TIPO DI PROGETTO: Compagnia assicurativa veicoli.

Scadenza 2 settembre.

3/8/2022

Raccolta di informazioni per costruire le prime entità da inserire nello schema ER e studio di come funzionano le assicurazioni per automobili.

Le entità necessarie raccolte finora sono: cliente, guidatore, guidatore\_veicolo, codiceinfrazionestradale, registroInfrazioneGuidatore, Auto, Assicurazione, Pagamento, Polizza.

La tabella cliente contiene i dati generici della persona con cui viene stipulata la polizza, guidatore serve a specificare tutti i guidatori di un determinato veicolo (utile per segnare chi ha commesso una infrazione), guidatore\_veicolo è semplicemente la relazione che intendo usare per ricollegare un determinato guidatore ad un veicolo, codiceInfrazioneStradale serve per conservare tutti i tipi di infrazioni che si possono commettere mentre registroInfrazioneGuidatore ci permetterà di ricollegare nello specifico le infrazioni commesse dai vari guidatori, Auto è l’entità per descrivere le varie caratteristiche principali per identificare le vetture, Assicurazione è l’entità che sfrutterò per contenere i vari tipi di assicurazioni disponibili, il pagamento serve per determinare quanto si paga per le specifiche polizze assicurative disponibili ed infine Polizza serve per tenere i vari dati generali che la compongono. Purtroppo, non sono un guidatore quindi al momento ci sono delle cose che non mi sono chiarissime ma nei prossimi giorni intendo riempire ogni lacuna.

4/8/2022

Ho aggiunto degli attributi alle entità e relazioni precedentemente dichiarate: cliente (ID\_Cliente, Nome, Cognome, Codice\_Fiscale), guidatore (ID\_Cliente, Nome, Cognome, Codice\_Fiscale, Data\_Nascita, Cod\_Patente, Indirizzo\_Abitazione), Guidatore\_veicolo (ID\_Cliente, ID\_Auto), CodiceInfrazioniStradali (ID\_Infrazione, Descrizione\_infrazione), RegistroInfrazioniGuidatore (ID\_Infrazione, ID\_Cliente), Auto (ID\_Auto, Targa, Immatricolazione, Kilometraggio, Marca, Modello), assicurazione (ID\_assicurazione, Massimale), Polizza (NumeroPolizza, Tipo\_Polizza, DataRilascio, DataScadenzaPolizza). Devo ancora ben capire cosa potrei aggiungere agli attributi Pagamento, Polizza e Assicurazione. Ovviamente ho già pensato come collegare molte di queste entità e presto farò un disegno del diagramma ER.

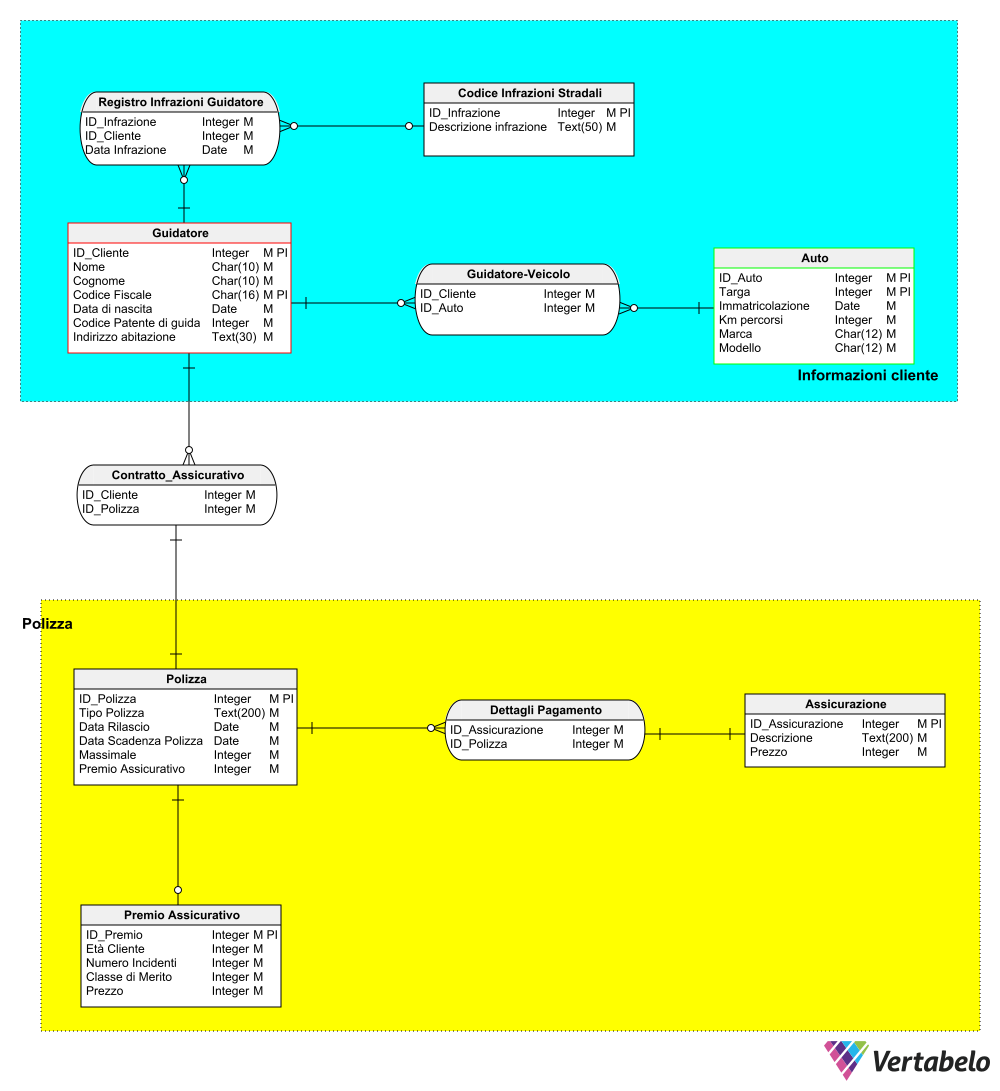
6/8/2022

Oggi costruisco una prima versione primitiva di uno schema ER, ho deciso di utilizzare un sito che si chiama Vertabelo per rendere il processo più semplice e pulito piuttosto che usare carta e penna. Utilizzerò una strategia mista andando successivamente a raffinare i concetti delle relazioni ove necesserio. Ho aggiunto il parametro data infrazione alla relazione Registro Infrazioni Guidatore.

7/8/2022

Sto aggiungendo una nuova entità Premio Assicurativo e la relazione DettagliPagamento. Il premio assicurativo è costituito da un ID, l’età del contraente, il numero di incidenti causato, la classe di merito e il prezzo finale. Rimossa tabella cliente per evitare ridondanza inutile.

11/08/2022



Questa è la prima versione del diagramma ER. Oggi ho aggiunto una nuova relazione, Contratto\_assicurativo, che serve per collegare ogni guidatore ad una o più polizze.

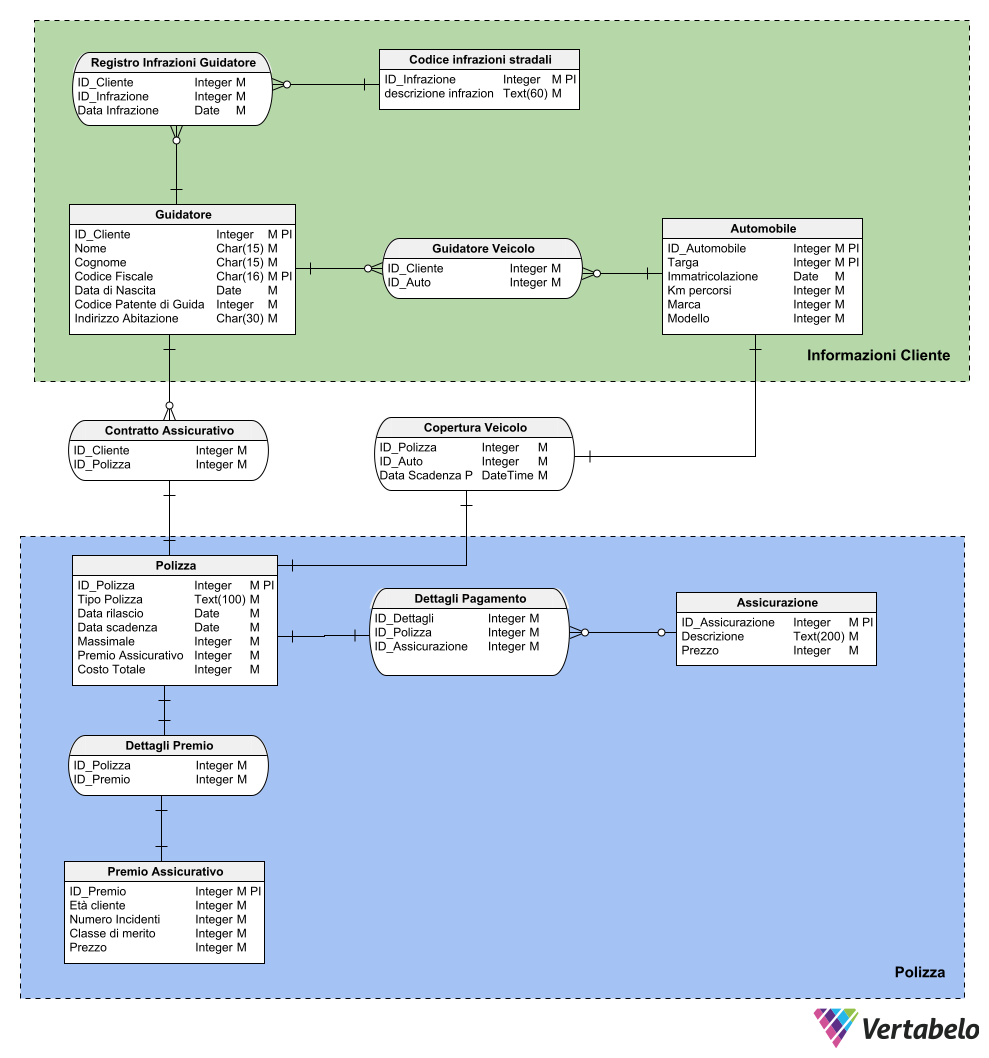
27/08/2022

Nelle ultime due settimane sono stato parecchio male e non ho avuto l’opportunità di lavorare al progetto ma ora sto nettamente meglio.

Ho deciso di eliminare una relazione inutile che avevo aggiunto in modo teorico durante il periodo in cui la febbre determinava le mie scelte poco sagge.

28/08/2022

Sto pensando di aggiungere la relazione Copertura\_Veicolo(ID\_Auto, ID\_Polizza, Data Scadenza Polizza) 1:1, collegata tra Polizza e Auto per gestire più velocemente e facilmente l’assicurazione appartenente ad un determinato veicolo. Oggi mi dedicherò alla conversione verso il modello relazionale. Il seguente è lo schema E-R finale. Ho anche aggiunto Costo Totale a Polizza.



Polizza (ID\_Polizza, Tipo Polizza, Data rilascio, Data scadenza, Massimale, **Premio Assicurativo**, Costo totale);

Premio Assicurativo (ID\_Premio, Età cliente, Numero incidenti, Classe di merito, Prezzo);

Dettagli Pagamento (ID\_Dettagli, **ID\_Polizza**, **ID\_Assicurazione**);

Assicurazione (ID\_Assicurazione, Descrizione, Prezzo);

Per quanto riguarda la relazione Dettagli Premio ho deciso di assimilarla nella trasposizione in schema relazionale con la chiave esterna “Premio assicurativo” in Polizza poiché l’associazione 1:1 ci permette di farlo; dunque, ho scelto di “rimuoverla” per snellire lo schema.

Guidatore (ID\_Cliente, Nome, Cognome, Codice Fiscale, Data di nascita, Codice Patente di guida, Indirizzo abitazione);

Contratto Assicurativo (**ID\_Cliente, ID\_Polizza**);

Guidatore Veicolo (**ID\_Cliente, ID\_Auto**);

Automobile (ID\_Automobile, Targa, data Immatricolazione, km percorsi, Marca, Modello, **Polizza**, Data scadenza Polizza);

Registro Infrazioni Guidatore (**ID\_Cliente, ID\_Infrazione**, **Targa**, Data Infrazione);

Codice Infrazioni Stradali (ID\_Infrazione, descrizione infrazione);

Anche qui ho assimilato Copertura Veicolo dentro all’entità Automobile grazie all’associazione 1:1 che ci permette di fare questa “magia” durante la trasposizione e ho aggiunto Targa alla relazione Registro Infrazioni Guidatore come chiave da Automobile esterna per facilitare il riconoscimento del veicolo con cui è stata commessa l’infrazione.

31/08/2022

Sto costruendo il dizionario delle relazioni su Excel.



È la prima volta che uso Excel, dunque, spero di non aver fatto qualche errore da principiante ma suppongo sia tutto corretto.

Nei prossimi giorni mi metterò al lavoro per scrivere i trigger dinamici della base di dati e creerò effettivamente la base di dati su MySQL.

01/09/2022

Oggi ho iniziato la creazione del database su MySQL Workbench, questo è il codice creato attraverso il forward engineering, devo ancora lavorare sui trigger attivi:

CREATE SCHEMA IF NOT EXISTS `autoassicurazioninadd` DEFAULT CHARACTER SET utf8 ;

USE `autoassicurazioninadd` ;

CREATE TABLE IF NOT EXISTS `autoassicurazioninadd`.`Premio Assicurativo` (

`ID\_Premio Assicurativo` INT NOT NULL AUTO\_INCREMENT,

`Età cliente` INT NOT NULL,

`Numero Incidenti` INT NOT NULL,

`Classe di merito` INT NOT NULL,

`Prezzo` INT NOT NULL,

PRIMARY KEY (`ID\_Premio Assicurativo`)

);

CREATE TABLE IF NOT EXISTS `autoassicurazioninadd`.`Polizza` (

`ID\_Polizza` INT NOT NULL AUTO\_INCREMENT,

`Tipo Polizza` LONGTEXT NOT NULL,

`Data Rilascio` DATE NOT NULL,

`Data Scadenza` DATE NOT NULL,

`Massimale` INT NOT NULL,

`Costo Totale` INT NOT NULL,

`Premio Assicurativo` INT NOT NULL,

PRIMARY KEY (`ID\_Polizza`, `Premio Assicurativo`),

INDEX `fk\_Polizza\_Premio Assicurativo1\_idx` (`Premio Assicurativo` ASC) VISIBLE,

UNIQUE INDEX `ID\_Polizza\_UNIQUE` (`ID\_Polizza` ASC) VISIBLE,

CONSTRAINT `fk\_Polizza\_Premio Assicurativo1`

FOREIGN KEY (`Premio Assicurativo`)

REFERENCES `autoassicurazioninadd`.`Premio Assicurativo` (`ID\_Premio Assicurativo`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

CREATE TABLE IF NOT EXISTS `autoassicurazioninadd`.`Assicurazione` (

`ID\_Assicurazione` INT NOT NULL AUTO\_INCREMENT,

`Descrizione` MEDIUMTEXT NOT NULL,

`Prezzo` INT NOT NULL,

PRIMARY KEY (`ID\_Assicurazione`),

UNIQUE INDEX `ID\_Assicurazione\_UNIQUE` (`ID\_Assicurazione` ASC) VISIBLE);

CREATE TABLE IF NOT EXISTS `autoassicurazioninadd`.`Dettagli Pagamento` (

`ID\_Dettagli Pagamento` INT NOT NULL AUTO\_INCREMENT,

`ID\_Assicurazione` INT NOT NULL,

`ID\_Polizza` INT NOT NULL,

PRIMARY KEY (`ID\_Dettagli Pagamento`, `ID\_Assicurazione`, `ID\_Polizza`),

INDEX `fk\_Dettagli Pagamento\_Assicurazione\_idx` (`ID\_Assicurazione` ASC) VISIBLE,

INDEX `fk\_Dettagli Pagamento\_Polizza1\_idx` (`ID\_Polizza` ASC) VISIBLE,

UNIQUE INDEX `ID\_Dettagli Pagamento\_UNIQUE` (`ID\_Dettagli Pagamento` ASC) VISIBLE,

CONSTRAINT `fk\_Dettagli Pagamento\_Assicurazione`

FOREIGN KEY (`ID\_Assicurazione`)

REFERENCES `autoassicurazioninadd`.`Assicurazione` (`ID\_Assicurazione`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Dettagli Pagamento\_Polizza1`

FOREIGN KEY (`ID\_Polizza`)

REFERENCES `autoassicurazioninadd`.`Polizza` (`ID\_Polizza`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

CREATE TABLE IF NOT EXISTS `autoassicurazioninadd`.`Guidatore` (

`ID\_Cliente` INT NOT NULL AUTO\_INCREMENT,

`Nome` VARCHAR(45) NOT NULL,

`Cognome` VARCHAR(45) NOT NULL,

`Codice Fiscale` VARCHAR(16) NOT NULL,

`Data di nascita` DATE NOT NULL,

`Codice patente di guida` VARCHAR(10) NOT NULL,

`Indirizzo abitazione` VARCHAR(100) NOT NULL,

PRIMARY KEY (`ID\_Cliente`),

UNIQUE INDEX `Codice Fiscale\_UNIQUE` (`Codice Fiscale` ASC) VISIBLE,

UNIQUE INDEX `ID\_Cliente\_UNIQUE` (`ID\_Cliente` ASC) VISIBLE,

UNIQUE INDEX `Codice patente di guida\_UNIQUE` (`Codice patente di guida` ASC) VISIBLE);

CREATE TABLE IF NOT EXISTS `autoassicurazioninadd`.`Contratto Assicurativo` (

`ID\_Cliente` INT NOT NULL,

`ID\_Polizza` INT NOT NULL,

PRIMARY KEY (`ID\_Cliente`, `ID\_Polizza`),

INDEX `ID\_Polizza\_idx` (`ID\_Polizza` ASC) VISIBLE,

CONSTRAINT `ID\_Polizza`

FOREIGN KEY (`ID\_Polizza`)

REFERENCES `autoassicurazioninadd`.`Polizza` (`ID\_Polizza`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `ID\_Cliente`

FOREIGN KEY (`ID\_Cliente`)

REFERENCES `autoassicurazioninadd`.`Guidatore` (`ID\_Cliente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

CREATE TABLE IF NOT EXISTS `autoassicurazioninadd`.`Automobile` (

`ID\_Auto` INT NOT NULL AUTO\_INCREMENT,

`Targa` VARCHAR(7) NOT NULL,

`Data immatricolazione` DATE NULL,

`Km percorsi` INT NULL,

`Marca` VARCHAR(20) NOT NULL,

`Modello` VARCHAR(45) NOT NULL,

PRIMARY KEY (`ID\_Auto`, `Targa`),

UNIQUE INDEX `Targa\_UNIQUE` (`Targa` ASC) VISIBLE,

UNIQUE INDEX `ID\_Auto\_UNIQUE` (`ID\_Auto` ASC) VISIBLE);

CREATE TABLE IF NOT EXISTS `autoassicurazioninadd`.` Guidatore\_Veicolo` (

`ID\_Cliente` INT NOT NULL,

`ID\_Auto` INT NOT NULL,

PRIMARY KEY (`ID\_Cliente`, `ID\_Auto`),

INDEX `ID\_Auto\_idx` (`ID\_Auto` ASC) VISIBLE,

CONSTRAINT `ID\_Cliente`

FOREIGN KEY (`ID\_Cliente`)

REFERENCES `autoassicurazioninadd`.`Guidatore` (`ID\_Cliente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `ID\_Auto`

FOREIGN KEY (`ID\_Auto`)

REFERENCES `autoassicurazioninadd`.`Automobile` (`ID\_Auto`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

CREATE TABLE IF NOT EXISTS `autoassicurazioninadd`.`Codice Infrazioni Stradali` (

`ID\_Infrazione` INT NOT NULL AUTO\_INCREMENT,

`Descrizione infrazione` MEDIUMTEXT NOT NULL,

PRIMARY KEY (`ID\_Infrazione`),

UNIQUE INDEX `ID\_Infrazione\_UNIQUE` (`ID\_Infrazione` ASC) VISIBLE);

CREATE TABLE IF NOT EXISTS `autoassicurazioninadd`.` Registro Infrazioni Guidatore` (

`ID\_Cliente` INT NOT NULL,

`ID\_Infrazione` INT NOT NULL,

`Targa` VARCHAR(7) NOT NULL,

`Data infrazione` DATE NOT NULL,

PRIMARY KEY (`ID\_Cliente`, `ID\_Infrazione`, `Targa`),

INDEX `ID\_Infrazione\_idx` (`ID\_Infrazione` ASC) VISIBLE,

INDEX `Targa\_idx` (`Targa` ASC) VISIBLE,

UNIQUE INDEX `Targa\_UNIQUE` (`Targa` ASC) VISIBLE,

CONSTRAINT `ID\_Cliente`

FOREIGN KEY (`ID\_Cliente`)

REFERENCES `autoassicurazioninadd`.`Guidatore` (`ID\_Cliente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `ID\_Infrazione`

FOREIGN KEY (`ID\_Infrazione`)

REFERENCES `autoassicurazioninadd`.`Codice Infrazioni Stradali` (`ID\_Infrazione`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `Targa`

FOREIGN KEY (`Targa`)

REFERENCES `autoassicurazioninadd`.`Automobile` (`ID\_Auto`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

05/09/2022

Oggi ho definito in linguaggio naturale i trigger e le viste che verranno usati nella base di dati

07/09/2022

Oggi ho riempito di dati le tabelle del database e creato dei trigger:

CREATE TRIGGER StimaPrezzo

AFTER INSERT ON premio\_assicurativo

FOR EACH ROW

update premio\_assicurativo

SET new.Prezzo = Prezzo\*((`Età cliente`/45)\*(`Numero Incidenti`\*1.15))+(`Classe di merito`\*10);

08/09/2022

Ho quasi finito di sistemare tutto ma ho deciso di prendere una soluzione drastica ad un problema eliminando le due tabelle “Registro Infrazioni Guidatore” e “Codice Infrazioni Stradali”; in realtà non servivano ad un granché visto che non influenzano il calcolo finale della polizza, dunque, ho snellito la base di dati.

Ho finito di aggiungere le viste ed i trigger. Questo è il resto del codice che ho scritto:

#vista per trovare i contratti scaduti usando la funzione di MySQL “curdate()” che ritorna la data di oggi

CREATE VIEW ContrattiScaduti AS

SELECT CA1.ID\_Cliente, P1.ID\_Polizza

FROM contratto\_assicurativo as CA1 JOIN polizza as P1

ON CA1.ID\_Polizza=P1.ID\_Polizza

WHERE curdate()>P1.`Data Scadenza`;

CREATE VIEW AutoClientePolizza AS

SELECT ca1.ID\_Polizza, p1.`Costo Totale`, G1.ID\_Cliente, G1.Nome, G1.Cognome, gv1.IDAuto, a1.Targa, a1.Marca, a1.Modello

FROM guidatore as G1 join guidatore\_veicolo as gv1

on G1.ID\_Cliente=gv1.IDCliente

join automobile as a1

on gv1.IDAuto=a1.ID\_Auto

join contratto\_assicurativo as ca1

on ca1.ID\_Cliente=G1.ID\_Cliente

join polizza as p1

on ca1.ID\_Polizza=p1.ID\_Polizza;

#trigger che impedisce di inserire un nuovo utente la cui età compromette la capacità di guidare

DELIMITER $$

CREATE TRIGGER Anzianita

BEFORE INSERT ON premio\_assicurativo

FOR EACH ROW

BEGIN

IF (new.`Età cliente` > '80' )THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = "Età troppo alta!";

END IF;

END$$

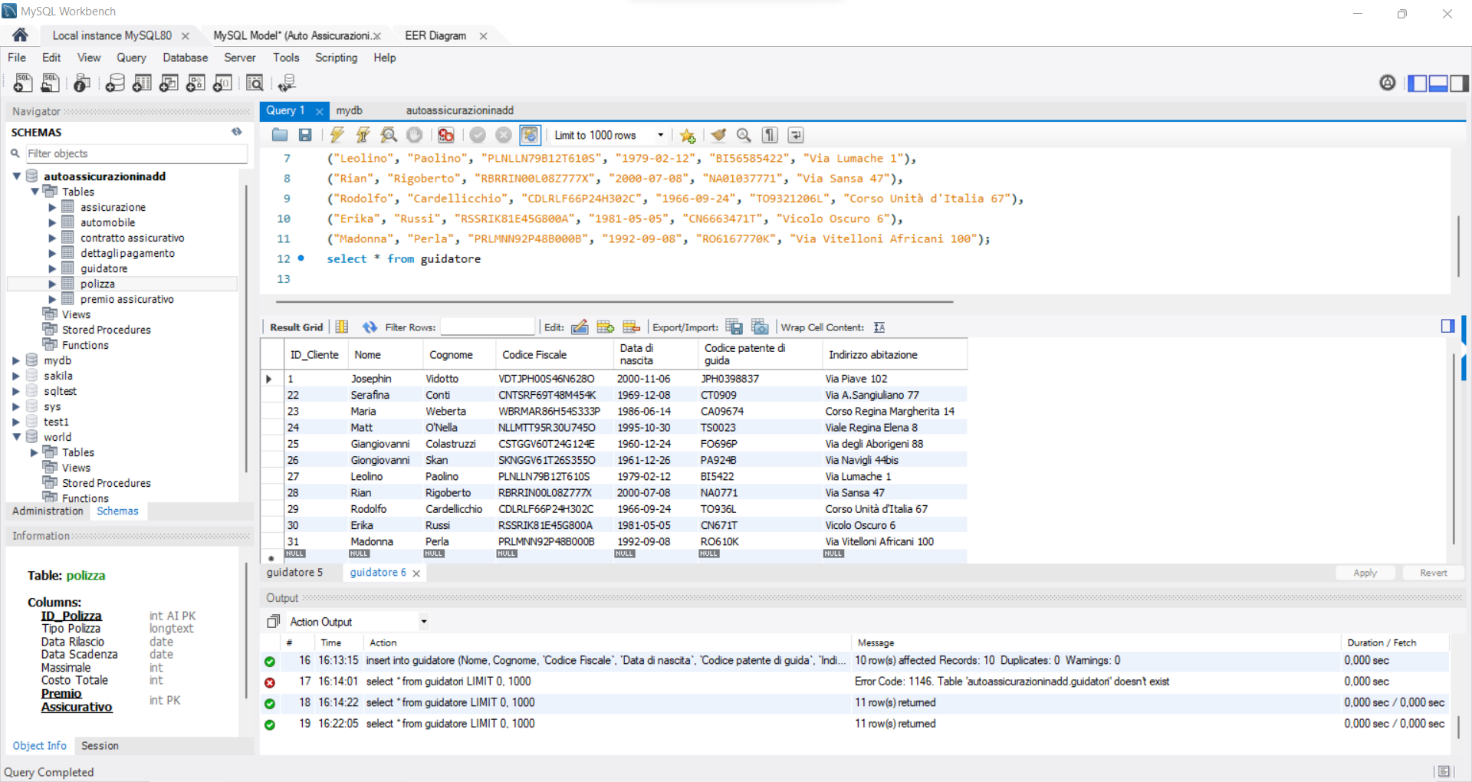
DELIMITER ;

Qui dovrebbe esserci tutto il database:



Ho aggiunto pure qualche screenshot delle query per inserire i dati.

Immagine che contiene testo

Descrizione generata automaticamenteImmagine che contiene testo

Descrizione generata automaticamente