**Data base- Final project**

**Project subject:**

Security company for civil purposes

**Date:**

27.05.2021

**Submit:**

Lital seyag , 313295412

Tom mandel , 205633688

Ido keren , 205614084

System requirement:

As part of the process of managing security companies for civilian purposes in Israel, it was decided to establish a new database.

The new database helps manage all the company's resources, services the company provides, customer base and types of employees.

The system contributes to choosing the ultimate security company that provides the best services.

General description:

Security company T.I.L has a unique ID, name, address and date of establishment.

A security company offers many types of services:

Patrol, personal security, escort and cyber security.

Each product has an ID, a product name.

The patrol service is accompanied by a vehicle identified with a unique number, patrol id, model, date of manufacture and quantity of fuel.

The security company employs workers in a different style:

Officer, Director, Bodyguard and NOC.

Each employee has a unique identity card, years of experience and a start date.

Some employees carry a weapon that is identified by a unique ID and type.

The company has a large variety of customers. Each customer has an identity card, name and address.

customer orders are managed according to the customer id and the job id. Each line in the order is represented by ID number, order id, product name, quantity, start and end date.

There is a NOC that is identified by noc name, shifts and shift time.

The NOC is control by security cameras characterized by a unique identity, manufacturing and model.

System users:

**Customer** - See the types of service that each security company offers and place a work order.

**Employees-** see the information about customers and security companies.

Description of entities:

fK

PK

**Security company**- id, name, address and establishment date.

**Customer**- id, name and address, company\_ID.

**Employee**- id,name, type, years of experience and start working date, company\_ID.

**Weapon**- id and type, employee\_ID.

**Product**- id ,type, Line\_ID.

**Vehicles-** vehicles id, product id, vehicles model, patrol vehicles production date and amount of fuel , employee\_ID.

**Order-** id, customer id ,cust\_ID.

**Line**- id, product name, quantity, start and end date, Order\_ID.

**NOC-** name, address, company\_ID.

**Security camera-** id, manufacturer and model, customer\_ID, NOC\_name.

Emp\_NOC- emp\_ID, NOC\_Name, Shift\_Date, Start\_Time, End\_Time, emp\_ID, NOC\_Name.

Creation-

create schema Security\_Company;

use Security\_Company;

Table 1:

create table Security\_Company

(

Company\_ID int not null,

SC\_Name char(25),

SC\_Address char(30),

EST\_Date date,

primary key(Company\_ID)

)engine = innoDB;

Table 2:

create table Employees

(

emp\_ID int not null,

emp\_Name char(25),

company\_ID int,

emp\_Type char(25) not null,

Years\_Of\_Exp int,

Start\_Working\_Date date,

primary key(emp\_ID),

constraint fk\_emp\_company foreign key(company\_ID) references security\_Company(company\_ID)

on update cascade

on delete no action

)engine = innoDB;

Table 3:

create table Weapon

(

weap\_ID int not null,

emp\_ID int not null,

Weap\_Type char(25),

primary key(weap\_ID),

constraint fk\_Weapon foreign key(emp\_ID) references Employees(emp\_ID)

on update cascade

on delete no action

)engine = innoDB;

Table 4:

create table Customer

(

Cust\_ID int not null,

cust\_Name char(25),

cust\_Address char(30),

company\_ID int,

primary key(Cust\_ID),

constraint fk\_cust\_company foreign key(company\_ID) references security\_Company(company\_ID)

on update cascade

on delete no action

)engine = innoDB;

Table 5:

create table NOC

(

NOC\_Name char(25) not null,

NOC\_Address char(30),

company\_ID int not null,

primary key(NOC\_Name),

CONSTRAINT fk\_NOC FOREIGN KEY (company\_ID) REFERENCES security\_Company(company\_ID)

)engine = innoDB;

Table 6:

create table Emp\_NOC

(

emp\_ID int not null,

NOC\_Name char(25) not null,

Shift\_Date date not null,

Start\_Time time,

End\_Time time,

primary key (emp\_ID,NOC\_Name),

constraint fk\_engineer\_noc foreign key (emp\_ID) references Employees(emp\_ID),

constraint fk\_noc\_engineer foreign key (NOC\_Name) references NOC(NOC\_Name)

on update cascade

on delete no action

)engine = InnoDB;

Table 7:

create table Security\_Camera

(

cam\_ID int not null,

cam\_manufacturer char(25),

cam\_model char(25),

NOC\_Name char(25),

Cust\_ID int not null,

primary key(cam\_ID),

constraint fk\_noc\_camera foreign key(NOC\_Name) references NOC(NOC\_Name),

constraint fk\_camera foreign key(Cust\_ID) references Customer(Cust\_ID)

on update cascade

on delete no action

)engine = innoDB;

Table 8:

create table Orders

(

order\_ID int not null,

cust\_ID int ,

primary key(order\_ID),

constraint fk\_cust\_order foreign key(cust\_ID) references customer(cust\_ID)

on update cascade

on delete no action

)engine = innoDB;

Table 9:

create table Products

(

Product\_ID int not null,

Product\_Type char(25) not null,

primary key(Product\_ID)

)engine = innoDB;

Table 10:

create table Line

(

Line\_ID int not null,

Product\_ID int not null,

Quantity int,

Start\_Date date,

End\_Date date,

order\_ID int not null,

primary key(Line\_ID),

constraint fk\_Line\_order foreign key(order\_ID) references Orders(order\_ID),

constraint fk\_products\_line foreign key(Product\_ID) references Products(Product\_ID)

on update cascade

on delete no action

)engine = innoDB;

Table 11:

create table Emp\_Products

(

emp\_ID int not null,

Product\_ID int not null,

primary key (emp\_ID,Product\_ID),

constraint fk\_Emp\_Products foreign key (emp\_ID) references Employees(emp\_ID),

constraint fk\_Product\_ID foreign key (Product\_ID) references Products(Product\_ID)

on update cascade

on delete no action

)engine = InnoDB;

Table 12:

create table Vehicle

(

V\_ID int not null,

Product\_ID int,

V\_Prod\_Year int,

V\_Model char(25),

Fuel\_Amount int,

primary key (V\_ID),

constraint fk\_patrol\_pv foreign key (Product\_ID) references Products(Product\_ID)

on update cascade

on delete no action

)engine = InnoDB;

Table 13:

create table V\_Employees

(

emp\_ID int not null,

V\_ID int not null,

Shift\_Date date not null,

Start\_Time time,

End\_Time time,

primary key (emp\_ID,V\_ID),

constraint fk\_emp\_ID foreign key (emp\_ID) references Employees(emp\_ID),

constraint fk\_V\_ID foreign key (V\_ID) references Vehicle(V\_ID)

on update cascade

on delete no action

)engine = InnoDB;

Insertion-

use security\_company;

**Table security\_Company:**

INSERT INTO security\_Company (company\_ID, sc\_Name, sc\_Address ,EST\_Date) VALUES (1, 'TIL', 'Tel aviv','1994-04-07');

**Table Employees:**

(11,'Alex', 1,'manager',27,'1994-06-26'),

(12,'David', 1,'manager',27,'1994-05-07'),

(13,'Roni', 1,'manager',26,'1995-12-07'),

(14,'Mila',1,'officer',3,'2018-07-26'),

(15,'Boni',1,'officer',25,'1996-03-07'),

(16,'Gal',1,'officer',26,'1995-06-07'),

(17,'Shlomo',1,'officer',24,'1997-06-26'),

(18,'Moshe',1,'bodyguard',21,'2000-04-07'),

(19,'Ami',1,'bodyguard',12,'2009-04-09'),

(20,'Hezi',1,'bodyguard',1,'2020-06-26'),

(21,'Menahem',1,'bodyguard',4,'2017-04-25'),

(22,'Haim',1,'bodyguard',2,'2019-04-07'),

(23,'Ofer',1,'bodyguard',0,'2021-05-07'),

(24,'Itzik',1,'bodyguard',7,'2014-04-20'),

(25,'Aharon',1,'NOC\_engineer',9,'2012-06-07'),

(26,'Arie',1,'NOC\_engineer',9,'2012-09-23'),

(27,'Daniel',1,'NOC\_engineer',10,'2011-04-07'),

(28,'Ran',1,'NOC\_engineer',11,'2010-03-18'),

(29,'Lotem',1,'NOC\_engineer',9,'2012-04-07'),

(30,'Lin',1,'NOC\_engineer',5,'2019-08-01');

**Table weapon:**

INSERT INTO weapon (weap\_ID,emp\_ID, Weap\_Type ) VALUES

(61,14,'Glock'),

(62,15,'Knife'),

(63,16,'Barita'),

(64,17,'Glock'),

(65,18,'Glock'),

(66,19,'M16'),

(67,20, 'M16),

(68,21, 'M16),

(69,22, 'Barita'),

(70,23, 'Glock'),

(71,24, 'Glock'),

**Table customer:**

INSERT INTO customer (cust\_ID,cust\_Name, cust\_Address,company\_ID) VALUES

('71','momo','ashdod',1),

('72','koko','rehovot',1),

('3','tiltil','Tel-Aviv',1),

('6','Ariel','yafo',1),

('10','Aviv','Haifa',1),

('2','Nir','Lod',1),

('57','Tal','Eilat',1),

('68','Tal','Eilat',1),

('74','Toyo','rishon lezion',1);

**Table NOC:**

INSERT INTO NOC (NOC\_name ,NOC\_Address,company\_ID) VALUES

('TILNOC','Yigal Alon 68, Tel Aviv',1);

**Table Emp\_NOC:**

INSERT INTO Emp\_NOC(emp\_ID, NOC\_Name ,Shift\_Date ,Start\_Time ,End\_Time) VALUES

(25,'TILNOC','2019-08-01','07:00:00','15:00:00'),

(26,'TILNOC','2020-11-09','07:00:00','15:00:00'),

(27,'TILNOC','2018-08-01','23:00:00','07:00:00'),

(28,'TILNOC','2107-12-18','15:00:00','23:00:00'),

(29,'TILNOC','2019-08-01','07:00:00','15:00:00'),

(30,'TILNOC','2019-08-01','15:00:00','23:00:00');

**Table security\_camera:**

INSERT INTO security\_camera (cam\_ID, cam\_manufacturer, Cust\_ID, NOC\_Name ,cam\_model) VALUES

('81', 'sony', 71,'TILNOC','ABC'),

('82', 'Cannon', 72,'TILNOC','DEF'),

('56', 'Nividia', 6,'TILNOC','DE3'),

('57', 'sony', 3,'TILNOC','DEI'),

('58', 'Nividia', 10,'TILNOC','EFI'),

('59', 'LG', 2,'TILNOC','D10I'),

('60', 'Samsung', 2,'TILNOC','D20I'),

('83', 'Nividia', 6,'TILNOC','ABC'),

('100', 'Cannon', 74,'TILNOC','DEFL');

**Table Orders:**

INSERT INTO Orders (order\_ID, cust\_ID) VALUES

(123,71),

(125,71),

(126,3),

(127,2),

(128,6),

(129,10),

(130,10),

(131,3),

(124,74);

**Table Products:**

INSERT INTO Products (Product\_ID,Product\_Type) VALUES

(120,'Escort'),

(121,'Personal Protection'),

(122,'Patrol'),

(123,'Cyber Security');

**Table Vehicle:**

INSERT INTO Vehicle (V\_ID ,Product\_ID ,V\_Prod\_Year ,V\_Model ,Fuel\_Amount) VALUES

(701, 120, '1995', 'M4', 30 ),

(703, 122, '2020', 'E4', 45),

(704, 120, '2021', 'Q4', 158),

(705, 120, '1980', 'E4', 230),

(706, 122, '1996', 'M4', 90),

(707, 122, '2019', 'B6', 115),

(708, 122, '2018', 'C9', 96);

INSERT INTO V\_Employees(emp\_ID, V\_ID ,Shift\_Date ,Start\_Time ,End\_Time) VALUES

(15,'707','2019-08-01','07:00:00','15:00:00'),

(15,'705','2019-08-05',20:30:00','04:30:00'),

(15,'701','2019-09-07',22:00:00','06:00:00'),

(16,'701','2020-06-08,'07:00:00','15:00:00'),

(17,'704','2021-01-18,16:00:00','21:00:00'),

(16,'708',2020-06-10','15:00:00','23:00:00'),

**Table Line:**

INSERT INTO Line (Line\_ID, Product\_ID ,Quantity,Start\_Date ,End\_Date ,order\_ID) VALUES

(1,120,3,'2019-08-01','2020-08-01',123),

(2,120,3,'2019-08-01','2020-08-01',123),

(3,120,3,'2019-08-01','2020-08-01',123),

(4,121,3,'2019-08-01','2020-08-01',124),

(5,122,3,'2019-08-01','2020-08-01',126),

(6,123,3,'2019-08-01','2020-08-01',131);

Oueries:

**Customer:**

1. Get the customer id of all the customer in the company by address.

solution

select cust\_ID

from customer

order by cust\_Address;

2. Get the customer name that starts with the letter 'A'.

solution

select cust\_Name

from customer

where cust\_Name like 'A%';

3. Get the customer id and name that the camera id is between 80-83 and the customer id is smaller than 71.

solution

select customer.cust\_Name ,customer.cust\_ID, security\_camera.cam\_ID

from customer, security\_camera

where customer.cust\_ID = security\_camera.cust\_ID

and cam\_ID between 80 and 83 and security\_camera.cust\_ID > 71;

4. Get the customer id and name and order id that the customer order in bigger than 1 and the customer name is like aviv and momo.

Solution

select customer.cust\_ID,customer.cust\_Name, count(order\_ID)

from customer, Orders

where Orders.cust\_ID = customer.cust\_ID

and order\_ID in (select order\_ID from Orders, customer where Orders.cust\_ID = customer.cust\_ID and

(cust\_Name like ('Aviv') or cust\_Name like (‘Momo')))

group by cust\_ID, cust\_Name

having count(order\_ID)>1

order by cust\_Name;

5. Get the customer name and order id and company that the customer order product.

Solution

Select customer.cust\_Name, Orders.order\_ID,

security\_company.sc\_Name

from customer,Orders,security\_company

where customer.cust\_ID = Orders.cust\_ID

group by Orders.cust\_ID

order by Orders.order\_ID;

6. Get customer name and camera id that the camera id id equal to customer id and the camera id is between 50-70.

Solution

select customer.cust\_Name, Security\_Camera.cam\_ID

from customer, Security\_Camera

where customer.Cust\_ID = Security\_Camera.Cust\_ID

and (Security\_Camera.cam\_ID) in (select cam\_ID from Security\_Camera where cam\_ID between 50 and 70);

7. Get camera id and manufacturer by sony that the customer named 'Nir' has.

Solution

select distinct Security\_Camera.cam\_ID as ID,Security\_Camera.cam\_manufacturer as Manufacturerfrom Security\_Camerawhere cam\_ID in (select cam\_ID from Security\_Camera, customer where cust\_Name = 'Nir' and customer.Cust\_ID = Security\_Camera.Cust\_ID)or cam\_ID in (select cam\_ID from Security\_Camera, customer where cam\_manufacturer = 'Sony' and customer.Cust\_ID = Security\_Camera.Cust\_ID)

order by cam\_manufacturer;

**Employees:**

1. Get employees data.

Solution

SELECT Employees.\*

FROM Employees

ORDER BY DATEDIFF(Employees.Start\_Working\_Date, CURDATE()) DESC;

2. Get the employee id from the employees that are officers.

Solution

SELECT emp\_ID

FROM Employees

WHERE emp\_Type = 'Officer';

3. Get the details of the employees that worked for more than 10 years.

Solution

SELECT Employees.\*

FROM Employees

WHERE YEAR(CURDATE()) - YEAR(Start\_Working\_Date) >= '10'ORDER BY Start\_Working\_Date ASC;

4. Get the employees details and weapon id and name from employees and weapon table and show who doesn't owns a weapon.

Solution

SELECT Employees.emp\_ID, Employees.emp\_Name, Employees.emp\_Type, Weapon.weap\_ID, Weapon.Weap\_Type

FROM Employees

LEFT OUTER JOIN Weapon ON Employees.emp\_ID = Weapon.emp\_IDWHERE Weapon.weap\_ID IS NULL;

5. Show how many of weapon type are the same held by the employees.

Solution

SELECT Employees.emp\_Type, Weapon.Weap\_Type, COUNT(Weapon.Weap\_Type) AS 'number\_Of\_weapon’

FROM Employees, Weapon

WHERE Employees.emp\_ID = Weapon.emp\_IDGROUP BY Weapon.Weap\_Type ,Employees.emp\_TypeORDER BY COUNT(Weapon.Weap\_Type) DESC;

6. Get the details of the "employee of the month" who worked the biggest amount of hours and show that amount of hours.

Solution

SELECT Employees.emp\_id ,Employees.emp\_Name

SUM(CASE

WHEN (HOUR(emp\_NOC.end\_time) - HOUR(emp\_NOC.start\_time)) > 0 THEN HOUR(emp\_NOC.end\_time) - HOUR(emp\_NOC.start\_time)

ELSE (24 - HOUR(emp\_NOC.start\_time) + HOUR(emp\_NOC.end\_time))

END) AS shift\_duration\_sum,

employees.emp\_type

FROM Employees

JOIN emp\_NOC ON Employees.emp\_id = emp\_NOC.emp\_id

GROUP BY emp\_NOC.emp\_id , Employees.emp\_name

UNION SELECT

Employees.emp\_id ,Employees.emp\_Name

SUM(CASE

WHEN (HOUR(v\_employees.end\_time) - HOUR(v\_employees.start\_time)) > 0 THEN HOUR(v\_employees.end\_time) - HOUR(v\_employees.start\_time)

ELSE (24 - HOUR(v\_employees.start\_time) + HOUR(v\_employees.end\_time))

END) AS shift\_duration\_sum,

employees.emp\_type

FROM Employees

JOIN

v\_employees ON Employees.emp\_id = v\_employees.emp\_id

GROUP BY v\_employees.emp\_id , Employees.emp\_name

ORDER BY shift\_duration\_sum DESC;

7. Calculate the amount of fuel per hour to a vehicle.

Solution

SELECT employees.emp\_id, employees.emp\_name, vehicle.v\_id, vehicle.v\_prod\_year, fuel\_amount / (CASE

AS avg\_fuel\_amount\_pWHEN (HOUR(v\_employees.end\_time) - HOUR(v\_employees.start\_time)) > 0 THEN HOUR(v\_employees.end\_time) - HOUR(v\_employees.start\_time) ELSE (24 - HOUR(v\_employees.start\_time) + HOUR(v\_employees.end\_time)) END)

er\_employee\_vhicle

FROM employees

JOIN v\_employees ON employees.emp\_id = v\_employees.emp\_id JOIN vehicle ON vehicle.v\_id = v\_employees.v\_idGROUP BY emp\_id , emp\_name , v\_id , v\_prod\_yearORDER BY avg\_fuel\_amount\_per\_employee\_vhicle DESC;

Insert/Update/Delete:

use security\_company;

create table Customer

(

Cust\_ID int not null,

cust\_Name char(25),

cust\_Address char(30),

company\_ID int,

primary key(Cust\_ID),

constraint fk\_cust\_company foreign key(company\_ID) references security\_Company(company\_ID)

on update cascade

on delete no action

engine = innoDB;(

create table customer\_log

(

Cust\_ID int not null,

old\_cust\_Name char(25),

new\_cust\_Name char(25

old\_cust\_Address char(30),

new\_cust\_Address char(30),

old\_company\_ID int,

new\_company\_ID int,

command\_ts timestamp,

command varchar(10));)

Triger 1:

delimiter $

CREATE TRIGGER Customer\_customer\_del\_trg AFTER DELETE ON customer

FOR EACH ROW

BEGIN

INSERT INTO customer\_log VALUES(old.Cust\_ID, old.cust\_Name, null, old.cust\_Address,null, old.company\_ID, null, now(), 'delete');

END$

Delimiter;

Triger 2:

Delimiter $

CREATE TRIGGER Customer\_customer\_upd\_trg AFTER UPDATE ON customer

FOR EACH ROW

BEGIN

INSERT INTO customer\_log VALUES(new.Cust\_ID, old.cust\_Name, new.cust\_Name, old.cust\_Address, new.cust\_Address, old.company\_ID , new.company\_ID, now(), 'update');

END$

Delimiter;