

JEGYZŐKÖNYV

Adatbázisrendszerek I.

Féléves feladat: Zeneadatbázis

Készítette: Simonyák János

Neptunkód: MZ727W

Gyakorlat: Szerda, 12-14

Gyakorlatvezető: Dr. Bednarik László

A feladat leírása

A feladatom egy zeneadatbázis létrehozása, amelyben lemezkiadókat, az általuk kiadott zenei albumokat, az azokon levő dalokat, az azokat szerző zenészerzőket, és egyéb énekeseket lehet tárolni.

A **Kiadó** egyed a lemezkiadók információt tárolja, elsődleges kulccsal együtt. Ezek a kiadó azonosítója, neve, alapítási dátuma, és a székhelye, ami többbelemű, tartalmazza az országot és a várost.

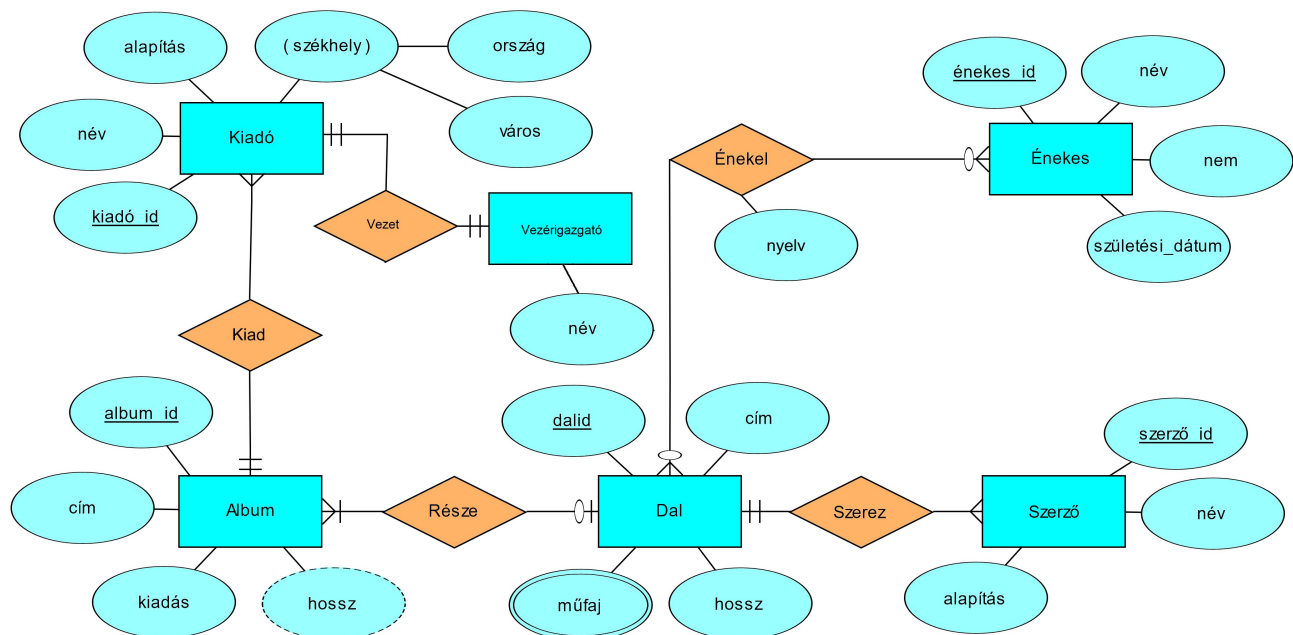
Az **Album** egyed 1:N kapcsolatban van a kiadóval. Továbbá elemei közé tartozik egy elsődleges kulcs, az album címe, a kiadás dátuma, és származtatott adatként az album hossza. Ennél az egyednél nem kötelező kiadót megadni, lévén, hogy akár függetlenül is ki lehet adni albumot.

A **Dal** egyed 1:N kapcsolatban van a Szerzővel. Hasonlóan a Kiadó-Szerző kapcsolathoz, egy szerzőnek lehet több dala is, viszont egy dal csak egy szerző munkája lehet. Ennek az egyednek is van egy elsődleges kulcsa, egy címe, egy hossza, és egy többértékű eleme, ami a műfaj.

A Dal egyedhez két másik egyed csatlakozik. Ezek egyike a **Szerző** egyed, 1:N kapcsolatban. Minden Szerző egyedhez tartozhat több dal is, viszont egy dal csak egy szerző munkássága lehet. Ezen kívül tartalmaz egy elsődleges kulcsot, egy nevet, és az alapítás évét, amennyiben együttesről van szó. Nem együttes esetén az első megjelent dal/album kiadási dátuma lesz az alapítás dátuma.

A Dal egyedhez kapcsolódó másik egyed az **Énekes**. Ezek N:M kapcsolatban vannak egymással, vagyis egy dalnak több énekes is lehet, valamint egy énekes több dalban is énekelhet. Ugyanúgy előfordulhat, hogy egy dalnak például nincs énekes. Az Énekes itt a szerzőtől független, szabadúszó, vagy közreműködő személyt jelent. Elemei az elsődleges kulcs, az énekes neve, neme és születési dátuma.

Az adatbázis ER modellje



Az adatbázis konvertálása relációs modellre

A **Kiadó** egyed átalakításakor arra kell figyelni, hogy a többemű *székhely* elem szét lesz bontva arra a két mezőre, amit tartalmaz; az *ország* és a *város* mezőkre.

A **Dal** egyed két idegen kulcsot kap; *album_id* és *szerző_id*. A *műfaj* elem további két táblázatra bomlik. Egy új **Műfaj** táblára, és az ezt és a Dal táblákat összekötő **Műfaj-Dal** kapcsolótáblára. Az előbbi az adott műfajok megnevezését és egyedi kulcsát tartalmazza, az utóbbi pedig két idegen kulcsot, amikkel a Műfaj és a Dal táblákhoz kapcsolódik.

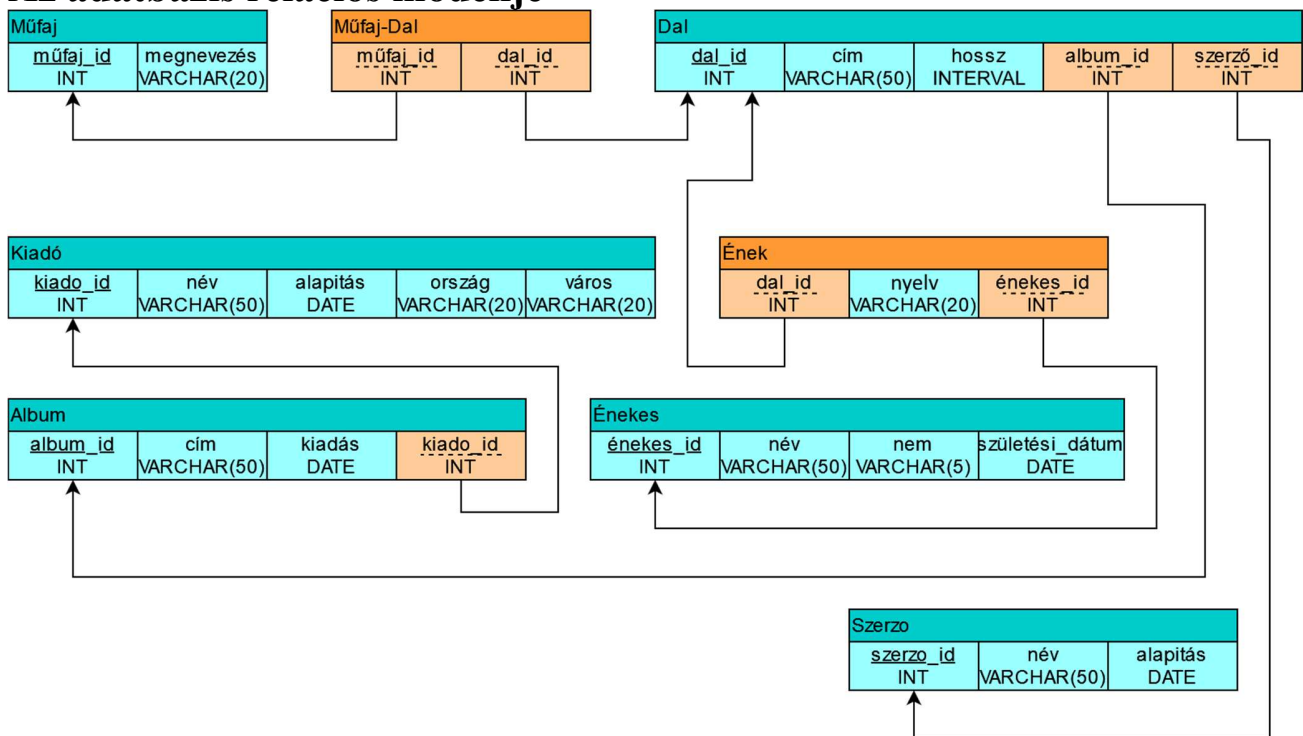
Az **Album** táblába a származtatott *hossz* elemen kívül minden eleme bekerül, mivel azt később lekérdezésből könnyen kiszámíthatjuk. Továbbá kap egy idegen kulcsot is, amelyik specifikus kiadóra mutathat.

A **Szerző** egyed meglévő elemein kívül kap egy *kiadó_id* nevű idegen kulcsot, amin keresztül hivatkozik a Kiadó egyed elemeire.

Az **Énekes és Dal** egyedeket összekötő kapcsolat külön kapcsolótáblává konvertálódik, melnek elemei az eddig is jelen levő *nyelv* mező, és két új idegen kulcs.

Így összesen nyolc darab tábla jön létre.

Az adatbázis relációs modellje



Az adatbázis relációs sémája:

Műfaj [műfaj_id, megnevezés]
Műfaj-Dal [műfaj_id, dal_id]
Dal [dal_id, cím, hossz, album_id, szerző_id]
Ének [dal_id, nyelv, énekes_id]
Énekes [énekes_id, név, nem, születési dátum]
Kiadó [kiadó_id, név, alapítás, ország, város]
Album [album_id, cím, kiadás, kiadó_id]
Szerző [szerző_id, név, alapítás]

A táblák létrehozása

A táblák létrehozásánál ügyelni kell a sorrendre. Nem lehet létrehozni olyan táblát, aminek egy idegen kulcs egy jelenleg még nem létező tábla mezőjére mutat. Ezért először az idegen kulcsot nem tartalmazó táblák lesznek létrehozva.

```
CREATE TABLE Kiado (  
    kiadoid INT PRIMARY KEY NOT NULL,  
    nev VARCHAR(50),  
    alapitas DATE,  
    orszag VARCHAR(20),  
    varos VARCHAR(20)  
);  
  
CREATE TABLE Szerzo (  
    szerzoid INT PRIMARY KEY NOT NULL,  
    nev VARCHAR(50),  
    alapitas DATE  
);  
  
CREATE TABLE Mufaj (  
    mufajid INT PRIMARY KEY,  
    megnevezes VARCHAR(20)  
);  
  
CREATE TABLE Enekes (  
    enekesid INT PRIMARY KEY NOT NULL,  
    nev VARCHAR(50),  
    nem VARCHAR(20),  
    szuletes DATE  
);  
  
CREATE TABLE Album (  
    albumid INT PRIMARY KEY NOT NULL,  
    cim VARCHAR(50),  
    kiadas DATE,  
    kiadoid int,  
    FOREIGN KEY (kiadoid) REFERENCES Kiado(kiadoid)  
);  
  
CREATE TABLE Dal (  
    dalid INT PRIMARY KEY NOT NULL,  
    cim VARCHAR(50),  
    hossz DATE,  
    szerzoid INT,  
    albumid INT,  
    FOREIGN KEY (szerzoid) REFERENCES Szerzo(szerzoid),  
    FOREIGN KEY (albumid) REFERENCES Album(albumid)  
);
```

```

CREATE TABLE Enek (
    dalid INT,
    nyelv VARCHAR(20),
    enekesid INT,
    FOREIGN KEY (dalid) REFERENCES Dal(dalid),
    FOREIGN KEY (enekesid) REFERENCES Enekes(enekesid)
);

CREATE TABLE Mufaj_Dal (
    mufajid INT,
    dalid INT,
    FOREIGN KEY (mufajid) REFERENCES Mufaj(mufajid),
    FOREIGN KEY (dalid) REFERENCES Dal(dalid)
);

```

A táblák feltöltése

Mivel az adatok mennyisége nagyon nagy lenne, albumonként csak pár darab dallal fogom feltölteni az adatbázist.

```

begin
    INSERT INTO Kiado VALUES(1, 'PRMD Music', '03/01/2013', 'Svédország', 'Stockholm');
    INSERT INTO Kiado VALUES(2, 'Universal Music Group', '09/01/1934', 'Egyesült Államok',
'Santa Monica');
    INSERT INTO Kiado VALUES(3, 'Sony Music Entertainment', '03/01/1968', 'Japán', 'Tokió');
    INSERT INTO Kiado VALUES(4, 'Odyssey Music Network', '01/01/2014', 'Németország',
'Berlin');
end;

begin
    INSERT INTO Szerzo VALUES(1, 'Avicii', '05/09/2008');
    INSERT INTO Szerzo VALUES(2, 'supercell', '12/07/2007');
    INSERT INTO Szerzo VALUES(3, 'kz', '09/22/2007');
    INSERT INTO Szerzo VALUES(4, 'Apocalyptica', '01/01/1993');
end;

begin
    INSERT INTO Mufaj VALUES(1, 'dance');
    INSERT INTO Mufaj VALUES(2, 'folktronica');
    INSERT INTO Mufaj VALUES(3, 'soul');
    INSERT INTO Mufaj VALUES(4, 'pop');
    INSERT INTO Mufaj VALUES(5, 'house');
    INSERT INTO Mufaj VALUES(6, 'j-pop');
    INSERT INTO Mufaj VALUES(7, 'rock');
    INSERT INTO Mufaj VALUES(8, 'EDM');
    INSERT INTO Mufaj VALUES(9, 'metal');
    INSERT INTO Mufaj VALUES(10, 'symphonic');
end;

```

```

begin
    INSERT INTO Enekes VALUES(1, 'Aloe Blacc', 'férfi', '01/07/1979');
    INSERT INTO Enekes VALUES(2, 'Audra Mae', 'nő', '02/20/1984');
    INSERT INTO Enekes VALUES(3, 'Simon Aldred', 'férfi', '09/01/1975');
    INSERT INTO Enekes VALUES(4, 'Hatsune Miku', 'nő', '08/31/2007');
    INSERT INTO Enekes VALUES(5, 'Franky Perez', 'férfi', '02/24/1976');
end;

begin
    INSERT INTO Album VALUES(1, 'True', '09/13/2013', 1);
    INSERT INTO Album VALUES(2, 'Stories', '10/02/2015', 1);
    INSERT INTO Album VALUES(3, 'TIM', '06/06/2019', 2);
    INSERT INTO Album VALUES(4, 'supercell', '08/16/2008', null);
    INSERT INTO Album VALUES(5, 'こっち向いて Baby / Yellow', '07/14/2010', 3);
    INSERT INTO Album VALUES(6, 'Shadowmaker', '04/17/2015', 4);
end;

begin
    INSERT INTO Dal VALUES(1, 'Wake Me Up', TO_DSINTERVAL('0 00:04:07'), 1, 1);
    INSERT INTO Dal VALUES(2, 'Addicted to You', TO_DSINTERVAL('0 00:02:28'), 1, 1);
    INSERT INTO Dal VALUES(3, 'Waiting for Love', TO_DSINTERVAL('0 00:03:50'), 1, 2);
    INSERT INTO Dal VALUES(4, 'SOS', TO_DSINTERVAL('0 00:02:37'), 1, 3);
    INSERT INTO Dal VALUES(5, 'ワールドイズマイン', TO_DSINTERVAL('0 00:04:14'), 2, 3);
    INSERT INTO Dal VALUES(6, 'こっち向いて Baby', TO_DSINTERVAL('0 00:03:30'), 2, 4);
    INSERT INTO Dal VALUES(7, 'ODDS&ENDS', TO_DSINTERVAL('0 00:05:37'), 2, null);
    INSERT INTO Dal VALUES(8, 'Yellow', TO_DSINTERVAL('0 00:03:20'), 3, 5);
    INSERT INTO Dal VALUES(9, 'Shadowmaker', TO_DSINTERVAL('0 00:07:36'), 4, 6);
    INSERT INTO Dal VALUES(10, 'Sea Song (You Waded Out)', TO_DSINTERVAL('0 00:04:54'), 4, 6);
    INSERT INTO Dal VALUES(11, 'Till Death Do Us Part', TO_DSINTERVAL('0 00:07:51'), 4, 6);
end;

begin
    INSERT INTO Enek VALUES(1, 'angol', 1);
    INSERT INTO Enek VALUES(2, 'angol', 2);
    INSERT INTO Enek VALUES(3, 'angol', 3);
    INSERT INTO Enek VALUES(4, 'angol', 1);
    INSERT INTO Enek VALUES(5, 'japán', 4);
    INSERT INTO Enek VALUES(6, 'japán', 4);
    INSERT INTO Enek VALUES(7, 'japán', 4);
    INSERT INTO Enek VALUES(8, 'japán', 4);
    INSERT INTO Enek VALUES(9, 'angol', 5);
    INSERT INTO Enek VALUES(10, 'angol', 5);
end;

begin
    INSERT INTO Mufaj_Dal VALUES(1, 1);
    INSERT INTO Mufaj_Dal VALUES(2, 1);

```

```

INSERT INTO Mufaj_Dal VALUES (3, 2);
INSERT INTO Mufaj_Dal VALUES (2, 2);
INSERT INTO Mufaj_Dal VALUES (4, 2);
INSERT INTO Mufaj_Dal VALUES (4, 3);
INSERT INTO Mufaj_Dal VALUES (5, 3);
INSERT INTO Mufaj_Dal VALUES (8, 4);
INSERT INTO Mufaj_Dal VALUES (6, 5);
INSERT INTO Mufaj_Dal VALUES (7, 5);
INSERT INTO Mufaj_Dal VALUES (6, 6);
INSERT INTO Mufaj_Dal VALUES (7, 6);
INSERT INTO Mufaj_Dal VALUES (6, 7);
INSERT INTO Mufaj_Dal VALUES (7, 7);
INSERT INTO Mufaj_Dal VALUES (6, 8);
INSERT INTO Mufaj_Dal VALUES (8, 8);
INSERT INTO Mufaj_Dal VALUES (1, 8);
INSERT INTO Mufaj_Dal VALUES (9, 9);
INSERT INTO Mufaj_Dal VALUES (10, 9);
INSERT INTO Mufaj_Dal VALUES (9, 10);
INSERT INTO Mufaj_Dal VALUES (10, 10);
INSERT INTO Mufaj_Dal VALUES (9, 11);
INSERT INTO Mufaj_Dal VALUES (10, 11);
end;

```

Lekérdezések

1. Albumon nem szereplő dalok adatainak listázása

```
SELECT * FROM Dal
```

```
WHERE albumid IS NULL;
```

$\sigma_{albumid = NULL} dal$

1

SELECT * FROM Dal

2

WHERE albumid IS NULL;

Results

Explain

Describe

Saved SQL

History

DALID	CIM	HOSSZ	SZERZOID	ALBUMID
7	ODDS&ENDS	[unsupported data type]	2	-

1 rows returned in 0.00 seconds

Download

2. Függetlenül kiadott albumok dalainak és szerzőinek listázása

```
SELECT Dal.cim, Album.cim FROM Dal
```

```
JOIN Album USING(albumid)
```

```
WHERE kiadoid IS NULL;
```

$\pi \text{ dal} . \text{cim}, \text{album} . \text{cim}$

$\sigma \text{ kiadoid} = \text{NULL} (\text{dal} \square \text{albumid album})$

```

1 SELECT Dal.cim, Album.cim FROM Dal
2 JOIN Album USING(albumid)
3 WHERE kiadoid IS NULL;

```

CIM	CIM
ワールドイズマイン	supercell

1 rows returned in 0.00 seconds [Download](#)

3. 2010 eleje és 2015 vége között kiadott albumok, és az azokat kiadó lemezkiadók listázása

```

SELECT DISTINCT Album.cim, Szerzo.nev, Kiado.nev, Album.kiadas FROM Album
JOIN Dal USING(albumid)
JOIN Szerzo USING(szerzoid)
JOIN Kiado USING(kiadoid)
WHERE kiadas BETWEEN '01/01/2010' AND '12/31/2015'

```

δ

$\pi \text{ album} . \text{cim}, \text{szerzo} . \text{nev}, \text{kiado} . \text{nev}, \text{album} . \text{kiadas}$

$\sigma "01/01/2010" \leq \text{kiadas} \text{ AND } \text{kiadas} \leq "12/31/2015" (\text{album} \square \text{albumid dal} \square \text{szerzoid szerzo} \square \text{kiadoid kiado})$

```

1 SELECT DISTINCT Album.cim, Szerzo.nev, Kiado.nev, Album.kiadas FROM Album
2 JOIN Dal USING(albumid)
3 JOIN Szerzo USING(szerzoid)
4 JOIN Kiado USING(kiadoid)
5 WHERE kiadas BETWEEN '01/01/2010' AND '12/31/2015'
6

```

CIM	NEV	NEV	KIADAS
こっち向いて Baby / Yellow	supercell	Sony Music Entertainment	07/14/2010
Stories	Avicii	PRMD Music	10/02/2015
True	Avicii	PRMD Music	09/13/2013
Shadowmaker	Apocalyptica	Odyssey Music Network	04/17/2015
こっち向いて Baby / Yellow	kz	Sony Music Entertainment	07/14/2010

5 rows returned in 0.01 seconds [Download](#)

4. Lemezkiadók csökkenő sorrendbe rendezése kiadott albumok száma alapján

```
SELECT nev, COUNT(*) AS albumSzam
FROM Kiado
JOIN Album USING(kiadoid)
GROUP BY nev
ORDER BY albumSzam DESC;
```

τ *albumszam* ↓

π *nev*, *COUNT*(*) → *albumszam*

γ *nev*, *COUNT*(*) (*kiado* ⋈ *kiadoid album*)

1	SELECT nev, COUNT(*) AS albumSzam
2	FROM Kiado
3	JOIN Album USING(kiadoid)
4	GROUP BY nev
5	ORDER BY albumSzam DESC;
6	

Results	Explain	Describe	Saved SQL	History
NEV		ALBUMSZAM		
PRMD Music		2		
Odyssey Music Network		1		
Universal Music Group		1		
Sony Music Entertainment		1		

5. Különböző nyelveken különböző műfajú dalok száma

```
SELECT nyelv, megnevezes, COUNT(*) AS mennyiseg FROM Mufaj
JOIN Mufaj_Dal USING(mufajid)
JOIN Dal USING(dalid)
JOIN Enek USING(dalid)
GROUP BY nyelv, megnevezes
ORDER BY mennyiseg DESC;
```

τ *mennyiseg* ↓

π *nyelv*, *megnevezes*, *COUNT*(*) → *mennyiseg*

γ *nyelv*, *megnevezes*, *COUNT*(*) (*mufaj* ⋈ *mufajid mufaj_dal* ⋈ *dalid dal* ⋈ *dalid enek*)

1	SELECT nyelv, megnevezes, COUNT(*) AS mennyiseg FROM Mufaj
2	JOIN Mufaj_Dal USING(mufajid)
3	JOIN Dal USING(dalid)
4	JOIN Enek USING(dalid)
5	GROUP BY nyelv, megnevezes
6	ORDER BY mennyiseg DESC;

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

NYELV	MEGNEVEZES	MENNYISEG
japán	j-pop	4
japán	rock	3
angol	metal	2
angol	folktronica	2
angol	pop	2
angol	symphonic	2
angol	EDM	1
angol	house	1
japán	EDM	1
angol	dance	1
angol	soul	1
japán	dance	1

6. Három és négy perc közötti hosszúságú dalok szerzőinek és címének kiválasztása

```
SELECT nev, cim FROM Dal JOIN Szerzo USING(szerzoid)
WHERE hossz BETWEEN INTERVAL '0 00:03:00' DAY TO SECOND
AND INTERVAL '0 00:05:00' DAY TO SECOND;
```

π *nev, cím*

σ "0 00:03:00" <= hossz AND hossz <= "0 00:05:00" (dal \square szerzoid szerzo)

1	SELECT nev, cim FROM Dal JOIN Szerzo USING(szerzoid)
2	WHERE hossz BETWEEN INTERVAL '0 00:03:00' DAY TO SECOND
3	AND INTERVAL '0 00:05:00' DAY TO SECOND;
4	

Results	Explain	Describe	Saved SQL	History
NEV		CIM		
Avicii		Wake Me Up		
Avicii		Waiting for Love		
supercell		ワールドイズマイン		
supercell		こっち向いて Baby		
kz		Yellow		
Apocalyptica		Sea Song (You Waded Out)		

7. Albumok csökkenő sorba rendezése a dalok összesített hossza szerint

```
SELECT Album.cim,
       NUMTODSINTERVAL(SUM(EXTRACT(MINUTE FROM hossz)), 'MINUTE') +
       NUMTODSINTERVAL(SUM(EXTRACT(SECOND FROM hossz)), 'SECOND')
AS osszHossz
FROM Album
JOIN Dal USING(albumid)
GROUP BY Album.cim
```

τ *osszhossz* ↓

π *album . cim*, *SUM(MINUTE) + SUM(SECOND)* → *osszhossz*

γ *cim*, (*album* □ *albumid dal*)

1	SELECT Album.cim,
2	NUMTODSINTERVAL(SUM(EXTRACT(MINUTE FROM hossz)), 'MINUTE') +
3	NUMTODSINTERVAL(SUM(EXTRACT(SECOND FROM hossz)), 'SECOND')
4	AS osszHossz
5	FROM Album
6	JOIN Dal USING(albumid)
7	GROUP BY Album.cim
8	ORDER BY osszHossz DESC;

Results	Explain	Describe	Saved SQL	History
CIM		OSSZHOSSZ		
Shadowmaker		+0000000000 00:20:21.0000000000		
こっち向いて Baby / Yellow		+0000000000 00:06:50.0000000000		
True		+0000000000 00:06:35.0000000000		
supercell		+0000000000 00:04:14.0000000000		
Stories		+0000000000 00:03:50.0000000000		
TIM		+0000000000 00:02:37.0000000000		

ORDER BY osszHossz DESC;

8. Énekesek csökkenő sorba rendezése dalok száma szerint

```
SELECT nev, COUNT(*) AS dalszam FROM Enekes
JOIN Enek USING(enekesid)
JOIN Dal USING(dalid)
GROUP BY nev
ORDER BY dalszam DESC;
```

τ *dalszam* ↓

π *nev*, *COUNT*(*) → *dalszam*

γ *nev*, *COUNT*(*) (*enekes* □ *enekesid* *enek* □ *dalid* *dal*)

1	SELECT nev, COUNT(*) AS dalszam FROM Enekes
2	JOIN Enek USING(enekesid)
3	JOIN Dal USING(dalid)
4	GROUP BY nev
5	ORDER BY dalszam DESC;

Results	Explain	Describe	Saved SQL	History
NEV		DALSZAM		
Hatsune Miku		4		
Aloe Blacc		2		
Franky Perez		2		
Simon Aldred		1		
Audra Mae		1		

5 rows returned in 0.01 seconds [Download](#)

9. Avicii azon dalainak kiválasztása, amiben női vokál van

```
SELECT cim FROM (
  SELECT Dal.cim, nem FROM Dal
  JOIN Album USING(albumid)
  JOIN Szerzo USING(szerzoid)
  JOIN Enek USING(dalid)
  JOIN Enekes USING(enekesid)
  WHERE Szerzo.nev LIKE 'Avicii'
) WHERE nem LIKE 'nő';
```

π cim

σ nem LIKE "no" (π dal . cim, nem

σ szerzo . nev LIKE "Avicii" (dal \square albumid album \square szerzoid szerzo \square dalid enek
 \square enekesid enekes))

1	SELECT cim FROM (
2	SELECT Dal.cim, nem FROM Dal
3	JOIN Album USING(albumid)
4	JOIN Szerzo USING(szerzoid)
5	JOIN Enek USING(dalid)
6	JOIN Enekes USING(enekesid)
7	WHERE Szerzo.nev LIKE 'Avicii'
8) WHERE nem LIKE 'nő';

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

CIM
Addicted to You

1 rows returned in 0.02 seconds [Download](#)

10. Azon albumok kiválasztása, amelyekben kettő vagy több szerző zenéi szerepelnek

```
SELECT Album.cim, COUNT(DISTINCT nev) AS kozremukodok FROM Album
JOIN Dal USING(albumid)
JOIN Szerzo USING(szerzoid)
GROUP BY Album.cim
HAVING COUNT(DISTINCT nev) >= 2;
```

$\pi_{album.cim, COUNT(\Delta nev)} \rightarrow kozremukodok$

$\sigma_{COUNT(\Delta nev) \geq 2}$

$\gamma_{cim, COUNT(\Delta nev)} (album \bowtie albumid \text{ dal } \bowtie szerzoid szerzo)$

1	SELECT Album.cim, COUNT(DISTINCT nev) AS kozremukodok FROM Album
2	JOIN Dal USING(albumid)
3	JOIN Szerzo USING(szerzoid)
4	GROUP BY Album.cim
5	HAVING COUNT(DISTINCT nev) >= 2;

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

CIM	KOZREMUKODOK
こっち向いて Baby / Yellow	2