

# levads datu bāzēs



LATVIJAS UNIVERSITĀTE  
**BIZNESĀ, VADĪBĀS  
UN EKONOMIKĀS  
FAKULTĀTE**

**VUMC** VADĪBAS UN  
UZNĒMĒJDARBĪBAS  
MĀCĪBU CENTRS

ESF projekts Nr. 8.4.1.0/16/I/001  
"Nodarbināto personu profesionālās kompetences pilnveide"

NACIONĀLAIS  
ATTĪSTĪBAS  
PLĀNS 2020



**EIROPAS SAVIENĪBA**  
Eiropas Sociālais  
fonds

I EGULDĪJUMS TAVĀ NĀKOTNĒ



# Tabulu datu tipi

3. lekcija



# Šodienas lekcijā

- **Kas ir datu tips**
- **Skaitliskie datu tipi**
- **Precizitāte**
- **Peldošā punkta tipi un problēmas**
- **Fiksētais tips**
- **Datuma un laika tipi**
- **Teksta tips**

# Par datu tipiem

- ✓ Datu tips ir piemēram integer, floating-point, Boolean utt.
- ✓ Datu tips apzīmē iespējamās vērtības, operācijas kuras var veikt ar šo tipu un veidu kā ši tipa vērtības tiek glabātas.

**Galvenās  
tipu  
kategorijas**

Numeric

DATE un  
TIME

String

# Kas ir datu tips

MySQL uztur visus SQL standarta skaitliskos datu tipus:

- ✓ INTEGER
- ✓ SMALLINT
- ✓ DECIMAL
- ✓ NUMERIC

MySQL uztur arī peldošā punkta skaitliskos tipus:

- ✓ FLOAT
- ✓ REAL
- ✓ DOUBLE PRECISION



# Skaitliskie datu tipi

MySQL uztur visus SQL standarta skaitliskos datu tipus :

## 💡 SQL standarta integer tipi

- INTEGER (vai INT)
- SMALLINT

## 💡 Uztur arī integer tipus

- TINYINT
- MEDIUMINT
- BIGINT

Group	Types
Integer Types	INTEGER, INT, SMALLINT, TINYINT, MEDIUMINT, BIGINT
Fixed Point Types	DECIMAL, NUMERIC
Floating Point Types	FLOAT, DOUBLE
Bit Value Type	BIT

# Skaitliskie datu tipi

Type	Length in bytes	Minimum Value (Signed)	Max Val (Signed)	Min Val (Unsigned)	Max Val (Unsigned)
TINYINT	1	-128	127	0	255
SMALLINT	2	-32768	32767	0	65535
MEDIUMINT	3	-8388608	8388607	0	16777215
INT	4	-2147483648	2147483648	0	429497295
BIGINT	8	-9223372036854775808	9223372036854775808	0	18446744073709551615

# Skaitlisko tipu precizitāte

```
mysql> CREATE TABLE Ages(Id SMALLINT, Age TINYINT) ENGINE=Memory;  
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> INSERT INTO Ages VALUES(1, 43);  
Query OK, 1 row affected (0.05 sec)
```

```
mysql> INSERT INTO Ages VALUES (2, 128);  
ERROR 1264 (22003): Out of range value for column 'Age' at row 1  
mysql> ALTER TABLE Ages MODIFY Age TINYINT UNSIGNED;  
Query OK, 1 row affected (0.06 sec)  
Records: 1 Duplicates: 0 Warnings: 0
```

```
mysql> INSERT INTO Ages VALUES(2, 128);  
Query OK, 1 row affected (0.05 sec)
```

```
mysql> SELECT * FROM Ages;  
+----+---+  
| Id | Age |  
+----+---+  
| 1 | 43 |  
| 2 | 128 |  
+----+---+  
2 rows in set (0.05 sec)
```

# Peldošā punkta tipi

- ✓ FLOAT un DOUBLE tipi attēlo aproksimētas skaitliskās datu vērtības
- ✓ MySQL atļauj nestandarta sintaksi
  - FLOAT(M,D)  Apzīmē vērtības līdz M cipariem,  
D decimālo zīmju skaitu
  - REAL(M,D)

FLOAT and DOUBLE represent approximate data types.

Type	Storage	Precision	Range
FLOAT	4 bytes	23 significant bits / ~7 decimal digits	$10^{+/-38}$
DOUBLE	8 bytes	53 significant bits / ~16 decimal digits	$10^{+/-308}$

REAL is a synonym for FLOAT. DOUBLE PRECISION is a synonym for DOUBLE.

# Peldošā punkta problēmas

```
mysql> CREATE TABLE Numbers (Id TINYINT, Floats FLOAT, Decimals DECIMAL(3, 2));  
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> INSERT INTO Numbers VALUES (1, 1.1, 1.1), (2, 1.1, 1.1), (3, 1.1, 1.1);  
Query OK, 3 rows affected (0.05 sec)  
Records: 3 Duplicates: 0 Warnings: 0
```

```
mysql> SELECT * FROM Numbers;  
+----+----+-----+  
| Id | Floats | Decimals |  
+----+----+-----+  
| 1 | 1.1 | 1.10 |  
| 2 | 1.1 | 1.10 |  
| 3 | 1.1 | 1.10 |  
+----+----+-----+  
3 rows in set (0.05 sec)
```

```
mysql> SELECT SUM(Floats), SUM(Decimals) FROM Numbers;  
+-----+-----+  
| SUM(Floats) | SUM(Decimals) |  
+-----+-----+  
| 3.300000715255737 | 3.30 |  
+-----+-----+  
1 row in set (0.05 sec)
```

# Fiksētais tips

- ✓ DECIMAL(m,n) m - zīmju skaits, n - decimālo ciparu skaits
  - DECIMAL(5,2) var saglabāt vērtību līdz 5 cipariem un 2 decimālajām zīmēm
- ✓ Vērtību intervāls -999.99 to 999.99
- ✓ DECIMAL(M) ir sinonīms DECIMAL(M,0)
- ✓ DECIMAL ir sinonīms to DECIMAL(M,0)



# Piemērs (decimal)

```
mysql> CREATE TABLE materials (
->     id INT AUTO_INCREMENT PRIMARY KEY,
->     description VARCHAR(255),
->     cost DECIMAL(19 , 4 ) NOT NULL
-> );
```

```
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> INSERT INTO materials(description,cost)
-> VALUES('Bicycle', 500.34),('Seat',10.23),('Break',5.21);
```

```
Query OK, 3 rows affected (0.05 sec)
```

```
Records: 3 Duplicates: 0 Warnings: 0
```

```
mysql> select * from materials;
+----+-----+-----+
| id | description | cost   |
+----+-----+-----+
| 1  | Bicycle    | 500.3400 |
| 2  | Seat        | 10.2300  |
| 3  | Break       | 5.2100  |
+----+-----+-----+
```

```
3 rows in set (0.05 sec)
```

# Date un Time tips

Tipi DATE, TIME,  
DATETIME,  
TIMESTAMP  
and YEAR

Katram tipam ir  
vērtību intervāls  
kā arī nulles  
vērtība

Tips	Apraksts	Displeja Formāts	Intervāls
DATETIME	satur abus datums un laiks .	YYYY-MM-DD HH:MM:SS	'1000-01-01 00:00:00' līdz '9999-12-31 23:59:59'.
DATE	Tikai datums.	YYYY-MM-DD	'1000-01-01' līdz '9999- 12-31'.
TIMESTAMP	pārveido tekošo laika zonu uz UTC saglabājot, un konvertē atpakaļ no UTC uz tekošo laika zonu kad nolasa	YYYY-MM-DD HH:MM:SS	'1970-01-01 00:00:01' UTC līdz '2038-01-19 03:14:07' UTC

# Date tips

```
mysql> SELECT CURDATE();  
+-----+  
| CURDATE() |  
+-----+  
| 2021-08-15 |  
+-----+  
1 row in set (0.05 sec)
```

```
mysql> SELECT DATE('2017-01-31 12:01:00');  
+-----+  
| DATE('2017-01-31 12:01:00') |  
+-----+  
| 2017-01-31 |  
+-----+  
1 row in set (0.05 sec)
```

```
mysql> SELECT ADDDATE('2017-01-20', 8);  
+-----+  
| ADDDATE('2017-01-20', 8) |  
+-----+  
| 2017-01-28 |  
+-----+  
1 row in set (0.05 sec)
```



# Date formāti

```
mysql> CREATE TABLE Dates(Id TINYINT, Dates DATE);
```

Query OK, 0 rows affected (0.07 sec)

```
mysql> INSERT INTO Dates VALUES(1, '2017-01-24');
```

Query OK, 1 row affected (0.06 sec)

```
mysql> INSERT INTO Dates VALUES(2, '2017/01/25');
```

Query OK, 1 row affected (0.05 sec)

```
mysql> INSERT INTO Dates VALUES(3, '20170126');
```

Query OK, 1 row affected (0.05 sec)

```
mysql> INSERT INTO Dates VALUES(4, '170127');
```

Query OK, 1 row affected (0.05 sec)

```
mysql> INSERT INTO Dates VALUES(5, '2017+01+28');
```

Query OK, 1 row affected (0.05 sec)

```
mysql> SELECT * FROM Dates;
```

```
+-----+-----+
```

Id	Dates
1	2017-01-24
2	2017-01-25
3	2017-01-26
4	2017-01-27
5	2017-01-28

```
+-----+-----+
```

```
+-----+-----+
```

5 rows in set (0.05 sec)

# Time

```
mysql> SELECT CURTIME();
+-----+
| CURTIME() |
+-----+
| 10:42:23 |
+-----+
1 row in set (0.04 sec)

mysql> SELECT TIMEDIFF('23:34:32', '22:00:00');
+-----+
| TIMEDIFF('23:34:32', '22:00:00') |
+-----+
| 01:34:32 |
+-----+
1 row in set (0.04 sec)

mysql> SELECT TIME('2017-01-31 11:06:43');
+-----+
| TIME('2017-01-31 11:06:43') |
+-----+
| 11:06:43 |
+-----+
1 row in set (0.05 sec)

mysql> SELECT TIMEDIFF('211344', 201123);
+-----+
| TIMEDIFF('211344', 201123) |
+-----+
| 01:02:21 |
+-----+
1 row in set (0.05 sec)
```



# Datetime

```
mysql> SELECT NOW();  
+-----+  
| NOW() |  
+-----+  
| 2021-08-15 11:17:04 |  
+-----+  
1 row in set (0.05 sec)
```

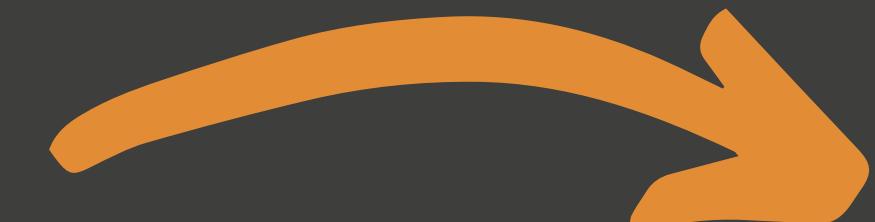
```
mysql> SELECT DAYNAME('2017@01@31 11@12@12');  
+-----+  
| DAYNAME('2017@01@31 11@12@12') |  
+-----+  
| Tuesday |  
+-----+  
1 row in set (0.05 sec)
```

Add a little bit of body text

# Year

```
mysql> SELECT YEAR(CURDATE()) AS 'Current year';  
+-----+  
| Current year |  
+-----+  
|      2021 |  
+-----+  
1 row in set (0.05 sec)
```

# Timestamp

- Timestamp ir tips ar kuru var saglabāt kāda notikuma datumu/laiku
  - Timestamp tipiski izmanto notikumu reģistrēšanai (logging)
  - TIMESTAMP kolonu izmanto INSERT vai UPDATE operācijas datuma/laika reģistrēšanai
- 

Data type	Format
TIMESTAMP(14)	YYYYMMDDHHMMSS
TIMESTAMP(12)	YYMMDDHHMMSS
TIMESTAMP(10)	YYMMDDHHMM
TIMESTAMP(8)	YYYYMMDD
TIMESTAMP(6)	YYMMDD
TIMESTAMP(4)	YYMM
TIMESTAMP(2)	YY

# Timestamp

```
mysql> CREATE TABLE Prices(Id TINYINT PRIMARY KEY,  
Price DECIMAL(8, 2), Stamp TIMESTAMP);  
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> INSERT INTO Prices(Id, Price) VALUES(1, 234.34);  
Query OK, 1 row affected (0.05 sec)
```

```
mysql> INSERT INTO Prices(Id, Price) VALUES(2, 344.12);  
Query OK, 1 row affected (0.06 sec)
```

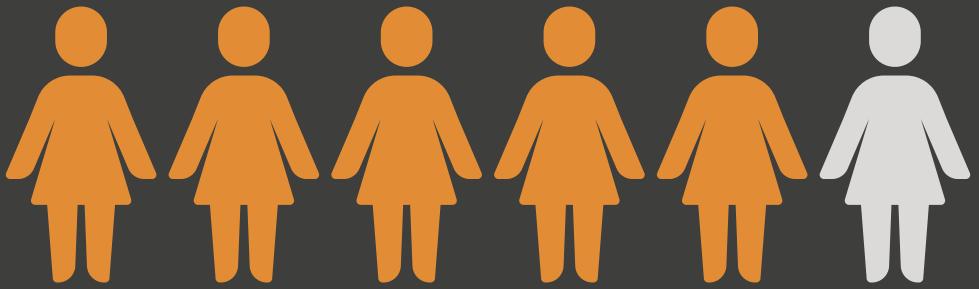
```
mysql> SELECT * FROM Prices;  
+----+----+-----+  
| Id | Price | Stamp      |  
+----+----+-----+  
| 1  | 234.34 | 2021-08-15 11:25:19 |  
| 2  | 344.12 | 2021-08-15 11:25:28 |  
+----+----+-----+  
2 rows in set (0.04 sec)
```

```
mysql> UPDATE Prices SET Price=250.50 WHERE Id=1;  
Query OK, 1 row affected (0.05 sec)  
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> SELECT * FROM Prices;  
+----+----+-----+  
| Id | Price | Stamp      |  
+----+----+-----+  
| 1  | 250.50 | 2021-08-15 11:25:49 |  
| 2  | 344.12 | 2021-08-15 11:25:28 |  
+----+----+-----+  
2 rows in set (0.05 sec)
```



# Teksta tips

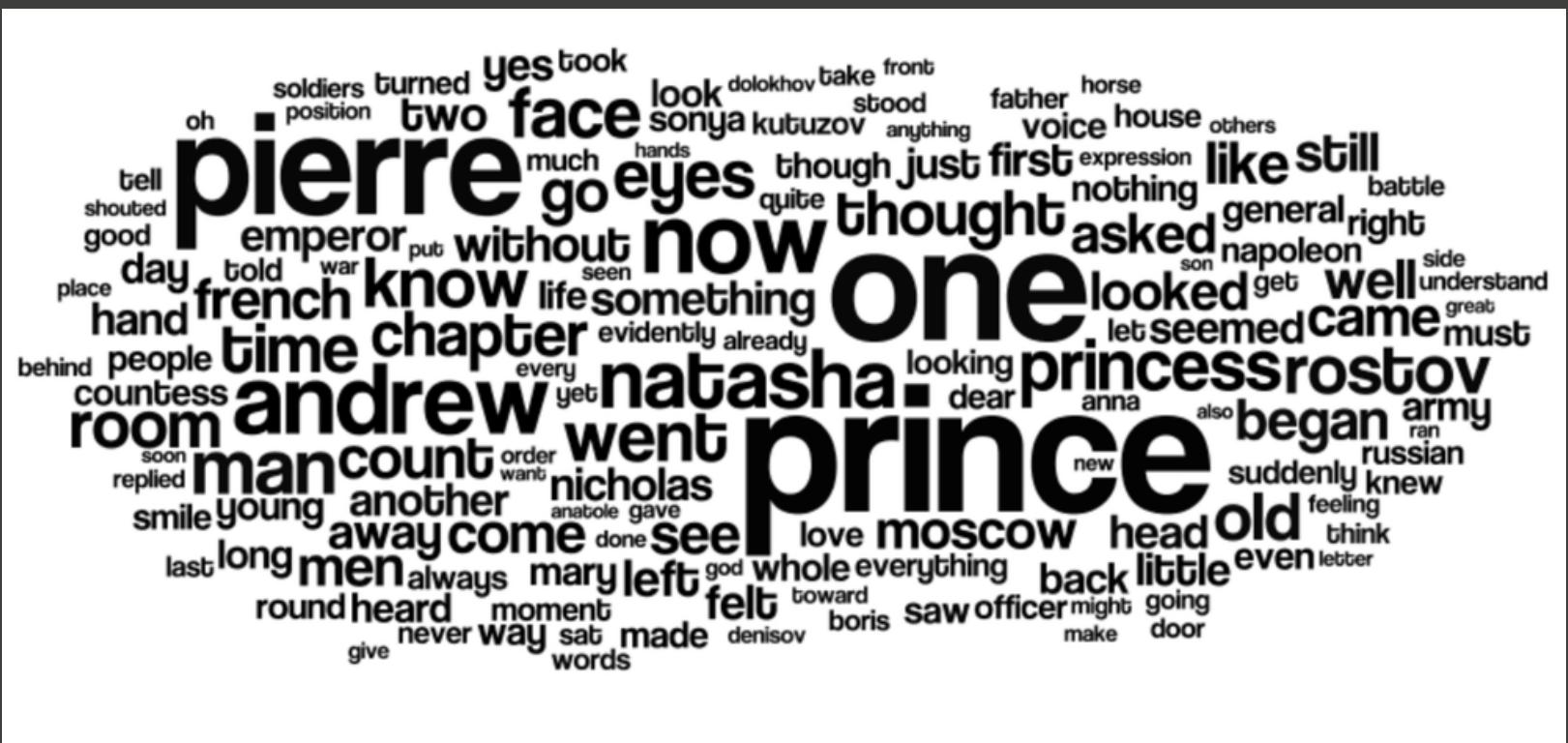


CHAR

VARCHAR

BINARY

VARBINARY



BLOB

TEXT

ENUM

SET



# Char

- ★ CHAR tipu datubāzē glabā ar tukšumiem līdz kolonas garumam
- ★ Datus nolasot "trailing spaces" jeb labās puses tukšumus izmet

```
mysql> CREATE TABLE Chars(Id TINYINT PRIMARY KEY, Chars CHAR(3));  
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> INSERT INTO Chars VALUES (1, 'a'), (2, 'ab'), (3, 'abc');  
Query OK, 3 rows affected (0.05 sec)
```

```
mysql> INSERT INTO Chars VALUES (1, 'abcd');  
ERROR 1406 (22001): Data too long for column 'Chars' at row 1
```

# Varchar



levads datu bāzēs  
2022

VARCHAR datu tips glabā  
mainīga garuma teksta  
rindu

Rindas garums var būt 0  
līdz 65535

VARCHAR netiek  
uzpildītas no labās pusēs

```
mysql> CREATE TABLE FirstNames(Id TINYINT, Firstname VARCHAR(20));
Query OK, 0 rows affected (0.08 sec)

mysql> INSERT INTO FirstNames VALUES (1, 'Tom'), (2, 'Lucy'), (3, 'Alice'),
-> (4, 'Robert'), (5, 'Timothy'), (6, 'Alexander');
Query OK, 6 rows affected (0.05 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> SELECT Id, Firstname, LENGTH(FirstName) AS Length FROM FirstNames;
+----+-----+-----+
| Id | Firstname | Length |
+----+-----+-----+
| 1 | Tom       | 3    |
| 2 | Lucy      | 4    |
| 3 | Alice     | 5    |
| 4 | Robert    | 6    |
| 5 | Timothy   | 7    |
| 6 | Alexander | 9    |
+----+-----+-----+
6 rows in set (0.09 sec)
```

# BLOB

- BLOB ir Binary Large Objekta datu tips
- Var saturēt mainīga garuma binārus datus
- Izmanto lai glabātu tādus datus kā bildes vai dokumenti

```
mysql> CREATE TABLE Images(Id INT PRIMARY KEY, Img LONGBLOB);
Query OK, 0 rows affected (0.06 sec)
mysql> SHOW VARIABLES LIKE "secure_file_priv";
+-----+-----+
| Variable_name | Value          |
+-----+-----+
| secure_file_priv | /secure_file_priv_dir/ |
+-----+-----+
1 row in set (0.04 sec)
```

```
mysql> INSERT INTO Images VALUES (1, LOAD_FILE('/Users/robertspolis/Pictures/1.png'));
Query OK, 1 row affected (0.05 sec)
```

<b>Blog type</b>	<b>Range in bytes</b>
TINYBLOB	0 - 255
BLOB	0 - 65535
MEDIUMBLOB	0 - 16777215
LONGBLOB	0 - 4294967295

# JSON Data Types

- JSON tipu izmanto JSON formāta datu glabāšanai

```
CREATE TABLE t1 (jdoc JSON);
INSERT INTO t1 VALUES('{"key1": "value1", "key2": "value2"}');
```

- Konstruēt JSON no elementiem

```
SELECT JSON_ARRAY('a', 1, NOW());
```

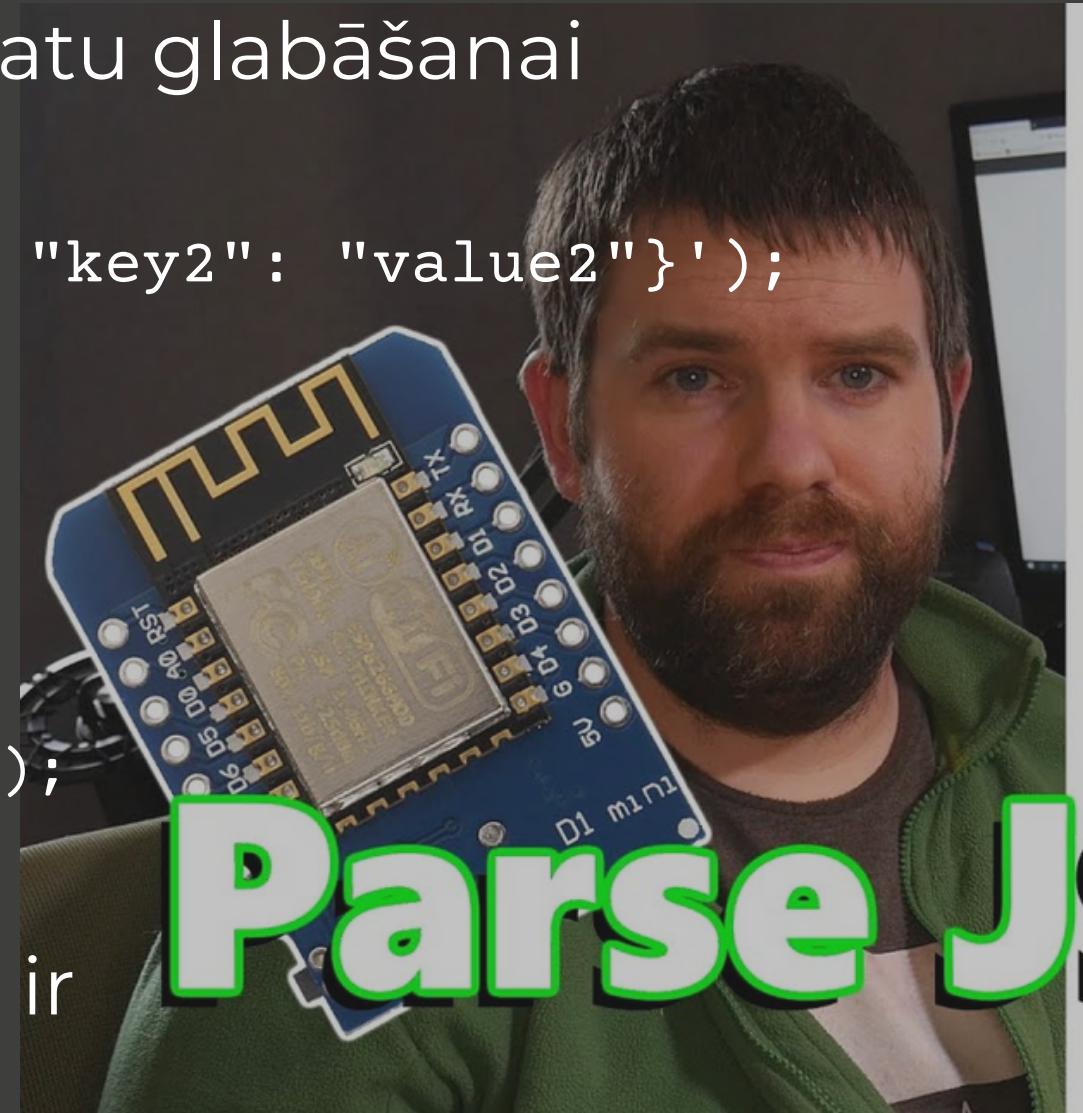
```
SELECT JSON_OBJECT('key1', 1, 'key2', 'abc');
```

- Pārbaudīt kāda tipa dokuments tas ir

```
SELECT JSON_TYPE('["a", "b", 1]');
```

- Izveidot JSON objektu no ierakstiem

```
SELECT json_object('name', name, 'phone', phone)
FROM person;
```



```
{
  "ethereum": {
    "usd": 3961.66,
    "eur": 3261.73
  },
  "bitcoin": {
    "usd": 48924,
    "eur": 40231
  }
}
```

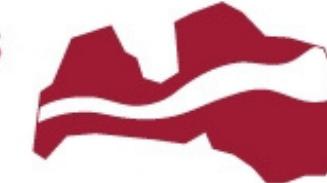


VADĪBAS UN  
UZNĒMĒJDARBĪBAS  
MĀCĪBU CENTRS

VUMC  
levads datu bāzēs  
2022

ESF projekts Nr. 8.4.1.0/16/I/001  
"Nodarbināto personu profesionālās kompetences pilnveide"

NACIONĀLAIS  
ATTĪSTĪBAS  
PLĀNS 2020



**EIROPAS SAVIENĪBA**  
Eiropas Sociālais  
fonds

I E G U L D Ī J U M S   T A V Ā   N Ą K O T N ē