

SQL w PostgreSQL

poziom podstawowy

Marcin Gwiżdż

15-16 października 2024r.

Podstawy pracy w środowisku bazodanowym

Bazy danych

Baza danych – w obecnie rozwijanych technologiach cyfrowych stanowi podstawę funkcjonowania systemów informatycznych.

Baza zawiera skonfigurowane przez użytkownika/ów kontenery zapisu informacji. Niezależnie od ich postaci, kontenery te mają za zadanie przechowywać informacje w sposób uporządkowany...

Uporządkowany = zrozumiały dla twórcy, użytkownika i każdego systemu, który korzysta z danej bazy.

Bazy danych - definicja

Techniczna:

Dane cyfrowe gromadzone zgodnie z zasadami przyjętymi dla danego programu komputerowego specjalizowanego do gromadzenia i przetwarzania tych danych (Wiki)

Prawna:

Zbiór danych lub jakichkolwiek innych materiałów i elementów zgromadzonych według określonej systematyki lub metody, indywidualnie dostępnych w jakikolwiek sposób, w tym środkami elektronicznymi, wymagający istotnego, co do jakości lub ilości, nakładu inwestycyjnego w celu sporządzenia, weryfikacji lub prezentacji jego zawartości (Ustawa o ochronie baz danych z 27.07.2001)

Największe istniejące bazy danych

Jaka jest największa baza danych na świecie...?

Jaka jest jej objętość...?

Jak szybko się rozwija...?

Czy można odpowiedzieć na te pytania?

The Google logo, consisting of the word "Google" in its characteristic multi-colored font (blue, red, yellow, blue, green, red).The Amazon logo, featuring the word "amazon" in a lowercase, black, sans-serif font with a curved orange arrow underneath it.The logo of the Central Statistical Office (Główny Urząd Statystyczny), featuring two overlapping circles: a light green one on top and a dark purple one on the bottom.

Główny
Urząd Statystyczny

Bazy danych – system zarządzania

Oprogramowanie bądź system informatyczny służący do zarządzania bazą danych. System zarządzania bazą danych może być również serwerem bazy danych (SBD) lub też może udostępniać bazę danych lokalnie – na określonym komputerze.

Zadania systemu zarządzania bazą danych:

- administrowanie danymi,
- zapewnienie integralności danych,
- narzędzia dostępu i przeszukiwania danych,
- narzędzia usprawniające wyszukiwanie danych (np. indeksy),
- narzędzia autoryzacji użytkowników,
- narzędzia umożliwiające równoczesny dostęp wielu użytkowników

Bazy danych - relacyjna



Sfinansowano ze środków
NARODOWEGO FUNDUSZU
OCHRONY ŚRODOWISKA
i GOSPODARKI WODNEJ

Baza przechowująca dane w formie tabel o ustalonej strukturze:

- Zbiór atrybutów dla pojedynczej tabeli jest z góry określony
- Obsługa relacji pomiędzy danymi
- Nie tak szybkie jak bazy klucz-wartość i dokumentowe dla poszukiwania po identyfikatorze, ale bardzo duże możliwości efektywnego przeszukiwania i analizy danych
- Zastosowanie - aplikacje Web, GIS, systemy bankowe, rejestry

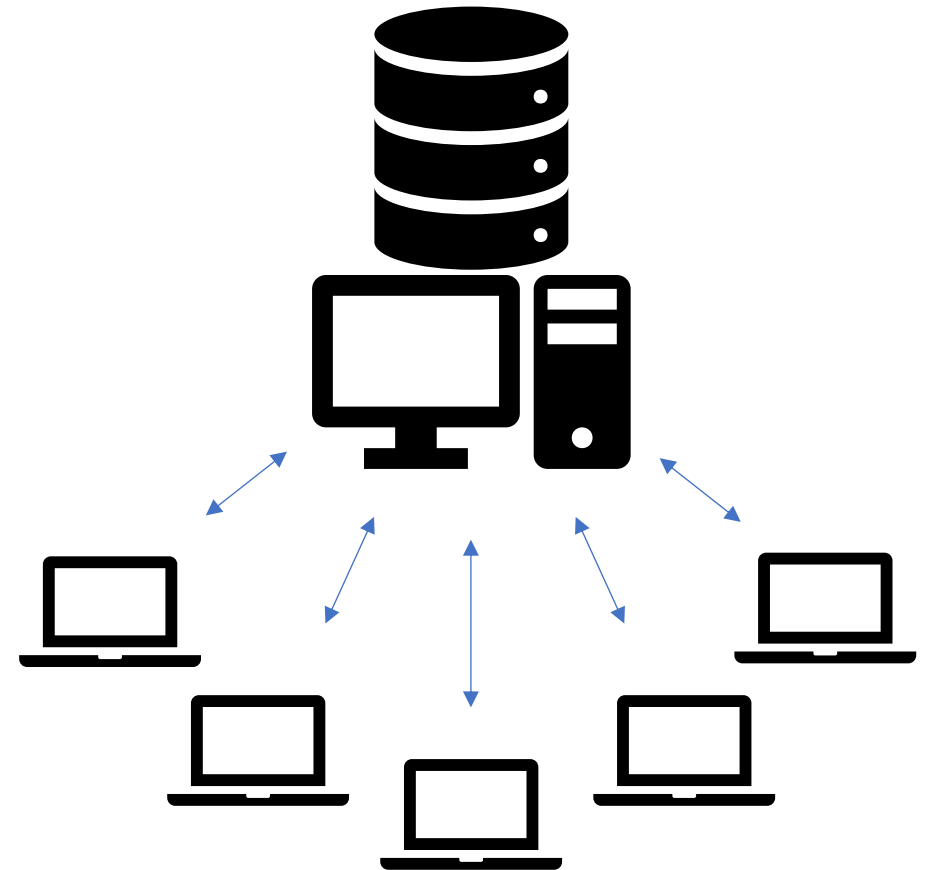
Przykłady systemów: **Oracle, SQL Server, Postgres, MySQL** publiczne, aplikacje mobilne

	sub_area	prot_categ	a_year	gat_gl	wiek	udzial	typ_dstan	typ_udz_1
2	4.59	OCH CENNE	2016	SO	46	7	IGLASTE	DOMINATED
3	4.18	OCH CENNE	2016	BRZ	38	4	LISCIASTE	MIXED
4	7.97	OCH CENNE	2016	BK	152	8	LISCIASTE	PURE
5	1.42	OCH CENNE	2016	MD	47	7	IGLASTE	DOMINATED
6	4.15	OCH CENNE	2016	SO	87	6	IGLASTE	DOMINATED
7	1.51	OCH CENNE	2016	SO	97	10	IGLASTE	PURE
8	10.81	OCH CENNE	2016	DB	137	7	LISCIASTE	DOMINATED
9	0.81	OCH CENNE	2016	DB	8	4	LISCIASTE	MIXED
10	2.96	OCH CENNE	2016	BK	20	5	LISCIASTE	DOMINATED
11	2.29	OCH CENNE	2016	MD	50	3	IGLASTE	MIXED
12	5.19	OCH CENNE	2016	SO	97	7	IGLASTE	DOMINATED
13	4.56	OCH CENNE	2016	BK	122	5	LISCIASTE	DOMINATED
14	4.78	OCH CENNE	2016	SO	97	9	IGLASTE	PURE
15	4.63	OCH CENNE	2016	DB	137	4	LISCIASTE	MIXED
16	8.85	OCH CENNE	2016	DB	162	5	LISCIASTE	DOMINATED
17	3.96	OCH CENNE	2016	DB	162	7	LISCIASTE	DOMINATED

Bazy danych - relacyjna

Baza przechowująca dane w specjalnej strukturze plików na serwerze

- Dane są przechowywane na serwerze i udostępniane poprzez sieć
- Pliki źródłowe nie są czytelne dla innego oprogramowania
- Z bazy może korzystać wielu użytkowników jednocześnie
- Wymaga stale działającego programu-serwera bazy danych



Wprowadzenie do PostgreSQL

Wprowadzenie do PostgreSQL





Sfinansowano ze środków
NARODOWEGO FUNDUSZU
OCHRONY ŚRODOWISKA
i GOSPODARKI WODNEJ

- PostgreSQL jest relacyjną bazą danych typu klient-serwer
- Następca bazy Ingres, rozwój od 1995 roku
- Aktualna wersja stabilna 15.4, rozwojowa 16
- Posiada własną licencję typu open source - "PostgreSQL License"
- Pakiety dostępne na platformy Mac, Linux i Windows, działa też na BSD i Solaris

11th August 2022: PostgreSQL 14.5, 13.8, 12.12, 11.17, 10.22, and 15 Beta 3 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 30 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 PostgreSQL community provides many helpful places to become familiar with the technology, discover how it works, and find career opportunities. Reach out to the community [here](#).

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

Latest Releases

2022-08-11 - PostgreSQL 14.5, 13.8, 12.12, 11.17, 10.22, and 15 Beta 3 Released!

The PostgreSQL Global Development Group has released an update to all supported versions of PostgreSQL, including 14.5, 13.8, 12.12, 11.17, and 10.22, as well as the third beta release of PostgreSQL 15. This release closes **one security vulnerability** and fixes over 40 bugs reported over the last three months.

For the full list of changes, please review the [release notes](#).

In the spirit of the open source PostgreSQL community, we strongly encourage you to test the new features of PostgreSQL 15 on your systems to help us eliminate bugs or other issues that may exist. While we do not advise you to run PostgreSQL 15 Beta 3 in production environments, we encourage you to find ways to run your typical application workloads against this beta release.

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

PostgreSQL - licencjonowanie

Licencja PostgreSQL należy do grupy tzw. liberalnych licencji open source, jest zbliżona do BSD lub MIT. Prawa autorskie należą do The PostgreSQL Global Development Group oraz Uniwersytetu Kaliforni.

Warunki:

- Zezwolenie na użytkowanie, kopiowanie, modyfikację, rozpowszechnianie oprogramowania i dokumentacji do dowolnych celów i bez pisemnego zezwolenia, pod warunkiem:
 - zachowania informacji o oryginalnych autorach
 - niewystępowania z żadnymi roszczeniami odnośnie oprogramowania do oryginalnych autorów.

Licencja nie zawiera zapisu typu copyleft, co oznacza że dozwolone jest tworzenie płatnych i zamkniętych wersji (np. EnterpriseDB Advanced Server, PostgresXL).

PostgreSQL - instalacja

Zalecanym sposobem instalacji w środowisku Windows jest użycie instalatora od EnterpriseDB.

Należy wybrać pakiet "PostgreSQL", a nie "EnterpriseDB Advanced Server" gdyż ten drugi jest wersją testową płatnego oprogramowania.

Możliwa jest instalacja więcej niż jednej wersji PostgreSQL na jednym systemie

(np. w celu testów, aktualizacji). Aktualizacja polega na instalacji nowszej wersji,

przeniesieniu danych i wyłączeniu starej.

Szczegółowy opis instalacji znajduje się w zeszycie ćwiczeń.

PostgreSQL - instalacja



Sfinansowano ze środków
NARODOWEGO FUNDUSZU
OCHRONY ŚRODOWISKA
i GOSPODARKI WODNEJ

The screenshot shows the EDB website's download page for PostgreSQL. The header includes the EDB logo, a webinar announcement, and navigation links. The main heading is 'Download PostgreSQL' with a sub-heading 'Open source PostgreSQL packages and installers from EDB'. Below this is a table listing various PostgreSQL versions and their availability for different operating systems.

PostgreSQL Version	Linux x86-64	Linux x86-32	Mac OS X	Windows x86-64	Windows x86-32
14.5	postgresql.org	postgresql.org			Not supported
13.8	postgresql.org	postgresql.org			Not supported
12.12	postgresql.org	postgresql.org			Not supported
11.17	postgresql.org	postgresql.org			Not supported
10.22					
9.6.24*					
9.5.25*					

<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>

PostgreSQL - PostGIS

PostgreSQL bez dodatków posiada wsparcie dla danych geometrycznych 2D, ale bez obsługi układów współrzędnych, integracji z oprogramowaniem GIS i funkcji analiz przestrzennych.

Dodatkiem rozszerzającym PostgreSQL o te funkcje jest PostGIS – osobny projekt rozwijany przez inną grupę programistów, ale bardzo dobrze zintegrowany z PostgreSQL.

Instalacja PostGIS w środowisku Windows jest wykonywana z użyciem narzędzia **StackBuilder**

Inne systemy - tzw. metapakiety (apt-get install postgresql-11-postgis, brew install postgis)



<https://postgis.net/docs/>

Strukturalne aspekty budowy i działania systemu bazodanowego PostgreSQL

Struktura systemu bazodanowego PostgreSQL



Sfinansowano ze środków
NARODOWEGO FUNDUSZU
OCHRONY ŚRODOWISKA
i GOSPODARKI WODNEJ

Instalacja systemu PostgreSQL tworzy **klaster baz danych (database cluster)**. Jest to kolekcja baz danych utrzymywanych przez pojedynczy proces serwera PostgreSQL. Klaster posiada swoje miejsce na dysku oraz zbiór kont użytkowników.

Wewnątrz klastra znajdują się **bazy danych (database)**. Bazy są izolowanymi od siebie przestrzeniami do przechowywania danych - w typowej instalacji PostgreSQL, z poziomu jednej bazy nie można odwołać się do danych zapisanych w innej. Istnieje rozszerzenie, które pozwala na takie odwołania - Foreign Data Wrapper (FDW).

Baza danych jest podzielona na **schematy (schema)**. Każda baza zawiera schemat public. Każdy użytkownik może w nim stworzyć tabelę. Możliwe jest tworzenie dodatkowych schematów w bazie, które można wykorzystać do logicznego grupowania danych (np. na własne i importowane ze źródeł zewnętrznych, aktualne i archiwalne) lub ograniczenia dostępu do tworzenia tabel.

Struktura systemu bazodanowego PostgreSQL

W schemacie znajdują się relacje (relations). Relacja w rozumieniu PostgreSQL to zbiór wierszy posiadających określone kolumny. Relacją może być:

- **tabela (table)** - podstawowy typ relacji, jest zapisywana na dysku, nie musi zależeć od danych już istniejących w bazie.
- **tabela obca (foreign table)** - tabela która odwołuje się do zewnętrznego źródła danych, może być używana tak jak natywna tabela, choć czasem z ograniczeniami (np. źródło nie pozwala na zapis)
- **widok (view)** - relacja która jest wynikiem wykonania określonego zapytania, dane widoku nie są zapisywane na dysku, a przeliczane na bieżąco z danych źródłowych.
- **zmaterializowany widok (materialized view)** - hybryda widoku i tabeli: dane są wynikiem zapytania na już istniejących danych, ale są zapisywane na dysku i odświeżane na żądanie.

Zarządzanie danymi – tworzenie i obsługa danych przestrzennych PgAdmin oraz QGIS

Zarządzanie danymi

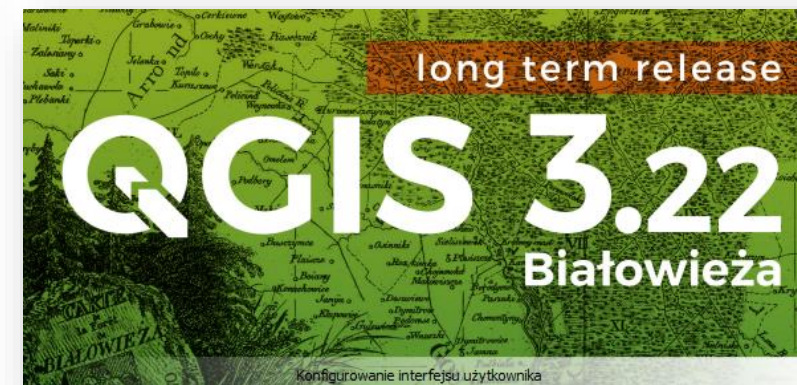


Sfinansowano ze środków
NARODOWEGO FUNDUSZU
OCHRONY ŚRODOWISKA
i GOSPODARKI WODNEJ

- psql - program działający w linii komend (cmd)
- pgAdmin - program działający w środowisku przeglądarki internetowej, posiada interfejs graficzny, umożliwia podgląd danych przestrzennych - instalowany razem z PostgreSQL
- QGIS - program GIS projektowany pierwotnie jako przeglądarka do danych przestrzennych w bazie PostGIS

```
Wiersz polecenia
Microsoft Windows [Version 10.0.19044.1889]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users>
```



Zarządzanie danymi - QGIS

Utwórz nowe połączenie z PostGIS

Informacja o połączeniu

Nazwa: szkolenie

Usługa:

Host: localhost

Port: 5432

Baza danych: szkolenie

Tryb SSL: wyłącz

Uwierzytelnianie

Konfiguracje Bez zabezpieczeń

Nazwa użytkownika: kursant Zapisz

Hasło: Zapisz

Warning: credentials stored as plain text in plik projektu.

Konwertuj na szyfrowaną konfigurację

Test połączenia

Wyświetlaj tylko zarejestrowane warstwy

Nie sprawdzaj typu dla kolumn GEOMETRY

Sprawdź tylko schemat "public"

Pokaż także tabele bez geometrii

Użyj szacunkowych metadanych tabeli

Zezwól na zapisywanie i wczytywanie z bazy projektów QGIS

OK Anuluj Pomoc

Warstwa -> Dodaj warstwę -> Dodaj warstwę PostGIS

Zarządzanie danymi - QGIS

Warstwa -> Dodaj warstwę -> Dodaj warstwę PostGIS

Zarządzanie źródłami danych | PostgreSQL

Przeglądarka

- Wektor
- Raster
- Siatka
- Chmura punktów
- CSV
- GeoPackage
- GPS
- SpatiaLite
- PostgreSQL
- MS SQL Server
- Oracle
- Warstwa wirtualna
- SAP HANA
- WMS/WMTS
- WFS / OGC API - Cechy
- WCS
- XYZ

Połączenia

SZKOLENIE

Połącz Nowe Edytuj Usuń Wczytaj Zapisz

Schema	Tabela	Komentarz	Kolumna	Typ danych	Dane przestrzenne	SRID	ID
public	bdot_ot_bubd_a		geom	Geometria	Polygon	2180	
public	demo_zaludnienie		geom	Geometria	MultiPolygon	2180	
public	egib_dzialki_ewidencyjne		geom	Geometria	MultiPolygon	2180	
public	gdos_obszary_chronionego_krajobrazu		geom	Geometria	MultiPolygon	2180	
public	gdos_obszary_specjalnej_ochrony		geom	Geometria	MultiPolygon	2180	
public	gdos_parki_krajobrazowe		geom	Geometria	MultiPolygon	2180	
public	gdos_parki_narodowe		geom	Geometria	MultiPolygon	2180	
public	gdos_pomniki_przyrody_a		geom	Geometria	MultiPolygon	2180	
public	gdos_pomniki_przyrody_p		geom	Geometria	MultiPoint	2180	
public	gdos_rezerwaty		geom	Geometria	MultiPolygon	2180	
public	gdos_specjalne_obszary_chrony		geom	Geometria	MultiPolygon	2180	
public	geol_osiadanie		geom	Geometria	Point	2180	
public	geol_warunki_budowlane		geom	Geometria	MultiPolygon	2180	
public	infr_swiatlowody		geom	Geometria	Point	2180	
public	nmt		rast	Raster	Raster	2180	
public	pokrycie_terenu		geom	Geometria	Polygon	2180	
public	prg_granice_gmin		geom	Geometria	MultiPolygon	2180	
public	prg_granice_jednostek_ewidencyjnych		geom	Geometria	MultiPolygon	2180	
public	prg_granice_obrebow_ewidencyjnych		geom	Geometria	MultiPolygon	2180	
public	prg_granice_powiatow		geom	Geometria	MultiPolygon	2180	
public	raster_columns		extent	Geometria	Polygon	2180	r...
public	zagrozenia_powodzie		geom	Geometria	MultiPolygon	2180	
tiger							

Pokaż także tabele bez geometrii

Ustaw filtr Zamknij Dodaj Pomoc

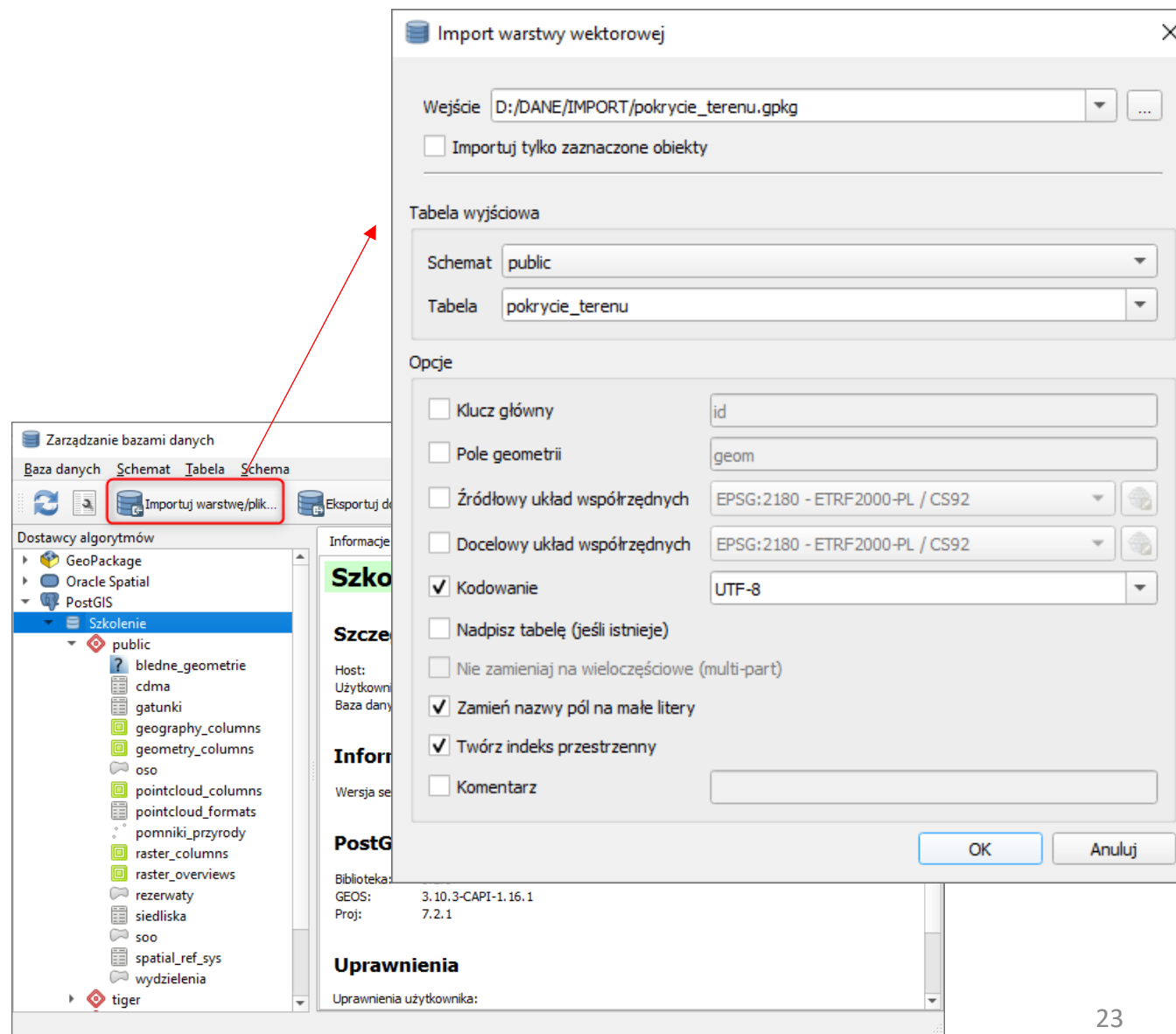
- **Połączenie**: zestaw parametrów (adres serwera, port, nazwa bazy, login i hasło). Dla każdej bazy danych musi istnieć osobne połączenie, nawet gdy znajdują się na tym samym serwerze.
- **Typ danych**: format przechowywania danych przestrzennych na serwerze - Geometry dla współrzędnych płaskich, Geography dla współrzędnych geograficznych
- **SRID**: liczbowy identyfikator układu współrzędnych (kod EPSG)
- **Metadane tabeli**: informacja o liczbie obiektów, ich typie geometrii oraz układzie współrzędnych. Szacunkowe metadane są obliczane dużo szybciej.
- **ID obiektu**: niepowtarzalny identyfikator obiektu. Jeśli tabela posiada zdefiniowany klucz główny, to będzie on automatycznie użyty. Jeśli nie, ale możliwe jest wskazanie kolumny (lub ich kombinacji) która da niepowtarzalne wartości, to można go wskazać ręcznie. Jeśli i ten warunek nie będzie mógł być spełniony, to dane będą dostępne w trybie tylko do odczytu niezależnie od uprawnień użytkownika.

Zarządzanie danymi - QGIS



Zasilenie bazy z użyciem funkcji QGIS:

- Przy pomocy QGIS można zasilić bazę dowolnymi danymi wektorowymi obsługiwany przez QGIS: z pliku SHP, GML, CSV, warstwy tymczasowej...
- Wadą narzędzia do ładowania w QGIS jest powolne działanie dla serwerów zdalnych i danych ponad 1000 obiektów.



Zarządzanie danymi - psql

psql jest narzędziem dostępnym z linii komend.

Uruchomienie:

- po otwarciu menu Start wpisać "cmd"
- odnaleźć w systemie plik psql.exe (np. C:\PostgreSQL\15\bin) i przeciągnąć go do okna konsoli
- dopisać parametry:
 - -h localhost
 - -d szkolenie
 - -U postgres

UWAGA: podczas wpisywania hasła nie pokazują się żadne znaki - jest to normalne.

Zarządzanie danymi - psql

Komendy psql - zatwierdzone przez wciśnięcie Enter:

\l - wyświetlenie listy baz danych w systemie

\l+ - jak wyżej, ale z podaniem rozmiaru na dysku

\connect gis - przełączenie się na bazę gis

\dt - wyświetlenie listy tabel w bazie

\dt+ - jak wyżej, ale z podaniem rozmiaru na dysku

\du - wyświetlenie listy użytkowników

\dv - wyświetlenie listy widoków

\di - wyświetlenie listy indeksów

Zarządzanie danymi - psql

\timing - pokazuje czas wykonania zapytania

\d <nazwa tabeli> - pokazuje dostępne kolumny w tabeli

\a - wyłącza / włącza justowanie tabeli

\x - pokazuje kolumny pionowo zamiast poziomo

\q - koniec sesji

Zarządzanie danymi - psql

Zapytanie może być wpisane w wielu wierszach - wiersz kończy się poprzez wciśnięcie Enter.

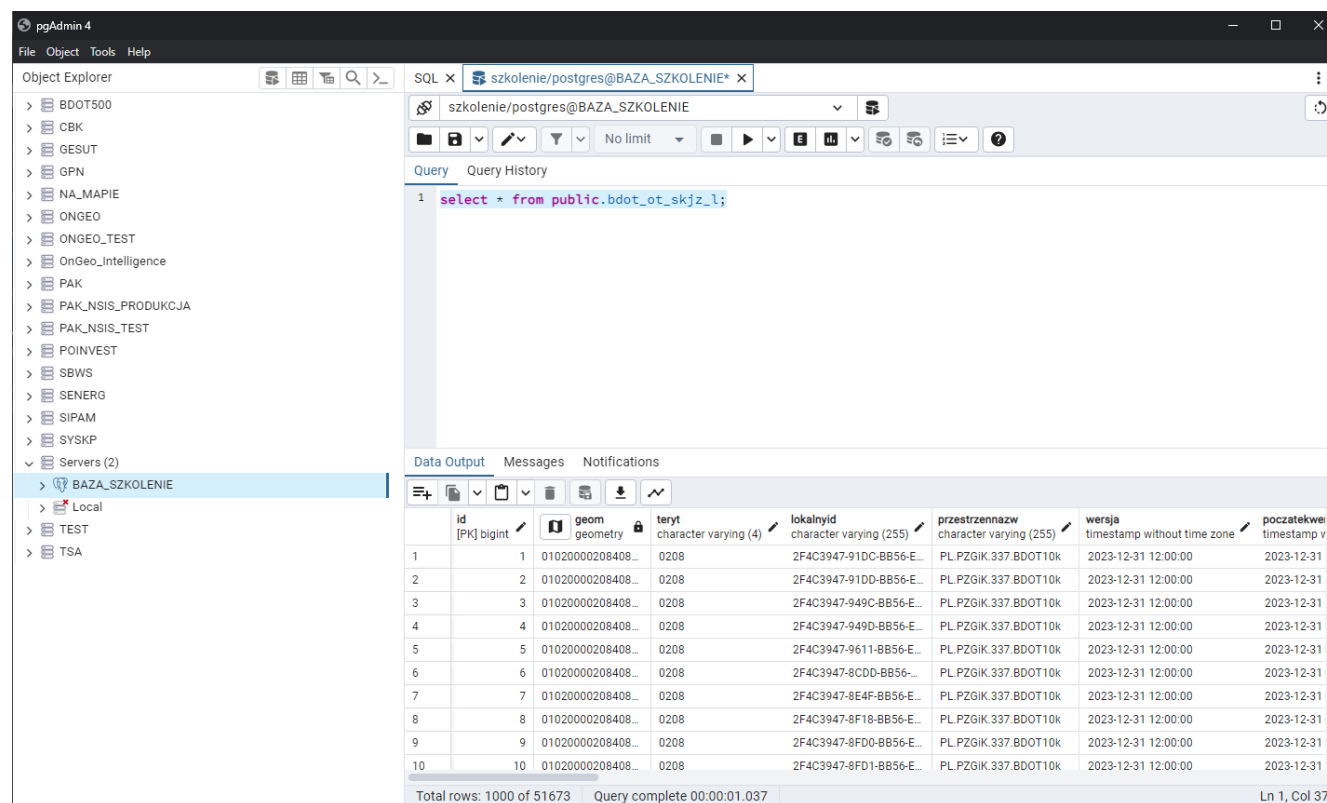
Zapytanie kończy się znakiem ; i wciśnięcie Enter.

Np.

```
szkolenie=# select *  
szkolenie-# from geometry_columns;
```

Zarządzanie danymi – pgAdmin

W programie pgAdmin definiuje się połączenia podobnie jak w QGIS, z tym, że wykorzystując jedną deklarację połączenia w pgAdmin można się przełączać pomiędzy różnymi bazami danych na tym samym serwerze - a w QGIS nie.



The screenshot shows the pgAdmin 4 interface. The left pane displays the Object Explorer with a tree view of databases, including 'BAZA_SZKOLENIE'. The main window shows a SQL query: `select * from public.bdod_ot_skjz_l1;`. Below the query, the Data Output pane displays a table with 10 rows of data. The table columns are: id [PK] bigint, geom geometry, teryt character varying (4), lokalnyid character varying (255), przestrzennazw character varying (255), wersja timestamp without time zone, and poczatekwe timestamp v.

	id [PK] bigint	geom geometry	teryt character varying (4)	lokalnyid character varying (255)	przestrzennazw character varying (255)	wersja timestamp without time zone	poczatekwe timestamp v
1	1	01020000208408...	0208	2F4C3947-91DC-BB56-E...	PL.PZGIK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31
2	2	01020000208408...	0208	2F4C3947-91DD-BB56-E...	PL.PZGIK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31
3	3	01020000208408...	0208	2F4C3947-949C-BB56-E...	PL.PZGIK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31
4	4	01020000208408...	0208	2F4C3947-949D-BB56-E...	PL.PZGIK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31
5	5	01020000208408...	0208	2F4C3947-9611-BB56-E...	PL.PZGIK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31
6	6	01020000208408...	0208	2F4C3947-8CDD-BB56-...	PL.PZGIK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31
7	7	01020000208408...	0208	2F4C3947-8E4F-BB56-E...	PL.PZGIK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31
8	8	01020000208408...	0208	2F4C3947-8F18-BB56-E...	PL.PZGIK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31
9	9	01020000208408...	0208	2F4C3947-8FD0-BB56-E...	PL.PZGIK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31
10	10	01020000208408...	0208	2F4C3947-8FD1-BB56-E...	PL.PZGIK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31

Dane w bazie danych

Typy danych i konwersja typów

Dane w bazie danych – kolumny i typy

- Różnica pomiędzy arkuszem kalkulacyjnym a bazą danych
- Kolumna może przechowywać tylko dane określonego typu, np. liczba całkowita, tekst
- Podobieństwo do tabeli atrybutów w GIS
- PostgreSQL posiada 56 typów + warianty tablicowe (lista wartości w pojedynczym polu)
- Silne typowanie: nie można wpisać do kolumny typu INTEGER wartości tekstowej itp. – w niektórych innych systemach baz danych można to zrobić

Dane w bazie danych – typy TEKSTOWE

- Typ **CHAR**: tekst stałej długości (np. suma kontrolna)
- Typ **VARCHAR**: tekst zmiennej długości, opcjonalny typmod dla ograniczenia maksymalnej długości (alias - CHARACTER VARYING)
- Typ **TEXT**: tekst dowolnej długości
- Typ **XML**: dokument XML - musi być well-formed
- Typ **TSVECTOR**: dla wyszukiwania pełnotekstowego

A, B, C

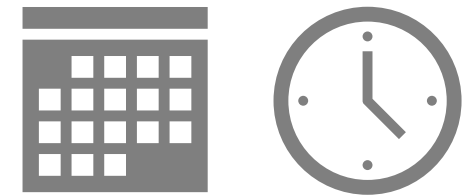
Dane w bazie danych – typy LICZBOWE

- Typ **INTEGER**: liczba całkowita 32 bit (alias - INT4)
- Typ **BIGINT**: liczba całkowita 64 bit (alias - INT8)
- Typ **REAL**: liczba zmiennoprzecinkowa pojedynczej precyzji
- Typ **DOUBLE PRECISION**: liczba zmiennoprzecinkowa podwójnej precyzji (alias - **FLOAT**)
- Typ **NUMERIC**: liczba dziesiętna o zdefiniowanej precyzji (alias - **DECIMAL**)

1, 2, 3

Dane w bazie danych – typy DATA i CZAS

- Typ **DATE**: tylko data
- Typ **TIME**: tylko czas (może być WITH TIME ZONE i WITHOUT TIME ZONE)
- Typ **TIMESTAMP**: data i czas - w wariantach WITH TIME ZONE i WITHOUT TIME ZONE



Dane w bazie danych – typy INNE

- Typ **BYTEA**: dane binarne
- Typ **BOOLEAN**: wartość prawda/fałsz
- Typ **HSTORE**: pary klucz-wartość
- Typy **JSON** i **JSONB**: obiekty JSON (można użyć do wpisania wielu wartości w jedno pole)
- Typ **OID**: identyfikator dla obiektów systemowych
- Pseudo-typ **SERIAL**: kolumna **INTEGER**+sekwencja+wartość domyślna, używany do nadawania identyfikatorów
- Typy **POINT**, **PATH**, **POLYGON**: typy grafiki wektorowej 2D
 - - nie używać do danych GIS!

Dane w bazie danych – typy POSTGIS

- Typ **GEOMETRY**: najczęściej używany. Dowolny układ współrzędnych, wszystkie obliczenia są wykonywane na współrzędnych płaskich, wyniki zwracane w jednostkach układu. Opcjonalny typmod na typ geometrii (POINT, LINESTRING, POLYGON...) oraz układ współrzędnych.
- Typ **GEOGRAPHY**: dla współrzędnych geograficznych długość-szerokość, obliczenia wykonywane metodami geodezji wyższej, wyniki i parametry podawane są w metrach.



Konwersja typów

- Konwersja typów nazywa się rzutowaniem (CAST)
- Rzutowanie w standardzie SQL: CAST ('2019-09-05' AS date); - słowo kluczowe CAST, następnie w nawiasie wartość, słowo kluczowe AS i nazwa typu.
- Dodatkowo w PostgreSQL jest dostępny operator ::, np. '2010-09-05'::date

Uwaga: słowo kluczowe AS ma w SQL także inną rolę - może służyć do nadawania aliasów dla kolumn - szczegóły w części poświęconej zapytaniom SELECT.

Podstawy SQL

Structured Query Language

- Strukturalny język zapytań używany do tworzenia, modyfikowania baz danych oraz do umieszczania i pobierania danych z baz danych. (def. uniwersalna → Wikipedia)
- Jest językiem deklaratywnym, co oznacza że użytkownik definiuje **co** chce uzyskać, natomiast decyzję o tym **jak** to zrobić podejmuje maszyna
- Jest zdefiniowany standard (ANSI-SQL) oraz implementacje (dialekty) mniej lub bardziej zgodne ze standardem
- Najważniejsze instrukcje i konwencje są wspólne dla wszystkich baz, ale...
- Różne sposoby formatowania zapytań, różne nazwy funkcji, różne podejście do typów danych

Elementy języka SQL

- **DML** - Data Manipulation Language

Polecenia wyszukiwania, dodawania, edycji i usuwania danych

- **DDL** - Data Definition Language

Polecenia tworzenia, zmiany i usuwania struktur danych (tabel, kolumn...)

- **DCL** - Data Control Language

Polecenia kontroli dostępu do danych (tylko bazy klient-serwer)

Instrukcja SELECT i filtrowanie wyników

Zapytania SELECT służą przede wszystkim do pobierania danych

- W PostgreSQL **nie** jest gwarantowane, że SELECT nie zmieni danych, np. SELECT AddGeometryColumn...
- Zapytanie SELECT może operować na "0 lub więcej relacji", co oznacza, że niekoniecznie musi odnosić się do danych zapisanych w bazie - może również sprawdzać ustawienia systemowe, lub... posłużyć za kalkulator
- Przykłady SELECT na 0 relacji:
 - *SELECT 1;* -- zapytanie zwróci 1, służy np. do sprawdzenia czy system działa
 - *SELECT PostGIS_full_version();* -- sprawdzenie wersji PostGIS
 - *SELECT now();* -- sprawdzenie aktualnej daty i godziny w systemie
 - *SELECT 50000::real / 365::real;* -- użycie PostgreSQL jako kalkulatora

SQL - Wybieranie danych

Przykładowe zapytania

Query Query History

```
1 select * from public.bdot_ot_skjz_l;
```

Data Output Messages Notifications

	id [PK] bigint	geom geometry	teryt character varying (4)	lokalnyid character varying (255)	przestrzennazw character varying (255)	wersja timestamp without time zone	początekwersjiobiektu timestamp without time zone	oznaczeniez character va
1	1	01020000208408...	0208	2F4C3947-91DC-BB56-E...	PL.PZGiK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31 12:00:00	GI-TOPO.60
2	2	01020000208408...	0208	2F4C3947-91DD-BB56-E...	PL.PZGiK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31 12:00:00	GI-TOPO.60
3	3	01020000208408...	0208	2F4C3947-949C-BB56-E...	PL.PZGiK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31 12:00:00	GI-TOPO.60
4	4	01020000208408...	0208	2F4C3947-949D-BB56-E...	PL.PZGiK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31 12:00:00	GI-TOPO.60
5	5	01020000208408...	0208	2F4C3947-9611-BB56-E...	PL.PZGiK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31 12:00:00	GI-TOPO.60
6	6	01020000208408...	0208	2F4C3947-8CDD-BB56-...	PL.PZGiK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31 12:00:00	GI-TOPO.60
7	7	01020000208408...	0208	2F4C3947-8E4F-BB56-E...	PL.PZGiK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31 12:00:00	GI-TOPO.60
8	8	01020000208408...	0208	2F4C3947-8F18-BB56-E...	PL.PZGiK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31 12:00:00	GI-TOPO.60
9	9	01020000208408...	0208	2F4C3947-8FD0-BB56-E...	PL.PZGiK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31 12:00:00	GI-TOPO.60
10	10	01020000208408...	0208	2F4C3947-8FD1-BB56-E...	PL.PZGiK.337.BDOT10k	2023-12-31 12:00:00	2023-12-31 12:00:00	GI-TOPO.60

Przykładowe zapytania

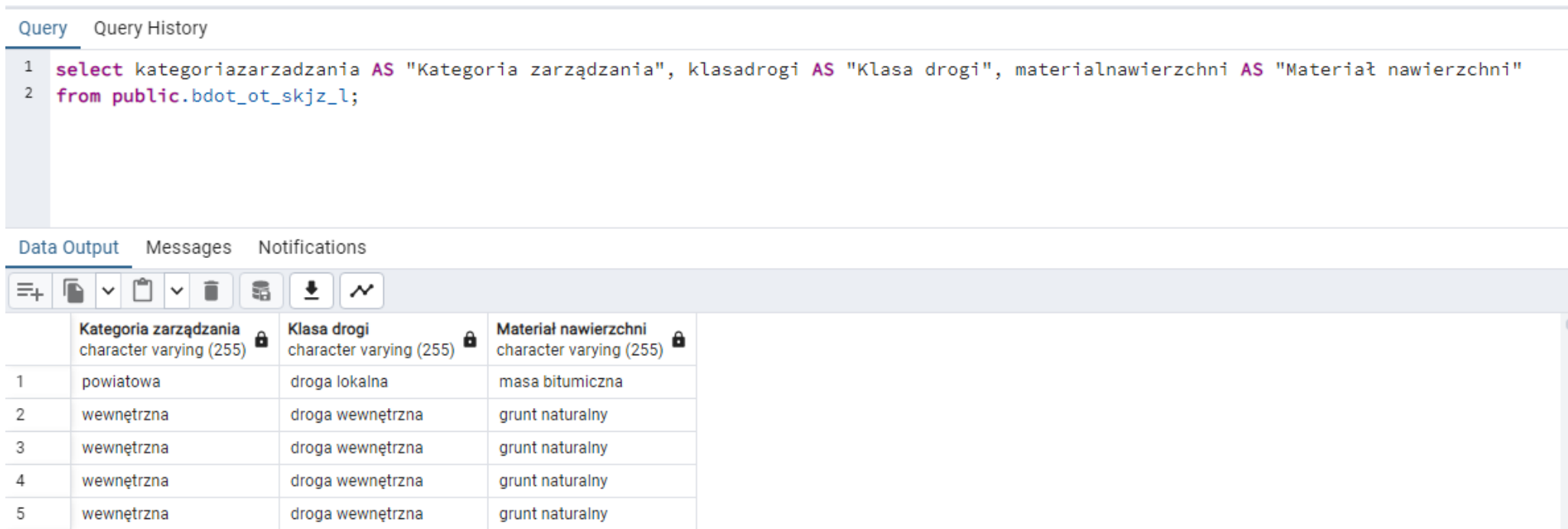
Query Query History

```
1 select kategoriazarzadzania, klasadrogi, materialnawierzchni from public.bdot_ot_skjz_l;
```

Data Output Messages Notifications

	kategoriazarzadzania character varying (255) 🔒	klasadrogi character varying (255) 🔒	materialnawierzchni character varying (255) 🔒
1	powiatowa	droga lokalna	masa bitumiczna
2	wewnętrzna	droga wewnętrzna	grunt naturalny
3	wewnętrzna	droga wewnętrzna	grunt naturalny
4	wewnętrzna	droga wewnętrzna	grunt naturalny
5	wewnętrzna	droga wewnętrzna	grunt naturalny
6	gminna	droga lokalna	masa bitumiczna
7	wewnętrzna	droga wewnętrzna	masa bitumiczna
8	gminna	droga lokalna	masa bitumiczna
9	gminna	droga lokalna	masa bitumiczna

Przykładowe zapytania



The screenshot shows a SQL query editor with a query window and a data output window. The query window contains the following SQL code:

```
1 select kategoriazarzadzania AS "Kategoria zarządzania", klasadrogi AS "Klasa drogi", materialnawierzchni AS "Materiał nawierzchni"  
2 from public.bdot_ot_skjz_l;
```

The data output window displays the results of the query in a table format:

	Kategoria zarządzania character varying (255)	Klasa drogi character varying (255)	Materiał nawierzchni character varying (255)
1	powiatowa	droga lokalna	masa bitumiczna
2	wewnętrzna	droga wewnętrzna	grunt naturalny
3	wewnętrzna	droga wewnętrzna	grunt naturalny
4	wewnętrzna	droga wewnętrzna	grunt naturalny
5	wewnętrzna	droga wewnętrzna	grunt naturalny

- Klauzula AS - pozwala na nadanie aliasu dla nazwy kolumny, dzięki czemu nazwa w wyniku zapytania może być inna niż zapisana w systemie. Czasem jest to koniecznością, np. jeśli łączy się dane z dwóch tabel o powtarzających się nazwach kolumn.

Przykładowe zapytania

- Słowa kluczowe mogą być pisane zarówno wielkimi, jak i małymi literami. Konwencją jest pisanie wielkimi, ale nie jest to wymagane
 - Kolumny mogą mieć dowolnie długie nazwy i zawierać dowolne znaki, jeśli jednak:
 - zawierają spacje, znaki specjalne
 - zawierają słowa kluczowe SQL (np. select, table, insert, order)
 - zaczynają się od cyfry
 - zawierają wielkie litery
- **ich nazwy muszą być ujęte w "podwójny cudzysłów"**.

Dla osób dobrze znających QGIS: zasady formułowania wyrażeń w QGIS zostały zaczerpnięte z SQL, dlatego składnia zapytań w bazie danych będzie podobna do składni wyrażeń w QGIS.

SQL – ĆWICZENIE

- Wybierz wszystkie wiersze i wszystkie kolumny z tabeli „gdos_parki_krajobrazowe”
- Wybierz kolumny: „lokalnyid”, „szerokosc nawierzchni” z tabeli „bdot_ot_skjz_l”

SQL – ĆWICZENIE - ODPOWIEDZI

- Wybierz wszystkie wiersze i wszystkie kolumny z tabeli „gdos_parki_krajobrazowe”

	id [PK] integer	geom geometry	gid integer	nazwa character varying (254)	kodinspire character varying (254)
1	1	01060000208408...	4077	Park Krajobrazowy Gór ...	PL.ZIPOP.1393.PK.1
2	2	01060000208408...	1004019	Śnieżnicki Park Krajobra...	PL.ZIPOP.1393.PK.106
3	3	01060000208408...	4019	Śnieżnicki Park Krajobra...	PL.ZIPOP.1393.PK.106

SELECT * FROM

gdos_parki_krajobrazowe;

- Wybierz kolumny: „lokalnyid”, „szerokoscnawierzchni” z tabeli „bdot_ot_skjz_l”

SELECT lokalnyid, szerokoscnawierzchni FROM

bdot_ot_skjz_l;

	lokalnyid character varying (255)	szerokoscnawierzchni double precision
1	2F4C3947-91DC-BB56-E...	2.8
2	2F4C3947-91DD-BB56-E...	[null]
3	2F4C3947-949C-BB56-E...	[null]
4	2F4C3947-949D-BB56-E...	[null]
5	2F4C3947-9611-BB56-E...	[null]
6	2F4C3947-8CDD-BB56-...	3
7	2F4C3947-8E4F-BB56-E...	2.9
8	2F4C3947-8F18-BB56-E...	3
9	2F4C3947-8FD0-BB56-E...	3

SQL – Filtrowanie

- Do filtrowania służy klauzula **WHERE**
- W klauzuli **WHERE** można używać operatorów: =, !=, >, >=, <, <=, **IN** (lista wartości), **LIKE** (wyszukiwanie przybliżone)
- Wymienione operatory nie działają dla wartości **NULL** - zawsze zwrócą fałsz. Do porównania z wartością **NULL** trzeba stosować specjalne operatory **IS NULL** oraz **IS NOT NULL**.
- Warunki można łączyć wykorzystując operatory logiczne: **AND**, **OR**, **NOT**



SQL – Filtrowanie

- W klauzuli **WHERE** można podać jeden lub więcej warunków. Warunki są łączone operatorami logicznymi: **AND** (i), **OR** (lub)
- Nazwy kolumn podajemy bez cudzysłowu lub w podwójnym cudzysłowie.
- Wartości liczbowe podajemy bez cudzysłowu, separatorem dziesiętnym jest zawsze kropka.
- Wartości tekstowe podajemy zawsze w pojedynczym cudzysłowie.



SQL – Filtrowanie



SELECT * FROM bdot_ot_skjz_l WHERE szerokoscnaawierzchni < 10;

	klasadrogi character varying (255)	materialnawierzchni character varying (255)	nazwadrogi character varying (255)	liczbajezdni double precision	polozenie character varying (255)	szerokoscnaawierzchni double precision
1	droga lokalna	masa bitumiczna	[null]	1	na powierzchni gruntu	2.8
2	droga lokalna	masa bitumiczna	[null]	1	na powierzchni gruntu	3
3	droga wewnętrzna	masa bitumiczna	[null]	1	na powierzchni gruntu	2.9
4	droga lokalna	masa bitumiczna	[null]	1	na powierzchni gruntu	3
5	droga lokalna	masa bitumiczna	[null]	1	na powierzchni gruntu	3
6	droga lokalna	masa bitumiczna	[null]	1	na powierzchni gruntu	3
7	droga wewnętrzna	masa bitumiczna	[null]	1	na powierzchni gruntu	6

SELECT * FROM bdot_ot_skjz_l WHERE szerokoscnaawierzchni > 10;

	klasadrogi character varying (255)	materialnawierzchni character varying (255)	nazwadrogi character varying (255)	liczbajezdni double precision	polozenie character varying (255)	szerokoscnaawierzchni double precision
1	droga główna	masa bitumiczna	[null]	1	na powierzchni gruntu	31.5
2	droga wewnętrzna	kostka prefabrykowana	[null]	1	na powierzchni gruntu	11.5
3	droga wewnętrzna	kostka kamienna	[null]	1	na powierzchni gruntu	11.5
4	droga główna	masa bitumiczna	[null]	1	na powierzchni gruntu	10.5
5	droga główna ruchu prz...	masa bitumiczna	[null]	1	na powierzchni gruntu	10.5
6	droga główna ruchu prz...	masa bitumiczna	[null]	1	na powierzchni gruntu	13
7	droga główna	masa bitumiczna	[null]	1	na powierzchni gruntu	10.5

SQL – Filtrowanie

```
SELECT * FROM bdot_ot_skjz_l WHERE kategoriazaradzania = 'powiatowa' AND szerokosc nawierzchni >= 10;
```

(drogi powiatowe o szerokości równej lub większej od 10 m)

```
SELECT kategoriazaradzania, klasa drogi, material nawierzchni FROM bdot_ot_skjz_l WHERE material nawierzchni = 'masa bitumiczna' OR material nawierzchni = 'płyty betonowe';
```

(odcinki dróg o nawierzchni asfaltowej lub betonowej)

```
SELECT material nawierzchni, klasa drogi FROM bdot_ot_skjz_l WHERE klasa drogi = 'droga wewnętrzna';
```

(drogi wewnętrzne)

SQL – Filtrowanie

```
SELECT lokalnyid, klasadrogi, materialnawierzchni FROM bdot_ot_skjz_l WHERE  
materialnawierzchni LIKE 'tł%';
```

(drogi o nawierzchni z tłucznia)

```
SELECT lokalnyid, klasadrogi, materialnawierzchni FROM bdot_ot_skjz_l WHERE  
materialnawierzchni ILIKE 'tł%';
```

(drogi o nawierzchni z tłucznia)

SQL – Funkcje skalarne i agregujące

SQL – Funkcje skalarne i agregujące

Funkcja jest fragmentem kodu - w języku SQL bądź innym obsługiwany przez bazę - nadającym się do wielokrotnego użytku.

W PostgreSQL funkcje nie muszą być "czyste", tj. przekształcać zbioru argumentów w zbiór wartości bez pozostawiania trwałych zmian w bazie.

Funkcja w PostgreSQL jest definiowana przez nazwę oraz liczbę i typy argumentów - w razie pomyłki w argumentach, komunikat o błędzie będzie brzmiał `function does not exist`.

Wyróżniamy **funkcje skalarne** (operujące na pojedynczych wierszach) i **agregujące** (operujące na grupach wierszy).

Oprócz wbudowanych funkcji, **użytkownik może tworzyć własne**.

PRZYKŁADY – funkcje tekstowe

SELECT LOWER(informacjadodatkowa) FROM bdot_ot_skjz_l WHERE informacjadodatkowa IS NOT NULL;

Funkcja LOWER zmienia wielkość liter w podanym tekście na małe.

SELECT UPPER(informacjadodatkowa) FROM bdot_ot_skjz_l WHERE informacjadodatkowa IS NOT NULL;

Funkcja UPPER zmienia wielkość liter w podanym tekście na wielkie.

SELECT INITCAP(informacjadodatkowa) FROM bdot_ot_skjz_l WHERE informacjadodatkowa IS NOT NULL;

Funkcja INITCAP zmienia wielkość liter w podanym tekście – pierwsza litera słowa jest wielka, pozostałe małe

SELECT lokalnyid AS "Identyfikator", REPLACE (zrodlodanychgeometrycznych, '10k', '1:10000') AS "Źródło danych" FROM bdot_ot_skjz_l WHERE zrodlodanychgeometrycznych LIKE 'mapa%';

Funkcja REPLACE przyjmuje 3 argumenty: tekst do zmiany, fragment podlegający zmianie oraz fragment, który ma zostać wstawiony.

PRZYKŁADY – funkcje tekstowe

```
SELECT lokalnyid, SPLIT_PART(lokalnyid,'-',2) FROM bdot_ot_skjz_l;
```

Funkcja `split_part` dzieli tekst na fragmenty według podanego separatora, oraz zwraca żądany fragment.

```
SELECT id, informacjadodatkowa, TRIM(informacjadodatkowa) FROM bdot_ot_skjz_l WHERE  
informacjadodatkowa ILIKE '%trasa%';
```

Funkcja `trim` obcina żądane znaki na początku, końcu lub z obu stron tekstu (najczęstsze zastosowanie: spacje na końcu).

```
SELECT informacjadodatkowa, LENGTH(informacjadodatkowa) FROM bdot_ot_skjz_l WHERE  
informacjadodatkowa IS NOT NULL;
```

Funkcja `length` oblicza długość ciągu znaków.

```
SELECT informacjadodatkowa, SUBSTRING(informacjadodatkowa,2,4) FROM bdot_ot_skjz_l WHERE  
informacjadodatkowa IS NOT NULL;
```

Funkcja `substring` ekstrahuje fragment tekstu według położenia pierwszego i ostatniego znaku.

PRZYKŁADY – funkcje liczbowe

```
SELECT id, ST_length(geom) AS dlugosc, CEIL(ST_length(geom)) FROM bdot_ot_skjz_l;
```

```
SELECT id, ST_length(geom) AS dlugosc, FLOOR(ST_length(geom)) FROM bdot_ot_skjz_l;
```

```
SELECT id, ST_length(geom) AS dlugosc, ROUND(ST_length(geom)) FROM bdot_ot_skjz_l;
```

Funkcja ceil zaokrągla liczbę dziesiętną w górę, floor - w dół, round - do końcówki 5 w dół, powyżej w górę.

```
SELECT id, ST_length(geom) AS dlugosc, ROUND(ST_length(geom)::numeric,2) FROM bdot_ot_skjz_l;
```

Funkcja round na typie numeric z podaniem drugiego argumentu - precyzji umożliwia zaokrąglenie do żądanej precyzji.

```
SELECT szerokosc nawierzchni, LOG(szerokosc nawierzchni) FROM bdot_ot_skjz_l;
```

```
SELECT szerokosc nawierzchni, LN(szerokosc nawierzchni) FROM bdot_ot_skjz_l;
```

Funkcja log oblicza logarytm przy podstawie 10, ln - przy podstawie e.

PRZYKŁADY – funkcje liczbowe

```
SELECT szerokosc nawierzchni, SQRT(szerokosc nawierzchni) FROM bdot_ot_skjz_l;
```

Funkcja sqrt oblicza pierwiastek kwadratowy. Dla liczb ujemnych zwraca błąd (ERROR: cannot take square root of a negative number) stąd konieczność ograniczenia do liczb nieujemnych.

```
SELECT szerokosc nawierzchni, szerokosc nawierzchni3 FROM bdot_ot_skjz_l;
```

```
SELECT szerokosc nawierzchni, szerokosc nawierzchni(1.0/3) FROM bdot_ot_skjz_l;
```

```
SELECT ABS(-3);
```

```
SELECT ABS(3);
```

```
SELECT SIGN(-3);
```

Funkcja abs oblicza wartość bezwzględną, sign - znak liczby.

```
SELECT RADIANS(90);
```

```
SELECT DEGREES(pi()/2);
```

Funkcja radians przelicza stopnie na radiany, degrees - radiany na stopnie.

SQL – Funkcje skalarne i agregujące

PRZYKŁADY – funkcje agregujące

```
SELECT MAX(szerokoscnaawierzchni) FROM bdot_ot_skjz_l;
```

Funkcja max zwraca największą wartość ze zbioru.

```
SELECT MIN(szerokoscnaawierzchni) FROM bdot_ot_skjz_l;
```

Funkcja min zwraca najmniejszą wartość ze zbioru.

```
SELECT SUM(szerokoscnaawierzchni) FROM bdot_ot_skjz_l;
```

Funkcja sum oblicza sumę wartości ze zbioru.

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

Funkcja count zwraca liczbę wartości w zbiorze.

SQL – Sortowanie i limitowanie

SQL – Sortowanie i limitowanie

- Do sortowania służy klauzula **ORDER BY**
- Domyślny kierunek sortowania - rosnąco, zmiana poprzez klauzulę **DESC**
- Domyślnie wartości puste są na początku, zmiana przez klauzulę **NULLS LAST**
- Możliwe jest sortowanie po więcej niż 1 kolumnie
- Ograniczenie liczby zwracanych wierszy - klauzula **LIMIT**

SQL – Sortowanie i limitowanie

***SELECT** lokalnyid, szerokosc nawierzchni **FROM** bdot_ot_skjz_l **ORDER BY** szerokosc nawierzchni;*

***SELECT** lokalnyid, szerokosc nawierzchni **FROM** bdot_ot_skjz_l **ORDER BY** szerokosc nawierzchni **ASC**;*

***SELECT** lokalnyid, szerokosc nawierzchni **FROM** bdot_ot_skjz_l **ORDER BY** szerokosc nawierzchni **DESC**;*

***SELECT** lokalnyid, szerokosc nawierzchni **FROM** bdot_ot_skjz_l **ORDER BY** szerokosc nawierzchni **DESC**
NULLS LAST;*

***SELECT** lokalnyid, szerokosc nawierzchni **FROM** bdot_ot_skjz_l **ORDER BY** szerokosc nawierzchni **ASC**
NULLS LAST;*

***SELECT** lokalnyid, szerokosc nawierzchni **FROM** bdot_ot_skjz_l **ORDER BY** szerokosc nawierzchni **ASC**
NULLS FIRST LIMIT 3;*

SQL – ĆWICZENIE

- Wybierz wszystkie wiersze z tabeli „infr_swiatlowody” sortując rosnąco po kolumnie „street_nam”
- Wybierz 5 wierszy z tabeli „ot_bubd_a” o najwyższych wartościach kolumny „liczbakondygnacji”

Uwzględnij fakt, że w kolumnach mogą występować wartości NULL.

SQL – ĆWICZENIE – ODPOWIEDZI

● ***SELECT * FROM infr_swiatlowody ORDER BY street_nam;***

	id [PK] integer	geom geometry	fid bigint	gml_id character varying (254)	terc bigint	simc bigint	city_name character varying (254)	ulic bigint	street_nam character varying (254)
1	8439258	01010000208408...	5635	s_dolnoslaskie_address...	208051	984522	Polanica-Zdrój	4	al. 1000-lecia
2	8677994	01010000208408...	27330	s_dolnoslaskie_address...	208051	984522	Polanica-Zdrój	4	al. 1000-lecia
3	8677993	01010000208408...	27329	s_dolnoslaskie_address...	208051	984522	Polanica-Zdrój	4	al. 1000-lecia
4	8677995	01010000208408...	27331	s_dolnoslaskie_address...	208051	984522	Polanica-Zdrój	4	al. 1000-lecia
5	8677996	01010000208408...	27332	s_dolnoslaskie_address...	208051	984522	Polanica-Zdrój	4	al. 1000-lecia

● ***SELECT * FROM bdot_ot_bubd_a ORDER BY liczbakondygnacji DESC NULLS LAST LIMIT 5;***

funkcjaogolnabudynku character varying (255)	przewazajacafuncjabudynku character varying (255)	liczbakondygnacji double precision	nazwa character varying (255)	fsbud character varying (32767)	idegib character varying (32767)
budynki mieszkalne	budynek wielorodzinny	11	[null]	budynek wielorodzinny	PL.PZGiK.262.EGiB.488da...
budynki mieszkalne	budynek wielorodzinny	11	[null]	budynek wielorodzinny	PL.PZGiK.262.EGiB.068d3...
budynki mieszkalne	budynek wielorodzinny	11	[null]	budynek wielorodzinny	PL.PZGiK.262.EGiB.7ae6c...
budynki mieszkalne	budynek wielorodzinny	11	[null]	budynek wielorodzinny	PL.PZGiK.262.EGiB.2f315...
budynki mieszkalne	budynek wielorodzinny	11	[null]	budynek wielorodzinny	PL.PZGiK.262.EGiB.754ee...

SQL – Grupowanie i operacje na zestawach danych

Możliwości funkcji agregujących są znacznie większe, gdy zastosuje się klauzulę GROUP BY. Możliwe jest wówczas obliczanie statystyk nie tylko dla całej tabeli, ale też podziału tabeli na kategorie.

Reguła: wszystkie kolumny, które mają być zwrócone przez zapytanie, muszą być:

- użyte w funkcji agregującej lub
- użyte w klauzuli GROUP BY.

Przykłady poprawnego użycia:

```
SELECT nazwa_stacji, MIN(srednia_temperatura_miesieczna::numeric) AS  
min_srednia_temp_mies, MAX(srednia_temperatura_miesieczna::numeric) AS  
max_srednia_temp_mies, ROUND(AVG(srednia_temperatura_miesieczna::numeric),1) AS  
sr_srednia_temp_mies FROM meteo_k_m_d_2024 GROUP BY nazwa_stacji ORDER BY 1;
```

SQL – ĆWICZENIE

Jaka jest łączna długość odcinków jezdni dla każdego rodzaju nawierzchni (materialnawierzchni)? Wynik wyraż w km z dokładnością do 3 miejsc po przecinku i posortuj malejąco względem obliczonej długości?

Ile jest budynków o poszczególnych funkcjach ogólnych (funkcjaogolnabudynku). Wynik posortuj po funkcji ogólnej budynku?

SQL – ĆWICZENIE – ODPOWIEDZI

Jaka jest łączna długość odcinków jezdni dla każdego rodzaju nawierzchni (materialnawierzchni)? Wynik wyraż w km z dokładnością do 3 miejsc po przecinku i posortuj malejąco względem obliczonej długości?

```
SELECT materialnawierzchni, ROUND((SUM(ST_length(geom))/1000)::numeric,3) AS  
"dlugosc [km]" FROM bdot_ot_skjz_l GROUP BY materialnawierzchni ORDER BY 2 DESC;
```

Ile jest budynków o poszczególnych funkcjach ogólnych (funkcjaogolnabudynku). Wynik posortuj po funkcji ogólnej budynku?

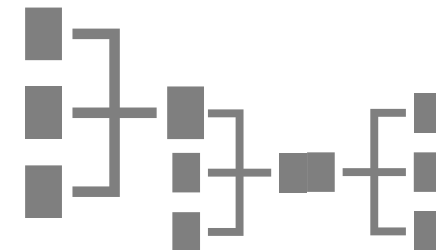
```
SELECT funkcjaogolnabudynku, COUNT(*) FROM bdot_ot_bubd_a GROUP BY  
funkcjaogolnabudynku ORDER BY 1;
```

SQL – Złączenia

SQL – Złączenia tabel

Złączenia tabel pozwalają na pobranie informacji z wielu tabel relacyjnej bazy danych, jeśli tylko posiadają one wspólną wartość na podstawie której można je połączyć.

Nie jest wymagane, aby relacja była utworzona na etapie projektowania bazy danych ani skonfigurowana przez administratora. Wystarczą identyczne wartości w kolumnach.



SQL – Złączenia domyślne

Złączenie domyślne wykorzystuje tylko klauzulę **WHERE**, bez użycia słowa kluczowego **JOIN**.

Przykład:

```
SELECT a.id, a.jpt_nazwa_ AS nazwa_gminy,  
b.jpt_nazwa_ AS nazwa_powiatu FROM  
prg_granice_gmin a, prg_granice_powiatow b  
WHERE LEFT(a.jpt_kod_je,4) = b.jpt_kod_je;
```

	id integer	nazwa_gminy character varying (128)	nazwa_powiatu character varying (128)
1	14	Nowa Ruda	powiat kłodzki
2	13	Bystrzyca Kłodzka	powiat kłodzki
3	12	Kłodzko	powiat kłodzki
4	11	Kudowa-Zdrój	powiat kłodzki
5	10	Polanica-Zdrój	powiat kłodzki
6	9	Radków	powiat kłodzki
7	8	Lądek-Zdrój	powiat kłodzki
8	7	Duszniki-Zdrój	powiat kłodzki
9	6	Stronie Śląskie	powiat kłodzki
10	5	Szczytna	powiat kłodzki
11	4	Kłodzko	powiat kłodzki
12	3	Nowa Ruda	powiat kłodzki
13	2	Międzylesie	powiat kłodzki
14	1	Lewin Kłodzki	powiat kłodzki

SQL – Złączenia wewnętrzne

Typ złączenia realizowany przez złączenie domyślne jest złączeniem **wewnętrznym**, oznacza to, że zwracane są tylko takie wyniki, które mają pasującą parę wartości w tabelach "a" i "b". Zapytanie można zapisać w następujący sposób:

```
SELECT a.id, a.jpt_nazwa_ AS nazwa_gminy, b.jpt_nazwa_ AS nazwa_powiatu FROM  
prg_granice_gmin a INNER JOIN prg_granice_powiatow b ON LEFT(a.jpt_kod_je,4) =  
b.jpt_kod_je;
```

lub krócej

```
SELECT a.id, a.jpt_nazwa_ AS nazwa_gminy, b.jpt_nazwa_ AS nazwa_powiatu FROM  
prg_granice_gmin a JOIN prg_granice_powiatow b ON LEFT(a.jpt_kod_je,4) = b.jpt_kod_je;
```

Warunek złączenia zapisywany jest za słowem kluczowym **ON**.

SQL – Złączenia zewnętrzne

Złączenie zewnętrzne pozwala na połączenie wielu tabel także wtedy, gdy nie zawsze wystąpią pasujące wiersze. Wówczas dla wiersza bez "pary" zostanie dopasowana wartość pusta (NULL).

SQL – Złączenia – LEFT/RIGHT

W złączeniu zewnętrznym **typu LEFT** zwracane są wszystkie wiersze z pierwszej tabeli, jeśli nie ma odpowiedników w drugiej tabeli, zwracana jest wartość NULL; jeśli zostanie użyta funkcja agregująca np. count lub sum, wówczas zwróci zero.

```
SELECT a.id, a.jpt_nazwa_ AS nazwa_gminy, b.jpt_nazwa_ AS nazwa_powiatu FROM  
prg_granice_gmin a LEFT JOIN prg_granice_powiatow b ON LEFT(a.jpt_kod_je,4) =  
b.jpt_kod_je;
```

W złączeniu zewnętrznym **typu RIGHT** zwracane są wszystkie wiersze z pierwszej tabeli, jeśli nie ma odpowiedników w drugiej tabeli, zwracana jest wartość NULL.

```
SELECT a.id, a.jpt_nazwa_ AS nazwa_gminy, b.jpt_nazwa_ AS nazwa_powiatu FROM  
prg_granice_gmin a RIGHT JOIN prg_granice_powiatow b ON LEFT(a.jpt_kod_je,4) =  
b.jpt_kod_je;
```

W złączeniu zewnętrznym **typu FULL OUTER** zwracane są wszystkie wiersze z wszystkich tabel. Dla niepowiązanych rekordów zwracane są wartości NULL.

```
SELECT a.id, a.jpt_nazwa_ AS nazwa_gminy, b.jpt_nazwa_ AS nazwa_powiatu FROM  
prg_granice_gmin a FULL OUTER JOIN prg_granice_powiatow b ON LEFT(a.jpt_kod_je,4) =  
b.jpt_kod_je;
```

SQL – Edycja/modyfikacja danych w bazie

SQL – Modyfikacja danych

Do modyfikacji danych służy komenda **UPDATE**.

Podstawowa składnia zapytania UPDATE jest następująca:

UPDATE <tabela> **SET** <kolumna> = <wartość>

Przykład:

UPDATE edycja_tabela **SET** uwagi = 'stacja: ' || INITCAP (nazwa_stacji);

każdy wyraz ma się zaczynać od wielkiej litery

UWAGA!

UPDATE w tej formie zmienia wartość we WSZYSTKICH wierszach tabeli!

SQL – Modyfikacja danych

Oprócz **UPDATE** dla całej tabeli, za pomocą SQL można modyfikować pojedyncze wiersze:

```
UPDATE edycja_tabela SET uwagi = 'brak danych' WHERE id = 9;
```

lub ich grupy według dowolnej klauzuli **WHERE**:

```
UPDATE edycja_tabela SET uwagi = 'luty' WHERE miesiac = '02';
```

SQL – Modyfikacja danych

Aby móc edytować dane za pomocą narzędzi graficznych takich jak pgAdmin lub QGIS, tabela musi mieć zdefiniowany **klucz główny**.

Dane importowane z zewnętrznych źródeł za pomocą QGIS czy ogr2ogr zwykle spełniają ten warunek.

Jeśli tabela nie posiada klucza głównego, ale posiada kolumnę z niepowtarzalnym identyfikatorem, należy wykonać:

```
ALTER TABLE tabela ADD CONSTRAINT tabela_pk PRIMARY KEY(identyfikator);
```

Jeśli nie ma takiej kolumny, można wygenerować identyfikatory automatycznie:

```
ALTER TABLE tabela ADD COLUMN identyfikator SERIAL PRIMARY KEY;
```

SQL – Usuwanie danych

Do usuwania danych służy komenda DELETE.

Uwaga!

Komenda DELETE bez podania klauzuli WHERE usunie WSZYSTKIE wiersze w tabeli!

DELETE FROM;

zwykle jednak usuwa się pojedyncze wiersze:

DELETE FROM edycja_tabela WHERE uwagi = 'brak danych';

lub ich grupy:

DELETE FROM edycja_tabela WHERE miesiac = '02';

SQL – Usuwanie danych

W praktyce do czyszczenia całej tabeli, wykorzystuje się komendę **TRUNCATE TABLE**:

```
CREATE TABLE edycja_tabela_copy AS SELECT * FROM edycja_tabela;  
TRUNCATE TABLE edycja_tabela_copy;
```

gdyż jest **szybsza od DELETE bez WHERE**.

Do usuwania danych w programach pgAdmin i QGIS stosuje się te same zasady, co do modyfikacji: **wymagany jest zadeklarowany klucz główny**.

SQL – Wstawianie danych

Do wstawiania danych służy komenda **INSERT**.

Podstawowa składnia:

```
INSERT INTO tabela VALUES ('wartosc1','wartosc2','wartosc3');
```

w takiej sytuacji muszą być podane wartości dla wszystkich kolumn w tabeli.

Możliwe jest też podanie podzbioru kolumn:

```
INSERT INTO tabela(kolumna1, kolumna2) VALUES('wartosc1','wartosc1');
```

```
INSERT INTO edycja_tabela_copy(id,nazwa_stacji,rok,  
miesiac,liczba_dni_z_opadem_sniegu) VALUES(1,'Śnieżka','2024','09','1');
```

Widoki w bazie PostgreSQL

Widok (View) - charakterystyka

Widok (VIEW) – jest specyficzną formą zapytania, które zapisane w bazie pozwala na wygenerowanie, np. w dowolnej chwili tabeli z danymi odpowiadającymi na treść przechowywanego polecenia. Za każdym razem gdy użyty zostanie widok, zapytanie jest ponownie wykonywane, zatem dane prezentowane w widoku zawsze są aktualne

Przykładowe zapytanie tworzące widok:

```
CREATE VIEW pomniki_przyrody_dzialki AS SELECT a.kodinspire, a.typ, a.obiekt, a.nazwa,  
a.data_utwor, b.identyfikator FROM gdos_pomniki_przyrody_p a,  
egib_dzialki_ewidencyjne b WHERE ST_INTERSECTS(a.geom, b.geom) AND a.obiekt NOT IN  
('drzewo','krzew');
```

Przykładowe zapytanie wywołujące widok:

```
SELECT * FROM pomniki_przyrody_dzialki;
```

Widok (View) – tworzenie na przykładzie QGIS

Zarządzanie bazami danych

Baza danych Tabela

Importuj warstwę/plik... Eksportuj do pliku...

Dostawcy algorytmów

- GeoPackage
- Oracle Spatial
- PostGIS
- Spatialite
- Warstwy wirtualne

Informacje Tabela Podgląd Zapytanie (SZKOLENIE) X

Zapisane zapytanie Name Zapisz Delete Wczytaj plik Zapisz jako plik

```
1 CREATE VIEW pomniki_przyrody_dzialki2 AS
2 SELECT a.kodinspire, a.typ, a.obiekt, a.nazwa, a.data_utwor, b.identyfikator
3 FROM gdos_pomniki_przyrody_p a, egib_dzialki_ewidencyjne b
4 WHERE ST_INTERSECTS(a.geom, b.geom)
5 AND a.obiekt NOT IN ('drzewo', 'krzew');
```

Uruchom Utwórz widok Wyczyść Historia zapytań

Wczytaj jako nową warstwę

Anuluj

Widok (View) – działanie na przykładzie pgAdmin

The screenshot shows the pgAdmin interface. On the left, a tree view displays the database structure, with 'pomniki_przyrody_dzialki2' selected under the 'Views (7)' folder. The main window shows a SQL query editor with the following query:

```
1 select * from pomniki_przyrody_dzialki2;
2
```

Below the query editor, the 'Data Output' tab is active, displaying a table with 7 rows of data. The table columns are: kodinspire, typ, obiekt, nazwa, data_utwor, and identyfikator.

	kodinspire character varying (254)	typ character varying (254)	obiekt character varying (254)	nazwa character varying (254)	data_utwor date	identyfikator character varying (60)
1	PL.ZIPOP.1393.PP.0208...	jednoobiektowy	skałka	Czartowski Kamień	2008-08-08	020803_1.0001.105/125
2	PL.ZIPOP.1393.PP.0208...	jednoobiektowy	skałka	Diabelskie Głazy	2008-08-08	020810_5.0006.406/120
3	PL.ZIPOP.1393.PP.0208...	jednoobiektowy	źródło	[null]	2008-08-08	020807_2.0035.194
4	PL.ZIPOP.1393.PP.0208...	jednoobiektowy	skałka	Skałki pasterskie	2008-08-08	020806_5.0005.226/1
5	PL.ZIPOP.1393.PP.0208...	jednoobiektowy	jaskinia	Solna Jama	2008-08-08	020810_5.0006.405/121
6	PL.ZIPOP.1393.PP.0208...	jednoobiektowy	skałka	[null]	2008-08-08	020811_2.0007.973/3
7	PL.ZIPOP.1393.PP.0208...	jednoobiektowy	jaskinia	Grota Radochowska	2008-08-08	020808_5.0009.166/1

Dziękuję za uwagę!