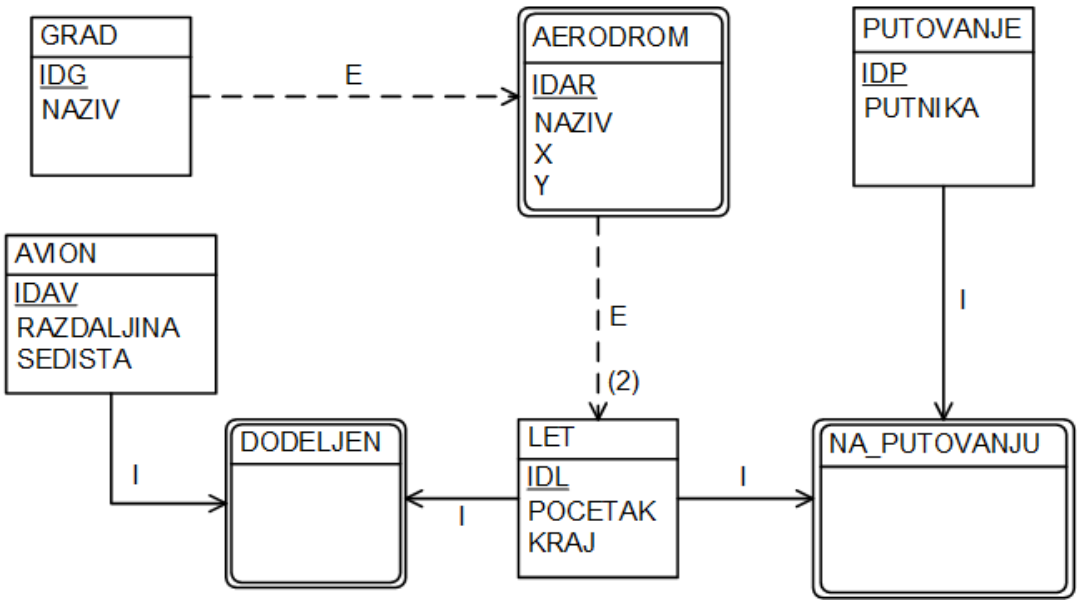
Универзитет у Београду **12.2.2022.**
Електротехнички факултет
Катедра за рачунарску технику и информатику

Базе података 1

(13С112БП1, 13Е113БП1, 13Е113БП)
- фебруарски испитни рок -

Посматра се део базе података авио компаније. У овој бази се чувају подаци о градовима, аеродромима, путовањима, авионима и њиховим летовима.

У наставку је дата релациона шема посматраног дела базе података.



GRAD (IDG, NAZIV)

IDG	- цео број, идентификује град, аутоматско додељивање наредног идентификатора
NAZIV	- низ до 200 знакова, обавезно

AERODROM (IDAR, NAZIV, X, Y, IDG)

IDAR	- цео број, идентификује аеродром, аутоматско додељивање наредног идентификатора
NAZIV	- низ до 200 знакова, обавезно
X	- реалан број, обавезно, представља географску дужину
Y	- реалан број, обавезно, представља географску ширину
IDG	- страни кључ (табела GRAD), обавезно, представља град у којем се налази аеродром

AVION (IDAV, RAZDALJINA, SEDISTA)

IDAV	- цео број, идентификује авион, аутоматско додељивање наредног идентификатора
RAZDALJINA	- цео број, већи од 0, обавезно, представља раздаљину коју авион може да прелети
SEDISTA	- цео број, већи од 0, обавезно, представља број седишта које авион поседује

LET (IDL, POCETAK, KRAJ, IDAR1, IDAR2)

IDL	- цео број, идентификује лет, аутоматско додељивање наредног идентификатора
POCETAK	- низ од тачно 16 знакова, обавезно, у формату ГГГГ-ММ-ДД ЧЧ:ММ, представља датум и време почетка лета
KRAJ	- низ од тачно 16 знакова, обавезно, у формату ГГГГ-ММ-ДД ЧЧ:ММ, представља датум и време краја лета
IDAR1	- страни кључ (табела AERODROM), обавезно, представља аеродром са којег се полеће
IDAR2	- страни кључ (табела AERODROM), обавезно, представља аеродром на који се слеће

PUTOVANJE (IDP, PUTNIKA)

IDP	- цео број, идентификује путовање, аутоматско додељивање наредног идентификатора
PUTNIKA	- цео број, већи од 0, обавезно, представља број путника на путовању

NA_PUTOVANJU (IDL, IDP)

IDL	- страни кључ (табела LET), део примарног кључа, обавезно, представља лет на путовању
IDP	- страни кључ (табела PUTOVANJE), део примарног кључа, обавезно, представља путовање

DODELJEN (IDL, IDAV)

IDL	- страни кључ (табела LET), део примарног кључа, обавезно, представља лет
IDAV	- страни кључ (табела AVION), део примарног кључа, обавезно, представља авион додељен лету

Задатак 1 [4 поена]

Потребно је направити SQL упит који за све градове који у називу имају бар две речи исписује информације о свим слетањима на аеродроме који се налазе у тим градовима. Резултат треба сортирати опадајуће по времену завршетка лета, а затим опадајуће по идентификатору лета.

Резултат дати у форми: IDG, NAZIV, IDL, POCETAK, KRAJ, AERODROM
У Cactus-у користити таб: Zadatak 1
Није дозвољено коришћење погледа.

[redacted]

Задатак 2 [4 поена]

Потребно је направити SQL упит који исписује оне градове код којих сваки аеродром има евидентирана бар 3 лета. Аеродром има евидентиран лет ако се приликом лета слетело или узлетело на аеродром. Резултат треба сортирати растуће по идентификатору града.

Резултат дати у форми: IDG, NAZIV
У Cactus-у користити таб: Zadatak 2
Није дозвољено коришћење погледа.

[redacted]

<u>Задатак 3 [4 поена]</u>

Потребно је направити SQL упит који за аеродроме исписује број полетања у мају на путовањима са мање од 23 путника. Резултат треба сортирати опадајуће по броју полетања, а затим растуће по идентификатору аеродрома.

Резултат дати у форми: IDAR, POLETANJA
У Cactus-у користити таб: Zadatak 3
Није дозвољено коришћење погледа.

[redacted]

<u>Задатак 4 [4 поена]</u>

Потребно је направити SQL скрипту која ако постоји табела ***AERODROM*** избацује табелу ***AERODROM*** из шеме, а затим формира нову табелу ***AERODROM*** која треба да има одговарајућу структуру и ограничења.

У Cactus-у користити таб: Zadatak 4

[redacted]

[redacted]

Задатак 5 [5 поена]

Потребно је направити SQL упит који исписује све аеродроме у градовима у којима се налази више од једног аеродрома и којима најближи аеродром не припада истом граду. Резултат треба сортирати опадајуће по X координати, затим опадајуће по Y координати.

Резултат дати у форми: IDAR, NAZIV, X, Y, IDG
У Cactus-у користити таб: Zadatak 5
Није дозвољено коришћење погледа.

```
SELECT *
FROM AERODROM A
WHERE (SELECT MIN(power(A.X -A2.X, 2)+power(A.Y -A2.Y, 2))
        FROM AERODROM A2
        WHERE A2.IDG=A.IDG AND A2.IDAR!=A.IDAR
        )
        =
        (SELECT MIN(power(A.X -A2.X, 2)+power(A.Y -A2.Y, 2))
        FROM AERODROM A2
        WHERE A2.IDG!=A.IDG)
ORDER BY X DESC, Y DESC
```

Задатак 6 [5 поена]

Потребно је направити SQL упит који исписује растојања између оних градова чије растојање није веће од 5. Растојање између два града је аритметичка средина растојања аеродрома та два града (сваки аеродром са сваким између градова). Приликом исписа градова исписати пар само једном и то тако да идентификатор првог града буде мање од идентификатора другог града. Резултат треба сортирати растуће по растојању, затим растуће по називу првог града, па растуће по називу другог града.

Резултат дати у форми: GRAD1, GRAD2, RASTOJANJE
У Cactus-у користити таб: Zadatak 6
Није дозвољено коришћење погледа.

```
SELECT G1.NAZIV AS GRAD1, G2.NAZIV AS GRAD2,
        AVG(SQRT(POWER(A1.X - A2.X,2) + POWER(A1.Y - A2.Y,2))) AS RASTOJANJE
FROM GRAD G1, AERODROM A1, GRAD G2, AERODROM A2
WHERE G1.IDG = A1.IDG AND G2.IDG = A2.IDG AND G1.IDG < G2.IDG
GROUP BY G1.IDG, G2.IDG
HAVING RASTOJANJE <= 5
ORDER BY RASTOJANJE ASC, GRAD1 ASC, GRAD2 ASC
```

Задатак 7 [5 поена]

Потребно је направити SQL упит који за сваки град исписује колико путника је узлетело, а колико слетело у периоду од '2014-05-10 00:00' до '2014-05-10 11:00'. Резултат треба сортирати растуће по идентификатору града.

Резултат дати у форми: IDG, NAZIV, UZLETELO, SLETELO
У Cactus-у користити таб: Zadatak 7
Није дозвољено коришћење погледа.

```
SELECT G.IDG, G.NAZIV
    COALESCE(
        SELECT SUM(P.PUTNIKA)
        FROM LET L INNER JOIN AERODROM A ON (L.IDAR1 = A.IDAR)
        INNER JOIN NA_PUTOVANJU NP ON (NP.IDL = L.IDL)
        INNER JOIN PUTOVANJE P ON (P.IDP = NP.IDP)
        WHERE A.IDG = G.IDG AND L.POCETAK >= '2014-05-10 00:00'
        AND L.POCETAK <= '2014-05-10 11:00'),0) AS UZLETELO,
    COALESCE(
        SELECT SUM(P.PUTNIKA)
        FROM LET L INNER JOIN AERODROM A ON (L.IDAR2 = A.IDAR)
        INNER JOIN NA_PUTOVANJU NP ON (NP.IDL = L.IDL)
        INNER JOIN PUTOVANJE P ON (P.IDP = NP.IDP)
        WHERE A.IDG = G.IDG AND L.KRAJ >= '2014-05-10 00:00'
        AND L.KRAJ <= '2014-05-10 11:00'),0) AS SLETELO
FROM GRAD G
ORDER BY G.IDG ASC
```

Задатак 8 [6 поена]

Потребно је направити SQL скрипту која за сваки град исписује његов ниво саобраћаја. Ниво саобраћаја зависи од збира броја свих полетања и слетања у том граду. Ниво саобраћаја је висок ако је збир већи од 6, средњи ако је збир већи од 3 а мањи или једнак 6, низак ако је збир мањи или једнак 3. Резултат сортирати по идентификатору града растуће.

Резултат дати у форми: IDG, NAZIV, SAOBRACAJ
У Cactus-у користити таб: Zadatak 8
Није дозвољено коришћење погледа.

[redacted]

Задатак 9 [6 поена]

Потребно је направити SQL упит који за она путовања на којима има летова исписује укупно чекање на преседањима за свако путовање, изражено у минутима. Резултат сортирати по идентификатору путовања растуће.

Препоручена документација: Date And Time Functions (sqlite.org)

Резултат дати у форми: IDP, CEKANJE
У Cactus-у користити таб: Zadatak 9

[redacted]

Задатак 10 [7 поена]

Потребно је направити SQL скрипту која излистава најкраћих 10 путева од града "New York" до града "Belgrade" са три преседања (укупно пет различитих аеродрома). Пут је дефинисан аеродромима, а збир раздаљина сукцесивних аеродрома на путу је дужина пута. Колона AERODROMI се исписује у формату "IDAR1.IDAR2.IDAR3.IDAR4.IDAR5". Сматрати да се са једног аеродрома може летети на све остале аеродроме. Резултат сортирати по дужини пута растуће.

Резултат дати у форми: AERODROMI, DUZINA_PUTA
У Cactus-у користити таб: Zadatak 10.

```
WITH RECURSIVE PUTANJE(AERODROMI, POLAZNI_AERODROM, RAZDALJINA, PRESEDANJA) AS (
    SELECT '.' || IDAR || '.', IDAR, 0, -1
    FROM AERODROM
    WHERE IDG IN (SELECT IDG FROM GRAD WHERE Naziv = 'Belgrade')
    UNION ALL
    SELECT '.' || A1.IDAR || P.AERODROMI, A1.IDAR,
        RAZDALJINA + sqrt(power(A1.X - A2.X,2) + power(A1.Y - A2.Y,2)),
        P.PRESEDANJA + 1
    FROM AERODROM A1, AERODROM A2, PUTANJE P
    WHERE A2.IDAR = P.POLAZNI_AERODROM
        AND P.AERODROMI NOT LIKE '%.' || A1.IDAR || '.%'
        AND P.PRESEDANJA <= 3
)
SELECT SUBSTR(AERODROMI, 2, length(AERODROMI)-2) AS AERODROMI, RAZDALJINA AS DUZINA_PUTA
FROM PUTANJE
WHERE PRESEDANJA = 3
    AND POLAZNI_AERODROM IN (
        SELECT IDAR
        FROM Aerodrom JOIN GRAD USING (IDG)
        WHERE GRAD.NAZIV = 'New York'
)
ORDER BY RAZDALJINA ASC
LIMIT 10
```
