

BAZY DANYCH

5

GRZEGORZ SURÓWKA

2018

PLAN WYKŁADU

1. WPROWADZENIE
2. DBMS, MODELE DANYCH, MODEL RELACYJNY
3. DIAGRAMY E/R
4. E/R → MODEL RELACYJNY
5. **ALGEBRA RELACJI**
6. ZŁĄCZENIA
7. NORMALIZACJA BAZY DANYCH
8. TRANSAKCJE
9. OPTYMALIZACJA ZAPYTAŃ
10. WYDAJNOŚĆ
11. PODZAPYTANIA SQL
12. GRUPOWANIE
13. WIDOKI, KURSORY, WYZWALACZE, SEKWENCJE
14. SQL

PLAN

- DZIAŁANIA NA RELACJACH
- FORMALIZM
- OGRANICZENIA INTEGRALNOŚCIOWE
- OPERACJE NA TABELACH
- OPERATORY SET
- RACHUNEK RELACYJNY

DZIAŁANIA NA RELACJACH

- relacyjne bazy danych są wyposażone w język, za pomocą którego wydobywa się dane
- istnieją dwa sposoby (języki) zapytań: algebra relacyjna i rachunek relacyjny

ALGEBRA RELACYJNA

- język proceduralny
- wejście: instancje relacji lub relacje,
wyjście: instancje relacji lub relacje
- operatory: unitarne, binarne
- działanie rekurencyjne na relacji,
wyniki pośrednie też są relacjami
→ dlatego możliwe są operacje złożone
- algebra relacji jest domknięta

formalizm relacji

baza danych to zbiór relacji (tabel)

schemat relacji R to zbiór $R = \{A_1, A_2, \dots, A_n\}$,
gdzie A_1, A_2, \dots, A_n są atrybutami (nazwami kolumn)

stopień relacji to liczba atrybutów schematu relacji

DOM(A_i) to dziedzina czyli zbiór wartości atrybutu

dziedzina relacji to suma dziedzin wszystkich atrybutów relacji
 $\text{DOM}(R) = \text{DOM}(A_1) \cup \text{DOM}(A_2) \cup \dots \cup \text{DOM}(A_n)$

$r(R)$ relacja **r** o schemacie **$R = \{A_1, A_2, \dots, A_n\}$**

- jest to skończony zbiór krotek postaci **$r = \{t_1, t_2, \dots, t_m\}$**
- pojedyncza krotka **t** jest uporządkowana listą n wartości **$t = \langle v_1, v_2, \dots, v_n \rangle$** , gdzie v_i ($1 \leq i \leq n$) jest elementem **$DOM(A_i)$** lub wartością NULL
- **$t[A_i]$** to i -ta (atomowa) wartość krotki t odpowiadająca wartości atrybutu A_i
- **$r(R)$** jest relacją matematyczną stopnia n zdefiniowaną na zbiorze domen **$DOM(A_1), DOM(A_2), \dots, DOM(A_n)$** będącą podzbiorem iloczynu kartezjańskiego domen definiujących R :
 $r(R) \subseteq DOM(A_1) \times DOM(A_2) \times \dots \times DOM(A_n)$

podsumowanie

- baza danych = zbiór relacji (tabel)
- schemat bazy danych = zbiór schematów relacji
- schemat relacji = zbiór {atrybut, dziedzina, [ograniczenia integralnościowe]}
- każdy atrybut relacji ma unikalną nazwę
- porządek atrybutów w relacji nie jest istotny
- wartości atrybutów są atomowe (elementarne)
- relacja = zbiór krotek,
z definicji relacji (która jest zbiorem),
wszystkie krotki relacji muszą być unikalne
- krotka = rekord = lista wartości
- porządek krotek w relacji nie jest istotny

jawnie zdefiniowane ograniczenia integralnościowe

- nad ich przestrzeganiem czuwa DBMS
- są definiowane na poziomie
 - ✓ pojedynczego atrybutu
 - ✓ całej relacji
- rodzaje
 - ✓ klucz podstawowy (primary key)
 - ✓ klucz obcy (foreign key)
 - ✓ unikalność (unique)
 - ✓ zawężenie domeny/dziedziny (check)
 - ✓ wartość pusta/niepusta (NULL/NOT NULL)

unikalność krotek relacji - klucze

superklucz relacji (super key)

- to każdy podzbiór S atrybutów relacji, taki że dla każdych dwóch krotek ze zbioru $r(R)$ zachodzi $t_1[S] \neq t_2[S]$
- w ogólności cały schemat relacji
- może posiadać nadmiarowe atrybuty

klucz K schematu relacji R

- superklucz schematu R o takiej własności, że usunięcie dowolnego atrybutu A z K powoduje, że $K' = K - A$ nie jest już superkluczem
- klucz jest minimalnym superkluczem zachowującym własność unikalności krotek relacji
- schemat relacji może posiadać więcej niż jeden klucz

klucz główny/podstawowy (PK, Primary Key)

- wyróżniony klucz

klucze wtórne/kandydackie

- pozostałe klucze

Pracownicy

idPrac	Imie	Nazwisko	Szef	IdZesp
100	Jan	Miś		10
110	Piotr	Wilk	100	10
120	Roman	Lis	100	20

↑
klucz obcy

Zespoły

idZesp	Nazwa
10	Reklama
20	Badania

↑
klucz główny

~~insert {130, "Zenon", "Szop", 100, 50}~~

naruszenie integralnościowe klucza obcego

klucz obcy relacji (FK, Foreign Key)

- dane: relacje A i B

klucz obcy relacji A to podzbiór FK atrybutów relacji A:

- ✓ atrybuty relacji A w FK mają taką samą domenę jak atrybuty klucza podstawowego PK relacji B
 $DOM(FK)=DOM(PK)$
- ✓ dla każdej krotki t_1 relacji A istnieje dokładnie jedna krotka t_2 relacji B, taka że $t_1[FK]=t_2[PK]$, lub $t_1[FK]=NULL$
- to atrybut (lub zbiór atrybutów), który wskazuje na klucz podstawowy
- służy do reprezentowania powiązań między danymi (łączenia relacji)
- gwarantuje, że rekordy z tabeli A występują w kontekście związanego z nim rekordu z tabeli B

zawężenie dziedziny

- etat: dziedzina: {"Analityk", "Dyrektor", "Referent", "Kierownik", "Sekretarka"}
- płaca: dziedzina: Placa > 1000
- klucz podstawowy: IdPrac

Pracownicy

IdPrac	Nazwisko	Etat	Placa	Szef	IdZesp
120	Kowalski	Analityk	1700	100	10
100	Tarzan	Dyrektor	5400		10
130	Nowak	Referent	2200	100	10
110	Józiak	Kierownik	2600	100	20
140	Nowacki	Analityk	3600	100	20
150	Bunio	Sekretarka	2400	100	10

~~insert {200, "Szop", "Strażak", 1900, 10}~~ → narusza integralność Etat'u

~~insert {130, "Borsuk", "Kierownik", 1500, 20}~~ → narusza klucz główny

~~insert {210, "Rosomak", "Kierownik", 400, 20}~~ → narusza Placa

algebra relacji

podstawowe operacje na tabelach

- selekcja σ
- rzutowanie π
- przemianowanie ρ
- złączenie \bowtie
- suma mnogościowa (unia) \cup
- przecięcie (iloczyn) zbiorów \cap
- różnica zbiorów \setminus
- iloczyn kartezjański \times

selekcja: σ

- wybiera z relacji krotki, które spełniają warunki selekcji
- warunek selekcji jest zbiorem predykatów
- notacja: $\sigma_p(r)$, $\sigma_p r$ r-relacja
p-predykat/zdanie logiczne: może używać łączników:
AND, OR, NOT i operatorów: =, \neq , \geq , $<$, $>$, \leq
- operacja selekcji jest komutatywna:

$$\sigma_{\langle \text{war1} \rangle} (\sigma_{\langle \text{war2} \rangle} (r)) = \sigma_{\langle \text{war2} \rangle} (\sigma_{\langle \text{war1} \rangle} (r))$$

- **SELECT ... SELECT DISTINCT ...**
- przykład: wybierz krotki z tablicy (relacji) Ksiazki, których dziedzina to "SQL" i kosztują mniej niż 100 (zł) lub opublikowano je nie wcześniej niż w 2012 roku
 $\sigma_{\text{Dziedzina}=\text{"SQL"} \text{ AND } \text{Cena} \leq 100 \text{ OR } \text{Rok} > \text{"2012"}} (\text{Ksiazki})$

przykłady

- $\sigma_{\text{IdZesp}=10}$ (Pracownicy)
select IdPrac, Nazwisko, Etat, Szef, Zatrudniony, Placa, IdZesp from Pracownicy where IdZesp=10
- $\sigma_{\text{Placa}>2000}$ (Pracownicy)
select * from Pracownicy where Placa>2000
- $\sigma_{(\text{IdZesp}=10 \text{ AND } \text{Placa}>7000) \text{ OR } (\text{IdZesp}=20 \text{ AND } \text{Placa}>8000)}$ (Pracownicy)
select * from Pracownicy where (IdZesp=10 and Placa>7000) or (IdZesp=20 and Placa>8000)
- $\sigma_{\text{Etat}=\text{"Księgowy"} \text{ AND } (\text{Placa}\geq 6000 \text{ AND } \text{Placa}<9000)}$ (Pracownicy)
select * from Pracownicy where Etat="KSIĘGOWY" and (Placa>=6000 and Placa<9000)

rzutowanie (projekcja): π

- wyodrębnia wybrane atrybuty relacji, które spełniają pewien warunek (predykat)
- notacja: $\pi_{A_1, A_2, A_3, \dots}(r)$, $\pi_{A_1, A_2, A_3, \dots}r$
r-relacja
 A_1, A_2, A_3, \dots -nazwy atrybutów (kolumn)
- zduplikowane wiersze są eliminowane
- operacja projekcji nie jest komutatywna
- składanie operacji projekcji jest możliwe jeżeli lista2 zawiera wszystkie atrybuty lista1
$$\pi_{\langle \text{lista1} \rangle}(\pi_{\langle \text{lista2} \rangle}(r)) = \pi_{\langle \text{lista1} \rangle}(r)$$
- przykład: wybierz z tablicy Ksiazki kolumny dziedzina i autor $\pi_{\text{Dziedzina}, \text{Autor}}(\text{Ksiazki})$

przykłady

- $\pi_{\text{Nazwisko}}(\text{Pracownicy})$
`select Nazwisko from Pracownicy`
- $\pi_{\text{Nazwisko, Etat, Placa}}(\text{Pracownicy})$
`select Nazwisko, Etat, Placa from Pracownicy`
- sekwencja wielu operacji, w której kolejne operacje są wykonywane na pośrednich wynikach operacji poprzednich, może być zastąpiona pojedynczą operacją złożoną, powstałą przez zagnieżdżenie operacji elementarnych

$\sigma_{\text{IdZesp}=10}(\text{Pracownicy}) \rightarrow \text{PracZesp10}$

$\pi_{\text{IdPrac, Nazwisko}}(\text{PracZesp10}) \rightarrow \text{PracZesp10Wynik}$

$\text{PracZesp10Wynik} = \pi_{\text{IdPrac, Nazwisko}}(\sigma_{\text{IdZesp}=10}(\text{Pracownicy}))$

przemianowanie: ρ

- przemianowuje nazwy atrybutów tabeli lub nazwę samej tabeli
- liczba i kolejność atrybutów zostaje zachowana
- notacja: $\rho_{s(A_1, A_2, A_3, \dots)}r$, $\rho_s r$
- przykład: zmień nazwę tabeli Książki na Books oraz zmień atrybuty na angielskie

$\rho_{\text{Books}(\text{Author}, \text{Title}, \dots)}$ Książki

operacje na zbiorach

kompatybilność relacji

- dwie relacje: $r(A_1, \dots, A_n)$ i $s(B_1, \dots, B_n)$ są kompatybilne, jeżeli mają ten sam stopień i jeżeli $\text{DOM}(A_i) = \text{DOM}(B_i)$ dla $1 \leq i \leq n$

dla dwóch kompatybilnych relacji

$r(A_1, \dots, A_n)$ i $s(B_1, \dots, B_n)$ możliwe są operacje

- suma
- iloczyn (przecięcie)
- różnica

należy uporządkować atrybuty obu tabel tak, aby kolejność atrybutów była taka sama

wielozbiory

- tabele w modelu relacyjny powinny być zbiorami (krotki nie mogą się powtarzać), ale niekiedy nie są - jeśli dopuszczamy powtórzenia krotek, czyli zbiory zastępujemy wielozbiorami, zmieniają się definicje operacji mnogościowych
- suma $r \cup s$: krotka w wyniku występuje tyle razy, ile występuje w r plus tyle razy, ile występuje w s
uwaga: jeśli nawet r i s są zbiorami, $r \cup s$ może być wielozbiorem!
- iloczyn $r \cap s$: krotka w wyniku występuje tyle razy, ile wynosi minimum jej wystąpień w r i s
- różnica $r \setminus s$: krotka w wyniku występuje tyle razy, ile występuje ona w r minus tyle razy, ile występuje ona w s , ale nie mniej niż 0 razy

przykład

$$r=\{A, B, B\}$$

$$s=\{A, B, C, C\}$$

$$r \cup s = \{A, A, B, B, B, C, C\}$$

$$r \cap s = \{A, B\}$$

$$r \setminus s = \{B\}$$

- wielozbiory pojawiają się (często) jako tabele wynikowe pewnych zapytań - tabele te mogą być tabelami wejściowymi kolejnych zapytań ...

suma mnogościowa (unia): \cup

- notacja: $r \cup s$
r,s-relacje (mogą też być chwilowe)
- $r \cup s = \{t, t \in r \text{ OR } t \in s\}$
wynikiem jest relacja zawierająca wszystkie krotki, które występują w r i wszystkie krotki, które występują w s
- warunek: r i s muszą mieć tę samą liczbę atrybutów, domeny muszą być kompatybilne, zduplikowane krotki są usuwane
- operacja sumy jest operacją komutatywną:
$$r \cup s = s \cup r$$
- przykład: podaj nazwiska autorów książek i artykułów
$$\pi_{\text{Autor}}(\text{Ksiazki}) \cup \pi_{\text{Autor}}(\text{Artykuly})$$

przecięcie (iloczyn) zbiorów: \cap

- notacja: $r \cap s$
r,s-relacje
- $r \cap s = \{t, t \in r \text{ AND } t \in s\}$
wynikiem jest relacja zawierająca wszystkie krotki,
które występują zarówno w r i s
- operacja iloczynu jest operacją komutatywną:
$$r \cap s = s \cap r$$

różnica zbiorów: \, -

- znajdź krotki obecne w tabeli r i nieobecne w tabeli s
- notacja: $r - s$
- wynikiem tej operacji jest relacja zawierająca wszystkie krotki, które występują w r i nie występują w s
- operacja różnicy nie jest komutatywna:

$$r - s \neq s - r$$

- przykład: znajdź nazwiska autorów książek,
ale nie artykułów $\pi_{\text{Autor}}(\text{Książki}) - \pi_{\text{Autor}}(\text{Artykuły})$

Uczniowie

Imie	Nazwisko
Ala	Kusiak
Edek	Musiał
Adam	Zajac
Olek	Struś
Ola	Buba

Instruktorzy

Imie	Nazwisko
Jan	Kuc
Edek	Musiał
Wacek	Misiek

Uczniowie \cap Instruktorzy

Imie	Nazwisko
Edek	Musiał

Uczniowie \cup Instruktorzy

Imie	Nazwisko
Ala	Kusiak
Edek	Musiał
Adam	Zajac
Olek	Struś
Ola	Buba
Jan	Kuc
Wacek	Misiek

Uczniowie - Instruktorzy

Imie	Nazwisko
Ala	Kusiak
Adam	Zajac
Olek	Struś
Ola	Buba

Instruktorzy - Uczniowie

Imie	Nazwisko
Jan	Kuc
Wacek	Misiek

select Imie, Nazwisko from Uczniowie **UNION**
select Imie, Nazwisko from Instruktorzy

z duplikatami:

select Imie, Nazwisko from Uczniowie **UNION ALL**
select Imie, Nazwisko from Instruktorzy

select Imie, Nazwisko from Uczniowie **INTERSECT**
select Imie, Nazwisko from Instruktorzy

select Imie, Nazwisko from Uczniowie **MINUS**
select Imie, Nazwisko from Instruktorzy

select Imie, Nazwisko from Instruktorzy **MINUS**
select Imie, Nazwisko from Uczniowie

operatory SET (UNION [ALL], INTERSECTION, MINUS)

- łączą wyniki dwóch lub więcej zapytań składowych w jeden rezultat
- kwerendy zawierające operatory SET to kwerendy złożone
- wszystkie operatory SET mają ten sam priorytet
- jeśli kwerenda SQL zawiera wiele operatorów SET, to jest wykonywana w porządku od lewej do prawej
- nawiasy zmieniają porządek operatorów (w szczególności powinny być stosowane w kwerendach, które zawierają operator INTERSECT i inne operatory SET)
- liczba kolumn i typy danych w zapytaniach składowych SELECT muszą być takie same

operator SET – dyrektywa ORDER BY

- może wystąpić wyłącznie na samym końcu kwerendy złożonej
- argumentem może być nazwa kolumny lub aliasy z pierwszego zdania SELECT

operator UNION

- zwraca sumę z zapytań składowych bez duplikatów
- domyślnie wynik jest sortowany w porządku rosnącym względem pierwszej kolumny
- liczba kolumn i typy danych w zapytaniach składowych muszą być takie same
- nazwy kolumn w tabelach składowych nie muszą być identyczne
- podczas sprawdzania duplikatów wartości NULL nie są ignorowane
- operator IN jest wyżej w hierchii niż operator UNION

operator UNION ALL

- zwraca sumę z zapytań składowych bez usuwania duplikatów
- domyślnie wynik nie jest sortowany
- słowo DISTINCT nie może być użyte
- liczba kolumn i typy danych w zapytaniach składowych muszą być takie same
- nazwy kolumn w tabelach składowych nie muszą być identyczne

operator INTERSECTION

- zwraca część wspólną z zapytań składowych
- liczba kolumn i typy danych w zapytaniach składowych muszą być takie same
- nazwy kolumn w tabelach składowych nie muszą być identyczne
- zmiana porządku zapytań składowych nie wpływa na wynik
- wartości NULL nie są ignorowane

operator MINUS

- zwraca krotki z pierwszego zapytania, których nie ma w zapytaniu drugim
- usuwa duplikaty
- liczba kolumn i typy danych w zapytaniach składowych muszą być takie same
- nazwy kolumn w tabelach składowych nie muszą być identyczne
- przy użyciu operatora MINUS wszystkie kolumny warunku WHERE muszą być zawarte w samym zapytaniu SELECT

iloczyn kartezjański: ✗

- zestawia informacje dwóch różnych relacji w jedną
- jeden z przypadków połączenia tabel (cartesian/cross join)
- notacja: $r \times s$
 $r(A_1, \dots, A_n), s(B_1, \dots, B_m)$ - relacje
- $r \times s = \{(p, t), p \in r \text{ AND } t \in s\}$
- wynikiem jest relacja Q stopnia $(n+m)$ i schemacie $q(A_1, \dots, A_n, B_1, \dots, B_m)$
- $(n \cdot m)$ krotkom w relacji q odpowiadają wszystkie kombinacje krotek z relacji r i s

- **SELECT ... FROM r cross join s**
- nazwy atrybutów są, o ile to możliwe, dziedziczone
- jeżeli wystąpią jakieś konflikty w nazwach kolumn, trzeba je rozwiązać przez przemianowanie przed dokonaniem iloczynu kartezjańskiego
- przykład: podaj wszystkie tytuły książek i artykułów napisanych przez autora="Kowalski"
 $\sigma_{\text{Autor}=\text{"Kowalski"}}(\text{Ksiazki} \times \text{Artykuly})$

Pracownicy

Imie	Nazwisko
Ala	Kusiak
Edek	Musiał
Adam	Zajęc

Zespoly

Nazwa	Lokalizacja
Reklama	Mostowa 10
Badania	Wolności 3

Pracownicy × Zespoly

Imie	Nazwisko	Nazwa	Lokalizacja
Ala	Kusiak	Reklama	Mostowa 10
Edek	Musiał	Reklama	Mostowa 10
Adam	Zajęc	Reklama	Mostowa 10
Ala	Kusiak	Badania	Wolności 3
Edek	Musiał	Badania	Wolności 3
Adam	Zajęc	Badania	Wolności 3

```
select Imie, Nazwisko, Nazwa, Lokalizacja from Pracownicy, Zespoly
select Imie, Nazwisko, Nazwa, Lokalizacja
from Pracownicy cross join Zespoly
```

RACHUNEK RELACYJNY

- nie-proceduralny język zapytań
(wykonuje akcje, ale nie wyjaśnia jak)
- formy

Rachunek Relacyjny Krotek

(TRC, Tuple Relational Calculus)

Rachunek Relacyjny Domen

(DRC, Domain Relational Calculus)

Rachunek Relacyjny Krotek

- filtruje zakres zmiennych po krotkach, zwraca krotki, które spełniają warunek
- notacja: $\{T \mid \text{Warunek}\}$
T-tabela (relacja)
- przykład: zwróć krotki z nazwiskiem autora, który napisał artykuły o SQL
T=Artykuly, inna tabela=Autorzy
 $\{T.\text{Nazwisko} \mid \text{Autorzy}(T) \text{ and } T.\text{Dziedzina} = \text{"SQL"}\}$
- może używać kwantyfikatorów: \exists , \forall
- przykład: (jak powyższy, wyprodukuję to samo wyjście)
 $\{R \mid \exists T \in \text{Autorzy}(T.\text{Dziedzina} = \text{"SQL"} \text{ and } R.\text{Nazwisko} = T.\text{Nazwisko})\}$

Rachunek Relacyjny Domen

- filtruje zakres zmiennych po krotkach, zwraca niektóre atrybuty krotek, które spełniają warunek
- notacja: $\{a_1, a_2, a_3, \dots, a_n \mid P(a_1, a_2, a_3, \dots, a_n)\}$
a₁, a₂, a₃, ...a_n-atrybuty
P()-formuła z atrybutów
- może używać kwantyfikatorów:
 \exists, \forall oraz operatorów relacyjnych
- przykład: zwróć Nazwa, Strona, Dziedzina z tabeli Artykuly
 $\{ \langle \text{Nazwa}, \text{Strona}, \text{Dziedzina} \rangle \mid \in \text{Artykuly} \wedge \text{Dziedzina} = \text{"SQL"} \}$