Labor dokumentáció – Adatbázisok Laboratórium

3. mérés: SQL2

|  |  |
| --- | --- |
| Név: | **Buga Péter** |
| Neptun kód: | **G50RDF** |
| Feladat kódja: | **33-Video** |
| Mérésvezető neve: | **Nagy Áron Artúr** |
| Mérés időpontja: | **2023-10-27 8:15** |
| Mérés helyszíne: | **HSZK J** |
| Megoldott feladatok: | **2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8** |
| Elérhető pontszám (plusz pontok nélkül): | **14p** |

# Mérési feladatok megoldása

## feladat: Video tábla összes oszlopának lekérdezése

### A megoldáshoz használt SQL utasítás

SELECT

\*

FROM

VIDEOS;

### Magyarázat

*A feladatban az szerepel, hogy ne használjuk az oszlopok neveit, ezt a \*-gal lehet megoldani, mivel ez kiválasztja az adott tábla összes oszlopát.*

A képen szöveg, képernyőkép, szám, Betűtípus látható

Automatikusan generált leírás

## feladat: Blu-ray típusú filmek lekérdezése

### A megoldáshoz használt SQL utasítás

SELECT

VIDEO\_CODE,

TITLE,

DIRECTOR

FROM

VIDEOS

WHERE

TYPE = 'B';

### Magyarázat

*A „SELECT” -tel kiválasztom a kívánt oszlopokat, a „FROM”-mal pedig hogy melyik táblából, végül a „WHERE”-nél megadom a feltételt hogy a típus ’B’ legyen ami reprezentálja a Blu-ray-t.*

*A képen szöveg, elektronika, képernyőkép, szoftver látható

Automatikusan generált leírás*

## feladat: 2000 jan 1 előtti filmek

### A megoldáshoz használt SQL utasítás

SELECT

VIDEO\_CODE,

TITLE,

RENTAL\_DATE,

RETURN\_DATE

FROM

VIDEOS,

LOANS

WHERE

VIDEOS.VIDEO\_ID = LOANS.VIDEO\_ID

AND RELEASE\_DATE < '01-JAN-2000'

ORDER BY

RENTAL\_DATE DESC, TITLE;

### Magyarázat

*A „SELECT”-tel kiválasztottam a megadott attribútumokat, a „FROM”-nál megadtam hogy melyik táblákban vannak ezek. A „WHERE”-nél az első feltétel biztosítja hogy csak olyan filmek kerülnek be a lekérdezésbe amik már szerepelnek a LOANS táblában, a második pedig hogy 2000.01.01 előtti megjelenésű az adott film. A rendezést az „ORDER BY” segítségével lehet megcsinálni, előszőr a kikölcsönzés napja szerint csökkenően, ezt a „DESC” biztosítja, majd a film címe szerint rendeztem.*

*A képen szöveg, képernyőkép, szám, menü látható

Automatikusan generált leírás*

## feladat: Kikölcsönzött DVD típusú filmek lekérdezése

### A megoldáshoz használt SQL utasítás

SELECT

VIDEO\_CODE,

TITLE,

RENTAL\_DATE

FROM

VIDEOS,

LOANS

WHERE

VIDEOS.VIDEO\_ID = LOANS.VIDEO\_ID

AND LOANS.RETURN\_DATE IS NULL

AND VIDEOS.TYPE = 'D'

ORDER BY

RENTAL\_DATE DESC, TITLE;

### Magyarázat

*Az eddigiekhez hasonló módon leírtakkal kiválasztottam a megfelelő attribútumokat. Az első feltétel biztosítja, hogy kikölcsönzött filmek kerülnek a lekérdezésbe. A második feltétel miatt csak olyan rekordok fognak szerepelni a lekérdezésben, ahol a „RETURN\_DATE” null, tehát még nem hozták vissza a filmet. A harmadik feltétel szerint csak DVD típusú filmek kerülnek be a lekérdezésbe. Végül rendeztem kikölcsönzési idő szerint csökkenően és ezen belül a film neve szerint*

*A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás*

## feladat: Kikölcsönzések lekérdezése

### A megoldáshoz használt SQL utasítás

SELECT

VIDEO\_CODE,

TITLE,

RENTAL\_DATE,

RETURN\_DATE

FROM

VIDEOS,

LOANS

WHERE

VIDEOS.VIDEO\_ID = LOANS.VIDEO\_ID(+)

ORDER BY

VIDEOS.VIDEO\_CODE, LOANS.RENTAL\_DATE DESC;

### Magyarázat

*A már megszokott módok kiválasztottam a megfelelő oszlopokat és táblákat. A feltételnél a „(+)”-jelölés biztosítja hogy ha nincs a „VIDEO.VIDEO\_ID”-nek megfelelő „VIDEO\_ID” attribútumú rekord a LOANS táblában akkor is szerepelni fog a lekérdezésben úgy hogy a LOANS táblából vett attribútumai null értéket kapnak. A végén rendeztem először a filmek kódja szerint, majd ezen belül az elvitel dátuma szerint csökkenően.*

*A képen szöveg, képernyőkép, szám, menü látható

Automatikusan generált leírás*

## feladat: Filmek kikölcsönzésének darabszáma

### A megoldáshoz használt SQL utasítás

SELECT

VIDEOS.VIDEO\_CODE,

VIDEOS.TITLE,

(

SELECT

NVL(COUNT(LOAN\_ID),0)

FROM

LOANS

WHERE

LOANS.VIDEO\_ID = VIDEOS.VIDEO\_ID

) AS TIMES

FROM

VIDEOS

ORDER BY

VIDEO\_CODE;

### Magyarázat

*A film kódja és a címe kiválasztása után beágyazott lekérdezésben megszámoltam a „COUNT” függvény segítségével, hogy hány rekord van a LOANS táblában egy-egy filmhez, ha nincs egy film se akkor 0-t rendel hozzá. Majd ezt elneveztem „times”-nak. Végül a külső lekérdezésben rendeztem a rekordokat a film kódja szerint.*

*A képen szöveg, képernyőkép, szám, Betűtípus látható

Automatikusan generált leírás*

## feladat: Legfeljebb kétszer kikölcsönzött filmek lekérdezése

### A megoldáshoz használt SQL utasítás

SELECT

VIDEOS.VIDEO\_CODE,

VIDEOS.TITLE,

(

SELECT

NVL(COUNT(LOAN\_ID),0)

FROM

LOANS

WHERE

LOANS.VIDEO\_ID = VIDEOS.VIDEO\_ID

) AS TIMES

FROM

VIDEOS

WHERE

(

SELECT

NVL(COUNT(LOAN\_ID),0)

FROM

LOANS

WHERE

LOANS.VIDEO\_ID = VIDEOS.VIDEO\_ID

) < 2

ORDER BY

VIDEO\_CODE;

### Magyarázat

*A feladat megoldása szinte ugyan az mint az előzőé, azzal a kivétellel hogy most a „WHERE”-nél is meg kellett csinálni ugyan azt a beágyazott lekérdezést hogy tudjunk rá feltételt rakni, ez esetben azt hogy ne legyen nagyobb 2-nél a kölcsönzések száma.*

*A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás*

## feladat: DVD-n lévő filmek lekérdezése

### A megoldáshoz használt SQL utasítás

SELECT

VIDEO\_CODE,

TITLE,

(

SELECT

MAX(RENTAL\_DATE)

FROM

LOANS

WHERE

LOANS.VIDEO\_ID = VIDEOS.VIDEO\_ID

AND RETURN\_DATE IS NULL

) AS RENTAL\_DATE

FROM

VIDEOS

WHERE

TYPE = 'D'

ORDER BY

TITLE;

### Magyarázat

*A film kódja és címe kiválasztása után egy beágyazott lekérdezésben megkerestem a legújabb kölcsönzést, ahol a visszahozatalnak nincs még értéke. Az jelenleg nem kikölcsönzött filmeknél a belső lekérdezés nem ad vissza eredményt így csak ott fog a „rental\_date”-nek értéke lenni ahol még nem hozták vissza a filmet. A külső lekérdezésben a „WHERE”-nél adtam meg a feltételt arra hogy DVD-nek kell lennie, majd sorba raktam a rekordokat a filmek címe szerint.*

*A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírás*