 **Технически Университет – София**

**Катедра „Компютърни системи”**

**Специалност “Компютърно и Софтуерно Инженерство”, бакалавър**

**КУРСОВА РАБОТА**

**по Бази от данни**

**Студент: Християн Цветанов Зарков**

**ФАК. № 501215038 Група: 52**

|  |
| --- |
| **Тема** |
| Разработете база данни за следене на хора. Пази се информация за проследяваните хора – три имена, ЕГН, адрес, имейл, телефонен номер, семеен статус, кола, номер на обувките, размер на дрехите и всякаква друга информация. Пази се информация за често посещавани места от субектите – държава, град, адрес, GPS координати, вид на мястото (заведение, библиотека, ресторант, бар, кръчма и т.н.). Пази се и информация за точните посещения – дата, час, колко време са седели, сами ли са били, ако не с кого. |

1. Да се проектира база от данни и да се представи ER диаграма със съответни CREATE TABLE заявки за средата MySQL.
2. Напишете заявка, в която демонстрирате SELECT с ограничаващо условие по избор.
3. Напишете заявка, в която използвате агрегатна функция и GROUP BY по ваш избор.
4. Напишете заявка, в която демонстрирате INNER и OUTER JOIN по ваш избор.
5. Напишете заявка, в която демонстрирате едновременно JOIN и агрегатна функция.
6. Създайте процедура, в която демонстрирате използване на курсор.

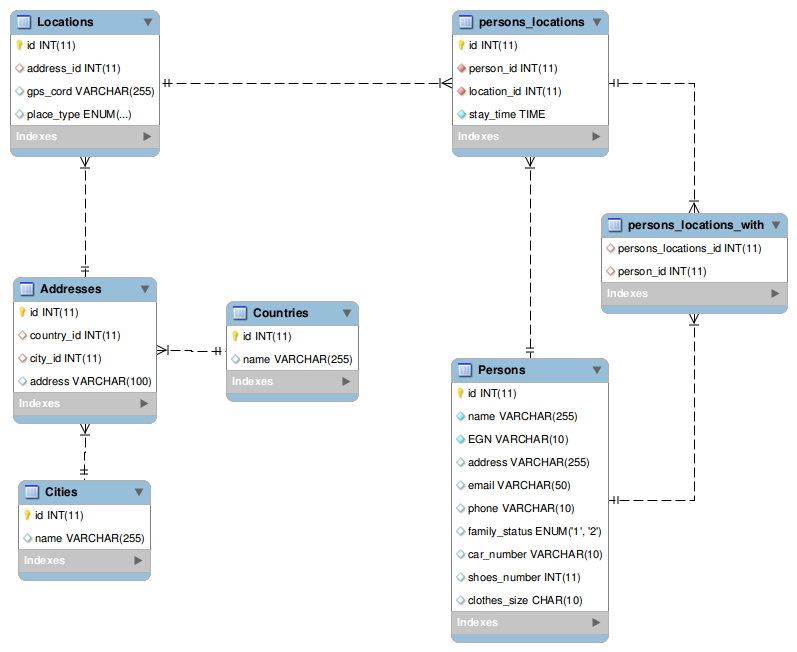
*Вашата работа трябва да включва: задание, ER–диаграма, CREATE TABLE заявки, всички останали заявки, решения на задачите от 2 до 5 и резултатите от тях.*

София, Подпис на преподавателя:**……………………………**

2016/2017 (ас. Петко Данов)

Подпис на студента:**………………….**

1. Да се проектира база от данни и да се представи ER диаграма със съответни CREATE TABLE заявки за средата MySQL.
2. ER диаграма



1. Код за създаване на таблиците

**DROP DATABASE IF EXISTS Stalkers;**

**CREATE DATABASE Stalkers;**

**use Stalkers;**

**CREATE TABLE Persons(**

**id INT PRIMARY KEY auto\_increment,**

**name varchar(255) NOT NULL,**

**EGN varchar(10) NOT NULL unique,**

**address varchar(255),**

**email varchar(50) unique,**

**phone varchar(10) unique,**

**family\_status enum('1', '2'),**

**car\_number varchar(10) unique,**

**shoes\_number int,**

**clothes\_size char(10)**

**);**

**CREATE TABLE Countries(**

**id int PRIMARY KEY auto\_increment,**

**name varchar(255)**

**);**

**CREATE TABLE Cities(**

**id int PRIMARY KEY auto\_increment,**

**name varchar(255)**

**);**

**CREATE TABLE Addresses(**

**id int PRIMARY KEY auto\_increment,**

**country\_id int,**

**city\_id int,**

**address varchar(100),**

**FOREIGN KEY (country\_id) REFERENCES Countries(id),**

**FOREIGN KEY (city\_id) REFERENCES Cities(id)**

**);**

**CREATE TABLE Locations(**

**id int PRIMARY KEY auto\_increment,**

**address\_id int,**

**gps\_cord varchar(255),**

**place\_type enum('bar', 'club', 'sport'),**

**FOREIGN KEY (address\_id) REFERENCES Addresses(id)**

**);**

**CREATE TABLE persons\_locations(**

**id int PRIMARY KEY auto\_increment,**

**person\_id int NOT NULL,**

**location\_id int NOT NULL,**

**stay\_time TIME NOT NULL,**

**FOREIGN KEY (person\_id) REFERENCES Persons(id),**

**FOREIGN KEY (location\_id) REFERENCES Locations(id),**

**UNIQUE KEY(person\_id, location\_id)**

**);**

**CREATE TABLE persons\_locations\_with(**

**persons\_locations\_id int,**

**person\_id int,**

**FOREIGN KEY (persons\_locations\_id) REFERENCES persons\_locations(id),**

**FOREIGN KEY (person\_id) REFERENCES Persons(id)**

**);**

1. Използвани Insert заявки за запълване на таблицата.

**Insert Into Persons values(NULL, 'Gosho', '46453198', NULL, 'hrerowe@grpiw.ngre', '0864685210', '1', 'Ca07763KS', '47', 'L');**

**Insert Into Persons values(NULL, 'Pesho', '46423199', NULL, 'gree@grpiw.ngre', '0864685211', '1', 'Ca07163KS', '47', 'L');**

**Insert Into Persons values(NULL, 'Petka', '46433197', NULL, 'wfv@grpiw.ngre', '0864685212', '1', 'Ca07743KS', '47', 'T');**

**Insert Into Persons values(NULL, 'Kolio', '54453196', NULL, 'greerht54@grpiw.ngre', '0864385210', '1', 'CO07453KS', '47', 'M');**

**Insert Into Persons values(NULL, 'Ivan', '95453194', NULL, '5yr@grpiw.ngre', '0864685216', '1', 'Ca07763MS', '47', 'M');**

**Insert Into Persons values(NULL, 'Maria', '56453193', NULL, '4ht@grpiw.ngre', '0864685218', '1', 'Ca07763LS', '47', 'S');**

**Insert Into Persons values(NULL, 'Ivanka', '01453192', NULL, 'gehn@grpiw.ngre', '0864685910', '1', 'Ca07763QS', '47', 'S');**

**Insert Into Countries values(NULL, 'Bulgaria');**

**Insert Into Countries values(NULL, 'Russia');**

**Insert Into Countries values(NULL, 'Germany');**

**Insert Into Cities values(NULL, 'Sofia');**

**Insert Into Cities values(NULL, 'Plovdiv');**

**Insert Into Cities values(NULL, 'Varna');**

**Insert Into Addresses values(NULL, 1, 1, 'greghjebgreo');**

**Insert Into Addresses values(NULL, 1, 2, 'hgrtrbgreo');**

**Insert Into Addresses values(NULL, 1, 3, 'ebgreo');**

**Insert Into Locations values(NULL, 1, '50, 2000 -20.6545', 'bar');**

**Insert Into Locations values(NULL, 1, '20, 5400 -10.6545', 'club');**

**Insert Into Locations values(NULL, 1, '30, 1000 -80.6545', 'sport');**

**Insert Into persons\_locations values(NULL, 1, 1, '2:12:50');**

**Insert Into persons\_locations values(NULL, 1, 2, '4:25:00');**

**Insert Into persons\_locations values(NULL, 1, 3, '1:55:40');**

**Insert Into persons\_locations\_with values(1, 2);**

**Insert Into persons\_locations\_with values(1, 3);**

**Insert Into persons\_locations\_with values(1, 4);**

1. Да се извадят всички хора който носят размер на дрехите L

**Select Persons.name From Persons Where clothes\_size='L';**

Избираме всичките хора и ограничаваме да вземем само тези който са с размер “L”.

Резултат:

'Gosho'

'Pesho'

1. Да се извадят бройката на посетените места за всеки човек

**Select Persons.clothes\_size, COUNT(Persons.id) From Persons Group by clothes\_size;**

Избират се всичките хора групират се по размер и се принтира броят на хората с даден размер.

Резултат:

'L', '2'

'M', '2'

'S', '2'

'T', '1'

1. Да се извади за всяка локация колко човека са я посетили включително придружителите

**Select Countries.name, Cities.name, Addresses.address, Locations.gps\_cord, Locations.place\_type, (COUNT(persons\_locations.id) + (Select COUNT( persons\_locations\_with.person\_id) From persons\_locations\_with**

**Left Join persons\_locations On persons\_locations\_with.persons\_locations\_id = persons\_locations.id where persons\_locations.location\_id = Locations.id)) as PersonsCount From Locations Join Addresses On Locations.address\_id = Addresses.id Join Cities On Addresses.city\_id = Cities.id Join Countries On Addresses.country\_id = Countries.id Left Join persons\_locations On Locations.id = persons\_locations.location\_id Group by Locations.id;**

Избират се всички локации и се добавят адресите. Добавят се и хората който са проследени до тези локации, след което се прави вложен Select за да може да се видят и хората, които може да не са проследени специално, но са били с друг проследен човек на даденото място.

Резултат:

'Bulgaria', 'Sofia', 'greghjebgreo', '50, 2000 -20.6545', 'bar', '4'

'Bulgaria', 'Sofia', 'greghjebgreo', '20, 5400 -10.6545', 'club', '1'

'Bulgaria', 'Sofia', 'greghjebgreo', '30, 1000 -80.6545', 'sport', '1'

1. Да се извади за всеки човек броят на посетените локации(може и като придружител само да е)

**Select Persons.name, (Count(persons\_locations.id)+(Select Count(\*) From persons\_locations\_with Where Persons.id = persons\_locations\_with.person\_id )) as LocationsCount From Persons Left Join persons\_locations on persons\_locations.person\_id = Persons.id Group by Persons.id;**

Показват се имената на всички хората, които има в базата и се броят местата на които са били. Използва се вложен Select за да може да се видят и локациите на който са ходили дори само като придружители (тоест не са били следени специално в това време, но се знае, че са били там).

Резултат:

'Gosho', '3'

'Pesho', '1'

'Petka', '1'

'Kolio', '1'

'Ivan', '0'

'Maria', '0'

'Ivanka', '0'

1. Да се изпишат имената на следените хора чрез курсор.

**drop procedure if exists CursorProcedure;**

**delimiter |**

**create procedure CursorProcedure()**

**begin**

**declare finished int;**

**declare current\_name varchar(255);**

**declare the\_cursor CURSOR for**

**Select distinct Persons.name From persons\_locations Join Persons on Persons.id=persons\_locations.person\_id;**

**declare continue handler FOR NOT FOUND set finished = 1;**

**set finished = 0;**

**OPEN the\_cursor;**

**coach\_loop: while( finished = 0)**

**DO**

**FETCH the\_cursor INTO current\_name;**

**IF(finished = 1)**

**THEN**

**LEAVE coach\_loop;**

**END IF;**

**Select current\_name;**

**end while;**

**CLOSE the\_cursor;**

**SET finished = 0;**

**SELECT 'Finished!';**

**end;**

**|**

**delimiter ;**

**Резултат 1:**

**'Gosho'**