# Összegzések

## A. Előkészületek

Ebben a leckében is a *filmek* adatbázist fogjuk használni.

1. Indítsd el az XAMPP programot, futtasd a web és a MySQL szervereket, majd indítsd el a PHPMyAdmin programot!
2. Kezdj egy új adatbázist *filmek* néven!
3. Importáld a táblát és az adatokat a *filmek.sql* fájlból!

## B. Összegzések

Az összegzések lényege, hogy nem kilistázzuk a sorokat, hanem valamilyen műveletet végzünk rajtuk (megszámlálás, összegzés, …) és az eredményt íratjuk ki egy sorban.

Írasd ki a **count()** függvény segítségével, hogy **hány darab** film szerepel a táblában!

SELECT count(\*) AS Darab FROM top10;

A \* azt jelzi, hogy nem egy adott mezőben lévő adatokat számlálunk, hanem teljes sorokat.

Írasd ki a **sum()** függvény segítségével, hogy hány néző látta **összesen** a top10 táblában szereplő filmeket! Írd ide az eredményt:

SELECT sum(nezo) AS Nézőszám FROM top10;

Írassuk ki az **avg()** függvény használatával, hogy a filmek eddig **átlagosan** mennyi bevételt hoztak! Írd ide az eredményt:

SELECT avg(bevetel) AS Átlag FROM top10;

Mennyi volt a **legnagyobb**, illetve a **legkisebb** bevétel?

SELECT max(bevetel) AS Legnagyobb, min(bevetel) AS Legkisebb FROM top10;

Az összegző függvények használatakor is meg lehet adni feltételeket a sorokra a WHERE kulcsszóval. Ilyenkor csak azokkal a sorokkal végzi el az összegzést, amelyekre igaz a feltétel. Például írasd ki, hogy hányan nézték meg az animációs filmeket!

SELECT sum(nezo) AS Néző FROM top10 WHERE mufaj='animációs';

Az eredmény:

## C. Csoportosítás

Ha nem csak egy műfajra szeretnél összegezni, hanem mindre, akkor csoportosítást kell alkalmaznod a **GROUP BY** kulcsszóval.

A SELECT és a FROM között csak összegzéseket és azokat a mezőket adhatod meg, amelyek szerint csoportosítasz!

Listázd ki műfajonként a nézőszámok összegét!

SELECT mufaj, sum(nezo) AS Néző FROM top10 GROUP BY mufaj;

Ez a parancs annyi csoportot képez, ahány féle műfaj szerepel a táblában, és mindegyik csoportra összeadja a nézőszámokat. Figyeld meg, hogy a műfajok ABC sorrendbe kerülnek!

Ha az eredményt más sorrendben szeretnéd, használhatod az **ORDER BY** kulcsszót. Rendezd a nézőszám szerint csökkenő sorrendbe az előző listát!

SELECT mufaj, sum(nezo) AS Néző FROM top10 GROUP BY mufaj ORDER BY Néző DESC;

Ha csak azokat a műfajokat szeretnéd kiíratni, amelyeknél az összes nézőszám 20 000 felett volt, akkor a csoportokra kell feltételt megadnod a **HAVING** kulcsszóval:

SELECT mufaj,sum(nezo) AS Néző   
FROM top10   
GROUP BY mufaj   
HAVING Néző>20000  
ORDER BY Néző DESC;

Az áttekinthetőség érdekében több sorba írtuk a parancsot.

Figyeld meg a kulcsszavak sorrendjét! A HAVING mindig a GROUP BY után következik.

Vigyázat! Ne keverd össze a WHERE és a HAVING kulcsszavakat!

* A WHERE a FROM után következik, és csak azok *rekordok* vesznek részt a műveletben, amelyekre igaz az utána megadott feltétel.
* A HAVING a GROUP BY után következik, és a parancs által létrehozott *csoportokat* szűri a megadott feltétel alapján.

Mentsd a parancsokat *megoldas.sql* néven!

## D. Feladatok

Végezd el a következő feladatokat az adatbázisban!

1. Írasd ki, hogy összesen mennyi bevételt hoztak a 100 percnél hosszabb filmek!  
   Parancs:
2. Szeretnénk kilistázni, hogy a (csak) amerikai filmek műfajonként mennyi bevételt hoztak. A listába csak azok a műfajok szerepeljenek, amelyek összbevétele legalább 100 000 000, és a bevétel szerint csökkenő sorrendben! Egészítsd ki az alábbi parancsot:   
   SELECT \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   FROM \_\_\_\_\_  
   WHERE \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   GROUP BY \_\_\_\_\_  
   HAVING \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   ORDER BY \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;