

SQL ESSENTIAL TRAINING

②

> SELECT 'Hello World';
- najjednostavniji QUERY, selecta 1 string
- ne queryja bazu

> USE world;

> SELECT * FROM Country;
- selecta sve iz Country tablice iz baze
↓
DAJE TABLICU ZA SELECT

> SELECT COUNT(*) FROM Country;
- upotrebila funkcije
↓
FUNKCIJA ZA SELECT

SELECTANJE POREĐENJA STUPACA

> SELECT Name, Life Expectancy FROM Country;
→ navedi stupce iz tablice
UMJESTO ZVIJERICE NAVODIMO STUPCE KOJE TREBAMO

PROJENA IMENA STUPACA - upotreba AS

> SELECT Name AS Country, Life Expectancy AS 'Life Exp.' FROM Country;
↓
UPOTREBA NAVODNIKA KAO NOVI IME IMA SPACE
- SINGLE QUOTES SU UVIJEK ISPRAVNE U SQL-U
↓
VRAĆA DRUGAČIJE NAZVANE STUPCE

ODABIR PODATAKA U SELECTU SA WHERE

ODNO JE UVJET
(CONDITION)

> SELECT Name, Continent, Region FROM Country WHERE Continent = 'Europe';

VRATA OVE STUPCE

VRATA RETKE KOJI ODO ZADOVOLJAVAJU

OPERATOR

COUNT FUNKCIJA

> SELECT COUNT(*) FROM Country;

↳ BROJI SVE RETKE U TABlici

> SELECT COUNT(IndepYear) FROM Country;

↓
STUPAC

→ VRATA BROJ PODATAKA KOJI IMAJU NEKU VRIJEDNOST U ODABRANOM STUPCU (DA VRIJEDNOST NIJE NULL)

GROUP BY + COUNT - group by radi s agregatorima

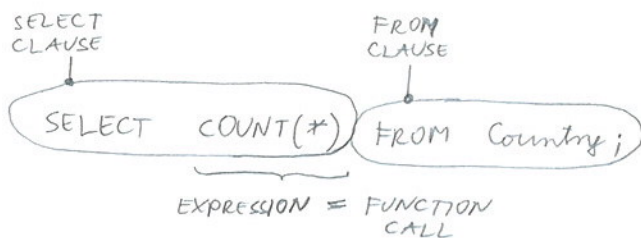
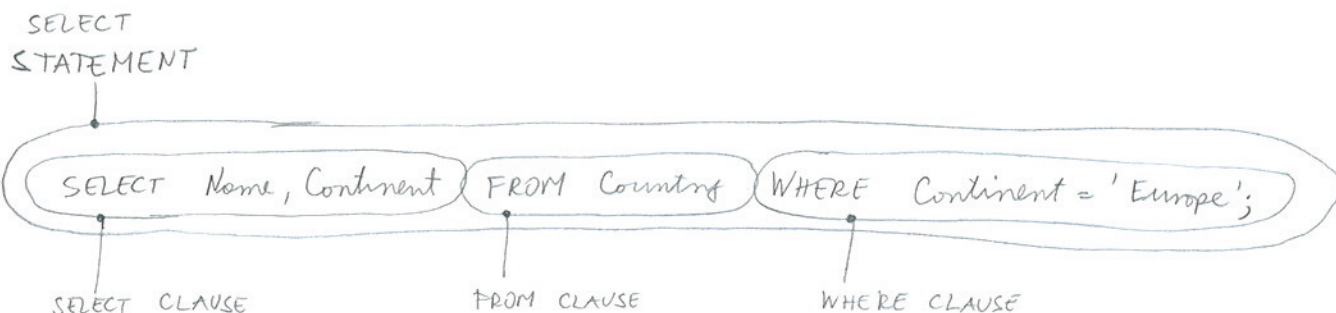
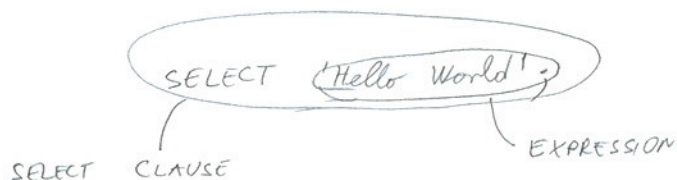
> SELECT Continent, COUNT(Name) AS Countries FROM Country GROUP BY Continent;

COUNT ZBRAJA GRUPE I NE SVE

GROUP BY RADI S COUNTOM

3

SQL statements se sastoje od:
 - clauses
 - expressions



FORMATIRANJE SQL-a - za readability

- whitespace se ignorira
- SPACE i NEWLINES su WHITESPACE

```
> SELECT Name, Continent
    FROM Country
    WHERE Continent = 'Europe';
```

} INDENTANJE FROM i WHERE KLAUZA
SE OPERIRAJU NA SELECT STATEMENTU

```
> SELECT c.Name, c.Continent, ct.Name AS Capital
    FROM Country AS c
    JOIN City AS ct
    ON ct.ID = c.Capital
    ORDER BY c.Name DESC;
```

} POMOCU INDENTATIONA
ZNAMO ŠTO SE NA
ŠTO DODOSI

KREIRANJE BAZE I TABLICA

Kreiranje tablice i baze

> CREATE DATABASE baza;

> CREATE TABLE customer(
 id INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
 name VARCHAR(255),
 address VARCHAR(255),
 city VARCHAR(255),
 state CHAR(2),
 zip CHAR(10)
);

→ NE SMIJE IMAT NULL
- AKO NETKO UMETNE
NULL DOGODIT ĆE
SE GREŠKA

Insert

INSERT PODATAKA

IME
TABLICE
↓
> INSERT INTO customer (name, address, city, state, zip)
VALUES ('John Smith', '123 Main Street', 'Liberty', 'CA', '98765');

LISTA
STUPACA
↑
VRJEDNOSTI ZA LISTU STUPACA

VALUES
CLAUSE

JOIN QUERY

= SELECT + JOIN CLAUSE

Join, where, like, in

> SELECT c.Name AS Country, c.Continent, ct.Name AS Capital
FROM Country AS c
JOIN City AS ct
ON ct.ID = c.Capital → UVIJET NA KOJEM SU
DVIJE TABLICE SPOJENE
ORDER BY Country;

DRUGI (STARIJI) NAČIN

> SELECT c.Name AS Country, c.Continent, ct.Name AS Capital
FROM Country AS c, City AS ct
WHERE ct.ID = c.Capital → ON CLAUSE POSTAJE WHERE
ORDER BY Country;

FILTRIRANJE SA WHERE, LIKE & IN

> SELECT CountryCode, Name, Population
FROM City
WHERE CountryCode = 'GBR';
UVJET

> SELECT CountryCode, Name, Population
FROM City
WHERE Population >= 5000000
UVJET

LIKE

> SELECT CountryCode, Name, Population
FROM City
WHERE Name LIKE 'Z%';
UVJET

- POČINJE SA Z
- SVI ZNAKOVI NAKON TOGA SU OK

→ WHERE Name LIKE '%Z';

↳ SVAKO IMENO KOJE ZAVRŠAVAJE NA Z

→ WHERE Name LIKE '%Z%';

↳ IMENA KOJA IMAJU Z BILU GDJE (POČETAK, SREDINA ILI KRAJ)

IN OPERATOR

```
> SELECT CountryCode, Name, Population  
    FROM City  
    WHERE CountryCode IN ('USA', 'CAN', 'MEX');
```

→ BIRA VRIJEDNOST IZ LISTE

KOMBINIRANJE UVJETA SA AND, OR

```
> SELECT CountryCode, Name, Population  
    FROM City  
    WHERE CountryCode IN ('USA', 'MEX', 'CAN')  
    AND Population > 1 000 000
```

• DISTINCT - uklanjanje duplikata

↳ upotreba sa SELECT da remove duplikate iz QUERY-a

```
> SELECT DISTINCT GovernmentForm, HeadOfState FROM Country WHERE HeadOfState  
    VRAĆA SAMO UNIQUE REZULTATE LIKE 'Elis %';
```

→ KAD PIŠEMO SAMO 'SELECT' TO ZNAČI 'SELECT ALL'

• ORDER BY - poredak rezultata

Update

```
> SELECT Name, District  
FROM City  
WHERE CountryCode = 'USA'
```

```
ORDER BY Name;
```

→ VRAĆA ABECEDNI POREDAK PO IMENU

```
ORDER BY District, Name;  
MOŽE BIT VIŠE  
PARAMETARA
```

→ PRVO POREDAK PO Districtu, a onda
'unutar' Districta po imenu

```
> SELECT Region, AVG(LifeExpectancy) AS AvgLE  
FROM Country  
WHERE LifeExpectancy  
GROUP BY Region
```

```
ORDER BY AVG(LifeExpectancy);  
IZRAZ
```

→ ORDER BY AvgLE;

BOJE JE OVAKO NAPIŠAT
TOČAN NAZIV STUPCA

• UPDATING DATA

! PRIJE UPDATEA TESTIRAJ IZRAZ SA SELECTOM

```
> SELECT * FROM track WHERE id = 16;
```

OVAJ DIO OSTANUJE
I ZA UPDATE

→ PROVJERA IMAMO LI PRAVI
REDAK

```
> UPDATE track SET title = 'blue suede shoes' WHERE id = 16;
```

↓
update koniti where do zna koji
podatke promijeniti

• DELETING DATA

Delete

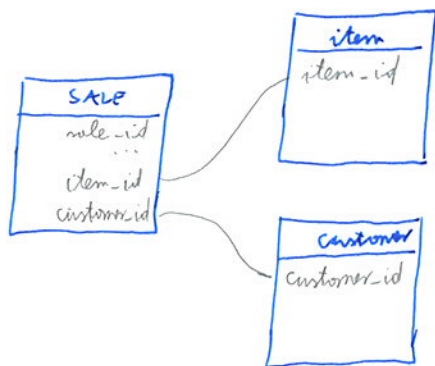
! UNJEK TESTIRATI WHERE CLAUSE SA SELECTOM PRIJE

> SELECT * FROM track WHERE id = 70;

> DELETE FROM track WHERE id = 70;

WHERE OSTAJE ISTI KAO U TESTNOM SELECTU
AKO NEMA WHERE - BRIŠU SE SVI RETCI IZ TABLICE.

4



BAZA test;

- provjera koliko jedinica 'itema' je prodatano

> SELECT SUM(s.quantity) AS QUANTITY, i.name AS Item
FROM sale AS s
JOIN item AS i ON s.item_id = i.id
GROUP BY i.id;

GROUP BY ZA SUM

INNER JOIN

1.

QUANTITY	ITEM
5	1
3	1
2	2
4	2

2.

QUAN	ITEM
8	1
6	2

RIGHT JOIN

- prikazuje iteme koji imaju NULL value

RIGHT JOIN item AS i ON s.item_id = i.id

JOIN S DVIJE TABLICE

Join with 2 tables

```
> SELECT s.date, c.name AS Customer, i.name AS item, s.quantity, s.price  
FROM sale AS s  
JOIN item AS i ON s.item_id = i.id  
JOIN customer AS c ON s.customer_id = c.id  
ORDER BY s.date
```

~~UVIJEK JE OSNOVNA
TABLICA ONA KOJA
SADRŽI ID-ove?~~

INDEXI

- mali, brzi i optimizirani
- kad u tablici kreiramo PRIMARY KEY stvara se INDEX

STVARANJE DOBATNIH INDEXA

```
> CREATE TABLE Customer (  
  id INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR (255),  
  address VARCHAR (255),  
  city VARCHAR (255),  
  state CHAR (2),  
  zip CHAR (10),  
  INDEX (name),  
  INDEX (zip).  
);
```

• STRINGOVI | STRING FJE

String funkcije

> SELECT ' Apostrof "' se tipka sa 2 apostrofa';
 ↓
 "

LENGTH - daje broj znakova u STRINGU

> SELECT LENGTH('Rock and Roll Medley') AS Length;
 ↳ 20

- length se može računati i na celovima
- length nije broj bajtova, nego broj znakova (valjda 1 znak može imati više bajta)

SUBSTRING - uzima dio stringa

> SELECT SUBSTR('Hello World', 1, 5) AS String;
 1 2 3 4 5 ↓ broj znakova
 ↳ 'Hello' ↑ POČINJE OD OVOG ZNAKA

- index 1 je prvi znak, nema indexa 0

RIGHT & LEFT - uzima broj znakova sa lijeve ili desne strane

> SELECT RIGHT('Hello World', 5) AS String;
 5 4 3 2 1
 ↳ 'World'

TRIM - uklanjanje nepotrebnog spacea na početku i kraju stringa

> SELECT TRIM(' Nesto spacea ') AS String;

UPPERCASE i LOWERCASE FJE

UPPER('...')

LOWER('...')

> SELECT UPPER('nesto') AS String;
↳ 'NESTO'

Numeric fje

• NUMERIC FJE

ABS

- absolutna vrijednost broja

> SELECT ABS(-12);

↳ 12

OVO JE
STRING!

> SELECT ABS('-12');

↳ 12

- SVI DBMS-ovi MOGU
PREPOZNAT BROJEVE U STRINGU
I RADITI NUMERIČKE FJE S TIM

- APO SE STRING SA ZNAKOM
NADE U NUM FJI MYSQL VRAĆA \emptyset ,
A NEKE DRUGE DBMS-ovi VRAĆE 'NAN'
NOT A NUMBER

ROUND

< .5 i niže - zaokružuje na VIŠE
< iznad .5 - na NIŽE

> SELECT ROUND(5.49, 0);

↳ 5

5.6

↳ 5

-5.49

↳ -5

-5.5

↳ -6

SEC_TO_TIME, DIV, MOD

> SELECT SEC_TO_TIME(320);

↳ 00:05:20

ZA SATI, MINUTE I SEKUNDE

> SELECT CONCAT(320 DIV 60,

':',

LPAD(320 MOD 60, 2, '0'));

↳ LPAD SE 'PUNI'
NULAMA

LPAD - POJE
IMA UVAZNO
2 ZNAKA

↳ '5:20'

• DATE & TIME FJE

- date i time su poseban tip podatka
- standardi se dosta razlikuju po platformama

MYSQL TIPOVI DATUMA (4 TIPA)

- DATE - sprema samo datume od godine 1000 do 9999 godine
- TIME - vrijeme dana ili vremenski period ($\pm 840h$)
- DATETIME - ima sve gore
- TIMESTAMP - uvijek spremljeno u UTC (londonsko vrijeme)
 - prevodi se u lokalno vrijeme (servera)
 - zamišljeno da bilježi evente

CURDATE()

> SELECT CURDATE();
↳ 2011-10-26

CURTIME()

> SELECT CURTIME();
↳ 18:54:24
↳ SADI SU U 24 H FORMATU

NOW()

> SELECT NOW();
↳ VRAĆA DATUM I VRIJEME

DATE_ADD - dodava vrijeme

> SELECT DATE_ADD(NOW(), INTERVAL 2 WEEK) AS Later;

DA JE OVUJE

CURDATE(), PRIKAZUJE
BI SE SAMO DATUM

↳ PRIKAZUJE DATE + TIME ZA 2 TJEDNA

DATE_SUB - oduzima vremena

> SELECT DATE_SUB(NOW(), INTERVAL 2 WEEK) AS Later;

• AGGREGATE FUNKCIJE

Date & time, aggregate fje

- COUNT - može se upotrijebiti sa group by

```
> SELECT a.title AS Album, COUNT(t.track_number) AS Tracks  
FROM track AS t  
JOIN album AS a ON a.id = t.album_id  
GROUP BY t.album_id  
ORDER BY Tracks DESC, Album;
```

1. dogodi se join
2. zatim se izvrši group by
3. zatim aggregate fja

```
> SELECT a.title AS album, COUNT(t.track_number) AS Tracks  
FROM track AS t  
JOIN album AS a ON a.id = t.album_id  
GROUP BY t.album_id
```

→ HAVING Tracks >= 10 → HAVING JE ZA COUNT ISTO ISTO
ORDER BY Tracks DESC, Album; | WHERE ZA SELECT

- having - za filtriranje na agregiranim vrijednostima

DISTINCT

```
> SELECT COUNT ( DISTINCT HeadOfState ) AS NumHeads FROM Country;
```

REGI SMO UNIQ VRIJEDNOSTI

- distinct se može koristiti u svim aggregate fja

SUM - sumira sve vrijednosti

AVG - daje srednju vrijednost (za neki stupac npr.)

MIN - daje najmanju/najveću vrijednost iz nize

MAX