



Pomorska Fundacja
Inicjatyw Gospodarczych

Relacyjne i nierelacyjne bazy danych cz.1

Michał Szymański

www.pfig.org.pl

Wprowadzenie



Kto tworzył zapytania SQL ?
Kto tworzył struktury DB?

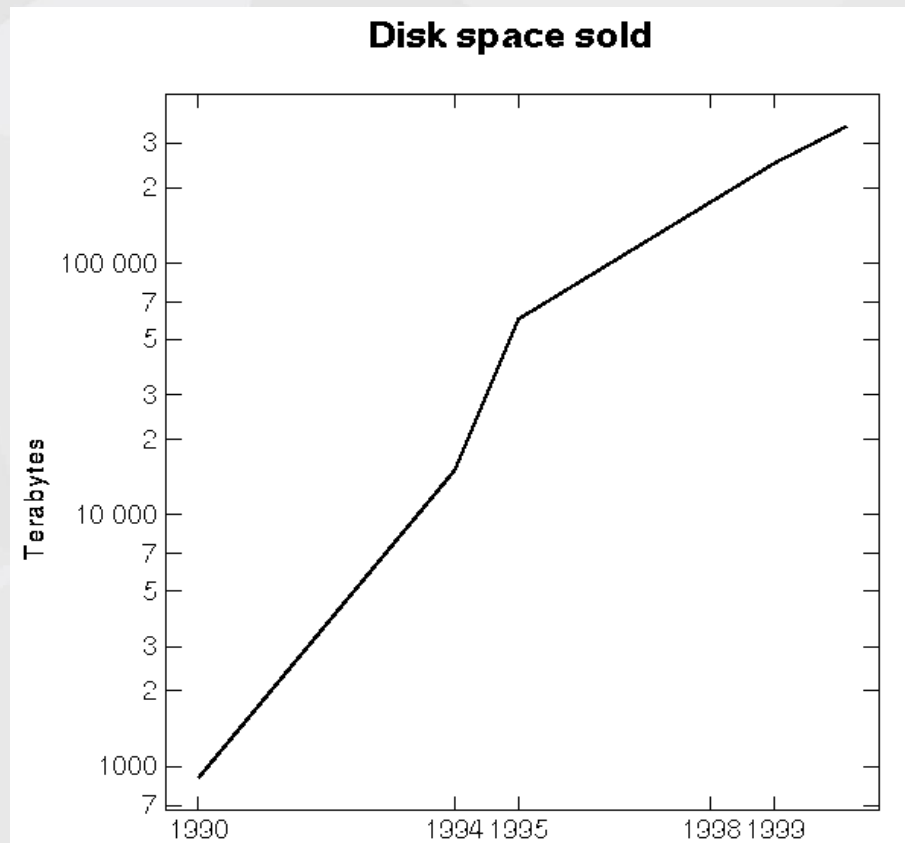


Wiek informacji

Ilość danych na świecie

(1EB=10¹⁸=1 milion TB)

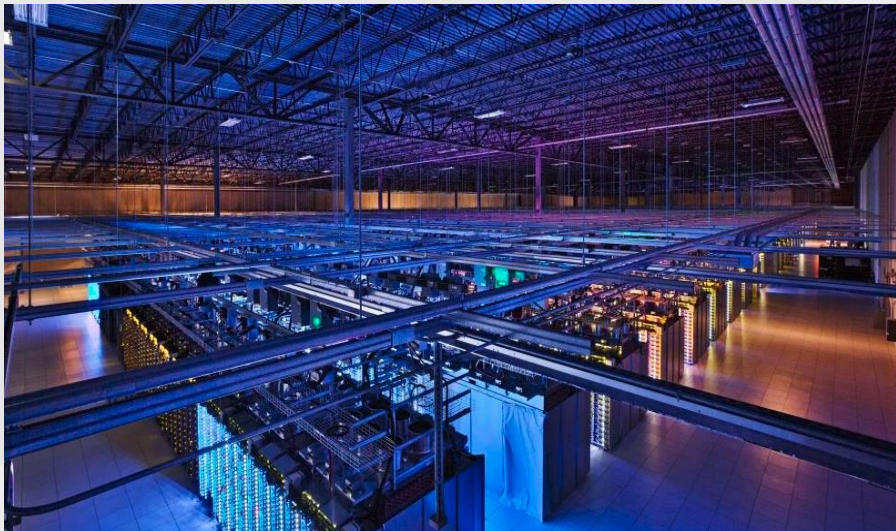
- 1986 → 2.6 EB
- 1993 → 15.8 EB
- 2000 → 54.5 EB
- 2007 → 295 EB
- 2025 → 25000 EB ?



Wiek informacji

- New York Stock Exchange tworzy około 1TB/dzień.
- Facebook dodawane 300mln zdjęć na dzień
- Large Hadron Collider koło Genewy produkuje 25GB/s
- Największa baza danych SQL (PostgreSQL) - Yahoo 2PB = 2048 TB

Wiek informacji



Czego się nauczyć

- Projektować bazy danych
- Dokonywać różnych operacji na bazach danych

Czego na kursie nie będzie

- Nie będziemy skupiali się na aspektach teoretycznych
- Nie będziemy mówili o specyficznych rozwiązaniach dostawców baz danych



Parę słów o prowadzącym

- Skończyłem studia na Wydz. Elektrycznym od 18 lat pracy w sektorze IT
- Pracowałem w Lufthansa Systems, GE Money Bank, Freeconet S.A, Apator aktualnie pracuję w Finastra
- Zajmowałem stanowiska - programisty, projektanta, kierownika projektów i szefa działu IT
- Projektowanie oprogramowania używanego w systemach wbudowanych, lotnictwie, bankowości i telekomunikacji
- Jestem współtwórcą platformy telekomunikacyjnej Carrier-eX – Freeconet



<https://www.linkedin.com/in/szymanskim/>



Baza danych

- Dane żyją zdecydowanie dłużej niż pojedyncze operacje
- Przechowują terabajty danych – dużo więcej niż ma przeciętny komputer pamięci RAM
- Przechowywane dane są bezpieczne – mowa o sprzęcie i oprogramowaniu
- Do danych w DBMS ma dostęp wielu użytkowników
- Dostęp do danych musi być łatwy
- Pobieranie i zmiany danych jest wydajne
- Niezawodne



Główne pojęcia

- Forma przechowywania danych – rekordy, XML, grafy
- Schematy i dane – odpowiednik typów i zmiennych
- DDL – Data definition language – pozwala tworzyć dane
- DML – Data manipulation language – pozwala modyfikować dane



Model relacyjny



Model relacyjny

Użytkownik

ID	Imię	Nazwisko	Data założenia
1	Jan	Kowalski	2014-01-01
2	Jan	Nowak	2014-02-01
3	Tomasz	NULL	2014-02-11

Połączenie

ID	Data rozpoczęcia	Czas trwania	Koszt	Stawka
1	2014-01-01 20:00	60	0.22	0.22
1	2014-01-01 11:23	120	0.40	0.20
3	2014-01-01 23:33	90	0.30	20



Model relacyjny

- Bardzo popularny – większość narzędzi bazuje na modelu relacyjnym
- Funkcjonuje od paru dekad , czyli sprawdzony
- Istnieje podbudowę teoretyczną
- Istnieją efektywne implementacje



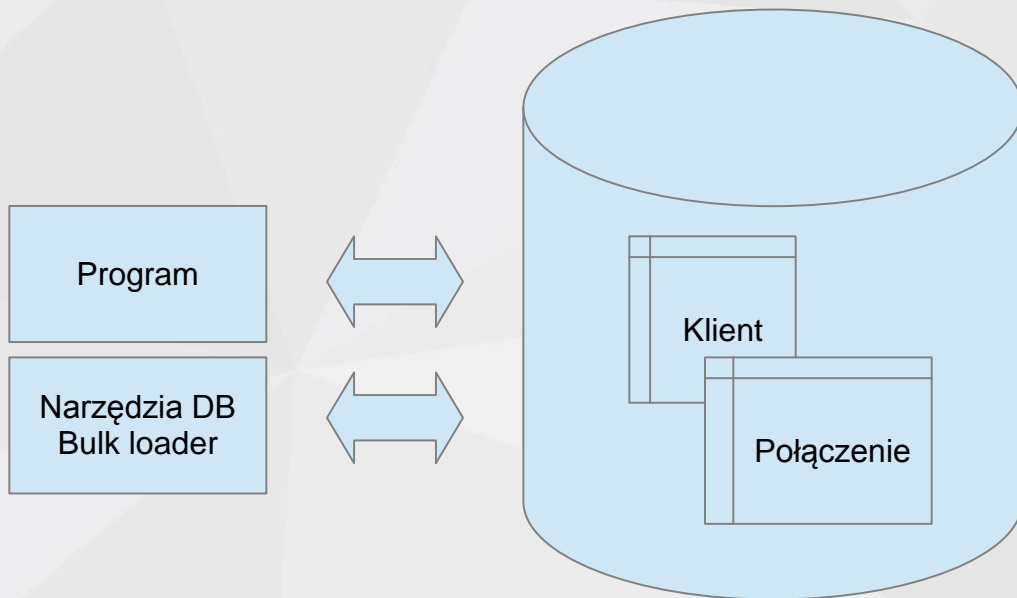
Przykłady danych...ćwiczenia

- Dane przechowujące przez instytucje finansowe – stany kont, operacje
- Różnego rodzaju spisy – katalog książek w bibliotekach
- Portale społecznościowe – informacje o klientach i usługach
- .. wasze przykłady..



Bazy danych – użycie

- Stworzenie struktur
- Zasilenie danymi
- Modyfikacja /
wyszukiwanie danych



Zapytania - DML

- Jakich mamy klientów?
- Jacy klienci zostali założeni w 2 lutego?
- Jakie połączenia zrobił klient o ID=1?
- Chciałbym skorygować nazwisko klienta o ID=3
- Chciałbym dodać nowego klienta



Język SQL

SQL – standard pozwalający na manipulowanie danymi w bazie danych

Przykład:

```
SELECT * FROM klient WHERE ID=2
```

```
UPDATE TABLE klient SET nazwisko='Kowal' WHERE ID=2
```



Dlaczego SQL

- Przyjęty jako standard
- Występuje w większości baz danych
- Prosty w użyciu
- Dostępne narzędzia do tworzenia, uruchamiania i analizy
- Oparty jest na modelu relacyjnym



Instalacja MySQL

<https://dev.mysql.com/downloads/windows/installer/5.7.html>

MySQL Workbench

<https://dev.mysql.com/downloads/workbench/>

(wymaga pobrania pakietów z Microsoft)



SQL - SELECT

```
select C1,C2,C3...  
from T1,T2...  
where c1>0
```



SQL – SELECT, podstawy

SELECT imie,nazwisko FROM klient

SELECT * FROM klient

SELECT * FROM klient WHERE nazwisko='kowalski'



SQL – Przykład danych

Serwis pozwalający użytkownikom na ocenę książek.

klient (klient_id, imie, imie, nazwisko, rok_urodzenia, plec, data_zalozenia)

ksiazka (ksiazka_id, rodzaj_ksiazki, autor_imie, autor_nazwisko, tytul, min_wiek)

ocena_ksiazki (id_ksiazka, id_klient, ocena)

rodzaj_ksiazki (rodzaj)

recenzja (recenzja_id, id_ksiazki, recenzja_tekst)

statystyka_klient (id_klient, liczba_przeczytanych_ksiazek)



SQL – SELECT, operatory

```
SELECT * FROM klient WHERE nazwisko='Kowalski'
```

```
SELECT * FROM klient WHERE nazwisko<>'Kowalski'
```

```
SELECT * FROM klient WHERE nazwisko LIKE 'Kow%'
```

```
SELECT * FROM klient WHERE nazwisko='Kowalski' OR  
nazwisko='Szymański'
```

```
SELECT * FROM klient WHERE nazwisko LIKE 'S%' AND  
rok_urodzenia>'1960-01-01'
```



SQL – SELECT, sortowanie

.... ORDER BY <kolumna> ASC - sortowanie rosnące

.... ORDER BY <kolumna> DESC – sortowanie malejące



SQL – SELECT, table variables

```
SELECT * FROM ksiazka K , recenzja R  
WHERE K.ksiazka_id=R.id_ksiazka
```

```
SELECT * FROM osoba rodzic , osoba dziecko  
WHERE rodzic.osoba_id= dziecko.id_rodzic
```



SQL – SELECT – UNION, INTERSECT, EXCEPT

UNION

klient				klient_archiwalny					
ID	Imie			ID	Imie			ID	Imie
1	Tomasz			4	Olga			1	Tomasz
2	Jan			5	Jan			2	Jan
								4	Olga
								5	Jan



SQL – SELECT – UNION, INTERSECT, EXCEPT

Tej konstrukcji nie ma MySQL !!

INTERSECT

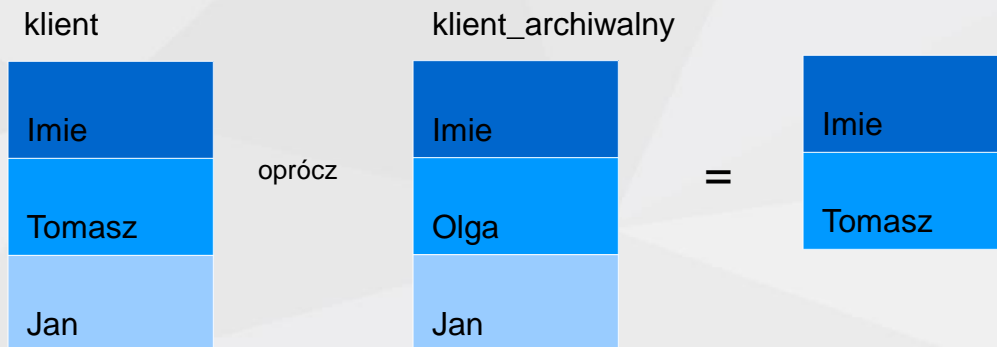
klient		klient_archiwalny		
Imie		Imie		Imie
Tomasz	AND	Olga	=	Jan
Jan		Jan		



SQL – SELECT – UNION, INTERSECT, EXCEPT

Tej konstrukcji nie ma MySQL !!

EXCEPT



SQL – SELECT, złączenia

```
SELECT * FROM ksiazka, recenzja  
WHERE ksiazka.ksiazka_id=recenzja.id_ksiazka
```

ksiazka

ksiazka_id=id_ksiazka



recenzja



SQL – SELECT, DISTINCT

DISTINCT – usuwa duplikaty

```
SELECT DISTINCT autor_imie FROM ksiazka
```

