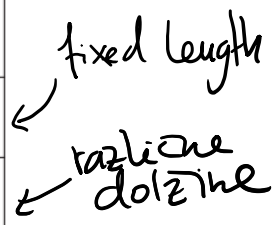
SQL oz. structured query language

(vir: GUPTA, Satinder Bal; MITTAL, Aditya. *Introduction to database management system*. Laxmi Publications, Ltd., 2009., Vir2: https://www.w3schools.com/sql)

Table 7.1. Basic data types

Datatype	Description	Size
Number(p, s)	It is used to store numeric data types. p stands for precision (total number of decimal digits) and s stands for scale (total number of digits after decimal point).	Range of p is from 1 to 38. And s is from -84 to 127.
Date	It is used to store date and time values.	Range of date is from Jan 1, 47 B.C. to Dec. 31, 9999 A.D.
Char(size)	It is used to store fixed size character data.	Range of char is 1 (By default) to 2000 bytes.
Varchar2(size)	It is used to store variable size character data.	Range of varchar2 is 1 (By default) to 4000 bytes.
Long	It is used to store variable size character data.	Range of long is upto 2 GB.
Clob	It is used to store variable size character data.	Range of clob is upto 4 GB
Raw(size)	It is used to store fixed binary data.	Maximum size is upto 2000 bytes.
Long raw	It is used to store variable binary data.	Maximum size is upto 2 GB.



SQL ima en osnovni stavek za pridobivanje informacij iz baze podatkov, in to je:

SELECT izjava.

Sql ima še tri druge DML stavke (z DML mislim data manipulation language), in to so:

update, delete in insert.

Osnovna oblika Stavka SELECT je sestavljena iz treh stavkov, in sicer SELECT, FROM in WHERE v obliki:

```
SELECT < attribute list >
  FROM < table list >
WHERE < condition >
```

SELECT:

Stavek Select se uporablja za pridobivanje informacij iz tabele.

Syntax :

```
SELECT < column list >
  FROM   < table name >.
```

Emp (Employee)

EID	Name	Salary	Hire-date	Job	DID	MID
701	Deepak	8000	5-Jan-2001	Analyst	30	707
702	Naresh	9000	10-Jan-2001	Manager	10	707
703	Sumesh	7000	5-Feb-2001	Salesman	20	705
704	Aditya	9000	27-Nov-2003	Analyst	30	707
705	Lalit	6500	8-Oct-2002	Manager	20	707
706	Amit		4-Nov-2004	Clerk	10	702
707	Vishal	9500	1-Jan-2001	Manager	30	
708	Sumit	8000	6-Jan-2006	Accountant	10	702

FIGURE 7.2. Employee table.

Dept (Department)

DID	DName	Loc	MID
10	Accounts	Bangalore	702
20	Sales	Delhi	705
30	Research	Pune	707
40	Developing	Hyderabad	

FIGURE 7.3. Department table.

stolpci

Za prikaz department ID in Department name imamo poizvedbo:

SELECT DID, DName

tabela FROM Dept ;

DID	DName
10	Accounts
20	Sales
30	Research
40	Developing

Za izbiro vseh stolpcev uporabimo '*'

```
SELECT *
FROM Dept;
```

DID	DName	Loc	Manager-ID
10	Accounts	Bangalore	702
20	Sales	Delhi	705
30	Research	Pune	707
40	Developing	Hyderabad	

Stolpce po svoji izbiri lahko poimenujete z uporabo ključne besede »AS«

(poda ime stolpca z velikimi črkami) ali z uporabo " " (poda ime stolpca, kot je navedeno v poizvedbi).

```
SELECT DID As Department_ID, DName
FROM Dept ;
```

Department-ID	DName
10	Accounts
20	Sales
30	Research
40	Developing

SELECT DISTINCT:

Stavek SELECT DISTINCT se uporablja za vrnitev samo različnih (različnih) vrednosti.

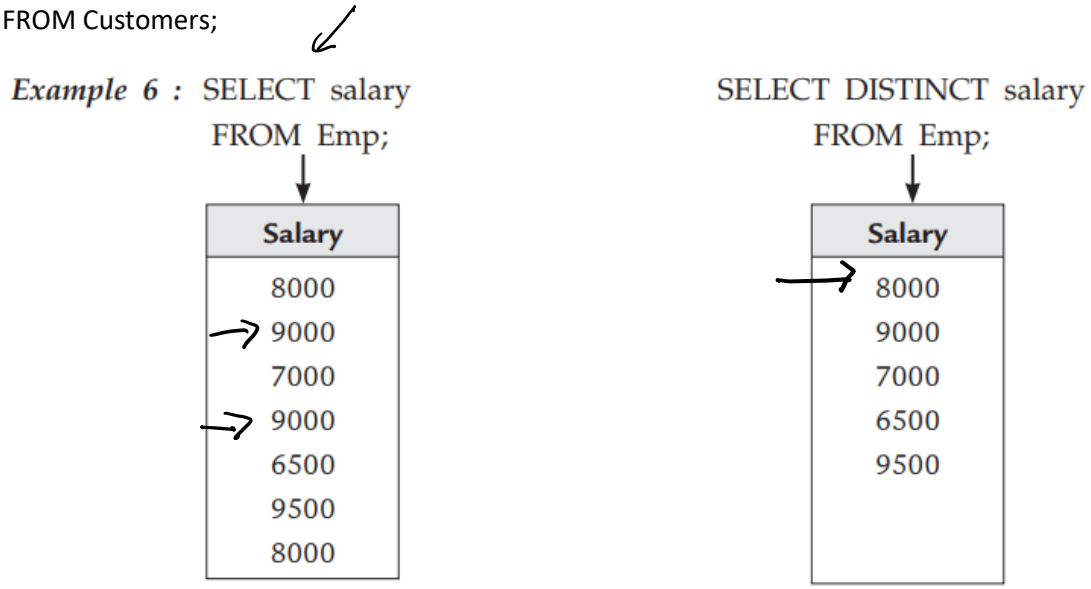
Znotraj tabele stolpec pogosto vsebuje veliko podvojenih vrednosti; in včasih želite samo navesti različne (različne) vrednosti.

Sintaksa:

```
SELECT DISTINCT column1, column2, ...
FROM table_name;
```

Primer:

SELECT DISTINCT Country
FROM Customers;



ARITMETIČNE OPERACIJE:

SQL ponuja aritmetične operatorje za izvajanje

izračuni. Aritmetični operatorji :

Description	Operator
Multiply	*
Divide	/
Add	+
Subtract	-

Example 7 : Suppose you want to increase salary of each employee by 500.

```
SELECT EID, salary + 500 "New Salary"
FROM Emp;
```

EID	New salary
701	8500
702	9500
703	7500
704	9500
705	7000
706	
707	10000
708	8500

WHERE:

WHERE se uporablja za izbiro določenih vrstic iz tabele.

Syntax :

```
SELECT   <column list>
FROM     <table name>
WHERE    <condition>.
```

Example 8 : List the name of employees having salary = 9000.

```
SELECT   name
FROM     emp
WHERE    salary = 9000;
```

atribut

Name
Naresh
Aditya

a) Primerjalni ali relacijski operatorji :

Operator	Description
=	Equal to
>	Greater than
<	Less than
>=	Greater than equal to
=<	Less than equal to
<>	Not equal to

<> , !=

Example 9 : List the name of employees having salary not equal to ₹ 9000.

```
SELECT name
FROM emp
WHERE salary <> 9000;
```

emp

Name
Deepak
Sumesh
Lalit
Amit
Vishal
Sumit

b) Posebni operatorji

Operator	Description
IN	Checking the value within a set
BETWEEN	Checking the value within a range
LIKE	Matching the pattern of characters
IS NULL	Checking the null value

IN:

Operator IN vam omogoča, da podate več vrednosti v členu WHERE.

Operator IN je okrajšava za več pogojev ALI.

Example 10 : List name and EID of employees having salary ₹ 8000 or ₹ 9500.

```
SELECT EID, name
  FROM Emp
WHERE salary IN (8000, 9500);
```

EID	Name
701	Deepak
707	Vishal
708	Sumit

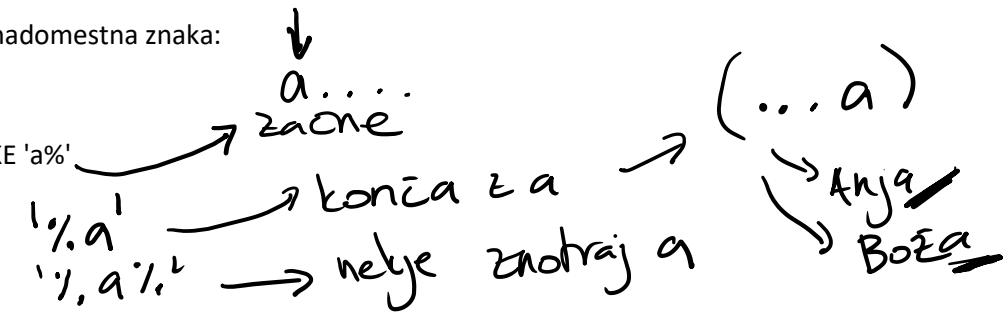
LIKE:

Operator LIKE se uporablja pri WHERE za iskanje določenega vzorca v stolpcu.

V povezavi z operatorjem LIKE se pogosto uporabljata dva nadomestna znaka:

% →predstavlja nič, enega ali več znakov

_ → predstavlja en sam znak primer: WHERE name LIKE 'a%'



BETWEEN, NOT BETWEEN:

Between izbere vrednosti med dvema vrednostima, NOT BETWEEN izbere vrednosti, ki niso znotraj postavljenih vrednosti.

NULL:

Polje z vrednostjo NULL je polje brez vrednosti.

Če polje v tabeli ni obvezno, je mogoče vstaviti nov zapis ali posodobiti zapis, ne da bi temu polju dodali vrednost. Nato bo polje shranjeno z vrednostjo NULL.

Vrednosti NULL ne moremo preveriti s primerjevalnimi operatorji kot so: <, <>, =; Uporabimo IS NULL ali IS NOT NULL

ORDER BY:

Sortira vrstice v naraščujoče ali padajoče rezultate. Če ni specificirano je rezultat razvrščen naraščujoče (uporabimo DESC ali ASC)

Syntax :

```
SELECT <column list>
FROM <table name>
WHERE <condition>
ORDER BY <column list> <ASC/DESC>;
```

naraščujoče

Name

padajoče

Example 18 : List name of employees in ascending order.

```
SELECT name
FROM Emp
ORDER BY name DESC
```

Name
Aditya
Amit
Deepak
Lalit
Naresh
Sumesh
Sumit
Vishal

JOIN:

Združitev tabel glede na povezanega stolpca med njimi.

The following is the common syntax for all types of JOIN.

```
SELECT table 1.columns, table 2.columns
  FROM          table 1, table 2.
WHERE           table 1.column N = table 2.column M;
```

- Kartezični produkt:

Za kartezični produkt se ne piše nobenega pogoja, vrnejo se vse možne kombinacije vrstic.

- Equijoin → Ko sta dve ali več tabel združeni z enakostjo vrednosti v eni ali več stolpcev, potem se imenuje Equijoin

Example 35 : Display EID and DName of all employees by joining over DID.

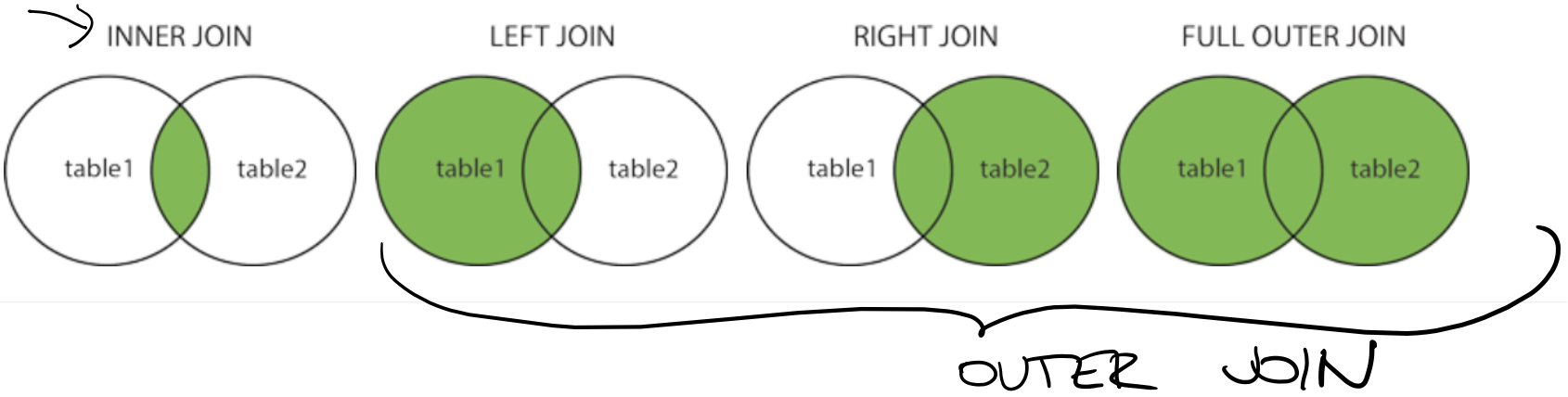
```
SELECT Emp.EID, Dept.DName
FROM Emp, Dept
WHERE Emp.DID = Dept.DID
```

EID	DName
701	Research
702	Accounts
703	Sales
704	Research
705	Sales
706	Accounts
707	Research
708	Accounts

Different Types of SQL JOINs

Here are the different types of the JOINs in SQL:

- `(INNER) JOIN`: Returns records that have matching values in both tables
- `LEFT (OUTER) JOIN`: Returns all records from the left table, and the matched records from the right table
- `RIGHT (OUTER) JOIN`: Returns all records from the right table, and the matched records from the left table
- `FULL (OUTER) JOIN`: Returns all records when there is a match in either left or right table



INNER JOIN:

Pokaže le zapise , ki se ujemajo v vrednostih v obeh tabelah.

Sintaksa:

```
SELECT * FROM table1
  JOIN table2
  ON relation;
```

LEFT JOIN:

Pokazalo bo vse zapise iz leve strani, na desni strani, pa bo pisalo ali NULL (v primeru, da se ne ujema) ali pa vrednost, ki se ujema.

FULL OUTER JOIN:

FULL OUTER JOIN si lahko predstavljate kot kombinacijo levega in desnega spoja. Obdržal bo vse vrstice iz obeh tabel, manjkajoči podatki pa bodo izpolnjeni z NULL.

SELF JOIN:

Example 39 : Display the name of employees and name of their managers.

```
SELECT e.Name "Employee", m.Name "Manager"
  FROM Emp e, Emp m
WHERE e.MID = m.MID;
```

Employee	Manager
Deepak	Vishal
Naresh	Vishal
Sumesh	Lalit
Aditya	Vishal
Lalit	Vishal
Amit	Naresh
Sumit	Naresh

UNION:

Operator UNION se uporablja za združevanje nabora rezultatov dveh ali več stavkov SELECT.

PRAVILA:

- Vsak stavek SELECT znotraj UNION mora imeti enako število stolpcev
- Stolpci morajo imeti podobne vrste podatkov.
- Stolpci v vsakem stavku SELECT morajo biti v istem vrstnem redu

(razlika z joinom -> union se ne združi po skupnem atributu, ampak najprej navede ene pol pa druge)

GROUP FUNKCIJE:

GROUP BY:

GROUP BY se uporablja za razdelitev tabele v skupine. Skupina lahko vsebuje celotno tabelo ali zbirko vrstic.
GROUP BY združi vrstice z enakimi vrednostmi v vrstice s povzetkom.
GROUP BY se pogosto uporablja z združevalnimi funkcijami (COUNT(), MAX(), MIN(), SUM(), AVG()) za združevanje nabora rezultatov po enem ali več stolpcih.

Syntax :

```
SELECT <column name>
  FROM <table name>
 WHERE <condition>
GROUP BY <column name>;
```

Example 44 : Display job and average salary paid by company for a particular job.

```
SELECT Job, AVG(salary)
  FROM emp
GROUP BY Job;
```

Job	AVG(Salary)
Accountant	8000
Analyst	8500
Manager	8333.33
Salesman	7000

HAVING:

HAVING je bil dodana v SQL, ker ključne besede WHERE ni mogoče uporabiti z GROUP BY

Syntax :

```
SELECT <column name>
  FROM <table name>
 WHERE <condition>
GROUP BY <column name>
HAVING <group condition>
```

← kako grupirati

KOMENTARJI:

```
SELECT * FROM Customers -- WHERE City='Berlin'; v isti vrstici

/* */  - čez več vrstic
```

Naloge:

(vir: https://techbeamers.com/sql-query-questions-answers-for-practice/)

(vir: https://sqlbolt.com/lesson/select_queries_introduction)

(vir: https://www.w3schools.com/sql/exercise.asp?filename=exercise_groupby2)

Editor : https://www.w3schools.com/sql/trysql.asp?filename=trysql_editor