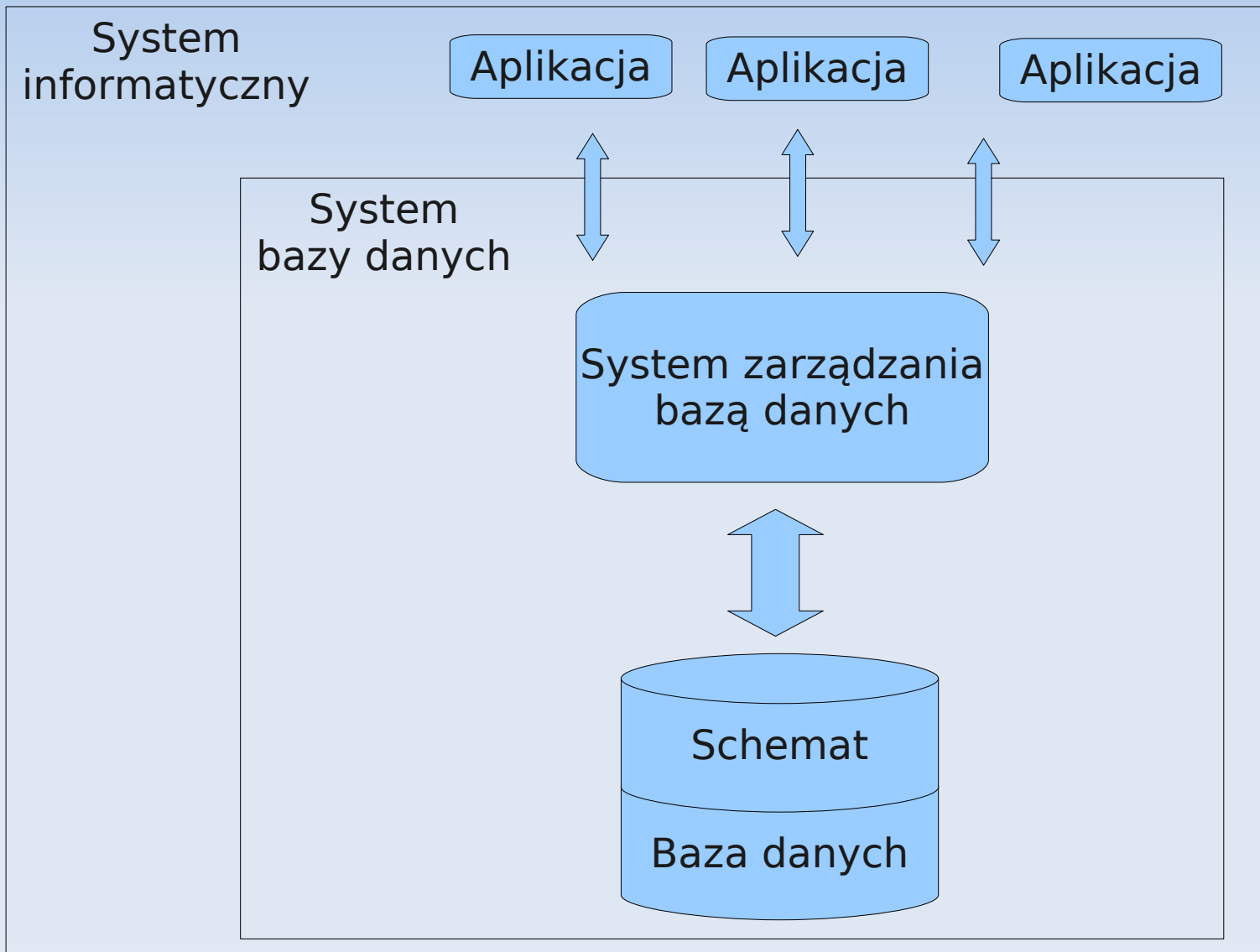


Podstawy języka SQL

Adam Cakudis
IFP UAM

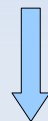
Użytkownicy



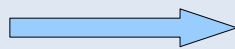
Konceptcje relacyjnych baz danych

Tabela (relacja)

atrybut



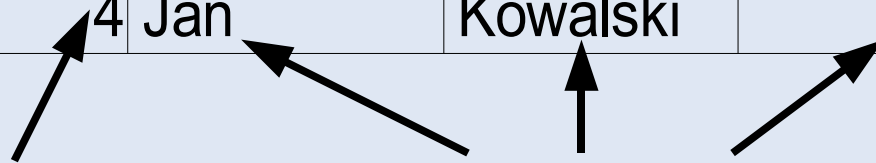
krotka
(wiersz,
rekord)



id	Imie	Nazwisko	Rurodz
1	Jan	Nowak	1959
2	Artur	Kowalski	1962
3	Maciej	Kucharski	1948
4	Jan	Kowalski	1981

klucz główny

wartość



SQL

Structured Query Language – strukturalny język zapytań używany do definiowania, modyfikowania i usuwania struktur danych oraz do umieszczania, pobierania, modyfikowania i usuwania danych z baz danych. Język SQL pozwala również na zarządzanie transakcjami i mechanizmami autoryzacji dostępu do danych.

Cechy SQL

- Język SQL jest zorientowany na przetwarzanie zbiorów
- SQL jest językiem deklaratywnym

Język deklaratywny

Język deklaratywny opisuje „co” ma być zrobione, ale nie „jak”

Język deklaratywny a proceduralny

Język proceduralny

Program Dziecko wersja 1.0

Procedura sprzątnia:

1. podnieś zabawkę z podłogi
2. zanieś do swojego pokoju
3. podejdź do półki
4. połóż zabawkę na pierwszym wolnym miejscu

Język deklaratywny a proceduralny

Język deklaratywny
Program Dziecko wersja 2.0

Przykład ogólnego polecenia:

Posprzątaj zabawki

Przykład polecenia szczegółowego:

**Pozbieraj zabawki i postaw
samochody na półce a klocki wrzuć
do pudełka**

Podział języka SQL

- DML (ang. Data Manipulation Language)
- DDL (ang. Data Definition Language)
- DCL (ang. Data Control Language)

DML

Język Manipulacji Danymi służy do operacji na danych – do ich umieszczania w bazie, kasowania, przeglądania, zmiany.

Najważniejsze polecenia z tego zbioru to:

- SELECT – pobranie z bazy danych,
- INSERT – umieszczenie danych w bazie,
- UPDATE – zmiana danych,
- DELETE – usunięcie danych z bazy

DDL

Język Definicji Danych służy do operowania na strukturach, w których dane są przechowywane

Polecenia z tego zbioru:

- CREATE – utworzenie struktury (np. tabeli /relacji, indeksu itp.)
- DROP – usunięcie struktury
- ALTER – zmiana struktury (np. dodanie kolumny do tabeli)

DCL

- Język Kontroli nad Danymi do nadawania uprawnień do obiektów bazodanowych. Najważniejsze polecenia w tej grupie to:
 - GRANT – przyznanie praw
 - REVOKE – odebranie praw
 - COMMIT – potwierdzenie transakcji
 - ROLBACK - wycofanie transakcji

Standardy języka SQL

- SQL86, SQL89 – historyczne
- SQL92 – najpowszechniej implementowany standard ISO/IEC 9075:1995, trzy poziomy zgodności:
 - wejściowy
 - pośredni
 - pełny
- SQL99, SQL2003 – najnowsze standardy

Cechy języka SQL

- Nie rozróżnia się małych i dużych liter
- Ignorowane są znaki końca linii
- Każde polecenie SQL powinno być zakończone średnikiem

```
SELECT imie, nazwisko FROM klienci;
```

jest równoznaczne

```
SELECT  
imie, nazwisko  
FROM KLIENCI;
```

Przykładowa baza

KSIAZKA

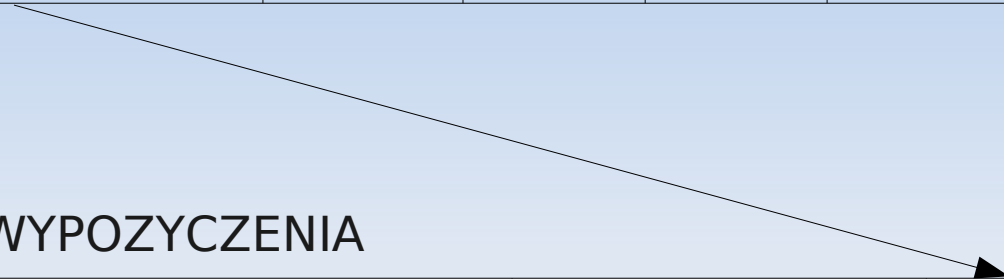
id_ksiazki	autor	tytul	isbn	cena_zakupu
------------	-------	-------	------	-------------

WYPOZYCZENIA

id_wypozyczenia	id_czytelnika	id_ksiazki	data	data_zwrotu
-----------------	---------------	------------	------	-------------

CZYTELNIK

id_czytelnika	imie	nazwisko	ulica	kod	miasto	telefon
---------------	------	----------	-------	-----	--------	---------



Przykładowa baza

KSIAZKA

id_ksiazki	autor	tytul	isbn	cena_zakupu
1	Ewa Lipska	Sklepy zoologiczne	830803117X	20,00
2	Andrzej Stasiuk	Fado	8389755750	29,00
3	Jerzy Pilch	Miasto utrapienia	837391370X	34,90
4	Wisława Szymborska	Poczta literacka	8308030777	25,50

Przykładowa baza danych

CZYTELNIK

id_czytelnika	imie	nazwisko	ulica	kod	miasto	telefon
1	Alicja	Nowak	Niepodległości 115	00-979	Warszawa	(022) 202 94 11
2	Bolesław	Kowalski	Kościuszki 19	60-204	Poznań	(061) 849 62 21
3	Jan	Wiśniewski	Wieniawskiego 11	60-301	Poznań	

Przykładowa baza danych

WYPOZYCZENIA

id_wypozyczenia	id_czytelnika	id_ksiazki	data	data_zwrotu
1	2	3	02.04.2008	16.04.2008
2	1	4	23.04.2008	07.03.2008
3	1	1	23.04.2008	07.03.2008

Proste zapytania

Polecenie `SELECT` – pozwala na odczytywanie danych z bazy oraz wykonywanie na tych danych prostych obliczeń i przekształceń. Pobiera krotki z relacji w bazie danych, przetwarza je (opcjonalnie) i zwraca wynik w postaci zbioru odczytanych krotek, czyli tzw. relację wynikową.

Proste zapytania

Zdanie w języku naturalnym:
Pokaż mi wszystkie książki

Zdanie w języku SQL:

```
SELECT * FROM ksiazka;
```

id_ksiazki	autor	tytul	isbn	cena_zakupu
1	Ewa Lipska	Sklepy zoologiczne	830803117X	20,00
2	Andrzej Stasiuk	Fado	8389755750	29,00
3	Jerzy Pilch	Miasto utrapienia	837391370X	34,90
4	Wisława Szymborska	Poczta literacka	8308030777	25,50

Proste zapytania - projekcja

*Pokaż tytuły i numery ISBN wszystkich
książek*

```
SELECT tytuł, isbn FROM ksiazka;
```

tytuł	isbn
Sklepy zoologiczne	830803117X
Fado	8389755750
Miasto utrapienia	837391370X
Poczta literacka	8308030777

Proste zapytania - wyrażenia

```
SELECT imie||' '||nazwisko  
FROM czytelnik;
```

imie nazwisko
Alicja Nowak
Bolesław Kowalski
Jan Wiśniewski

Proste zapytania - wyrażenia

```
SELECT imie||' '||nazwisko  
AS Czytelnik,  
miasto AS „Miejsce Zamieszkania”  
FROM czytelnik;
```

Czytelnik	Miejsce Zamieszkania
Alicja Nowak	Warszawa
Bolesław Kowalski	Poznań
Jan Wiśniewski	Poznań

Proste zapytania – wartości puste

```
SELECT nazwisko, telefon  
FROM czytelnik;
```

nazwisko	telefon
Nowak	(022) 202 94 11
Kowalski	(061) 849 62 21
Wiśniewski	

Eliminacja duplikatów

```
SELECT miasto FROM czytelnik;
```

miasto
Warszawa
Poznań
Poznań

Eliminacja duplikatów

```
SELECT DISTINCT miasto FROM czytelnik;
```

miasto
Warszawa
Poznań

Sortowanie

```
SELECT nazwisko FROM czytelnik  
ORDER BY nazwisko;
```

nazwisko
Kowalski
Nowak
Wiśniewski

Sortowanie

```
SELECT nazwisko FROM czytelnik  
ORDER BY nazwisko ASC;
```

nazwisko
Kowalski
Nowak
Wiśniewski

Sortowanie

```
SELECT nazwisko FROM czytelnik  
ORDER BY nazwisko DESC;
```

nazwisko
Wiśniewski
Nowak
Kowalski

Sortowanie

```
SELECT nazwisko FROM czytelnik  
ORDER BY imie DESC;
```

nazwisko
Wiśniewski
Kowalski
Nowak

Sortowanie

```
SELECT nazwisko FROM czytelnik  
ORDER BY miasto, nazwisko DESC;
```

nazwisko
Wiśniewski
Kowalski
Nowak

Selekcja

```
SELECT nazwisko FROM czytelnik  
WHERE miasto='Poznań'  
ORDER BY nazwisko DESC;
```

nazwisko

Wiśniewski

Kowalski

Selekcja – operatory logiczne

- = – równe
- != – nie równe
- <> – nie równe
- > – większe
- >= – większe lub równe
- < – mniejsze
- <= – mniejsze lub równe

Selekcja – operatory logiczne

- BETWEEN ... AND ...
- IN (wartosc1, ..., wartoscN)
- LIKE (np. ... WHERE miasto LIKE 'P%' ;)
- IS NULL
(np. ... WHERE telefon IS NULL;)

Selekcja – zanegowane operatory logiczne

- NOT BETWEEN ... AND ...
- NOT IN (wartosc1, ..., wartoscN)
- NOT LIKE
(np. ... WHERE miasto NOT LIKE 'P%' ;)
- IS NOT NULL
(np. ... WHERE telefon IS NOT NULL ;)

Selekcja – warunki złożone

```
SELECT nazwisko FROM czytelnik  
WHERE miasto LIKE 'P%'  
AND imie='Jan';
```

nazwisko

Wiśniewski

Selekcja – warunki złożone

```
SELECT nazwisko FROM czytelnik  
WHERE miasto LIKE 'W%'  
OR imie='Jan'  
ORDER BY nazwisko DESC;
```

nazwisko
Wiśniewski
Nowak

Łączenie tabel

- relacja jeden do wielu

KSIAZKA

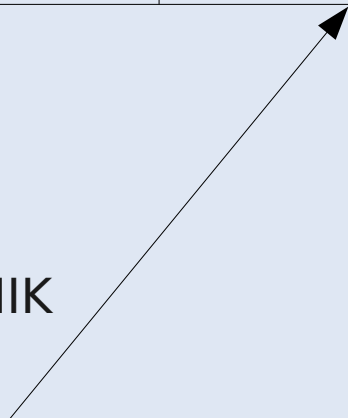
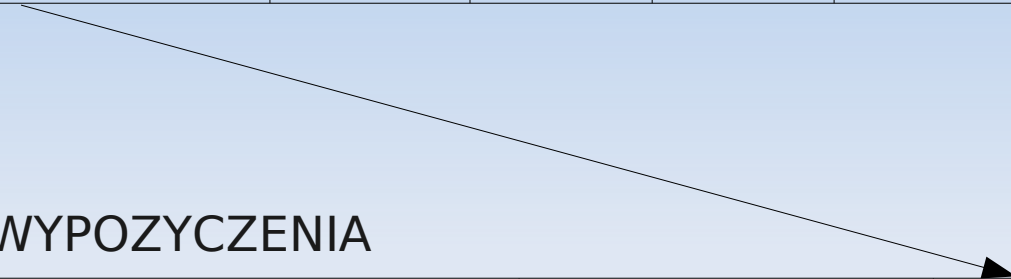
id_ksiazki	autor	tytul	isbn	cena_zakupu
------------	-------	-------	------	-------------

WYPOZYCZENIA

id_wypozyczenia	id_czytelnika	id_ksiazki	data	data_zwrotu
-----------------	---------------	------------	------	-------------

CZYTELNIK

id_czytelnika	imie	nazwisko	ulica	kod	miasto	telefon
---------------	------	----------	-------	-----	--------	---------



Łączenie tabel

Pytanie w języku naturalnym:
Kto wypożyczył książkę(i)
i kiedy zwróci?

Łączenie tabel

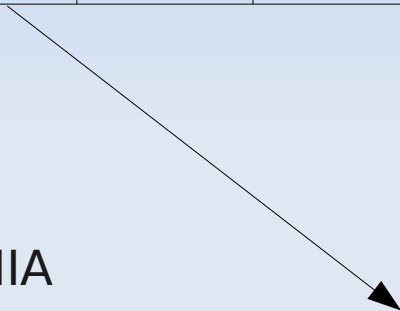
- relacja jeden do wielu

CZYTELNIK

id_czytelnika	imie	nazwisko	ulica	kod	miasto	telefon
---------------	------	----------	-------	-----	--------	---------

WYPOZYCZENIA

id_wypozyczenia	id_czytelnika	id_ksiazki	data	data_zwrotu
-----------------	---------------	------------	------	-------------



Łączenie tabel

SELECT (lista kolumn: tabela.kolumna)
FROM (lista tabel) WHERE (warunki
powiązania) AND (warunki wyboru wierszy)

```
SELECT czytelnik.nazwisko,  
wypozyczenia.data_zwrotu  
FROM czytelnik, wypozyczenia  
WHERE czytelnik.id_czytelnika=  
wypozyczenia.id_czytelnika;
```

Łączenie tabel

Pytanie w języku naturalnym:
Jakie książki zostały wypożyczone
i kiedy powinny być zwrócone?

Łączenie tabel

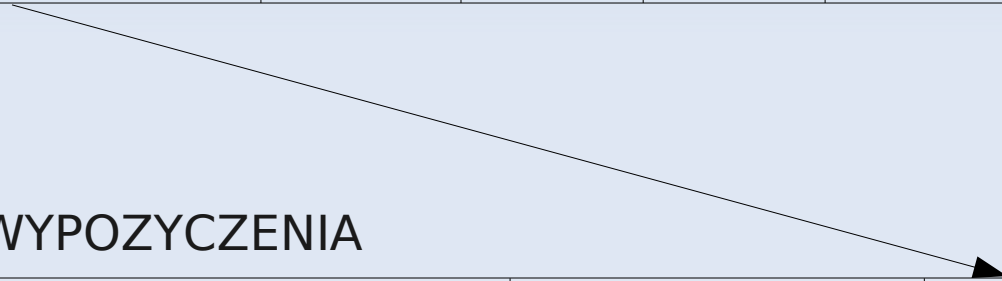
- relacja jeden do wielu

KSIAZKA

id_ksiazki	autor	tytul	isbn	cena_zakupu
------------	-------	-------	------	-------------

WYPOZYCZENIA

id_wypozyczenia	id_czytelnika	id_ksiazki	data	data_zwrotu
-----------------	---------------	------------	------	-------------



łączenie tabel

SELECT (lista kolumn: tabela.kolumna)
FROM (lista tabel) WHERE (warunki
powiązania) AND (warunki wyboru wierszy)

```
SELECT ksiazka.tytuł,  
wypozyczenia.data_zwrotu  
FROM ksiazka, wypozyczenia  
WHERE ksiazka.id_ksiazki=  
wypozyczenia.id_ksiazki;
```

Łączenie tabel

A jak zapytać:

Kto pożyczył książkę,
jaką książkę
i kiedy odda?

Łączenie tabel

SELECT (lista kolumn: tabela.kolumna)
FROM (lista tabel) WHERE (warunki
powiązania) AND (warunki wyboru wierszy)

```
SELECT czytelnik.nazwisko,  
ksiazka.tytuł,  
wypozyczenia.data_zwrotu  
FROM czytelnik, ksiazka, wypozyczenia  
WHERE czytelnik.id_czytelnika=  
wypozyczenia.id_czytelnika,  
AND ksiazka.id_ksiazki=  
wypozyczenia.id_ksiazki;
```

Powiązania zewnętrzne

Pokaż wszystkich czytelników i daty, kiedy wypożyczali książki, jeśli wypożyczali

Powiązania zewnętrzne

```
SELECT nazwisko FROM czytelnik  
ORDER BY nazwisko;
```

nazwisko
Kowalski
Nowak
Wiśniewski

Powiązania zewnętrzne

```
SELECT czytelnik.nazwisko,  
wypozyczenia.data_zwrotu  
FROM czytelnik, wypozyczenia  
WHERE czytelnik.id_czytelnika=  
wypozyczenia.id_czytelnika;
```

nazwisko	data_zwrotu
Kowalski	2.04.2008
Nowak	23.04.2008
Nowak	23.04.2008

Powiązania zewnętrzne

```
SELECT czytelnik.nazwisko  
FROM czytelnik  
WHERE czytelnik.id_czytelnika  
NOT IN  
(SELECT wypozyczenia.id_czytelnika  
FROM wypozyczenia);
```

nazwisko

Wiśniewski

Powiązania zewnętrzne

```
SELECT czytelnik.nazwisko,  
wypozyczenia.data_wyp  
FROM czytelnik  
LEFT JOIN wypozyczenia  
ON czytelnik.id_czytelnika=  
wypozyczenia.id_czytelnika;
```

nazwisko	data_zwrotu
Kowalski	2.04.2008
Nowak	23.04.2008
Nowak	23.04.2008
Wiśniewski	

Powiązania zewnętrzne

Ogólna postać:

Tabela chroniona



```
SELECT kolumny FROM tabela1  
LEFT JOIN tabelą2  
ON tabela1.kolumna = tabelą2.kolumna
```

Polecenia modyfikujące

- INSERT INTO nazwa_tabeli (lista kolumn)
VALUES (lista wartości)

Polecenia modyfikujące

- UPDATE [tabela] SET [kolumna] = wartość [, ...] WHERE warunek;

Polecenia modyfikujące

- **DELETE FROM nazwa_tabeli WHERE warunek**
(DELETE działa na wierszach, dlatego nie ma nazw kolumn w poleceniu)

Pytania?