

EVILCORP



Lluc Galceran Jodar
28/05/2023

Índice

1- Introducción.....	3
2- Health Care.....	4
2.1- Diagrama Health Care.....	4
2.2- Explicación de las Tablas:.....	5
2.2.1- Users:.....	5
2.2.2- Medicines:.....	5
2.2.3- Diagnoses:.....	6
2.2.4- Doctors:.....	6
2.2.5- Treatments:.....	7
2.2.6- Conditions.....	7

1- Introducción

Un Evil Corp es una empresa ficticia o real que se dedica a actividades ilegales, muy poco éticas las cuales solo van en busca de dinero y poder.

Estas empresas a menudo tienen objetivos nefastos y persiguen sus propios intereses sin tener en cuenta las consecuencias para la sociedad o el medio ambiente.

Los objetivos de una evil corp suelen estar centrados en obtener el control y la dominación en diferentes ámbitos. Pueden buscar el monopolio en un sector determinado, manipular mercados para maximizar sus ganancias. En nuestro caso nuestra Evil Corp está separada por 3 partes, la parte de las conspiraciones (**WORLD**), en la parte de la salud (**HEALTH CARE**) y en la parte de las viviendas (**REAL STATE**).

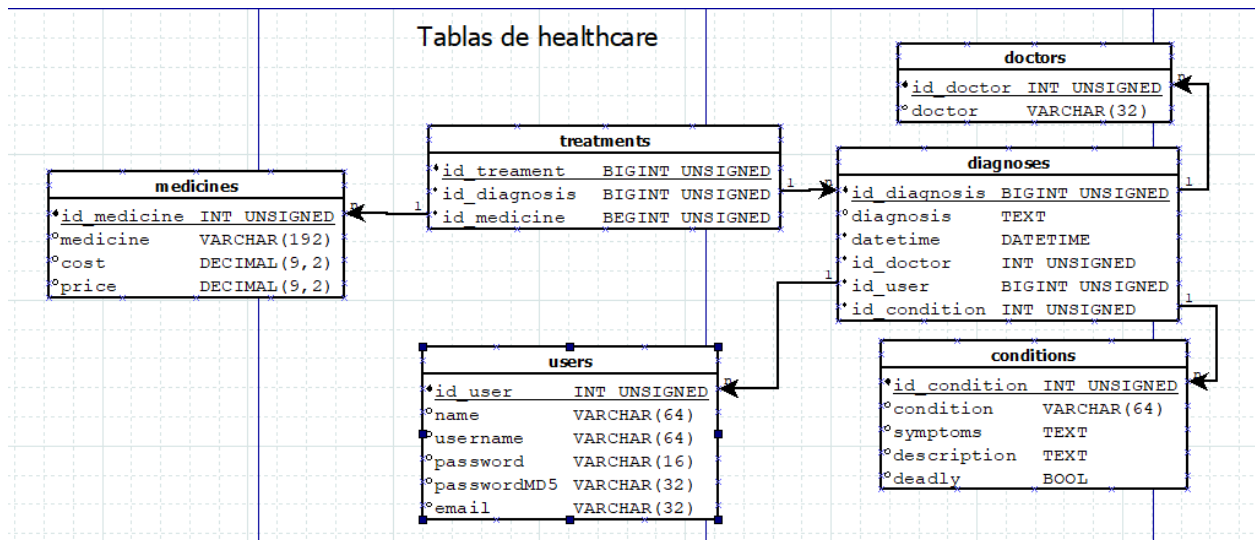
Estas corporaciones suelen utilizar tácticas despiadadas para lograr sus objetivos, como el soborno, la corrupción o el chantaje. También pueden emplear la tecnología y la innovación para desarrollar herramientas o productos peligrosos que pongan en riesgo la privacidad, la seguridad o la salud de las personas como en nuestro caso.



2- Health Care

En este apartado veremos la parte de la salud, veremos varias tablas con las medicinas los diagnósticos, los tratamientos entre otras, también veremos lo que de verdad le interesa a nuestra empresa que es el dinero que saca por cada medicina que vende.

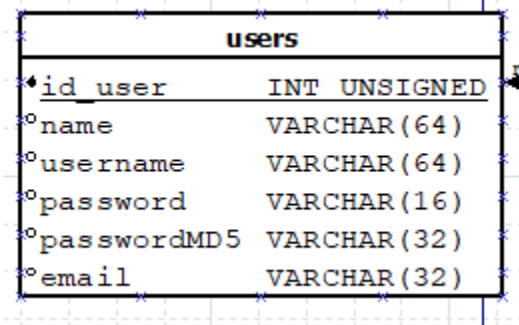
2.1- Diagrama Health Care



2.2- Explicación de las Tablas Health Care:

2.2.1- Users:

Esta tabla es la tabla en la que se basa toda la información de esta base de datos podríamos decir que es la tabla más importante de todas, esta tabla no es exactamente de la parte de Health Care, pero como es el primero el cual voy a explicar la he puesto aquí.



users	
•id_user	INT UNSIGNED
°name	VARCHAR(64)
°username	VARCHAR(64)
°password	VARCHAR(16)
°passwordMD5	VARCHAR(32)
°email	VARCHAR(32)

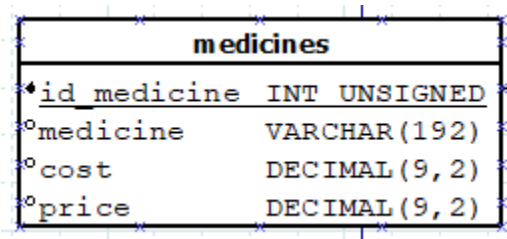
2.2.2- Medicines:

Esta tabla almacena las medicinas que Evil Corp tiene disponible, esta tiene los siguientes campos:

Medicine: Este campo es un **VARCHAR(192)** porque es el campo que almacena el nombre de la medicina y como ya sabemos hay medicinas con nombres muy largos de hecho la medicina con el nombre más largo tiene 192 caracteres.

Cost: Este campo es **Decimal (9,2)**, ya que representa el coste de los medicamentos para Evil Corp, él (9,2) significa que la columna "cost" puede almacenar números con hasta 9 dígitos en total, y de esos 9 dígitos, 2 se reservan para la parte decimal.

Price: Este campo es **Decimal (9,2)**, ya que representa el precio al cual Evil Corp vende estos productos al mercado, él (9,2) significa que la columna "cost" puede almacenar números con hasta 9 dígitos en total, y de esos 9 dígitos, 2 se reservan para la parte decimal.



medicines	
•id_medicine	INT UNSIGNED
°medicine	VARCHAR(192)
°cost	DECIMAL(9,2)
°price	DECIMAL(9,2)

2.2.3- Diagnoses:

Esta tabla es donde se guardan los datos de una diagnóstico completo el cual tiene estos campos:

Diagnosis: Esta es tipo **TEXT**, ya que es una pequeña explicación del Diagnóstico completo i requiere de demasiados caracteres para ser un **CHAR** o **VARCHAR**

Id_user: Este campo hace referencia al identificador de la tabla **USERS** mediante una **FOREIGN KEY**, sirve para poder vincular un usuario a una diagnóstico en concreto.

Id_doctor: Este campo hace referencia al identificador de la tabla **DOCTORS** mediante una **FOREIGN KEY**, nos sirve para poder vincular un doctor a una diagnóstico en concreto.

Id_condition: Este campo hace referencia al identificador de la tabla **CONDITIONS** mediante una **FOREIGN KEY**, nos sirve para poder vincular una condición a un diagnostico concreto.

diagnoses	
*id_diagnosis	BIGINT UNSIGNED
°diagnosis	TEXT
*datetime	DATETIME
*id_doctor	INT UNSIGNED
*id_user	BIGINT UNSIGNED
*id_condition	INT UNSIGNED

2.2.4- Doctors:

En esta tabla se almacenan los nombres de todos los doctores que están dentro de nuestra Evil corp, esta tabla tiene solo este campo:

Doctor: Este campo es **VARCHAR(32)** este indica el nombre del doctor el cual tenemos.

doctors	
*id_doctor	INT UNSIGNED
°doctor	VARCHAR(32)

2.2.5- Treatments:

En esta tabla se almacenan las vinculaciones entre los **medicamentos y los diagnósticos**, podemos decir que si juntamos el medicamento más un diagnóstico tenemos un **tratamiento**.

Id_diagnosis: Este campo hace referencia al identificador de la tabla **DIAGNOSES** mediante una **FOREIGN KEY**, nos sirve para poder vincular un diagnóstico a un tratamiento específico.

Id_medicine: Este campo hace referencia al identificador de la tabla **MEDICINES** mediante una **FOREIGN KEY**, esto nos permite poder vincular una medicina a un tratamiento específico.

treatments	
*id_treatment	BIGINT UNSIGNED
*id_diagnosis	BIGINT UNSIGNED
*id_medicine	BIGINT UNSIGNED

2.2.6- Conditions

Esta tabla es donde se almacenan todas las distintas enfermedades que puede haber o las más importantes para que Evil Corp saque su dinero.

Condición: Este campo hace referencia al nombre de la enfermedad, por eso es un **VARCHAR(64)**, ya que puede haber enfermedades con nombres bastante largos.

Symptoms: Este campo almacena los síntomas de la enfermedad correspondiente, es de tipo **TEXT**, ya que los síntomas son como una pequeña descripción y con los otros tipos no habría suficiente espacio.

Description: Este campo almacena las descripciones de las enfermedades por esto es tipo **TEXT**.

Deadly: Este campo hace referencia a si una enfermedad es mortal, por esto es un **BOOLEAN**, ya que solo puede ser o **TRUE** o **FALSE**.

conditions	
*id_condition	INT UNSIGNED
°condition	VARCHAR(64)
°symptoms	TEXT
°description	TEXT
°deadly	BOOL

2.3- Script Health Care:

2.3.1- Drops:

```
DROP TABLE IF EXISTS treatments;
DROP TABLE IF EXISTS diagnoses;
DROP TABLE IF EXISTS doctors;
DROP TABLE IF EXISTS conditions;
DROP TABLE IF EXISTS medicines;
```

2.3.2- Creates:

```
CREATE TABLE medicines(
    id_medicine INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    medicine VARCHAR(192) NOT NULL,
    cost DECIMAL(9,2),
    price DECIMAL(9,2)
);

CREATE TABLE conditions(
    id_condition INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    symptom TEXT NOT NULL,
    description TEXT NOT NULL,
    deadly BOOL NOT NULL
);

CREATE TABLE doctors(
    id_doctor INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    doctor VARCHAR(32)
);

CREATE TABLE diagnoses(
    id_diagnosis INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    diagnoses TEXT NOT NULL,
    id_user INT UNSIGNED NOT NULL,
    id_condition INT UNSIGNED NOT NULL,
    id_doctor INT UNSIGNED NOT NULL,
    FOREIGN KEY (id_user) REFERENCES users(id_user),
    FOREIGN KEY (id_condition) REFERENCES conditions(id_condition),
    FOREIGN KEY (id_doctor) REFERENCES doctors(id_doctor)
);
```



```

CREATE TABLE treatments(
    id_treatments INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    id_diagnosis INT UNSIGNED NOT NULL,
    id_medicine INT UNSIGNED NOT NULL,
    FOREIGN KEY (id_treatments) REFERENCES diagnoses(id_diagnosis),
    FOREIGN KEY (id_medicine) REFERENCES medicines(id_medicine)
);

```

2.3.3- Inserts:

```

----- MEDICINES -----
INSERT INTO medicines (medicine, cost, price)
VALUES ('Aspirin', 5.99, 9.99);

INSERT INTO medicines (medicine, cost, price)
VALUES ('Paracetamol', 3.49, 7.99);

INSERT INTO medicines (medicine, cost, price)
VALUES ('Ibuprofen', 4.99, 8.99);

INSERT INTO medicines (medicine, cost, price)
VALUES ('Amoxicillin', 7.99, 12.99);

INSERT INTO medicines (medicine, cost, price)
VALUES ('Omeprazole', 6.49, 10.99);

----- CONDITIONS -----
INSERT INTO conditions (name, symptom, description, deadly)
VALUES ('Flu', 'Fever, cough, sore throat', 'Common viral infection',
FALSE);

INSERT INTO conditions (name, symptom, description, deadly)
VALUES ('Diabetes', 'Increased thirst, frequent urination', 'Chronic
metabolic disease', TRUE);

INSERT INTO conditions (name, symptom, description, deadly)
VALUES ('Hypertension', 'High blood pressure', 'Cardiovascular disorder',
FALSE);

INSERT INTO conditions (name, symptom, description, deadly)

```

```
VALUES ('Asthma', 'Shortness of breath, wheezing', 'Chronic respiratory disease', FALSE);
```

```
INSERT INTO conditions (name, symptom, description, deadly)
VALUES ('Cancer', 'Unexplained weight loss, fatigue', 'Malignant tumor growth', TRUE);
```

```
----- DOCTORS -----
```

```
INSERT INTO doctors (doctor)
VALUES ('Dr. Smith');
```

```
INSERT INTO doctors (doctor)
VALUES ('Dr. Johnson');
```

```
INSERT INTO doctors (doctor)
VALUES ('Dr. Anderson');
```

```
INSERT INTO doctors (doctor)
VALUES ('Dr. Davis');
```

```
INSERT INTO doctors (doctor)
VALUES ('Dr. Wilson');
```

```
----- DIAGNOSES -----
```

```
INSERT INTO diagnoses (diagnoses, id_user, id_condition, id_doctor)
VALUES ('You have the flu. Get plenty of rest and drink fluids.', 1, 1, 1);
```

```
INSERT INTO diagnoses (diagnoses, id_user, id_condition, id_doctor)
VALUES ('You have diabetes. Follow a healthy diet and take medication as prescribed.', 2, 2, 2);
```

```
INSERT INTO diagnoses (diagnoses, id_user, id_condition, id_doctor)
VALUES ('You have hypertension. Monitor your blood pressure regularly.', 3, 3, 3);
```

```
INSERT INTO diagnoses (diagnoses, id_user, id_condition, id_doctor)
VALUES ('You have asthma. Use an inhaler as needed.', 4, 4, 4);
```

```
INSERT INTO diagnoses (diagnoses, id_user, id_condition, id_doctor)
VALUES ('You have cancer. Further tests and treatment options will be discussed.', 5, 5, 5);
```

```
----- TREATMENTS -----
```

```
INSERT INTO treatments (id_diagnosis, id_medicine)
VALUES (1, 1);

INSERT INTO treatments (id_diagnosis, id_medicine)
VALUES (2, 3);

INSERT INTO treatments (id_diagnosis, id_medicine)
VALUES (3, 2);

INSERT INTO treatments (id_diagnosis, id_medicine)
VALUES (4, 5);

INSERT INTO treatments (id_diagnosis, id_medicine)
VALUES (5, 4);
```

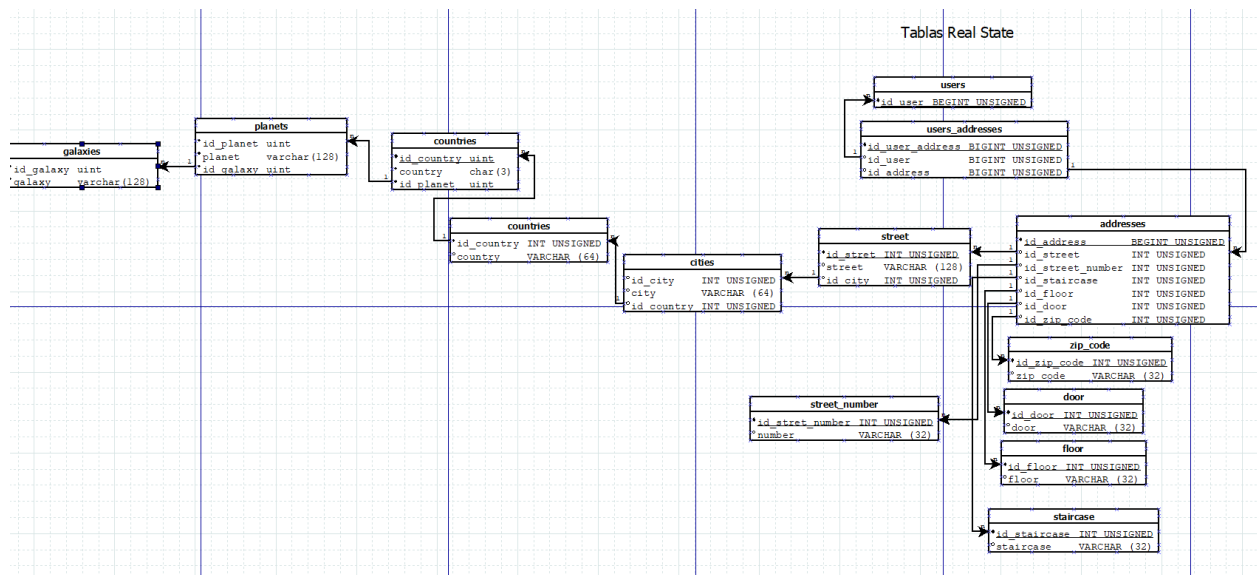
2.3.4- Views:

```
----- VIEW PROFIT MEDICINES -----
create view costes as select treatments.id_medicine,
COUNT(medicine) medicines, cost_production,
SUM(cost_production) total_cost_production, cost_sell,
SUM(cost_sell) total_cost_sell from medicines LEFT JOIN
treatments on medicines.id_medicine=treatments.id_medicine group
by cost_production;
```

3- Real State

En este apartado veremos la parte de la vivienda i de toda la información que tenemos sobre cada usuario, en esta parte tenemos estas tablas: streets, streets numbers, addresses, planets entre otras, este apartado es el más extenso y el más complejo, también veremos aquí las diferentes funciones y procedimientos creados para varios usos.

3.1- Diagrama



3.2- Explicación de las tablas Real State:

3.2.1-Addresses:

Esta tabla es la encargada de almacenar el conjunto de calle, número de calle, piso, puerta y código postal.

Id_street: Este campo hace referencia al identificador de la tabla **STREETS** mediante una **FOREIGN KEY**, esto nos sirve para vincular una calle a una dirección.

Id_street_num: Este campo hace referencia al identificador de la tabla **STREETS_NUM** mediante una **FOREIGN KEY**, esto es porque vinculamos un número calle a una dirección.

Id_staircase: Este campo hace referencia al identificador de la tabla **STAIRCASES** mediante una **FOREIGN KEY**, nos sirve para vincular un número de escalera a una dirección.

Id_floor: Este campo hace referencia al identificador de la tabla **FLOORS** mediante una **FOREIGN KEY**, nos sirve para vincular una planta a una dirección.

Id_door: Este campo hace referencia al identificador de la tabla **DOORS** mediante una **FOREIGN KEY**, esto es porque queremos saber a qué puerta pertenece esta dirección.

Id_zipcode: Este campo hace referencia al identificador de la tabla **ZIPCODES** mediante una **FOREIGN KEY**, esto es porque queremos saber qué código postal pertenece esta dirección.

addresses	
*id_address	BIGINT UNSIGNED
°id_street	INT UNSIGNED
°id_street_number	INT UNSIGNED
°id_staircase	INT UNSIGNED
°id_floor	INT UNSIGNED
°id_door	INT UNSIGNED
°id_zip_code	INT UNSIGNED

3.2.2- User_addresses:

Esta tabla es una tabla intermedia entre la tabla **users** y la tabla **addresses**, nos sirve para poder vincular a un usuario a su dirección correspondiente. Esta tabla está compuesta por:

Id_user: Este campo se refiere al identificador de la tabla **USERS** mediante una **FOREIGN KEY**, y no sirve para saber a qué usuario pertenece una dirección concreta.

Id_address: Este campo se refiere al identificador de la tabla **ADDRESSES** mediante una **FOREIGN KEY**, nos sirve para saber que direcciones pertenecen a cada usuario.

users_addresses	
*id_user_address	BIGINT UNSIGNED
°id_user	BIGINT UNSIGNED
°id address	BIGINT UNSIGNED

3.2.3- Doors:

Esta tabla se encara de almacenar los números de puerta y tienes estos campos:

Door: Este campo es el que almacena el numero de la puerta, es tipo **VARCHAR(8)** porque solo necesitamos almacenar esta cantidad de caracteres.

3.2.4- Doors:

Esta tabla se encara de almacenar los pisos y tienes estos campos:

Floor: Este campo es el que almacena el piso, es tipo **VARCHAR(8)** porque solo necesitamos almacenar números y no superara esta cantidad de caracteres.

3.2.5- Zipcodes:

Esta tabla se encara de almacenar los codigos postal y tienes estos campos:

Zipcode: Este campo es el que almacena el numero de la puerta, es tipo **VARCHAR(16)** porque los codigos postales más largos solo tienen 16 caracteres.

3.2.6- Staircase:

Esta tabla se encara de almacenar los números de escalera y tienes estos campos:

Door: Este campo es el que almacena el numero de la puerta, es tipo **VARCHAR(8)** porque con 8 caracteres tenemos suficientes para almacenar todas las escaleras.

3.2.7- Streets_Numbers:

Esta tabla se encara de almacenar los números de calle y tienes estos campos:

Door: Este campo es el que almacena el numero de la puerta, es tipo **VARCHAR(16)** porque con 16 caracteres tenemos suficientes para almacenar todos los números de calle posibles.