

# UVOD U BAZE PODATAKA

Predavanje 9-10

# MI1 - komentar

Ukupno upisanih studenata	257	
Prijavljenih	217	84%
Nije predalo rješenje	49	23%
Ocijenjenih rješenja	168	65%

Prošli ishoda		
sva tri ishoda	73	43%
dva ishoda	48	29%
jedan ishod	23	14%
nijedan ishod	24	14%

- Nedostatno čitanje upute na ispitu
  - Neimenovana rješenja
- Prepisivanje
  - Zabrana korištenja interneta
  - AI

Prošli I1	76%
Prošli I2	71%
Prošli I3	54%

# Ponavljanje osnovnih operacija

- Podaci se umeću naredbom **INSERT INTO**

```
INSERT INTO tablica (stupac_1, ..., stupac_n)  
VALUES (vrijednost_1, ..., vrijednost_n)
```

- Podaci se mijenjaju naredbom **UPDATE**

```
UPDATE tablica  
SET  
    stupac_1 = vrijednost_1,  
    ...  
    stupac_n = vrijednost_n  
WHERE uvjet
```

- Podaci se brišu naredbom **DELETE**

```
DELETE FROM tablica  
WHERE uvjet
```

# Spajanje tablica

# Uvod

- U SELECT naredbi izvor podataka nam je bila jedna tablica:

```
SELECT ...
```

```
...
```

```
FROM izvor_podataka
```

```
WHERE ...
```

```
ORDER BY ...
```

- Često se podaci koji nam trebaju nalaze u dvije ili više tablica
- **Spajanje** (engl. *join*) služi za dohvaćanje podataka iz više tablica koje su semantički povezane
- Veze između tablica su najčešće **strani ključevi**

# Primjer potrebe za spajanjima

- Imamo tablice Osoba i Mjesto:

IDOsoba	Ime	Prezime	PrebivalisteID
1	Miro	Mirić	2
2	Ana	Anić	4

Strani ključ

IDMjesto	Naziv
1	Zagreb
2	Osijek
3	Pula
4	Rijeka

- Želimo prikazati sljedeće podatke:

OsobaIme	OsobaPrebivaliste
Miro Mirić	Osijek
Ana Anić	Rijeka

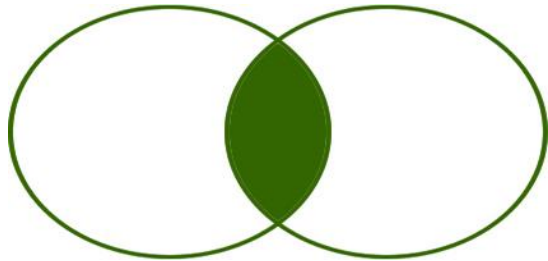
- Kako napisati SELECT upit?

# Tipovi spajanja

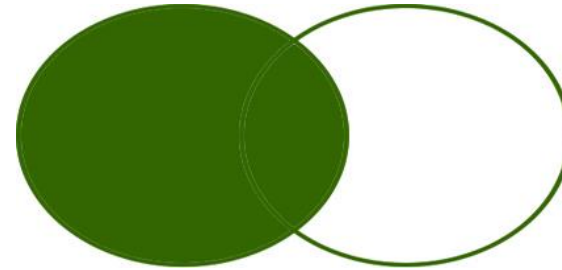
- Tipovi spajanja:
  - Unutarnje spajanje (engl. *inner join*)
  - Vanjsko spajanje:
    - Lijevo vanjsko spajanje (engl. *left outer join*)
    - Desno vanjsko spajanje (engl. *right outer join*)
    - Puno vanjsko spajanje (engl. *full outer join*)
    - Unakrsno spajanje (engl. *cross join*)
- U jednom spajanju uvijek sudjeluju dvije tablice
  - Poseban slučaj je spajanje **jedne tablice sa samom sobom**
- Spajanje je operacija nad elementima skupa

# Grafički prikaz spajanja

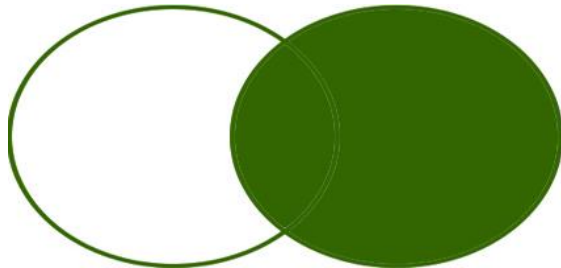
Unutarnje spajanje



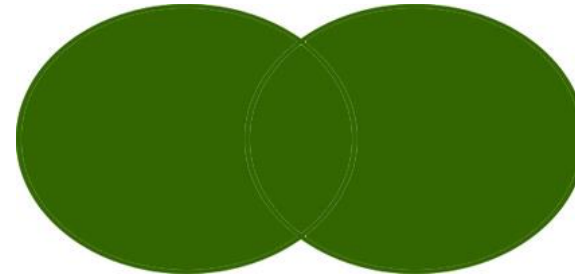
Lijevo vanjsko spajanje



Desno vanjsko spajanje



Puno vanjsko spajanje





# Napomene kod spajanja

- Kod spajanja je najbolje svakoj tablici koja sudjeluje u spajanju dodijeliti **alias**
  - Alias tablicama dodjeljujemo naredbom **AS**, jednako kao i stupcima u **SELECT** naredbi
  - Tablici smo time dali drugi naziv, obično kraći
- Kod svakog korištenje stupaca najbolje je ispred svakog stupca navesti **alias**
  - Alias i naziv stupac se odvajaju znakom '.' (točka)
    - Primjerice, **g.Naziv**
  - Na taj način smo jasno rekli iz koje tablice dolazi stupac
  - Ponekad u više tablica imamo stupac istog imena

# Spajanje više tablica

- Ako želimo spojiti više od dvije tablice, spajanja se izvršavaju po sljedećem principu:
  1. Prvo se spajaju **prva** i **druga** navedena tablica
  2. Tablica dobivena kao **rezultat koraka 1** se spaja s **trećom** navedenom tablicom
  3. Tablica dobivena kao **rezultat koraka 2** se spaja s **četvrtom** navedenom tablicom
  4. Postupak se ponavlja dok se ne iscrpe **sve navedene tablice**
- Svako spajanje može biti bilo kojeg navedenog tipa

# Unutarnje spajanje

# Unutarnje spajanje

- Unutarnje spajanje je najčešći oblik spajanja
- Logički opis postupka spajanja:
  1. Zadaje se **uvjet spajanja**:
    - Najčešće po jedan stupac iz svake tablice i operator '=' (tzv. *equi-join*)
  2. Za svaki redak u prvoj tablici RDBMS pronalazi sve retke u drugoj tablici koji zadovoljavaju uvjet spajanja
  3. Rezultat koraka 2 je privremena tablica koja sadrži spojene retke i koja **služi kao izvor podataka**
    - Strukturu dobivene tablice čini **unija stupaca** iz obje tablice
- Nije bitno koja tablica se navede prva, a koja druga

# Slikoviti opis postupka spajanja

- Uzmimo tablice Drzava i Grad sa strukturom i podacima:

IDDrzava	Naziv	IDGrad	Naziv	DrzavaID
1	Hrvatska	1	Rijeka	1
2	Škotska	2	Zadar	1
3	Norveška	3	Glasgow	2

- Alias: Drzava = d, Grad = g
- Uvjet spajanja:  $d.IDDrzava = g.DrzavaID$
- Rezultat spajanja je sljedeća tablica:

IDDrzava	Naziv	IDGrad	Naziv	DrzavaID
1	Hrvatska	1	Rijeka	1
1	Hrvatska	2	Zadar	1
2	Škotska	3	Glasgow	2

# SQL sintaksa

- Sintaksa unutarnjeg spajanja je sljedeća:

**FROM**

*prva\_tablica AS alias1*

**INNER JOIN** *druga\_tablica AS alias2*

**ON** *uvjet\_spajanja*

- Ako koristimo najčešći uvjet spajanja, sintaksa postaje:

**FROM**

*prva\_tablica AS alias1*

**INNER JOIN** *druga\_tablica AS alias2*

**ON** *alias1.stupac1 = alias2.stupac2*

# Primjer unutarnjeg spajanja

- Želimo ispisati nazive gradova i uz svaki grad naziv države. Zanimaju nas samo gradovi koji počinju slovom S ili Z i želimo ih sortirati silazno po nazivu države:

```
SELECT
```

```
  g.Naziv AS NazivGrada,
```

```
  d.Naziv AS NazivDrzave
```

```
FROM Grad AS g
```

```
INNER JOIN Drzava AS d
```

```
  ON g.DrzavaID = d.IDDrzava
```

```
WHERE g.Naziv LIKE 'S%' OR g.Naziv LIKE 'Z%'
```

```
ORDER BY NazivDrzave DESC
```

# Primjeri

1. Dohvatiti imena i prezimena svih kupaca i uz svakog ispisati naziv grada.
2. Dohvatiti imena i prezimena svih kupaca i uz svakog ispisati naziv grada i države.
3. Za sve račune izdane 01.08.2002. i plaćene American Expressom ispisati njihove ID-eve i brojeve te ime i prezime i grad kupca, ime i prezime komercijaliste te broj i podatke o isteku kreditne kartice. Rezultate sortirati prema prezimenu kupca.
4. Ispisati nazive proizvoda koji su na nekoj stavci računa prodani u više od 35 komada. Svaki proizvod navesti samo jednom.



# Vanjska spajanja

# Lijevo vanjsko spajanje

- Unutarnje spajanje vraća samo one retke iz prve tablice koji imaju odgovarajući redak u drugoj tablici i obratno
- **Lijevo vanjsko spajanje** radi na sljedeći način:
  1. Vraća sve retke kao i unutarnje spajanje
  2. Vraća sve retke iz **prve** tablice koji nisu spojeni u koraku 1
    - Pri tom su sve vrijednosti stupaca iz druge tablice jednake NULL
- Sintaksa je ista, samo se umjesto **INNER JOIN** koristi naredba **LEFT OUTER JOIN**
- **Važno** je paziti koju tablicu navedimo prvu, a koju drugu

# Desno vanjsko spajanje

- Analogno lijevom vanjskom spajanju
- **Desno vanjsko spajanje** radi na sljedeći način:
  1. Vrati sve retke kao i unutarnje spajanje
  2. Vrati sve retke iz **druge** tablice koji nisu spojeni u koraku 1
    - Pri tom su sve vrijednosti stupaca iz prve tablice jednake NULL
- Koristi se naredba **RIGHT OUTER JOIN**
- **Važno** je paziti koju tablicu navedimo prvu, a koju drugu
- Ako u lijevom vanjskom spajanju zamijenimo redoslijed tablica, dobijemo desno vanjsko spajanje
  - Vrijedi i obratno

# Puno vanjsko spajanje

- **Puno vanjsko spajanje** radi na sljedeći način:
  1. Vrati sve retke kao i unutarnje spajanje
  2. Vrati sve retke iz **prve** tablice koji nisu spojeni u koraku 1
    - Pri tom su sve vrijednosti stupaca iz druge tablice jednake NULL
  3. Vrati sve retke iz **druge** tablice koji nisu spojeni u koraku 1
    - Pri tom su sve vrijednosti stupaca iz prve tablice jednake NULL
- Koristi se naredba **FULL OUTER JOIN**
- Nije bitno koja tablica se navede prva, a koja druga

# Primjeri

5. Koristeći lijevo vanjsko spajanje prikazati nazive država i nazive pripadajućih gradova. Za državu za koju nema definiranih gradova ispišite NULL.
6. Koristeći lijevo vanjsko spajanje prikazati nazive proizvoda i nazive njihovih potkategorija i kategorija. Za proizvode koji nemaju definiranu potkategoriju ispisati NULL.
7. Koristeći lijevo vanjsko spajanje dohvatiti sve proizvode koji nisu nikad prodani.
8. Koristeći desno vanjsko spajanje dohvatiti sve proizvode koji nisu nikad prodani.

# Primjeri

9. Koristeći lijevo vanjsko spajanje dohvatiti imena svih kupaca koji nisu ništa kupili.
10. Koristeći desno vanjsko spajanje dohvatiti imena svih kupaca koji nisu ništa kupili.
11. Koristeći puno vanjsko spajanje ispisati nazive država i nazive gradova.
12. Koristeći puno vanjsko spajanje ispisati nazive država i nazive gradova. Ispisati samo one gradove koji nemaju definiranu državu i one države koji nemaju upisanih gradova.

# Unakrsno spajanje

- Unakrsno spajanje ima **dva načina rada**:
  - Ako ne postoji `WHERE` uvjet, unakrsno spajanje je Kartezijev produkt relacija koje sudjeluju u spajanju
    - **Kartezijev produkt**: svaki redak iz prve tablice se spaja sa svakim retkom iz druge tablice
  - Ako postoji definiran `WHERE` uvjet, unakrsno spajanje se ponaša kao **unutarnje spajanje**
- Koristi se naredba **CROSS JOIN** bez **ON** dijela
- Primjer: napraviti unakrsno spajanje gradova i država
- Primjer: napraviti unakrsno spajanje gradova i država uz uvjet da su im ID-evi država jednaki