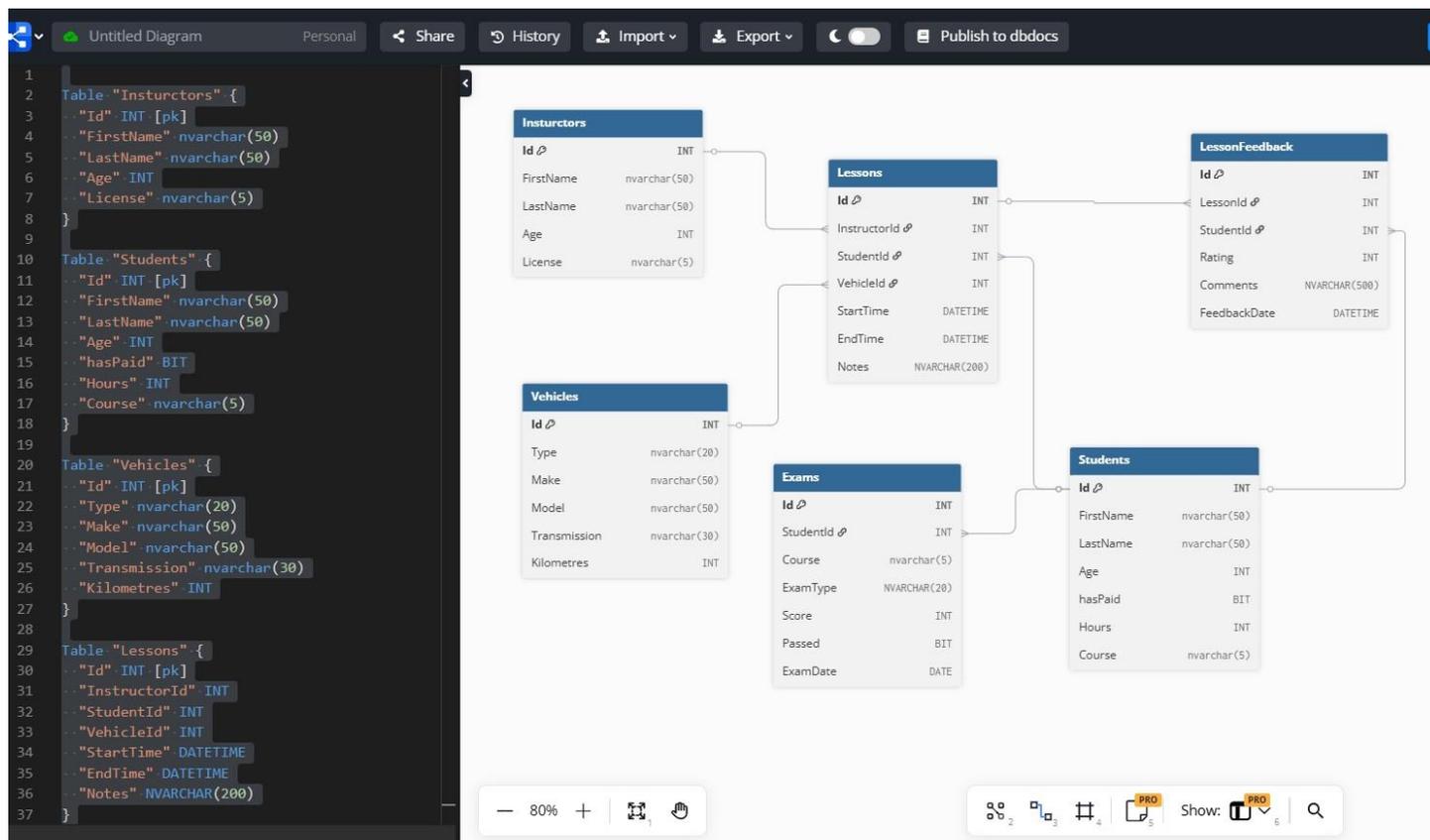


База данни за управление на дейността на автошкола / учебен център за шофьори



```

CREATE TABLE Insturctors(
  Id INT PRIMARY KEY ,
  FirstName nvarchar(50),
  LastName nvarchar(50),
  Age INT,
  License nvarchar(5)
);

```

```

Create TABLE Students(
  Id INT PRIMARY KEY,
  FirstName nvarchar(50),
  LastName nvarchar(50),
  Age INT,
  hasPaid BIT,
  Hours INT,
  Course nvarchar(5)
);

```

```

CREATE TABLE Vehicles(
  Id INT PRIMARY KEY,
  Type nvarchar(20),
  Make nvarchar(50),
  Model nvarchar(50),
  Transmission nvarchar(30),
  Kilometres INT
);

```

```

CREATE TABLE Lessons(
  Id INT PRIMARY KEY,
  InstructorId INT,

```

```

StudentId INT,
VehicleId INT,
StartTime DATETIME,
EndTime DATETIME,
Notes NVARCHAR(200),
FOREIGN KEY (InstructorId) REFERENCES Insturctors(Id),
FOREIGN KEY (StudentId) REFERENCES Students(Id),
FOREIGN KEY (VehicleId) REFERENCES Vehicles(Id)
);

```

```

CREATE TABLE Exams(
  Id INT PRIMARY KEY,
  StudentId INT ,
  Course nvarchar(5) ,
  ExamType NVARCHAR(20), -- Theory / Practical
  Score INT,
  Passed BIT,
  ExamDate DATE,
  FOREIGN KEY(StudentId) REFERENCES Students(Id)
);

```

```

CREATE TABLE LessonFeedback (
  Id INT PRIMARY KEY,
  LessonId INT,
  StudentId INT,
  Rating INT, -- Rating from 1 to 5
  Comments NVARCHAR(500),
  FeedbackDate DATETIME,
  FOREIGN KEY (LessonId) REFERENCES Lessons(Id),
  FOREIGN KEY (StudentId) REFERENCES Students(Id)
);

```

```

INSERT INTO Insturctors (Id, FirstName, LastName, Age, License) VALUES
(1, 'John', 'Miller', 45, 'B1'),
(2, 'Sarah', 'Johnson', 38, 'B1'),
(3, 'Michael', 'Brown', 50, 'C1'),
(4, 'Emily', 'Davis', 29, 'A1'),
(5, 'Chris', 'Wilson', 42, 'B1'),
(6, 'Anna', 'Thompson', 33, 'C1'),
(7, 'Robert', 'Garcia', 47, 'Ai'),
(8, 'Olivia', 'Martinez', 36, 'B1');

```

```

INSERT INTO Students (Id, FirstName, LastName, Age, hasPaid, Hours, Course) VALUES
(1, 'Mark', 'Stevens', 19, 1, 3, 'B1'),
(2, 'Julia', 'Roberts', 22, 0, 1, 'A1'),
(3, 'Anthony', 'Lopez', 18, 1, 5, 'B1'),
(4, 'Sophie', 'Lee', 24, 1, 7, 'C1'),
(5, 'Kevin', 'Harris', 21, 0, 2, 'B1'),
(6, 'Linda', 'Clark', 27, 1, 10, 'A1'),
(7, 'Brian', 'White', 20, 0, 0, 'C1'),
(8, 'Emma', 'Young', 23, 1, 4, 'B1');

```

```

INSERT INTO Vehicles (Id, Type, Make, Model, Transmission, Kilometres) VALUES
(1, 'Car', 'Toyota', 'Corolla', 'Automatic', 32000),
(2, 'Car', 'Honda', 'Civic', 'Manual', 41000),
(3, 'Motorcycle', 'Yamaha', 'MT-07', NULL, 15000),
(4, 'Truck', 'Volvo', 'FH16', 'Automatic', 92000),
(5, 'Car', 'Ford', 'Focus', 'Automatic', 28000),

```

```
(6, 'Truck', 'Mercedes', 'Actros', 'Manual', 110000),
(7, 'Motorcycle', 'Kawasaki', 'Ninja 400', NULL, 8000),
(8, 'Car', 'BMW', '320i', 'Automatic', 50000);
```

```
INSERT INTO Lessons (Id, InstructorId, StudentId, VehicleId, StartTime, EndTime, Notes) VALUES
(1, 1, 1, 1, '2025-02-01 09:00', '2025-02-01 10:00', 'B1'),
(2, 2, 3, 2, '2025-02-02 11:00', '2025-02-02 12:00', 'B1'),
(3, 4, 2, 3, '2025-02-03 14:00', '2025-02-03 15:00', 'A1'),
(4, 3, 4, 4, '2025-02-04 10:00', '2025-02-04 12:00', 'C1'),
(5, 5, 5, 5, '2025-02-05 09:30', '2025-02-05 10:30', 'A1'),
(6, 7, 6, 7, '2025-02-06 13:00', '2025-02-06 14:00', 'A1'),
(7, 6, 7, 6, '2025-02-07 08:00', '2025-02-07 10:00', 'C1'),
(8, 1, 8, 8, '2025-02-08 15:00', '2025-02-08 16:00', 'B1');
```

```
INSERT INTO Exams (Id, StudentId, Course, ExamType, Score, Passed, ExamDate) VALUES
(1, 1, 'CAR', 'Theory', 78, 1, '2025-03-01'),
(2, 1, 'CAR', 'Practical', 65, 1, '2025-03-05'),
(3, 2, 'MOTO', 'Theory', 55, 0, '2025-03-02'),
(4, 3, 'CAR', 'Theory', 82, 1, '2025-03-03'),
(5, 4, 'TRK', 'Theory', 90, 1, '2025-03-04'),
(6, 5, 'CAR', 'Theory', 40, 0, '2025-03-06'),
(7, 6, 'MOTO', 'Practical', 88, 1, '2025-03-07'),
(8, 8, 'CAR', 'Practical', 72, 1, '2025-03-08');
```

```
-- Изтрива таблицата за обратна връзка от уроци
DROP TABLE LessonFeedback;
```

```
-- Добавя телефонен номер към инструкторите
ALTER TABLE Insturctors
ADD Phone NVARCHAR(20);
```

```
-- Добавя имейл към студентите
ALTER TABLE Students
ADD Email NVARCHAR(50);
```

```
-- Изтрива всички студенти, които не са платили
DELETE FROM Students
WHERE hasPaid = 0;
```

```
-- Променя възрастта на студент с Id = 1
UPDATE Students
SET Age = 25
WHERE Id = 1;
```

```
-- Изчислява общия брой километри на всички превозни средства
SELECT SUM(Kilometres) AS TotalKilometres
FROM Vehicles;
```

```
-- Намира средната възраст на студентите
SELECT AVG(Age) AS AverageAge
FROM Students;
```

```
-- Намира най-високия и най-ниския резултат от изпитите
SELECT
    MAX(Score) AS MaxScore,
    MIN(Score) AS MinScore
```

FROM Exams;

-- Създава изглед с всички успешно издържани изпити

CREATE VIEW PassedExams AS

SELECT

S.FirstName,
S.LastName,
E.Course,
E.ExamType,
E.Score,
E.Passed,
E.ExamDate

FROM Students S

JOIN Exams E ON S.Id = E.StudentId

WHERE E.Passed = 1;

-- Показва всички записи от изгледа PassedExams

SELECT * FROM PassedExams;

-- Показва всички инструктори

SELECT * FROM Insturctors;

-- Показва всички студенти

SELECT * FROM Students;

-- Показва основна информация за превозните средства

SELECT Id, Make, Model, Type FROM Vehicles;

-- Извежда инструктори над 40 години

SELECT FirstName, LastName, Age

FROM Insturctors

WHERE Age > 40;

-- Показва пълнолетни студенти, които са платили

SELECT FirstName, LastName

FROM Students

WHERE hasPaid = 1 AND Age > 18;

-- Извежда студенти с курс B1 или под 20 години

SELECT *

FROM Students

WHERE Course = 'B1' OR Age < 20;

-- Подрежда превозните средства по километри във възходящ ред

SELECT Make, Model, Kilometres

FROM Vehicles

ORDER BY Kilometres ASC;

-- Показва уроци с информация за студент и превозно средство

SELECT

L.Id AS LessonId,
CONCAT(S.FirstName, ' ', S.LastName) AS StudentName,
CONCAT(V.Make, ' ', V.Model) AS Vehicle,
L.StartTime,
L.EndTime

FROM Lessons L

```
JOIN Students S ON L.StudentId = S.Id  
JOIN Vehicles V ON L.VehicleId = V.Id;
```

```
-- Брой колко уроци е провел всеки инструктор  
SELECT  
    CONCAT(I.FirstName, ' ', I.LastName) AS Instructor,  
    COUNT(L.Id) AS TotalLessons  
FROM Lessons L  
INNER JOIN Instructors I ON L.InstructorId = I.Id  
GROUP BY I.FirstName, I.LastName;
```