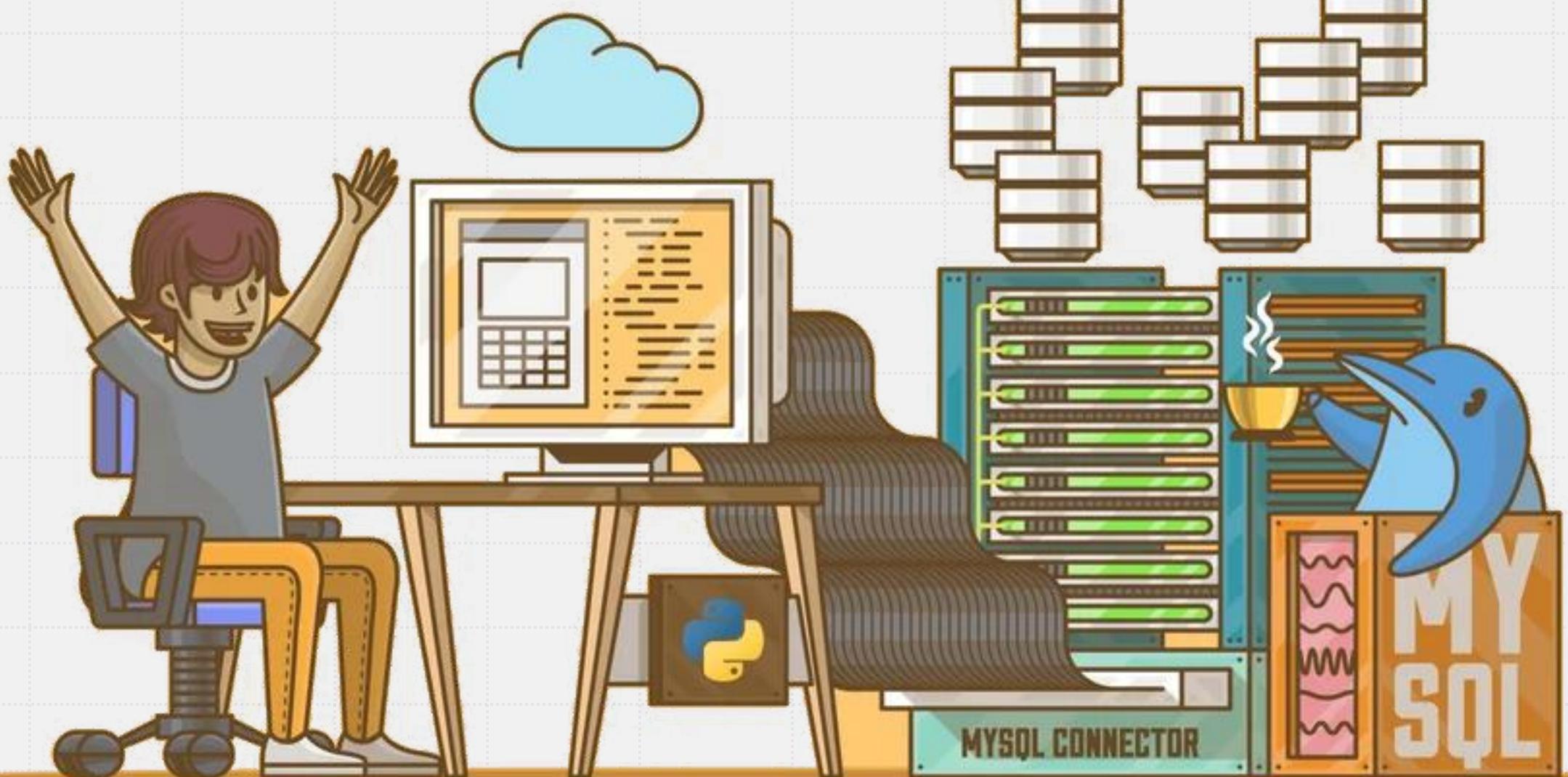


# Python

Programowanie baz danych

Dodatki



# Bazy danych

# Baza danych

Zbiór danych opisujący pewien wybrany fragment rzeczywistości.

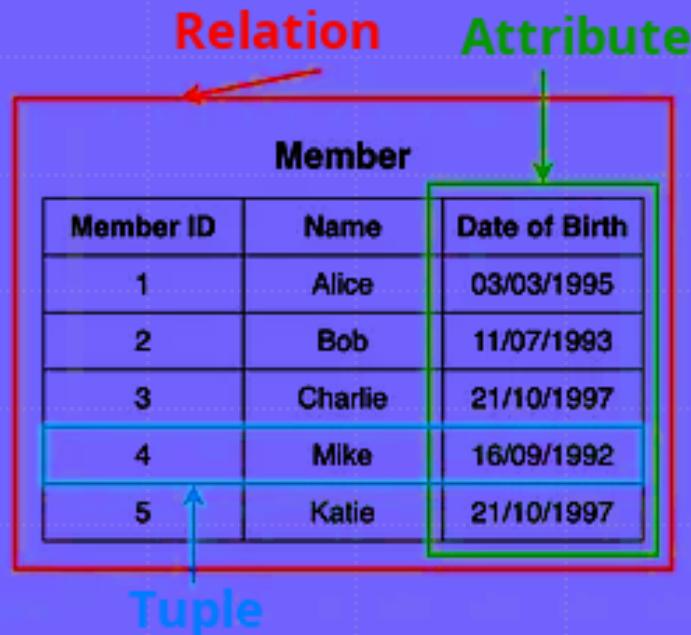


# Rodzaje baz danych

W zależności od sposobu zorganizowania danych (tzw. modelu danych) wyróżniamy:

- **Relacyjne bazy danych**
  - Bazy danych NoSQL
  - Obiektowe bazy danych
  - Hierarchiczne bazy danych
  - Sieciowe bazy danych
  - Grafowe bazy danych
  - ...

# Relacyjny model danych



Relational  
Algebra  
Operators

$\pi$   $\sigma$   $\rho$   
 $\times$   $\cup$

# Relacyjny model danych

Relacyjne bazy danych oparte są o relacyjny model danych. W relacyjnym modelu danych podstawową strukturą danych jest **relacja**. Każda relacja posiada tzw. schemat. Schemat relacji to lista atrybutów. Każdy atrybut posiada swoją domenę (dziedzinę). Dziedzina definiuje zbiór wartości jakie może przyjmować atrybut. W relacyjnym modelu danych baza danych jest zbiorem relacji.

Twórcą relacyjnego modelu danych jest brytyjski matematyk (computer scientist) **Edgar Frank "Ted" Codd**. Koncepcję relacyjnego modelu danych przedstawił w **1970** roku w swojej pracy "A Relational Model of Data for Large Shared Data Banks". W pracy sformułował 13 reguł znanych dzisiaj pod nazwą "**12 postulatów Codda**". Dwa lata później, w swojej kolejnej pracy "Relational Completeness of Data Base Sublanguages" Codd rozwinął swoją teorię tworząc fundamenty nowej dziedziny matematycznej znanej dzisiaj pod nazwą **algebra relacyjna**.

Chociaż relacja posiada ściśle określona definicję matematyczną, a cały model opiera się na abstrakcyjnym koncepcie matematycznym, to ze względu na prostotę i intuicyjność relacje najczęściej wprowadza się obrazowo w postaci **tabelek**.



## **System zarządzania bazą danych**

(DBMS – Database Management System)

Oprogramowanie służące do zarządzania bazą danych (w tym przetwarzania danych oraz administracji)

# Baza danych vs system zarządzania bazą danych

**Baza danych** poprzez model danych definiuje w jakiej postaci będą przechowywane umieszczone w niej dane oraz definiuje zestaw możliwych operacji na tych danych.

**System zarządzania bazą danych** to oprogramowanie umożliwiające umieszczanie danych w tej bazie oraz przetwarzanie ich zgodnie z modelem danych.

# DBMS

DataBase Management System

# System zarządzania bazą danych

Database Files



DBMS

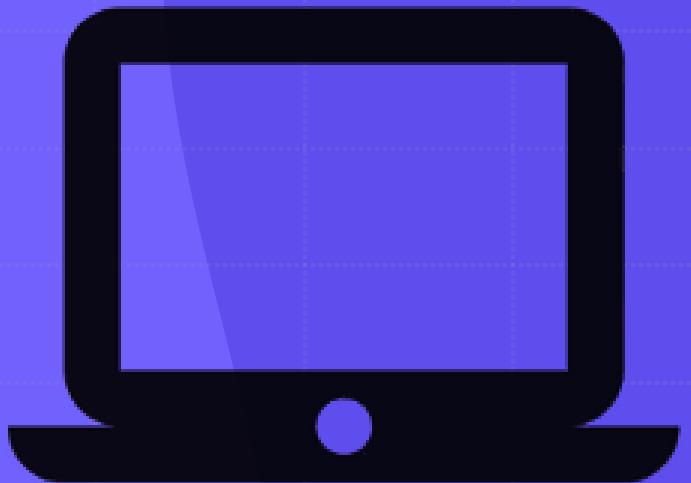


Większość współczesnych systemów zarządzania bazami danych oparta jest na architekturze klient-serwer

# Architektura klient-serwer



Server



Client

# RDBMS

Relational DataBase Management System



# **Relacyjny system zarządzania bazą danych**

(RDBMS – Relational Database Management System)

Oprogramowanie służące do  
zarządzania relacyjną bazą  
danych

# Popularne relacyjne systemy zarządzania bazami danych

- MySQL
- PostgreSQL
- SQLite
- Microsoft SQL Server (MSSQL)
- Oracle Database
  - MariaDB
  - IBM Db2
  - ...

System zarządzania  
relacyjną bazą  
danych

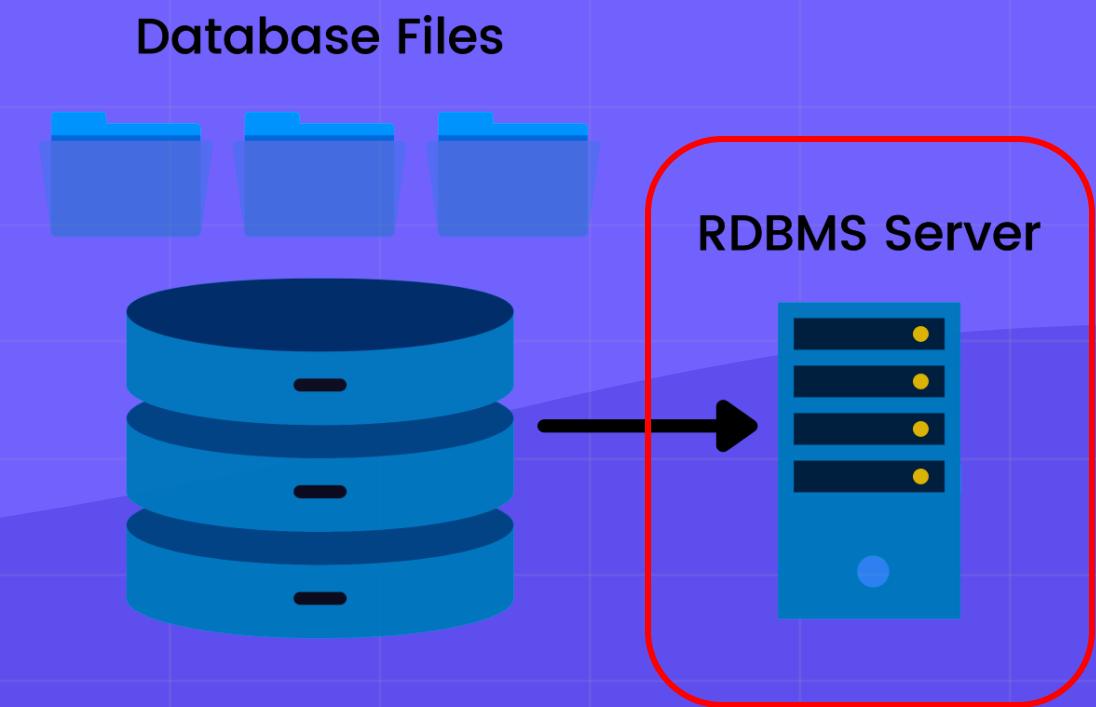
Database Files



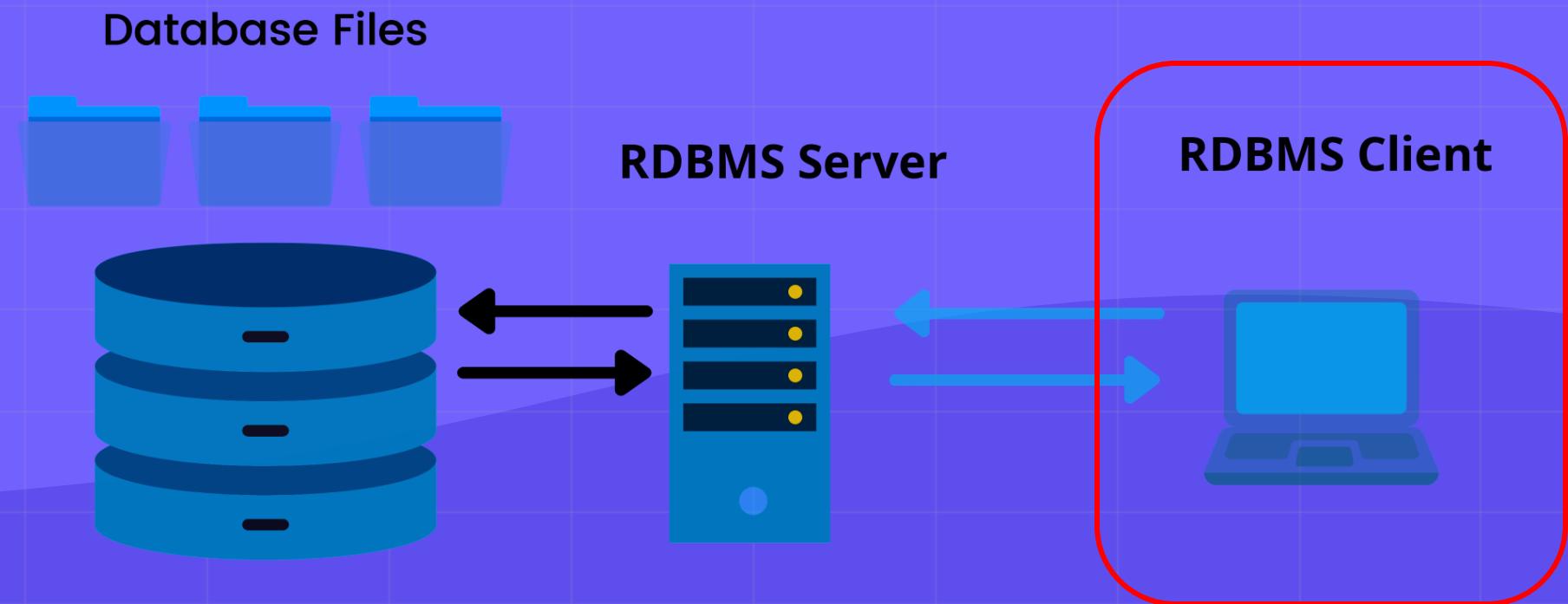
RDBMS



# Serwer bazodanowy

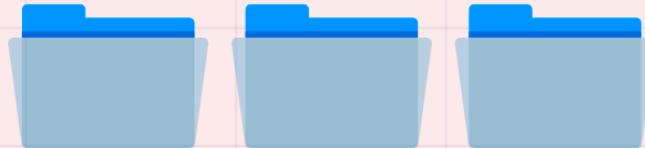


# Klient bazodanowy

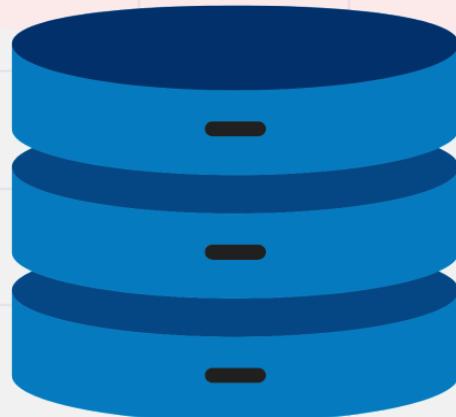


# MySQL

Database Files



MySQL Server  
(mysqld)



MySQL Client  
(Workbench)



# MySQL



*The world's most popular open source database*



Contact MySQL | Login | Register

mysol.com

DOWNLOAD

DOCUMENTATION

DFVFI OPER ZON



# MySQL Shorts

## Video Series

**WATCH NOW**



MySQL HeatWave

MySQL HeatWave is a fully managed database service for transactions, real- time analytics across data warehouses and data lakes, and machine learning services, without the complexity, latency, and cost of ETL duplication. It is available on OCI, AWS, and Azure.

[Learn More](#)



MySQL for OEM/ISV

Over 2000 ISVs, OEMs, and VARs rely on MySQL as their products' embedded database to make their applications, hardware and appliances more competitive, bring them to market faster, and lower their cost of goods sold.

[Learn More](#)



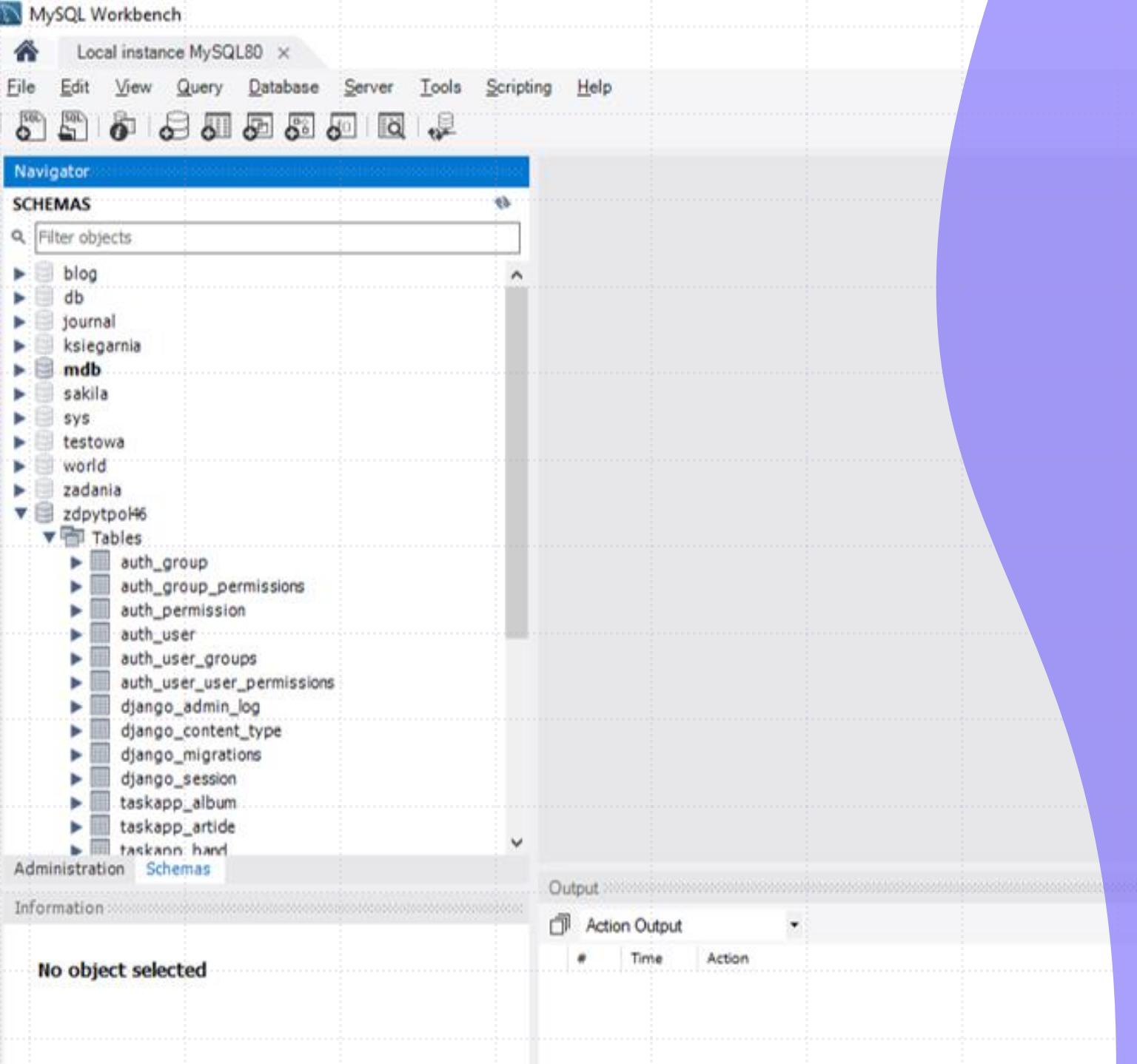
MySQL Enterprise Edition

The most comprehensive set of advanced features, management tools and technical support to achieve the highest levels of MySQL availability, security, reliability, and uptime.

[Learn More](#)



MySQL Cluster CG



# MySQL GUI Client Workbench

The screenshot shows the phpMyAdmin interface running on a local server at `localhost / 127.0.0.1 | phpMyAdmin`. The main navigation bar includes links for Bazy danych, SQL, Status, Konta użytkowników, Eksport, Import, Ustawienia, Dziennik binarny, and Replikacja.

The left sidebar lists databases: Nowa, blog, db, information\_schema, journal, ksiegarnia, mdb, mysql, performance\_schema, sakila, sys, testowa, world, zadania, and zdpytppol46.

The central area displays two configuration panels:

- Ustawienia ogólne (General Settings):** Includes options to change password, set connection sorting (currently set to `utf8mb4_unicode_ci`), and access more settings.
- Ustawienia wyglądu (Appearance Settings):** Shows the language set to Polish (Polski - Polish) and the theme set to pmahomme, with a "View all" button.

A large watermark in the bottom right corner reads "MySQL GUI Client" and "phpMyAdmin".

**Serwer bazy danych (Database Server):**

- Serwer: 127.0.0.1 via TCP/IP
- Typ serwera: MySQL
- Połączenie z serwerem: SSL nie jest używany
- Wersja serwera: 8.0.26 - MySQL Community Server - C
- Wersja protokołu: 10
- Użytkownik: root@localhost
- Kodowanie znaków serwera: UTF-8 Unicode (utf8mb4)

**Serwer WWW:**

- Apache/2.4.53 (Win64) OpenSSL/1.1.1n PHP/7.4.29
- Wersja klienta bazy danych: libmysql - mysqlnd 7.4.29
- Rozszerzenie PHP: mysqli, MySQLi, MySQLiString
- Wersja PHP: 7.4.29

**phpMyAdmin:**

- Informacja o wersji: 5.2.0 (aktualna)
- Dokumentacja
- Oficjalna strona phpMyAdmin

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select \* from zdpypol46.django\_migrations;

+-----+ <th>+-----+<th>+-----+</th></th>	+-----+ <th>+-----+</th>	+-----+	
id   app	name	applied	
1	contenttypes	0001_initial	2022-03-28 17:45:13.807284
2	auth	0001_initial	2022-03-28 17:45:14.926662
3	admin	0001_initial	2022-03-28 17:45:15.208500
4	admin	0002_logentry_remove_auto_add	2022-03-28 17:45:15.227488
5	admin	0003_logentry_add_action_flag_choices	2022-03-28 17:45:15.244480
6	contenttypes	0002_remove_content_type_name	2022-03-28 17:45:15.560297
7	auth	0002_alter_permission_name_max_length	2022-03-28 17:45:15.674246
8	auth	0003_alter_user_email_max_length	2022-03-28 17:45:15.710210
9	auth	0004_alter_user_username_opts	2022-03-28 17:45:15.726201
10	auth	0005_alter_user_last_login_null	2022-03-28 17:45:15.839137
11	auth	0006_require_contenttypes_0002	2022-03-28 17:45:15.844132
12	auth	0007_alter_validators_add_error_messages	2022-03-28 17:45:15.858125
13	auth	0008_alter_user_username_max_length	2022-03-28 17:45:15.989051
14	auth	0009_alter_user_last_name_max_length	2022-03-28 17:45:16.086009
15	auth	0010_alter_group_name_max_length	2022-03-28 17:45:16.125987
16	auth	0011_update_proxy_permissions	2022-03-28 17:45:16.140977
17	auth	0012_alter_user_first_name_max_length	2022-03-28 17:45:16.255906
18	sessions	0001_initial	2022-03-28 17:45:16.308355
19	taskapp	0001_initial	2022-03-28 17:45:16.354036
20	taskapp	0002_band	2022-03-28 17:45:16.393835
21	taskapp	0003_category	2022-03-28 17:45:16.438843
22	taskapp	0004_alter_category_description	2022-03-28 17:45:16.500810
23	taskapp	0005_article	2022-03-28 17:45:16.542807
24	taskapp	0006_album	2022-03-28 17:45:16.596777
25	taskapp	0007_capitol_language_movie_framework_country_character	2022-03-28 17:45:17.217439
26	taskapp	0008_album_band	2022-03-28 17:45:17.330310

26 rows in set (0.00 sec)

mysql>

MySQL CLI Client  
mysql



# PostgreSQL



PostgreSQL: The world's most advanced open source relational database

8th February 2024: PostgreSQL 16.2, 15.6, 14.11, 13.14, and 12.18 Released!

# PostgreSQL: The World's Most Advanced Open Source Relational Database

[Download →](#) [New to PostgreSQL?](#)



## New to PostgreSQL?

PostgreSQL is a powerful, open source object-relational database system with over 35 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance. There is a wealth of information to be found describing how to [install](#) and [use PostgreSQL](#) through the [official documentation](#). The [open source community](#) provides many helpful places to become familiar with PostgreSQL, discover how it works, and find career opportunities. Learn more on how to [engage with the community](#).

[Learn More](#) [Feature Matrix](#)



## Latest Releases

**2024-02-08 - PostgreSQL 16.2, 15.6, 14.11, 13.14, and 12.18 Released!**

The PostgreSQL Global Development Group has released an update to all supported versions of PostgreSQL, including 16.2, 15.6, 14.11, 13.14, and 12.18. This release fixes one [security vulnerability](#) and over 65 bugs reported over the last several months.

If you use GIN indexes, you may need to reindex after updating to this release. Please see the [release notes](#) for more information.

For the more information about this release, please see the [release notes](#). You can download Postgre

**16.2 · 2024-02-08**

pgAdmin

File ▾ Object ▾ Tools ▾ Help ▾

Browser

- zdpypol46
  - Casts
  - Catalogs
  - Event Triggers
  - Extensions
  - Foreign Data Wrappers
  - Languages
  - Publications
  - Schemas (1)
    - public
      - Collations
      - Domains
      - FTS Configurations
      - FTS Dictionaries
      - FTS Parsers
      - FTS Templates
      - Foreign Tables
      - Functions
      - Materialized Views
      - Procedures
      - Sequences
    - Tables (22)
      - auth\_group

Dashboard

Properties SQL Statistics Dependencies Dependents

Database sessions

Total	Active	Idle
1	1	0

Transactions per second

	2	1	0
1	2	1	0

Tuples in

Inserts	Updates	Delete
1	0	0

Tuples out

Fetched	Returned
3500	3000
3000	2500
2500	2000
2000	1500
1500	1000
1000	500
500	0

Server activity

Sessions Locks Prepared Transactions

PID	User	Application	Client	Backend start
15904	postgres	pgAdmin 4 - DB:zdpypol46	::1	2022-03-29 13:11:57 CEST

# PostgreSQL GUI Client PgAdmin

The screenshot shows the phpPgAdmin interface for PostgreSQL 14.1. The title bar indicates the application is running on localhost:5432 as the "postgres" user. The left sidebar lists several databases: db, exercises, financeblog, postgres, python\_module5, python\_module5\_dev, tdd, tdd\_dev, test, and zdpytpol46. The main area displays a table titled "Bazy danych" (Databases) with the following data:

Baza danych	Właściciel	Kodowanie	Collation	Character Type	Przestrzeń tabel	Rozmiar	Akcje	Komentarz	
db	postgres	UTF8	Polish_Poland.1250	Polish_Poland.1250	pg_default	9001 kB	Usuń	Uprawnienia	Zmień
exercises	postgres	UTF8	Polish_Poland.1250	Polish_Poland.1250	pg_default	9729 kB	Usuń	Uprawnienia	Zmień
financeblog	postgres	UTF8	Polish_Poland.1250	Polish_Poland.1250	pg_default	9393 kB	Usuń	Uprawnienia	Zmień
postgres	postgres	UTF8	Polish_Poland.1250	Polish_Poland.1250	pg_default	8409 kB	Usuń	Uprawnienia	Zmień
python_module5	postgres	UTF8	Polish_Poland.1250	Polish_Poland.1250	pg_default	9913 kB	Usuń	Uprawnienia	Zmień
python_module5_dev	postgres	UTF8	Polish_Poland.1250	Polish_Poland.1250	pg_default	10 MB	Usuń	Uprawnienia	Zmień
tdd	postgres	UTF8	Polish_Poland.1250	Polish_Poland.1250	pg_default	9025 kB	Usuń	Uprawnienia	Zmień
tdd_dev	postgres	UTF8	Polish_Poland.1250	Polish_Poland.1250	pg_default	9025 kB	Usuń	Uprawnienia	Zmień
test	postgres	UTF8	Polish_Poland.1250	Polish_Poland.1250	pg_default	8401 kB	Usuń	Uprawnienia	Zmień
zdpytpol46	postgres	UTF8	Polish_Poland.1250	Polish_Poland.1250	pg_default	9409 kB	Usuń	Uprawnienia	Zmień

Below the table, there is a section for "Actions on multiple lines" with buttons for "Select all / Unselect all", "Wykonaj" (Execute), and a dropdown menu. A link "Utwórz bazę danych" (Create database) is also present.

**PostgreSQL GUI Client  
phpPgAdmin**

Password for user postgres:

psql (14.1)

WARNING: Console code page (852) differs from Windows code page (1250)  
8-bit characters might not work correctly. See psql reference  
page "Notes for Windows users" for details.

Type "help" for help.

postgres=# \c zdpytpol46

You are now connected to database "zdpytpol46" as user "postgres".

zdpytpol46=# select \* from django\_migrations;

id	app	name	applied
1	contenttypes	0001_initial	2022-03-28 19:28:11.83978+02
2	auth	0001_initial	2022-03-28 19:28:12.010683+02
3	admin	0001_initial	2022-03-28 19:28:12.050658+02
4	admin	0002_logentry_remove_auto_add	2022-03-28 19:28:12.068648+02
5	admin	0003_logentry_add_action_flag_choices	2022-03-28 19:28:12.07964+02
6	contenttypes	0002_remove_content_type_name	2022-03-28 19:28:12.116621+02
7	auth	0002_alter_permission_name_max_length	2022-03-28 19:28:12.128614+02
8	auth	0003_alter_user_email_max_length	2022-03-28 19:28:12.142609+02
9	auth	0004_alter_user_username_opts	2022-03-28 19:28:12.153598+02
10	auth	0005_alter_user_last_login_null	2022-03-28 19:28:12.164591+02
11	auth	0006_require_contenttypes_0002	2022-03-28 19:28:12.166591+02
12	auth	0007_alter_validators_add_error_messages	2022-03-28 19:28:12.176586+02
13	auth	0008_alter_user_username_max_length	2022-03-28 19:28:12.201572+02
14	auth	0009_alter_user_last_name_max_length	2022-03-28 19:28:12.212564+02
15	auth	0010_alter_group_name_max_length	2022-03-28 19:28:12.234551+02
16	auth	0011_update_proxy_permissions	2022-03-28 19:28:12.245545+02
17	auth	0012_alter_user_first_name_max_length	2022-03-28 19:28:12.25454+02
18	sessions	0001_initial	2022-03-28 19:28:12.279527+02
19	taskapp	0001_initial	2022-03-28 19:28:12.307522+02
20	taskapp	0002_band	2022-03-28 19:28:12.309512+02
21	taskapp	0003_category	2022-03-28 19:28:12.322503+02
22	taskapp	0004_alter_category_description	2022-03-28 19:28:12.3295+02
23	taskapp	0005_article	2022-03-28 19:28:12.351487+02
24	taskapp	0006_album	2022-03-28 19:28:12.362479+02
25	taskapp	0007_capitol_language_movie_framework_country_character	2022-03-28 19:28:12.453437+02
26	taskapp	0008_album_band	2022-03-28 19:28:12.467426+02

(26 rows)

zdpytpol46=#

PostgreSQL CLI Client  
psql

# SQLite

SQLite Home Page + <https://www.sqlite.org>

# SQLite

Small. Fast. Reliable.  
Choose any three.

Home About Documentation Download License Support Purchase Search

## What Is SQLite?

SQLite is a C-language library that implements a [small](#), [fast](#), [self-contained](#), [high-reliability](#), [full-featured](#), SQL database engine. SQLite is the [most used](#) database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day. [More Information...](#)

The SQLite [file format](#) is stable, cross-platform, and backwards compatible and the developers pledge to keep it that way [through the year 2050](#). SQLite database files are commonly used as containers to transfer rich content between systems [\[1\]](#) [\[2\]](#) [\[3\]](#) and as a long-term archival format for data [\[4\]](#). There are over 1 trillion (1e12) SQLite databases in active use [\[5\]](#).

SQLite [source code](#) is in the [public-domain](#) and is free to everyone to use for any purpose.

## Latest Release

Version 3.45.1 (2024-01-30). [Download](#) [Prior Releases](#)

### Common Links

- Features
- When to use SQLite
- Getting Started
- Try it live!
- Prior Releases
- SQL Syntax
  - Pragmas
  - SQL functions
  - Date & time functions
  - Aggregate functions
  - Window functions
  - Math functions
  - JSON functions
- C/C++ Interface Spec
  - Introduction
  - List of C-language APIs
- The TCL Interface Spec
- Quirks and Gotchas
- Frequently Asked Questions
- Commit History
- Bugs
- News

Ongoing development and support of SQLite is made possible in part by [SQLite Consortium](#) members, including:



Nowa baza danych Otwórz bazę danych... Zapisz zmiany Wycofaj zmiany Otwórz projekt Zapisz projekt Dołącz bazę danych Zamknij bazę danych

Struktura danych Przeglądarka danych Polecenia Pragmy Polecenia SQL

Utwórz tabelę... Utwórz indeks... Zmień tabelę... Usuń tabelę... Drukuj

Nazwa Rodzaj Polecenie tworzące

<b>Tabele (23)</b>		
>  auth_group		CREATE TABLE "auth_group" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(150) NOT NULL UNIQUE)
>  auth_group_permissions		CREATE TABLE "auth_group_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "group_id" integer NOT NULL REFERENCES "auth_group" ("id") NOT NULL, "permission_id" integer NOT NULL REFERENCES "auth_permission" ("id") NOT NULL)
>  auth_permission		CREATE TABLE "auth_permission" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "content_type_id" integer NOT NULL REFERENCES "django_content_type" ("id") NOT NULL, "codename" varchar(100) NOT NULL, "name" varchar(50) NOT NULL, "model" varchar(100) NOT NULL)
>  auth_user		CREATE TABLE "auth_user" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "password" varchar(128) NOT NULL, "last_login" datetime NULL, "is_superuser" integer NOT NULL, "username" varchar(150) NOT NULL, "first_name" varchar(30) NOT NULL, "last_name" varchar(30) NOT NULL, "email" varchar(254) NOT NULL, "is_staff" integer NOT NULL, "groups_id" integer NOT NULL REFERENCES "auth_user_groups" ("id"), "user_permissions_id" integer NOT NULL REFERENCES "auth_user_user_permissions" ("id"))
>  auth_user_groups		CREATE TABLE "auth_user_groups" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "user_id" integer NOT NULL REFERENCES "auth_user" ("id") NOT NULL, "group_id" integer NOT NULL REFERENCES "auth_group" ("id"))
>  auth_user_user_permissions		CREATE TABLE "auth_user_user_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "user_id" integer NOT NULL REFERENCES "auth_user" ("id") NOT NULL, "permission_id" integer NOT NULL REFERENCES "auth_permission" ("id"))
>  django_admin_log		CREATE TABLE "django_admin_log" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "action_time" datetime NOT NULL, "object_id" text NULL, "change_message" text NULL, "content_type_id" integer NOT NULL REFERENCES "django_content_type" ("id") NOT NULL, "user_id" integer NOT NULL REFERENCES "auth_user" ("id"))
>  django_content_type		CREATE TABLE "django_content_type" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app_label" varchar(100) NOT NULL, "model" varchar(100) NOT NULL)
>  django_migrations		CREATE TABLE "django_migrations" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app" varchar(255) NOT NULL, "name" varchar(255) NOT NULL, "applied" datetime NOT NULL)
>  django_session		CREATE TABLE "django_session" ("session_key" varchar(40) NOT NULL PRIMARY KEY, "session_data" text NOT NULL, "expire_date" datetime NOT NULL)
>  sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
>  taskapp_album		CREATE TABLE "taskapp_album" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "title" varchar(128) NOT NULL, "year" integer NOT NULL, "rating" float NOT NULL, "description" text NULL)
>  taskapp_article		CREATE TABLE "taskapp_article" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "title" varchar(128) NOT NULL, "author" varchar(64) NULL, "text" text NULL)
>  taskapp_band		CREATE TABLE "taskapp_band" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(64) NOT NULL, "year" integer NOT NULL, "style" varchar(64) NOT NULL)
>  taskapp_capitol		CREATE TABLE "taskapp_capitol" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(64) NOT NULL)
>  taskapp_category		CREATE TABLE "taskapp_category" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(64) NOT NULL, "description" text NULL)
>  taskapp_character		CREATE TABLE "taskapp_character" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(64) NOT NULL)
>  taskapp_character_movies		CREATE TABLE "taskapp_character_movies" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "character_id" bigint NOT NULL REFERENCES "taskapp_character" ("id"), "movie_id" bigint NOT NULL REFERENCES "taskapp_movie" ("id"))
>  taskapp_country		CREATE TABLE "taskapp_country" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(64) NOT NULL, "capitol_id" bigint NOT NULL REFERENCES "taskapp_capitol" ("id"))
>  taskapp_framework		CREATE TABLE "taskapp_framework" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(64) NOT NULL, "language_id" integer NOT NULL REFERENCES "taskapp_language" ("id"))
>  taskapp_language		CREATE TABLE "taskapp_language" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(64) NOT NULL)
>  taskapp_movie		CREATE TABLE "taskapp_movie" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "title" varchar(128) NOT NULL)
>  taskapp_task		CREATE TABLE "taskapp_task" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(64) NOT NULL)
Indeksy (20)		
>  auth_group_permissions_group_id_b120cbf9...		CREATE INDEX "auth_group_permissions_group_id_b120cbf9" ON "auth_group_permissions" ("group_id")
>  auth_group_permissions_group_id_pe...		CREATE UNIQUE INDEX "auth_group_permissions_group_id_permission_id_0cd325b0_uniq" ON "auth_group_permissions" ("group_id", "permission_id")
>  auth_group_permissions_permission_id_...		CREATE INDEX "auth_group_permissions_permission_id_a0c99a29" ON "auth_group_permissions" ("permission_id")

# SOLITE GUI Client DB Browser

```
C:\Windows\System32\cmd.exe - sqlite3.exe db.sqlite3
Microsoft Windows [Version 10.0.19044.1586]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\jerem\PycharmProjects\ZDPYTpol46_backend>sqlite3.exe db.sqlite3
SQLite version 3.38.2 2022-03-26 13:51:10
Enter ".help" for usage hints.

sqlite> select * from django_migrations;
1|contenttypes|0001_initial|2022-03-24 19:10:57.309449
2|auth|0001_initial|2022-03-24 19:10:57.370417
3|admin|0001_initial|2022-03-24 19:10:57.400400
4|admin|0002_logentry_remove_auto_add|2022-03-24 19:10:57.424383
5|admin|0003_logentry_add_action_flag_choices|2022-03-24 19:10:57.449367
6|contenttypes|0002_remove_content_type_name|2022-03-24 19:10:57.491344
7|auth|0002_alter_permission_name_max_length|2022-03-24 19:10:57.516330
8|auth|0003_alter_user_email_max_length|2022-03-24 19:10:57.534318
9|auth|0004_alter_user_username_opts|2022-03-24 19:10:57.553308
10|auth|0005_alter_user_last_login_null|2022-03-24 19:10:57.575310
11|auth|0006_require_contenttypes_0002|2022-03-24 19:10:57.585290
12|auth|0007_alter_validators_add_error_messages|2022-03-24 19:10:57.604284
13|auth|0008_alter_user_username_max_length|2022-03-24 19:10:57.625283
14|auth|0009_alter_user_last_name_max_length|2022-03-24 19:10:57.644256
15|auth|0010_alter_group_name_max_length|2022-03-24 19:10:57.667249
16|auth|0011_update_proxy_permissions|2022-03-24 19:10:57.682234
17|auth|0012_alter_user_first_name_max_length|2022-03-24 19:10:57.707235
18|sessions|0001_initial|2022-03-24 19:10:57.729217
19|taskapp|0001_initial|2022-03-24 19:38:00.782212
20|taskapp|0002_band|2022-03-24 19:38:00.791208
21|taskapp|0003_category|2022-03-24 19:46:17.921446
22|taskapp|0004_alter_category_description|2022-03-24 19:47:03.022597
23|taskapp|0005_article|2022-03-24 19:55:11.537016
24|taskapp|0006_album|2022-03-24 20:03:30.393453
25|taskapp|0007_capitol_language_movie_framework_country_character|2022-03-28 16:31:17.736793
26|taskapp|0008_album_band|2022-03-28 16:50:35.520866
27|taskapp|0009_alter_character_table|2022-03-28 17:08:13.934086
28|taskapp|0010_alter_character_table|2022-03-28 17:10:53.656707
sqlite>
```

SQLite CLI Client  
sqlite3

SQLite Tutorial - An Easy Way to +

https://www.sqlitetutorial.net

SQLITE TUTORIAL

HOME START HERE VIEWS INDEXES TRIGGERS FUNCTIONS API TRY IT

## SQLite Tutorial

This **SQLite tutorial** teaches you everything you need to know to start using SQLite effectively. In this tutorial, you will learn SQLite step by step through extensive hands-on practices.

This SQLite tutorial is designed for developers who want to use SQLite as the back-end database or to use SQLite to manage structured data in applications including desktop, web, and mobile apps.

SQLite is an open-source, zero-configuration, self-contained, stand-alone, transaction relational database engine designed to be embedded into an application.

### Getting started with SQLite

You should go through this section if this is the first time you have worked with SQLite. Follow these 4-easy steps to get started with SQLite fast.

- First, help you answer the first and important question: what is SQLite? You will have a brief overview of SQLite.
- Second, show you step by step how to download and install the SQLite tools on your computer.
- Third, introduce you to an SQLite sample database and walk you through the steps of using the sample database for practicing.
- Finally, guide you on how to use the sqlite3 commands.

### Basic SQLite tutorial

This section presents basic SQL statements that you can use with SQLite. You will first start querying data from the sample database. If you are already familiar with SQL, you will notice the differences between SQL standard and SQL dialect used in SQLite.

Section 1. Simple query

Search ...

GETTING STARTED

What Is SQLite  
Download & Install SQLite  
SQLite Sample Database  
SQLite Commands

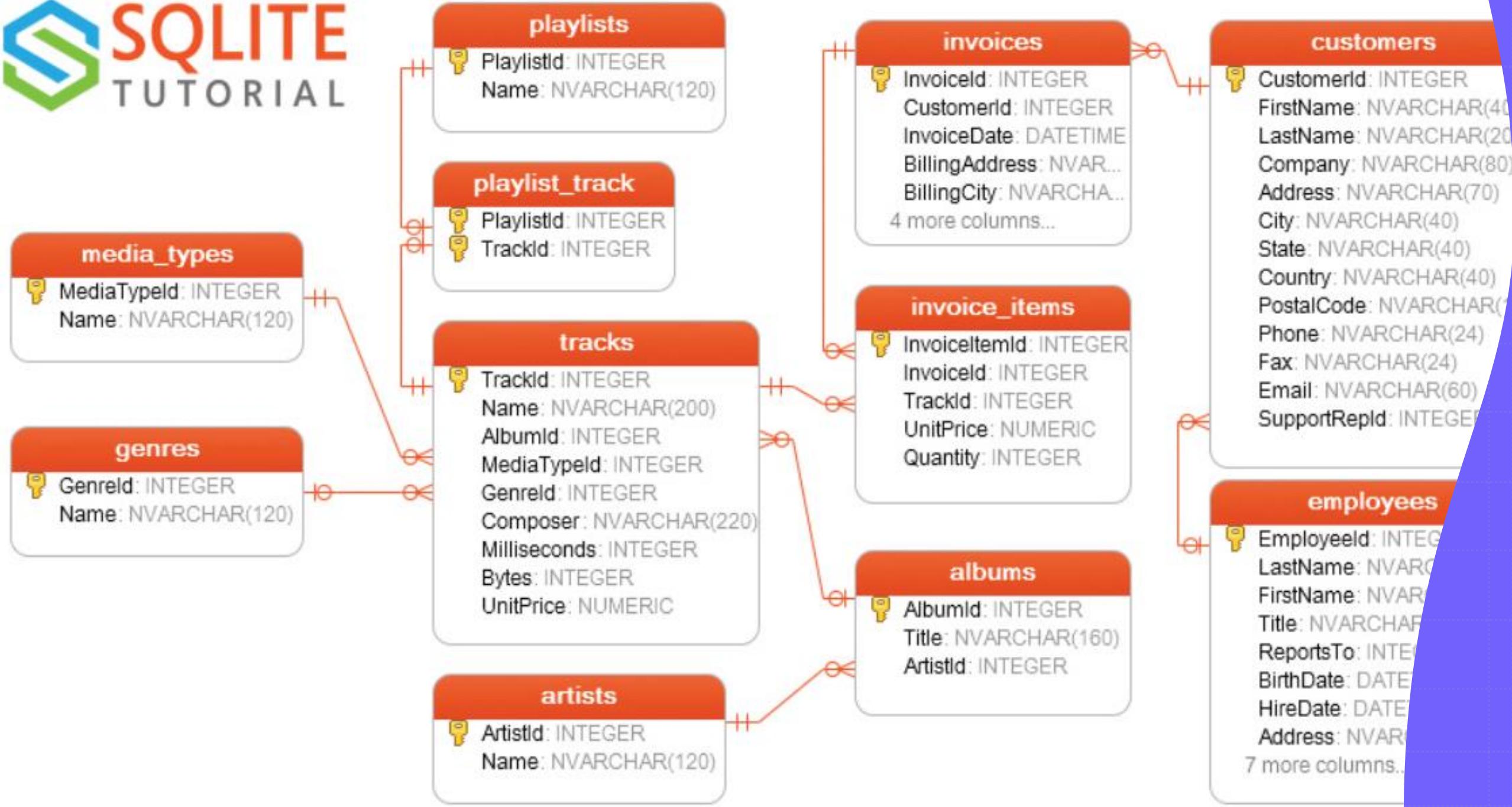
ADVERTISEMENTS

Ad closed by Google

SQLITE TUTORIAL

SQLite Select  
SQLite Order By  
SQLite Select Distinct  
SQLite Where  
SQLite Limit

# SQLITE TUTORIAL



# Klienci ogólnego przeznaczenia



DBeaver 22.0.0 - football\_game

File Edit Navigate Search SQL Editor Database Window Help

SQL Commit Rollback Auto census.sqlite 4 < N/A >

Database Navig... X Projects

Enter a part of object name here

census.sqlite  
census.sqlite 2  
census.sqlite 3  
census.sqlite 4  
census.sqlite 5  
census.sqlite 6  
coderslab\_vm - localhost:5432  
Cookies  
Cookies 2  
Cookies 3  
cookies.sqlite  
cookies.sqlite 2  
db.sqlite3  
db.sqlite3 2  
db.sqlite3 3  
db.sqlite3.txt  
exercises - localhost:5432  
postgres - localhost:5432

Properties Data ER Diagram

Grid Text

football\_game | Enter a SQL expression to filter results (use Ctrl+Space)

123 id 123 team\_home\_goals 123 team\_away\_goals 123 team\_away\_id 123 team\_home\_id

Project - General X

Name Data Source

Bookmarks ER Diagrams

The screenshot shows the DBeaver interface with the following details:

- Toolbar:** File, Edit, Navigate, Search, SQL Editor, Database, Window, Help.
- Top Status Bar:** SQL, Commit, Rollback, Auto, census.sqlite 4, < N/A >, search icon.
- Left Sidebar:** Database Navigator (closed), Projects, a search bar ("Enter a part of object name here"), and a list of databases and connections.
- Central Area:** A table named "football\_game" is displayed in a "Grid" view. The table has five columns: "id", "team\_home\_goals", "team\_away\_goals", "team\_away\_id", and "team\_home\_id". Each column has a numeric type indicator (123) and a sorting arrow icon.
- Bottom Left:** Project - General tab, Data Source configuration panel with icons for settings, minus, plus, and refresh.
- Bottom Navigation:** Name, Data Source, Bookmarks, and ER Diagrams.

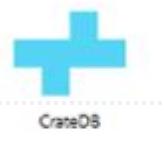
# DBeaver

## Select your database

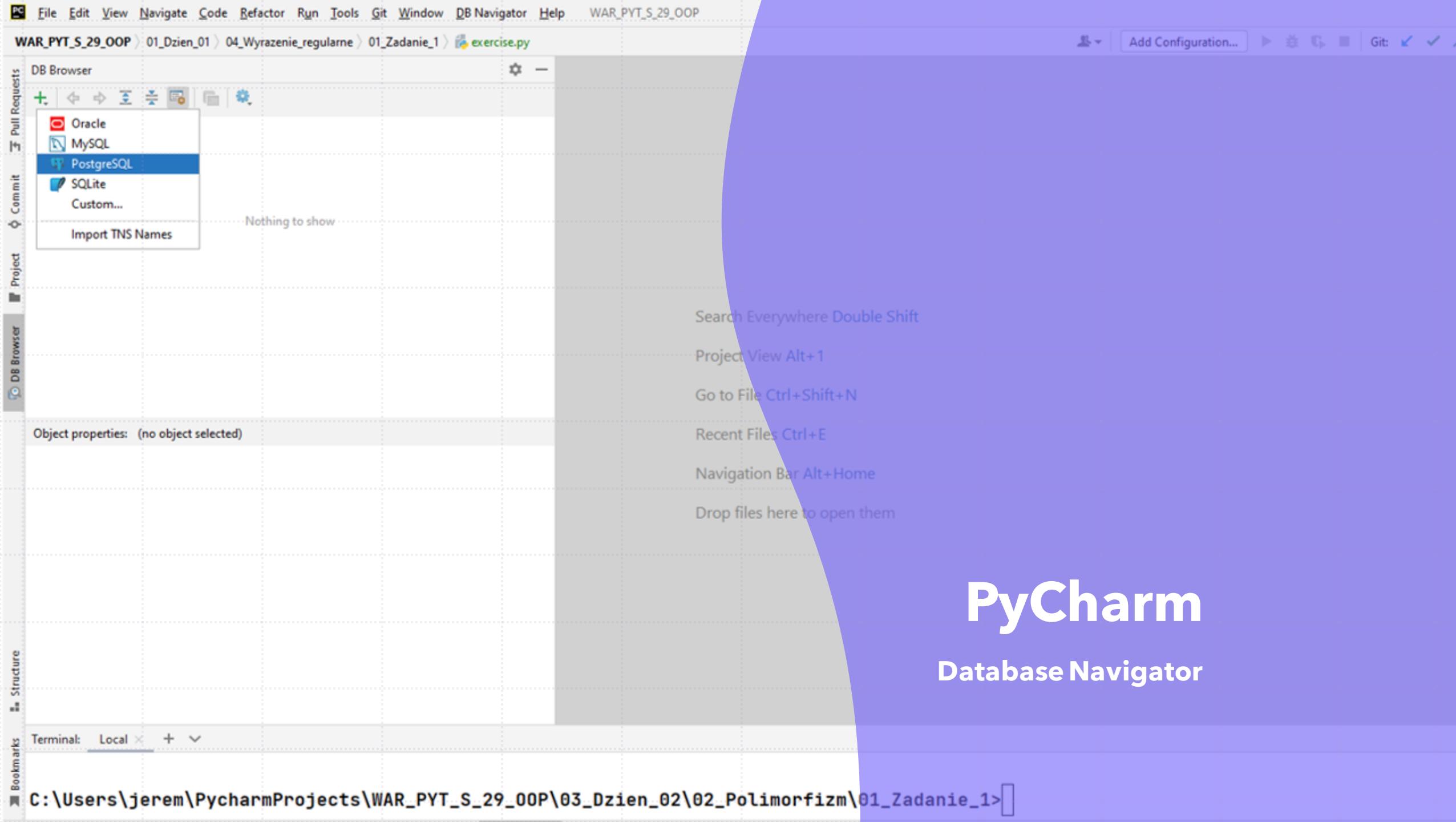
Create new database connection. Find your database driver in the list below.

Type part of database/driver name to filter

Sort by:  Title  Score

All	Popular	SQL	NoSQL	Analytical	Timeseries	Embedded	Hadoop / BigData	Full-text search	Graph databases		
 SQLite	 PostgreSQL	 IBM DB2 LUW	 DuckDB	 MariaDB	 MySQL	 ODBC	 ORACLE	 SQL Server	 Apache Calcite Avatica	 Apache Drill	 Apache Hive
 Apache Ignite	 Apache Kylin	 Apache Phoenix	 Apache Spark	 Athena	 Azure SQL Server	 Babelfish via TDS (beta)	 Cache	 ClickHouse	 ClickHouse (Legacy)	 Cloudera Impala	 CockroachDB
 CrateDB	 CSV	 CUBRID	 Data Virtuality	 IBM DB2 Series/AS 400	 IBM DB2 LUW (Old 8x)	 IBM DB2 z/OS	 DBF	 Denodo 8	 Derby Embedded	 Derby Server	 Dremio
 Elasticsearch	 EnterpriseDB POSTGRES	 Exasol	 Firebird	 Gemfire XD	 Google BigQuery	 Google Cloud Spanner	 Google Cloud Spanner (Deprecated)	 Google Cloud SQL - PostgreSQL	 Greenplum	 H2 Embedded	 H2 Embedded V.2
 H2 Server	 H2GIS Embedded	 H2GIS Server	 SAP HANA	 HSQL Embedded	 HSQL Server	 Informix	 Ingres	 InterSystems IRIS	 Kognito	 Machbase	
 SAP	 MongoDB	 Oracle Database	 Apache Derby	 Apache Flink	 Apache Ignite	 Apache Drill	 Apache Calcite Avatica	 Apache Drill	 Apache Drill	 Apache Drill	

DBeaver



# PyCharm

## Database Navigator

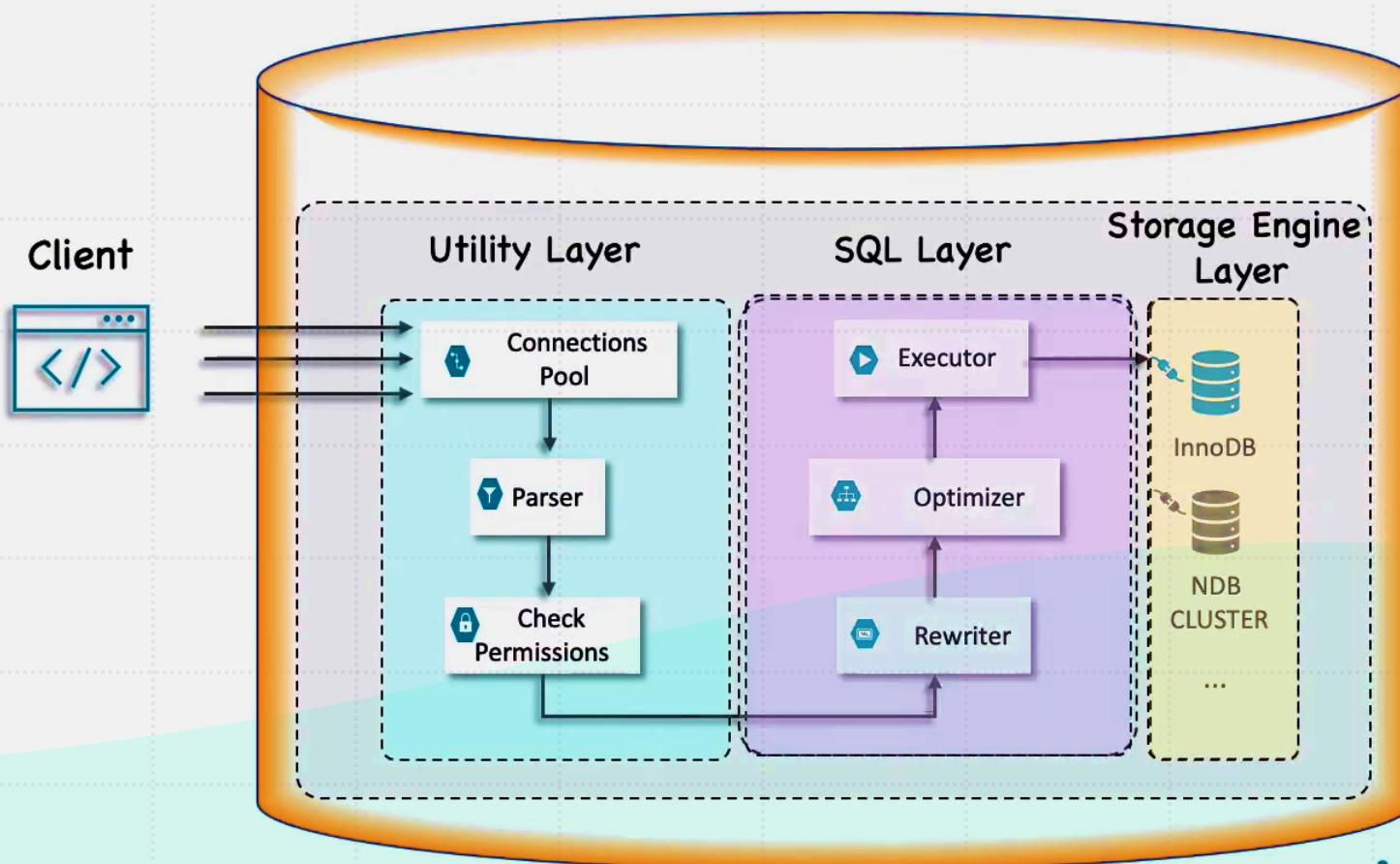
The screenshot shows the PyCharm Database Navigator interface. On the left, the Project and DB Browser toolbars are visible. The main area displays a connection to a PostgreSQL database named 'main'. The 'Connection' tab is selected, showing the query `select * from django_migrations;`. The results pane shows the contents of the `django_migrations` table:

	id	app	name	applied
1	1	contenttypes	0001_initial	23 cze 2022, 17:00:35
2	2	auth	0001_initial	23 cze 2022, 17:00:35
3	3	admin	0001_initial	23 cze 2022, 17:00:35
4	4	admin	0002_logentry_remove_auto_add	23 cze 2022, 17:00:35
5	5	admin	0003_logentry_add_action_flag_choices	23 cze 2022, 17:00:35
6	6	contenttypes	0002_remove_content_type_name	23 cze 2022, 17:00:35
7	7	auth	0002_alter_permission_name_max_length	23 cze 2022, 17:00:35
8	8	auth	0003_alter_user_email_max_length	23 cze 2022, 17:00:35
9	9	auth	0004_alter_user_username_opts	23 cze 2022, 17:00:35
10	10	auth	0005_alter_user_last_login_null	23 cze 2022, 17:00:35

Below the table, a message states: "Connection (main): 19 records - executed in 9 ms. / fetched in 1 ms."

**PyCharm**  
Database Navigator

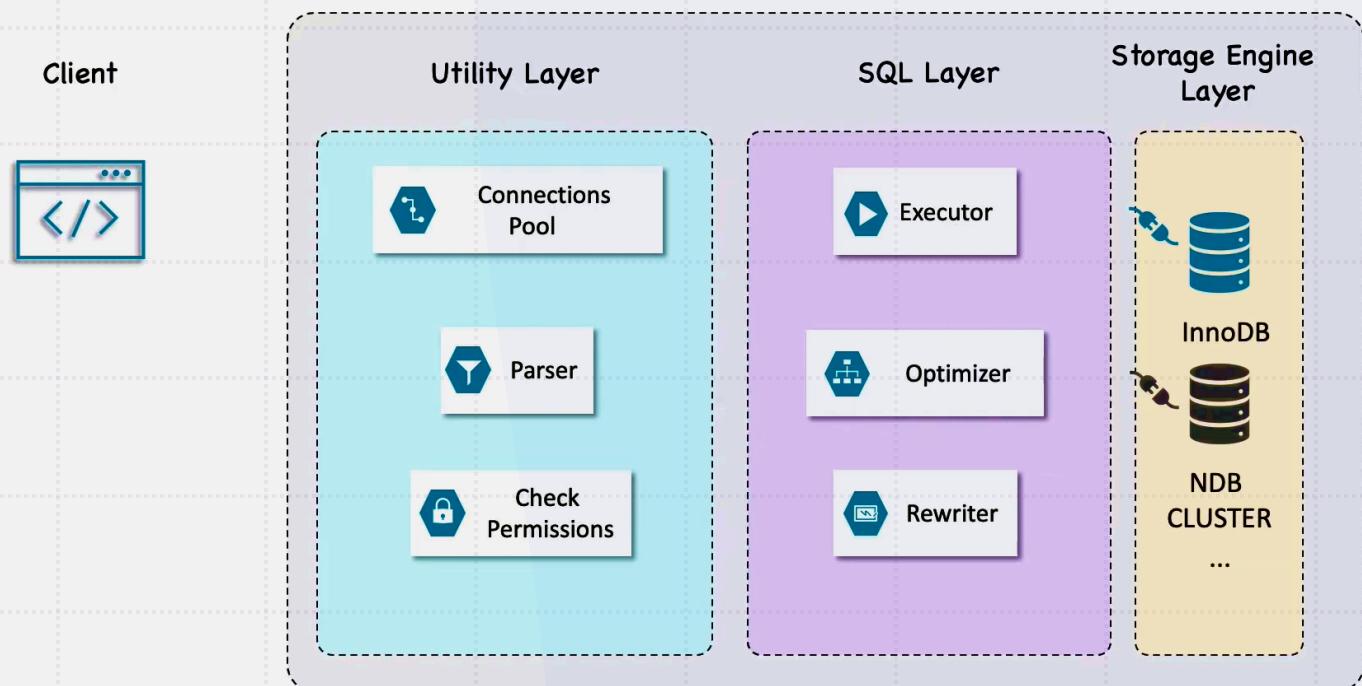
# Architektura serwera MySQL



# Architektura serwera MySQL

W architekturze serwera MySQL można wydzielić trzy podstawowe warstwy:

1. Utility Layer
2. SQL Layer
3. Storage Engine Layer



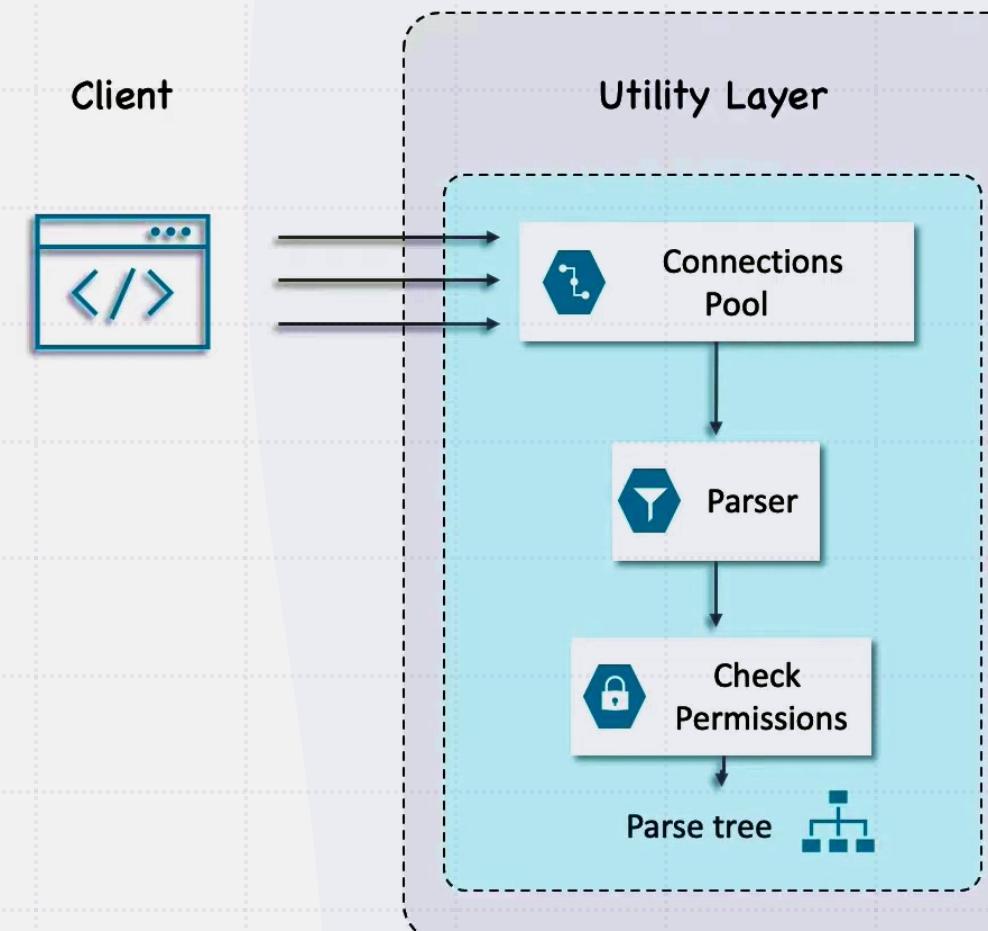
# 1. Utility Layer

Najwyższa warstwa serwera MySQL składa się z usług typowych dla większości narzędzi klient-serwer bazujących na sieci, takich jak obsługa połączenia oraz wstępne przetwarzanie zapytania, w tym: parsowanie oraz autoryzacji zapytania. Najważniejszymi elementami warstwy są analizator składni (ang. parser) oraz preprocesor.

Analizator weryfikuje poprawność składniową zapytania oraz dzieli je na tokeny. Na podstawie wyodrębnionych tokenów buduje "drzewo analizy składniowej" aka "drzewo przetwarzania" (ang. parse tree).

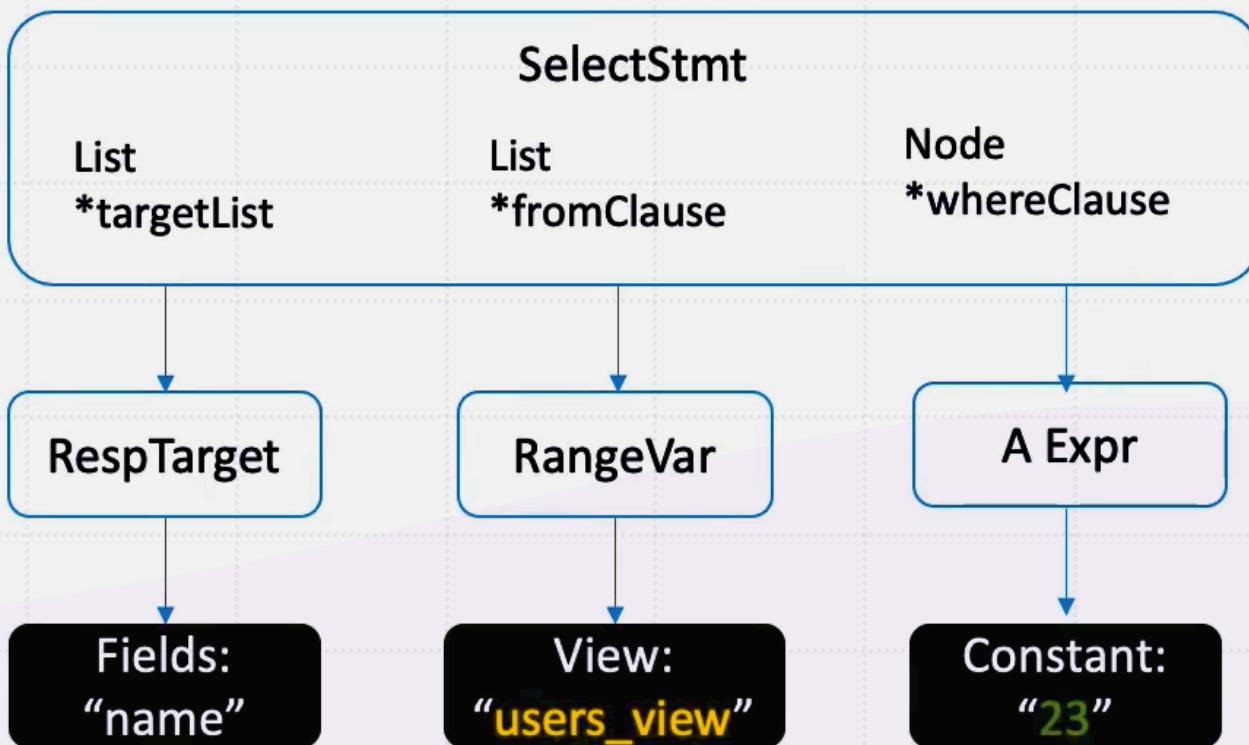
Zbudowane drzewo przekazywane jest do preprocesora. Preprocesor weryfikuje zgodność drzewa z istniejącymi danymi oraz sprawdza uprawnienia.

Jeżeli wszystkie etapy wstępnego przetwarzania zakończyły się pomyślnie, drzewo analizy jest przekazywane do kolejnego warstwy.



# Przykładowe drzewo analizy składniowej w MySQL serwer

```
SELECT name FROM users_view WHERE id = 23;
```



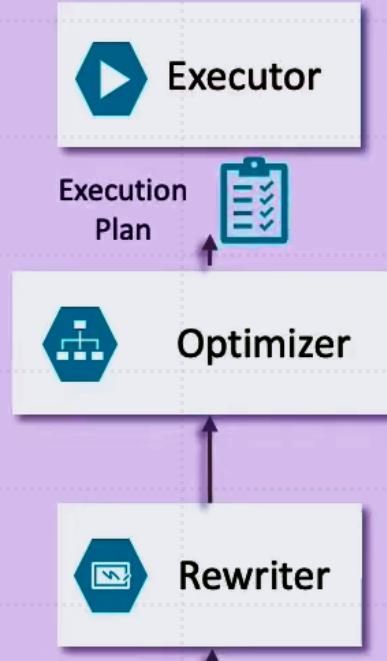
## 2. SQL Layer

Środkowa warstwa serwera MySQL jest odpowiedzialna za opracowanie planu wykonywania zapytania (ang. *execution plan*) na podstawie otrzymanego drzewa analizy składniowej. Podstawowymi elementami warstwy są: optymalizator zapytań (ang. *optimizer*) oraz silnik wykonywania zapytań (ang. *executor*).

Optymalizator zapytania buduje plan wykonywania na podstawie drzewa analizy składniowej, a następnie przekazuje stworzony plan do silnika wykonywania zapytań.

Silnik wykonywania zapytań wywołuje odpowiednią metodą API silnika magazynu danych (ostatnia warstwa serwera MySQL)

### SQL Layer



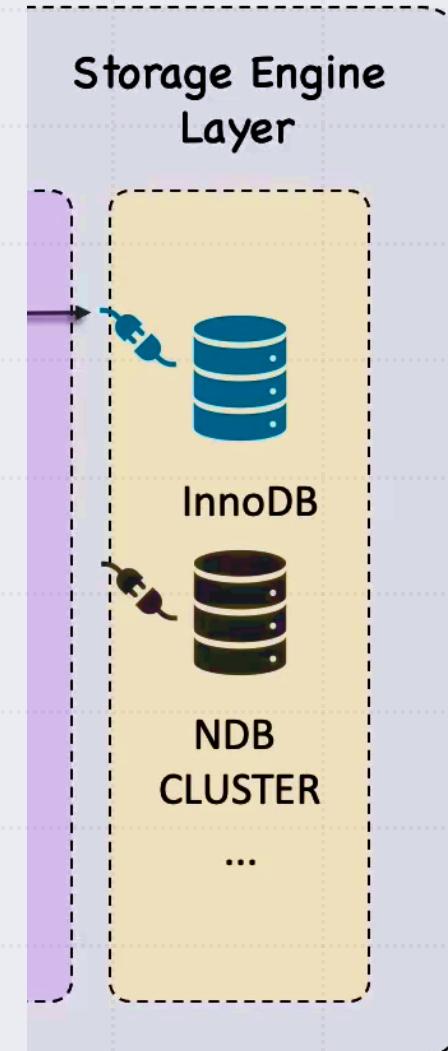
### 3. Storage Engine Layer

Najniższa warstwa serwera MySQL - silnik magazynu danych odpowiedzialny jest za przechowywanie danych, wykonywanie operacji na tych danych oraz zwracanie wyniku. Jest zaimplementowana w tzw. architekturze pluginowej, dlatego stosunkowo łatwo jest przełączyć się pomiędzy różnymi silnikami magazynu danych.

Domyślnym silnikiem magazynu danych w MySQL jest **InnoDB**. InnoDB w pełni wspiera transakcyjność i osiąga dobre wyniki w warunkach silnego obciążenia równoległymi operacjami (ale na tym polu prym wiedzie Oracle i PostgreSQL).

Inne silniki magazynu danych to m.in.:

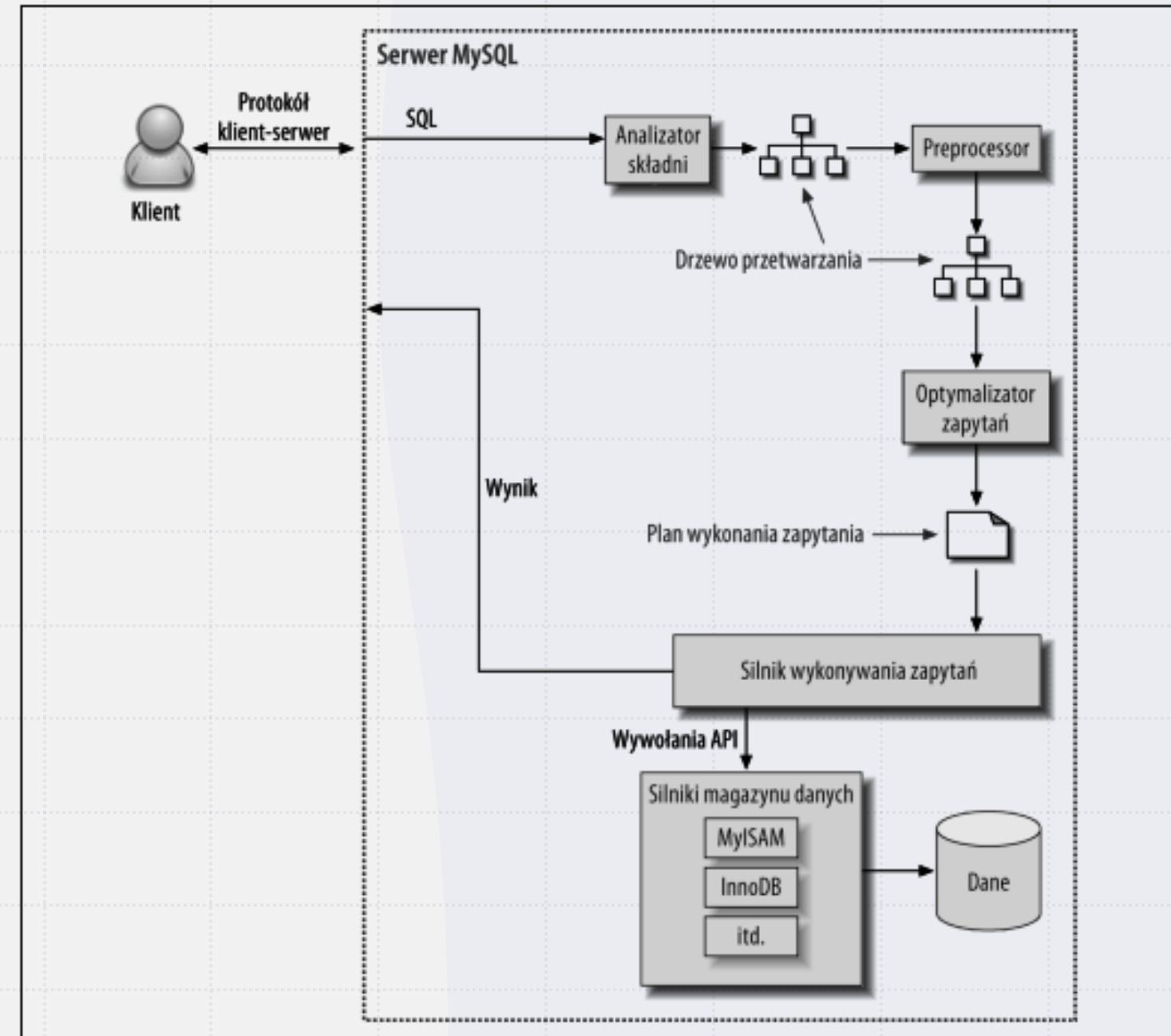
- NDB Cluster
- MyISAM Storage Engine
- MEMORY Storage Engine
- CSV Storage Engine
- ARCHIVE Storage Engine
- BLACKHOLE Storage Engine
- MERGE Storage Engine
- FEDERATED Storage Engine
- EXAMPLE Storage Engine



Na koniec otrzymany wynik zwracany jest do klienta w postaci **result set-a**.

# Architektura serwera MySQL

Diagram przedstawiający pełny proces przetwarzania zapytania przez serwer MySQL.



# Błędy

# SQLSTATE

Zgodnie ze standardem ANSI SQL każdy błąd posiada kod wyjścia nazywany SQLSTATE. Dwa pierwsze znaki SQLSTATE oznaczają klasę błędu, trzy pozostałe oznaczają numer błędu danej klasy. Przykłady SQLSTATE:

SQLSTATE	Numer klasy	Opis klasy	Numer błędu	Opis błędu
28000	28	Błąd uwierzytelnienia	000	Nieprawidłowa nazwa użytkownika lub hasło
22004	22	Błąd danych	004	Niedozwolona wartość null
22007	22	Błąd danych	007	Nieprawidłowy format daty

Standard ANSI SQL wyróżnia 60 klas i daje swobodę poszczególnym RDBMS do robudowania hierarchii własne klasy. Więcej o SQLSTATE można przeczytać tutaj <https://en.wikipedia.org/wiki/SOLSTATE>

## Kody błędów (MySQL)

Niezależnie od SQLSTATE wiele RDBMS definiuje swoje kody błędów (ang. error code). Przykłady kodu błędów dla MySQL:

SQLSTATE	MySQL error code	MySQL Symbol
28000	1045	ER_ACCESS_DENIED_ERROR
22004	1138	ER_INVALID_USE_OF_NULL
22007	1292	ER_TRUNCATED_WRONG_VALUE

Więcej o kodach błędów MySQL można przeczytać tutaj <https://dev.mysql.com/doc/mysql-errors/8.0/en/server-error-reference.html>

# Transakcyjność

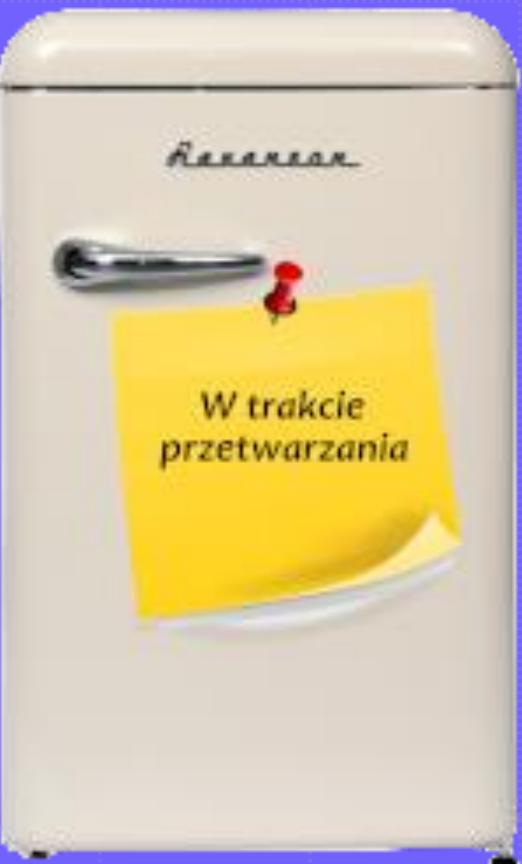
All or nothing











W trakcie  
przetwarzania



# Lock











START TRANSACTION;  
UPDATE konto SET saldo = saldo - 100 WHERE konto\_id = 1;  
UPDATE konto SET saldo = saldo + 100 WHERE konto\_id = 2;  
COMMIT;



START TRANSACTION;

UPDATE konto SET saldo = saldo - 100 WHERE konto\_id = 1;

UPDATE konto SET saldo = saldo + 100 WHERE konto\_id = 2;

COMMIT;

Korzyści z użycia transakcji:

1. Miejsce wprowadzania zmian jest kontrolowane, zmiany są aplikowane w momencie wywołania komendy COMMIT, a nie natychmiast.
2. Rozpoczęcie transakcji zakłada locka na modyfikowane zasoby.
3. Transakcja grupuje operacje, które potem, w razie wystąpienia błędu można wycofać jedną komendą.

**UWAGA!** Zmiany nie są wycofywane automatycznie w przypadku wystąpienia błędu w trakcie wykonywania transakcji. Taki przypadek trzeba samodzielnie obsłużyć. Służy do tego instrukcja ROLLBACK.

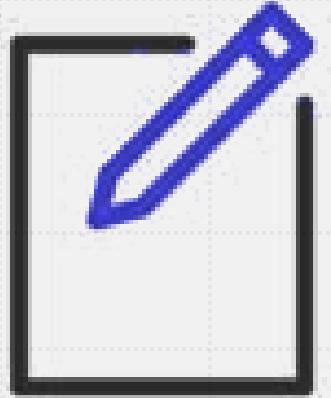


## Wycofywanie zmian (rollback)

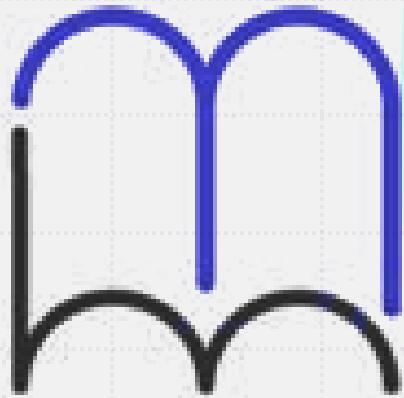
```
-- Tworzenie procedury przenoszenia środków pomiędzy kontami
CREATE PROCEDURE make_transfer()
BEGIN
    -- Obsługa wyjątków
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;    -- cofnięcie wszelkich zmian dokonanych w trakcie transakcji
    END;

    -- Rozpoczęcie transakcji
    START TRANSACTION;
    -- Aktualizacja salda kont
    UPDATE konto SET saldo = saldo - 100 WHERE konto_id = 1;
    UPDATE konto SET saldo = saldo + 100 WHERE konto_id = 2;
    COMMIT;
END// 
DELIMITER ; 

-- Wywołanie procedury make_transfer
CALL make_transfer;
```



CREATE



READ



UPDATE



DELETE

---

C

R

U

D

# Programowanie imperatywne vs programowanie deklaratywne

```
# Znajdź największy element na liście arr  
arr = [10, -2, 334, -45, 50, 35]
```

```
# Styl imperatywny (nakaz jak zrobić)  
max_value = arr[0]  
idx = 1  
while idx < len(arr):  
    if arr[idx] > max_value:  
        max_value = arr[idx]  
    idx += 1  
  
print(max_value)
```

Jak?

```
# Znajdź największy element na liście arr  
arr = [10, -2, 334, -45, 50, 35]
```

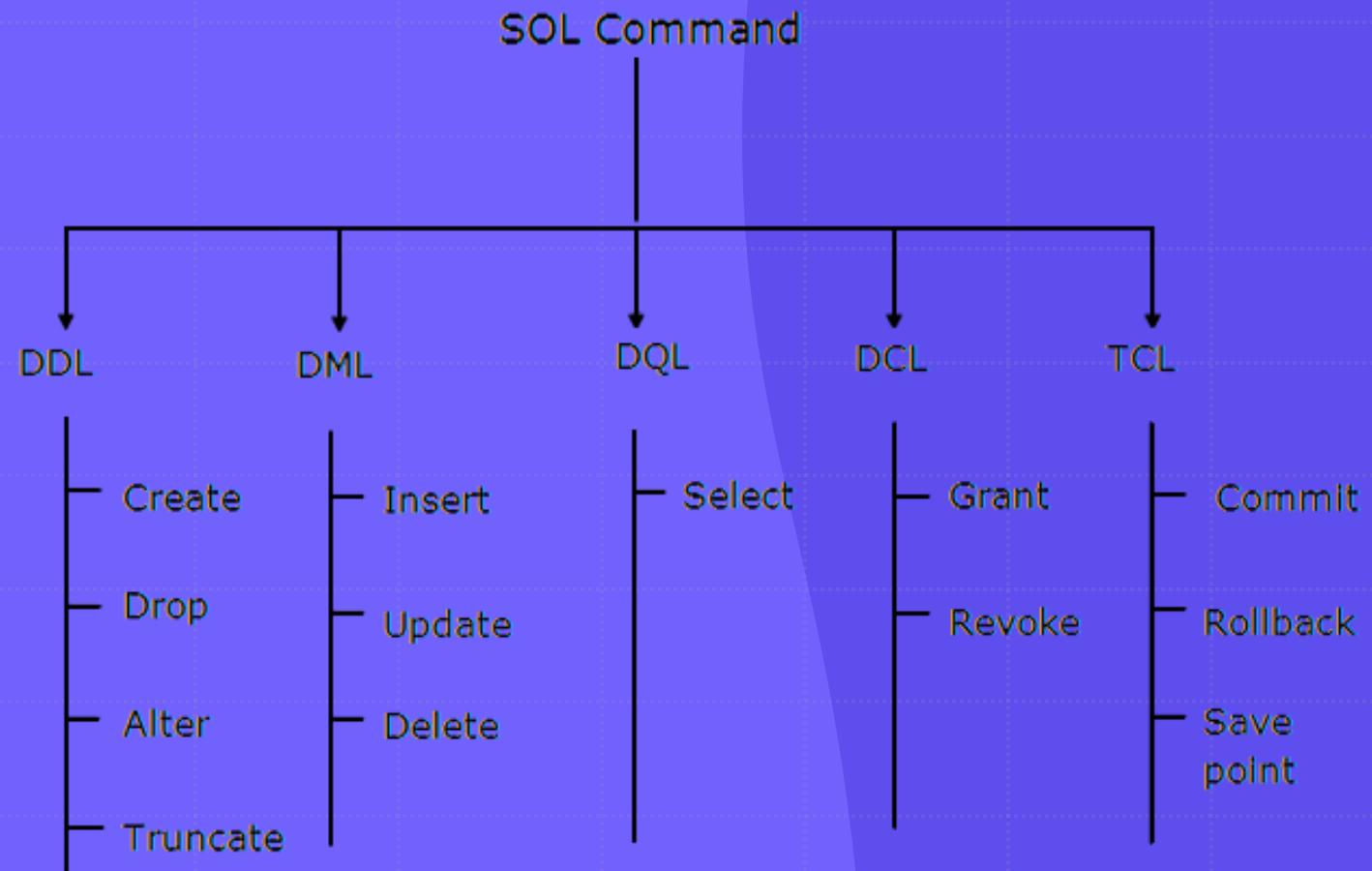
```
# Styl deklaratywny (deklaracja co zrobić)  
max_value = max(arr)  
  
print(max_value)
```

Co?

# Deklaratywny SQL

# Deklaratywny SQL

1. **DDL** - Data Definition Language
2. **DML** - Data Manipulation Language
3. **DQL** - Data Query Language
4. **DCL** - Data Control Language
5. **TCL** - Transaction Control Language



# Imperatywny SQL

# Imperatywny SQL

Standard SQL poza podstawowymi strukturami deklaratywnymi takimi jak: SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ... definiuje również dobrze nam znane z Python-a struktury imperatywne:

1. zmienne
2. warunki
3. pętle

# 1. Zmienna

```
SET @moja_zmienna = 10;
```

## 2. Warunki

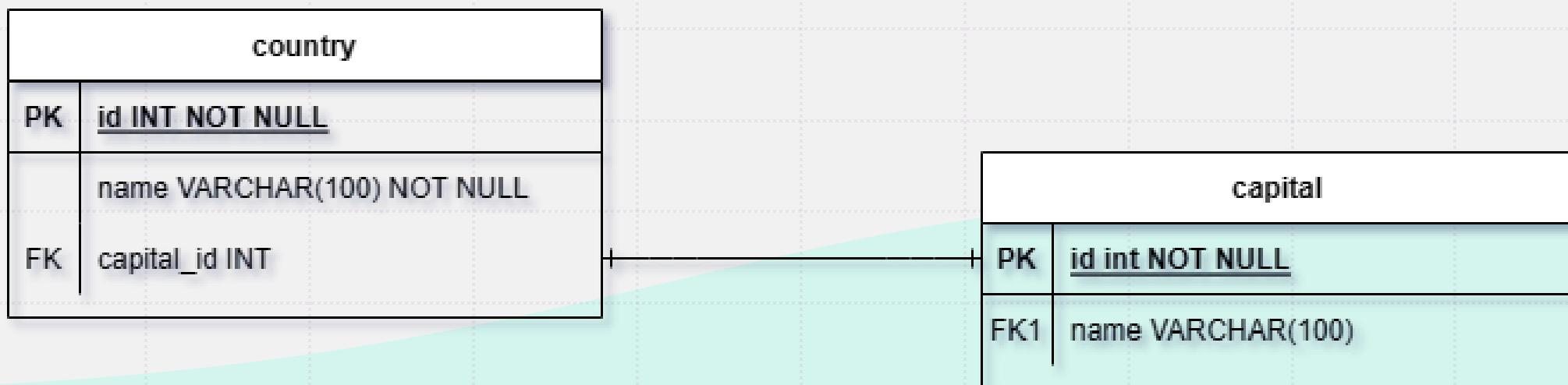
```
SET @moja_zmienna = 10;  
IF @moja_zmienna > 5 THEN  
    SELECT 'Większe od 5';  
ELSE  
    SELECT 'Mniejsze lub równe 5';  
END IF;
```

## 3. Pętle

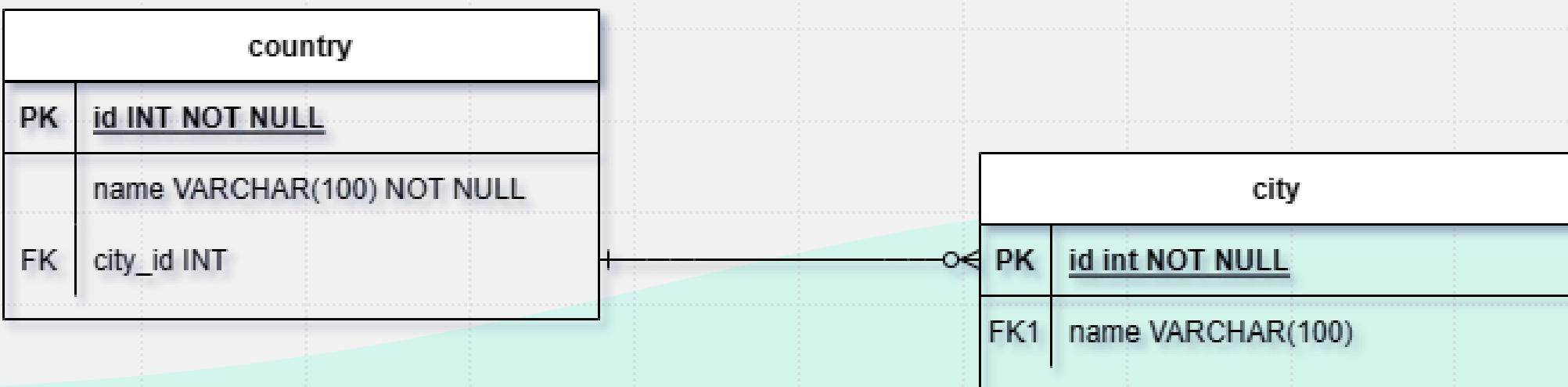
```
SET @licznik = 0;  
WHILE @licznik < 5 DO  
    SELECT @licznik;  
    SET @licznik = @licznik + 1;  
END WHILE;
```

# Powiązania

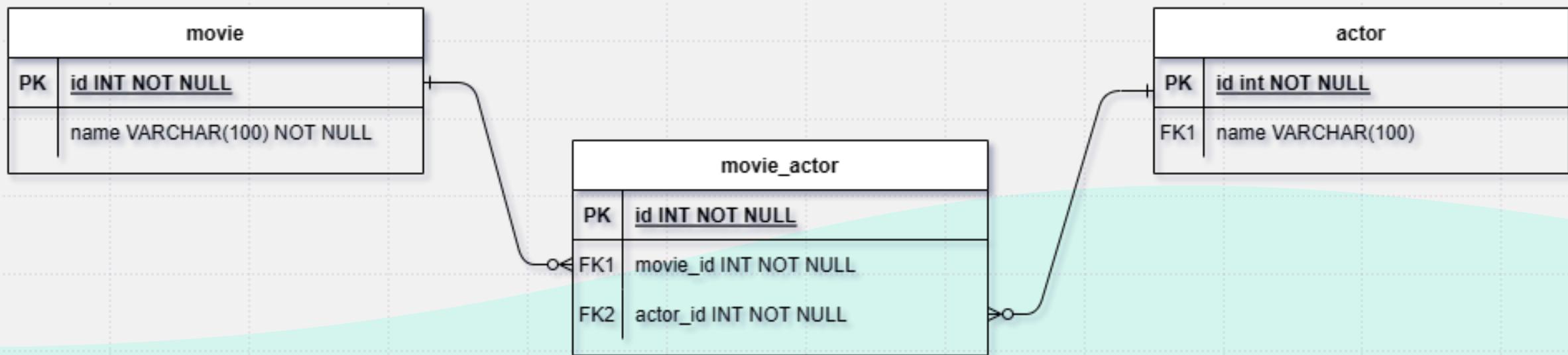
# Jeden do jeden



# Jeden do wielu



# Wiele do wielu



# SQL Injection

# SQL Injection

**Username:**

**Password:**

# SQL Injection

```
SELECT * FROM users  
WHERE username = username AND password = password;
```

# SQL Injection

**Username:**

harry

**Password:**

12345

# SQL Injection

```
SELECT * FROM users  
WHERE username = username AND password = password;
```

# SQL Injection

```
SELECT * FROM users  
WHERE username = "harry" AND password = "12345";
```

# SQL Injection

**Username:**

hacker" --

**Password:**

# SQL Injection

```
SELECT * FROM users  
WHERE username = username AND password = password;
```

# SQL Injection

```
SELECT * FROM users  
WHERE username = "hacker"--" AND password = "";
```

# SQL Injection

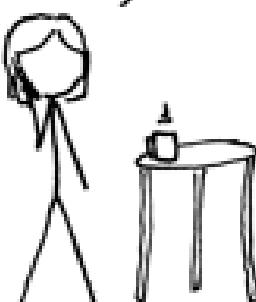
```
SELECT * FROM users  
WHERE username = "hacker"--" AND password = "";
```

# SQL Injection

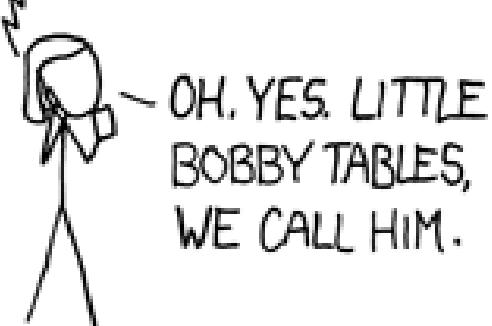
HI, THIS IS  
YOUR SON'S SCHOOL.  
WE'RE HAVING SOME  
COMPUTER TROUBLE.



OH, DEAR – DID HE  
BREAK SOMETHING?  
IN A WAY –



DID YOU REALLY  
NAME YOUR SON  
Robert'); DROP  
TABLE Students;-- ?



OH, YES. LITTLE  
BOBBY TABLES,  
WE CALL HIM.

WELL, WE'VE LOST THIS  
YEAR'S STUDENT RECORDS.  
I HOPE YOU'RE HAPPY.



AND I HOPE  
YOU'VE LEARNED  
TO SANITIZE YOUR  
DATABASE INPUTS.

# SQL Injection



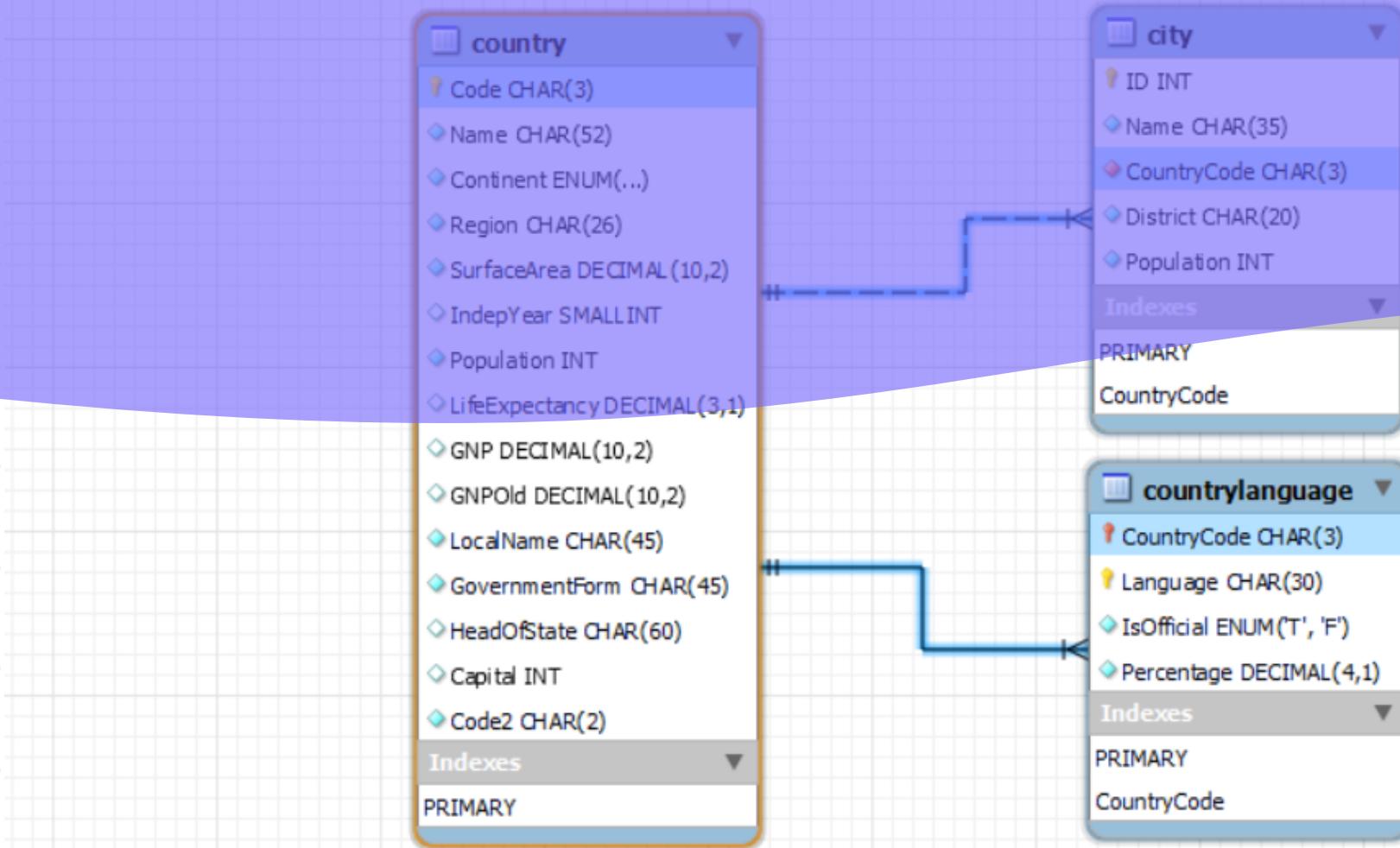
# Popularne bazy danych

**MySQL**

# Popularne bazy danych MySQL

Title	DB Download	HTML Setup Guide	PDF Setup Guide
employee data (large dataset, includes data and test/verification suite)	<a href="#">GitHub</a>	<a href="#">View</a>	<a href="#">US Ltr   A4</a>
world database	<a href="#">TGZ   Zip</a>	<a href="#">View</a>	<a href="#">US Ltr   A4</a>
sakila database	<a href="#">TGZ   Zip</a>	<a href="#">View</a>	<a href="#">US Ltr   A4</a>
airportdb database (large dataset, intended for MySQL on OCI and HeatWave)	<a href="#">TGZ   Zip</a>	<a href="#">View</a>	<a href="#">US Ltr   A4</a>
menagerie database	<a href="#">TGZ   Zip</a>		





# Word sample database



Contact MySQL | Login | Register

# MySQL

mysol.com

## DOWNLOADS

DOCUMENTATION

DEVELOPER ZONE



# Sakila sample database

 Documentation Home

## Sakila Sample Database

- Preface and Legal Notices
  - Introduction
  - **History**
  - Installation
  - Structure
  - Usage Examples
  - Known Issues
  - Acknowledgments
  - License for the Sakila Sample Database
  - Note for Authors
  - Sakila Change History

Sakila Sample Database / History

## 3 History

The Sakila sample database was designed as a replacement to the [world sample database](#), also provided by Oracle.

The world sample database provides a set of tables containing information on the countries and cities of the world and is useful for basic queries, but lacks structures for testing MySQL-specific functionality and features found in MySQL 5 and higher.

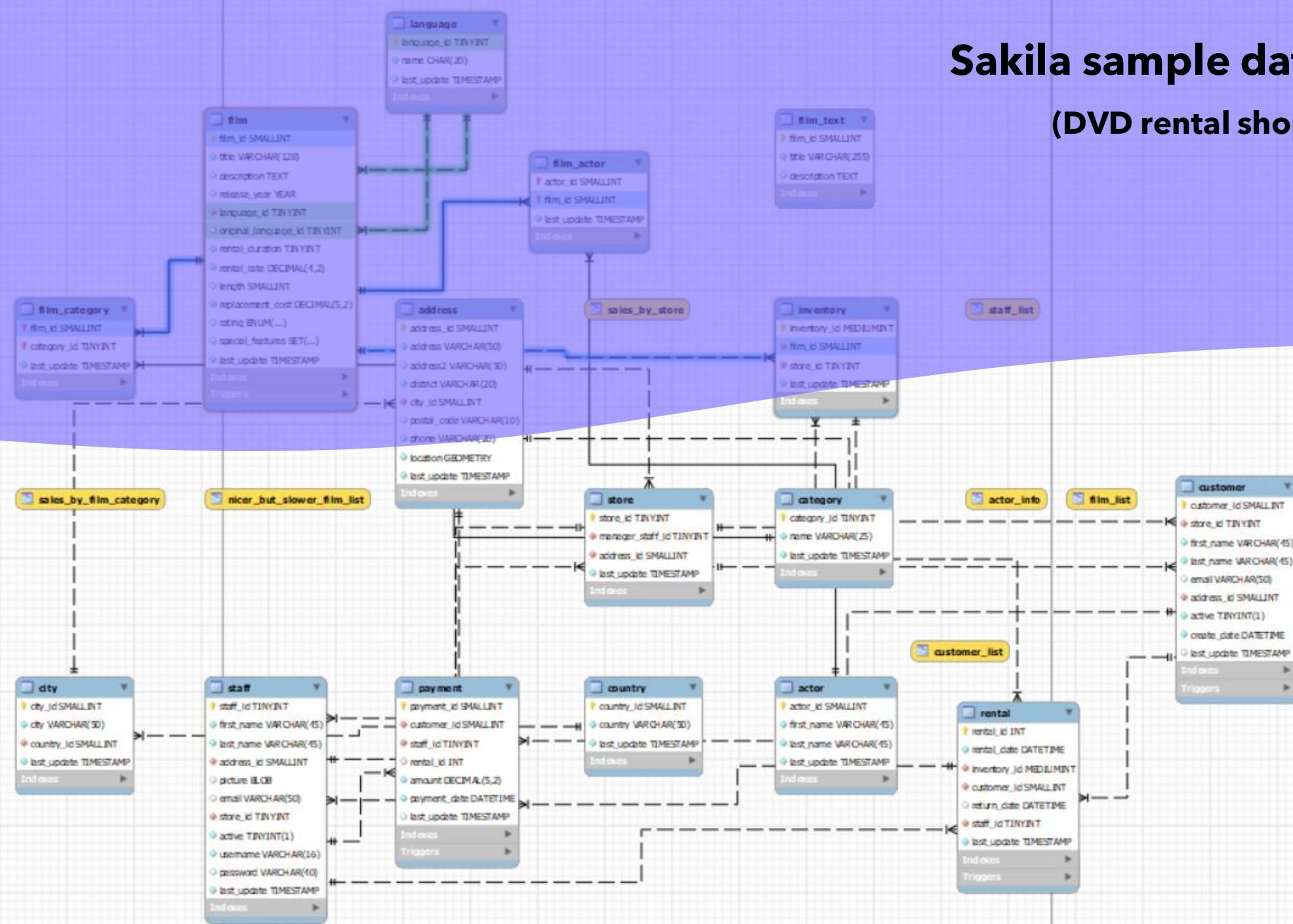
Development of the Sakila sample database began in early 2005. Early designs were based on the database used in the Dell whitepaper [Three Approaches to MySQL Applications on Dell PowerEdge Servers](#).

Where Dell's sample database was designed to represent an online DVD store, the Sakila sample database is designed to represent a DVD rental store. The Sakila sample database still borrows film and actor names from the Dell sample database.

Development was accomplished using MySQL Query Browser for schema design, with the tables being populated by a combination of MySQL Query Browser and custom scripts, in addition to contributor efforts (see [Section 8, "Acknowledgments"](#)).

After the basic schema was completed, various views, stored routines, and triggers were added to the schema; then the sample data was populated. After a series of review versions, the first official version of the Sakila sample database was released in March 2006.

# Sakila sample database (DVD rental shop)



# Materiały do nauki SQL



 DataWars: Practice Data Science/Analysis with +100 Real Life Projects



# SQL Tutorial

## Tutorials: Learn SQL step by step

Language: English • 日本語 • 中文

- 1 SELECT basics**  
Some simple queries to get you started
  - 2 SELECT name**  
Some pattern matching queries
  - 3 SELECT from World**  
In which we query the World country profile table.
  - 4 SELECT from Nobel**  
Additional practice of the basic features using a table of Nobel Prize winners.
  - 5 SELECT within SELECT**  
In which we form queries using other queries.
  - 6 SUM and COUNT**  
In which we apply aggregate functions. [more the same](#)
  - 7 JOIN**  
In which we join two tables; game and goals. [previously music tutorial](#)
  - 8 More JOIN operations**  
In which we join actors to movies in the Movie Database.
  - 9 Using Null**  
In which we look at teachers in departments. [previously Scottish Parliament](#)
  - 8+ Numeric Examples**  
In which we look at a survey and deal with some more complex calculations.
  - 9- Window function**  
In which we examine UK general election results.
  - 9+ COVID 19**  
In which we measure the impact of COVID-19
  - 9 Self join**

# SQLZoo

<https://sqlzoo.net/>

Zaloguj Zamknij

Learn to Code - for Free | Codecademy

https://www.codecademy.com

codecademy Catalog ▾ Resources ▾ Community ▾ Pricing ▾ Career Center New For Teams Log In

# THE NEXT GENERATION OF LEARNING

- + Build skills quicker with AI assistance
- + Gain experience with modern developer tools

Join the millions learning to code with Codecademy for free

Email

Password  ⓘ

Sign up

By signing up for Codecademy, you agree to Codecademy's [Terms of Service](#) & [Privacy Policy](#).

Or sign up using:

<https://www.codecademy.com/>

**SQLBolt**

Learn SQL with simple, interactive exercises.



Interactive Tutorial



More Topics

## Introduction to SQL

Welcome to SQLBolt, a series of interactive lessons and exercises designed to help you quickly learn SQL right in your browser.

### What is SQL?

SQL, or Structured Query Language, is a language designed to allow both technical and non-technical users query, manipulate, and transform data from a relational database. And due to its simplicity, SQL databases provide safe and scalable storage for millions of websites and mobile applications.

#### Did you know?

There are many popular SQL databases including SQLite, MySQL, Postgres, Oracle and Microsoft SQL Server. All of them support the common SQL language standard, which is what this site will be teaching, but each implementation can differ in the additional features and storage types it supports.

### Relational databases

Before learning the SQL syntax, it's important to have a model for what a relational database actually is. A relational database represents a collection of related (two-dimensional) tables. Each of the tables are similar to an Excel spreadsheet, with a fixed number of named columns (the attributes or properties of the table) and any number of rows of data.

For example, if the Department of Motor Vehicles had a database, you might find a table containing all the known vehicles that people in the state are driving. This table might need to store the model name, type, number of wheels, and number of doors of each vehicle for example.

Table: Vehicles

# SQLBold

<https://sqlbolt.com/>

