



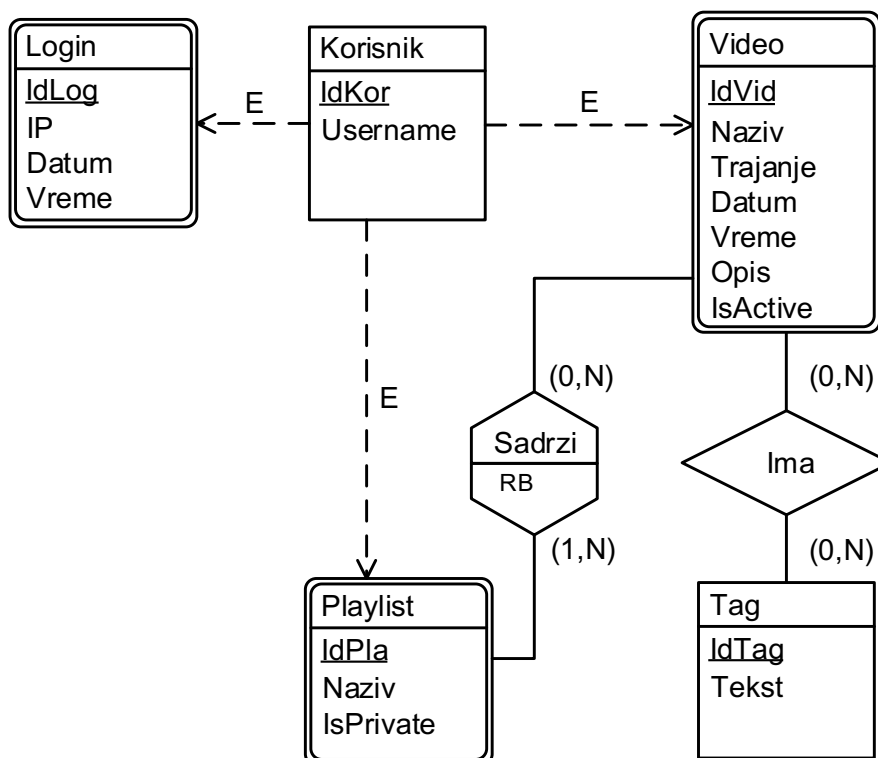
## Базе података 1

(13C112БП1, 13E113БП1, 13E113БП)  
- јулски испитни рок -

### Група А

Посматра се део система веб портала за дељење видео садржаја. Овај портал има регистроване кориснике за које се прати корисничко име (*Username*), као и записник када и са које IP адресе се корисник пријавио на портал. Корисник може да објави видео садржај и том приликом се памти назив, трајање садржаја у секундама, датум и време поставке, као и опис који није обавезан. Корисник може да обрише видео садржај, али се том приликом не брише видео садржај заувек, већ се поставља индикатор да видео није више активан (1 – активан, 0 – неактиван). Корисник може да формира листу видео садржаја – *Playlist*. Листа има назив, као и индикатор да ли је листа приватна или јавна. Листа мора да има бар један видео и нема ограничење о максималном броју видео записа. За сваки видео на листи се прати под којим редним бројем се налази (редни бројеви су сукцесивни). Ради лакше претраге видео може да има листу тагова. Тагови могу да буду заједнички за више видеа. У систему може да постоји таг који се не односи ни на један видео. У систему се датуми памте као целобројне величине у формату ууууммдд (уууу – година, мм – месец, дд – дан), док се времена памте као низ од тачно 12 карактера у формату hh:mm:ss.zzz (hh – сат, мм – минут, ss – секунд, zzz – милисекунде). Сматрати да је комбинација датума и времена јединствена у табелама где постоје колоне за датум и време.

У наставку је дата релациона шема посматраног дела базе податка.



#### Korisnik (IdKor, Username)

IdKor	- цео број, идентификује корисника, аутоматско додељивање наредног идентификатора
Username	- низ до 50 знакова, обавезно

#### Login (IdLog, IP, Datum, Vreme, IdKor)

IdLog	- цео број, идентификује пријаву корисника, аутоматско додељивање наредног идентификатора
IP	- низ до 15 знакова, обавезно, IP адреса у формату (xxx.yyy.zzz.www; xxx, ууу, зzz, www – бројеви 0-255)
Datum	- цео број, у формату (ууууmmdd), јединствено у комбинацији са пољем Vreme, обавезно
Vreme	- низ од 12 знакова, у формату (hh:mm:ss.zzz), јединствено у комбинацији са пољем Datum, обавезно
IdKor	- страни кључ (табела Korisnik), обавезно

#### Video (IdVid, Naziv, Trajanje, Datum, Vreme, Opis, IsActive, IdKor)

IdVid	- цео број, идентификује видео, аутоматско додељивање наредног идентификатора
Naziv	- низ до 200 знакова, обавезно
Trajanje	- цео број, обавезно, вредност већа од 0
Datum	- цео број, у формату (ууууmmdd), јединствено у комбинацији са пољем Vreme, обавезно
Vreme	- низ од 12 знакова, у формату (hh:mm:ss.zzz), јединствено у комбинацији са пољем Datum, обавезно
Opis	- низ до 200 знакова
IsActive	- цео број, обавезно, могуће вредности 0 или 1
IdKor	- страни кључ (табела Korisnik), обавезно

#### Playlist (IdPla, Naziv, IsPrivate, IdKor)

IdPla	- цео број, идентификује видео листу, аутоматско додељивање наредног идентификатора
Naziv	- низ до 200 знакова, обавезно
IsPrivate	- цео број, обавезно, могуће вредности 0 или 1
IdKor	- страни кључ (табела Korisnik), обавезно

#### Sadrzi (IdPla, IdVid, RB)

IdPla	- страни кључ (табела Playlist), део примарног кључа, обавезно
IdVid	- страни кључ (табела Video), део примарног кључа, обавезно
RB	- цео број, обавезно, вредност већа од 0, сукцесивно се додељује видео садржају на видео листи

#### Tag (IdTag, Tekst)

IdTag	- цео број, идентификује таг, аутоматско додељивање наредног идентификатора
Tekst	- низ до 50 знакова, обавезно

#### Ima (IdVid, IdTag)

IdVid	- страни кључ (табела Video), део примарног кључа, обавезно
IdTag	- страни кључ (табела Tag), део примарног кључа, обавезно

Задатак 1 [4 поена]

Потребно је направити SQL упит који брише из базе све приватне видео листе корисника који се нису улоговали од почетка 2020 године. Након претходног брисања потребно је исписати све преостале видео листе. Сортирати резултат по IdPla опадајуће.

Резултат дати у форми: IdPla, Naziv, IsPrivate, IdKor

У Сactus-у користити таб: Zadatak 1

**Није дозвољено коришћење погледа.**

---

```
DELETE FROM Playlist
WHERE IsPrivate=1 AND IdKor NOT IN (
                                SELECT IdKor
                                FROM Login
                                WHERE Datum / 10000 >= 2020
                                );
SELECT * FROM Playlist ORDER BY IdPla DESC;
```

---

Задатак 2 [4 поена]

Потребно је написати SQL упит који исписује све тагове видео листе са називом “Моја muzika” (може се сматрати да постоји јединствена видео листа са овим називом). Тагови видео листе представљају скуп свих тагова активних видео садржаја који се садрже у листи. Сортирати резултат по IdTag растуће. Тагове је потребно исписати само једном, чак и ако се појављују у више видео садржаја.

Резултат дати у форми: IdTag, Tekst

У Сactus-у користити таб: Zadatak 2

**Није дозвољено коришћење погледа.**

---

```
SELECT DISTINCT IdTag, Tekst
FROM Playlist JOIN Sadrzi USING(IdPla) JOIN Video USING (IdVid) JOIN Ima USING(IdVid) JOIN
Tag USING(IdTag)
WHERE Playlist.Naziv = 'Moja muzika' AND IsActive=1
ORDER BY IdTag
```

---

### Задатак 3 [4 поена]

Потребно је направити SQL упит који исписује резултат претраге активних видео садржаја који у називу имају реч “SRB” (реч може да представља комплетан назив или да буде на почетку, у средини или на крају назива). У резултату треба исписати и који је корисник поставио видео. Трајање видео записа треба исписати у формату mm:ss где mm представља број минута, а ss број секунди трајања видео записа. Сортирати резултат прво по датуму постављања растуће, па онда по времену постављања растуће.

Резултат дати у форми: IdVid, Naziv, Trajanje, Username

У Sactus-у користити таб: Zadatak 3

**Није дозвољено коришћење погледа.**

---

```
SELECT IdVid, Naziv, (Trajanje/60) || ':' || (Trajanje % 60) AS Trajanje, Username
FROM Video NATURAL JOIN Korisnik
WHERE IsActive=1 AND ( Naziv = 'SRB' OR Naziv LIKE 'SRB %' OR Naziv LIKE '% SRB %' OR
Naziv LIKE '% SRB' )
ORDER BY Datum, Vreme
```

---

### Задатак 4 [4 поена]

Потребно је направити SQL скрипту која као резултат враћа колико је (различитих) корисника приступило порталу искључиво са мреже 10.20.30.0/24 и ни са једне друге.

Резултат дати у форми: Broj korisnika

У Sactus-у користити таб: Zadatak 4

**Није дозвољено коришћење погледа.**

---

```
SELECT COUNT(DISTINCT IdKor) AS "Broj korisnika"
FROM Login l1
WHERE IP LIKE '10.20.30.%'
      AND NOT EXISTS (
          SELECT * FROM Login l2
          WHERE l2.IdKor = l1.IdKor AND IP NOT LIKE '10.20.30.%'
      )
```

---

### Задатак 5 [5 поена]

Потребно је направити SQL скрипту која ако постоји табела **Video** избацује табелу **Video** из шеме, а затим формира нову табелу **Video** која треба да има одговарајућу структуру и ограничења. Није потребно да се реализује ограничење да су датум и време у исправном формату.

У Cactus-у користити таб: Zadatak 5

---

```
DROP TABLE IF EXISTS Video;
CREATE TABLE Video (
    IdVid          INTEGER PRIMARY KEY AUTOINCREMENT,
    Naziv          VARCHAR (200) NOT NULL,
    Trajanje       INTEGER CHECK (Trajanje > 0) NOT NULL,
    Datum          INTEGER NOT NULL,
    Vreme          CHAR (12) NOT NULL,
    Opis          VARCHAR (200),
    IsActive       INTEGER CHECK (IsActive = 0 OR IsActive = 1) NOT NULL,
    IdKor          INTEGER NOT NULL REFERENCES Korisnik (IdKor),
    UNIQUE (Datum, Vreme)
);
```

---

### Задатак 6 [5 поена]

Потребно је направити SQL упит који исписује све видео листе код којих су сви видео записи активни и имају опис. У резултату приказати и корисника који је направио видео листу. Сортирати резултат по IdPla опадајуће.

Резултат дати у форми: IdPla, Naziv, Username

У Cactus-у користити таб: Zadatak 6

**Није дозвољено коришћење погледа.**

---

```
SELECT IdPla, Naziv, Username
FROM Playlist p NATURAL JOIN Korisnik
WHERE (SELECT COUNT (*) FROM Video NATURAL JOIN Sadrzi WHERE IdPla = p.IdPla)
      =
      (SELECT COUNT (*) FROM Video NATURAL JOIN Sadrzi WHERE IdPla = p.IdPla
       AND IsActive = 1 AND Opis IS NOT NULL)
ORDER BY IdPla DESC
```

---

### Задатак 7 [5 поена]

Потребно је направити SQL упит који приказује активан видео садржај (или садржаје) који се налази на највише јавних видео листа. Ако постоји више таквих видеа, исписати их све. Сортирати резултат по IdVid опадајуће.

Резултат дати у форми: IdVid, Naziv

У Сactus-у користити таб: Zadatak 7

**Није дозвољено коришћење погледа.**

---

WITH VideoPlaylist AS

```
(
  SELECT IdVid, Video.Naziv, COUNT (*) AS Br
  FROM Playlist JOIN Sadrzi USING (IdPla) JOIN Video USING (IdVid)
  WHERE IsActive = 1 AND IsPrivate = 0
  GROUP BY IdVid, Video.Naziv
)
SELECT IdVid, Naziv FROM VideoPlaylist
WHERE Br = (SELECT MAX(Br) FROM VideoPlaylist)
ORDER BY IdVid DESC
```

---

### Задатак 8 [6 поена]

Потребно је направити SQL упит који пребројава за сваког корисника колико тагова од оних које је он искористио на својим видео садржајима се користи само на једном видео садржају на нивоу целог система. Кориснике који нису постављали видео садржај или нису постављали такве тагове треба исписати са резултатом једнаким нула. Сортирати резултат по IdKor растуће.

Резултат дати у форми: IdKor, Username, Broj tagova

У Сactus-у користити таб: Zadatak 8

**Није дозвољено коришћење погледа.**

---

WITH KorisnikITacnoJedanTag AS (

```
  SELECT IdKor, IdTag
  FROM Video NATURAL JOIN Ima
  WHERE IdTag IN (
    SELECT IdTag
    FROM Ima
    GROUP BY IdTag
    HAVING COUNT(*)=1)
)
SELECT IdKor, Username, COUNT(IdTag) AS "Broj tagova"
FROM Korisnik LEFT OUTER JOIN KorisnikITacnoJedanTag USING (IdKor)
GROUP BY IdKor, Username
ORDER BY IdKor
```

---

### Задатак 9 [6 поена]

Потребно је направити SQL упит који врши категоризацију видео листа, тако што све видео листе на основу трајања смешта у једну од три групе - "kratka", "srednja" или "dugacka". Трајање видео листе представља збир трајања свих активних видео записа те листе. Кратка видео листа је она листа чије трајање припада првој трећини свих трајања видео листа. Средња видео листа је она листа чије трајање припада другој трећини свих трајања видео листа. Дугачка видео листа је она листа чије трајање припада трећој трећини свих трајања видео листа. Резултат треба да садржи и текстуалну представу категорије видео листе ("kratka", "srednja" или "dugacka"). Резултат сортирати прво по категоризацији тако да се прво приказују кратке, па средње и на крају дугачке видео листе, а унутар категоризације сортирати по IdPla растуће.

Резултат дати у форми: IdPla, Naziv, IsPrivate, Naziv kategorije

У Sactus-у користити таб: Zadatak 9

**Није дозвољено коришћење погледа.**

---

WITH PlaylistTrajanje AS

```
(
    SELECT IdPla, Playlist.Naziv, IsPrivate, (
        SELECT COALESCE (SUM(Trajanje), 0)
        FROM Sadrzi JOIN Video USING (IdVid)
        WHERE IsActive=1 AND IdPla=Playlist.IdPla
    ) AS Trajanje
    FROM Playlist
```

```
),
PlaylistPozicija AS
```

```
(
    SELECT IdPla, Naziv, IsPrivate, (SELECT COUNT (*)
                                     FROM PlaylistTrajanje
                                     WHERE PlaylistTrajanje.Trajanje < pt.Trajanje) AS Pozicija
    FROM PlaylistTrajanje pt
```

```
),
PlaylistKategorija AS
```

```
(
    SELECT IdPla, Naziv, IsPrivate, CASE
        WHEN Pozicija < (SELECT COUNT (*) FROM PlaylistPozicija)/3 THEN 0
        WHEN Pozicija < 2* (SELECT COUNT (*) FROM PlaylistPozicija)/3 THEN 1
        ELSE 2
    END AS Kat
    FROM PlaylistPozicija
```

```
)
SELECT IdPla, Naziv, IsPrivate, CASE Kat
    WHEN 0 THEN 'kratka'
    WHEN 1 THEN 'srednja'
    ELSE 'dugacka'
END AS "Naziv kategorije"
```

```
FROM PlaylistKategorija
ORDER BY Kat ASC, IdPla
```

---

Задатак 10 [7 поена]

Потребно је направити SQL упит који приказује наредних 20 видео садржаја ако би се укључила опција аутоматског прелаза са видео садржаја на видео садржај. Узети да се тренутно приказује видео садржај са IdVid=2. Алгоритам по ком се рачуна наредни видео садржај је следећи:

- прво се тражи најсроднији видео садржај (најсроднији видео садржај је онај који има највећи број заједничких тагова, ако има више таквих бира се онај са мањим идентификатором), у случају да ово правило враћа видео који је претходно гледан, онда се примењује следеће правило
- ако не постоји ниједан видео садржај из претходне ставке, онда се узима први видео садржај који има већи идентификатор, у случају да ово правило враћа видео који је претходно гледан, онда се примењује следеће правило
- ако не постоји ниједан видео садржај из претходне ставке, онда се узима први видео садржај који има мањи идентификатор, у случају да ово правило враћа видео који је претходно гледан, онда се примењује следеће правило
- ако не постоји ниједан видео садржај из претходне ставке, онда се понавља исти видео садржај

При испису резултата потребно је приказати редослед (1, 2, 3, ...). Резултат треба сортирати растуће по редоследу.

Резултат дати у форми: Redosled, IdVid, Naziv

У Sactus-у користити таб: Zadatak 10

**Није дозвољено коришћење погледа.**

---

```
WITH BrZajednickihTagova (Idvid1, IdVid2, Br) AS
(
SELECT ima1.IdVid, ima2.IdVid, COUNT(*)
FROM Ima ima1, Ima ima2
WHERE ima1.IdVid != ima2.IdVid AND ima1.IdTag=ima2.IdTag
GROUP BY ima1.IdVid, ima2.IdVid
),
AlgoritamPraviloJedan (IdVidTren, IdVidSled) AS
(
SELECT IdVid1, IdVid2 FROM BrZajednickihTagova b1 WHERE
NOT EXISTS (SELECT * FROM BrZajednickihTagova b2 WHERE (b1.IdVid1=b2.IdVid1 AND
b1.Br<b2.Br) OR (b1.IdVid1=b2.IdVid1 AND b1.IdVid2>b2.IdVid2 AND b1.Br=b2.Br))
),
AlgoritamPraviloDva (IdVidTren, IdVidSled) AS
(
SELECT v1.IdVid, v2.IdVid
FROM Video v1, Video v2
WHERE v2.IdVid>v1.IdVid AND NOT EXISTS (SELECT * FROM Video WHERE
Video.IdVid>v1.IdVid AND Video.IdVid<v2.IdVid)
),
AlgoritamPraviloTri (IdVidTren, IdVidSled) AS
(
SELECT v1.IdVid, v2.IdVid
FROM Video v1, Video v2
WHERE v2.IdVid<v1.IdVid AND NOT EXISTS (SELECT * FROM Video WHERE
Video.IdVid<v1.IdVid AND Video.IdVid>v2.IdVid)
),
```



```

Algoritam (IdVidTren, IdVidSled, Pravilo) AS
(
SELECT *, 1 FROM AlgoritamPraviloJedan
UNION ALL
SELECT *, 2 FROM AlgoritamPraviloDva
UNION ALL
SELECT *, 3 FROM AlgoritamPraviloTri
UNION ALL
SELECT IdVid,IdVid,4
FROM Video v
),
Autoplay (Redosled, IdVid, Naziv, IdVidPred) AS
(
SELECT 0, IdVid, Naziv, IdVid FROM Video WHERE IdVid=2
UNION ALL
SELECT Redosled+1, IdVidSled, Video.Naziv, Autoplay.IdVid
FROM Algoritam alg, Video, Autoplay WHERE Redosled<20 AND alg.IdVidSled=Video.IdVid AND
alg.IdVidTren=Autoplay.IdVid AND NOT EXISTS (SELECT * FROM Algoritam WHERE alg.Pravilo >
Algoritam.Pravilo AND Algoritam.IdVidTren=Autoplay.IdVid AND (IdVidPred<>Algoritam.IdVidSled
OR Algoritam.Pravilo=4))
AND (IdVidPred<>alg.IdVidSled OR alg.Pravilo=4)
LIMIT -1 OFFSET 1
)
SELECT Redosled, IdVid, Naziv FROM Autoplay

```

---