**Курсов проект по бази данни**

**на тема**

**Охранителна фирма**

Изготвено от :

Владимир Янакиев ФН: 62344

Борис Трифонов ФН: 62343

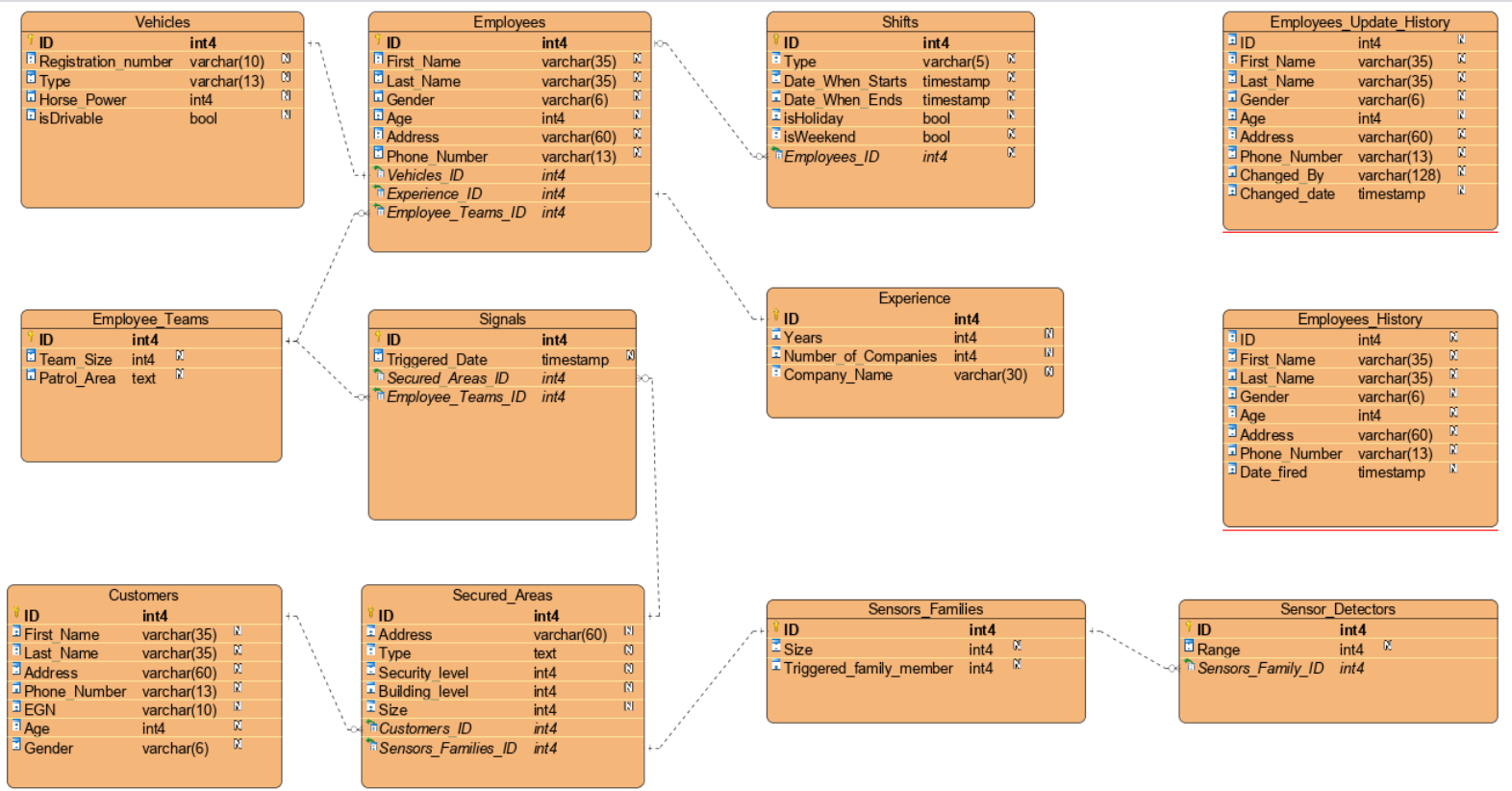
Йордан Ангелов ФН: 62263

Разпределение на задачите

|  |  |
| --- | --- |
| Задача | Фак.№ |
| 1. Създаване на схемите и релациите | 62263 |
| 2. Добавяне на съдържание | 62263 |
| 3. Примерни прости заявки и заявки върху две и повече релации | 62343 |
| 4. Примери с подзаявки : | 62344 |
| 5. Примери със съединения | 62343 |
| 6. Примери с групиране и аграгация : | 62344 |
| 7. Примери с ограничения | 62263 |
| 8. Примери с изгледи и индекси : | 62344 |
| 9. Примери с тригери | 62343 |
| 10. Подготовка на презентацията | 62343 |

Проекта, който изпълняваме, ще бъде за конкретна охранителна фирма, намираща се на територията на град Видин. Фирмата има нужда от база данни за схематизиране и оптимизиране на работата на конкретните компоненти и връзките между тях. Наличието на голям брой автомобили, обекти и служители изисква дигитализация на тази информация. В базата от данни на Security Company ™ се съхранява информация за охраняваните обекти, техните семейства от детектори, клиенти както и служителите заедно с тяхната информация,група и превозно средство.

**Дефинициране на схемата на релациите:**

Таблицата **Vehicles** съдържа информация за превозните средства:

* ID – идентификационен номер, първичен ключ ;
* Registration Number – регистрационен номер ;
* Type – тип на автомобила, hatchback – ‘hatchback’ , sedan – ‘sedan’, coupe – ‘coupe’, високопроходима – ‘high passable’;
* Horse\_Power – конски сили,измерва се в HP;
* IsDrivable – дали колата е в състояние, подходящо за работа , Boolean – ‘0’, ‘1’ .

Таблицата **Employees** съдържа информация за служителите във фирмата. Атрибутите Vehicles\_ID ,Experience\_ID и Employee\_Teams\_ID са външни ключове съответно към Vehicles, Experience и Employee\_Teams :

* ID – идентификационен номер, първичен ключ ;
* First\_name - име;
* Last\_name - фамилия;
* Gender - пол , мъж – ‘male’ , жена – ‘female’;
* Age – възраст;
* Address – адрес;
* Phone\_number – телефонен номер;
* Vehicles\_ID – идентификационен номер на превозното средство, външен ключ към Vehicles.ID ;
* Experience\_ID – идентификационен номер на превозното средство, външен ключ към Experience.ID ;
* Employee\_Teams\_ID – идентификационен номер на групата, от която е служителят (ако съществува такава) – външен ключ към Employee\_Teams ;

Таблицата **Employee\_Teams** съдържа информация за групите от служители:

* ID – идентификационен номер, първичен ключ ;
* Team\_Size – броя на членовете в екипа ;
* Patrol\_Area – областта, в която обикаля екипът .

Таблицата **Shifts** съдържа информация за графика на служителите във фирмата. Атрибутът Employees\_ID е външен ключ към Employees :

* ID – идентификационен номер, първичен ключ ;
* Type – тип , дневна – ‘day’ , нощна – ‘night’ ;
* Date\_When\_Starts – начало на смяната ;
* Date\_When\_Ends – край на смяната ;
* isHoliday – дали денят е празник, празник – ‘yes’ , не е празник – ‘no’ ;
* isWeekend – дали денят е делник, уикенд – ‘yes’, делник – ‘no’ ;
* Employees\_ID – идентификационен номер на служител, външен ключ към Employees.ID .

Таблицата **Experience** съдържа информация за предишен опит на служителите от фирмата.

* ID – идентификационен номер, първичен ключ ;
* Years – години стаж ;
* Number\_of\_Companies – броят на компаниите, в които е бил служителят ;
* Company\_Name – Име на компания , в които е бил служителят .

Таблицата **Customers** съдържа информация за клиентите на фирмата.

* ID – идентификационин номер, първичен ключ ;
* First\_Name – Име ;
* Last\_Name – Фамилия ;
* Address – Адрес на клиента ;
* Phone\_Number – телефонен номер ;
* EGN – ЕГН ;
* Age – Възраст ;
* Gender – пол на клиента , мъж – ‘male’, жена – ‘female’ .

Таблицата **Secured\_Areas** съдържа информация за територията, в която е осигурена защита от компанията, както и типът на защитата, действаща на дадения адрес, нивото на сигурност и квадратурата на територията.Атрибутите Customers\_ID и Sensors\_Families\_ID са външни ключове съответно към Customers и Sensors\_Families :

* ID – идентификационен номер; първичен ключ;
* Address – точен адрес;
* Type – тип на действащата защита;
* Security level – ниво на сигурност;
* Building level – етаж;
* Size – размер на територията (в кв.м.);
* Customers ID – идентификационен номер на клиента, поръчал услугата, външен ключ към Customers ;
* Sensors\_Families\_ID - идентификационен номер на фамилията от сензори, работещи на съответната територия, външен ключ към Sensors\_Families .

Таблицата **Sensors\_Families** съдържа информация за фамилиите от сензори по различните обекти.

* ID – идентификационен номер, първичен ключ ;
* Size – големина на семейство ;
* Triggered\_family\_member – конкретен задействан сензор .

Таблицата **Sensor\_Detectors** съдържа информация за сензорите, използвани за улавяне на движение на територията, в която е активирана услуга за защита.Атрибутът Sensors\_Family\_ID е външен ключ към Sensors\_Family :

* ID – идентификационен номер на сензор; първичен ключ ;
* Range – размер на обхват (в кв.см.) ;
* Sensors\_Family\_ID – идентификационен номер на фамилията от сензори, на която принадлежи текущият, външен ключ към Sensors\_Families .

Таблицата **Signals** съдържа информация за сигналите за влизане с взлом. Атрибутът Secured-Areas\_ID и Employee\_Teams\_ID са външни ключове съответно на Secured\_Areas и Employee\_Teams :

* ID – идентификационен номер, първичен ключ ;
* Triggered\_Date – дата на която е записан сигнала ;
* Secured\_Areas\_ID – информация за мястото на сигнала, външен ключ към Secured\_Areas.ID .
* Employee\_Teams\_ID – идентификационен номер на групата, от която е служителят (ако съществува такава) – външен ключ към Employee\_Teams ;

**Създаване на таблиците в Microsoft SQL Server:**

CREATE TABLE Vehicles

(

ID int primary key identity,

Registration\_Number varchar(10),

Type varchar(13),

Horse\_Power int,

isDrivable int

);

CREATE TABLE Employees

(

ID int primary key identity,

First\_Name varchar(35),

Last\_Name varchar(35),

Gender char(6),

Age int,

Address varchar(60),

Phone\_Number varchar(13),

Vehicles\_ID int,

Experience\_ID int,

Employee\_Teams\_ID int

)

CREATE TABLE Employee\_Teams

(

ID int primary key identity,

Team\_Size int,

Patrol\_Area text

)

CREATE TABLE Shifts

(

ID int primary key identity,

Type varchar(5),

Date\_When\_Starts datetime2(0),

Date\_When\_Ends datetime2(0),

isHoliday int,

isWeekend int,

Employees\_ID int

)

CREATE TABLE Experience

(

ID int primary key identity,

Years int,

Number\_of\_Companies int,

Company\_Name varchar(30)

)

CREATE TABLE Customers

(

ID int primary key identity,

First\_Name varchar(35),

Last\_Name varchar(35),

Address varchar(60),

Phone\_Number varchar(13),

EGN varchar(10),

Age int,

Gender varchar(6)

)

CREATE TABLE Secured\_Areas

(

ID int primary key identity,

Address varchar(60),

Type text,

Security\_Level int,

Building\_Level int,

Size int,

Sensors\_Families\_ID int,

Customers\_ID int

)

CREATE TABLE Sensors\_Families

(

ID int primary key identity,

Size int,

Triggered\_Family\_Members int

)

CREATE TABLE Sensor\_Detectors

(

ID int primary key identity,

range int,

Sensors\_Family\_ID int

)

CREATE TABLE Signals

(

ID int primary key identity,

Triggered\_Date datetime2(0),

Secured\_Areas\_ID int,

Employee\_Teams\_ID int,

)

Таблици, които показват промяната и изтриването на данните от съответната таблица, датата на промяна и потребителят,който ги е променил.

CREATE TABLE Employees\_Update\_History(

ID int,

First\_Name varchar(35),

Last\_Name varchar(35),

Gender char(6),

Age int,

Address varchar(60),

Phone\_Number varchar(13),

Changed\_By varchar(128),

Changed\_date datetime2

)

CREATE TABLE Employees\_History(

ID int ,

First\_Name varchar(35),

Last\_Name varchar(35),

Gender char(6),

Age int,

Address varchar(60),

Phone\_Number varchar(13),

Date\_fired datetime2

)

**Добавяне на примерно съдържание :**

INSERT INTO VEHICLES(Registration\_Number,Type,Horse\_Power,isDrivable)

VALUES( 'ВН 2012 АК' , 'sedan', 120, 1 ),

('ВН 2301 АК' , 'hatchback', 120, 1 ),

('ВН 3021 АК' , 'sedan', 60, 0 ),

( 'ВН 5015 АК' , 'coupe', 160, 0 ),

( 'ВН 2701 АК' , 'hatchback', 75, 1 ),

( 'ВН 2302 АК' , 'coupe', 230, 1 ),

( 'ВН 9960 АК' , 'high passable', 135, 0 ),

( 'ВН 7260 АК' , 'high passable', 90, 1 )

INSERT INTO EMPLOYEES( First\_Name,Last\_name,Gender,Age,Address,Phone\_Number,Vehicles\_ID,Experience\_ID,Employee\_Teams\_ID)

VALUES

('Boris','Ivanov','MALE',24,'Himik 12','+359894452348', 1 , 2, 3),

('Ivana','Todorova','FEMALE',21,'Himik 13','+359884865132', 2 , 3, 4),

('Bako','Zahariev','MALE',45,'Himik 14','+359888761365', 1 , 3 , 5),

('Toshko','Nedev','MALE',29, 'BabaTonka 4','+359894586663', 5 , 6 , 2),

('Gergana','Zaharieva','FEMALE',45,'Balnava 1','+359895216542', 2 , 1 , 6),

('Teodor','Ivanov','MALE',36,'Kiranaru 45','+359875895219', 7, 1 , 5)

INSERT INTO Experience( Years,Number\_of\_Companies, Company\_Name)

VALUES ( 4 , 2 , 'СОТ 161'),

( 0 , 0 , NULL),

( 3 , 2 , 'СОТ СОВА 7'),

( 1 , 3 , 'АЛФА СОТ'),

( 7 , 1 , 'АЛФА ЩИТ'),

( 10 , 1 , 'СОТ 161')

INSERT INTO Employee\_Teams(Team\_Size,Patrol\_Area)

VALUES ( 3 , 'Bononia'),

( 5 , 'GM'),

( 2 , 'Simeonovo'),

( 4 , 'Bosna'),

( 3 , 'Botevo'),

( 4 , 'Vardulevo')

Insert into CUSTOMERS(First\_Name,Last\_Name, Address , Phone\_Number , EGN, Age , Gender)

VALUES

('Iordan','Manchev','Bononiq','+359895603861','7511017896', 26, 'MALE'),

('Mani','Manchev','Himik 12','+359895659874','7505097896', 45, 'MALE'),

('Romeo','Ivanov','Bononiq','+359892679071','7511127896', 36, 'MALE'),

('Snejana','Petrova','Kaleto','+359897669824','7808069521', 25, 'FEMALE'),

('Veni','Palata','Akjamiq','+359895692199','7801014825', 21, 'MALE'),

('Joro','Grebeca','Bononiq','+359898958872','8512124589', 27, 'MALE'),

('Iordan','Ganchev','Bononiq','+359893603011','7511012729', 30, 'MALE')

INSERT INTO Shifts( Type,Date\_when\_Starts, Date\_when\_ends,isHoliday,isWeekend,Employees\_ID)

VALUES

('day', '20120101 06:00', '20120101 14:00', 0, 1, 6),

('night', '2001-02-06 20:00', '20010207 06:00', 0, 1, 2),

('day', '19990501 13:00', '19990501 21:00', 1, 1, 3),

('night', '19510701 22:00', '19510702 06:00', 0, 1, 5),

('night', '20151205 00:00', '20151205 08:00', 0, 1, 7),

('day', '19781101 07:00', '19781101 15:00', 1, 0, 4),

('night', '20200502 16:00', '20200503 02:00', 1, 0, 3)

Insert into Sensors\_families(Size,Triggered\_Family\_Members)

VALUES

(12, null),

(8, null),

(4, null),

(6, null),

(10, null),

(2, null)

INSERT INTO Sensor\_Detectors(range,Sensors\_Family\_ID)

VALUES

(1000, 1),

(1542, 4),

(999, 2),

(863, 3),

(756, 5),

(1024, 6)

Insert into Secured\_Areas( Address,Type,Security\_Level,Building\_level,Size,Sensors\_Families\_ID,Customers\_ID)

VALUES

('Baba Tonka 3', 'Building', 5, 1, 45, 1, 1),

('Botevo', 'Field', 6, null, 4500, 2, 2),

('Simeonovo', 'Farm', 9, null, 1200, 3, 3),

('Himik 12', 'Building', 8, 4, 80, 4, 4),

('Akjamiq 4', 'Building', 10, 12, 182, 5, 5),

('Dianabad 45', 'Pool', 4, null, 200, 6, 6),

('Baba Tonka 3', 'Building', 5, 1, 45, 1,9)

INSERT INTO Signals(Triggered\_Date,Secured\_Areas\_ID,Employee\_Teams\_ID)

VALUES

('20180203 16:00', 4 , 2),

('20011106 08:45', 2 , 6),

('20190112 22:50', 3 , 4),

('20130615 12:25', 1 , 3),

('19950712 05:20', 6 , 5),

('19970322 14:30', 5 , 4),

('20050430 23:00', 1 , 7)

**Примерни заявки:**

Прости заявки:

1. Заявка, която извежда броят високопроходими автомобили, които имат идентификационен номер по-голям от 5

и чиито регистрационен номер съдържа '7260' .

select \*

from vehicles

where ID > 5 and Type = 'high passable' and Registration\_Number like '%7260%'



2. Заявка, която извежда броят служители с възраст под 35 години и чиито имена започват с 'T'.

select \*

from Employees

where First\_Name like 'T%' and Age < 35



3. Заявка, която извежда идентификационен номер и област на патрулиране за групите служители, при

които броят на служители в екипа е 3 и адресът е Ботево.

select ID,Patrol\_Area

from Employee\_Teams

where Team\_Size = 3 and Patrol\_Area like 'Botevo'



4. Заявка, която извежда защитените територии от тип сграда или поле със размер - по-голям от 100 кв.м.

select \*

from Secured\_Areas

where (Type like 'Building' or Type like 'Field') and Size > 100



5. Заявка, която извежда сензорните детектори с обхват по-голям от 900, но по-малък от 1300 и с

идентификационен номер на сензорно семейство 2.

select \*

from Sensor\_Detectors

where range > 900 and range < 1300 and Sensors\_Family\_ID = 2



Заявки с много релации:

1. Заявка, която извежда всички данни за Сензорните семейства с размер най-много 10 и сензорните детектори

с обхват поне 1001, сортирани по размер на сензорните семейства.

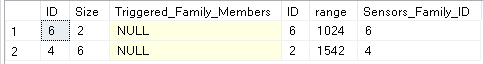
select distinct \*

from Sensors\_Families,Sensor\_Detectors

where Range > 1000 and Size <= 10

and Sensors\_Families.ID = Sensor\_Detectors.Sensors\_Family\_ID

order by Size



2. Заявка, която извежда всички данни за автомобилите и служителите, които имат адрес 'Himik', сортирани

по първо име.

Select \*

from Vehicles, Employees

where vehicles.ID = Employees.Vehicles\_ID

and Address like 'Himik %'

order by First\_Name



3. Заявка, която извежда първото,последно име,адреса и телефона на служителите, които имат в телефонния

си номер '3' и първото им име започва с 'B', и отборът на служителите.

(select First\_name,Last\_Name,Address,Phone\_Number,Team\_Size

from Employee\_Teams,Employees

where Employees.Employee\_Teams\_ID = Employee\_Teams.ID

and Phone\_Number like '%3%')

INTERSECT

(select First\_name,Last\_Name,Address,Phone\_Number,Team\_Size

from Employee\_Teams,Employees

where Employees.Employee\_Teams\_ID = Employee\_Teams.ID

and First\_Name like 'B%')



4. Заявка, която извежда сигналите и защитените места с адрес,започващ с 'B' , при които нивото на сградата не съществува.

select \*

from Signals, Secured\_Areas

where signals.Secured\_Areas\_ID = Secured\_Areas.ID

and Address like 'B%' and Building\_Level is NULL



5. Заявка, която извежда първо,последно име,адрес и идентификационен номер на опита на служителите, които работят

в компания, чието име започва с 'СОТ' и с опит над 5 години, сортирани по последно име на служителя.

select First\_Name,Last\_Name,Address,Experience\_ID

from Experience,Employees

where Experience.ID = Employees.Experience\_ID

and Company\_Name like 'СОТ%' and Years > 5

order by Last\_Name



Подзаявки:

1. Заявка, която извежда всички служители с трудов стаж над 5 години и съответнита им група.

select \*

from Employees, Employee\_Teams

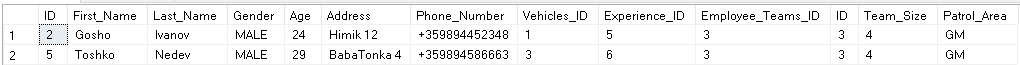
where Employees.Employee\_Teams\_ID = Employee\_Teams.ID

and First\_Name in

(select First\_Name

from Employees,Experience

where employees.Experience\_ID = Experience.ID and Years > 5)



2. Заявка, която извежда всички защитени места и клиенти,чиито адрес е същият като този

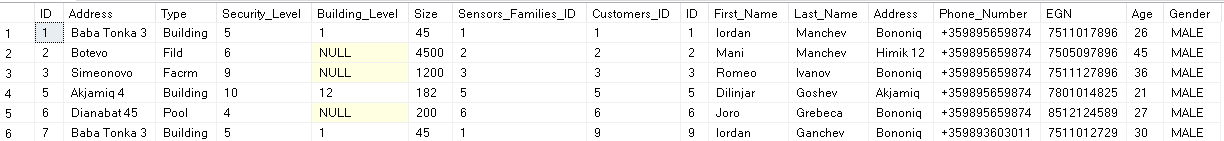
на хората,чието име свършва с 'hev'.

select \*

from Secured\_Areas,Customers

where Secured\_Areas.Customers\_ID = Customers.ID

and customers.Address in (select address from Customers where Last\_Name like '%hev')



3. Заявка, която извежда първо,последно име,телефонен номер,възраст и адрес на най-възрастните служители,

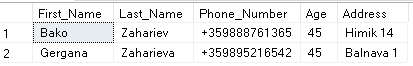
чиито адрес не съвпада с адресите на клиентите.

select First\_Name,Last\_Name,Phone\_Number,Age,Address

from employees,Employee\_Teams

where Employees.Employee\_Teams\_ID = Employee\_Teams.ID

and Age >= all(select Age from Customers) and Address not in (select Address from Customers)



4. Заявка, която извежда първо,последно име,адрес на служителите,чийто идентификационен номер на групата

не е в групите с максимален брой служители .

select First\_Name,Last\_Name,Address,Registration\_Number

from Vehicles,Employees

where Vehicles.ID = Employees.Vehicles\_ID

and Employees.Employee\_Teams\_ID not in

(select ID from Employee\_Teams where Team\_Size >=

all(select Team\_Size from Employee\_Teams))



5. Заявка, която извежда защитените зони, клиентите им и сензорните семейства,които са свързани с тях

и имат най-голям размер.

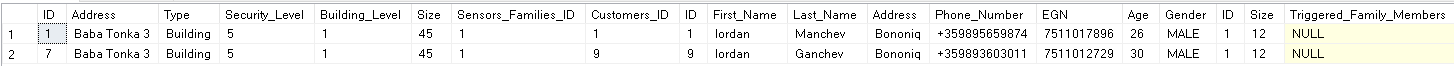
select \*

from Secured\_Areas, Customers, Sensors\_Families

where Secured\_Areas.Customers\_ID = Customers.ID

and Secured\_Areas.Sensors\_Families\_ID = Sensors\_Families.ID

and Sensors\_Families.Size >= all( select Size from Sensors\_Families)



Съединения:

1. Заявка, която извежда служителите и групите им,които имат максимален брой служители и зона на патрулиране 'GM'.

select \*

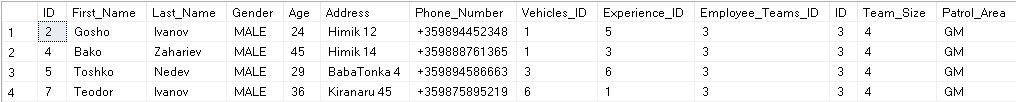
from Employees e

join Employee\_Teams e2

on e.Employee\_Teams\_ID = e2.ID

where e2.Team\_Size >= all(select Team\_Size from Employee\_Teams)

and e2.Patrol\_Area like 'GM'



2. Заявка, която извежда защитеноте места(може и да не имат NULL стойности), при които няма етажи на сградата, и техните сигнали

са на дата започваща с '20' .

select \*

from Secured\_Areas s

left join signals s2 on s2.Secured\_Areas\_ID = s.ID

where Building\_Level is not NULL and Triggered\_Date like '20%'



3. Заявка, която извежда сигналите, служителите(мъже), чието първо име е същото като това на

клиентите с възраст над 30, и техните групи.

select \*

from Signals s

join Employee\_Teams e

on s.Employee\_Teams\_ID = e.ID

join Employees e2

on e.ID = e2.Employee\_Teams\_ID

where Gender like 'Male' and First\_Name in (select First\_Name from Customers where age >30)



4. Заявка, която извежда сензорните детектори с най-малкия обхват,сензорните им семейства, както и защитените места ,

които използват тези семейства с ниво на сигурност поне 5.

select \*

from Sensor\_Detectors s

join Sensors\_Families s2

on s.Sensors\_Family\_ID = s2.ID

join Secured\_Areas s3

on s2.ID = s3.Sensors\_Families\_ID

where Range <= all(select Range from Sensor\_Detectors)

and Security\_Level >= 5



5. Заявка, която извежда опита и служителите, работили в компания,която има име 'АЛФА' ,или с опит повече от 7 години.

select \*

from Employees e

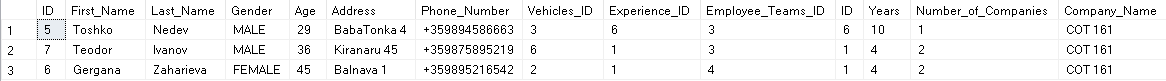
join Experience e2

on e.Experience\_ID = e2.ID

where Company\_Name in

(select Company\_Name from Experience where Company\_Name like 'АЛФА' or Years > 7)

order by Age



Групови заявки / Агрегация

1. Заявка,която извежда средната големина на групата от служители

и общо групите, при които служителите са мъжи и имат първо име 'Gosho',

групирани по големина на групата.

select AVG(Team\_Size) as AVG\_TEAM\_SIZE, COUNT(\*) as ALL\_COUNT

from Employees e1

join Employee\_Teams e2

on e1.Employee\_Teams\_ID = e2.ID

where Gender like 'Male' and First\_name like 'Gosho'

Group by Team\_Size



2. Заявка, която извежда броят на сигналите и сумата от размерите

на защитените места от тип 'Building', при които няма

етаж на сградата.

select COUNT(\*) as ALL\_MATCHED\_LINES,SUM(s2.size) as SUM\_OF\_ALL\_SIZES

from Signals s

join Secured\_Areas s2

on s.Secured\_Areas\_ID = s2.ID

where Building\_Level is not NULL and Type like 'Building'



3. Заявка,която извежда броят на компаниите,които имат име на компания,

започващо със 'СОТ',ако те са поне 3.

select COUNT(\*) as COUNT\_COMPANY

from Experience e1

join Employees e2

on e1.ID = e2.Experience\_ID

where Company\_Name like 'СОТ%'

group by Company\_Name

having COUNT(\*) >= 3



4. Заявка, която извежда средния размер на сензорните семейства

и броят на идентификационните номера на сензорните детектори,

които имат обхват по-голям от 1000.

select AVG(SIZE) as AVG\_SIZE, COUNT(s.ID) as ID\_COUNT

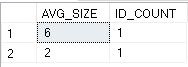
from Sensor\_Detectors s

join Sensors\_Families s2

on s.Sensors\_Family\_ID = s2.ID

where range > 1000

group by Size,s.ID



5. Заявка, която извежда най-голямата възраст на служители,работели

в компанията 'СОТ СОВА 7' и адрес, който съдържа 'Himik' .

select MAX(Age) as MAX\_AGE

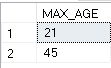
from Experience e1

join Employees e2

on e1.ID = e2.Experience\_ID

where Company\_Name like 'СОТ СОВА 7' and Address like 'Himik%'

group by Age



6. Заявка,която извежда броят на клиентите и фирмите,които имат

ниво на сигурност поне 5, групирани по първо име.

select COUNT(\*) as COUNT\_CUSTOMERS\_SECURED\_AREAS

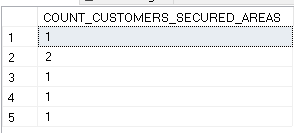
from Customers c1

join Secured\_Areas c2

on c1.ID = c2.Customers\_ID

where Security\_level >= 5

group by First\_Name



7. Заявка, която извежда всички коли, които не са в състояние,

подходящо за работа, групирани по регистрационен номер.

select COUNT(\*) as COUNT\_EMPLOYEE\_VEHICLES\_FOR\_REPAIR

from Vehicles v1

join Employees e1

on v1.ID = e1.Vehicles\_ID

where isDrivable = 0

group by Registration\_number



8. Заявка, която извежда средния размер на защитените места, които

имат ниво на сигурност най-много 5, групирани по размер.

select AVG(s2.Size) as AVG\_SIZE

from Sensors\_Families s1

join Secured\_Areas s2

on s1.ID = s2.Sensors\_Families\_ID

where Security\_level <= 5

group by s2.Size



9. Заявка, която извежда средната възраст на клиентите,които

са на фирми с ниво на сигурност поне 3.

select AVG(Age) as AVG\_CUSTOMERS\_AGE

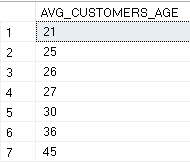
from Customers c1

join Secured\_Areas s1

on c1.ID = s1.Customers\_ID

where Security\_level >=3

group by Age



10. Заявка, която извежда броя и идентификационния на служителите,

които имат смени,когато е и празник, и уикенд.

select COUNT(\*) COUNT\_EMPLOYEE\_SHIFTS, Employees\_ID

from Shifts s1

join Employees e1

on e1.ID = s1.Employees\_ID

where (isHoliday = 0) AND (isWeekend = 0)

group by e1.ID,Employees\_ID



**Ограничения**

FOREIGN KEYS:

TABLE EMPLOYEES:

ALTER TABLE EMPLOYEES ADD CONSTRAINT VEHICLES\_ID\_FK FOREIGN KEY(VEHICLES\_ID) REFERENCES VEHICLES(ID)

ALTER TABLE EMPLOYEES ADD CONSTRAINT EXPERIENCE\_ID\_FK FOREIGN KEY(EXPERIENCE\_ID) REFERENCES EXPERIENCE(ID)

ALTER TABLE EMPLOYEES ADD CONSTRAINT EMPLOYEE\_TEAMS\_ID\_FK FOREIGN KEY(EMPLOYEE\_TEAMS\_ID) REFERENCES EMPLOYEE\_TEAMS(ID)

TABLE SHIFTS :

ALTER TABLE SHIFTS ADD CONSTRAINT EMPLOYEES\_ID\_FK FOREIGN KEY(EMPLOYEES\_ID) REFERENCES EMPLOYEES(ID)

TABLE SIGNALS :

ALTER TABLE SIGNALS ADD CONSTRAINT SECURED\_AREAS\_ID\_FK FOREIGN KEY(SECURED\_AREAS\_ID) REFERENCES SECURED\_AREAS(ID)

ALTER TABLE SIGNALS ADD CONSTRAINT EMPLOYEE\_TEAMS\_ID\_SIGNALS\_FK FOREIGN KEY(EMPLOYEE\_TEAMS\_ID) REFERENCES EMPLOYEE\_TEAMS(ID)

TABLE SECURED AREAS:

ALTER TABLE SECURED\_AREAS ADD CONSTRAINT CUSTOMERS\_ID\_FK FOREIGN KEY(CUSTOMERS\_ID) REFERENCES CUSTOMERS(ID)

ALTER TABLE SECURED\_AREAS ADD CONSTRAINT SENSORS\_FAMILIES\_ID\_FK FOREIGN KEY(SENSORS\_FAMILIES\_ID) REFERENCES SENSORS\_FAMILIES(ID)

TABLE SENSOR DETECTORS:

ALTER TABLE SENSOR\_DETECTORS ADD CONSTRAINT SENSORS\_FAMILY\_ID\_SENSORS\_FK FOREIGN KEY(SENSORS\_FAMILY\_ID) REFERENCES SENSORS\_FAMILIES(ID)

CHECK :

VEHICLES :

ALTER TABLE VEHICLES ADD CONSTRAINT CHK\_HORSE\_POWER CHECK (HORSE\_POWER > 50 AND HORSE\_POWER < 1000)

ALTER TABLE VEHICLES ADD CONSTRAINT CHK\_ISDRIVABLE CHECK (ISDRIVABLE = 0 OR ISDRIVABLE = 1)

EMPLOYEES :

ALTER TABLE EMPLOYEES ADD CONSTRAINT CHK\_GENDER CHECK (GENDER LIKE 'MALE' OR GENDER LIKE 'FEMALE')

ALTER TABLE EMPLOYEES ADD CONSTRAINT CHK\_AGE CHECK (AGE > 20 AND AGE < 50)

ALTER TABLE EMPLOYEES ADD CONSTRAINT CHK\_PHONE\_NUMBER CHECK (PHONE\_NUMBER LIKE '+359%')

EMPLOYEE TEAMS:

ALTER TABLE EMPLOYEE\_TEAMS ADD CONSTRAINT CHK\_TEAM\_SIZE CHECK (TEAM\_SIZE >= 2 AND TEAM\_SIZE <= 5)

SHIFTS:

ALTER TABLE SHIFTS ADD CONSTRAINT CHK\_TYPE CHECK (TYPE LIKE 'DAY' OR TYPE LIKE 'NIGHT')

ALTER TABLE Shifts ADD CONSTRAINT CHK\_TIME CHECK( Date\_When\_Starts <= Date\_When\_Ends)

ALTER TABLE SHIFTS ADD CONSTRAINT CHK\_IS\_HOLIDAY CHECK (ISHOLIDAY LIKE '1' OR ISHOLIDAY LIKE '0')

ALTER TABLE SHIFTS ADD CONSTRAINT CHK\_IS\_WEEKEND CHECK (ISWEEKEND LIKE '1' OR ISWEEKEND LIKE '0')

-- EXPERIENCE:

ALTER TABLE EXPERIENCE ADD CONSTRAINT CHK\_YEARS CHECK (YEARS >= 0)

ALTER TABLE EXPERIENCE ADD CONSTRAINT CHK\_NUMBER\_OF\_COMPANIES CHECK (NUMBER\_OF\_COMPANIES >= 0)

CUSTOMERS:

ALTER TABLE CUSTOMERS ADD CONSTRAINT CHK\_PHONE\_NUMBER\_CUSTOMER CHECK (PHONE\_NUMBER LIKE '+359%')

ALTER TABLE CUSTOMERS ADD CONSTRAINT CHK\_GENDER\_CUSTOMER CHECK (GENDER LIKE 'MALE' OR GENDER LIKE 'FEMALE')

SECURED\_AREAS :

ALTER TABLE SECURED\_AREAS ADD CONSTRAINT CHK\_SECURITY\_LEVEL CHECK (SECURITY\_LEVEL >= 1 AND SECURITY\_LEVEL <= 10)

ALTER TABLE SECURED\_AREAS ADD CONSTRAINT CHK\_BUILDING\_LEVEL CHECK (BUILDING\_LEVEL >= 0)

ALTER TABLE SECURED\_AREAS ADD CONSTRAINT CHK\_SIZE CHECK (SIZE >= 0)

SENSORS\_FAMILIES :

ALTER TABLE SENSORS\_FAMILIES ADD CONSTRAINT CHK\_SIZE\_SENSORS CHECK (SIZE >= 0)

SENSORS\_DETECTORS :

ALTER TABLE SENSOR\_DETECTORS ADD CONSTRAINT CHK\_RANGE CHECK (RANGE >= 0)

**Индекси:**

CREATE INDEX idx\_Secured\_Areas\_address ON Secured\_Areas (Address)

CREATE INDEX idx\_Customers\_phone\_number ON Customers (Phone\_Number)

CREATE INDEX idx\_Experience\_Company\_Name ON Experience (Company\_Name)

CREATE INDEX idx\_IsDrivable ON Vehicles(isDrivable)

CREATE INDEX idx\_Date\_When\_Starts ON Shifts(Date\_When\_Starts,Date\_When\_Ends)

CREATE INDEX idx\_Triggered\_Date ON Signals(Triggered\_Date)

**Тригери:**

1.

CREATE TRIGGER Employees\_Update\_History\_trigger

ON Employees

AFTER UPDATE

AS

DECLARE @login\_name VARCHAR(128)

SELECT @login\_name = login\_name

FROM sys.dm\_exec\_sessions

WHERE session\_id = @@SPID

BEGIN

INSERT INTO Employees\_Update\_History

SELECT ID,First\_Name,Last\_name,Gender,Age,Address,Phone\_Number, @login\_name, GETDATE()

FROM deleted

END;

2.

CREATE TRIGGER Employees\_Fired\_trigger

ON Employees

AFTER DELETE

AS

BEGIN

INSERT INTO Employees\_History

SELECT ID, First\_Name,Last\_name,Gender,Age,Address,Phone\_Number, GETDATE()

FROM deleted

END;