

# Seminarska naloga PB za leto 2016

Jernej Habjan

Študent VSŠ FRI

ŠIFRA: 63150106

## Uvod:

Bazo sem dodal v Heidi.sql, lokalni strežnik sem pa gnal iz programa XAMPP, na katerega sem se pozneje povezal tudi s pythonom.

Vnos je bil hiter tudi z vnosom znotraj HeidiSQL, saj je kolega predelal albums.sql datoteko. Vsak pogled, trigger, tabela in funkcija imajo pred definiranjem zapisan "drop if exists" stavek, da sem lahko poizvedbo večkrat pognal

Jernej Habjan, 29 maj 2016

# Prva Naloga:

a)

Koda:

```
SELECT AlbumID, Price FROM Price;
```

b)

Koda:

```
SELECT AlbumID, Price
FROM Price
WHERE Price IS NOT NULL;
```

c)

Koda:

```
SELECT AlbumID, st_medijev, Medium FROM (
    SELECT AlbumID, NumberOfDiscs AS st_medijev, Medium, Price/NumberOfDiscs AS
razmerje
    FROM Price JOIN DiscNumber USING (albumID)
        JOIN Medium USING (AlbumID)
        JOIN MediumNames USING (MediumID)
    GROUP BY AlbumID
    ORDER BY razmerje) AS T;
```

Result #1 (3×38,531)		
AlbumID	st_medijev	Medium
544	1	Audio CD
441	1	Audio CD
440	1	Audio CD
439	1	Audio CD
436	1	Audio CD
138	1	Audio CD
434	1	Audio CD
433	1	Audio CD
143	1	Audio CD
145	(NULL)	Audio CD

- Ta naloga je bila meni ena težjih, saj sem moral dodati novo "select" poizvedbo za razmerje v gornji "select" stavek.

d)

Koda:

```
SELECT MIN(Year), MAX(Year)
FROM Year;
```

- Rezultata sta 1069 in 2029

e)

Koda:

```

SELECT AuthorID, AuthorName, COUNT(AlbumID) AS st_albumov
FROM Author JOIN AuthorNames USING (AuthorID)
GROUP BY AuthorID, AuthorName
ORDER BY st_albumov DESC;

```

f)

Koda:

```

DELIMITER //
DROP PROCEDURE IF EXISTS avgPric //
CREATE PROCEDURE avgPric(IN cena DOUBLE)
BEGIN
    SELECT Label, ((SELECT AVG(Price)
                    FROM Price JOIN Album USING(AlbumID) JOIN Label
                    USING(AlbumID))-(AVG(Price))
                ) as Razlika_Te_minus_povprecna
    FROM Price p JOIN Album USING(AlbumID) JOIN Label USING(AlbumID) JOIN
    LabelNames USING(LabelID)
    WHERE p.Price=cena
    GROUP BY LabelID

    HAVING COUNT(AlbumID) >= 10
    ORDER BY Razlika_Te_minus_povprecna DESC;
END //
DELIMITER ;
CALL avgPric(5.99);
16 CALL avgPric(5.99);

```

labelnames (2x9)	
Label	Razlika_Te_minus_povprecna
Polydor / Umgd	8.972323782409797
A&M;	8.972323782409797
Motown	8.972323782409797
Geffen Records	8.972323782409795
Mca Special Products	8.972323782409795
Island / Mercury	8.972323782409795
Mca Nashville	8.972323782409795
Mca	8.972323782409791
Rhino Flashback	8.972323782409791

- Zelo čudna navodila, zato sem naredil tako kot sem razumel.
- 5.99 je le primer cene.

g)

Koda:

```
SELECT COUNT(*) AS dodatna_dolocila
FROM Album
WHERE AlbumName LIKE '%[%]';
```

- Na koncu LIKE nisem dodal še en % znak, saj tako šteje, da je zaklepaj obvezno na koncu imena.

h)

Koda:

```
DROP PROCEDURE IF EXISTS Priporocam;
DELIMITER $
CREATE PROCEDURE Priporocam(IN isci VARCHAR(50))
BEGIN
    SELECT a.AlbumID, a.AlbumName, (SELECT recommended.AlbumName
    FROM Album recommended
    WHERE r.RecommendationID=recommended.AlbumID) AS RecommendedAlbum
    FROM Album a JOIN Recommendation r USING(AlbumID)
    WHERE UPPER(a.AlbumName)=UPPER(isci);
END $
DELIMITER ;
CALL Priporocam("Disturbed - Indestructible");
```

album (3×9)		
AlbumID	AlbumName	RecommendedAlbum
23764	Disturbed - Indestructible	Linkin Park - Hybrid Theory
23764	Disturbed - Indestructible	Rage Against The Machine - Battle of Los Angeles
23764	Disturbed - Indestructible	Metallica - Metallica
23764	Disturbed - Indestructible	Alice In Chains - Dirt [Explicit Lyrics]
23764	Disturbed - Indestructible	System of a Down - Toxicity
23764	Disturbed - Indestructible	Metallica - Death Magnetic
23764	Disturbed - Indestructible	Disturbed - Believe [Enhanced, Explicit Lyrics]
23764	Disturbed - Indestructible	Disturbed - Sickness [Explicit Lyrics]
23764	Disturbed - Indestructible	Disturbed - Ten Thousand Fists

- V notranjem WHERE delu sem klical UPPER, da lahko vpišemo ime albuma z malimi črkami in ga bo še vseeno našlo.
- "Disturbed - Indestructible" je primer mojega iskanja po tem albumu.

i)

Koda:

```
SELECT DISTINCT aN2.AuthorName AS Recommended
FROM AuthorNames aN1 JOIN Author a1 ON(aN1.AuthorID=a1.AuthorID)
      JOIN Recommendation r1 ON(a1.AuthorID=r1.AlbumID)
      JOIN Recommendation r2 ON(r1.RecommendationID=r2.AlbumID)
      JOIN Author a2      ON(r2.RecommendationID=a2.AlbumID)
      JOIN AuthorNames aN2 ON(a2.AuthorID=aN2.AuthorID)
WHERE aN1.AuthorName="U2";
```



The screenshot shows a database query result window titled 'authornames (1x64,825)'. It displays a list of recommended authors under the column header 'Recommended'. The authors listed are: John Pizzarelli, Tierney Sutton, The Tierney Sutton Band, Jeremy Pelt, Incognito, Kirk Whalum, Bill Evans, and Quantic & Alice Russel.

Recommended
John Pizzarelli
Tierney Sutton, The Tierney Sutton Band
Jeremy Pelt
Incognito
Kirk Whalum
Bill Evans
Quantic & Alice Russel

- Ena izmed zahtevnejših nalog, saj si sprva nisem dobro predstavljal kako se povezujejo te relacije.
- Povežem se z avtorjem na vse njegove albume in potem še na vsa priporočila od teh albumov, od teh pa izvem, kdo jih izvaja.

j)

Koda:

```
ALTER TABLE Recommendation ADD INDEX(AlbumID);
```

- Index na tabelo Recommendation je pohitрил poizvedbo na 0.078 sekund na mojem računalniku.

## Druga Naloga:

a)

Koda:

```
DROP TABLE IF EXISTS razsirjeni_izvajalci;
DROP VIEW IF EXISTS razsirjeni_izvajalci;
CREATE VIEW razsirjeni_izvajalci AS
    SELECT AuthorID, AuthorName, COUNT(AlbumID) AS st_njegovih_albumov
    FROM Author JOIN AuthorNames USING(AuthorID)
    GROUP BY AuthorID, AuthorName;
SELECT * FROM razsirjeni_izvajalci;
```

Result #1 (3×12,384)		
AuthorID	AuthorName	st_njegovih_albumov
1	Santana	39
2	Hot Club of San Francisco	1
3	Richard Barone	1
4	Johnny Cash	95
5	Blitzen Trapper	5
6	MF Doom	8

- Ta primer vsebuje še stavek drop table, tako da ga lahko požnem z primerom b.

b)

Koda:

```
DROP TABLE IF EXISTS razsirjeni_izvajalci;
DROP VIEW IF EXISTS razsirjeni_izvajalci;
CREATE TABLE razsirjeni_izvajalci AS
    SELECT AuthorID, AuthorName, COUNT(AlbumID) AS st_njegovih_albumov
    FROM Author JOIN AuthorNames USING(AuthorID)
    GROUP BY AuthorID, AuthorName;
SELECT * FROM razsirjeni_izvajalci;
```

- Enako kot primer a, z razliko "create table".

c)

Koda:

```
DROP TRIGGER IF EXISTS on_Razsirjeni_Insert;
DELIMITER $
CREATE TRIGGER on_Razsirjeni_Insert
AFTER INSERT ON Album
FOR EACH ROW
BEGIN
    UPDATE Razsirjeni_izvajalci SET

        st_njegovih_albumov = (SELECT COUNT(AlbumID)

                                FROM Author
                                WHERE AuthorID =

AuthorID);
END $
DELIMITER ;
#####
DROP TRIGGER IF EXISTS on_Razsirjeni_Delete;
DELIMITER $
CREATE TRIGGER on_Razsirjeni_Delete
AFTER DELETE ON Album
FOR EACH ROW
BEGIN
    UPDATE Razsirjeni_izvajalci SET

        st_njegovih_albumov = (SELECT COUNT(AlbumID)

                                FROM Author
                                WHERE AuthorID =

AuthorID);
END $
DELIMITER ;
```

- Za vsakega avtorja sem še enkrat izračunal število njegovih albumov, tako pri brisanju iz album in pri insertu v albums.

# Tretja Naloga:

Koda:

```
DROP TABLE IF EXISTS GroupedData;
CREATE TABLE GroupedData AS

    SELECT y.Year, p.Price, NumberOfDiscs, MediumID
    FROM Year y JOIN Price p      USING (AlbumID)
                JOIN DiscNumber USING (AlbumID)
                JOIN Medium      USING (AlbumID);

SELECT * FROM GroupedData;

DROP TABLE IF EXISTS Year;
DROP TABLE IF EXISTS Price;
DROP TABLE IF EXISTS DiscNumber;
DROP TABLE IF EXISTS Medium;

ALTER TABLE GroupedData ADD FOREIGN KEY(MediumID) REFERENCES MediumNames(MediumID);
```

albums.groupeddata: 38,956 rows total (approximately), limited

Year	Price	NumberOfDiscs	 MediumID
2012	9.99	1	1
2012	14.99	1	1
2010	12.99	1	1
2006	8.85	1	1
2009	8.98	1	1
2009	10.8	1	1
2004	(NULL)	1	1
2003	14.99	1	1
1964	(NULL)	4	1
(NULL)	22.61	2	1
(NULL)	8.84	1	1
2000	13.42	2	1
2005	15.9	1	1
2008	5.99	1	1
2000	10.87	1	1



## Četrta Naloga:

Python koda je v datoteki "odbcJernejHabjan.py"

Imel sem nekaj težav s povezavo strežnika s pythonom in inštalacijo pyodbc, poleg tega so bile pa težave minimalne.

Še malo razlage o kodi:

- Def clearTables() zbriše obe tabeli. Prvo zbriše ExtraData, ki vsebuje tuji ključ ki veže primarni ključ v ExtraDataNames.
- Pri prvi nalogi razdelim vrstico po simbolu "[" in potem obdelujem vsebino desno od tega. Nakoncu pa še odstranim desni zaklepaj.
- Pri drugi nalogi pa naredim tabelo, kjer je ExtraID primarni ključ, potem pa vstavljam v to tabelo dodatna določila, ki jih imam shranjene v seznamu, ki sem ga sestavil v prvi nalogi.
- Pri tretji nalogi pa naredim dva slovarja s ključi, ki so ID, in vrednostmi, ki so v prvem seznamu deli besede med oglatimi oklepaji in celimi imeni v drugem  
Potem samo v for zanki primerjam vrednost iz prvega seznama s substringom vrednosti drugega.
- Potem še zaprem connection



ExtraDataNames:

albums.extradataNames: 661 rows total (approximately)

ExtraID	ExtraName
1	MCA 15024 LP Vinyl Record
2	13 Disc Set
3	140g Vinyl LP
4	15th Anniversary Deluxe Limited Edition
5	180 Gram
6	180 Gram Vinyl
7	180g Vinyl
8	2 CD
9	2 CD + 1 DVD Combo
10	2 CD + DVD
11	2 CD + 1 DVD + 1 DVD + 1 DVD

## ExtraData:

albums.extradata: 15,899 rows tot

 AlbumID	 ExtraID
30452	412
11431	416
7783	569
7762	383
9654	412
13547	460
28467	344
13842	536
30186	60
19298	312
21093	101
10666	412