

Wrocławska Wyższa Szkoła Informatyki Stosowanej

Charakterystyka języka SQL. Elementy obliczeń relacyjnych.

Dr hab. inż. Krzysztof Pieczarka

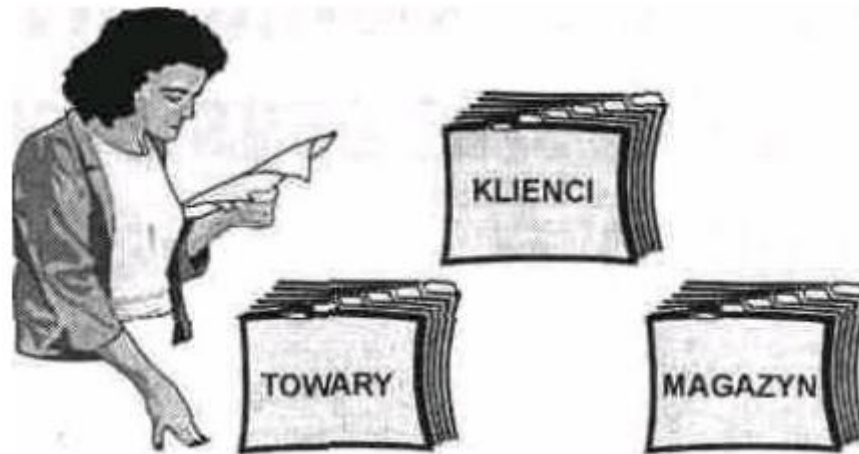
Email: krzysztof.pieczarka@gmail.com

Tradycyjne bazy danych

W czasie kiedy komputery nie były wykorzystywane tak powszechnie jak dzisiaj, informacje były gromadzone na papierze.

Pracownik firmy musiał ręcznie organizować dane. Zapisywał je, wyszukiwał, aktualizował itd.

Dzisiaj, oprócz operacji, które musi wykonać ręcznie, pozostałe wykonuje za pomocą funkcji, które dostarcza relacyjny system bazy danych.



Relacyjny system bazy danych

Relacyjny system bazy danych przechowuje wszystkie dane w tabelach.

Każda tabela zawiera dane na konkretny temat, np dane o klientach, dane o pracownikach, towarach itp.

System bazy danych zarządza tymi danymi, pozwala m.in. na szybsze ich wyszukanie i zorganizowanie.

Za każdym razem gdy potrzebujemy informacji z bazy danych, musimy „zapytać” system bazy danych w języku, który on rozumie.

Tym językiem jest SQL - Structured Query Language.

	NR_KLIENTA	IMIE	NAZWISKO	NR_KARTY_KREDYT	FIRMA	ULICA	NUM...	MIASTO	KOD	NIP	NR_TELEFONU
1	00000001	JAN	KOWALSKI	NULL	NULL	KOCHANOWSKIEGO	3	WROCLAW	37-300	NULL	167-763-234
2	00000002	TOMASZ	ADAMCZAK	HH 12345678	KOWALSKI S.C.	KWIATOWA	4/9	WARSZAWA	01-900	543-123-456	46-744-431
3	00000003	PIOTR	MALCZYK	HF 12445661	ADA S.C.	ROZANA	9	WARSZAWA	01-900	443-133-251	16-742-114
4	00000004	PAWEL	FIODOROWICZ	DD 76545321	KRAWIECTWO	ARMII KRAJOWEJ	22A	WARSZAWA	01-200	555-233-256	44-342-116
5	00000005	ANIELA	DALGIEWICZ	NULL	MODNA PANI	BOHATEROW GETTA	24	WROCLAW	37-200	456-134-153	144-188-415
6	00000006	JOANNA	KWIATKOWSKA	NULL	NULL	TUWIMA	2/5	SWIDNICA	58-100	NULL	963-733-231
7	00000007	BOZENA	MALINOWSKA	NULL	NULL	LELEWELA	34/1	SWIDNICA	58-100	NULL	965-553-778
8	00000008	TOMASZ	NOWAK	NULL	NULL	ZEROMSKIEGO	5A/8	SWIDNICA	58-100	NULL	911-135-536
9	00000009	KRZYSZTOF	DOMAGALA	NULL	NULL	LESNA	5	SWIDNICA	58-100	NULL	922-233-232

	NR_PRACOWNIKA	IMIE	NAZWISKO	DATA_ZATR	DZIAL	STANOWISKO	PENSJA	DODATEK	NR_MIEJSCA	NR_TELEFONU		
10	000000											
11	000000	1	0001	JAN	KOWALSKI	1997-02-01 00:00:00.000	OBSLUGA KLIENTA	SPRZEDAWCA	1100.00	123.00	000001	987-231-123
		2	0002	ANNA	KAMINSKA	1997-01-01 00:00:00.000	OBSLUGA KLIENTA	SPRZEDAWCA	1200.00	115.00	000002	987-231-124
		3	0003	KRZYSZTOF	ADAMSKI	1997-05-01 00:00:00.000	OBSLUGA KLIENTA	KIEROWNIK	2000.00	NULL	000001	987-231-125
		4	0004	PIOTR	MICHALSKI	1998-06-01 00:00:00.000	TECHNICZNY	MECHANIK	1700.00	76.00	000001	987-231-131
		5	0005	BOZENA	DOMANSKA	1997-02-01 00:00:00.000	OBSLUGA KLIENTA	SPRZEDAWCA	1300.00	134.00	000003	987-231-126
		6	0006	WOJCIECH	BURZALSKI	1998-12-01 00:00:00.000	TECHNICZNY	MECHANIK	1800.00	80.00	000003	987-231-132

	NR_SAMOCHODU	MARKA	TYP	ROK_PROD	KOLOR	POJ_SILNIKA	PRZEBIEG	KASJER	1400.00	105.00 <th>000001</th> <th>987-231-141</th>	000001	987-231-141
1	000001	MERCEDES	190D	1999-01-01 00:00:00.000	BIALY	1800	23000	KIEROWNIK	2200.00	NULL	000001	987-231-133
2	000002	MERCEDES	230D	1999-01-01 00:00:00.000	NIEBIESKI	2000	35000	SPRZEDAWCA	1400.00	120.00	000004	933-241-525
3	000003	FIAT	SEICENTO	2000-01-01 00:00:00.000	CZERWONY	1100	13000	SPRZEDAWCA	1200.00	100.00	000001	457-531-143
4	000004	FIAT	SEICENTO	1999-01-01 00:00:00.000	BIALY	900	10000					
5	000005	FIAT	TIPO	1998-01-01 00:00:00.000	BORDOWY	1400	43000					

	NR_WYPOZYCZENIA	NR_KLIENTA	NR_SAMOCHODU	NR_PRACOW_WYP	NR_PRACOW_ODD	NR_MIEJSCA_WYP	NR_MIEJSCA_ODD	DATA_WYP	DATA_ODD		
6	000006										
7	000007	1	00000001	00000001	000003	0002	0002	000001	000001	1998-09-18 00:00:00.000	1998-09-23 00:00:00.000
8	000008	2	00000002	00000003	000004	0001	0001	000001	000001	1998-09-26 00:00:00.000	1998-09-27 00:00:00.000
9	000009	3	00000003	00000002	000004	0009	0009	000002	000002	1998-10-04 00:00:00.000	1998-10-04 00:00:00.000
10	000010	4	00000004	00000004	000003	0010	0010	000003	000003	1998-10-19 00:00:00.000	1998-10-25 00:00:00.000
11	000011	5	00000005	00000006	000007	0010	0010	000003	000003	1998-10-29 00:00:00.000	1998-11-02 00:00:00.000
12	000012	6	00000006	00000005	000008	0010	0002	000001	000003	1998-11-07 00:00:00.000	1998-11-09 00:00:00.000
13	000013	7	00000007	00000008	000011	0009	0002	000001	000001	1998-11-20 00:00:00.000	1998-11-25 00:00:00.000
14	000014	8	00000008	00000006	000011	0001	0005	000004	000004	1998-11-28 00:00:00.000	1998-12-02 00:00:00.000
15	000015	9	00000009	00000007	000017	0002	0002	000001	000002	1998-12-01 00:00:00.000	1998-12-03 00:00:00.000
16	000016	SEAT	IBIZA	1998-01-01 00:00:00.000	ZOLTY	1800	63000				
17	000017	FORD	SIERRA	1995-01-01 00:00:00.000	CZERWONY	1600	87000				
18	000018	OPEL	CORSA	2000-01-01 00:00:00.000	ZIELONY	1400	9000				
		FORD	KA	1999-01-01 00:00:00.000	ZOLTY	1400	20000				

Wydobywanie informacji z tradycyjnej bazy danych oraz z systemu relacyjnej bazy danych

Do dzisiaj istnieją tzw. tradycyjne bazy danych. Są to bazy informacji, których nośnikiem jest po prostu papier. Wiele instytucji w tym niestety policja oraz szpitale korzysta do dziś z takich baz.

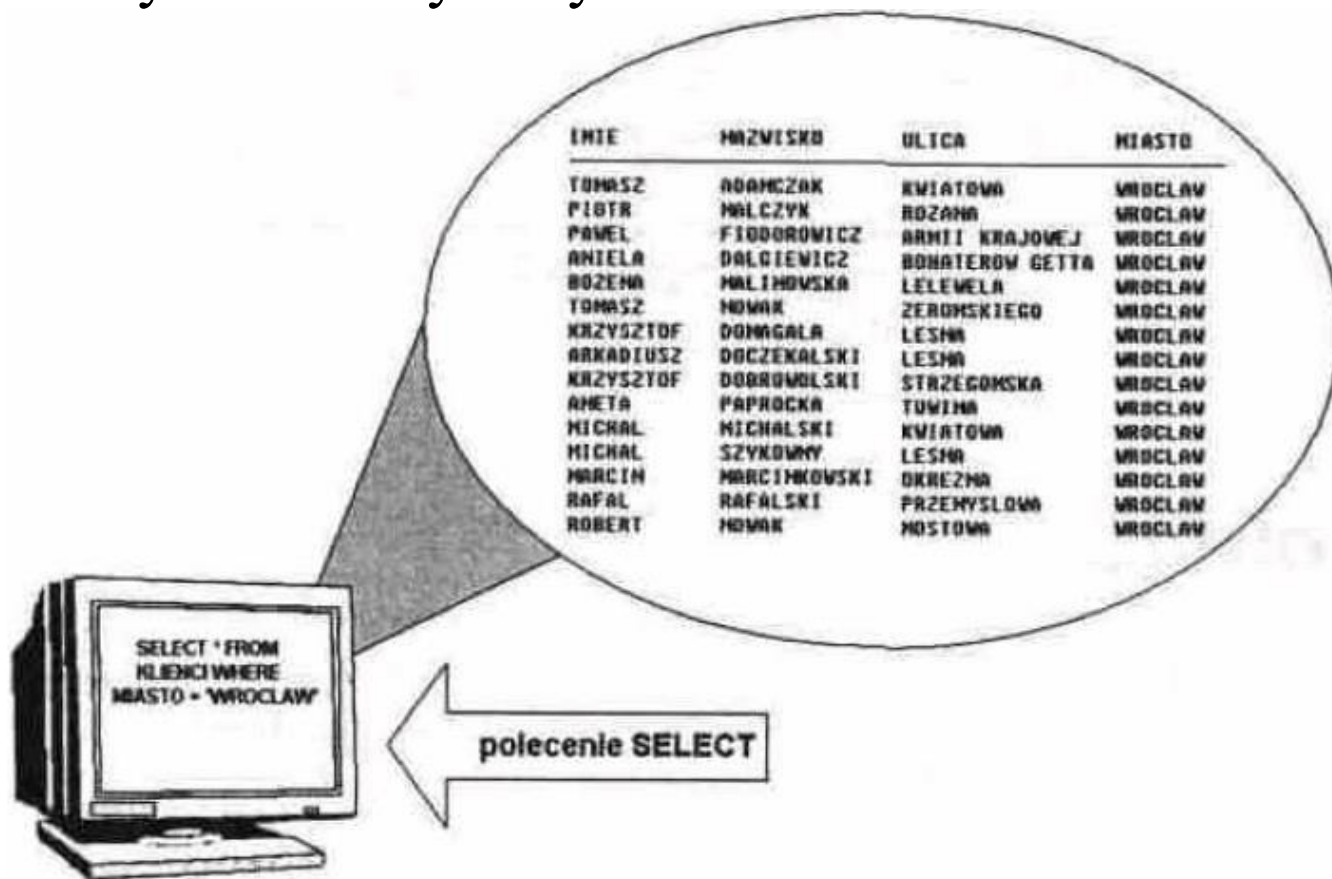


Scenariusz wydobywania danych w takich instytucjach jest następujący:

- 1 osoba potrzebująca danych np. przełożony prosi drugą osobę o pewne dane;
- pracownik po otrzymaniu polecenia szuka informacji wśród dokumentów, które zostały złożone albo w archiwum lub po prostu w szafie;
- pracownik po zebraniu pewnej ilości segregatorów z danego okresu przegląda je, a następnie przygotowuje zbiorczy dokument zawierający żądane informacje;
- po wydobyciu danych i zorganizowaniu ich do odpowiedniej formy wysyła je do przełożonego.

Czas wykonania tych operacji jest różny i zależy od wielkości organizacji, jej struktury, ilości potrzebnych danych oraz od pracowitości osób je zbierających.

W systemie relacyjnej bazy danych wszystkie powyższe operacje sprowadzają się do sformułowania tego samego pytania o dane ale w formie zrozumiałej dla komputera, a ściślej mówiąc, w formie zrozumiałej dla systemu bazy danych.



Cała operacja wydobywania danych trwa w tym przypadku znacznie krócej. Jakość tych danych jest przy tym lepsza.

Mamy więcej pewności, że dane są prawdziwe, że ktoś się nie pomylił lub pominął pewną część danych przy zestawianiu żądanych informacji.

(pomyłka jednak może mieć miejsce, ale jej charakter jest zgoła odmienny!!!)

Język SQL (ang. Structured Query Language) jest strukturalnym językiem zapewniającym możliwość wydawania poleceń do systemu zarządzania bazą danych (SZBD).

Dzięki temu językowi możliwe jest tworzenie poleceń, które nakazują SZBD odnaleźć potrzebne przez nas dane w bazie danych, jak również poleceń, dzięki którym można składowane dane zmodyfikować, wstawić nowe dane, bądź je usunąć.

Dzięki temu językowi możliwe jest również definiowanie, modyfikacja i usuwanie struktur danych, w których dane są składowane.

Język SQL pozwala również na zarządzanie transakcjami i mechanizmami autoryzacji dostępu do danych.

Język SQL jest językiem deklaratywnym. Charakterystyczną cechą języków deklaratywnych jest to, że opisują „co” ma być zrobione, ale nie „jak”.

W praktyce oznacza to, że użytkownik opisuje w języku SQL efekt jaki chce uzyskać (odczytanie telefonów wszystkich pracowników o nazwisku zaczynającym się na literę ‘A’, podniesienie pensji wszystkim profesorom itp.), ale nie sposób w jaki ma to być zrobione (sekwencja operacji dyskowych, które prowadzą do wykonania polecenia).

Sposób wykonania polecenia jest automatycznie dobierany przez SZBD i zależy od fizycznego sposobu składowania danych.

Język SQL jest zorientowany na przetwarzanie zbiorów.

Ponieważ dane są przechowywane w relacjach, które są zbiorami krotek, konstrukcje języka SQL dotyczą przetwarzania zbiorów i nie zawierają poleceń uwzględniających jakiś porządek na krotkach.

W języku SQL można wyróżnić trzy grupy poleceń:

DML – (ang. Data Manipulation Language) język manipulacji danymi pozwalający na odczytywanie danych z relacji (polecenie SELECT) oraz na wstawianie, modyfikację i usuwanie danych z relacji (polecenia: INSERT, UPDATE, MERGE i DELETE).

DDL – (ang. Data Definition Language) język definicji danych pozwalający na tworzenie, modyfikację i usuwanie relacji (polecenia CREATE, ALTER i DROP).

DCL – (ang. Data Control Language) język kontroli danych pozwalający na zapewnienie autoryzacji dostępu do danych oraz zarządzanie transakcjami. Najważniejsze polecenia to GRANT i REVOKE (czasem zaliczane do DDLa) oraz COMMIT, ROLLBACK i SAVEPOINT.

W języku SQL nie rozróżnia się dużych i małych liter. Wszystkie słowa kluczowe i nazwy (np. relacji i atrybutów) w języku SQL można pisać zarówno dużymi jak i małymi literami. Przykładowo, wyrażenia: nazwisko, Nazwisko i NAZWISKO są identyczne.

W poleceniach SQL ignorowane są znaki końca linii. Wszystkie słowa kluczowe i wyrażenia w języku SQL można rozdzielać zarówno spacjami jak i znakami końca linii. W rezultacie dowolne polecenie SQL można sformatować w dowolny sposób (może ono zajmować jeden długi wiersz, lub kilka krótszych wierszy).

Każde polecenie SQL powinno być zakończone średnikiem. Próba wykonania polecenia nie zakończonego w ten sposób może zakończyć się błędem, chociaż istnieją sytuacje, gdy pominięcie średnika jest dozwolone.

Tworzenie tabeli - CREATE TABLE

Tworzenie tabeli polega na definiowaniu jej kolumn. Dla każdej kolumny należy określić nazwę kolumny, typ danych i długość (w zależności od typu) oraz to, czy jest dozwolone pozostawienie wartości pustej w kolumnie.

```
CREATE TABLE PRACOWNICY (  
    NR_PRACOWNIKA    CHAR(4)        NOT NULL,  
    IMIE             VARCHAR(20) NOT NULL,  
    NAZWISKO        VARCHAR(20) NOT NULL,  
    DATA_ZATR      DATE           NOT NULL,  
    DZIAL           VARCHAR(20) NOT NULL,  
    STANOWISKO      VARCHAR(20) NOT NULL,  
    PENSJA          DECIMAL(8,2)    ,  
    DODATEK        DECIMAL(8,2)    ,  
    NR_MIEJSCA      CHAR(6)        NOT NULL,  
    NR_TELEFONU     CHAR(16)        ,  
    PRIMARY KEY (NR_PRACOWNIKA));
```

Wartość pusta NULL

Wartość NULL jest to wartość nieokreślona, która może zostać użyta w każdym polu tabeli niezależnie od typu kolumny. Wartość NULL jest różna od zera lub spacji.

	NR_PRACOWNIKA	IMIE	NAZWISKO	DATA_ZATR	DZIAL	STANOWISKO	PENSJA	DODATEK	NR_MIEJSCA	NR_TELEFONU
1	0001	JAN	KOWALSKI	1997-02-01 00:00:00.000	OBSLUGA KLIENTA	SPRZEDAWCA	1100.00	123.00	000001	987-231-123
2	0002	ANNA	KAMINSKA	1997-01-01 00:00:00.000	OBSLUGA KLIENTA	SPRZEDAWCA	1200.00	115.00	000002	987-231-124
3	0003	KRZYSZTOF	ADAMSKI	1997-05-01 00:00:00.000	OBSLUGA KLIENTA	KIEROWNIK	2000.00	NULL	000001	987-231-125
4	0004	PIOTR	MICHALSKI	1998-06-01 00:00:00.000	TECHNICZNY	MECHANIK	1700.00	76.00	000001	987-231-131
5	0005	BOZENA	DOMANSKA	1997-02-01 00:00:00.000	OBSLUGA KLIENTA	SPRZEDAWCA	1300.00	134.00	000003	987-231-126
6	0006	WOJCIECH	BURZALSKI	1998-12-01 00:00:00.000	TECHNICZNY	MECHANIK	1800.00	80.00	000003	987-231-132
7	0007	MARZENA	KOWNACKA	1997-05-01 00:00:00.000	KSIEGOWOSC	KASJER	1400.00	105.00	000001	987-231-141
8	0008	DAMIAN	MACHALICA	1997-05-01 00:00:00.000	TECHNICZNY	KIEROWNIK	2200.00	NULL	000001	987-231-133
9	0009	ALICJA	MAKOWIECKA	1999-07-01 00:00:00.000	OBSLUGA KLIENTA	SPRZEDAWCA	1400.00	120.00	000004	933-241-525
10	0010	WOJCIECH	BAGIELSKI	1998-04-01 00:00:00.000	OBSLUGA KLIENTA	SPRZEDAWCA	1200.00	100.00	000001	457-531-143

Przy konstruowaniu tabeli poleceniem CREATE TABLE w poprzedniej sekcji określono dla pewnych kolumn parametr NOT NULL. Oznacza to, że przy wstawianiu nowych wierszy musimy określić wartości dla tych kolumn, nie mogą one być wartością NULL.

Definicja kolumny w poleceniu CREATE TABLE pozostawiona bez klauzuli NOT NULL określa, że dozwolone jest wstawienie do tej kolumny wartości NULL. Istnieje jeszcze opcja o następującej składni:

NOT NULL WITH DEFAULT 'wartość'

gdzie parametr wartość określa domyślną wartość dla kolumny. Wartość domyślna zostanie nadana dla kolumny automatycznie, gdy nie określimy jej wprost przy wstawianiu nowego wiersza do tabeli.

Autoryzacja dostępu do tabeli

Możemy udostępnić nasze dane innym użytkownikom, a ściślej mówiąc możemy udostępnić tabele innemu użytkownikowi.

W tym celu stosuje się polecenie języka SQL GRANT. Poniższy przykład nadaje uprawnienia użytkownikowi o nazwie UŻYTKOWNIK do tabeli PRACOWNICY:

GRANT SELECT ON PRACOWNICY TO UŻYTKOWNIK;

Od tej pory UŻYTKOWNIK może wybierać (wykonywać zapytania SELECT) dane z naszej tabeli.

Prawa do tabeli można odebrać poleceniem REVOKE. Oto przykład:

REVOKE SELECT ON PRACOWNICY FROM UŻYTKOWNIK;

Poniższy przykład umożliwia wybieranie, wstawianie i aktualizowanie danych w tabeli:

GRANT SELECT, INSERT, UPDATE ON PRACOWNICY TO UŻYTKOWNIK;

Widoki

Za pomocą widoków możemy ograniczyć zakres danych dostępnych dla użytkownika. Widok może ograniczać dane z jednej tabeli lub może to być kompozycja danych z kilku tabel. Dane w widoku mogą być ograniczone do kilku kolumn lub do pewnego zakresu wierszy.

Widoki stosuje się w różnych celach:

- w celu zabezpieczenia danych przed niepowołanym dostępem;
- uproszczenia korzystania z danych dla końcowego użytkownika.

Przykładem zwiększenia bezpieczeństwa może być widok, który nie obejmuje kolumny z danymi o zarobkach. Wiadomo, że nie wszyscy użytkownicy powinni mieć dostęp do takich danych.

Tabela PRACOWNICY

	NR_PRACOWNIKA	IMIE	NAZWISKO	DATA_ZATR	DZIAL	STANOWISKO	PENSJA	DODATEK	NR_MIEJSCA	NR_TELEFONU
1	0001	JAN	KOWALSKI	1997-02-01 00:00:00.000	OBSLUGA KLIENTA	SPRZEDAWCA	1100.00	123.00	000001	987-231-123
2	0002	ANNA	KAMINSKA	1997-01-01 00:00:00.000	OBSLUGA KLIENTA	SPRZEDAWCA	1200.00	115.00	000002	987-231-124
3	0003	KRZYSZTOF	ADAMSKI	1997-05-01 00:00:00.000	OBSLUGA KLIENTA	KIEROWNIK	2000.00	NULL	000001	987-231-125
4	0004	PIOTR	MICHALSKI	1998-06-01 00:00:00.000	TECHNICZNY	MECHANIK	1700.00	76.00	000001	987-231-131
5	0005	BOZENA	DOMANSKA	1997-02-01 00:00:00.000	OBSLUGA KLIENTA	SPRZEDAWCA	1300.00	134.00	000003	987-231-126
6	0006	WOJCIECH	BURZALSKI	1998-12-01 00:00:00.000	TECHNICZNY	MECHANIK	1800.00	80.00	000003	987-231-132
7	0007	MARZENA	KOWNACKA	1997-05-01 00:00:00.000	KSIEGOWOSC	KASJER	1400.00	105.00	000001	987-231-141
8	0008	DAMIAN	MACHALICA	1997-05-01 00:00:00.000	TECHNICZNY	KIEROWNIK	2200.00	NULL	000001	987-231-133
9	0009	ALICJA	MAKOWIECKA	1999-07-01 00:00:00.000	OBSLUGA KLIENTA	SPRZEDAWCA	1400.00	120.00	000004	933-241-525
10	0010	WOJCIECH	BAGIELSKI	1998-04-01 00:00:00.000	OBSLUGA KLIENTA	SPRZEDAWCA	1200.00	100.00	000001	457-531-143

Widok utworzony na podstawie tabeli PRACOWNICY

	IMIE	NAZWISKO	DATA_ZATR	DZIAL	STANOWISKO
▶	JAN	KOWALSKI	1997-02-01 00:...	OBSLUGA KLIENTA	SPRZEDAWCA
	ANNA	KAMINSKA	1997-01-01 00:...	OBSLUGA KLIENTA	SPRZEDAWCA
	KRZYSZTOF	ADAMSKI	1997-05-01 00:...	OBSLUGA KLIENTA	KIEROWNIK
	PIOTR	MICHALSKI	1998-06-01 00:...	TECHNICZNY	MECHANIK
	BOZENA	DOMANSKA	1997-02-01 00:...	OBSLUGA KLIENTA	SPRZEDAWCA
	WOJCIECH	BURZALSKI	1998-12-01 00:...	TECHNICZNY	MECHANIK
	MARZENA	KOWNACKA	1997-05-01 00:...	KSIEGOWOSC	KASJER
	DAMIAN	MACHALICA	1997-05-01 00:...	TECHNICZNY	KIEROWNIK
	ALICJA	MAKOWIECKA	1999-07-01 00:...	OBSLUGA KLIENTA	SPRZEDAWCA
	WOJCIECH	BAGIELSKI	1998-04-01 00:...	OBSLUGA KLIENTA	SPRZEDAWCA

Podsumowanie-wstęp do języka SQL

1. Relacyjna baza danych jest kolekcją tabel.
2. Użytkownicy nie muszą się martwić o to, jak dane są przechowywane w bazie danych oraz jak są wydobywane.
3. SQL jest językiem do komunikowania się z baza danych.
4. Język SQL jest używany do:
 - wydobywania danych (SELECT);
 - manipulowania danymi (INSERT, UPDATE, DELETE);
 - definiowania, redefiniowania i usuwania obiektów wchodzących w skład struktury bazy danych (CREATE, ALTER, DROP);
 - definiowania uprawnień do danych (GRANT, REVOKE).
5. Użytkownicy mogą mieć dostęp do danych poprzez widoki.

OPERACJE RELACYJNE

SELEKCJA

Operacja selekcji umożliwia pobieranie krotek (wiersz) spełniających określony warunek. Operacja ta nazwana jest również podzbiorem poziomym.

Pesel	Imię	Nazwisko	Wykształcenie
72030403987	Małgorzata	Albinos	WT
65081002987	Damian	Jędrzejek	SO
44101202034	Barbara	Bibicka	P
70010101231	Piotr	Burzyński	SO
55121201223	Mateusz	Manicki	ST



W języku SQL wykonanie selekcji umożliwia polecenie SELECT z klauzulą WHERE.

Przykładowe polecenie:

```
SELECT * FROM osoby;
```

Spowoduje wybranie wszystkich krotek (wiersz) z relacji (tabeli) osoby (to nie jest selekcja!!!).

W celu pobrania wierszy (wykonania projekcji), dla których pole w kolumnie 'Wykształcenie' jest równe 'SO' (średnie ogólne) należy zapytanie sformułować następująco:

```
SELECT * FROM osoby  
WHERE Wykształcenie = 'SO';
```


Warunki selekcji mogą być złożone. Przykładowo, aby wybrać wszystkie osoby, które mają wykształcenie średnie (średnie techniczne - ST lub średnie ogólne - SO) można odpowiednie warunki połączyć operatorem logicznym OR, wówczas zapytanie wygląda w sposób następujący:

```
SELECT * FROM osoby  
WHERE Wykształcenie = 'ST' OR Wykształcenie = 'SO';
```

Budowa wyrażeń i warunków zostanie omówiona dokładniej w dalszej części wykładu.

PROJEKCJA

Projekcja umożliwia pobieranie wartości wybranych atrybutów, wymienionych po słowie kluczowym SELECT z wszystkich krotek relacji. Operacja ta jest nazwana także podzbiorem pionowym.

Przykładową operację której wynik przedstawiono na rysunku poniżej można wykonać za pomocą następującego zapytania:
SELECT Pesel, Wykształcenie FROM osoby;

Pesel	Imię	Nazwisko	Wykształcenie
72030403987	Małgorzata	Albinos	WT
65081002987	Damian	Jędrzejek	SO
44101202034	Barbara	Bibicka	P
70010101231	Piotr	Burzyński	SO
55121201223	Mateusz	Manicki	ST

Wyrażenia i funkcje

72030403987	WT
65081002987	SO
44101202034	P
70010101231	SO
55121201223	ST

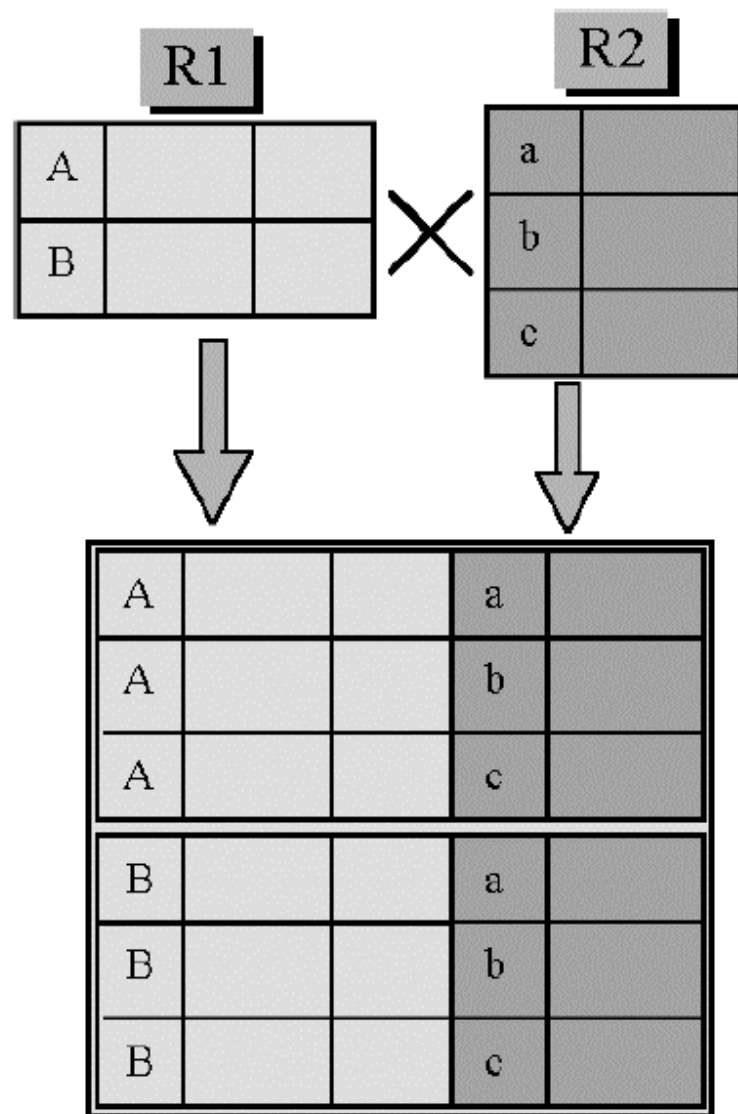
Operacje selekcji i projekcji mogą być łączone w jednym zapytaniu SELECT. I tak aby otrzymać kolumny zawierające Pesel i Nazwisko osób mających średnie wykształcenie wyrażenie przyjmie następującą postać:

```
SELECT Pesel, Nazwisko FROM osoby  
WHERE Wkształcenie = 'ST' OR Wykształcenie = 'SO';
```

PRODUKT

Produkt (iloczyn kartezjański) jest operacją teorii zbiorów. Operacja ta umożliwia łączenie dwóch tabel lub więcej relacji w taki sposób, że każda krotka pierwszej relacji, jest łączona z każdą krotką drugiej relacji.

W przypadku większej ilości relacji, operacja ta jest wykonywana na pierwszych dwóch, a następnie na otrzymanym wyniku i relacji następnej, aż do wyczerpania wszystkich argumentów. Przykładowe wykonanie iloczynu kartezjańskiego przedstawia poniższy rysunek.



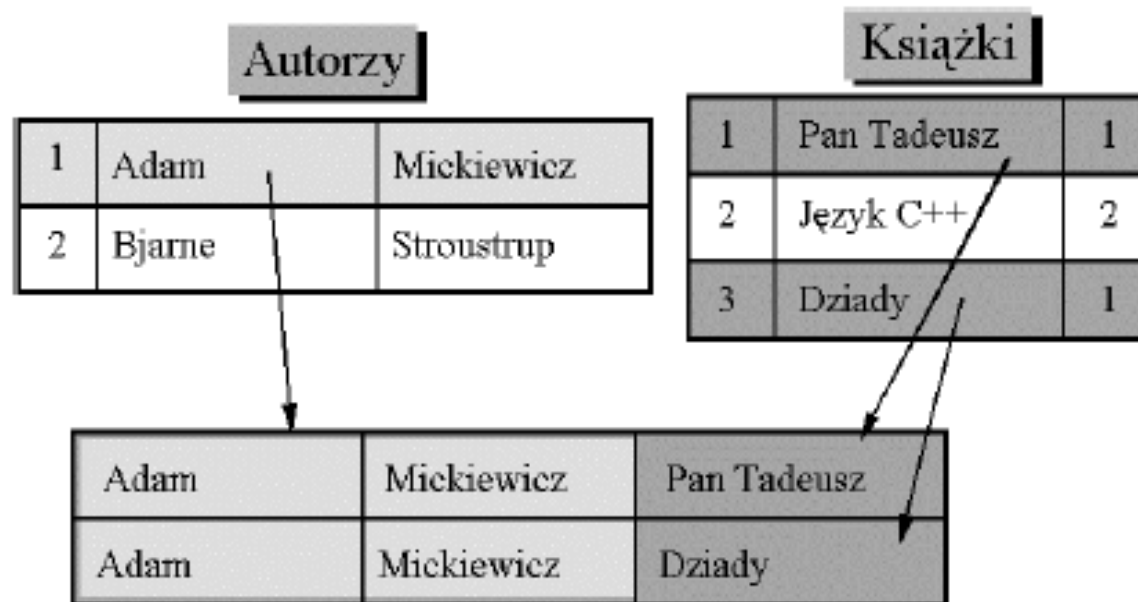
Znajdowanie iloczynu kartezyjskiego dwóch relacji (tabel) jest również wykonywane przez rozkaz SELECT. Przedstawioną na powyższym rysunku operację można wykonać za pomocą następującego zapytania:

SELECT * FROM R1, R2;

Operacja znajdowania iloczynu kartezyjskiego może być łączona zarówno z operacją selekcji, jak również projekcji lub oboma równocześnie.

POŁĄCZENIE

Operacja ta polega na łączeniu krotek dwóch lub więcej relacji z zastosowaniem określonego warunku łączenia. Wynikiem połączenia jest podzbiór produktu relacji.



Operację której wynik przedstawiono na powyższym rysunku można wykonać za pomocą następującego zapytania:

```
SELECT imie, nazwisko, tytul  
FROM autorzy, ksiazki  
WHERE autorzy.nazwisko = 'Mickiewicz' AND  
Autorzy.nr = ksiazki.autor;
```

OPERACJE MNOGOŚCIOWE

Unia

Unia pozwala na zsumowanie zbiorów krotek dwóch lub więcej relacji (bez powtórzeń – zgodnie z teorią mnogości). Warunkiem poprawności tej operacji jest zgodność liczby i typów atrybutów (kolumn) zsumowanych relacji.

PLSQL>SQL>Query 1.sql Summary

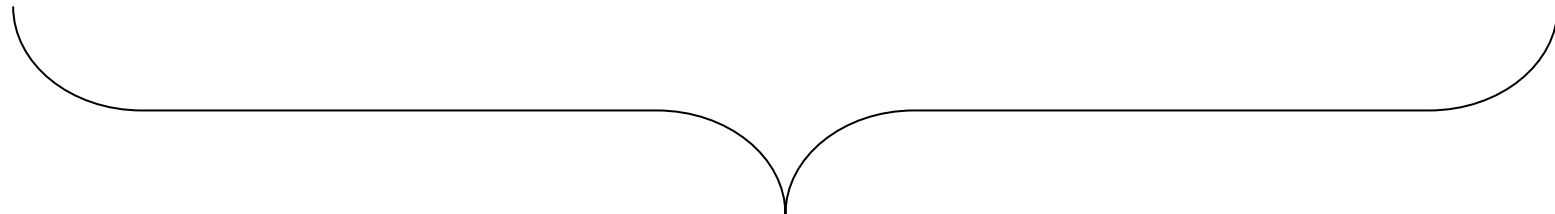
```
select imie, nazwisko from pracownicy
```

	imie	nazwisko
1	JAN	KOWALSKI
2	ANNA	KAMINSKA
3	KRZYSZTOF	ADAMSKI
4	PIOTR	MICHALSKI
5	BOZENA	DOMANSKA
6	WOJCIECH	BURZALSKI
7	MARZENA	KOWNACKA
8	DAMIAN	MACHALICA
9	ALICJA	MAKOWIECKA
10	WOJCIECH	BAGIELSKI

+

```
select imie, nazwisko from klienci
```

	imie	nazwisko
1	JAN	KOWALSKI
2	TOMASZ	ADAMCZAK
3	PIOTR	MALCZYK
4	PAWEL	FIODOROWICZ
5	ANIELA	DALGIEWICZ
6	JOANNA	KWIATKOWSKA
7	BOZENA	MALINOWSKA
8	TOMASZ	NOWAK
9	KRZYSZTOF	DOMAGALA
10	ARKADIUSZ	DOCZEKALSKI
11	ANNA	KOWALSKA
12	KRZYSZTOF	DOBROWOLSKI
13	MARCIN	KRZYKALA
14	ANETA	PAPROCKA
15	SEBASTIAN	KOWNACKI
16	MICHAL	MICHALSKI
17	MICHAL	SZYKOWNY
18	MARCIN	MARCINKOWSKI
19	RAFAL	RAFALSKI
20	ROBERT	NOWAK



```
select imie, nazwisko from pracownicy
union
select imie, nazwisko from klienci
```

Results Messages

	imie	nazwisko
1	ALICJA	MAKOWIECKA
2	ANETA	PAPROCKA
3	ANIELA	DALGIEWICZ
4	ANNA	KAMINSKA
5	ANNA	KOWALSKA
6	ARKADIUSZ	DOCZEKALSKI
7	BOZENA	DOMANSKA
8	BOZENA	MALINOWSKA
9	DAMIAN	MACHALICA
10	JAN	KOWALSKI
11	JOANNA	KWIATKOWSKA
12	KRZYSZTOF	ADAMSKI
13	KRZYSZTOF	DOBROWOLSKI
14	KRZYSZTOF	DOMAGALA
15	MARCIN	KRZYKALA
16	MARCIN	MARCINKOWSKI
17	MARZENA	KOWNACKA
18	MICHAL	MICHALSKI
19	MICHAL	SZYKOWNY
20	PAWEL	FIODOROWICZ
21	PIOTR	MALCZYK
22	PIOTR	MICHALSKI
23	RAFAL	RAFALSKI
24	ROBERT	NOWAK
25	SEBASTIAN	KOWNACKI
26	TOMASZ	ADAMCZAK
27	TOMASZ	NOWAK
28	WOJCIECH	BAGIELSKI
29	WOJCIECH	BURZALSKI

Przekrój

Przekrój pozwala znaleźć iloczyn dwóch lub więcej zbiorów krotek tzn. takich, które występują zarówno w jednej jak i w drugiej relacji. Podobnie jak w przypadku unii, warunkiem poprawności tej operacji jest zgodność liczby i typów atrybutów relacji bazowych.

PERSONAL-00...Query1.sql Summary

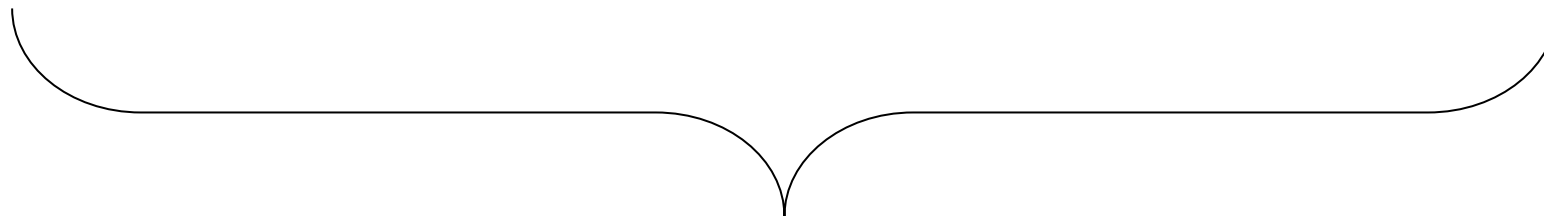
```
select imie, nazwisko from pracownicy
```

	imie	nazwisko
1	JAN	KOWALSKI
2	ANNA	KAMINSKA
3	KRZYSZTOF	ADAMSKI
4	PIOTR	MICHALSKI
5	BOZENA	DOMANSKA
6	WOJCIECH	BURZALSKI
7	MARZENA	KOWNACKA
8	DAMIAN	MACHALICA
9	ALICJA	MAKOWIECKA
10	WOJCIECH	BAGIELSKI

```
select imie, nazwisko from klienci
```

	imie	nazwisko
1	JAN	KOWALSKI
2	TOMASZ	ADAMCZAK
3	PIOTR	MALCZYK
4	PAWEL	FIODOROWICZ
5	ANIELA	DALGIEWICZ
6	JOANNA	KWIATKOWSKA
7	BOZENA	MALINOWSKA
8	TOMASZ	NOWAK
9	KRZYSZTOF	DOMAGALA
10	ARKADIUSZ	DOCZEKALSKI
11	ANNA	KOWALSKA
12	KRZYSZTOF	DOBROWOLSKI
13	MARCIN	KRZYKALA
14	ANETA	PAPROCKA
15	SEBASTIAN	KOWNACKI
16	MICHAL	MICHALSKI
17	MICHAL	SZYKOWNY
18	MARCIN	MARCINKOWSKI
19	RAFAL	RAFALSKI
20	ROBERT	NOWAK

*



```
select imie, nazwisko from pracownicy  
intersect  
select imie, nazwisko from klienci
```

Results Messages

	imie	nazwisko
1	JAN	KOWALSKI

Różnica

Operacja obliczania różnicy dwóch relacji polega na znalezieniu wszystkich krotek, które występują w pierwszej relacji, ale nie występują w drugiej.

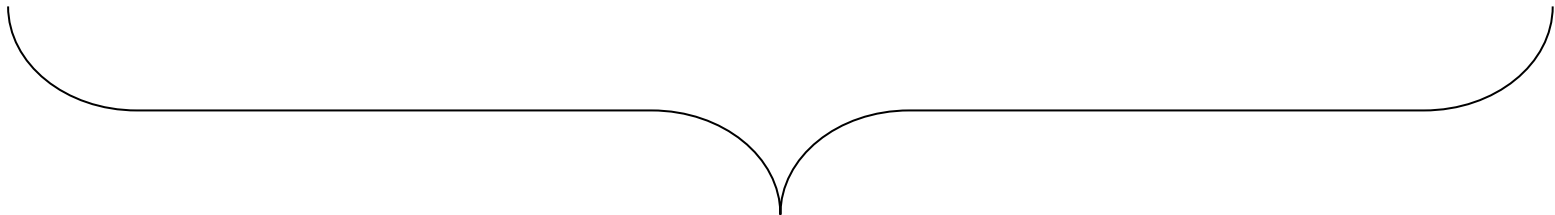
PERSONAL-000...Query1.sql Summary

```
select imie, nazwisko from pracownicy
```

	imie	nazwisko
1	JAN	KOWALSKI
2	ANNA	KAMINSKA
3	KRZYSZTOF	ADAMSKI
4	PIOTR	MICHALSKI
5	BOZENA	DOMANSKA
6	WOJCIECH	BURZALSKI
7	MARZENA	KOWNACKA
8	DAMIAN	MACHALICA
9	ALICJA	MAKOWIECKA
10	WOJCIECH	BAGIELSKI

```
select imie, nazwisko from klienci
```

	imie	nazwisko
1	JAN	KOWALSKI
2	TOMASZ	ADAMCZAK
3	PIOTR	MALCZYK
4	PAWEL	FIODOROWICZ
5	ANIELA	DALGIEWICZ
6	JOANNA	KWIATKOWSKA
7	BOZENA	MALINOWSKA
8	TOMASZ	NOWAK
9	KRZYSZTOF	DOMAGALA
10	ARKADIUSZ	DOCZEKALSKI
11	ANNA	KOWALSKA
12	KRZYSZTOF	DOBROWOLSKI
13	MARCIN	KRZYKALA
14	ANETA	PAPROCKA
15	SEBASTIAN	KOWNACKI
16	MICHAL	MICHALSKI
17	MICHAL	SZYKOWNY
18	MARCIN	MARCINKOWSKI
19	RAFAL	RAFALSKI
20	ROBERT	NOWAK



```
select imie, nazwisko from pracownicy  
except  
select imie, nazwisko from klienci
```

Results

Messages

	imie	nazwisko
1	ALICJA	MAKOWIECKA
2	ANNA	KAMINSKA
3	BOZENA	DOMANSKA
4	DAMIAN	MACHALICA
5	KRZYSZTOF	ADAMSKI
6	MARZENA	KOWNACKA
7	PIOTR	MICHALSKI
8	WOJCIECH	BAGIELSKI
9	WOJCIECH	BURZALSKI

**Czas teraz zaprezentować dokładnie przykłady
zapytań na pojedynczej tabeli.**