**BEADANDÓ FELADAT – 2024.05.15.**

Az adott témának megfelelő SQL adatbázis megtervezése:

**Autósiskola adatbázisa**

Medgyesi Szabolcs G02

**Tartalom**

[I. Az adatbázis ismertetése 2](#_Toc166632780)

[II. Adatbázis megtervezése (logikai modell) 3](#_Toc166632781)

[III. Adatbázis megvalósításai (Fizikai modell) 4](#_Toc166632782)

[IV. Lekérdezések 6](#_Toc166632783)

[1. Lekérdezés 6](#_Toc166632784)

[2. Lekérdezés 6](#_Toc166632785)

[3. Lekérdezés 7](#_Toc166632786)

[4. Lekérdezés 7](#_Toc166632787)

[5. Lekérdezés 7](#_Toc166632788)

[6. Lekérdezés 8](#_Toc166632789)

# Az adatbázis ismertetése

Az alábbiakban bemutatom az adatbázis struktúráját és annak célját, valamint a felhasználási lehetőségeit. Az adatbázis, egy autósiskola adatainak adminisztrációjára és oktatási folyamatainak nyomon követésére lett tervezve. Ez egy strukturált és jól szervezett adattárolási lehetőséget biztosít, amely lehetővé teszi az oktatók, diákok, órák, foglalások és járművek adatainak kezelését, valamint az adatok könnyű lekérdezhetőségét. Az adatbázis 5 táblából áll. Ezek:

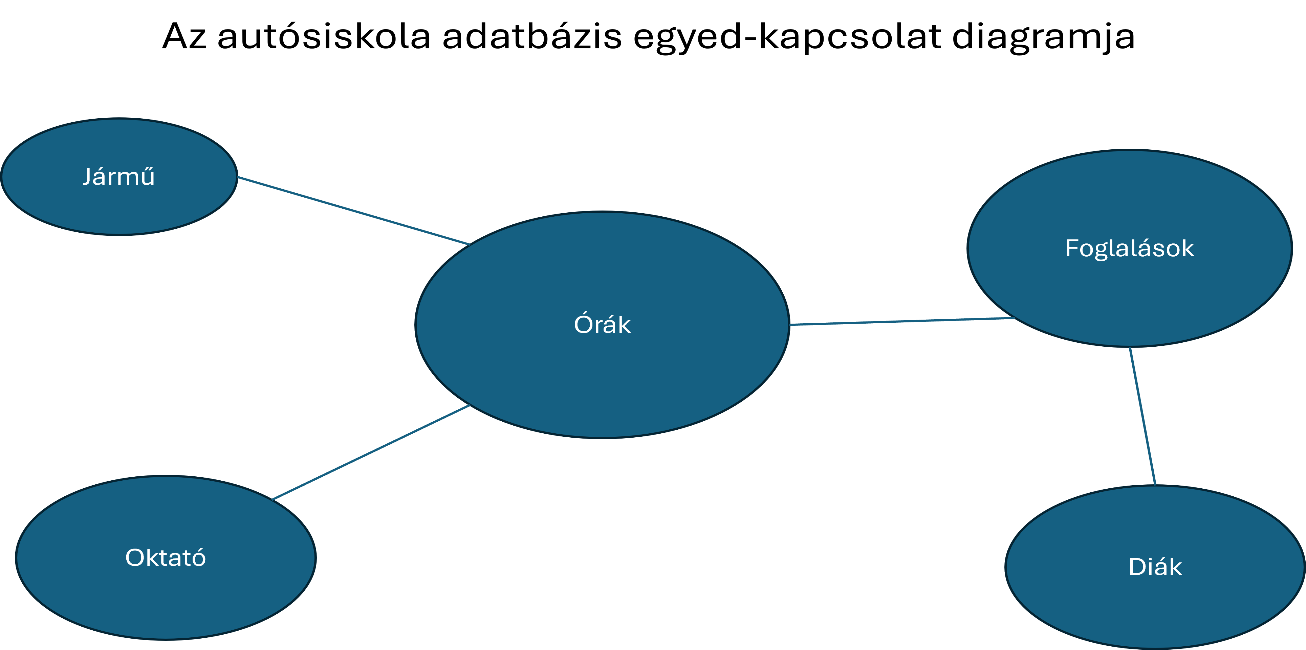
a diák tábla (diak) – amely a diákok adatait tárolja

a foglalás tábla (foglalas) – ebben a táblában vannak rögzítve a diákok által tett foglalások

a jármű tábla (jarmu) – ez a tábla az autósiskolán belül használt járművek adatait tárolja, amelyeket az oktatási órákhoz használnak

az oktató tábla (oktató) – ebben a táblában tárolják az autósiskolánál dolgozó oktatók adatait

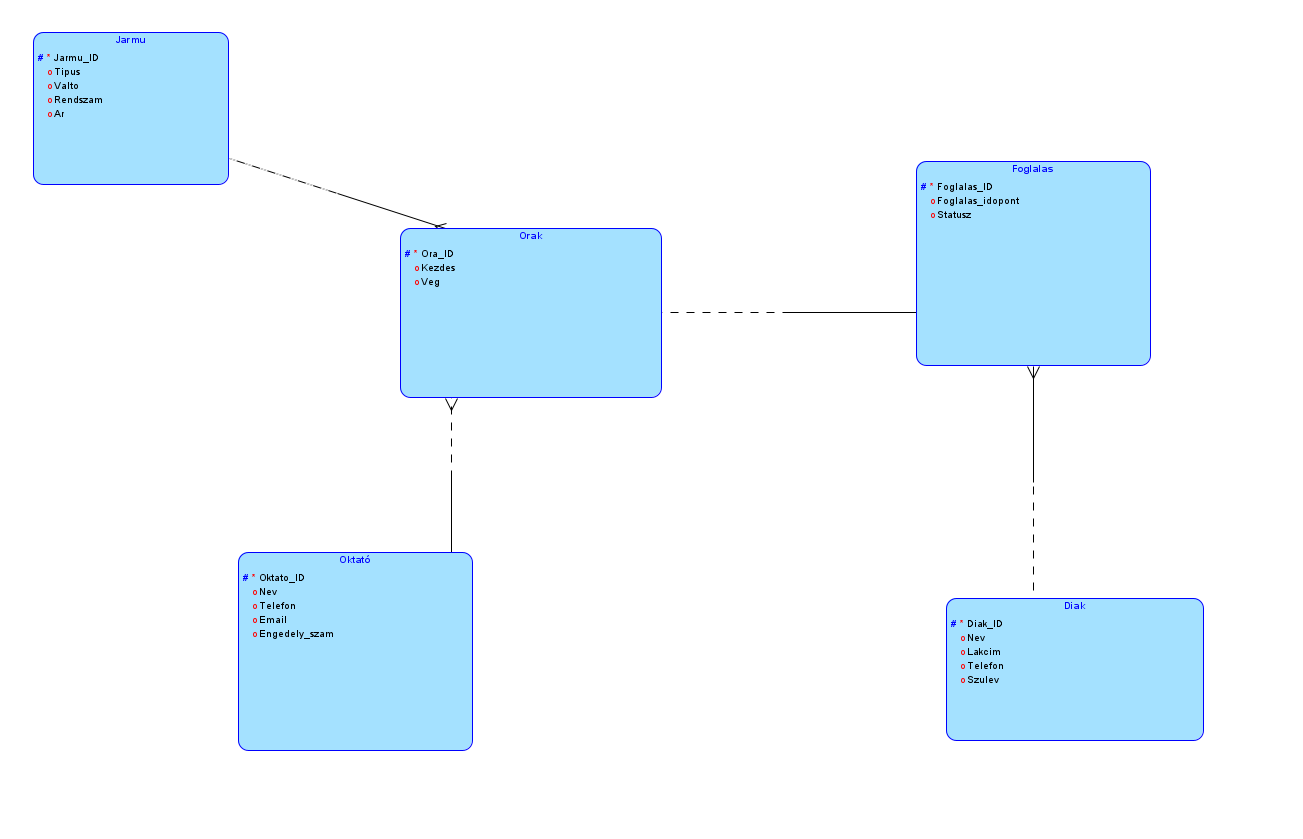
az órák tábla (orak) – ez a tábla rögzíti az egyes foglalásokhoz kapcsolt oktatási órákat

Az adatbázisban található adattáblák kapcsolatát az alábbi egyed-kapcsolat diagram mutatja. 

A diákok adatai az iskolánál való regisztrálás után bekerülnek az adatbázisba és a *Diak* táblában tárolódnak. A tanulni vágyó diákok ezután megkezdhetik a képzését, vagyis időpontokat foglalhatnak, amelyek adatai bekerülnek a *Foglalas* táblába. Innen a rendszer egy tanórát rendel a foglaláshoz, amelynek adatait az *Orak* táblában tárolja. Ezekhez az órákhoz tartozik egy jármű, amelynek adatai a *Jarmu* táblában találhatóak. Ezen kívül kapcsolódik hozzá egy, a tanórát tartó oktató is, akinek adatait az *Oktató* táblában tárolja a rendszer.

Így bármilyen adatot gyorsan és könnyedén el lehet érni, az adatbázison belül. Ez könnyebbé teszi a logisztikai és adminisztrációs feladatokat az iskola számára.

# Adatbázis megtervezése (logikai modell)



A logikai modell célja, hogy strukturált formában ábrázolja az adatokat és az azok közötti kapcsolatokat, anélkül, hogy részletezné az adatbázis fizikai implementációját. Az alábbiakban röviden összefoglalom az autósiskola adatbázisának logikai modelljét, bemutatva az egyes entitásokat és azok attribútumait:

**diak entitás**:

* Diak\_id (PK): Minden diákhoz egyedi azonosító tartozik.
* Nev: A diák neve.
* Lakcim: A diák lakcíme.
* Telefon: A diák telefonszáma.
* Szulev: A diák születési dátuma.

**foglalas entitás**:

* Foglalas\_id (PK): Az egyes foglalások egyedi azonosítója.
* Foglalas\_idopont: A foglalás időpontja, percre pontosan.
* Statusz: A foglalás jelenlegi státusza, attól függ, hogy elfogadták-e az órát, megtörtént már vagy lemondták.

**órák entitas**:

* Ora\_id (PK): Minden órának egyedi azonosítója van.
* Kezdes: Az óra kezdésének időpontja, percre pontosan.
* Veg: Az óra befejezésének időpontja.

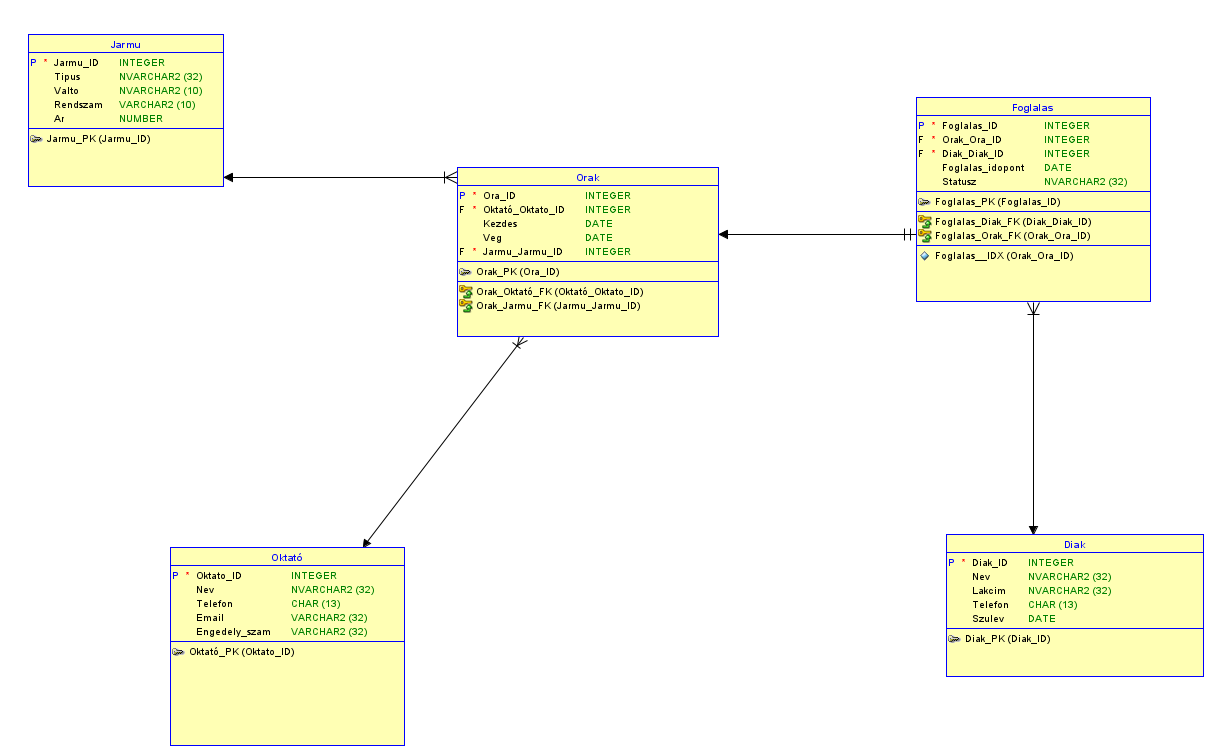
**jarmu entitás**:

* Jarmu\_id (PK): A egyes járműveknek is van egyedi azonosítója.
* Tipus: A jármű típusa.
* Valto: A jármű váltójának típusa, fontos mivel más jogosítvány jár az automata és a manuális váltóval rendelkező autókért.
* Rendszam: A jármű rendszáma.
* Ar: A jármű tanóránkként fizetendő ára.

**oktató entitás**:

* Oktato\_id (PK): Az oktatókhoz tartozó egyedi azonosító.
* Nev: Az oktató neve.
* Telefon: Az oktató telefonszáma.
* Email: Az oktató e-mail címe.
* Engedely\_szam: Az oktató oktatási engedély száma.

# Adatbázis megvalósításai (Fizikai modell)

(Az adatbázis kódja, még a fizikai modellhez tartozik, ez külön fájlban található.)

A fizikai modell részletezi az adatbázis fizikai implementációját, vagyis azt, hogy az adatok és a kapcsolatok hogyan vannak tárolva a valóságban, egy konkrét adatbázis-kezelő rendszerben.

Az egyes táblák közötti kapcsolatokat a következőképpen lehet bemutatni:

A diák megtesz egy foglalást (Diak - Foglalas): 1/N Kapcsolat mivel egy diák sok foglalást megtehet.

A foglaláshoz a rendszer rendel egy tanórát (Foglalas - Orak): 1/1 Kapcsolat mivel egy foglaláshoz csupán egy óra tartozhat.

A tanórához a rendszer foglal egy járművet (Orak – Jarmu): 1/N Kapcsolat mivel egy jármű több tanórán részt vehet.

A tanórához a rendszer hozzáad egy oktatót is (Orak - Oktató): 1/N Kapcsolat mivel egy oktató sok órát megtarthat.

Az adatbázisban szereplő adatok típus szerint:

**Diak tábla:**

Diak\_ID: (INTEGER) NOT NULL

Nev: (NVARCHAR (32))

Lakcim: (NVARCHAR (32))

Telefon: CHAR (13)

Szulev: DATE

Elsődleges kulcs: Diak\_PK (Diak\_ID)

**Foglalas tábla:**

Foglalas\_ID: (INTEGER) NOT NULL

Orak\_Ora\_ID: (INTEGER) NOT NULL

Diak\_Diak\_ID: (INTEGER) NOT NULL

Foglalas\_idopont: (DATETIME)

Statusz: (NVARCHAR (32))

Elsődleges kulcs: Foglalas\_PK (Foglalas\_ID)

Idegen kulcsok: Foglalas.Diak\_Fk (Diak.Diak\_ID), Foglalas.Orak\_FK (Orak.Ora\_ID)

**Orak tábla**

Ora\_ID: (INTEGER) NOT NULL

Oktató\_Oktato\_ID: (INTEGER)

Kezdes: (DATETIME)

Veg: (DATETIME)

Jarmu\_Jarmu\_ID: (INTEGER) NOT NULL

Elsődleges kulcs: Orak\_PK (Orak\_ID)

Idegen kulcsok: Orak.Oktató\_Fk (Oktató.Oktato\_ID), Orak.Jarmu\_FK (Jarmu.Jarmu\_ID)

**Jarmu tábla**

Jarmu\_ID: (INTEGER) NOT NULL

Tipus: (NVARCHAR (32))

Valto: (NVARCHAR (10))

Rendszam: (VARCHAR (10))

Ar: (DECIMAL)

Elsődleges kulcs: Jarmu\_PK (Jarmu\_ID)

**Oktató tábla**

Oktató\_ID: (INTEGER) NOT NULL

Nev: (NVARCHAR (32))

Telefon: (VARCHAR (13))

Email: VARCHAR (32))

Engedely\_szam: VARCHAR (32))

Elsődleges kulcs: Oktató\_PK (Oktato\_ID)

Miután az üres adatbázis elkészült, ez átmenetileg fel lett töltve random generált, teszt adatokkal, melynek a kódja megtalálható egy külön fájlban.

# Lekérdezések

### Lekérdezés

Az első lekérdezés kilistázza az összes foglalást és ezek részleteit. Ilyen a foglalás azonosító, a tanuló neve, a foglalás ideje, a foglalás státusza és az órát tartó oktató neve.

Ez a lekérdezés hasznos lehet az autósiskola adminisztrációjának, mivel átlátható és strukturált módon mutatja be a foglalásokat, valamint az azokhoz kapcsolódó tanulókat és oktatókat. Segítségével könnyebben nyomon követhetők az egyes foglalások, és hatékonyabban lehet kezelni az oktatási órákat.

SELECT F.foglalas\_id AS 'Foglalás azonosítója',

D.nev AS 'Tanuló neve',

F.foglalas\_idopont AS 'Foglalás ideje',

F.statusz AS 'Foglalás státusza',

OK.nev AS 'Oktató neve'

FROM diak D JOIN foglalas F ON D.diak\_id = F.diak\_diak\_id

JOIN orak O ON F.orak\_ora\_id = O.ora\_id

JOIN oktató OK ON O.oktató\_oktato\_id = OK.oktato\_id

### Lekérdezés

Ez a lekérdezés megmutatja az egyes oktatókat, az engedély számukat, valamint csoportosítja őket és kiszámolja az összes bevételüket az iskolánál.

Ez a lekérdezés lehetővé teszi az egyes oktatók teljesítményének és bevételének nyomon követését. Ezáltal összehasonlíthatók az oktatók teljesítményei és az oktatási órákból származó bevételük, ami segíthet a hatékonyabb üzleti döntések meghozatalában és a teljesítményük javításában.

SELECT O.nev AS 'Oktató neve',

O.engedely\_szam,

SUM(J.ar) AS 'Teljes bevétel'

FROM oktató O JOIN orak R ON O.oktato\_id = R.oktató\_oktato\_id

JOIN foglalas F ON R.ora\_id = F.orak\_ora\_id

JOIN jarmu J ON R.jarmu\_jarmu\_id = J.jarmu\_id

GROUP BY O.nev, O.engedely\_szam;

### Lekérdezés

A következő lekérdezés járműnkként listázza az egyes típusokat, az óránként fizetendő árát, az utolsó használatának időpontját, státuszát és a legtöbbet rajta tanult diáknak az azonosítóját.

Ez a lekérdezés megmutatja a járművekkel kapcsolatos fontos információkat, valamint segíti a diákok preferenciájának összegyűjtését és áttekintését.

SELECT j.jarmu\_id AS 'Jármű azonosítója',

j.tipus AS 'Jármű tipusa',

j.ar AS 'Óránkénti ára',

MAX(R.kezdes) AS 'Utolsó használat időpontja',

f.statusz AS 'A foglalás státusza',

MAX(F.diak\_diak\_id) AS 'Az autón legtöbbet tanult diák azonosítója'

FROM foglalas F JOIN orak R ON F.orak\_ora\_id = R.ora\_id

JOIN jarmu J ON J.jarmu\_id = R.jarmu\_jarmu\_id

GROUP BY j.jarmu\_id, j.tipus, j.ar, f.statusz

### Lekérdezés

Ez a lekérdezés megmutatja egy bizonyos oktatónál tanult diákokat, valamint az általuk használt különböző járművek számát. Ez az információ segíthet az oktatók teljesítményének elemzésében és a járművek használatának nyomon követésében.

SELECT D.nev AS 'Diák neve',

COUNT (DISTINCT J.jarmu\_id) AS 'Különböző járművek száma'

FROM diak D JOIN foglalas F ON D.diak\_id = F.diak\_diak\_id

JOIN orak O ON F.orak\_ora\_id = O.ora\_id

JOIN jarmu j ON O.jarmu\_jarmu\_id = J.jarmu\_id

WHERE O.oktató\_oktato\_id = (

SELECT oktato\_id

FROM oktató

WHERE nev = 'Szabó András')

GROUP BY D.nev;

### Lekérdezés

Az 5. lekérdezés listázza ez egyes foglalásokat és azok információit, valamint kategorizálja az egyes órákat a kifizetett egységár alapján 4 csoportba. Ez segíthet az árak menedzselésében és a különböző árkategóriákhoz tartozó járművek megfelelő elhelyezésében az autósiskola kínálatában.

SELECT

F.foglalas\_id AS 'Foglalás azonosító',

F.foglalas\_idopont AS 'Foglalás időpont',

J.ar AS 'Foglalás egységára',

CASE

WHEN J.ar <5500 THEN 'Alacsony ár'

WHEN J.ar BETWEEN 5500 AND 7000 THEN 'Közepes ár'

WHEN J.ar BETWEEN 7000 AND 8500 THEN 'Magas ár'

ELSE 'Nagyon magas ár'

END AS 'Ár kategória'

FROM diak D JOIN foglalas F ON D.diak\_id = F.diak\_diak\_id

JOIN orak O ON F.orak\_ora\_id = O.ora\_id

JOIN jarmu j ON O.jarmu\_jarmu\_id = J.jarmu\_id

### Lekérdezés

Az utolsó lekérdezés megmutatja azon oktatókat és azon óráik számát, amelyek 90 percnél hosszabb ideig tartottak. Ez az információ segíthet az oktatók teljesítményének értékelésében és az oktatási órák időtartamának optimalizálásában.

SELECT O.nev AS 'Oktató neve',

COUNT(R.ora\_id) AS 'Összes 90 percnél hosszabb óra'

FROM oktató O JOIN orak R ON O.oktato\_id = R.oktató\_oktato\_id

JOIN foglalas F ON R.ora\_id = F.orak\_ora\_id

WHERE DATEDIFF (minute, r.kezdes, r.veg) > 90

GROUP BY o.nev;