

# NULL VRIJEDNOSTI

VIII auditorne vježbe

# Teme

---

- ▶ **NULL vrijednosti**
  - ▶ IS NULL operator
  - ▶ Uslovi selekcije i spajanja u prisustvu NULL vrijednosti
  - ▶ Agregatne funkcije i NULL vrijednosti
  - ▶ Grupisanje i NULL vrijednosti
  - ▶ Podupiti i NULL vrijednosti
  - ▶ Poredak rezultata upita u prisustvu NULL vrijednosti
  - ▶ NULL konstanta
- ▶ **Vanjsko spajanje**
  - ▶ Djelovanje uslova selekcije u upitima sa vanjskim spajanjem
  - ▶ Vanjsko spajanje s tri ili više relacija



# IS NULL operator

---

- ▶ Koristi se za ispitivanje da li je vrijednost nekog atributa postavljena na NULL vrijednost.

`column IS NULL` ili `column IS NOT NULL`

- ▶ Rezultat ispitivanja je TRUE ili FALSE

ISPLATA

<b>sifra</b>	<b>datum</b>	<b>koef</b>	<b>iznos</b>
NULL	NULL	0.5	50
1	01.01.2006	1.0	100
NULL	01.01.2008	1.5	200
1	01.01.2007	NULL	250
NULL	NULL	2.0	NULL
2	01.01.2007	2.5	300
2	NULL	NULL	NULL
2	NULL	3.0	300



# IS NULL operator

---

```
SELECT * FROM isplata  
WHERE koef IS NULL
```

<b>sifra</b>	<b>datum</b>	<b>koef</b>	<b>iznos</b>
1	01.01.2007	NULL	250
2	NULL	NULL	NULL



# Uslovi selekcije i spajanja u prisustvu NULL vrijednosti

---

- ▶ Sve n-torke čije vrijednosti atributa uvrštene u uslov za dohvat ili uslov spajanja daju rezultat TRUE, kvalifikuju se kao rezultat SELECT naredbe.
- ▶ n-torke za koje se uslov za dohvat ili uslov za spajanje evaluira kao FALSE ili UNKNOWN, ne ulaze u rezultat.
- ▶ Isto vrijedi i u naredbama UPDATE ili DELETE

```
SELECT * FROM isplata  
WHERE koef*iznos <> 100
```

<b>sifra</b>	<b>datum</b>	<b>koef</b>	<b>iznos</b>
NULL	NULL	0.5	50
NULL	01.01.2008	1.5	200
2	01.01.2007	2.5	300
2	NULL	3.0	300



# Agregatne funkcije i NULL vrijednosti

---

- ▶ Pri korištenju agregatnih funkcija izračunavaju se slijedeće vrijednosti:
- ▶ **COUNT(\*)** - broji n-torke, pojava NULL vrijednosti u n-torki ne utiče na rezultat
- ▶ **COUNT(x)** - broj n-torki u kojima atribut x ima vrijednost različitu od NULL
- ▶ **COUNT(DISTINCT x)** - broj n-torki sa različitim vrijednostima od x (n-troke čija je vrijednost atributa x jednaka NULL vrijednosti se ne broje)
- ▶ Pri korištenju ostalih agregatnih funkcija, bez obzira koristi li se ključna riječ DISTINCT, funkcije ignorišu n-torke u kojima atribut x poprima vrijednost NULL i izračunavaju rezultat na temelju ostalih vrijednosti, osim ako je vrijednost x postavljena na NULL za sve n-torke: tada je rezultat NULL.

- ▶ **Primjeri:**

<code>SELECT COUNT(*) FROM isplata</code>	→ 8
<code>SELECT COUNT(iznos) FROM isplata</code>	→ 6
<code>SELECT COUNT(DISTINCT sifra) FROM isplata</code>	→ 2
<code>SELECT COUNT(DISTINCT iznos) FROM isplata</code> <code>WHERE sifra = 2</code>	→ 1
<code>SELECT SUM(iznos) FROM isplata</code>	→ 1200.00
<code>SELECT SUM(iznos) FROM isplata</code> <code>WHERE koef = 2.0</code>	→ NULL



# Grupisanje i NULL vrijednosti

---

- Ukoliko vrijednosti izraza prema kojima se obavlja grupisanje, za neke dvije n-torke predstavljaju međusobnu kopiju, te dvije n-torke ulaze u istu grupu.

```
SELECT sifra, datum, COUNT(*) FROM isplata  
      GROUP BY sifra, datum
```

<b>sifra</b>	<b>datum</b>	<b>(count (*))</b>
NULL	NULL	2
1	01.01.2006	1
NULL	01.01.2008	1
1	01.01.2007	1
2	01.01.2007	1
2	NULL	2



# Podupiti i NULL vrijednosti

---

- ▶ Naročitu pažnju pri korištenju podupita čiji rezultat može sadržavati NULL vrijednosti treba obratiti na uslove za dohvat koji imaju jedan od ova dva oblika:

WHERE expression relationalOperator ALL (subquery) ili  
WHERE expression NOT IN (subquery)

ISPLATA

<b>sifra</b>	<b>datum</b>	<b>koef</b>	<b>iznos</b>
NULL	NULL	0.5	50
1	01.01.2006	1.0	100
NULL	01.01.2008	1.5	200
1	01.01.2007	NULL	250
NULL	NULL	2.0	NULL
2	01.01.2007	2.5	300
2	NULL	NULL	NULL
2	NULL	3.0	300

UPLATA

<b>sifra</b>	<b>datum</b>	<b>koef</b>	<b>iznos</b>
1	01.01.2001	0.8	400
NULL	01.01.2004	0.9	150
4	01.01.2006	1.5	NULL





# Podupiti i NULL vrijednosti

---

```
SELECT * FROM uplata
      WHERE iznos > ANY (SELECT iznos FROM isplata)
```

<b>sifra</b>	<b>datum</b>	<b>koef</b>	<b>iznos</b>
1	01.01.2001	0.8	400
NULL	01.01.2004	0.9	150

```
SELECT * FROM uplata
      WHERE iznos > ALL (SELECT iznos FROM isplata)
```

**nema zapisa**

```
SELECT * FROM uplata
      WHERE sifra IN (SELECT sifra FROM isplata)
```

<b>sifra</b>	<b>datum</b>	<b>koef</b>	<b>iznos</b>
1	01.01.2001	0.8	400

```
SELECT * FROM uplata
      WHERE sifra NOT IN (SELECT sifra FROM isplata)
```

**nema zapisa**

---



# Poredak rezultata upita u prisustvu NULL vrijednosti

---

- ▶ NULL vrijednost je manja pri poretku od svake druge vrijednosti. Zbog toga pri sortiranju rezultata u uzlaznom nizu NULL vrijednost dolazi prije svih ostalih vrijednosti. Pri sortiranju rezultata u silaznom nizu, NULL vrijednost dolazi iza svih ostalih vrijednosti.

```
SELECT * FROM isplata  
ORDER BY sifra
```

<b>sifra</b>	<b>datum</b>	<b>koef</b>	<b>iznos</b>
NULL	NULL	0.5	50
NULL	01.01.2008	1.5	200
NULL	NULL	2.0	NULL
1	01.01.2006	1.0	100
1	01.01.2007	NULL	250
2	01.01.2007	2.5	300
2	NULL	NULL	NULL
2	NULL	3.0	300



# Vanjsko spajanje

---

- ▶ Kod prirodnog spajanja se relacije koje se spajaju tretiraju «simetrično» (jednako su «vrijedne»).

STUD

<u>mbr</u>	<u>prezime</u>	<u>pbr</u>
1	Mehanović	75000
2	Mićanović	71000
3	Marković	72000

MJESTO

<u>pbr</u>	<u>nazMjesto</u>
75000	Tuzla
72000	Zenica
88000	Mostar

```
SELECT stud.*, mjesto.* FROM stud INNER JOIN mjesto
      ON stud.pbr = mjesto.pbr
```

<u>mbr</u>	<u>prezime</u>	<u>pbr</u>	<u>pbr</u>	<u>nazMjesto</u>
1	Mehanović	75000	75000	Tuzla
3	Marković	72000	72000	Zenica



# Vanjsko spajanje

---

- ▶ Kod vanjskog spajanja relacije se tretiraju nesimetrično. Jedna je dominantna, te se njene n-torke pojavljuju u rezultatu bez obzira postoji li odgovarajuća n-torka u podređenoj relaciji.

```
SELECT stud.*, mjesto.* FROM stud  
      LEFT OUTER JOIN mjesto ON stud.pbr = mjesto.pbr
```

<b>mbr</b>	<b>prezime</b>	<b>pbr</b>	<b>pbr</b>	<b>nazMjesto</b>
1	Mehanović	75000	75000	Tuzla
2	Mićanović	71000	NULL	NULL
3	Marković	72000	72000	Zenica



# Vanjsko spajanje

---

- ▶ U prethodnom upitu stud je dominantna relacija. Kod vanjskog spajanja relacije se tretiraju nesimetrično:
  - ▶ dominantna (*dominant, preserved*) – njene se n-torke (ukoliko zadovoljavaju uslov za dohvat) pojavljuju u rezultatu bez obzira postoji li odgovarajuća n-torka u podređenoj relaciji.
  - ▶ podređena (*subservient*) – njezine se n-torke pojavljuju jedino u slučaju kada zadovoljavaju uslov spajanja i uslov za dohvat (ukoliko uslov za dohvat postoji)
- ▶ Ukoliko za n-torku dominantne relacije ne postoji odgovarajuća n-torka iz podređene relacije, u rezultatu se kao vrijednosti atributa podređene relacije pojavljuju NULL vrijednosti.



# Djelovanje uslova selekcije u upitima sa vanjskim spajanjem

---

## ► **Bez uslova selekcije:**

```
SELECT stud.*, mjesto.* FROM stud  
      LEFT OUTER JOIN mjesto ON stud.pbr = mjesto.pbr
```

<u>mbr</u>	<u>prezime</u>	<u>pbr</u>	<u>pbr</u>	<u>nazMjesto</u>
1	Mehanović	75000	75000	Tuzla
2	Mićanović	71000	NULL	NULL
3	Marković	72000	72000	Zenica



# Djelovanje uslova selekcije u upitima sa vanjskim spajanjem

---

## ► **Uslov selekcije postavljen na dominantnu relaciju**

```
SELECT stud.*, mjesto.* FROM stud  
    LEFT OUTER JOIN mjesto ON stud.pbr = mjesto.pbr  
WHERE stud.mbr > 1
```

<u>mbr</u>	<u>prezime</u>	<u>pbr</u>	<u>pbr</u>	<u>nazMjesto</u>
2	Mićanović	71000	NULL	NULL
3	Marković	72000	72000	Zenica



# Djelovanje uslova selekcije u upitima sa vanjskim spajanjem

---

- ▶ ***Uslov selekcije postavljen na podređenu relaciju***
- ▶ MySQL konvertuje ovakve upite u upit bez vanjskog spajanja ukoliko je WHERE uslov *NULL-rejected* (evaluira se u FALSE ili UNKNOWN za svaki red koji sadrži NULL vrijednosti koji se formira u operaciji vanjskog spajanja)
- ▶ Opšte pravilo provjere da li je uslov *NULL-rejected* za operaciju vanjskog spajanja je vrlo jednostavno. Uslov je *NULL-rejected* u sljedećim slučajevima:
  - ▶ Ukoliko je u formi *A IS NOT NULL* gdje je *A* atribut bilo koje *inner* relacije
  - ▶ Ukoliko je predikat koji sadrži referencu na *inner* relaciju koji se evaluira kao UNKNOWN kada je jedan od njegovih argumenata NULL
  - ▶ Ukoliko je AND operacija koja sadrži *NULL-rejected* uslov
  - ▶ Ukoliko je OR operacija *NULL-rejected* operacija





# Djelovanje uslova selekcije u upitima sa vanjskim spajanjem

---

- ▶ Zbog toga će upit:

```
SELECT stud.*, mjesto.* FROM stud
  LEFT OUTER JOIN mjesto ON stud.pbr = mjesto.pbr
 WHERE mjesto.pbr > 72000
```

- ▶ Dati sljedeći rezultat:

<b>mbr</b>	<b>prezime</b>	<b>pbr</b>	<b>pbr</b>	<b>nazMjesto</b>
1	Mehanović	75000	75000	Tuzla

- ▶ Ali će upit:

```
SELECT stud.*, mjesto.* FROM stud
  LEFT OUTER JOIN mjesto ON stud.pbr = mjesto.pbr
 WHERE mjesto.pbr > 72000 OR mjesto.pbr IS NULL
```

- ▶ Dati pravi rezultat vanjskog spajanja:

<b>mbr</b>	<b>prezime</b>	<b>pbr</b>	<b>pbr</b>	<b>nazMjesto</b>
1	Mehanović	75000	75000	Tuzla
2	Mićanović	71000	NULL	NULL
3	Marković	72000	NULL	NULL



# Vanjsko spajanje s tri ili više relacija

---

- ▶ Postoje tri osnovna oblika spajanja:
  - a) neka su zadane relacije S, R i T. N-torke iz relacije R vanjski se spajaju sa rezultatom «običnog» spajanja relacija S i T

```
SELECT * FROM R LEFT OUTER JOIN  
    (S INNER JOIN T ON S ⋈ T ) ON R ⋈ S
```

- ▶ U rezultatu će se pojaviti sve n-torke iz relacije R, a uz njih n-torke iz S i T ukoliko postoji odgovarajući rezultat spajanja relacija S i T.



# Vanjsko spajanje s tri ili više relacija

---

AVION		PILOT		LET	
tip	dolet	prezime	tip	brLet	udaljenost
B747	6000	JOHNSON	B747	CA-825	7200
DC9	3000	SMITH	AIRBUS	A-224	3300
B727	4500	WALKER	F16	CA-878	4200
AIRBUS	1800				

```
SELECT * FROM pilot LEFT OUTER JOIN  
    (avion INNER JOIN let ON avion.dolet >= udaljenost)  
    ON pilot.tip = avion.tip
```

Prezime	tip	tip	dolet	brLeta	udaljenost
JOHNSON	B747	B747	6000	A-224	3300
JOHNSON	B747	B747	6000	CA-878	4200
SMITH	AIRBUS	NULL	NULL	NULL	NULL
WALKER	F16	NULL	NULL	NULL	NULL



## Vanjsko spajanje s tri ili više relacija

---

- b) neka su zadane relacije S, R i T. N-torke iz R vanjski se spajaju sa n-tokama iz S, te se nezavisno o tom spajanju, spajaju sa n-torkama iz T

```
SELECT * FROM R LEFT OUTER JOIN S R ⋈ S  
      LEFT OUTER JOIN T ON R ⋈ T
```

- U rezultatu će se pojaviti sve n-torke iz relacije R, a uz svaku takvu n-torku pojaviće se n-torka iz relacije S (ukoliko u relaciji S postoji odgovarajuća), te n-torka iz relacije T (ukoliko postoji odgovarajuća n-torka u relaciji T).



# Vanjsko spajanje s tri ili više relacija

---

```
SELECT * FROM avion LEFT OUTER JOIN pilot  
  ON pilot.tip = avion.tip  
  LEFT OUTER JOIN let ON avion.dolet >= udaljenost
```

<b>tip</b>	<b>dolet</b>	<b>prezime</b>	<b>tip</b>	<b>brLet</b>	<b>udaljenost</b>
B747	6000	JOHNSON	B747	A-224	3300
B747	6000	JOHNSON	B747	CA-878	4200
DC9	3000	NULL	NULL	NULL	NULL
B727	4500	NULL	NULL	A-224	3300
B727	4500	NULL	NULL	CA-878	4200
AIRBUS1800		SMITH	AIRBUS	NULL	NULL



## Vanjsko spajanje s tri ili više relacija

---

- c) neka su zadane relacije S, R i T. N-torke iz relacije R vanjski se spajaju sa rezultatom vanjskog spajanja relacija S i T

```
SELECT * FROM R LEFT OUTER JOIN  
    (S LEFT OUTER JOIN T ON S  $\bowtie$  T)  
    ON R  $\bowtie$  S
```

- U rezultatu će se pojaviti sve n-torke iz relacije R, a uz njih n-torke iz S (ukoliko postoji odgovarajuća n-torka u S). Pri tome će se uz n-toku iz relacije S pojaviti i n-toka iz relacije T (ukoliko postoji odgovarajuća n-toka u relaciji T)



# Vanjsko spajanje s tri ili više relacija

---

```
SELECT * FROM pilot LEFT OUTER JOIN  
    (avion LEFT OUTER JOIN let  
    ON avion.dolet >= udaljenost)  
    ON pilot.tip = avion.tip
```

<b>prezime</b>	<b>tip</b>	<b>tip</b>	<b>dolet</b>	<b>brLeta</b>	<b>udaljenost</b>
JOHNSON	B747	B747	6000	A-224	3300
JOHNSON	B747	B747	6000	CA-878	4200
SMITH	AIRBUS	AIRBUS	1800	NULL	NULL
WALKER	F16	NULL	NULL	NULL	NULL

