

SQL

# Stavki skupine SQL DML

---

- **DML skupina** zajema SQL stavke za manipulacijo s podatki
  - SELECT → Izbira
  - INSERT → Dodajanje
  - DELETE → Brisanje
  - UPDATE → Spreminjanje
- Sintaksa SELECT stavka najbolj kompleksna

# SELECT stavek...

---

SELECT [DISTINCT | ALL]

{\* | [columnExpression [AS newName]] [,...]} }

FROM            TableName [alias] [, ...]

[WHERE           condition]

[GROUP BY       columnList] [HAVING           condition]

[ORDER BY       columnList]

# SELECT stavek...

---

- **SELECT** Določa stolpce, ki naj se pojavijo v izhodni relaciji
- **FROM** Določa tabele za poizvedbo
- **WHERE** Filtrira vrstice
- **GROUP BY** Združuje vrstice po vrednostih izbranih stolpcev
- **HAVING** Filtrira skupine glede na določene pogoje
- **ORDER BY** Določa vrstni red vrstic na izhodu

Vrstnega reda sklopov ni možno spreminjati!  
Obvezna sta samo SELECT in FROM sklopa!

# SELECT stavek - primer

---

- Izpiši vsa naročila – vse podatke o vseh naročilih:

```
SELECT *  
FROM Naročila
```

- Izpiši vsa naročila – samo podatke o številu naročila in šifri kupca:

```
SELECT [Številka naročila], [Šifra kupca]  
FROM Naročila
```

# Uporaba DISTINCT

---

- Izpiši izdelke, ki nastopajo v naročilih – brez dvojnikov

```
SELECT DISTINCT [Šifra izdelka]  
FROM Postavke
```



Ukaz DISTINCT odstrani dvojnike

Naročila

# Izračunana polja

---

- Izpiši ceno postavke v EUR

```
SELECT [Šifra izdelka] AS izdelek,  
       cena AS cena_SIT,  
       cena/239 AS cena_EUR
```

```
FROM Postavke
```

Uporabljamo formule

Izračunanemu stolpcu  
dodelimo naziv

# Iskalni kriteriji

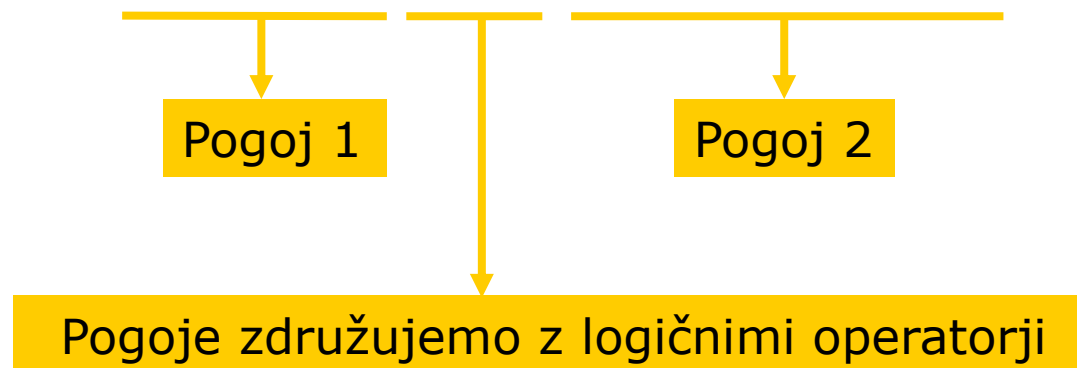
---

- Izpiši vse postavke naročil, ki imajo ceno večjo od 20.000 SIT in količino manjšo od 10

```
SELECT *
```

```
FROM Postavke
```

```
WHERE cena > 20000 AND količina < 10
```



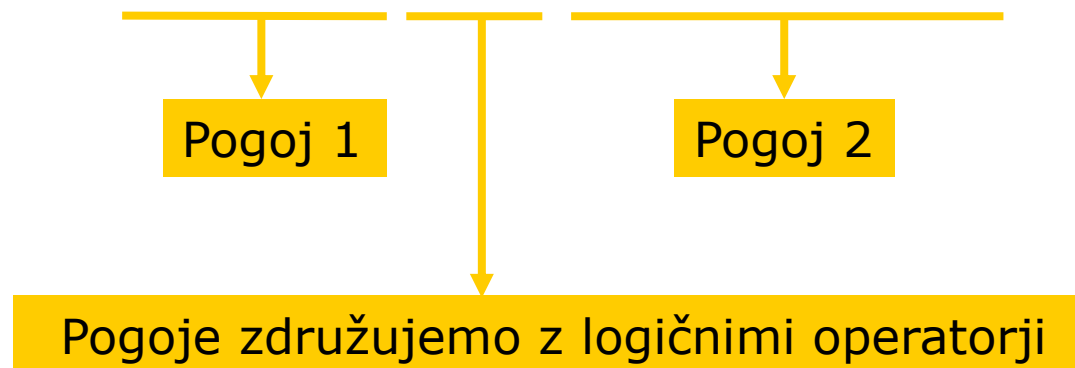


# Iskalni kriteriji

---

- Izpiši vsa naročila, kjer sta datum naročila in rok plačila enaka

```
SELECT *  
FROM Naročila  
WHERE datum=[rok plačila]
```



# Iskanje z uporabo BETWEEN

---

- Izpiši vse postavke s ceno med 10.000 in 20.000 SIT

```
SELECT *  
FROM Postavke  
WHERE cena BETWEEN 10000 AND 20000
```

BETWEEN vključuje spodnjo in zgornjo mejo!

Uporabimo lahko tudi negacijo NOT BETWEEN

BETWEEN ne doda veliko SQL moči, možno izraziti posredno

# Iskanje po članstvu množice

---

- Izpiši naročila za kupce 100 in 300

```
SELECT *  
FROM Naročila  
WHERE [šifra kupca] IN ( '100' , '300' )
```



Članstvo množice

Uporabimo lahko tudi negacijo NOT IN

IN ne doda veliko SQL moči; koristno pri večjih množicah

# Iskanje z vzorcem

---

- Izpiši vse postavke, kjer šifra izdelka začne z 2

```
SELECT *  
FROM Postavke  
WHERE [šifra izdelka] LIKE '2%'
```



Iskanje z vzorcem

SQL ima dva posebna znaka za iskanje z vzorcem:

Znak **%** nadomešča katerikoli niz znakov

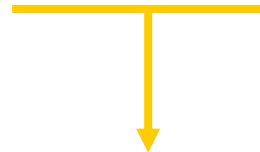
Znak **\_** nadomešča katerikoli znak

# Iskanje z NULL vrednostjo v pogoju

---

- Izpiši vse delavce, ki nimajo opombe

```
SELECT *  
FROM Delavci  
WHERE opombe IS NULL
```



Iskanje z NULL vrednostjo

Uporabljamo lahko tudi negacijo **IS NOT NULL**

# Sortiranje vrstic v izhodni relaciji

---

- Izpiši vse podatke o delavcih, urejene po priimku delavca

```
SELECT *  
FROM Delavci  
ORDER BY priimek ASC
```

ASC – ascending → naraščujoče

DESC – descending → padajoče

# Naloge – baza Prodaja (1)

---

- Napiši poizvedbo, ki:
  - vrne (izpiše) vse delavce in sicer:
    - vse podatke o delavcih
    - samo naslednje podatke: ime, priimek, spol, ulica
  - vrne vse delavce, ki delajo na del.mestu s šifro 10
  - vrne vsa naročila za kupca s šifro 300
  - vrne vse postavke s šifro izdelka 20002
  - vrne vse postavke s šifro izdelka 20002, urejene padajoče po znesku
  - vrne vse delavce in številke mobilnih telefonov – tiste, ki imajo številke
  - vrne vse delavce pri katerih jim ime začne s črko M
  - vrne vse delavce, ki delajo v oddelkih 10 ali 30
  - vrne število zapisov v postavkah naročil
  - vrne kateri izdelki so se pojavili v postavkah naročil (samo enkrat!)
  - vrne povprečje vrednosti postavk

# Naloge – baza Prodaja (1)

---

- ...vrne vse delavce, in sicer vse podatke o delavcih

```
SELECT * FROM Delavci;
```



# Naloge – baza Prodaja (1)

---

- ...vrne vse delavce, ampak samo naslednje podatke:  
ime, priimek, spol, ulica

```
SELECT Ime, Priimek, Spol, Ulica FROM Delavci;
```

# Naloge – baza Prodaja (1)

---

- ...vrne vse delavce, ki delajo na del. mestu s šifro 10

```
SELECT * FROM Delavci WHERE Delovno_mesto = 10;
```

# Naloge – baza Prodaja (1)

---

- ...vrne vsa naročila za kupca s šifro 300

```
SELECT * FROM Naročila WHERE Šifra_kupca = 300;
```

# Naloge – baza Prodaja (1)

---

- ...vrne vse postavke s šifro izdelka 20002

```
SELECT * FROM Postavke WHERE Šifra_izdelka = 20002;
```

# Naloge – baza Prodaja (1)

---

- ...vrne vse postavke s šifro izdelka 20002, urejene padajoče po znesku

```
SELECT * FROM Postavke WHERE Šifra_izdelka = 20002  
ORDER BY Cena DESC;
```

# Naloga – baza Prodaja (1)

---

- ...vrne vse delavce in številke mobilnih telefonov (samo tiste, ki imajo številke)

```
SELECT Priimek, Ime, Mobilni_telefon FROM Delavci  
WHERE Mobilni_telefon NOT NULL;
```

# Naloge – baza Prodaja (1)

---

- ...vrne vse delavce, katerih ime se začne s črko M

```
SELECT * FROM Delavci WHERE Ime LIKE 'M%';
```

# Naloge – baza Prodaja (1)

---

- ...vrne vse delavce, ki delajo v oddelkih 10 ali 30

```
SELECT * FROM Delavci WHERE Oddelek IN (10, 30);
```



# Naloge – baza Prodaja (1)

---

- ...vrne število zapisov v postavkah naročil

```
SELECT COUNT(*) FROM Postavke;
```

# Naloge – baza Prodaja (1)

---

- ...vrne (samo enkrat) kateri izdelki so se pojavili v postavkah naročil

```
SELECT DISTINCT Šifra_izdelka FROM Postavke;
```

# Naloge – baza Prodaja (1)

---

- ...vrne povprečje vrednosti postavk

```
SELECT AVG(Cena) FROM Postavke;
```

# Agregacija podatkov...

---

- Standard ISO opredeljuje pet agregacijskih operacij:
  - COUNT vrne število vrednosti v določenem stolpcu
  - SUM vrne seštevek vrednosti v določenem stolpcu
  - AVG vrne povprečje vrednosti v določenem stolpcu
  - MIN vrne najmanjšo vrednost v določenem stolpcu
  - MAX vrne največjo vrednost v določenem stolpcu
- Vse operacije delujejo na enem stolpcu in vračajo eno samo vrednost.

# Agregacija podatkov...

---

- COUNT, MIN in MAX se uporabljajo za numerične in ne-numerične vrednosti, SUM in AVG zahtevata numerične vrednosti.
- Vse operacije razen COUNT(\*) najprej odstranijo vrstice z vrednostjo NULL v stolpcu, po katerem agregiramo.
- COUNT(\*) prešteje vse vrstice, ne glede na vrednosti NULL ali podvojene vrednosti.

# Agregacija podatkov...

---

- Če se želimo znebiti podvojenih vrednosti, uporabimo DISTINCT pred imenom stolpca.
- DISTINCT nima učinka na MIN/MAX, lahko pa vpliva na SUM/AVG.
- Agregacijske operacije lahko uporabimo le v SELECT ali HAVING sklopu.

# Agregacija podatkov

---

- SELECT AVG(Cena) FROM Postavke;
- ~~• SELECT [Šifra izdelka], AVG(Cena) FROM Postavke;~~

Napačna raba agregacije

- SELECT [Šifra izdelka], AVG(Cena) FROM Postavke GROUP BY [Šifra izdelka];

# Uporaba COUNT in DISTINCT

---

- Za koliko različnih izdelkov velja, da jih je bilo prodanih več kot 10 (v sklopu posameznega naročila)?

```
SELECT COUNT (DISTINCT [Šifra izdelka]) AS  
Števílo FROM Postavke WHERE Količina > 10;
```



# Uporaba več agregatov istočasno

---

- Agregati se lahko uporabljajo tudi istočasno.
- Npr: izpiši povprečno, minimalno in maksimalno ceno postavke naročila:

```
SELECT AVG(Cena), MIN(Cena), MAX(Cena)  
FROM Postavke;
```

# Združevanje podatkov...

- Sklop **GROUP BY** uporabimo za združevanje podatkov v skupine.
- **SELECT** in **GROUP BY** sta tesno povezana

Šifra izdelka	Cena
20001	39.900,00
20001	29.000,00
20001	25.000,00
20002	17.900,00
20002	13.500,00

Šifra izdelka	Avg(Cena)
20001	31.300,00
20002	15.700,00

# Združevanje podatkov...

---

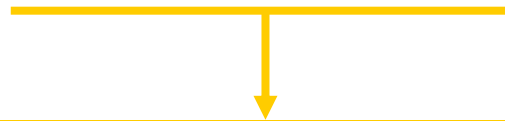
- Vsi stolpci, ki so navedeni v SELECT sklopu, se morajo nahajati tudi v GROUP BY sklopu, razen tistih, ki nastopajo samo v agregacijskih operacijah.
- Če uporabljamo WHERE sklop v kombinaciji z GROUP BY, se WHERE upošteva najprej, združevanje pa se izvede na preostalih vrsticah.
- Standard ISO smatra vrednosti NULL kot enake, ko gre za združevanje.

# Primer združevanja (grupiranja)

---

- Izpiši število pojavljanj izdelkov v postavkah

```
SELECT [Šifra izdelka], COUNT([Šifra izdelka])  
FROM Postavke  
GROUP BY [Šifra izdelka]
```



Za vsako šifro izdelka, vrne število pojavljanj v postavkah

# Omejitev skupin

---

- Sklop **HAVING** je namenjen uporabi v kombinaciji z GROUP BY kot omejitev skupin, ki se lahko pojavijo v rezultatu.
- Deluje podobno kot WHERE:
  - WHERE filtrira posamezne vrstice,
  - HAVING filtrira skupine.
- Stolpci, ki so navedeni v sklopu HAVING, morajo biti tudi v sklopu SELECT ali v agregatih.

# Uporaba sklopa HAVING

---

- Izpiši izdelke, ki so bili na več kot enem naročilu.

```
SELECT [Šifra izdelka]
FROM Postavke
GROUP BY [Šifra izdelka]
HAVING COUNT([Šifra izdelka]) > 1;
```

# Naloge – baza Prodaja (2)

---

Napiši poizvedbo, ki ...

# Naloge – baza Prodaja (2)

---

- ...izpiše, koliko naročil je imel posamezni kupec.

```
SELECT [Šifra kupca], COUNT(*)  
FROM Naročila  
GROUP BY [Šifra kupca];
```



# Naloga – baza Prodaja (2)

---

- ...izpiše, koliko naročil z gotovinskim plačilom je imel posamezni kupec.

```
SELECT [Šifra kupca], COUNT(*)  
FROM Naročila  
WHERE [Način plačila] = 'Gotovina'  
GROUP BY [Šifra kupca];
```

# Naloga – baza Prodaja (2)

---

- ...izpiše kupce z več kot enim naročilom, ki je bilo plačano z naročilnico.

```
SELECT [Šifra kupca], COUNT(*)  
FROM Naročila  
WHERE [Način plačila] = 'Naročilnica'  
GROUP BY [Šifra kupca]  
HAVING COUNT(*) > 1;
```

# Naloge – baza Prodaja (2)

---

- ...izpiše povprečno, minimalno in maksimalno ceno izdelkov (iz šifranta izdelkov).

```
SELECT AVG(Cena), MIN(Cena), MAX(Cena)
FROM Izdelki;
```

# Naloge – baza Prodaja (2)

---

- Napiši poizvedbo, ki:
  - izpiše število delavcev, ki imajo oz. nimajo mobilni telefon
  - izpiše, koliko delavcev dela na posameznem delovnem mestu
  - izpiše delovna mesta, kjer dela več kot en delavec
  - izpiše, koliko delavcev dela na posameznem oddelku
  - izpiše, koliko je naročil imel posamezni delavec, kjer je način plačila naročilnica
  - izpiše, koliko je naročil imel posamezni delavec za posameznega kupca

# Poizvedbe po več tabelah...

- V poizvedbah, ki vračajo stolpce različnih tabel, moramo uporabljati **stik** (*ang. join*).
- Stik izvedemo tako, da v sklopu FROM navedemo tabele, v sklopu WHERE pa določimo stolpce za stik.  
Namesto stika z WHERE se priporoča uporaba INNER JOIN.
- Za tabele v razdelku FROM lahko uvedemo sinonime (alias).

- Sintaksa:

```
SELECT nar.[šifra kupca], pos.[šifra izdelka]  
FROM naročila nar, postavke pos  
...
```

Sinonim za tabelo  
**Postavke**

- Sinonime uporabljamo za ločevanje med istoimenskimi stolpci različnih tabel (lahko pa uporabimo kar imena tabel).

# Primer poizvedbe po dveh tabelah

---

- Izpiši postavke naročil za kupca s šifro 100:

```
SELECT nar.[Številka naročila],  
       nar.[Šifra kupca],  
       pos.[Šifra izdelka],  
       pos.Količina,  
       pos.Cena  
FROM Naročila nar, Postavke pos  
WHERE pos.[Številka naročila] = nar.[Številka naročila]  
       AND nar.[Šifra kupca] = 100;
```

Kako pa to zapišemo z INNER JOIN?

# Primer poizvedbe po več tabelah

---

- Izpiši nazive in e-naslove kupcev, ki so naročili vsaj 5 izdelkov na naročilu. Izpiši še šifre teh izdelkov.

```
SELECT kup.[Šifra kupca],  
       kup.[E-mail],  
       pos.[Šifra izdelka]  
FROM Kupci kup, Naročila nar, Postavke pos  
WHERE kup.[Šifra kupca] = nar.[Šifra kupca]  
      AND nar.[Številka naročila] = pos.[Številka naročila]  
      AND pos.Količina >= 5;
```

Kako pa to zapišemo z INNER JOIN?

# Naloge – baza Prodaja (3)

---

- Napiši poizvedbo, ki:
  - izpiše vse podatke o postavkah in poleg še naziv izdelka
  - izpiše vse podatke o naročilih ter poleg še ime in priimek delavca
  - izpiše matične številke, priimek in ime delavcev in poleg še naziv njihovega delovnega mesta
  - izpiše številko naročila, datum in način plačila, poleg pa še naziv kupca
  - izpiše nazive kupcev, ki imajo samo eno naročilo
  - izpiše, koliko naročil je imel posamezni kupec (izpiše naj naziv kupca)
  - izpiše vse podatke o postavkah naročil za kupca z id. št. 300
  - izpiše vse kraje, kjer bodisi živi delavec ali ima sedež kupec, urejeno padajoče



# Naloge – baza Prodaja (3)

---

- Napiši poizvedbo, ki:
  - izpiše številke naročil, v katerih je prodan izdelek s šifro 20002
  - izpiše nazive kupcev, ki so naročili izdelek 20002
  - izpiše za vsako postavko naziv izdelka in skupno vrednost postavke
  - izpiše matične številke delavcev, ki imajo naročila
  - izpiše matične številke delavcev, ki imajo na postavkah svojih naročil več kot en izdelek
  - izpiše, koliko naročil je imel posamezni delavec
  - izpiše, koliko postavk je imel posamezni delavec (na vseh naročilih skupaj)
  - izpiše, koliko izdelkov je imel posamezni delavec (na vseh postavkah vseh naročil skupaj)

# Alternativni načini stika več tabel

---

- SQL omogoča alternativne načine stika med več tabelami:
  - **LEFT (OUTER) JOIN** – vse iz leve tabele, iz desne tiste zapise, ki ustrezajo pogojem:

```
SELECT izd.[šifra izdelka], izd.[naziv izdelka]
  FROM izdelki izd LEFT OUTER JOIN postavke pos ON izd.[šifra izdelka] =
  pos.[šifra izdelka]
```
  - **RIGHT (OUTER) JOIN** – vse iz desne tabele, iz leve tiste, ki ustrezajo pogojem:

```
SELECT del.priimek, p.[naziv pošte]
  FROM delavci del RIGHT OUTER JOIN pošte p ON del.[poštna številka] =
  p.[poštna številka]
```
  - **FULL (OUTER) JOIN** – vse iz leve in iz desne tabele (kombinacija **LEFT** in **RIGHT JOIN**).

# Gnezdenje poizvedb...

---

- Nekateri SQL stavki imajo lahko **ugnezdene stavke SELECT**.
- Ugnezdeni stavki SELECT se lahko uporabijo v sklopih WHERE ali HAVING drugega stavka SELECT (**subselect**).
- Ugnezdeni stavki SELECT se lahko pojavijo tudi v stavkih INSERT, UPDATE in DELETE.

# Primer ugnezdenega stavka SELECT

---

- Izpiši nazive kupcev, ki imajo vsaj 2 naročili

```
SELECT naziv
FROM kupci
WHERE [šifra kupca] IN
      (SELECT [šifra kupca]
       FROM naročila
       GROUP BY [šifra kupca]
       HAVING COUNT(*) >= 2)
```

Kaj moramo  
spremeniti, da  
vidimo zraven še  
šifro kupca?

# Pravila gnezdenja stavkov SELECT...

---

- Ugnezdeni stavki SELECT ne smejo uporabljati sklopa ORDER BY.
- Sklop SELECT ugnezdenega stavka SELECT lahko zajema samo en stolpec, razen v primeru uporabe ukaza **EXISTS**.
- Imena stolpcev v ugnezdenem stavku SELECT se privzeto nanašajo na tabele iz ugnezdenega ali zunanjega stavka SELECT.
- Ko je ugnezdeni stavek SELECT operand v primerjavi, se mora nahajati na desni strani enačbe.

# Uporaba EXISTS in NOT EXISTS

---

- EXISTS in NOT EXISTS lahko uporabljamo le v ugnezdenih poizvedbah.
- Vračajo logičen rezultat pravilno (angl. true)/nepravilno (angl. false).
  - True dobimo, če obstaja vsaj ena vrstica v tabeli, ki je rezultat ugnezdene poizvedbe.
  - False dobimo, če ugnezdena poizvedba vrača prazno množico.
- NOT EXISTS je negacija EXISTS.

# Uporaba EXISTS in NOT EXISTS

---

- (NOT) EXISTS preveri samo, če v rezultatu ugnezdene poizvedbe (ne) obstajajo vrstice.
- Število stolpcev v sklopu SELECT ugnezdene poizvedbe je zato irelevantno.
- Navadno uporabimo sintakso:  
(SELECT \* ...)

# Primer uporabe EXISTS

---

- Izpiši vse kupce, ki so kdaj kaj naročili:

```
SELECT [šifra kupca]
  FROM kupci kup
 WHERE EXISTS
   (SELECT *
     FROM naročila nar
    WHERE nar.[šifra kupca] = kup.[šifra kupca])
```

- Namesto EXISTS lahko uporabimo stik:

```
SELECT DISTINCT nar.[šifra kupca]
  FROM kupci kup, naročila nar
 WHERE nar.[šifra kupca] = kup.[šifra kupca]
```



# Uporaba operacij nad množicami

---

- Rezultate dveh ali več poizvedb lahko združujemo z ukazi:
  - **Union** (unija),
  - **Intersection** (presek)
  - **Difference (EXCEPT)** (razlika)
- Da lahko izvajamo naštetе operacije, morata tabeli A in B biti **skladni** (domene atributov morajo biti enake).

# Primer unije

---

- Izpiši vsa mesta, kjer bodisi živi delavec ali ima sedež kupec.

```
(SELECT [poštna številka]  
FROM delavci)
```

```
UNION
```

```
(SELECT [poštna številka] AS pošta2  
FROM kupci)
```

Kakšen bo naziv  
stolpca,  
ki ga poizvedba vrne?