

BAZY SKRYPT

1. **Baza danych** – trwały, uporządkowany zbiór tematycznie powiązanych danych.
 - operacyjna: nie tylko gromadzenie danych, ale także ich modyfikowanie(CRUD)
 - analityczna: jej dane pochodzą często z operacyjnej, ale po wprowadzeniu są stałe – podlegają tylko analizie
2. **Dobrze zaprojektowana:**
 - brak redundancji
 - bezpieczeństwo danych
 - atomowe dane(atrybuty w l.p)
 - rozróżnialne rekordy
 - SCHEMATY PRZYNAJMNIEJ W 3PN
3. **Podstawowe operacje na danych:** CRUD (create,read,update,delete)
4. **Hurtownia danych** – zintegrowana, zorientowana tematycznie, zmienna w czasie baza danych(analityczna, wykorzystująca dane z wielu operacyjnych) – baza zorganizowana i zoptymalizowana pod kątem pewnego wycinka rzeczywistości
5. **Dana** – nazwa + wartość
6. **Informacja** – zinterpretowana dana
7. **Wiedza** – zbiór reguł (w połączeniu z danymi możemy podjąć decyzję)
8. **Mądrość** – wykorzystanie wiedzy
9. **System Zarządzania Bazą Danych (SZBD)** – system informatyczny służący do zarządzania bazą danych – może być serwerem baz danych lub udostępniać bazę lokalnie – **oprogramowanie narzędziowe służące do zakładania, przechowywania i wydajnego przetwarzania dużych zbiorów informacji.** (np. Access, Oracle) baza danych +oprogramowanie + sprzęt + ludzie.
 - tworzy konta, operacje na danych,
 - tworzy strukturę,
 - kontrola dostępu i bezpieczeństwa,
 - obsługa SQL,
 - przetwarzanie transakcyjnie,
 - tworzenie kopii zapasowych,
 - praca współbieżna.

10. Ludzie związani z bazą danych:

- projektanci, testerzy, architekci, programiści,
- administratorzy,
- użytkownicy.

11. Języki baz danych:

- DDL (Data definition language) – języki definiowania, czyli opisy danych:

CREATE, DROP, ALTER

- DML (Data manipulation language) – języki manipulowania danymi:

INSERT, UPDATE, DELETE

- DQL (Data query language) – języki zapytań baz danych: SELECT

- DCL – języki nadzoru, czyli nadawanie uprawnień i zarządzanie nimi: GRANT, REVOKE, DENY

DDL+DML+DCL = SQL

12. Model danych – zbiór zasad opisujących strukturę danych w BD, reprezentuje sposób przechowywania danych, określone są również dozwolone operacje.

- hierarchiczny – dane układane są hierarchicznie w strukturę drzewa, uporządkowane od ogółu do szczegółu, np. katalogi danych na dysku. Wady: redundancja, możliwy brak spójności, konieczna znajomość całej ścieżki dostępu, problem: usuwanie elementu usuwa wszystkie jego dzieci.
- sieciowy – rozszerzenie hierarchicznego, tutaj poddrzewa mogą być powiązane z wieloma komórkami, tj. jedno dziecko może mieć wiele rodziców. Wady: ścieżka dostępu i usuwanie.
- relacyjny - oparty na matematycznych zasadach logiki predykatów i teorii zbiorów. Dane są przechowywane w relacjach (nazwa + atrybuty), czyli powstają krotki, które tworzą całą relację. Można wykonywać operacje zgodne ze zbiorem relacji.
relacja to zbiór krotek $\{t_1, t_2, \dots, t_n\}$, podzbiór iloczynu kartezjańskiego.
- obiektowy – reprezentacja danych w formie obiektów, dostęp przez referencje (zawiera pola i metody).
- logiczny – dane przedstawione są w formie tabeli, operuje się na nich przy pomocy rachunku predykatów i logiki zdań.
- temporalny – posiada informację o czasie wprowadzenia lub czasie ważności danych.

13. **Administrator BD** – osobowa odpowiedzialna za tworzenie, instalowanie i zarządzanie BD:
- dodawanie i odbieranie uprawnień,
 - zmniejszenie zapamiętanych danych,
 - poprawność danych,
 - wprowadzenie standardów w BD,
 - zachowanie integralności danych.
14. **System bazy danych** – baza danych+ludzie+sprzęt+oprogramowanie
15. **Przykłady systemów z potrzebną autoryzacją**: Facebook, e-mail, edukacja.ci
16. **Bez autoryzacji**: rozkład jazdy, pogoda, Filmweb
17. **Dziedzina** – zbiór wartości jakie może przyjąć atrybut
18. **Atrybut** – użycie dziedziny, ma nazwę i przyjmuje wartości z ustalonego zbioru-dziedziny
19. **Relacja n-członowa** – podzbiór iloczynu kartezjańskiego składający się z n-dziedzin(zbiorów wartości)
20. **Krotka** – pojedynczy element iloczynu kartezjańskiego - obiekt w relacji – zbiór par postaci: nazwa atrybutu – wartość atrybutu.
21. **Składowa krotki** – wartość atrybutu w krotce.
22. **Schemat relacji** – specyfikacja relacji opisująca jej nazwę oraz unikalne atrybuty
23. **Instancja relacji** – zbiór krotek
24. **Liczebność relacji** – liczba krotek w relacji – $\text{card}(R)$
25. **Stopień relacji** – liczba atrybutów relacji
26. **Zbiór identyfikujący relację** – zbiór atrybutów, których kombinacje wartości jednoznacznie identyfikują każdą krotkę relacji
27. **Klucz** – minimalny zbiór identyfikujący relacji, atrybut lub minimalny zbiór atrybutów, którego wartości jednoznacznie identyfikuje krotkę w relacji
28. **Klucz główny** – klucz wybrany spośród kluczy kandydujących
29. **Klucz kandydujący** – jeden z kluczy jednoznacznie identyfikujących relację

30. **Klucz obcy** – atrybut(y) który jest kluczem w innej relacji
31. **Klucz sztuczny** – sztucznie utworzony klucz, gdy potrzeba klucza głównego a nie wytworzył się naturalnie w relacji(np. IdCoś)
32. **Klucz prosty** – klucz z jednego atrybutu(zbiór identyfikujący relacji jest jednoelementowy)
33. **Złożony** – klucz z wielu atrybutów
34. **Nadklucz** – zbiór atrybutów, który zawiera klucz i który zapewnia rozróżnialność krotek(w szczególności zbiór wszystkich atrybutów)

<<Przykłady relacji>>

35. PROJEKTOWANIE BAZY DANYCH

1. **Temat, cel, użytkownicy**
2. **Szczegółowa analiza wycinka rzeczywistości:**

Wycinek rzecz – rzeczywistość fizyczna(książka,bilet) i abstrakcyjna(wypożyczenie,lot,kupno)

- Analiza wycinka: wywiad z ekspertem dziedzinowym i określenie założeń: użytkownicy, reguły funkcjonowania, wymagania funkcyjne, co ma zawierać baza, atrybuty obiektów i dziedziny, zgromadzenie materiałów(przepisy regulujące itd.)
 - Słownik pojęć: istotne pojęcia bardzo precyzyjnie zdefiniować
 - Analiza istniejącej BD
 - Analiza wymagań funkcjonalnych: ustalenie, wyszczególnienie i precyzyjne opisanie operacji
 - Analiza wymagań niefunkcjonalnych: sprzęt, system, środowisko implementacyjne, rodzaj BD, oszacowanie liczby danych, archiwizowanie
 - Analiza kosztów: projekt i implementacja, wdrożenie i eksploatacja, sprzęt
3. **Wyodrębnienie kategorii:** z analizy rzecz. Wyodrębniamy rzeczy o których chcemy przetrzymywać informacje(rzeczowniki w LP) + ustalamy atrybuty(KAT/x Nazwa |Opis: |Atrybuty:)
 4. **Wyodrębnienie reguł funkcjonowania** (REG/x <sformułowanie w j.natur>)
 5. **Ograniczenia dziedzinowe:** nakładamy na kategorie i atrybuty (OGR/00x <sformułowanie w j.natur>)
 6. **Transakcje(operacje bazodanowe):** niepodzielne,spójne,niezależne,trwałe (TRA/00x Nazwa |Opis: |Uwarunkowania: |We/wy
 7. **Definiowanie encji i związków:** silne(niezależne), słabe. (ENC/X Nazwa |Semantyka – opis |Atrybuty – opis i dziedziny |Klucze kandydujące |

Klucz główny | Charakter – silna lub słaba) – NIE MA KLUCZY OBCYCH! ; Związki – powiązanie między encjami – NIE MA ZWIĄZKÓW N do N

8. Definicje predykatowe encji i związków: zwarte i czytelne

ENC/x NAZWA_TYPU(Lista atr)

ZWI/X Nazwa(ENCJA1(liczność):ENCJA2(liczność);lista atr związku)

9. Diagram ERD – diagram związków encji - bez zbędnych połączeń

10. Transformacja do modelu logicznego – z encji i związków w relacje 3PN

11. Definiowanie schematów relacji

REL/00x Nazwaschematu/NAZWA.TYP.ENCJI

+ tabela: atrybuty, dziedziny, maski, obl/opc, wart. Domyślna, unikalność, klucz, referencje do relacji, źródło danych

12. Schemat bazy danych wraz ze słownikiem atrybutów:

NAZWA BAZY DANYCH

Nazwa.schematu.relacji1(lista_atrybutów)

Słownik – nazwa, dziedzina, przynależność do schematu relacji

DataSprzedazy, Date, Faktury

13. Użytkownicy i perspektywy

(Tabela uprawnień)

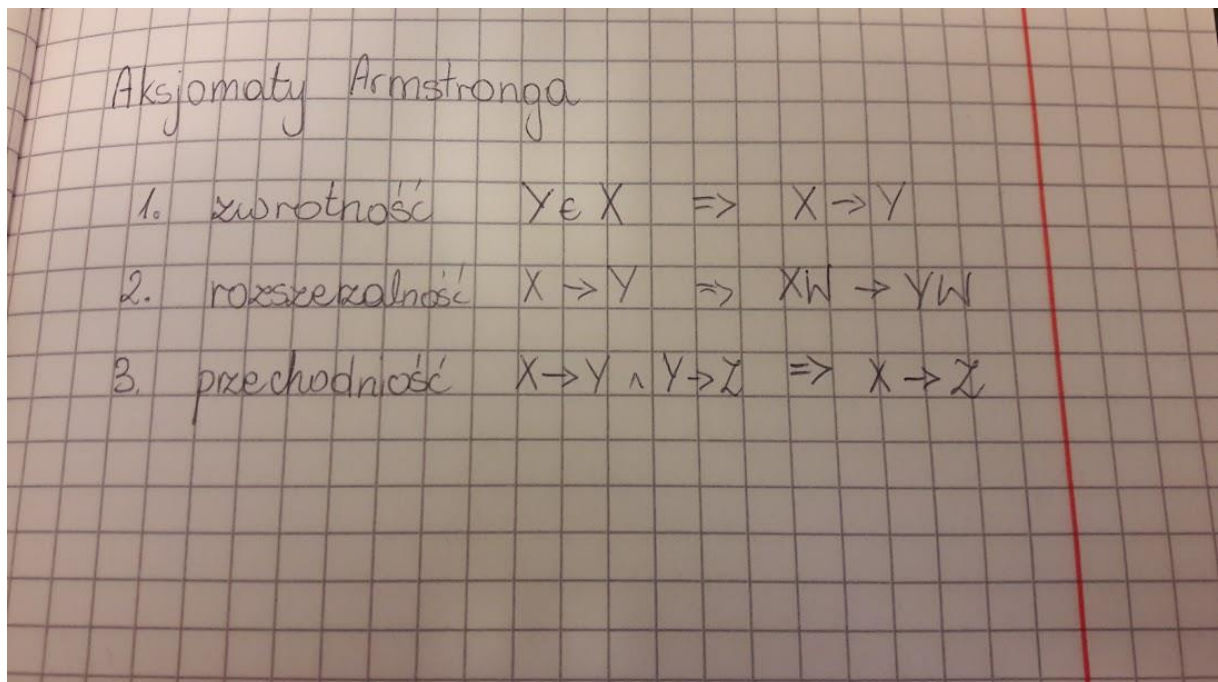
PER/001 Dane pracownika

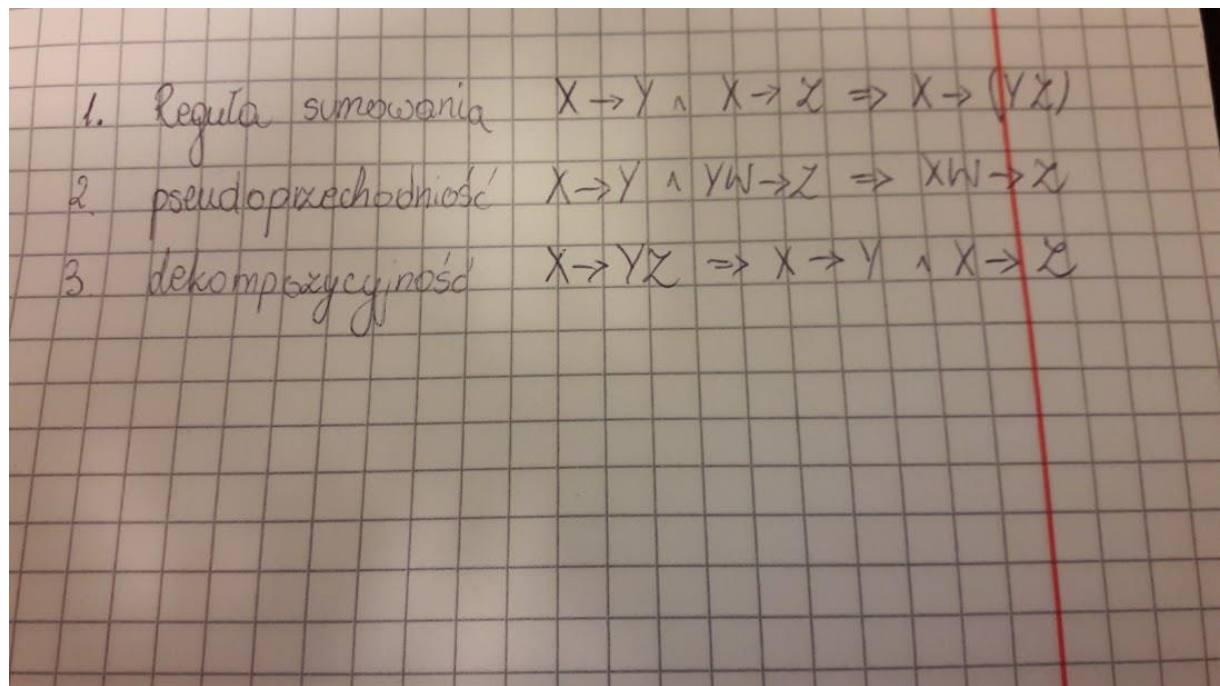
Użytkownik: Administrator

Transakcje: TRA/001

W tabelce tylko te atrybuty które biorą udział.

AKSJOMATY ARMSTRONGA





36. **Pierwsza postać normalna** – schemat relacji jest w 1PN kiedy każda wartość z dziedziny jest elementarna(atomowa)

Np. Klienci(NrKlienta,Nazwisko,Imie) a nie Osoby(NrOsoby,Imie,ImionaDzieci)

37. **Zależność funkcyjna(jednoznaczna identyfikacja, $A \rightarrow B$)** – atrybut B jest funkcyjnie zależny od A, jeśli każdej wartości a atrybutu A odpowiada co najwyżej jeden atrybut b z B np. Osoby(PESEL,Nazwisko): PESEL \rightarrow Nazwisko, ale nie Nazwisko \rightarrow PESEL

Dla zbiorów: jeśli każdej krotce wartości atrybutów Z1 odpowiada nie więcej niż jedna krotka atrybutów Z2. Wtedy Z1=wyznacznik zależności funkcyjnej.

Np. Egzaminy(NumerIndeksu,Kurs,Ocena,Data)

{NumerIndeksu,Kurs} \rightarrow {Ocena,Data}

lecz {Ocena,Data} $\not\rightarrow$ {NumerIndeksu,Kurs}

38. **Pełna zależność funkcyjna(\Rightarrow)** gdy zbiór Z2 jest funkcyjnie zależny od Z1, ale nie jest funkcyjnie zależny od żadnego podzbioru właściwego Z1

Np. {Student,Przedmiot} \Rightarrow Ocena, bo Student $\not\rightarrow$ Ocena i Przedmiot $\not\rightarrow$ Ocena

ogólnie na zajęciach był używany jakiś łatwy sposób określania tego ale nie pamiętam, masz to?

39. **Druga postać normalna** – schemat jest w 2PN, jeżeli każdy atrybut, nie wchodzący w skład żadnego klucza kandydującego, jest w pełni funkcyjnie zależny od każdego klucza kandydującego//każdy atrybut niekluczowy zależny od klucza//

Np. Studenci(Indeks,Imię,Nazwisko,Dur,Adres)

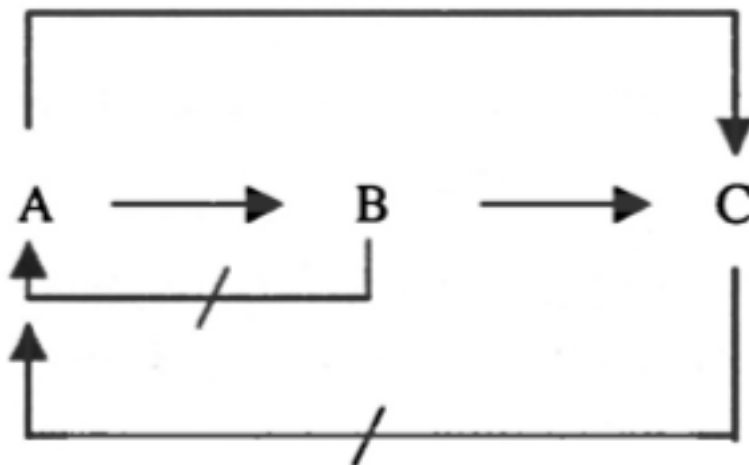
//Zawsze jeśli nie ma atrybutów niekluczowych

//Zawsze jeśli wszystkie klucze kandydujące są kluczami prostymi ->

Klucze proste = 2PN

zawsze kiedy jest klucz prosty?

40. Przechodnia zależność funkcyjna(A->C)



41. Trzecia postać normalna – jeżeli jest 2PN oraz każdy atrybut niekluczowy nie jest przechodnio funkcyjnie zależny od żadnego klucza kandydującego

//każdy atrybut zależy tylko od klucza, a nie między sobą//

np. Zajęcia(Student,Kurs,Wykładowca)

{Student,Kurs}->Wykładowca – student może uczęszczać na dany kurs tylko do 1 wykładowcy

//Zawsze jeśli jest tylko jeden atrybut niekluczowy

<< SPRAWDZANIE PN W SCHEMATACH, DEKOMPOZYCJA, RYSOWANIE ERD>>

//zacyna się od wskazania kluczy kandydujących, wie ktoś jak to się robi?

//kluczami kandydującymi są wszystkie atrybuty określające zależności (Wszystko po lewej od strzałki przy zależnościach)

R(Makler, Biuro, Akcjonariusz, Kapitał, Akcja, Dywidenda) 1PN

- Kapitał \rightarrow Dywidenda
- Akcjonariusz \rightarrow Makler
- { Akcjonariusz, kapitał } \rightarrow Akcja
- Makler \rightarrow Biuro

Schemat R nie jest w 3PN, bo nie jest w 2PN, bo nie wszystkie atrybuty niezależne są w pełni zależne od tłuszcza, np. Dywidenda jest zależna tylko od kapitału.

KK: { Akcjonariusz, kapitał }

R1 (Akcjonariusz, Makler ~~Biuro~~, Biuro) 2PN

*R2 (Kapitał, Dywidenda) 3PN

*R3 (Akcjonariusz, Kapitał, Akcja) 3PN

KK: { Akcjonariusz, Makler }

*R11 (Akcjonariusz, Makler) 3PN

*R12 (Makler, Biuro) 3PN

42. **Selekcja** – operator jednoargumentowy oznaczający wybór z relacji krotek spełniających zadany warunek dotyczący atrybutów danej wartości

σ_F - operator selekcji krotek spełniających formułę F,

43. **Projekcja //rzutowanie//** - operator jednoargumentowy polegający na wyborze z danej relacji R określonego zbioru atrybutów Z, np. z relacji Pracownicy wybieramy Nazwisko, Imię i Pensje

π_Z - operator projekcji (rzutowania) na zbiór atrybutów Z,

44. **Złączenie//join//** - operator dwuargumentowy, polega na złączeniu krotek z dwóch relacji posiadających **atrybuty o takich samych dziedzinach**(przecież jak łączę sobie np. tabelę Nauczyciele z tabelą Przedmioty to gdzie tam atrybuty o takich samych dziedzinach? O co z tym chodzi?). Z relacji wynikowej usuwane są powtarzające się krotki.

$\bowtie \theta$ - operator teta-złączenia.

JOIN - operator złączenia naturalnego,

//ocb z teta? jakis warunek? //dokładnie, teta-złączenie to złączenie pod jakimś warunkiem //masz może jakiś przykład?

//Przy ilorazie to tłumacząc łopatologicznie:

1. Masz kolumny A i B z powiedzmy relacji R1

2. Dzielisz przez kolumnę B z relacji R2
 3. Wynikowo dostajesz te wartości z kolumny A z R1 które miały krotki z wszystkimi wartościami z dziedziny B z R2
- $AB / B = A$

Iloraz – operator dwuargumentowy

Iloraz: przykład i interpretacja

Operacja dzielenia relacji R przez relację S ma istotne znaczenie praktyczne i stanowi rozszerzenie operacji antyprojekcji. W przypadku operacji antyprojekcji, można było uzyskać odpowiedź na pytanie typu:

Którzy dostawcy dostarczają wszystkie produkty?

Natomiast w przypadku operacji ilorazu można uzyskać odpowiedź na pytanie typu:

Którzy dostawcy dostarczają produkty z określonego zbioru?

Interesujący nas zbiór zadawany jest właśnie relacją S . Poniżej przedstawiono prosty przykład.

Przykład:

<i>Dostawca</i>	<i>Element</i>
<i>Budex</i>	<i>cegła</i>
<i>Budex</i>	<i>pustak</i>
<i>Budex</i>	<i>cement</i>
<i>Budex</i>	<i>piasek</i>
<i>Matbud</i>	<i>pustak</i>
<i>Matbud</i>	<i>cement</i>
<i>Matbud</i>	<i>gips</i>
<i>Matbud</i>	<i>piasek</i>
<i>Probud</i>	<i>cegła</i>
<i>Probud</i>	<i>cement</i>
<i>Probud</i>	<i>piasek</i>
<i>Probud</i>	<i>żwir</i>

<i>Element</i>
<i>cegła</i>
<i>cement</i>
<i>piasek</i>

÷

=

<i>Dostawcy</i>
<i>Budex</i>
<i>Probud</i>

Operacja dzielenia pozwala zatem na realizację zapytań formalizowalnych logicznie z zastosowaniem kwantyfikatora ogólnego (\forall).

45. **Suma** – dla relacji o takich samych atrybutach. Wynik: relacja, której krotki są sumą teoriomnogościową krotek rel wyjściowych
46. **Różnica** – tak samo, tylko wynik jest relacją której krotki są różnicą teoriomnogościową
47. **Iloczyn kartezjański** – dwuargumentowy, dla relacji o różnych nazwach atrybutów. Relacja wynikowa zawiera atrybuty jednej i drugiej relacji. Zbiór jej krotek powstaje poprzez utworzenie iloczynu kartezjańskiego krotek dwu relacji, a następnie każda krotka, która jest parą krotek z relacji wyjściowych zostaje zastąpiona konkatenacją tych krotek, tzn. kiedy k_1 należy do R_1 , k_2 należy do R_2 to do iloczynu kartezjańskiego należy krotka utworzona ze składowych krotek k_1 i k_2 .

8. iloczyn kartezjański – operator dwuargumentowy określony dla relacji o różnych nazwach atrybutów.

$R_1 \times R_2$ Relacja wynikowa zawiera atrybuty obu relacji. Zbiór krotek relacji wyjściowej powstaje przez utworzenie iloczynu kartezjańskiego krotek obu relacji wyjściowych, następnie każda para zostaje zastąpiona łąčeniem tych krotek. Łączenie relacji nie zawiera "każdy z każdym".

Zadanie 2.

$R_1(A, B) = \{(1, 1), (1, 2), (2, 3), (0, 2)\}$ $R_2(A, B) = \{(1, 1), (1, 3), (2, 1), (0, 2), (2, 2)\}$

A	B
1	1
1	2
2	3
0	2

A	B
1	1
1	3
2	1
0	2
2	2

a) $R_1 \cup R_2 = \{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (0, 2)\}$

b) $R_1 \cap R_2 = \{(1, 1), (0, 2)\}$

c) $R_1 - R_2 = \{(1, 2), (2, 3)\}$

d) $R_2 - R_1 = \{(1, 3), (2, 1), (2, 2)\}$

e) $R_1 \times R_2' = \{(1, 1, 1, 1), (1, 1, 1, 3), (1, 1, 2, 1), (1, 1, 0, 2), (1, 1, 2, 2), (1, 2, 1, 1), (1, 2, 1, 3), (1, 2, 2, 1), (1, 2, 0, 2), (1, 2, 2, 2), (2, 3, 1, 1), (2, 3, 1, 3), (2, 3, 2, 1), (2, 3, 0, 2), (2, 3, 2, 2), (0, 2, 1, 1), (0, 2, 1, 3), (0, 2, 2, 1), (0, 2, 0, 2), (0, 2, 2, 2)\}$

Wymaga przemianowania atrybutów.
 $RENAME(R_2: A \text{ AS } C, B \text{ AS } D)$
 $AS R_2'$

f) $R_1 \text{ JOIN } R_2 = R_1 \cap R_2 = \{(1, 1), (0, 2)\}$ ponieważ $sch(A) = sch(B)$

g) $\pi_{A < B}(R_1) = \{(1, 2), (2, 3), (0, 2)\}$

h) $\pi_{A=A \vee B=2}(R_1) \cup \pi_{A \leq B, A}(R_1) = \{(1, 1), (1, 2), (0, 2)\} \cup \{(1, 2), (2, 3)\} = \{(1, 1), (1, 2), (2, 3), (0, 2)\}$

// o co chodzi z przemianowaniem atrybutów

BD, Ćwiczenia 3. cd.

3. b) • nazwiska i imiona wszystkich studentów i pracowników: $\Pi_{\{Nazwisko, Imie\}} (STUDENCI) \cup \Pi_{\{Nazwisko, Imie\}} (PRACOWNICY)$
 • dane par studentów o tych samych nazwiskach
 1) $RENAME (STUDENCI: Indeks, A: Indeks', Nazwisko A: Nazwisko', Imie A: Imie, Data-ur A: Data-ur') AS STUDENCI'$
 2) $STUDENCI \bowtie (Nazwisko = Nazwisko' \wedge Indeks < Indeks') STUDENCI'$

- indeks, nazwisko, imię studentów, którzy zaliczyli algebrę na 4

$$\Pi_{\{Indeks, Nazwisko, Imie\}} (STUDENCI \text{ JOIN } (Ocena = 4 (ZALICZENIA \text{ JOIN } (Nazwa = 'Algebra' (PRZEDMIOTY)))))$$

- nazwy przedmiotów zaliczonych przez co najmniej jednego studenta $\Pi_{\{Nazwa\}} (Ocena \geq 3 (PRZEDMIOTY \text{ JOIN } ZALICZENIA))$

- numery przedmiotów zaliczone przez wszystkich studentów

$$\Pi_{\{Numer, Indeks\}} (Ocena \geq 3 (ZALICZENIA)) \div \Pi_{\{Indeks\}} (STUDENCI)$$

- numery indeksów studentów, którzy zaliczyli wszystkie przedmioty

$$\Pi_{\{Indeks, NrPredm\}} (Ocena \geq 3 (ZALICZENIA)) \div \Pi_{\{NrPredm\}} (PRZEDMIOTY)$$

4. a) Zastępowanie $R3 = R1 \text{ REPLACE } R2$

$R1 (A_1, \dots, A_N, Z_1, \dots, Z_k)$

$R2 (Z_1, \dots, Z_{k'}, B_1, \dots, B_M) \quad M, N, k, k' \in \mathbb{N}$

$R3 = \Pi_{\{A_1, \dots, A_N, Z_1, \dots, Z_{k'}, B_1, \dots, B_M\}} (R1 \text{ JOIN } R2)$

JOIN Rel1 AND Rel2 OVER L.atr GIVING Wynik

przydatne operandy: NOT, isNull(...)

JOIN tabela1 AND tabela2 OVER wyplnne - kolumna GIVING wynik

↑
zwraca unikalny set

} 3PN

ROB

$$R = 6_{\text{addres}, \text{wires}}(0)$$

HYNIK

$$R1 = \bar{11} \text{ NfD}_3 \text{ Na.22m'sko} \quad (D)$$
$$H = \prod_{N \in D} N_{\text{Zamirsko}}(G_{\text{Adres} = \text{"Wrocław"}}(D))$$

Nowasko GIVING WYNIK

GET Wynik(L.atr) : Warunek

Język predykatów:

RANGE nazwa-tabeli alias

przydatny operator: COUNT(...)

GET Wynik(nazwa-tabeli.nazwa-kolumny) : Warunek

gdzie warunek \forall alias (Warunki) lub \exists alias (Warunki)

- ☐ Ad b)
Pokaż pełne dane o dostawcach

GET H(D)

- ☐ Pokaż nr, nazwisko dostawców z Wrocławia

GET H(D.NrD, D.Nazwisko)

GET H(D.NrD, D.Nazwisko) : D.Adres = 'Wrocław'

H(1)

↑

LIMIT (ile rekordów wyświetlić)

Sortowanie wyników
UP D.NrD
DOWN

- ☐ Pokaż nazwiska dostawców dostarczających część C1

RANGE DC X (zmienna zakresu)

GET H(D.NrD, D.Nazwisko) : $\exists X (X.NrD = D.NrD \wedge X.NrC = 'C1')$

dostawca dostarcza jakąkolwiek część

zmienna zakresu - zastosowanie

Dostawcy (NrD, Nazwisko, Adres)

Pokaż nr, nazwisko dostawcy

RANGE Dostawcy D

GET (D.NrD, D.Nazwisko)

□ Podaj nazwiska dostawców zielonych części

GET Wynik (D.NrD, D.Nazwisko); $\exists DCX (DCX.NrD = D.NrD \wedge \exists CX (DCX.NrC = CX.NrC \wedge CX.kolor = zielony))$

RANGE DC DCX
RANGE C CX

□ Podaj nazwiska dostawców nie dostarczających części C1

RANGE DC DCX
GET W(D.Nazwisko); $\forall DCX (DCX.NrD \neq D.NrD \vee DCX.NrC \neq C1)$
 $:\neg (\exists DCX (DCX.NrD = D.NrD \wedge DCX.NrC = C1))$

Podaj nazwiska dostawców wszystkich części

RANGE C CX
RANGE DC DCX
GET W(D.Nazwisko); $\forall CX \exists DCX (DCX.NrD = D.NrD \wedge DCX.NrC = CX.NrC)$

□ Podaj nazwiska dostawców co najmniej jednej części dostarczanej przez dostawcę D2

DC		
NrD	NrC	Wartość
D1	C1	
D2	C1	
D2	C2	
D3	C4	

D2 → C1, C2
↗ ↘

RANGE DC DCX
RANGE DC DCY
GET W(D.Nazwisko); $\exists DCX (DCX.NrD = D.NrD \wedge \exists DCY (DCY.NrC = DCX.NrC \wedge DCY.NrD = D2))$
do D2

□ Aktualizacja danych

- dopisz nowego dostawcę

W.NrD = 'D10'
W.Nazwisko = 'Kowalski'
W.Adres = 'Opole'
PUT W(D)

- usunięcie dostawcy D1

HOLD W(D) \wedge D.NrD = 'D1'
DELETE W

HOLD W(D) \wedge
DELETE W

□ Funkcje biblioteczne

Podaj liczbę dostawców

GET W(COUNT(D.NrD)); D.Adres = 'wrocław'
MAX
MIN
AVG (?)

Query by example

SELECT:

tabela	kol1	kol2	...
P.			

tabela	kol1	kol2	...
	P.		

SELECT * FROM tabela

SELECT kol1 FROM tabela

WHERE:

tabela	kol1	kol2
P.	= 'x'	> 10

SELECT * FROM tabela

WHERE kol1 = 'x' and kol2 > 10

tabela	kol1	kol2
P.K	= war1	
K	= war2	

dwie definicje
zmiennej ten
sam warunek

SELECT kol1 FROM tabela

WHERE kol2 = war1 AND

kol1 IN (SELECT kol1 FROM tabela

WHERE kol2 = war2)

tabela	kol1	kol2
	X	P.K
	≠ X	K

SELECT kol2 FROM tabela

WHERE kol1 = X AND

kol2 IN (SELECT kol2 FROM tabela

WHERE kol1 != X)

tabela	kol1	kol2
	P	X
	wartosc	X

SELECT kol1 FROM tabela

WHERE kol2 IN

(SELECT kol2 FROM tabela

WHERE kol1 = wartosc)

5. a)

Kasety	NrKasety	Tytul	Rok
		P.	

SELECT Tytul FROM Kasety

SQL-procedura

b)

Klienci	NrK	Imie	Nazwisko	Adres
		P.	P.	

SELECT Imie, Nazwisko FROM Klienci

c)

Kasety	NrKasety	Tytul	Rok
	link	P.	

SELECT Tytul FROM Kasety

WHERE NrKasety NOT IN

(SELECT NrKasety FROM Wypożyczenia)

Wypożyczenia	NrK	NrKasety	DataWyp	DataZw
		link		

d)

Klienci	NrK	Imie	Nazwisko	Adres
		P.	P. = 'Kowalski'	P.

SELECT Imie, Nazwisko, Adres

FROM Klienci

WHERE Nazwisko = 'Kowalski'

e)

Klienci	NrK	Imie	Nazwisko	Adres
	link1		P.	

Kasety	NrKasety	Tytul	Rok
	link2	P.	

SELECT Nazwisko FROM Klienci WHERE NrK IN

(SELECT NrK FROM Kasety JOIN Wypożyczenie

ON Kasety.NrKasety = Wypożyczenie.NrKasety

WHERE Tytul = 'Lalka')

Klienci	NrK	Imię	Nazwisko	Adres
		P.	P. = 'Kowalski'	P.

```
SELECT Imię, Nazwisko, Adres
FROM Klienci
WHERE Nazwisko = 'Kowalski'
```

e)

Klienci	NrK	Imię	Nazwisko	Adres
	link1		P.	

Kasety	NrKasety	Tytuł	Rek.
	link2	P. = 'Pan Tadeusz'	

Wypożyczenia	NrK	NrKasety	DataWyp	DataZw
Lp	link1	link2		

```
SELECT Nazwisko FROM Klienci WHERE NrK IN
(SELECT NrK FROM Kasety JOIN Wypożyczenia
ON Kasety.NrKasety = Wypożyczenia.NrKasety
WHERE Tytuł = 'Lutka')
```

f)

Klienci	NrK	Imię	Nazwisko	Adres
	link1			

Wypożyczenia	NrK	NrKasety	DataWyp	DataZw
Lp	link1			
link2				

Warunek
COUNT(link2) >= 3

```
SELECT Nazwisko FROM Klienci
WHERE ((SELECT COUNT(Lp) FROM Wypożyczenia
WHERE Klienci.NrK = Wypożyczenia.NrK) >= 3)
```

Thema Bazy Danych

Teilnehmer Ćwiczenia 6. cd.

Ort 5. cd.

Datum

Seite

Notizen

Deutschsprachige SAP® Anwendergruppe

h)

Klienci	NrK	Imię	Nazwisko	Adres
	link		P.	

Wypożyczenia	Lp	NrK	NrKasety	DataWyp	DataZw
		link			

```
SELECT Nazwisko FROM Klienci
WHERE NrK NOT IN
(SELECT NrK FROM Wypożyczenia)
```

i)

Klienci	NrK	Imię	Nazwisko	Adres
	link1		P.	

Wypożyczenia	Lp	NrK	NrKasety	DataWyp	DataZw
	link2	link1			
	link3	link4			

```
SELECT Nazwisko FROM Klienci WHERE
(SELECT MAX(COUNT(Lp)) FROM Wypożyczenia
GROUP BY NrK) - pogrupuj, zlicz dla w grupie, weź max
=
(SELECT COUNT(Lp) FROM Wypożyczenia
WHERE NrK = Wypożyczenia.NrK)
Klienci.
```

Warunek

MAX(COUNT(link3)) = COUNT(link2)

6. a)

Klienci	NrK	Imię	Nazwisko	Adres
		= Adam	= Kowalski	U = Nasty Adres

Ad d) QbE

☐ Podaj numery wszystkich dostarczanych części

DC	NrD	NrC	Ilość
		P.	

☐ Podaj pełną informację o dostawcach

D	NrD	Nazwisko	Adres
	P.NrD	P.N	P.A

lub P.

☐ Podaj numery, nazwiska dostawców z Qpola

D	NrD	Nazwisko	Adres
	P.NrD	P.N	= Qpole

C	Masa
	> 1000

Podaj numery części w kolorze zielonym (i) masie większej od 100

C	NrC	Nazwa	Kolor	Masa	Miasto
	P.N		= zielony	> 100	(lub)
	P.N		= zielony	> 100	
	P.N				

Podaj numery dostawców części C1 i C2

DC	NrD	NrC	Ilość
	P.N	C1	
	N	C2	

Podaj nazwiska dostawców części C1
(nie dostarczających)

D	NrD	Nazwisko	Adres
	DX	P.N	

DC	NrD	NrC	Ilość
(1)	DX	C1	

Podaj nazwiska dostawców co najmniej jednej części jaka pochodzi od D2.

DC	NrD	NrC	Ilość
	DX	CX	
	D2	CX	

Podaj nr części pochodzących od co najmniej dwóch dostawców.

DC	NrD	NrC	Ilość
	DX	PNC	
	#DX	NC	

Podaj nr dostawców wszystkich części

C	NrC	...
	ALL.CX	

Ramiona warunków
DX #

DC	NrD	NrC	wartość
	P.N	ALL.CX	

Aktualizacja danych

D	NrD	Nazwisko	Adres
	ND	'Kowalski'	
UPDATE	ND	'Nowak'	

DC	NrD	NrC	wartość
	D1	C1	I
UPDATE	D1	C1	I + 10

D	NrD	Nazwisko	Adres
Insert	D7	'Kowalski'	'Nysa'
delete	D9		

Podaj wartość wszystkich dostawców

D	NrD	
	P.Count, ALL.DX	

Podaj nr dostawców aktualnie dostarczających części

DC	NrD	NrC
	P.Count, U, ALL.DX	
	↑	
	UNIQUE	

Podaj wartość dostawców części C1

DC	NrD	NrC
	P.Count, ALL.DC	'C1'

D	NrD	NrC	wartość
		C1	P.SUM, ALL.I

SUM, MAX, MIN, AVG

51. SQL

```
SELECT kolumny  
FROM tabele  
WHERE warunek-selekcji  
GROUP BY kolumny-grupowane  
HAVING warunek-dotyczący-pogrupowanych-danych  
ORDER BY kolumny-klucze-sortowania  
;
```

4d c) SQL

- ☐ Podaj numery wszystkich dostawców

```
SELECT NrD FROM D
```

- ☐ Podaj numery części aktualnie dostarczanych

```
SELECT UNIQUE NrC FROM DC
```

- ☐ Podaj numery, nazwiska dostawców z Wrocławia lub Opola

```
SELECT NrD FROM D WHERE Adres = 'Wrocław' OR Adres = 'Opole'
```

Nie musi być cała nazwa, tylko rzeź.

```
SELECT NrD, Nazwisko  
FROM D
```

```
WHERE Adres = 'Wrocław'
```

```
ORDER BY Nazwisko ASC, (sortowanie rosnące)  
NrD DESC (malejące)
```

Podaj nazwiska dostawców części C1

```
SELECT Nazwisko FROM D AS D1  
FROM D WHERE NrD IN  
(SELECT NrD FROM DC  
WHERE NrC = 'C1')
```

```
SELECT Nazwisko  
FROM D, DC  
WHERE D.NrD = DC.NrD  
and NrC = 'C1'
```


Podaj nazwiska dostawców zielonych części

```
SELECT Nazwisko
FROM D (NOT)
WHERE NrD IN
( SELECT NrD
FROM DC
WHERE NrC IN
( SELECT NrC
FROM C
WHERE Kolor = 'zielony' ) )
```

dostawcy części zielonych
części zielone

```
SELECT Nazwisko
FROM D, DC, C
WHERE D.NrD = DC.NrD and
DC.NrC = C.NrC and
Kolor = 'zielony'
```

Podaj nazwiska dostawców nie dostarczających zielonych części
• NOT IN zamiast IN

Podaj numery dostawców co najmniej jednej części, jaka pochodzi od dostawcy D2.

```
SELECT NrD
FROM DC
WHERE NrC IN
( SELECT NrC FROM DC
WHERE NrD = 'D2' )
```

zb. części, które produkuje D2

$D2 = \{C1, C2, C3\}$

Podaj nazwiska dostawców wszystkich części ($A = B$)

```
SELECT Nazwisko
FROM D
WHERE ( SELECT NrC
FROM DC
WHERE NrD = D.NrD )
= ( SELECT NrC
FROM C )
```

zb. części dostarczonych = zb. wszystkich części

Podaj numery dostawców co najmniej takich części jakie pochodzą od dostawcy D2.
($A \supset B$)
zawieranie

$D2 = \{C1, C2, C3\}$

```
SELECT UNIQUE NrD
FROM DC DCX
WHERE ( SELECT NrC
FROM DC
WHERE NrD = DCX.NrD )
CONTAINS
( SELECT NrC
FROM DC
WHERE NrD = 'D2' )
```

A
B
części pochodzące od D2

```
SELECT NrD
FROM DC
GROUP BY NrD
HAVING SET (NrC)
CONTAINS
( SELECT NrC
FROM DC
WHERE NrD = 'D2' )
```

GRUPOWANIE:

DC	NrD	NrC	Ilość
D1	D1	C1	
	D1	C2	
D2	D2	C1	
	D2	C2	
	D1	C3	
	D2	C3	
	D1	C4	
D5	D5	C1	

D1: C1 C2 C3 C4 \supset C1, C2, C3
D5:

numery
Podaj ~~numer~~ dostawców, którzy aktualnie nie dostarczają żadnych części (A-B)

```
(SELECT NcD
FROM D )
MINUS
(SELECT NcD
FROM DC)
```

średnia arytmetyczna

□ Funkcje biestandardne: COUNT, MAX, MIN, AVG, SUM
Podaj liczbę wszystkich dostawców

```
SELECT COUNT(NcD)
FROM D
WHERE Adres = 'Opole'
```

DISTINCT (bez powtórzeń)

```
SELECT SUM(ilosc)
FROM DC
WHERE NcD = 'C2'
```

□ Aktualizacje danych

- Zmiana kolor części C1 na zielony
UPDATE C
SET Kolor = 'zielony'
WHERE NcD = 'C1'
- Dodawanie wrotek
INSERT INTO D
<'05', Kawasaki, Opole>
- Usuwanie
DELETE D
WHERE NcD = 'D1'

<<KORZYSTANIE Z W/W JĘZYKÓW>>

POUFNOŚĆ W SYSTEMACH BAZ DANYCH

1. Sterowanie dostępem, przyznanie praw dostępu.

- identyfikacja użytkowników,
- przyznawanie prawa dostępu do określonego zbioru danych.

- a) read - odczyt
- b) insert - wprowadzanie danych
- c) update - edycja
- d) delete - usuwanie
- e) drop - usuwanie całych tabel
- f) index - edycja indeksów

GRANT ALL [PRIVILEGES] | lista przywilejów
ON [TABLE] | lista tabel
TO PUBLIC | lista użytkowników
[WITH GRANT OPTION] (możliwość przekazywania uprawnień)

Przyznanie wszystkich praw do tabeli uczniowie dla trzech osób:

GRANT ALL
ON Nauczyciele
TO MArcin, Łukasz, Piotr

Przyznanie praw do odczytu tabeli książki dla wszystkich użytkowników:

GRANT READ
ON Książki
TO PUBLIC

Przyznanie praw kierownikowi:

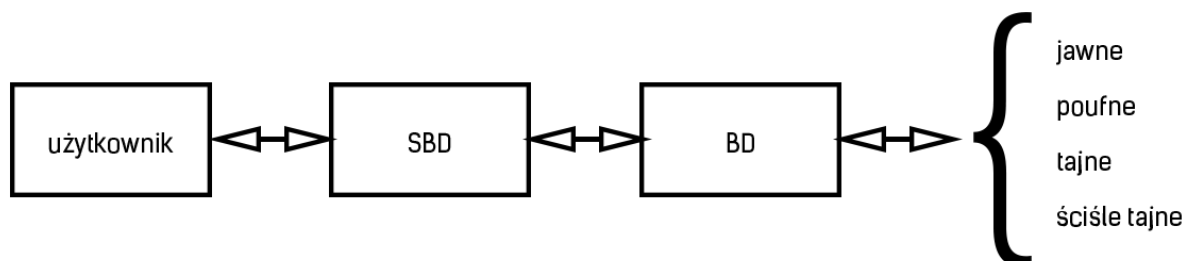
GRANT UPDATE, READ, INSERT
ON Pracownicy
TO Kierownik
WITH GRANT OPTION (możliwość przekazywania uprawnień podwładnym)

Cofanie uprawnień:

REVOKE ALL [PRIVILEGES]	lista przywilejów
ON [TABLE]	lista tabel
FROM PUBLIC	lista użytkowników

REVOKE INSERT, UPDATE
ON Uczniowie
FROM Piotr, Marcin

2. Sterowanie przepływem danych.



Użytkownik posiadający uprawnienia niższej warstwy posiada również uprawnienia to wszystkich warstw poniżej, tj.:

- jawne - dostęp do danych jawnych BD, odmowa dla danych poufnych, tajnych i ściśle tajnych,
- poufne - dostęp do danych jawnych i poufnych BD, odmowa dla danych tajnych i ściśle tajnych,
- tajne - dostęp do danych jawnych, poufnych i tajnych BD, odmowa dla danych ściśle tajnych,
- ściśle tajne - dostęp do danych jawnych, poufnych, tajnych i ściśle tajnych BD.

Jeśli dane są:

- jawne - dostęp mają użytkownicy o uprawnieniach jawnych, poufnych, tajnych i ściśle tajnych,
- poufne - dostęp mają użytkownicy o uprawnieniach poufnych, tajnych i ściśle tajnych, odmowa dla użytkowników z uprawnieniami jawnymi,
- tajne - dostęp mają użytkownicy o uprawnieniach tajnych i ściśle tajnych, odmowa dla użytkowników z uprawnieniami jawnymi, poufnymi,
- ściśle tajne - dostęp mają użytkownicy o uprawnieniach ściśle tajnych, odmowa dla użytkowników z uprawnieniami jawnymi, poufnymi i tajnymi.

3. Ograniczenie - możliwości warunkowe

- statystyczne bazy danych

```
SELECT COUNT (*)  
FROM ...  
WHERE ...
```

Wynikiem jest liczba rekordów spełniających dane kryteria!

Przykład:

Pacjenci:

Nazwisko	Płeć	Wiek	Stan cywilny	...
----------	------	------	--------------	-----

Formułując odpowiednie pytania jesteśmy w stanie wydedukować odpowiedź na pytanie stopniowo zawężając kryteria wyszukiwania:

Zapytanie 1: Ilu chorych?

- 1) Płeć: męska
- 2) Wiek 45-50 lat
- 3) Żonaty

Odpowiedź systemu: 100 osób

Zapytanie 2: Ilu chorych?

- 1) Płeć: męska
- 2) Wiek 45-50 lat
- 3) Żonaty
- 4) Dyplom zdobyty w Tokio

Odpowiedź systemu: 10 osób

Zapytanie 3: Ilu chorych?

- 1) Płeć: męska
- 2) Wiek 45-50 lat
- 3) Żonaty
- 4) Dyplom zdobyty w Tokio
- 5) Chory na AIDS

Odpowiedź systemu: 0/1 – nie jest / jest chory na AIDS

Pytamy tak długo, aż uzyskamy odpowiedź tak/nie – 1/0.

Jak się zabezpieczyć ?

- a) Udzielanie odpowiedzi tylko do progowej wartości odpowiedzi, tj. gdy liczba m odpowiedzi jest większa ustalonego k: $m > k$ (np. $k = 10$)
- b) Prowadzenie dziennika wprowadzanych kwerend przez użytkownika (podejrzliwość co do wprowadzanych kwerend, nadmiernego ruchu i sekwencji powtarzających się zapytań od danego użytkownika). Monitoringu ruchu w bazie danych.

4. Szyfrowanie danych

- a) Szyfry przestawieniowe - szyfr Cezara – przestawiamy litery wg określonego schematu,
- b) Algorytm DES
- c) Algorytm RSA
- d) Podpis elektroniczny

5. Zabezpieczenie bazy danych:

- ograniczanie dostępu,
- nadawanie i odbieranie uprawnień,
- stosowanie odpowiednich zabezpieczeń.