, his/her first name and his/her last name. */
21 • SELECT customer.id AS customer_id,first_name,last_name,contract.id AS contract_id,phone_number
22     FROM customer,contract
23    WHERE customer.id = contract.customer_id AND datediff(ending_date,curdate()) < 60;
24

```

Result Grid					
Filter Rows:					
Export:					
Wrap Cell Content:					
	customer_id	first_name	last_name	contract_id	phone_number
▶	1	Rita	Sakellariou	1	6912345187
	1	Rita	Sakellariou	2	6956192310
	3	Maria	Antouaneta	3	2104449104
	3	Maria	Antouaneta	4	2151237890

e. For each contract id and each month of 2018, show the average duration of calls

```

26  -- e) For each contract id and each month of 2018, show the average duration of calls
27  • SELECT contract_id AS contractId ,MONTH(date_call) AS month, avg(duration) AS avg_duration
28      FROM callT
29      WHERE YEAR(date_call) = '2018'
30      GROUP BY contractId,month
31      ORDER BY contractId,month;

```

Result Grid			
Filter Rows:			
Export:			
Wrap Cell Content:			
	contractId	month	avg_duration
▶	1	6	26.500000
	1	7	61.333333
	10	7	120.000000
	11	7	121.000000
	12	7	110.000000
	13	7	10.000000
	2	6	25.000000
	2	7	23.000000
	3	7	28.000000
	6	7	23.000000
	7	7	150.000000

	8	7	124.500000
	9	7	33.000000

f. Show the total duration of calls in 2018 per plan id

```

31
32  -- f) Show the total duration of calls in 2018 per plan id
33  • SELECT plan.id AS plan_id, SUM(duration) AS total_duration
34      FROM callT,plan,contract
35      WHERE plan.id = contract.plan_id AND contract.id = callT.contract_id AND YEAR(date_call) = '2018'
36      GROUP BY plan.id
37      ORDER BY plan_id;

```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	plan_id	total_duration			
▶	1	639.00			
	2	241.00			
	3	422.00			

g. Show the top called number among TP's customers in 2018

```

39  -- g) Show the top called number among TP's customers in 2018
40  •   SELECT phone_number, SUM(duration) AS max_duration
41         FROM callT
42        WHERE YEAR(date_call) = '2018' AND phone_number IN
43              (SELECT phone_number
44                 FROM contract
45                WHERE contract.phone_number = callT.phone_number)

```

```

46      GROUP BY phone_number
47      ORDER BY max_duration DESC
48      LIMIT 1;
49

```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	phone_number	max_duration				
▶	6912345187	246.00				

h. Show the contract ids and the months where the total duration of the calls was greater than the free minutes offered by the plan of the contract

```

50  /*h. Show the contract ids and the months where the total duration
51     of the calls was greater than the free minutes offered by the plan of the contract */
52  •   SELECT contract.id AS contractId, MONTH(date_call) AS month, SUM(duration) AS sum_duration
53         FROM contract, callT, plan
54        WHERE callT.contract_id = contract.id AND contract.plan_id = plan.id
55        GROUP BY contractId, month, minutes
56        HAVING sum_duration > minutes

```

```

57      ORDER BY contractId, month, minutes;
58

```

Result Grid			
	contractId	month	sum_duration
▶	7	10	622.00

i. For each month of 2018, show the percentage change of the total duration of calls compared to the same month of 2017

```

59 -- i) For each month of 2018, show the percentage change of the total duration of calls compared to the same month of 2017
60 • SELECT month2018, ((duration2018.sum2018 - duration2017.sum2017)/duration2017.sum2017)*100 AS percentage_change
61     FROM (SELECT MONTH(date_call) AS month2017, SUM(duration) AS sum2017 FROM CallT WHERE YEAR(date_call) = '2017' GROUP BY MONTH(date_call)) AS duration2017,
62     (SELECT MONTH(date_call) AS month2018, SUM(duration) AS sum2018 FROM CallT WHERE YEAR(date_call) = '2018' GROUP BY MONTH(date_call)) AS duration2018
63     GROUP BY month2017, month2018
64     HAVING month2017 = month2018
65     ORDER BY month2017, month2018;
66

```

Result Grid		
	month2018	percentage_change
▶	6	290.000000
	7	4120.689655

j. For each city id and calls made in 2018, show the average call duration by females and the average call duration by males (i.e. three columns)

```

67 /* j) For each city id and calls made in 2018, show the average call duration
68    by females and the average call duration by males (i.e. three columns) */
69 • SELECT cityM, male_avg.male_duration AS avg_male, female_avg.female_duration AS avg_female
70     FROM (SELECT city_id AS cityM, avg(duration) AS male_duration FROM callT, customer, contract
71            WHERE gender = 'm' AND customer.id = contract.customer_id AND contract.id = callT.contract_id AND YEAR(date_call) = '2018'
72            GROUP BY city_id) AS male_avg,
73     (SELECT city_id AS cityF, avg(duration) AS female_duration FROM callT, customer, contract
74            WHERE gender = 'f' AND customer.id = contract.customer_id AND contract.id = callT.contract_id AND YEAR(date_call) = '2018'

```

```

75     GROUP BY city_id) AS female_avg
76     GROUP BY cityM, cityF
77     HAVING cityM = cityF
78     ORDER BY cityM, cityF;
79

```


Result Grid			
	cityM	avg_male	avg_female
▶	1	90.666667	38.500000
	2	150.000000	76.500000
	3	10.000000	86.333333

k. For each city id, show the city id, the ratio of the total duration of the calls made from customers staying in that city in 2018 over the total duration of all calls made in 2018, and the ratio of the city's population over the total population of all cities (i.e three columns)

```

80  /* k. For each city id, show the city id, the ratio of the total duration of the calls made from customers staying in that city
81     in 2018 over the total duration of all calls made in 2018, and the ratio of the city's population over the total population of all cities (i.e three columns) */
82  • SELECT cityPop , Duration.sum_duration / total_duration.total AS ratio_duration , Population.population / SUM(city.population) AS ratio_population
83     FROM (SELECT city_id AS cityDur,SUM(duration) AS sum_duration FROM callT,customer,contract
84            WHERE customer.id = contract.customer_id AND contract.id = callT.contract_id AND YEAR(date_call) = '2018' GROUP BY city_id) AS Duration,
85            (SELECT city_id AS cityPop,population FROM callT,customer,contract,city
86            WHERE city.id = customer.city_id AND customer.id = contract.customer_id AND contract.id = callT.contract_id GROUP BY city_id) AS Population,
87            (SELECT sum(duration) AS total FROM callT WHERE YEAR(date_call) = '2018') total_duration,

```

```

88  city
89  GROUP BY cityDur,cityPop
90  HAVING cityDur = cityPop
91  ORDER BY cityDur,cityPop;

```

Result Grid			
	cityPop	ratio_duration	ratio_population
▶	1	0.445469	0.397059
	2	0.347926	0.323529
	3	0.206605	0.279412

4. (20%) Using the programming language of your choice, connect to the database and implement query (k) above – **without using GROUP BY SQL statements**,

The programming language that we used in order to connect to the database and implement the query, was JAVA through eclipse application. We downloaded the “mysql-connector-java-5.1.48-bin.jar” and then we applied it as an external jar in the eclipse program. After having connected to the database, we implemented 2 queries and saved the results in four lists :

cityDurList: the list of cities that the duration of calls of 2018 is refer to.

durationList: the list durations of calls made in 2018.

cityPopList: the list of cities that the population is refer to. (all the cities of the database)
populationList : the list of populations of cities.

For every city j of the total cities of the database, if there are calls made from this city , then calculate the total duration of calls referring to this city and the ratio of duration. Calculate the ratio of population, as well.
Here is the code:

```
*telecom.java
1 package f2821907;
2
3 import java.sql.*;
4
5
6 class telecom{
7     public static void main(String args[]){
8         try{
9             // This will load the MySQL driver
10            Class.forName("com.mysql.jdbc.Driver");
11            //connecting to database using username and password of database
12            Connection con=DriverManager.getConnection(
13                "jdbc:mysql://localhost:3306/telecom","root","putyourSqlCode");
14            // Statements allow to issue SQL queries to the database
15            Statement stmt=con.createStatement();
16            Statement stmt2=con.createStatement();
17            // Result set get the result of the SQL query
18            ResultSet rs=stmt.executeQuery("select city_id,duration from customer,callt,contract "
19                + " where customer.id = contract.customer_id AND contract.id = callt.contract_id AND YEAR(date_call) = '2018' ");
20            ResultSet rs2 =stmt2.executeQuery("select id,population from city");
21
22            //ArrayList that will contain the city per duration
23            ArrayList<String> cityDurList = new ArrayList<String>();
24            //ArrayList for duration
25            ArrayList<Double> durationList = new ArrayList<Double>();
26            while(rs.next()){
27                cityDurList.add(rs.getString(1));
28                durationList.add(rs.getDouble(2));
29            }
30
31
32            //ArrayList that will contain the city per population
33            ArrayList<String> cityPopList = new ArrayList<String>();
34            //ArrayList for duration
35            ArrayList<Double> populationList = new ArrayList<Double>();
36            while(rs2.next()){
37                cityPopList.add(rs2.getString(1));
38                populationList.add(rs2.getDouble(2));
39            }
40            con.close();
41
42            //total duration of calls of all cities
43            double total_duration = 0;
44            for(Double temp:durationList){
45                total_duration += temp;
46            }
47        }
48    }
49 }
```

```

47     //total population of all cities
48     double total_population = 0;
49     for(Double temp:populationList){
50         total_population += temp;
51     }
52
53     //titles of columns
54     System.out.println("CityId   DurationRatio       PopulationRatio ");
55
56     for(int j=0; j < cityPopList.size(); j++ ){
57         double ratio_duration = 0;
58         //if there are calls made from this city
59         if(cityDurList.contains(cityPopList.get(j))){
60             double sum = 0;
61             for(int i=0; i < durationList.size(); i++){
62
63                 if(cityDurList.get(i).equals(cityPopList.get(j))){
64                     sum += durationList.get(i);
65                 }
66             }
67             ratio_duration = sum / total_duration;
68         }
69         double ratio_population = populationList.get(j) / total_population;
70
71         //printing the results
72         System.out.println(cityPopList.get(j)+"      "+ratio_duration+"      "+ratio_population);
73
74     }
75 }
76 }catch(Exception e){
77     System.out.println(e);
78 }
79
80
81
82
83 }
84 }

```

Problems @ Javadoc Declaration Console

<terminated> telecom [Java Application] C:\Program Files\Java\

Wed Nov 20 03:55:21 EET 2019 WARN: Establishing

CityId	DurationRatio	PopulationRatio
1	0.445468509984639	0.39705882352941174
2	0.347926267281106	0.3235294117647059
3	0.206605222734255	0.27941176470588236