

Bazy danych 2022

na podstawie slajdów Przemysławy Kanarek

2 marca 2022

BD i SZBD

Baza danych (BD,DB) — zbiór danych zawierających zarówno informacje rzeczowe, jak i strukturę tych informacji; zazwyczaj **duży, długotrwały, dostępny dla wielu użytkowników** na różne sposoby;

System Zarządzania Bazami Danych (SZBD, DBMS) — oprogramowanie pozwalające **definiować strukturę bazy danych, gromadzić dane w bazie i je efektywnie udostępniać** (Oracle, PostgreSQL, MySQL, SQL Server, . . .);

ACID — atomowość (Atomic), poprawność (Consistent), niezależność (Independent), trwałość (Durable).

Security —kontrola dostępu

Języki baz danych

Diagramy E-R, UML — projektowanie konceptualne (modelowanie);

Język definiowania danych (DDL) — polecenia tworzenia elementów struktury bazy danych;

Język zapytań (query language) — polecenia wyszukiwania danych;

Język modyfikacji danych (DML) — polecenia dodawania, usuwania i modyfikacji danych;

Język aplikacji — język służący do pisania aplikacji odwołujących się do bazy danych.

SQL (Structured Query Language) zawiera DDL, query language oraz DML i jest zaimplementowany praktycznie we wszystkich relacyjnych SZBD (*dialekty SQL*).

Języki aplikacji mogą to być języki programowania, dla których zaprogramowano biblioteki dostępu do bazy danych, własny język programowania SZBD stanowiący rozszerzenie SQL lub programistycznego SQL.

Object-relational mapping (ORM) — pozwala na serializację i deserializację obiektów do bazy danych z poziomu języka programowania, konwertuje niezgodne typy, pozwala abstrahować od SZBD. Zmniejsza ilość kodu. *Object-relational impedance mismatch*. — do omówienia na innych zajęciach (POO).

Po co nam bazy danych? Do czego?

- Pozwalają na abstrakcję (nieważne co w środku!),
- Uwalniają twórcę systemu od implementowania warstwy danych w każdym projekcie,
- Zapewniają integralność danych (ACID) i Security,
- Pozwalają na optymalizacje.

- OLTP (duża liczba małych transakcji, scale-independence),
- OLAP (analiza danych, zapytania przetwarzają duże dane, ML-in-database).

Plan wykładu

Będzie o:

- 1 **Model relacyjny teoretycznie:** elementy składowe modelu, języki zapytań, zapytania koniunkcyjne, rekursja, postaci normalne (BCNF, 3NF, 4NF).
- 2 **Model relacyjny praktycznie:** zapytania SQL, projektowanie baz danych oraz diagramy E-R i UML, język definicji danych SQL.
- 3 **Systemy zarządzania relacyjnymi bazami danych:** przetwarzanie zapytań, transakcje i wielodostęp, bezpieczeństwo danych.

Na innych przedmiotach:

- Bazy grafowe
- MapReduce, NoSQL
- DataMining i hurtownie danych, bazy analityczne, kolumnowe
- Budowa DBMS (Bazy danych 2)
- Bazy geograficzne (PostGIS)
- Przetwarzanie strumieniowe, Bazy rozproszone

Relacyjne bazy danych?

Relacyjne bazy danych



ORACLE[®]



Relacyjne bazy danych?

"n.id"	"n.screen_name"	"n.followers_count"	"n.friends_count"	"n.location"
2531159968	"traceyhappymom"	3696	3353	"Washington, DC"
100345056	"SCOTTGOHARD"	1053	1055	"still #Block\Corner\street"
247165706	"Beckster319"	650	896	"Chicago, IL"
249538061	"skatewake1994"	44	154	""
449689677	"KadirovRussia"	94773	7	""
471868887	"MargoSavazh"	23305	8021	"Санкт-Петербург, Россия"
1039581360	"darknally"	22	40	"Amerika"
1510488662	"lagonehoe"	3080	2369	"USA"
1513801268	"YouJustCtrlC"	2760	2700	"USA"
1517678892	"MrMoraan"	879	758	"Philadelphia, PA"
1518857420	"NoJonathonNo"	789	440	"USA"

Relacyjne bazy danych?

"n.id"	"n.screen_name"	"n.followers_count"	"n.friends_count"	"n.location"
2531159968	"traceyhappymom"	3696	3353	"Washington, DC"
100345056	"SCOTTGOHARD"	1053	1055	"still #BlockCornerstreet"
247165706	"Beckster10"	1000	1000	"Chicago, IL"

"n.tweet_id"	"n.user_key"	"n.created_str"	"text"
836227891897651201	"kathiemrr"	"2017-02-27 14:54:00"	"ThingsDoneByMist"
765198948239810561	"traceyhappymom"	"2016-08-15 14:50:20"	"T @mc_derpin: #I"
616002306572746752	"evewebster373"	"2015-06-30 21:56:09"	"T @dnataconis: F"
77693302926147584	"blacktolive"	"2016-09-16 08:04:48"	"men! #blacklives"
777594647875059712	"jacuelinisbest"	"2016-09-18 19:46:25"	"T @NahBabyNah: T"
718040061649031168	"judelambertusa"	"2016-04-07 11:37:45"	"T @mcicero10: #E-
785586729579196416	"carriethornton"	"2016-10-10 21:04:06"	"T @ItsJustJaynie-
664592113775022084	"johnbranchh"	"2015-11-11 23:54:42"	"TodayCleveland "
782408661389840384	"march_for_trump"	"2016-10-02 02:35:35"	"NickTomawBRE Hi,
811580868573691904	"puredavie"	"2016-12-21 14:35:32"	"hat. Is. A. Resc
800655998335774720	"daileyjaddon"	"2016-11-21 11:04:00"	"ifetime movie
822266545728585728	"evagreen69"	"2017-01-20 02:16:36"	"T @Conservatexia
795661161282091776	"cassiewelch"	"2016-11-07 16:16:18"	"T @HillaryClintc
777859822679158784	"_nickluna_"	"2016-09-19 13:20:08"	"T @leonpui: Hil

Relacyjne bazy danych?

"n.id"	"n.screen_name"	"n.followers_count"	"n.friends_count"	"n.location"		
2531159968	"traceyhappymom"	3696	3353	"Washington, DC"		
180345056	"SCOTTGOHARD"	1853	1855	"still #Block\Corner\street"		
247165786	"Beckster10"	1853	1855	"Chicago, IL"		
249538861	"skatewak"					
		"n.tweet_id"	"n.user_key"	"n.created_str"	"text"	
449689677	"Kadirov"	836227891897651201	"kathiemrr"	"2017-02-27 14:54:00"	"ThingsDoneByMist"	
47		39810561	"traceyhappymom"	"2016-08-15 14:50:20"	"T @mc_derpin: #I"	
10	"n.name"	"count"				
10	"Trump "	8029	72746752	"evewebster373"	"2015-06-30 21:56:09"	"T @dnataconis: F"
15	"CNN "	1263	26147584	"blacktolive"	"2016-09-16 08:04:48"	"men! #blacklives"
15	"GOP "	914	75059712	"jacquelinisbest"	"2016-09-18 19:46:25"	"T @NahBabyNah: T"
15	"FBI "	762	49031168	"judelambertusa"	"2016-04-07 11:37:45"	"T @mcicero10: #E-
15	"TRUMP "	675	79196416	"carriethornton"	"2016-10-10 21:04:06"	"T @ItsJustJaynie-
	"Clinton "	672	75022084	"johnbranchh"	"2015-11-11 23:54:42"	"TodayCleveland "
	"ISIS "	541	89840384	"march_for_trump"	"2016-10-02 02:35:35"	"NickTomaWBRE Hi,
	"MAGA "	479	73691904	"puredavie"	"2016-12-21 14:35:32"	"hat. Is. A. Resc
	"YouTube "	411	35774720	"daileyjaddon"	"2016-11-21 11:04:00"	"lifetime movie
	"Twitter "	403	28585728	"evagreen69"	"2017-01-20 02:16:36"	"T @Conservatexia
	"Wikileaks "	377	82891776	"cassiewelch"	"2016-11-07 16:16:18"	"T @HillaryClintc
	"Obama "	352	79158784	"_nickluna_"	"2016-09-19 13:20:08"	"T @leonpui: Hil
	"HRC "	340				
	"America "	335				
	"Wikileaks "	321				

Relacyjne bazy danych?

"n.id"	"n.screen_name"	"n.followers_count"	"n.friends_count"	"n.location"
253115998	"traceyhappymon"	3696	3353	"Washington, DC"
100345056	"SCOTTGOHARD"	1053	1055	"still #BlockCornerstreet"
247165706	"Beckster10"	1004		"Chicago, IL"
"n.tweet_id"	"n.user_key"	"n.created_str"	"text"	
83622789189765520	"kathierf"	"2017-02-27 14:54:00"	"ThingsDoneByMist"	
39810561	"traceyhappymon"	"2016-08-15 14:50:20"	"T @mc_derpin: #I"	
"n.name"	"count"	"n.created_str"	"text"	
47	72746752	"2015-06-30 21:56:09"	"T @dnataconis: F"	
10	26147584	"2016-09-16 08:04:48"	"men! #blacklives"	
15	75059712	"2016-09-18 19:46:25"	"T @NahBabyNah: T"	
15	49031168	"2016-04-07 11:37:45"	"T @mcicero10: #E-	
15	79196416	"2016-10-10 21:04:06"	"T @ItsJustJaynie-	
15	75022084	"2015-11-11 23:54:42"	"TodayCleveland "	
"Clinton "	672	89840384	"march_for_trump" "NickTomawBRE Hi,	
"ISIS "	541	73691904	"puredavie" "hat. Is. A. Resc	
"MAGA "	479	35774720	"daileyjaddon" "ifetime movie	
"YouTube "	411	28585728	"evagreen69" "T @Conservatexia	
"Twitter "	403	82091776	"cassiewelch" "T @HillaryClintc	
"Wikileaks "	377	79158784	"_nickluna_" "T @leonpui: Hil	
"Obama "	352			
"HRC "	340			
"America "	335			
"Wikileaks "	321			

Relacyjne bazy danych?

Schematy - encje, związki, normalizacja

```
Szkolenie(id int, temat text, trener_nazwisko text, trener_mail text,  
          trener_opis text, data date, sluchacze text ARRAY, trudnosc int)
```

Relacyjne bazy danych?

Schematy - encje, związki, normalizacja

```
Szkolenie(id int, temat text, trener_nazwisko text, trener_mail text,  
         trener_opis text, data date, sluchacze text ARRAY, trudnosc int)
```



```
Szkolenie(id int, temat text, data date, trudnosc int)  
Trener(id int, nazwisko text, mail text, opis text)  
Sluchacz(id int, nazwisko text)
```

Relacyjne bazy danych?

Schematy - encje, związki, normalizacja

```
Szkolenie(id int, temat text, trener_nazwisko text, trener_mail text,  
         trener_opis text, data date, sluchacze text ARRAY, trudnosc int)
```



```
Szkolenie(id int, temat text, data date, trudnosc int)  
Trener(id int, nazwisko text, mail text, opis text)  
Sluchacz(id int, nazwisko text)
```

Relacyjne bazy danych?

Schematy - encje, związki, normalizacja

Szkolenie(id int, temat text, trener_nazwisko text, trener_mail text, trener_opis text, data date, sluchacze text ARRAY, trudnosc int)

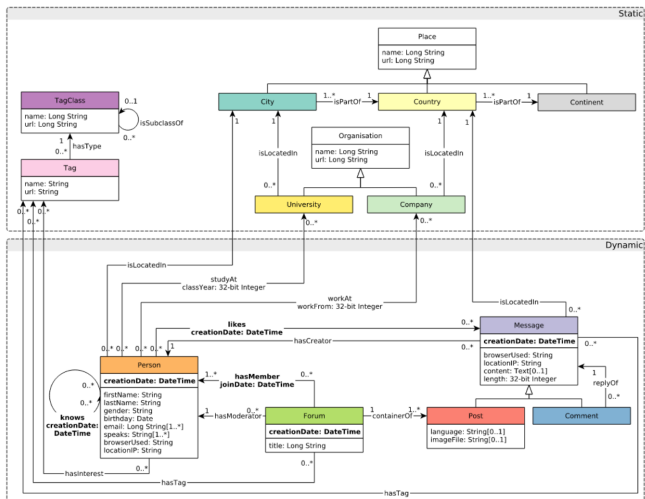


Szkolenie(id int, temat text, data date, trudnosc int)
 Trener(id int, nazwisko text, mail text, opis text)
 Sluchacz(id int, nazwisko text)



Szkolenie(id int, temat text, trudnosc int)
 Edycja(id int, szkolenie int, data date)
 Trener(id int, nazwisko text, mail text, opis text)
 Sluchacz(id int, nazwisko text)

Relacyjne bazy danych?



LDBC
Benchmark

Dlaczego relacyjne bazy danych?

Jak do tego doszło?

- **Lata 50-60-te:** powstaje model hierarchiczny (IMS) i sieciowy (CODASYL).
- **Lata 70-te:** Codd proponuje model relacyjny i powstają pierwsze relacyjne SZBD: Ingres (Ingres Corp, PostgreSQL, Sybase, MS SQL Server, ...) oraz System R (DB2, Oracle,...).
- **Lata 90-te:** model relacyjny rządzi, ale... staje się za ciasny, bo nie zawsze dobrze sobie radzi ze skomplikowanymi danymi, specyficznym przetwarzaniem danych i funkcjonowaniem w nowych środowiskach (chmura, urządzenia mobilne, dyski SSD).

Jack of all trades, master of none

XXI wiek

- NoSQL → NewSQL (ACID powraca!), Polyglot persistence
- Zasady ACID (czasem) nie są kluczowe: bazy sieci społecznościowych (mniejsza niezawodność), informacji (opóźniona spójność), brak powtarzalnej struktury, ...
- Powstają nowe modele danych: obiektowy, XML, JSON, *wide column stores* (BigTable: row&column keys, timestamp→bytes),...
- Powstają specjalistyczne systemy baz danych: temporalne, probabilistyczne, geograficzne, tekstowe, grafowe,...
- Bazy są większe niż kiedykolwiek wcześniej: skalowalność, przetwarzanie analityczne (hurtownie danych), przetwarzanie strumieniowe,...
- Autonomiczne bazy danych (zarządzane przez AI), Bazy w chmurze,

NoSQL: key-values stores, document stores

- brak schematu / elastyczny schemat
- duża wydajność kosztem organizacji danych wg pojedynczego klucza
- brak łatwej możliwości modelowania związków pomiędzy danymi (dodawanie id jednego dokument jako wartość w drugim...)
- przegrupowanie danych, joiny -> obliczenia po stronie aplikacji
- typowe zastosowania: profile użytkownika (web, gry, ...), koszyk zakupowy, rekomendacje, ogłoszenia, cache

What's Really New with NewSQL?

<https://db.cs.cmu.edu/papers/2016/pavlo-newsq1-sigmodrec2016.pdf>

BTW <https://sigmodrecord.org/sigmod-record-current-issue/>


[Current Issue](#)
[Previous Issues](#)
[Info for Authors](#)
[Record Editors](#)
[FAQ](#)
[Credits](#)
[About SIGMOD](#)


Home / SIGMOD Record – Current Issue December 2020 (Vol. 49, No. 4)

SIGMOD Record – Current Issue December 2020 (Vol. 49, No. 4)

ENTIRE ISSUE – 32 pages (in: [PDF](#))

31 Downloads

Back Of Front Cover (also in: [PDF](#))

Table Of Contents (also in: [PDF](#))

SIGMOD Officers, Committees and Awardees (also in: [PDF](#))

Editor's Notes (in: [PDF](#))

Vision Article

Agora: Bringing Together Datasets, Algorithms, Models and More in a Unified Ecosystem [Vision]

Jonas Traub, Zoi Kaoudi, Jorge-Arnulfo Quiané-Ruiz, Volker Markl

Available in: [PDF](#)

106 Downloads

[Click here to like and comment this article](#)

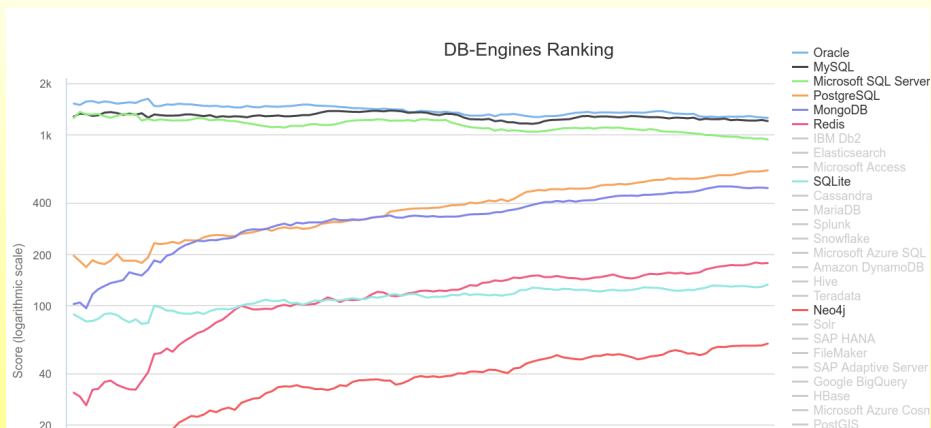
Categories

[Announcements](#)
[Database Principles](#)
[Distinguished Profiles](#)
[Industry Perspectives](#)
[Messages](#)
[Open Forum](#)
[Reports](#)
[Research Articles](#)

Popularność: <https://db-engines.com/en/ranking>

Rank			DBMS	Database Model	Score
Mar 2022	Feb 2022	Mar 2021			
1.	1.	1.	Oracle +	Relational, Multi-model ⓘ	1251.32
2.	2.	2.	MySQL +	Relational, Multi-model ⓘ	1198.23
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model ⓘ	933.78
4.	4.	4.	PostgreSQL + 🗨️	Relational, Multi-model ⓘ	616.93
5.	5.	5.	MongoDB +	Document, Multi-model ⓘ	485.66
6.	6.	↑ 7.	Redis +	Key-value, Multi-model ⓘ	176.76
7.	7.	↓ 6.	IBM Db2	Relational, Multi-model ⓘ	162.15
8.	8.	8.	Elasticsearch	Search engine, Multi-model ⓘ	159.95
9.	9.	↑ 10.	Microsoft Access	Relational	135.43
10.	10.	↓ 9.	SQLite +	Relational	132.18
11.	11.	11.	Cassandra +	Wide column	122.14
12.	12.	12.	MariaDB +	Relational, Multi-model ⓘ	108.31
13.	13.	13.	Splunk	Search engine	95.36
14.	↑ 15.	↑ 30.	Snowflake +	Relational	86.23
15.	↓ 14.	↑ 16.	Microsoft Azure SQL Database	Relational, Multi-model ⓘ	84.68
16.	↑ 17.	↑ 17.	Amazon DynamoDB +	Multi-model ⓘ	81.80
17.	↓ 16.	↓ 14.	Hive +	Relational	81.22

Trendy: https://db-engines.com/en/ranking_trend



Ostrzeżenie:

<http://avid.cs.umass.edu/courses/691LL/f2006/papers/SH05.pdf>

What Goes Around Comes Around

Michael Stonebraker
Joseph M. Hellerstein

Abstract

This paper provides a summary of 35 years of data model proposals, grouped into 9 different eras. We discuss the proposals of each era, and show that there are only a few basic data modeling ideas, and most have been around a long time. Later proposals inevitably bear a strong resemblance to certain earlier proposals. Hence, it is a worthwhile exercise to study previous proposals.

Ćwiczenia i pracownia

Będziemy się uczyć:

- 1 Rozumieć model relacyjny (algebra relacji i rachunki relacyjne, postaci normalne);
- 2 Korzystać z gotowej bazy danych — wyszukiwać w niej informacje, odpowiedzi na interesujące nas pytania (język SQL);
- 3 Konstruować poprawne bazy danych dla zagadnień rzeczywistych — projektować bazy (modelować) i na podstawie projektów definiować elementy baz danych;
- 4 Tworzyć aplikacje korzystające z bazy danych.

Materiały i informacje: <https://skos.ii.uni.wroc.pl/course/view.php?id=463>
— kurs Bazy Danych 2022.

Literatura

- Jeffrey D. Ullman, Jennifer Widom, Podstawowy Kurs Systemów Baz Danych, WNT, Warszawa 1999;
- Garcia-Molina H., Ullman J.D., Widom J., Implementacja systemów baz danych, WNT, 2003 (seria: Klasyka Informatyki);
- Garcia-Molina H., Ullman J.D., Widom J., Database Systems: The Complete Book (suma dwóch powyższych pozycji);
- Thomas Connolly, Carolyn Begg, Database Systems, Addison Wesley 2002, także po polsku: ReadMe 2004;
- Date C. J., An Introduction to Database System, vol. II, Addison-Wesley Pub. Comp., również WNT W-wa, (seria: Klasyka Informatyki), 2000;
- R. Ramakrishnan, J. Gehrke, Database Management Systems, 2nd edition, WCB/McGraw-Hill, 2001. Jest też wydanie 3-cie.

Elementy modelu

Relacja (tabela) — jedyna struktura dla danych w modelu; ma ustaloną liczbę kolumn, w które można wpisywać wartości ustalonego typu i dowolną liczbę wierszy.

Więzy (warunki poprawności, warunki spójności) — dane wpisywane do tabel muszą spełniać zdefiniowane warunki: typ danych, zakres,...

Baza danych — zbiór tabel z danymi spełniającymi nałożone na nie więzy.

Język zapytań (*query language*) — algebra relacji, relacyjny rachunek krotek i relacyjny rachunek dziedzin — formalne języki pozwalające wyszukać w relacjach określoną informację.

Relacja, czyli tabela

Osoba

Nazwisko : varchar(20)	PESEL : char(11)	dataUr : date
...		
Abacki	80121304455	'20-02-1980'
...		

Mieszkanie

PESEL : char(11)	Adres : varchar(50)	Metraż : real
...		
80121304455	Elk, Kwiatowa 100	60,2
80121304455	Poznań, Szeroka 10/2	30,2
NULL	Elk, Kwiatowa 102	64,2
...		

Elementy relacji

- *Atrybut* — nazwa kolumny;
- *Dziedzina* — typ danych;
- *Krotność (arność)* — liczba atrybutów;
- *Krotka (wiersz)* — element relacji;
- *Atrybuty krotki* — *Osoba*[3] lub *Mieszkanie*.Adres;
- *Schemat relacji* — nazwa relacji, nazwy i typy kolumn;
- *Stan relacji* to zawarte w niej krotki.

Notacja matematyczna

Dla atrybutów A_1, \dots, A_k i związanych z nimi dziedzin D_1, \dots, D_k relacja R ma:

schemat $R = A_1 \dots A_k$ lub $R(A_1, \dots, A_k)$,

arność k ,

stan $r \subseteq D_1 \times \dots \times D_k$,

krotki $(v_1, v_2, \dots, v_k) \in r$.

Relacyjna baza danych (schemat i stan) to zbiór relacji o różnych nazwach.

W przykładzie:

- Osoba(Nazwisko,PESEL,dataUr),
- Mieszkanie(PESEL,Adres,Metraż)

Wartość pusta (NULL)

PESEL : char(11)	Adres : varchar(50)	Metraż : real	
...			
NULL	Poznań, Szeroka 10/12	64,2	← t_1
NULL	Elk, Kwiatowa 102	64,2	← t_2
...			

- t_1 .PESEL = t_2 .PESEL UNKNOWN!!!
- t_1 .PESEL = t_1 .Adres UNKNOWN!!!
- t_1 .Metraż = t_2 .Metraż TRUE
- t_1 .Adres = t_2 .Adres FALSE
- t_1 .PESEL = *NULL* UNKNOWN!!!
- t_1 .PESEL = '' UNKNOWN!!!
- t_1 .PESEL IS NULL TRUE
- t_1 .Adres IS NOT NULL TRUE

Klucze

Klucz relacji

Podzbiór atrybutów relacji, których wartości zawsze pozwalają jednoznacznie zidentyfikować krotkę relacji. Oznacza, to że nie dopuszczamy, by w danych znalazły się dwie różne krotki o jednakowych wartościach klucza. Relacja może mieć kilka kluczy:

`Student (indeks, PESEL, Nazwisko, ...)`

Klucz główny

Jeden z kluczy relacji. Zazwyczaj wybieramy ten, według którego najczęściej będziemy wyszukiwać dane z relacji. Pozostałe klucze nazywamy *kandydującymi* lub *alternatywnymi*. Na przykład `indeks` może być kluczem głównym relacji `Student`, a `PESEL` — kluczem alternatywnym.

Klucz z wielu atrybutów

Stosujemy takie rozwiązanie, gdy jeden atrybut nie wystarcza do zidentyfikowania krotki. Na przykład w relacji `Zaliczenie (indeks, kod_przedmiotu, ocena, data)`.

Klucz obcy

Dane w bazie muszą często zostać rozmieszczone w różnych relacjach, pomimo że się ze sobą wiążą. Do połączenia danych z różnych relacji służą **klucze obce**.

Zaliczenie:

indeks	kod_przedm	ocena
123456	BD2011	5.0
123456	SK2011	4.5
654321	BD2011	3.5
999999	BD2012	2.0

Student:

indeks	PESEL	nazwisko
123456	AB123456	Abacka
654321	CD345678	Babacka
987654	DE534343	Cabacka

- Zamieszczony w relacji `Zaliczenie` atrybut `indeks` służy do zidentyfikowania osoby z relacji `Student`.
- W relacji `Student` atrybut `indeks` jest kluczem.
- W relacji `Zaliczenie` atrybut `indeks` może powtarzać się lub być pusty.
- Jeśli `indeks` jest użyty w relacji `Zaliczenie`, to w relacji `Student` powinna występować osoba o tym indeksie (integralność referencyjna).

Więzy — podsumowanie

Więzy kolumnowe — nakładanie ograniczeń na wartość atrybutu: dziedzina, wartość nie pusta (NOT NULL), zakres;

Więzy tabeli — własność klucza, unikalność w ramach tabeli;

Więzy między tabelami — własność klucza obcego;

Inne więzy ogólne — bardziej złożone warunki (np. maksymalnie dwa podejścia do przedmiotu w sesji, dostęp do wybranych przedmiotów dla studentów określonej sekcji, limit liczby osób zapisanych na zajęcia itp.)

Języki

Język definiowania danych

Musi pozwolić opisać schematy relacji oraz więzy (warunki poprawności) danych.

Język manipulacji danymi

Pozwala dodawać/usuwać krotki z relacji.

Języki zapytań

Mamy trzy propozycje:

- algebra relacji** — kilka operacji pozwalających działać na relacjach jako na zbiorach;
- relacyjny rachunek dziedzin** — język wykorzystujący formuły logiczne do opisu wartości, które należy znaleźć;
- relacyjny rachunek krotek** — język wykorzystujący formuły logiczne do opisu krotek, które należy znaleźć;

Standard: **SQL**

Różne podejścia do budowania zapytań

- `SELECT indeks, adres FROM Student`
- `{(indeks, adres) | ∃ nazwisko Student(indeks, nazwisko, adres)}`
- $\pi_{\{\text{indeks, adres}\}}(\text{Student})$
- `for krotka in Student`
 `print (krotka.indeks, krotka.adres)`

Algebra relacji

Argumentami są całe relacje (tabele), na których wykonujemy operacje.

Zestaw operacji jest nieliczny: rzutowanie, selekcja, iloczyn kartezjański, suma, różnica i przemianowanie

Zapytanie to poprawne wyrażenie algebry relacji, a odpowiedź, to wartość tego wyrażenia obliczona na podstawie aktualnego stanu bazy danych.

Operacje podstawowe - unarne

Rzut — $\pi_{\alpha}(R)$ zwraca relację o schemacie $\alpha \subseteq attr(R)$ powstałą z obcięcia relacji R do kolumn α . Na przykład $\pi_{nazwisko}(Student)$. Duplikaty są eliminowane.

Selekcja — $\sigma_F(R)$ zwraca krotki wybrane z relacji R spełniające warunek F . Na przykład $\sigma_{Adres='Koszalin'}(Student)$.

Przemianowanie — $\rho_{S(B_1, \dots, B_k)}(R)$ zmienia nazwę relacji R na S i nazwy odpowiednich atrybutów R na B_1, \dots, B_k . Na przykład $\rho_{Osoba(id, nazwisko, miasto)}(\pi_{indeks, nazwisko, adres}(Student))$.

Student

Indeks	Nazwisko	Adres
123456	Abacka	Koszalin
654321	Babacka	Szczecin
765678	Cabacka	Koszalin
234565	Abacka	Legnica

Wynik rzutu na Nazwisko

Nazwisko
Abacka
Babacka
Cabacka
Abacka

Wynik selekcji Adres='Koszalin'

Indeks	Nazwisko	Adres
123456	Abacka	Koszalin
765678	Cabacka	Koszalin

Tabela po przemianowaniu: Osoba

Id	Nazwisko	Miasto
123456	Abacka	Koszalin
654321	Babacka	Szczecin
765678	Cabacka	Koszalin
234565	Abacka	Legnica

Operacje teoriomnogościowe — suma, różnica, przekrój

Suma (\cup), różnica (\setminus), przekrój (\cap) — „zwykłe” operacje na zbiorach; $R \setminus S$ i $R \cup S$ wymagają, by $\text{attr}(R) = \text{attr}(S)$; w praktyce mogą być zastępowane operacjami na **multizbiorach** (dlaczego?).

Dodawane (odejmowane, krojone) relacje muszą mieć zgodne schematy.

StudentII

Indeks	Nazwisko	Adres
123456	Abacka	Koszalin
654321	Babacka	Szczecin
234565	Abacka	Legnica

StudentIM

Indeks	Nazwisko	Adres
012345	Zetowski	Kielce
654321	Babacka	Szczecin

Relacja wynikowa:

Indeks	Nazwisko	Adres	Indeks	Nazwisko	Adres
			123456	Abacka	Koszalin
			654321	Babacka	Szczecin
			234565	Abacka	Legnica
			012345	Zetowski	Kielce
			654321	Babacka	Szczecin
Indeks	Nazwisko	Adres	Indeks	Nazwisko	Adres
123456	Abacka	Koszalin	654321	Babacka	Szczecin
234565	Abacka	Legnica			

Złączenia

Iloczyn kartezjański (\times) — dla relacji o rozłącznych schematach ($attr(R) \cap attr(S) = \emptyset$) $R \times S$ jest relacją o atrybutach $attr(R) \cup attr(S)$ zawierająca krotki $t = rs$, gdzie $r \in R$ i $s \in S$ oraz $t.attr(R) = r$ i $t.attr(S) = s$.

Student

Indeks	Nazwisko	Adres
123456	Abacka	Koszalin
654321	Babacka	Szczecin
234565	Abacka	Legnica

Przedmiot

Kod	Nazwa	Typ
BD	Bazy danych	podst
AM	Analiza mat.	obow

Student \times Przedmiot

Indeks	Nazwisko	Adres	Kod	Nazwa	Typ
123456	Abacka	Koszalin	BD	Bazy danych	podst
654321	Babacka	Szczecin	BD	Bazy danych	podst
234565	Abacka	Legnica	BD	Bazy danych	podst
123456	Abacka	Koszalin	AM	Analiza mat.	obow
654321	Babacka	Szczecin	AM	Analiza mat.	obow
234565	Abacka	Legnica	AM	Analiza mat.	obow

Złączenie naturalne

Złączenie naturalne (\bowtie) Dla relacji R i S *złączeniem naturalnym* $R \bowtie S$ jest relacja o schemacie $attr(R) \cup attr(S)$ zawierająca krotki t , dla których istnieją krotki $r \in R$ i $s \in S$, takie że $r.(attr(R) \cap attr(S)) = s.(attr(R) \cap attr(S))$ oraz $t.attr(R) = r$ i $t.attr(S) = s$.

Student

Indeks	Nazwisko	Adres
654321	Babacka	Szczecin
234565	Abacka	Legnica
123456	Abacka	Koszalin

Ocena

Indeks	Kod	Stopien
654321	BD	5.0
234565	BD	4.5
234565	AM	4.5
012345	AM	3.5

Student \bowtie Ocena

Indeks	Nazwisko	Adres	Kod	Stopien
654321	Babacka	Szczecin	BD	5.0
234565	Abacka	Legnica	BD	4.5
234565	Abacka	Legnica	AM	3.5

Krotki, które nie mają pary, nie wchodzą do wyniku!

Wszystkie operacje algebry relacji

Złączenie θ_F to iloczyn kartezjański połączony z selekcją:

$$R \bowtie_F S = \sigma_F(R \times S)$$

Złączenia zewnętrzne to złączenie naturalne, do którego wyniku dorzuca się krotki, które nie znalazły pary. W polach, które są niewypełnione, wpisywana jest wartość NULL.

Półzłączenia to operacja wybierająca z relacji krotki, które połączyłyby się, gdyby wykonywano złączenie naturalne.

Inne operacje np. iloraz, złączenie lewostronne i prawostronne.

Zapytania budujemy poprawne wyrażenia używając operatorów algebry relacji, nawiasów i stałych.

Wszystkie operacje algebry relacji są wyrażalne za pomocą: $\pi, \sigma, \rho, \times, \cup, \setminus$.

Wszystkie operacje algebry relacji

$\pi, \sigma, \rho, \times, \cup, \setminus, \bowtie$

`\pi, \sigma, \rho, \times, \cup, \setminus, \Join`

Baza do przykładów

- **Student**=(indeks,nazwisko, rok), czyli indeks, nazwisko i rok studiów studenta;
- **Przedmiot**=(nazwa, typ), czyli nazwa i typ przedmiotu;
- **Ocena**=(indeks,przed,data,stop), czyli ocena uzyskana przez studenta za przedmiot wraz z datą wystawienia.

Klucze główne relacji są podkreślone. Dodatkowo w relacji *O* występują klucze obce:

- *O.indeks* odnoszący się do *S.indeks*,
- *O.przed* odnoszący się do *P.nazwa*,
- Czy pola *data* i *stop* w relacji *Ocena* mogą być puste?

Przykłady 1-3

Baza danych

$S = (\underline{\text{indeks}}, \text{nazwisko}, \text{rok}), P = (\underline{\text{nazwa}}, \text{typ}), O = (\underline{\text{indeks}}, \underline{\text{przed}}, \underline{\text{data}}, \text{stop})$

1. $\pi_{S.\text{indeks}, \text{nazwisko}}(\sigma_{\text{stop}=5.0 \wedge \text{przed} \neq 'BD'}(S \bowtie O));$
2. $\pi_{S.\text{indeks}, \text{nazwisko}, \text{rok}}(S \bowtie \sigma_{\text{stop}=5.0}(O));$
3. $\pi_{S.\text{indeks}, \text{nazwisko}}(S \bowtie \sigma_{i1=\text{indeks} \wedge p1=\text{przed} \wedge \text{przed} \neq 'BD' \wedge \text{data} \neq d1}(\rho_{O1}(i1, p1, d1, s1)(O) \times O)).$

Znaczenie zapytań

1. Indeksy i nazwiska studentów, którzy dostali 5.0 z *BD*.
2. Pełne dane studentów, którzy dostali jakąś ocenę 5.0.
3. Studenci, którzy podchodzili do *BD* co najmniej dwa razy.

Przykład 4

Baza danych

$$S = (\underline{indeks}, \underline{nazwisko}, rok), P = (\underline{nazwa}, \underline{typ}), O = (\underline{indeks}, \underline{przed}, \underline{data}, \underline{stop})$$

Studenci, którzy nie dostali 5.0.

- 4a. $\pi_{S.indeks, nazwisko, rok}(S \bowtie \sigma_{stop \neq 5.0}(O));$
- 4b. $\pi_{S.indeks, nazwisko, rok}(S \bowtie \sigma_{stop \text{ IS NULL}}(O));$
- 4c. $\pi_{S.indeks, nazwisko, rok}(S) \setminus \pi_{S.indeks, nazwisko, rok}(S \bowtie \sigma_{stop = 5.0}(O));$
- 4d. $\pi_{S.indeks, nazwisko, rok}(S) \setminus \pi_{S.indeks, nazwisko, rok}(S \bowtie \sigma_{stop \neq 5.0}(O));$
- 4e. $\pi_{S.ind, naz, rok}(S \bowtie O) \setminus \pi_{S.indeks, nazwisko, rok}(S \bowtie \sigma_{stop \neq 5.0}(O));$

Znaczenie zapytań

- 4a. Studenci, którzy dostali jakąś ocenę inną niż 5.0.
- 4b. Studenci, którzy nie dostali wpisu (niezgodne z więzami relacji)
- 4c. Studenci, którzy nie dostali żadnej piątki.
- 4d. Studenci, którzy mają tylko oceny 5.0 (być może nie mają żadnych).
- 4e. Studenci, którzy dostają tylko piątki, przy czym bierzemy pod uwagę tylko tych, którzy mają jakikolwiek wpis.

Przykład 5

Baza danych

$S = (\underline{indeks}, \underline{nazwisko}, rok)$, $P = (\underline{nazwa}, typ)$, $O = (\underline{indeks}, \underline{przed}, \underline{data}, stop)$

Jak szukać czegoś, czego nie ma?

- 5a. $\pi_{S.indeks, nazwisko}(S) \setminus \pi_{S.indeks, nazwisko}(S \bowtie O)$;
- 5b. $\pi_{S.indeks, nazwisko}(S \bowtie \sigma_{stop \text{ IS NULL}}(O))$;
- 5c. $\pi_{S.indeks, nazwisko}(S \bowtie \sigma_{stop= \text{NULL}}(O))$;
- 5d. $\pi_{S.indeks, nazwisko}(S \bowtie \sigma_{stop \neq \text{NULL}}(O))$;

Krotka jest wybierana przez selekcję, gdy warunek ma dla niej wartość TRUE. Wartość UNKNOWN nie wystarcza.

Przykład 6

Baza danych

$$S = (\underline{\text{indeks}}, \text{nazwisko}, \text{rok}), P = (\underline{\text{nazwa}}, \text{typ}), O = (\underline{\text{indeks}}, \underline{\text{przed}}, \underline{\text{data}}, \text{stop})$$

Można pytać o to samo na różne sposoby. Czy to ma jakieś znaczenie?

- (6) $\pi_{\text{nazwisko}, \text{indeks}}(\sigma_{\text{stop}=5.0 \wedge \text{typ}='zaaw'}(\sigma_{\text{nazwa}=\text{przed}}(P \times O))) \bowtie \sigma_{\text{rok}=4}(S)$
- $\cup \pi_{\text{nazwisko}, \text{indeks}}(\sigma_{\text{stop}=5.0 \wedge \text{typ}='obow'}(\sigma_{\text{nazwa}=\text{przed}}(P \times O))) \bowtie \sigma_{\text{rok}=3}(S);$
- (6a) $\pi_{\text{nazwisko}, \text{indeks}}(\sigma_{((\text{rok}=3 \wedge \text{typ}='obow') \vee (\text{rok}=4 \wedge \text{typ}='zaaw'))}(\sigma_{\text{rok}=3 \vee \text{rok}=4}(S))) \bowtie \pi_{\text{indeks}, \text{typ}}(\rho P(\text{przed}, \text{typ})(\sigma_{\text{typ}='zaaw' \vee \text{typ}='obow'}(P)))) \bowtie \pi_{\text{indeks}, \text{przed}}(\sigma_{\text{stop}=5.0}(O))))$

Przykład 7 - poszukajmy najlepszych z BD

Baza danych

$S = (\textit{indeks}, \textit{nazwisko}, \textit{rok})$, $P = (\textit{nazwa}, \textit{typ})$, $O = (\textit{indeks}, \textit{przed}, \textit{data}, \textit{stop})$

$$(7a) \pi_{\textit{indeks}}(\sigma_{\textit{stop} > s1 \wedge \textit{przed} = \textit{BD}'' \wedge p1 = \textit{przed}}(O \bowtie \rho_{O1}(i1, p1, d1, s1)(O)))$$

$$(7b) \pi_{\textit{indeks}}(\sigma_{\textit{stop} < s1 \wedge \textit{przed} = \textit{BD}'' \wedge p1 = \textit{przed}}(O \bowtie \rho_{O1}(i1, p1, d1, s1)(O)))$$

$$(7c) \pi_{\textit{indeks}}(S) \setminus \pi_{\textit{indeks}}(\sigma_{\textit{stop} < s1 \wedge \textit{przed} = \textit{BD}'' \wedge p1 = \textit{przed}}(O \bowtie \rho_{O1}(i1, p1, d1, s1)(O)))$$

$$(7d) \pi_{\textit{indeks}}(\sigma_{\textit{przed} = \textit{BD}''}(O)) \setminus \pi_{\textit{indeks}}(\sigma_{\textit{stop} < s1 \wedge \textit{przed} = \textit{BD}'' \wedge p1 = \textit{przed}}(O \bowtie \rho_{O1}(i1, p1, d1, s1)(O)))$$

Znaczenie zapytań

- 7a. Indeksy studentów, którzy z BD mają ocenę lepszą niż ktoś inny, czyli nie są najgorsi.
- 7b. Indeksy studentów, którzy z BD mają ocenę gorszą niż ktoś inny, czyli nie są najlepsi (dopełnienie tego, czego szukamy).
- 7c. Indeksy studentów, którzy nie są od nikogo gorsi z BD.
- 7d. Indeksy studentów, którzy są najlepsi z BD.

Wnioski i uwagi:

- 1 Algebra relacji jest językiem imperatywnym (operacyjnym).
- 2 Znaczenie zapytania (w języku naturalnym) nie zawsze jest oczywiste, gdyż algebra relacji nie przypomina języka naturalnego.
- 3 To samo zapytanie może mieć wiele równoważnych postaci — mogą one różnić się złożonością wykonania.
- 4 Na podstawie samego opisu trudno określić moc tego języka.
- 5 Algebra relacji jest podstawą SQL.

Materiały na skosie:

- 1 Trudne zapytania w algebrze relacji:
<https://skos.ii.uni.wroc.pl/mod/resource/view.php?id=28050>
- 2 Kalkulator algebry relacji:
<https://dbis-uibk.github.io/relax/landing>