



PostgreSQL

Analysing Open Data

Matej Kovačič

`matej.kovacic@ijs.si`

Jožef Stefan Institute

Centre for Knowledge Transfer
in Information Technologies

Artificial Intelligence Laboratory

SQL and PostgreSQL

SQL (*Structured Query Language*) – is a structured query language for working with databases.

PostgreSQL is opensource application for database management which could be compared with the most powerful commercial solutions.

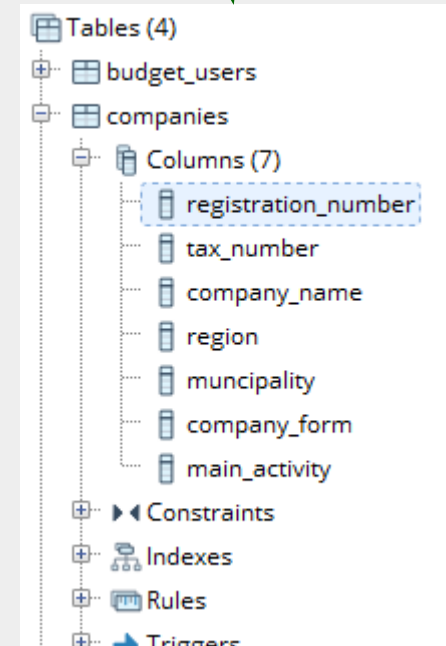
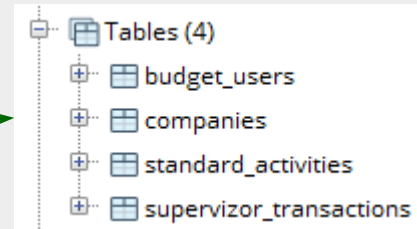
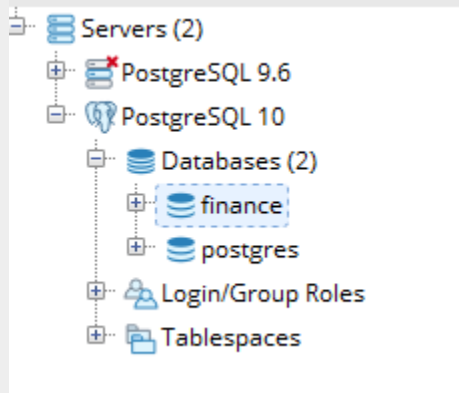
Features:

- high reliability,
- speed,
- ability of analysing big amount of data (several million queries),
- with addons it is possible to perform analysis of spacial data (PostGIS), pictures (PostPic), text indexing (OpenFTS), etc.
- it is free.

Data organisation



Database server → Database → Table → Column → Row.



	registration_number text	tax_number integer	company_name text	region text	municipality text	company_form text	main_activity text
1	1001779000	46272305	RESTAVRACIJA BLEJ...	GORENJ...	Bled	Samostojni podjet...	55.301
2	1006622000	69010757	ORGANIZIRANJE GL...	POMUR...	Radenci	Samostojni podjet...	90.040
3	1016024000	72457708	EVA - CVETJE ZA D...	GORIŠKA	Nova Gorica	Samostojni podjet...	47.761
4	1028685000	83379908	ERKER LOVRENC JA...	OSRED...	Logatec	Samostojni podjet...	74.203
5	1028774000	27282562	MIZARSTVO MOLK ...	OSRED...	Logatec	Samostojni podjet...	16.230
6	1028871000	97507741	KNJIGOVODSKI SER...	OSRED...	Logatec	Samostojni podjet...	69.200
7	1033280000	48212105	GUSTINČIČ ZDENK...	GORIŠKA	Nova Gorica	Notar	69.102
8	1033514000	90215605	ROJEC ANTON - NO...	SAVINJS...	Celje	Notar	69.102
9	1033549000	66945267	ŠOEMEN ANDREJ - ...	PODRA...	Ptuj	Notar	69.102
10	1033794000	85108448	MALI-LEMUT VIDA ...	GORIŠKA	Ajdovščina	Evidentirani odvet...	69.101
11	1033824000	70068607	BENKOČ IGOR - OD...	SAVINJS...	Celje	Evidentirani odvet...	69.101
12	1033883000	78257441	FEGUŠ MARJAN - O...	SAVINJS...	Celje	Evidentirani odvet...	69.101
13	1033891000	35889047	FINK ROK - ODVET...	SAVINJS...	Celje	Evidentirani odvet...	69.101
14	1033930000	59739363	INKRET IGOR - ODV...	SAVINJS...	Celje	Evidentirani odvet...	69.101
15	1033964000	77597176	JAZBINŠEK-GORIČA...	SAVINJS...	Celje	Evidentirani odvet...	69.101
16	1033999000	47728809	KARLOVŠEK MARIJA...	SAVINJS...	Celje	Evidentirani odvet...	69.101
17	1034006000	59471808	KLEP MATJAŽ - OD...	SAVINJS...	Celje	Evidentirani odvet...	69.101

Data organisation

Database → table → column → row.

```
finance=# \d
```

```
List of relations
```

Schema	Name	Type	Owner
public	supervizor_transactions	table	matej
public	budget_users	table	matej
public	companies	table	matej
public	standard_activities	table	matej

```
finance=# \d companies
```

```
Table "public.compnies"
```

Column	Type	Modifiers
registration_number	text	
tax_number	integer	
company_name	text	
region	text	
municipality	text	
company_form	text	
main_activity	text	

Main data types

The most used data types are:

- text data type (*text, char...*)
- numeric data type (*decimal, numeric, integer,...*)
- time data type (*timestamp* (with or without time zone), *date, time...*),
- other (*geometric, boolean,...*).

PostgreSQL installation

PostgreSQL is available for many operating systems (Linux, Windows, Mac OS, BSD, Solaris).

<<https://www.postgresql.org/download>>

Installation process is specific for target operating system.

After installation you need to set access rights to databases, create database users, set firewall, etc. ... especially if database server will be accessible from the internet:

- create new databases users,
- set strong passwords,
- limit user rights,
- set firewall and encrypted connections to database server,...

PgAdmin

PgAdmin is a graphical tool for working with PostgreSQL. It allows users to connect to remote database servers too.

The screenshot displays the PgAdmin 4 interface. On the left is a 'Browser' pane showing a tree view of the database structure, including servers, databases, and various objects like tables and views. The main area is a dashboard with several charts and a table:

- Database sessions:** A chart showing the number of active database sessions.
- Transactions per second:** A chart showing the rate of transactions, with a legend for Commits (blue), Rollbacks (yellow), and Transactions (red).
- Tuples in:** A chart showing the rate of tuples being inserted, updated, or deleted, with a legend for Inserts (blue), Updates (yellow), and Deletes (red).
- Tuples out:** A chart showing the rate of tuples being fetched or returned, with a legend for Fetched (blue) and Returned (yellow).
- Block I/O:** A chart showing the rate of block reads and hits, with a legend for Reads (blue) and Hits (yellow).
- Database activity:** A table showing active database sessions.

Database activity										
Sessions										
		PID	User	Application	Client	Backend start	State	Wait Event	Blocking PIDs	
✖	■	▶	3080	postgres	pgAdmin 4 - CONN:5127806	:::1	2017-11-18 20:29:12 CET	idle	Client: ClientRead	
✖	■	▶	3312	postgres	pgAdmin 4 - DB:finance	:::1	2017-11-18 20:25:49 CET	idle	Client: ClientRead	

PgAdmin vs. command line

Instead of PgAdmin we can use command line.

finance on postgres@PostgreSQL 10

```
1 SELECT tax_number, region FROM companies LIMIT 10;
```

Data Output Explain Messages Query Hi

	tax_number integer	region text
1	46272305	GORENJSKA
2	69010757	POMURSKA
3	72457708	GORIŠKA
4	83379908	OSREDNJESLOVENSKA
5	27282562	OSREDNJESLOVENSKA
6	97507741	OSREDNJESLOVENSKA
7	48212105	GORIŠKA
8	90215605	SAVINJSKA
9	66945267	PODRAVSKA
10	85108448	GORIŠKA

matej@cryptoloop: ~/Namizje/VirtualShared/psql

```
registration_number | text |
tax_number          | integer |
company_name        | text |
region              | text |
municipality        | text |
company_form        | text |
main_activity       | text |
```

```
finance=> SELECT tax_number, region FROM companies LIMIT 10;
```

```
tax_number | region
-----+-----
46272305 | GORENJSKA
69010757 | POMURSKA
72457708 | GORIŠKA
83379908 | OSREDNJESLOVENSKA
27282562 | OSREDNJESLOVENSKA
97507741 | OSREDNJESLOVENSKA
48212105 | GORIŠKA
90215605 | SAVINJSKA
66945267 | PODRAVSKA
85108448 | GORIŠKA
```

(10 rows)

finance=> █

Preparing the data

Getting the data:

- public sources,
- Access To Public Information Act.

Understanding form of data:

- CSV, tab-delimited, MDB, XML, JSON...
- character encoding (i. e. selecting correct character table, like CP-1250, ISO-8859-2, UFT-8,...).

Preparing and cleaning of data:

- manually,
- Excel/LibreOffice,
- Google Refine, TextEdit/Gedit, bash...

```
grep -v '^$' data.txt >  
data_without_empty_lines.txt
```

Creating new database

The screenshot shows the pgAdmin 4 interface. In the left-hand 'Browser' pane, the tree view is expanded to 'PostgreSQL 10' > 'Databases'. A context menu is open over the 'Databases' folder, with 'Create' > 'Database...' selected. A red arrow points from this menu item to the 'Database...' option in the main dashboard area.

The main dashboard displays several performance metrics:

- Server sessions:** A line chart showing Active (blue), Idle (yellow), and Total (red) sessions over time. The y-axis ranges from 1.0 to 7.0.
- Transactions per second:** A line chart showing Commits (blue), Rollbacks (yellow), and Transactions (red) over time. The y-axis ranges from 0.0 to 8.0.
- Tuples in:** A line chart showing Inserts (blue), Updates (yellow), and Deletes (red) over time. The y-axis ranges from 0.00 to 1.00.
- Tuples out:** A line chart showing Fetched (blue) and Returned (yellow) tuples over time. The y-axis ranges from 0 to 1200.
- Block I/O:** A line chart showing Reads (blue) and Hits (yellow) over time. The y-axis ranges from 0 to 80.

At the bottom, the 'Server activity' section shows a table of active sessions:

		PID	Database	User	Application	Client	Backend start	State	Wait Event	Blocking PIDs
✖	■	▶ 216					2017-11-16 10:50:15 CET		Activity: WalWriterMain	
✖	■	▶ 376	postgres	postgres	pgAdmin 4 - DB:postgres	::1	2017-11-16 11:03:10 CET	active		
✖	■	▶ 2428		postgres			2017-11-16 10:50:15 CET		Activity: LogicalLauncherMain	
✖	■	▶ 2928					2017-11-16 10:50:15 CET		Activity: BgWriterHibernate	
✖	■	▶ 2984					2017-11-16 10:50:15 CET		Activity: CheckpointerMain	
✖	■	▶ 3788					2017-11-16 10:50:15 CET		Activity: AutoVacuumMain	

Databases – Create - Database.

Creating new database

The screenshot displays the pgAdmin 4 interface. A 'Create - Database' dialog box is open, showing the 'General' tab. The 'Database' field contains the text 'finance', and the 'Owner' dropdown is set to 'postgres'. A red arrow points to the 'Database' field. The background shows a dashboard with various charts and a table of server sessions.

State	Wait Event	Blocking PIDs
T	Activity: WalWriterMain	
T	active	
T	Activity: LogicalLauncherMain	
T	Activity: BgWriterHibernate	
T	Activity: CheckpointerMain	
T	Activity: AutoVacuumMain	

Set database name and owner. Database will contain tables.

Creating new database

The screenshot shows the pgAdmin 4 interface with a sidebar on the left displaying a tree view of servers and databases. The main area contains several performance charts: 'Database sessions' (Active, Idle, Total), 'Transactions per second' (Commits, Rollbacks, Transactions), 'Tuples in' (Inserts, Updates, Deletes), 'Tuples out' (Fetched, Returned), and 'Block I/O' (Reads, Hits). Below these charts is a 'Database activity' section with tabs for Sessions, Locks, and Prepared Transactions. A table displays active sessions, with one session highlighted. A context menu is open over the 'public' schema in the sidebar, with a red arrow pointing to the 'Query Tool...' option.

	PID	User	Application	Client	Backend start	State	Wait Event	Blocking PIDs
■ ▶	2484	postgres	pgAdmin 4 - DB:finance	:::1	2017-11-16 11:11:37 CET	idle	Client: ClientRead	

Databases - "my_database" - Schemas - public - Query Tool

Creating new user and new database

On Linux systems:

```
sudo su
su - postgres
psql
```

Creating new user (in database *postgres*):

```
CREATE USER student WITH PASSWORD
'myproject';
ALTER USER student VALID UNTIL 'infinity';
```

Creating new database:

```
CREATE DATABASE finance WITH OWNER
student;
```

Creating new table

Create new table and define its structure:

```
create table supervisor_transactions
(debit_sub_account text,
date_of_transaction date,
amount_of_transaction numeric(16,4),
currency_code text, recipient_bank_account
text, name_of_recipient text,
registration_number text, tax_number
integer, budget_user_code integer,
zr_sns_oe text, purpose text);
```

Why »*text*« data type?

Creating new table

The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of the database structure, with 'supervisor_transactions' selected under 'Tables (4)'. The main window shows the 'Columns' tab for the 'supervisor_transactions' table. The table has 11 columns, each with a checkbox for selection, a data type, length, precision, and 'Not NULL?' and 'Primary key?' options.

	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	debit_sub_account	text			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	date_of_transaction	date			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	amount_of_transaction	numeric	16	4	<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	currency_code	text			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	recipient_bank_account	text			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	name_of_recipient	text			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	registration_number	text			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	tax_number	integer			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	budget_user_code	integer			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	zr_sns_oe	text			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	purpose	text			<input type="checkbox"/> No	<input type="checkbox"/> No

Databases - "my_database" - Schemas - public - Tables - "my_table" - Properties - Columns

Importing data into table - "manually"

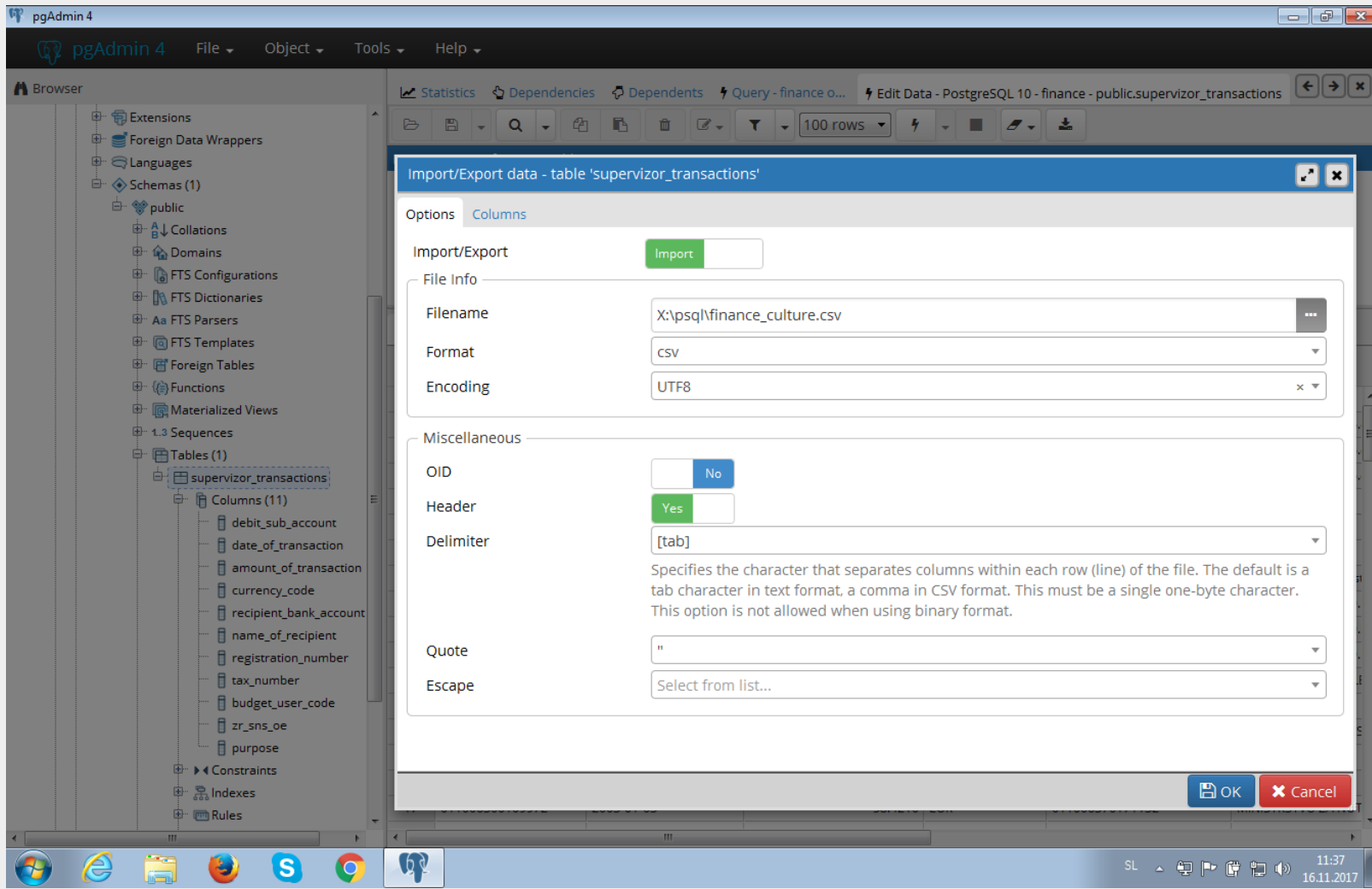
Rule:

```
INSERT INTO my_table (col1, col2) VALUES  
('value 1', 'value 2');
```

Our example:

```
INSERT INTO supervizor_transactions  
(debit_sub_account, date_of_transaction,  
amount_of_transaction, currency_code,  
recipient_bank_account, name_of_recipient,  
registration_number, tax_number,  
budget_user_code, zr_sns_oe, purpose)  
VALUES ('011006300109972', '2003-01-03',  
8419.9633, 'EUR', '010000001910013', 'DBS  
d.d.', '5349907000', 18787762, 16110,  
'161140-02-01220', '3054913 POROŠTVA RS  
GLAVNICA');
```


Importing data into table - from file



Databases - "*my_database*" - Import/Export.
Format, encoding, delimiter, does file has header?

Importing data into table - from file

Rule:

```
COPY my_table FROM 'filename.csv' with csv
header delimiter E'\t';
```

```
COPY my_table2 FROM 'filename.txt' with
csv header delimiter ';' encoding
'windows-1250';
```

Our example:

```
COPY supervisor_transactions from
'C:\psql\finance_culture.csv' with csv
header delimiter E'\t';
```

On Linux systems use \COPY instead of COPY.

Creating indexes

Indexes are used to increase performance of data operations. For indexing »special« data types, you need to add extensions. For text indexing and fast text searching you need to add *pg_trgm* extension:

```
CREATE EXTENSION pg_trgm;
```

Create index on numeric data (*tax_number*):

```
CREATE INDEX itax_number on  
supervizor_transactions(tax_number);
```

Create index on text data (*purpose*):

```
CREATE INDEX ipurpose on  
supervizor_transactions USING gist  
(purpose gist_trgm_ops);
```

Adding data

Adding additional columns:

```
ALTER TABLE supervisor_transactions ADD  
COLUMN tax_number_text text;
```

Adding values into additional columns:

```
UPDATE supervisor_transactions SET  
tax_number_text = tax_number::text;
```

Adding rows: with INSERT or COPY.

Deleting data

Deleting columns:

```
ALTER TABLE supervisor_transactions DROP  
COLUMN tax_number_text;
```

Deleting rows:

```
DELETE FROM supervisor_transactions  
WHERE [budget_user_code = 16110];
```

Deleting table:

```
DROP TABLE supervisor_transactions;
```

Let's import our sample data...

```
CREATE TABLE supervizor_transactions (debit_sub_account text,  
date_of_transaction date, amount_of_transaction numeric(16,4),  
currency_code text, recipient_bank_account text, name_of_recipient  
text, registration_number text, tax_number integer,  
budget_user_code integer, zr_sns_oe text, purpose text);  
COPY supervizor_transactions from 'C:\psql\finance_culture.csv'  
with csv header delimiter E'\t';
```

```
CREATE TABLE budget_users (budget_user_code integer, name text,  
category text);  
COPY budget_users from 'C:\psql\budget_users.csv' with csv header  
delimiter E'\t';
```

```
CREATE TABLE companies (registration_number text, tax_number  
integer, company_name text, region text, municipality text,  
company_form text, main_activity text);  
COPY companies from 'C:\psql\companies.csv' with csv header  
delimiter E'\t';
```

```
CREATE TABLE standard_activities (level text, category text,  
description_slo text, description_eng text, parent_level text);  
COPY standard_activities from  
'C:\psql\standard_classification_of_activities.csv' with csv header  
delimiter E'\t';
```

Basic overview of data

```
SELECT * FROM companies;  
SELECT company_name FROM companies;  
SELECT region FROM companies;  
SELECT company_name, region FROM companies;
```

-S

q

```
SELECT * FROM companies LIMIT 5;
```

```
SELECT * FROM supervisor_transactions ORDER  
BY amount_of_transaction;
```

```
SELECT purpose FROM supervisor_transactions  
ORDER BY amount_of_transaction DESC;
```

```
SELECT company_name, tax_number FROM  
companies ORDER BY 1 DESC;
```

Conditions

```
SELECT * FROM supervisor_transactions WHERE  
[amount_of_transaction > 100000];
```

```
SELECT * FROM supervisor_transactions WHERE  
[budget_user_code = 33405];
```

```
SELECT * FROM supervisor_transactions WHERE  
[budget_user_code = 33405] ORDER BY  
date_of_transaction DESC;
```

```
SELECT * FROM supervisor_transactions WHERE  
[amount_of_transaction > 100000] AND  
[budget_user_code = 16110];
```


Conditions

```
SELECT name_of_recipient FROM
supervizor_transactions WHERE
(registration_number = '5860571000') AND NOT
(budget_user_code = 16110);
```

```
SELECT name_of_recipient,
amount_of_transaction, currency_code FROM
supervizor_transactions WHERE
(registration_number = '5860571000') AND
(budget_user_code = 16110) ORDER BY 2 DESC
LIMIT 5;
```

Conditions

```
SELECT * FROM supervisor_transactions WHERE  
[purpose ~* 'obres'];
```

```
SELECT * FROM supervisor_transactions WHERE  
[purpose ~* '^obres'];
```

“obres” → “obresti” (eng. “money interest”)

Numeric operations

```
SELECT sum(amount_of_transaction) FROM  
supervisor_transactions;
```

```
SELECT count(*) FROM  
supervisor_transactions;
```

```
SELECT count(*) FROM supervisor_transactions  
WHERE (amount_of_transaction > 100000);
```

```
SELECT min(amount_of_transaction) FROM  
supervisor_transactions;
```

```
SELECT max(amount_of_transaction) FROM  
supervisor_transactions;
```

Numeric operations

```
SELECT avg(amount_of_transaction) FROM  
supervizor_transactions;
```

```
SELECT round(avg(amount_of_transaction), 2)  
AS average_payment FROM  
supervizor_transactions WHERE  
(budget_user_code = 16110);
```

```
SELECT round(avg(amount_of_transaction), 2)  
AS average_payment FROM  
supervizor_transactions WHERE NOT  
(budget_user_code = 16110);
```

Data aggregation

```
SELECT sum(amount_of_transaction),  
currency_code FROM supervisor_transactions  
GROUP BY currency_code;
```

```
SELECT count(*), currency_code FROM  
supervisor_transactions GROUP BY  
currency_code;
```

```
SELECT count(*) AS num_of_payments,  
EXTRACT(year from date_of_transaction) AS  
fiscal_year FROM supervisor_transactions  
WHERE (currency_code = 'EUR') GROUP BY  
fiscal_year ORDER BY fiscal_year;
```

Linking multiple tables

Table: companies

	registration_number text	tax_number integer	company_name text	region text	municipality text	company_form text	main_activity text
1	1001779000	46272305	RESTAVRACIJA BLEJ...	GORENJ...	Bled	Samostojni podjet...	55.301
2	1006622000	69010757	ORGANIZIRANJE GL...	POMUR...	Radenci	Samostojni podjet...	90.040
3	1016024000	72457708	EVA - CVETJE ZA D...	GORIŠKA	Nova Gorica	Samostojni podjet...	47.761
4	1028685000	83379908	ERKER LOVRENC JA...	OSRED...	Logatec	Samostojni podjet...	74.203
5	1028774000	27282562	MIZARSTVO MOLK ...	OSRED...	Logatec	Samostojni podjet...	16.230
6	1028871000	97507741	KNJIGOVODSKI SER...	OSRED...	Logatec	Samostojni podjet...	69.200
7	1033280000	48212105	GUSTINČIČ ZDENK...	GORIŠKA	Nova Gorica	Notar	69.102
8	1033514000	90215605	ROJEC ANTON - NO...	SAVINJS...	Celje	Notar	69.102
9	1033549000	66941067	ŠOEMEN ANDREJ - ...	PODRA...	Ptuj	Notar	69.102
10	1033794000	95108448	MARJETA SMITVIDA	GORIŠKA	Ajdovščina	Evidenčni svetov...	69.101

	debit_sub_account text	date_of_transaction date	amount_of_transaction numeric (16,4)	currency_code text	recipient_bank_account text	name_of_recipient text	registration_number text	tax_numt integer
11	011006300109972	2003-01-03	8419.9633	EUR	010000001910013	DBS d.d.	5349907000	18
12	011006300109972	2003-01-03	444.7922	EUR	043020000217749	ČASNIK FINANCE, d.o.o.	1353942000	19
13	011006300109972	2003-01-03	813.0758	EUR	043020000217749	ČASNIK FINANCE, d.o.o.	1353942000	19
14	011006300109972	2003-01-03	335.4031	EUR	043020000217749	ČASNIK FINANCE, d.o.o.	1353942000	19
15	011006300109972	2003-01-03	18.4839	EUR	060000026390798	CETIS d.d.	5042208000	24
16	011006300109972	2003-01-03	16720.0801	EUR	060000026390798	CETIS d.d.	5042208000	24
17	011006300109972	2003-01-03	2388.5829	EUR	060000026390798	CETIS d.d.	5042208000	24
8	011006300109972	2003-01-03	31.0278	EUR	242009004409728	PROTOCOL BANK d.o.o.	5670829000	28
9	011006300109972	2003-01-03	72143.5883	EUR	010000002400057	RAIFFEISEN BANKA D.O.O.	5706491000	29
10	011006300109972	2003-01-03	67315.5191	EUR	010000002400057	RAIFFEISEN BANKA D.O.O.	5706491000	29
11	011006300109972	2003-01-03	14345.5600	EUR	010000002400057	RAIFFEISEN BANKA D.O.O.	5706491000	29
12	011006300109972	2003-01-03	109304.5071	EUR	010000001800015	NLB BANKA DOMŽALE...	5101727000	38
13	011006300109972	2003-01-03	265982.3068	EUR	010000003029005	Sberbank d.d.	5496527000	41
14	011006300109972	2003-01-03	333.8341	EUR	020110018848763	SLOVENSKA KADROVS...	5213835000	46
15	011006300109972	2003-01-03	1500.8768	EUR	000100013724496	AMIDAS d.o.o.	5504643000	50
1	011006300109972	2003-01-03	1500.8768	EUR	000100013724496	AMIDAS d.o.o.	5504643000	50
1	011006300109972	2003-01-03	500000.0000	EUR	01000000011132	MINISTRSTVO ZA NOT...	5030200000	52

Table: supervizor_transactions

Linking multiple tables

```
SELECT * FROM supervisor_transactions,  
companies WHERE  
[supervisor_transactions.registration_number  
= companies.registration_number];
```

```
SELECT amount_of_transaction, currency_code,  
name_of_recipient, company_name FROM  
supervisor_transactions, companies WHERE  
[supervisor_transactions.registration_number  
= companies.registration_number] LIMIT 100;
```

Linking multiple tables

```
SELECT company_name AS "Company Name",
supervisor_transactions.tax_number AS "Tax
Number", sum(amount_of_transaction) AS
"Received Money", count(*) AS "Number Of
Payments" FROM supervisor_transactions,
companies WHERE
(companies.registration_number =
supervisor_transactions.registration_number)
GROUP BY company_name,
supervisor_transactions.tax_number ORDER BY
3 DESC;
```


Linking multiple tables

```
SELECT name AS "Budget User", company_name
AS "Company Name",
round(sum(amount_of_transaction),1) AS
"Received Money", count(*) AS "Number Of
Payments",
round(sum(amount_of_transaction)/count[*],1)
AS "Amount Per Payment" FROM
supervisor_transactions, companies,
budget_users WHERE
(companies.registration_number =
supervisor_transactions.registration_number)
AND (budget_users.budget_user_code =
supervisor_transactions.budget_user_code)
GROUP BY name, company_name ORDER BY "Amount
Per Payment" DESC;
```

Linking multiple tables

```
SELECT name AS "Budget User", company_name AS
"Company Name",
round(sum(amount_of_transaction),1) AS "Received
Money", count(*) AS "Number Of Payments",
round(sum(amount_of_transaction)/count(*),1) AS
"Amount Per Payment", description_slo,
description_eng FROM supervisor_transactions,
companies, budget_users, standard_activities
WHERE (companies.registration_number =
supervisor_transactions.registration_number) AND
(budget_users.budget_user_code =
supervisor_transactions.budget_user_code) AND
(right(standard_activities.category,6) =
main_activity) GROUP BY name, company_name,
description_slo, description_eng ORDER BY
"Amount Per Payment" DESC;
```

Linking multiple tables

The screenshot shows the pgAdmin 4 interface. On the left, the 'Browser' pane displays a tree view of the database structure, including 'Servers (2)', 'PostgreSQL 9.6', 'PostgreSQL 10', and 'Databases (2)'. The 'finance' database is selected, showing its various components like 'Casts', 'Catalogs', 'Event Triggers', 'Extensions', 'Foreign Data Wrappers', 'Languages', and 'Schemas'. The main window shows a SQL query in the 'Query' editor:

```
1 tegrity,6) = main_activity) GROUP BY name, company_name, description_slo, description_eng ORDER BY "Amount Per Payment" DESC;
```

Below the query editor, the 'Data Output' pane displays a table with 17 rows of data. The columns are: Budget User text, Company Name text, Received Money numeric, Number Of Payments bigint, Amount Per Payment numeric, description_slo text, and description_eng text.

	Budget User text	Company Name text	Received Money numeric	Number Of Payments bigint	Amount Per Payment numeric	description_slo text	description_eng text
1	MINISTRSTVO ZA FINANCE	KOŠAK MIRO - NOTAR	49500000.0	1	49500000.0	Notariat	Activities of notaries pu
2	MINISTRSTVO ZA FINANCE	ZAVOD ZA POKOJNINSKO I...	15902081886.4	538	29557773.0	Dejavnost obvezn...	Compulsory social sec
3	MINISTRSTVO ZA FINANCE	UPRAVA REPUBLIKE SLOVE...	22345774612.8	849	26320111.4	Splošna dejavnost ...	General public adminis
4	MINISTRSTVO ZA FINANCE	SKLAD OBRTNIKOV IN POD...	109775801.0	14	7841128.6	Dejavnost pokojni...	Pension funding
5	MINISTRSTVO ZA FINANCE	Factor banka d.d.	167288933.7	49	3414059.9	Drugo denarno po...	Other monetary interm
6	MINISTRSTVO ZA FINANCE	Nafta Lendava, proizvodnj...	10000050.1	3	3333350.0	Proizvodnja naftn...	Manufacture of refinec
7	MINISTRSTVO ZA FINANCE	KAPITALSKA DRUŽBA POK...	90052515.8	30	3001750.5	Dejavnost pokojni...	Pension funding
8	MINISTRSTVO ZA FINANCE	PROBANKA, d.d.	163839978.5	63	2600634.6	Drugo denarno po...	Other monetary interm
9	MINISTRSTVO ZA FINANCE	NADŠKOFIJA MARIBOR	7444964.5	3	2481654.8	Dejavnost verskih ...	Activities of religious o
10	MINISTRSTVO ZA FINANCE	HOLDING SLOVENSKE ELE...	1998689.6	1	1998689.6	Trgovanje z elektri...	Trade of electricity
11	MINISTRSTVO ZA FINANCE	MINISTRSTVO ZA FINANCE	19988831.4	11	1817166.5	Splošna dejavnost ...	General public adminis
12	MINISTRSTVO ZA FINANCE	SID - Slovenska izvozna in ...	258379494.5	143	1806849.6	Drugo denarno po...	Other monetary interm
13	MINISTRSTVO ZA FINANCE	NOVA LJUBLJANSKA BANK...	4282342027.7	2538	1687290.0	Drugo denarno po...	Other monetary interm
14	MINISTRSTVO ZA FINANCE	SLOVENSKE ŽELEZNICE, d...	111286830.7	67	1660997.5	Dejavnost uprav p...	Activities of head office
15	MINISTRSTVO ZA FINANCE	NOVA KREDITNA BANKA M...	889261092.4	558	1593657.9	Drugo denarno po...	Other monetary interm
16	MINISTRSTVO ZA FINANCE	DRAVSKE ELEKTRARNE MA...	1436882.5	1	1436882.5	Proizvodnja elektri...	Production of electricit
17	MINISTRSTVO ZA FINANCE	JOHNSON CONTROLS SLO...	1292920.7	1	1292920.7	Proizvodnja drugih...	Manufacture of other p

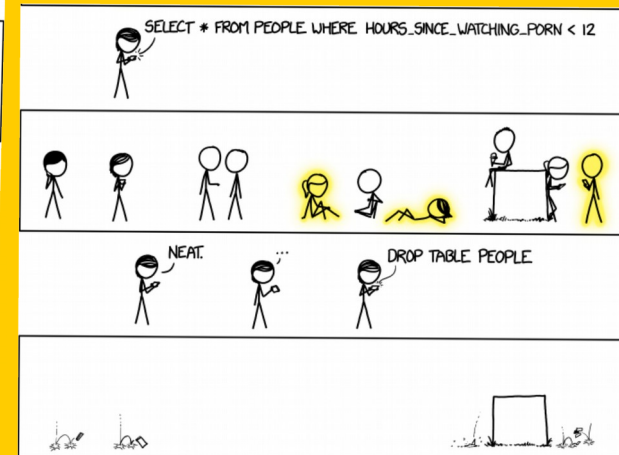
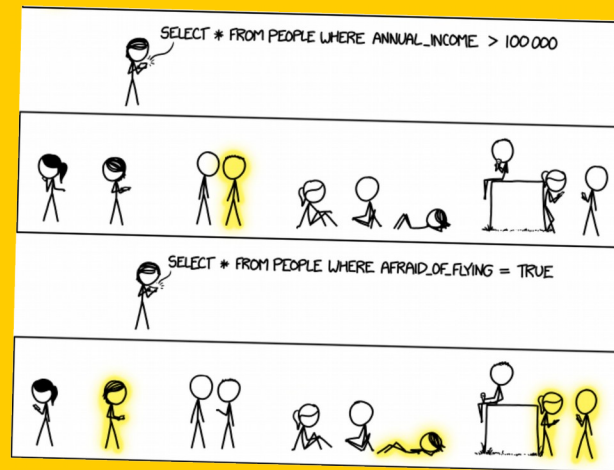
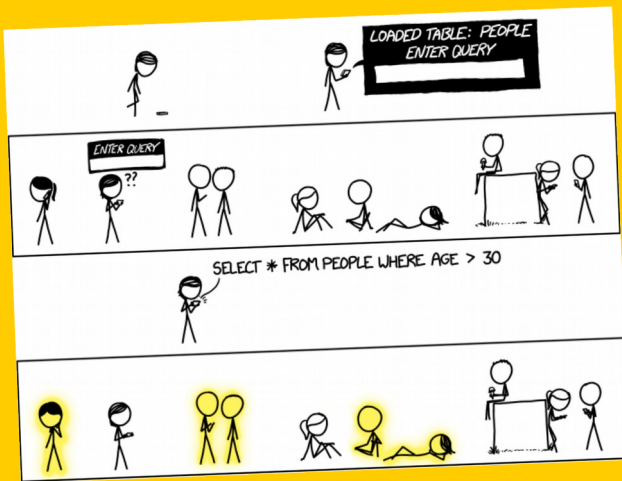
The Windows taskbar at the bottom shows the system tray with the date 17:51 and 18.11.2017.

Saving results to external file

```
COPY (SELECT count(*) as num_of_payments,  
EXTRACT(year from date_of_transaction) as  
fiscal_year FROM supervisor_transactions  
GROUP BY fiscal_year) to  
'C:\psql\results.csv' with csv header  
delimiter ';';
```

... and have fun! :-)

SELECT *questions* FROM *audience*;



Cartoon: (CC) xkcd.com

Matej Kovačič
matej.kovacic@ijs.si

Personal blog: <https://telefoncek.si>