Autósiskola SQL adatbázis beadandó

# Logikai modell

# Fizikai modell

# Az adatbázis leírása

1. Diák tábla (Diak)

* általános leírás: alkalmas az autósiskola diákjairól adatokat eltárolni
* oszlopok
  + diakID: szám (int) típusú elsődleges kulcsa ennek a táblának
  + Nev: a diák neve; 50, vagy annál kevesebb UNICODE kódolású karakter lehet; KÉNYSZER: NEM NULL
  + Cim: a diák címe; 100, vagy annál kevesebb UNICODE kódolású karakter lehet
  + Email: a diák emailcíme: 50, vagy annál kevesebb UNICODE kódolású karakter lehet; KÉNYSZER: egyedi érték
  + Telefonszám: a diák telefonszáma; 50, vagy annál kevesebb UNICODE kódolású karakter lehet

1. Oktató tábla (Oktato)

* általános leírás: alkalmas az autósiskola oktatóiról adatokat eltárolni
* oszlopok
  + oktatoID: szám (int) típusú elsődleges kulcsa ennek a táblának
  + Nev: az oktató neve; 50, vagy annál kevesebb UNICODE kódolású karakter lehet; KÉNYSZER: NEM NULL
  + Cim: a diák címe; 100, vagy annál kevesebb UNICODE kódolású karakter lehet
  + Auto: az oktató autója; 50, vagy annál kevesebb UNICODE kódolású karakter lehet
  + Telefonszám: az oktató telefonszáma; 50, vagy annál kevesebb UNICODE kódolású karakter lehet
  + Email: az oktató emailcíme; 50, vagy annál kevesebb UNICODE kódolású karakter lehet; KÉNYSZER: egyedi érték

1. Foglalás tábla (Foglalas)

* általános leírás: alkalmas foglalások rögzítésére az autósiskolánál melyeket diákok tesznek a tanfolyam és az oktató megjelölésével. Rögzíthető benne a dátum is mikor a foglalás történt
* oszlopok
  + foglalasID: szám (int) típusú elsődleges kulcsa ennek a táblának
  + diakID: idegen kulcs a Diak táblából
  + oktatoID: idegen kulcs az Oktato táblából
  + tanfolyamID: idegen kulcs a Tanfolyam táblából
  + foglalasdatum: a foglalás dátumát lehet ide rögzíteni, dátum formátumban

1. Tanfolyam tábla (Tanfolyam)

* általános leírás: alkalmas az autósiskola tanfolyamainak a tárolására névvel, típussal, árral, kezdő és befejező dátummal
  + tanfolyamID: szám (int) típusú elsődleges kulcsa ennek a táblának
  + Tnev: a tanfolyam megnevezése; 50, vagy annál kevesebb UNICODE kódolású karakter lehet
  + Ttipus: a tanfolyam típusa 20, vagy annál kevesebb UNICODE kódolású karakter lehet
  + Dij: a tanfolyam ára; decimális értékkel; 0-99.999.999,99-ig; KÉNYSZER: nagyobb, mint 0
  + Tdatumtol: a tanfolyam kezdési dátuma; dátum formátumban megadható
  + Tdatumig: a tanfolyam befejezési dátuma; dátum formátumban megadható

1. Fizetés tábla (Fizetes)

* általános leírás: alkalmas az autósiskola számára a diákok által befizetett tanfolyami díjak nyomonkövetesére összeggel, fizetési dátummal és a diák azonosítójával
  + fizetesID: szám (int) típusú elsődleges kulcsa ennek a táblának
  + DiakID: idegen kulcs a Diak táblából
  + Fosszeg: a diák által befizetett összeg; decimális értékkel; 0-99.999.999,99-ig; KÉNYSZER: nagyobb, mint 0
  + Fdatum: a diák által befizetett összeg beérkezésének a dátuma; dátumérték

# Az adatbázis létrehozása + Kényszerek

CREATE TABLE Diak (

DiakID INT PRIMARY KEY,

Nev NVARCHAR(50),

Cim NVARCHAR(100),

Telefonszam NVARCHAR(50),

Email NVARCHAR(50)

);

ALTER TABLE Diak

ADD CONSTRAINT DNevnemNull CHECK (Nev IS NOT NULL);

ALTER TABLE Diak

ADD CONSTRAINT DEmailEgyedi UNIQUE (Email);

CREATE TABLE Oktato (

OktatoID INT PRIMARY KEY,

Nev NVARCHAR(50),

Cim NVARCHAR(100),

Auto NVARCHAR(50),

Telefonszam NVARCHAR(50),

Email NVARCHAR(50)

);

ALTER TABLE Oktato

ADD CONSTRAINT ONevnemNull CHECK (Nev IS NOT NULL);

ALTER TABLE Oktato

ADD CONSTRAINT OEmailEgyedi UNIQUE (Email);

CREATE TABLE Tanfolyam (

TanfolyamID INT PRIMARY KEY,

Tnev NVARCHAR(50),

Ttipus NVARCHAR(20),

Dij DECIMAL(10, 2),

Tdatumtol DATE,

Tdatumig DATE

);

ALTER TABLE Tanfolyam

ADD CONSTRAINT DijNemNegativ CHECK (Dij >= 0);

CREATE TABLE Foglalas (

FoglalasID INT PRIMARY KEY,

DiakID INT,

OktatoID INT,

TanfolyamID INT,

FoglalasDatum DATE,

FOREIGN KEY (DiakID) REFERENCES Diak(DiakID),

FOREIGN KEY (OktatoID) REFERENCES Oktato(OktatoID),

FOREIGN KEY (TanfolyamID) REFERENCES Tanfolyam(TanfolyamID)

);

CREATE TABLE Fizetes (

FizetesID INT PRIMARY KEY,

DiakID INT,

Fosszeg DECIMAL(10, 2),

Fdatum DATE,

FOREIGN KEY (DiakID) REFERENCES Diak(DiakID)

);

ALTER TABLE Fizetes

ADD CONSTRAINT OsszegNemNegativ CHECK (Fosszeg >= 0);

# Az adatbázis feltöltése

-- Diak tábla

INSERT INTO Diak (DiakID, Nev, Cim, Telefonszam, Email)

VALUES

(1, 'John Doe', '123 Main St, Anytown', '123-456-7890', 'john.doe@example.hu'),

(2, 'Jane Smith', '456 Elm St, Othertown', '987-654-3210', 'jane.smith@example.hr'),

(3, 'Alice Johnson', '789 Oak St, Somewhere', '555-123-4567', 'alice.johnson@example.com'),

(4, 'Bob Jones', '321 Pine St, Anycity', '111-222-3333', 'bob.jones@example.hu'),

(5, 'Emma White', '654 Maple St, Othercity', '444-555-6666', 'emma.white@example.hr'),

(6, 'Michael Brown', '987 Cedar St, Anothercity', '777-888-9999', 'michael.brown@example.com'),

(7, 'Emily Davis', '123 Oak St, Anytown', '222-333-4444', 'emily.davis@example.hu'),

(8, 'David Wilson', '456 Elm St, Othertown', '555-666-7777', 'david.wilson@example.hr'),

(9, 'Sarah Taylor', '789 Pine St, Somewhere', '888-999-0000', 'sarah.taylor@example.com'),

(10, 'Daniel Martinez', '321 Maple St, Anycity', '999-000-1111', 'daniel.martinez@example.hu'),

(11, 'Laura Lee', '987 Oak St, Anytown', '123-456-7890', 'laura.lee@example.hr'),

(12, 'Michael Johnson', '456 Elm St, Othertown', '987-654-3210', 'michael.johnson@example.com'),

(13, 'Sophia Smith', '789 Oak St, Somewhere', '555-123-4567', 'sophia.smith@example.hu'),

(14, 'William Davis', '321 Pine St, Anycity', '111-222-3333', 'william.davis@example.hr'),

(15, 'Olivia Brown', '654 Maple St, Othercity', '444-555-6666', 'olivia.brown@example.com'),

(16, 'James Taylor', '987 Cedar St, Anothercity', '777-888-9999', 'james.taylor@example.hu'),

(17, 'Emily Wilson', '123 Oak St, Anytown', '222-333-4444', 'emily.wilson@example.hr'),

(18, 'Benjamin Lee', '456 Elm St, Othertown', '555-666-7777', 'benjamin.lee@example.com'),

(19, 'Abigail Taylor', '789 Pine St, Somewhere', '888-999-0000', 'abigail.taylor@example.hu'),

(20, 'Ethan Martinez', '321 Maple St, Anycity', '999-000-1111', 'ethan.martinez@example.com');

-- Oktato tábla

INSERT INTO Oktato (OktatoID, Nev, Cim, Auto, Telefonszam, Email)

VALUES

(1, 'Michael Brown', '123 Oak St, Anytown', 'VW Passat B5', '111-222-3333', 'michael.brown@example.com'),

(2, 'Emily Davis', '456 Elm St, Othertown', 'BMW E46', '444-555-6666', 'emily.davis@example.com'),

(3, 'David Wilson', '789 Pine St, Somewhere', 'Mercedes E class', '777-888-9999', 'david.wilson@example.com'),

(4, 'Sarah Taylor', '321 Maple St, Anycity', 'VW Passat B5', '999-888-7777', 'sarah.taylor@example.com'),

(5, 'Daniel Martinez', '654 Cedar St, Othercity', 'Dacia Duster', '333-222-1111', 'daniel.martinez@example.com');

-- Tanfolyam tábla

INSERT INTO Tanfolyam (TanfolyamID, Tnev, Ttipus, Dij, Tdatumtol, Tdatumig)

VALUES

(1, 'Driving Course but more expensive', 'B2', 50000.00, '2024-06-01', '2024-06-30'),

(2, 'Fast Driving', 'B2+', 30000.00, '2024-07-15', '2024-07-28'),

(3, 'Driving Course', 'B2', 40000.00, '2024-08-10', '2024-08-31');

-- Foglalas tábla

INSERT INTO Foglalas (FoglalasID, DiakID, OktatoID, TanfolyamID, FoglalasDatum)

VALUES

(1, 1, 1, 1, '2024-05-14'),

(2, 2, 2, 2, '2024-05-15'),

(3, 3, 4, 3, '2024-05-16'),

(4, 4, 4, 1, '2024-05-17'),

(5, 5, 5, 2, '2024-05-18'),

(6, 6, 1, 3, '2024-05-19'),

(7, 7, 2, 1, '2024-05-20'),

(8, 8, 4, 2, '2024-05-21'),

(9, 9, 4, 3, '2024-05-22'),

(10, 10, 5, 1, '2024-05-23');

-- Fizetes tábla

INSERT INTO Fizetes (FizetesID, DiakID, Fosszeg, Fdatum)

VALUES

(1, 1, 50000.00, '2024-04-14'),

(2, 2, 30000.00, '2024-04-15'),

(3, 3, 40000.00, '2024-04-16'),

(4, 4, 50000.00, '2024-04-17'),

(5, 5, 30000.00, '2024-04-25'),

(6, 6, 40000.00, '2024-04-27');

# 6 db hasznos lekérdezés, melyek kihasználják a tanultakat

## Április 21-től a mai dátumig befizetett tandíjak összege (Ft)

SELECT concat(sum(fosszeg), ' Ft') as 'aprilis 21-tol a mai datumig befizetett osszeg:'

FROM Fizetes

WHERE fdatum BETWEEN '2024-04-21'and GETDATE()

## Milyen típusú járművön hány diák fog vezetni? Csökkenő sorba rendezve a legnépszerűbb autótól

SELECT o.Auto, count(\*) as 'Darab diak'

FROM Oktato o JOIN Foglalas f ON o.OktatoID = f.OktatoID

JOIN Diak d ON f.DiakID = d.DiakID

GROUP BY o.auto

ORDER BY count(\*) DESC

## Csoportosítsuk emailcímeik végződése szerint, hogy mennyi diák nem fizetett még . Csak a .hu-s és a .hr-es emailcímek érdekelnek.

SELECT COUNT(\*) as 'diakdb', RIGHT(d.Email, CHARINDEX('.', REVERSE(d.Email)) - 1) AS 'Email vegzodes'

FROM Diak d

LEFT JOIN Fizetes f ON d.DiakID = f.DiakID

WHERE f.DiakID IS NULL

GROUP BY RIGHT(d.Email, CHARINDEX('.', REVERSE(d.Email)) - 1)

HAVING RIGHT(d.Email, CHARINDEX('.', REVERSE(d.Email)) - 1) = 'hr' OR

RIGHT(d.Email, CHARINDEX('.', REVERSE(d.Email)) - 1) = 'hu'

## Mely ID-jú oktatókhoz van óra felvéve és melyekhez nincs? (Halmazművelettel)

SELECT OktatoID, 'Van hozzá óra felvéve' AS 'Foglalt'

FROM Foglalas

UNION

SELECT OktatoID, 'Nincs hozzá óra felvéve' AS 'Foglalt'

FROM Oktato

WHERE OktatoID NOT IN (SELECT DISTINCT OktatoID FROM Foglalas);

## Tanfolyam típusonként és oktatónként hány tanfolyam lesz? Kíváncsiak vagyunk a típusonkénti összegre is és az összes típus összegére is. (ROLLUP-pal)

SELECT

CASE

WHEN t.Ttipus IS NULL THEN 'összes fajta tanfolyam'

ELSE t.Ttipus

END AS 'Tanfolyam típus',

CASE

WHEN o.Nev IS NULL THEN '-összesen oktatott tanfolyamok:'

ELSE o.Nev

END AS 'Oktato',

COUNT(\*) AS 'Oktatott tanfolyam db'

FROM Tanfolyam t

LEFT JOIN Foglalas f ON t.TanfolyamID = f.TanfolyamID

LEFT JOIN Oktato o ON f.OktatoID = o.OktatoID

GROUP BY ROLLUP(t.Ttipus, o.Nev)

## A VW Passat B5-ön oktató kollégák elérhetősége (ha van hozzájuk foglalva): (beágyazott selectekkel)

SELECT Nev, Telefonszam, Email

FROM Oktato

WHERE

OktatoID IN (SELECT OktatoID FROM Foglalas

WHERE OktatoId IN (SELECT OktatoID FROM Oktato WHERE Auto = 'VW Passat B5')

);