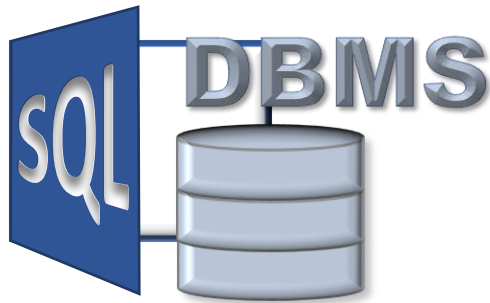


Bazy danych



Złączenia i agregacja danych

złączenia wewnętrzne i zewnętrzne
obliczenia i agregacja danych
sortowanie i ograniczanie liczby krotek

Przeszukiwanie danych – podstawowa operacja wykonywana w bazie danych, poszukiwanie informacji spełniających określone przez użytkownika kryteria.

Język zapytań (*ang. query language*) – język programowania umożliwiający formułowanie zapytań, na podstawie których system zarządzania bazą danych wykonuje przeszukiwanie danych.

Operatory algebry relacji

- **rzutowanie** wybiera_atrybuty (kolumny) relacji pierwotnej,
- **selekcja** wybiera krotki (wiersze) relacji pierwotnej spełniające określone kryterium,
- **iloczyn kartezyjski** dwóch relacji łączy wiersze tworząc wszystkie pary krotek należących do obydwu relacji,
- **złączenie naturalne** dwóch relacji łączy wiersze, tworząc pary krotek, które mają identyczne wartości atrybutów wspólnych w obydwu relacjach,
- **złączenie teta** dwóch relacji łączy wiersze, tworząc pary krotek, dla których spełnione jest pewne kryterium,
- **przemianowanie** zmienia nazwę relacji i/lub jej atrybutów.

Wykład01 – formułowanie zapytań w SQL

SQL (*ang. Structured Query Language*, strukturalny język zapytań) – język manipulowania danymi utworzony przez firmę IBM. Obowiązuje jako standardowe narzędzie w systemach zarządzania relacyjnymi bazami danych. Zestaw poleceń związanych z przeszukiwaniem danych jest oparty na algebrze relacji.

Podstawowa składnia zapytania w języku SQL

```
SELECT atrybuty  
FROM   źródła_danych  
WHERE  warunek;
```

- *atrybuty* to rozdzielona przecinkami lista atrybutów (kolumn), których wartości są zwracane jako wynik zapytania, realizuje operację rzutowania;
- *źródła_danych* to rozdzielona przecinkami lista nazw źródeł, z których pobierane są dane (np. lista tabel), określa źródła danych, które podlegają złączeniu;
- *warunek* jest opcjonalny, może być dowolnym wyrażeniem logicznym, realizuje operację selekcji, a w SQL-89 dodatkowo określa warunek złączenia.

P(PESEL, NazwiskoP, ImięP, PWZ)

PESEL	NazwiskoP	ImięP	PWZ
82012090803	Kowalski	Jan	1234567
70052000234	Nowak	Andrzej	1234567
85102303225	Wiśniewska	Anna	7654321
93041020146	Kowalczyk	Julia	7654321

S(PESEL, ICD9, Data, PWZ)

PESEL	ICD9	Data	PWZ
70052000234	A14	15.01.21	1234567
70052000234	A15	15.01.21	1234567
82012090803	A14	18.01.21	7654321
82012090803	C09	18.01.21	7654321
82012090803	C19	18.01.21	7654321
70052000234	C09	18.01.21	1234567
82012090803	A14	11.02.21	1234567
70052000234	C41	25.02.21	1234567

L(PWZ, NazwiskoL, ImięL)

PWZ	NazwiskoL	ImięL
1234567	Adamska	Alicja
7654321	Wójcik	Zofia

B(ICD9, Kategoria, Cena)

ICD9	Kategoria	Cena
A14	Leukocyty	25
A15	Glukoza	40
C09	Erytrogram	37
C19	Hemoglobina	28
C41	Limfocyty T	45

Standard SQL-92 wprowadza operator **JOIN**, który wykonuje złączenie w klauzuli `FROM`. Wynik złączenia jest traktowany jako nowa wirtualna tabela, która stanowi wejście do kolejnego kroku przetwarzania danych realizowanego w `WHERE`.

Rodzaje złączeń

- Złączenie wewnętrzne **INNER JOIN** – złączenie oparte o zależności pomiędzy wybranymi kolumnami łączonych źródeł danych,
- Złączenie zewnętrzne **OUTER JOIN** – złączenie którego wynikiem są wszystkie krotki (wiersze) jednego źródła i odpowiadające im krotki drugiego źródła danych.

Podstawowa składnia

```
źródło1 INNER JOIN źródło2 ON warunek
```

```
źródło1 LEFT|RIGHT|FULL OUTER JOIN źródło2 ON warunek
```

Uwaga: niezależnie od typu złączenia wynikowa tabela wirtualna zawiera wszystkie kolumny łączonych źródeł (wszystkie kolumny mogą być używane w `WHERE` i `SELECT`)

Złączenie wewnętrzne INNER JOIN

Złączenie wewnętrzne łączy dane z dwóch źródeł na podstawie określonego warunku (wartości składników występujących w łączonych źródłach danych). Warunek złączenia może być skonstruowany przy pomocy operatorów relacyjnych i logicznych definiowanych przez algebrę relacji (s. 1-20).

Składnia INNER JOIN

```
SELECT atrybuty
FROM źródło1 INNER JOIN źródło2 ON warunek
```

Etapy realizacji INNER JOIN

1. Obliczenie iloczynu kartezyjskiego źródeł danych (wszystkich par krotek),
2. Wyznaczenie wartości warunku dla każdej pary krotek,
3. Odrzucenie wszystkich wierszy niespełniających warunku.

Uwaga 1: Wynikiem złączenia wewnętrznego będą tylko te pary krotek, które mają odpowiedniki w obydwu łączonych źródłach.

Uwaga 2: Złączenie wewnętrzne jest symetryczne, więc kolejność źródeł oraz kolejność składników w warunku nie wpływa na wynik zapytania.

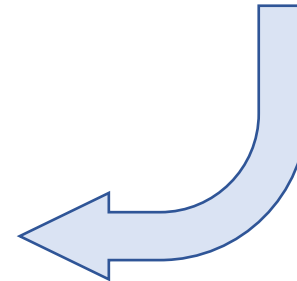
Złączenie wewnętrzne – przykład

Lista pacjentów z nazwiskami lekarzy rodzinnych

```
SELECT PESEL, NazwiskoP, ImięP, NazwiskoL, ImięL
FROM P INNER JOIN L ON P.PWZ = L.PWZ;
```

PESEL	NazwiskoP	ImięP	P.PWZ	warunek	L.PWZ	NazwiskoL	ImięL
82012090803	Kowalski	Jan	1234567	=	1234567	Adamska	Alicja
82012090803	Kowalski	Jan	1234567	-	7654321	Wójcik	Zofia
70052000234	Nowak	Andrzej	1234567	=	1234567	Adamska	Alicja
70052000234	Nowak	Andrzej	1234567	-	7654321	Wójcik	Zofia
85102303225	Wiśniewska	Anna	7654321	-	1234567	Adamska	Alicja
85102303225	Wiśniewska	Anna	7654321	=	7654321	Wójcik	Zofia
93041020146	Kowalczyk	Julia	7654321	-	1234567	Adamska	Alicja
93041020146	Kowalczyk	Julia	7654321	=	7654321	Wójcik	Zofia

PESEL	NazwiskoP	ImięP	NazwiskoL	ImięL
82012090803	Kowalski	Jan	Adamska	Alicja
70052000234	Nowak	Andrzej	Adamska	Alicja
85102303225	Wiśniewska	Anna	Wójcik	Zofia
93041020146	Kowalczyk	Julia	Wójcik	Zofia



Złączenia wewnętrzne equi-JOIN i NATURAL JOIN

Standard SQL-92 wprowadza dwa szczególne przypadki złączenia wewnętrznego

Equi-JOIN – złączenie sprawdza równość wskazanych atrybutów.

```
SELECT atrybuty
FROM źródło1 INNER JOIN źródło2 USING (atrybuty);
```

Atrybuty o nazwach określonych po USING muszą występować w obydwu źródłach danych. Złączeniu podlegają krotki, które mają równe wartości wszystkich wymienionych atrybutów.

Złączenie naturalne (NATURAL JOIN) – szczególny przypadek equi-JOIN, porównuje wartości wszystkich kolumn o identycznych nazwach (realizuje operację złączenia naturalnego z algebry relacji).

```
SELECT atrybuty
FROM źródło1 NATURAL JOIN źródło2;
```

Łączone źródła muszą zawierać atrybuty o identycznych nazwach. Złączeniu podlegają krotki, które mają równe wartości we wszystkich parach atrybutów.

Uwaga: equi-JOIN oraz NATURAL JOIN nie są obsługiwane przez niektóre systemy zarządzania bazami danych.

Złączenia wewnętrzne – przykłady

Lista pacjentów z nazwiskami lekarzy rodzinnych

- **INNER JOIN:**

```
SELECT PESEL, NazwiskoP, ImięP, NazwiskoL, ImięL
FROM P INNER JOIN L ON P.PWZ = L.PWZ;
```
- **Equi-JOIN:**

```
SELECT PESEL, NazwiskoP, ImięP, NazwiskoL, ImięL
FROM P INNER JOIN L USING (PWZ);
```
- **NATURAL JOIN:**

```
SELECT PESEL, NazwiskoP, ImięP, NazwiskoL, ImięL
FROM P NATURAL JOIN L;
```

Lista badań zleconych 18-01-2021 roku (PESEL, ICD9, Kategoria)

- **INNER JOIN:**

```
SELECT PESEL, S.ICD9, Kategoria
FROM S INNER JOIN B ON S.ICD9 = B.ICD9
WHERE Data = '2021-01-18';
```
- **Equi-JOIN:**

```
SELECT PESEL, S.ICD9, Kategoria
FROM S INNER JOIN B USING (ICD9)
WHERE Data = '2021-01-18';
```
- **NATURAL JOIN:**

```
SELECT PESEL, S.ICD9, Kategoria
FROM S NATURAL JOIN B
WHERE Data = '2021-01-18';
```



Złączenie zewnętrzne OUTER JOIN

Złączenie zewnętrzne łączy dane z dwóch źródeł danych na podstawie określonego warunku, wynik jest uzupełniany o krotki źródła znajdującego się po jednej ze stron złączenia (określonej odpowiednią dyrektywą), dla których warunek nie był spełniony.

Składnia

```
SELECT atrybuty  
FROM źródło1 LEFT | RIGHT | FULL OUTER JOIN źródło2 ON warunek;
```

LEFT, **RIGHT**, **FULL** wykluczają się, określają źródło którego krotki są dodane do wyniku.

Etapy realizacji OUTER JOIN

1. Obliczenie iloczynu kartezyjskiego źródeł danych (wszystkich par krotek),
2. Wyznaczenie wartości warunku dla każdej pary krotek,
3. Odrzucenie wszystkich wierszy niespełniających warunku,
4. Dołączenie krotek niespełniających warunku złączenia ze źródła znajdującego się po lewej (**LEFT**), prawej (**RIGHT**) lub obydwu (**FULL**) stronach **JOIN**.

Uwaga 1: Wynikiem złączenia zewnętrznego są wszystkie krotki z jednego źródła (obydwu w przypadku **FULL**) oraz ich odpowiedniki z drugiego. Jeżeli krotka z jednego źródła nie ma odpowiednika w drugim to jej atrybuty w wyniku są puste (**NULL**).

Uwaga 2: Złączenie zewnętrzne nie jest symetryczne.

Złączenie zewnętrzne – przykład

Lista wszystkich pacjentów z kodami badań.

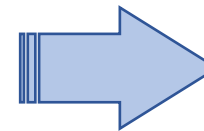
```
SELECT NazwiskoP, ImięP, ICD9  
FROM P LEFT OUTER JOIN S ON P.PESEL = S.PESEL;
```

Wynik złączenia wewnętrznego

P.PESEL	NazwiskoP	ImięP	P.PWZ	S.PESEL	ICD9	Data	S.PWZ
70052000234	Nowak	Andrzej	1234567	70052000234	A14	15.01.21	1234567
70052000234	Nowak	Andrzej	1234567	70052000234	A15	15.01.21	1234567
82012090803	Kowalski	Jan	1234567	82012090803	A14	18.01.21	7654321
82012090803	Kowalski	Jan	1234567	82012090803	C09	18.01.21	7654321
82012090803	Kowalski	Jan	1234567	82012090803	C19	18.01.21	7654321
70052000234	Nowak	Andrzej	1234567	70052000234	C09	18.01.21	1234567
82012090803	Kowalski	Jan	1234567	82012090803	A14	11.02.21	1234567
70052000234	Nowak	Andrzej	1234567	70052000234	C41	25.02.21	1234567

Po uzupełnieniu i selekcji

NazwiskoP	ImięP	ICD9
Nowak	Andrzej	A14
Nowak	Andrzej	A15
Kowalski	Jan	A14
Kowalski	Jan	C09
Kowalski	Jan	C19
Nowak	Andrzej	C09
Kowalski	Jan	A14
Nowak	Andrzej	C41
Wiśniewska	Anna	
Kowalczyk	Julia	



Lista pacjentów, którzy nigdy nie byli skierowani na badania.

```
SELECT NazwiskoP, ImięP  
FROM P LEFT OUTER JOIN S ON P.PESEL=S.PESEL  
WHERE ICD9 IS NULL;
```

NazwiskoP	ImięP
Wiśniewska	Anna
Kowalczyk	Julia

Możliwe warianty warunku WHERE:

Data IS NULL, S.PESEL IS NULL, S.PWZ IS NULL



```
SELECT atrybuty
FROM   źródło1 INNER|OUTER JOIN źródło2 ON warunek1
       INNER|OUTER JOIN źródło3 ON warunek2
...
       INNER|OUTER JOIN źródłoN ON warunekN-1;
```

Etapy realizacji złączenia wielokrotnego

1. Wykonanie złączenia wewnętrznego/zewnętrznego *źródła1* ze *źródłem2* na podstawie *warunku1* (zgodnie z regułami przedstawionymi na s. 6 i 10) i utworzenie pierwszej tabeli wirtualnej.
2. Wykonanie złączenia *źródła3* z tabelą wirtualną utworzoną w kroku 1 i utworzenie drugiej tabeli wirtualnej.
3. Wykonanie kolejnych złączeń kolejnych źródeł danych z tabelami wirtualnymi utworzonymi w krokach poprzednich, aż do *źródłaN* i utworzenia ostatecznej tabeli wirtualnej, która stanowi wejście do kolejnego kroku przetwarzania danych.

Uwaga: Złączenia domyślnie wykonywane są od lewej do prawej (zgodnie z kolejnością w klauzuli FROM). W celu zmiany kolejności należy użyć nawiasów.

Złączenie wielokrotne – przykład

Nazwiska i imiona pacjentów z kodami i kategoriami badań zleconych 18-01-2021.

```
SELECT NazwiskoP, ImięP, S.ICD9, Kategoria
FROM P INNER JOIN S ON P.PESEL = S.PESEL
INNER JOIN B ON S.ICD9 = B.ICD9
WHERE Data = '2021-01-18';
```

← Złączenie tabel P, S, B

P. PESEL=S. PESEL S. ICD9=B. ICD9

P.PESEL	NazwiskoP	ImięP	P.PWZ	S.PESEL	S.ICD9	Data	S.PWZ	B.ICD9	Kategoria	Cena
70052000234	Nowak	Andrzej	1234567	70052000234	A14	15.01.21	1234567	A14	Leukocyty	25
70052000234	Nowak	Andrzej	1234567	70052000234	A15	15.01.21	1234567	A15	Glukoza	40
82012090803	Kowalski	Jan	1234567	82012090803	A14	18.01.21	7654321	A14	Leukocyty	25
82012090803	Kowalski	Jan	1234567	82012090803	C09	18.01.21	7654321	C09	Erytrogram	37
82012090803	Kowalski	Jan	1234567	82012090803	C19	18.01.21	7654321	C19	Hemoglobina	28
70052000234	Nowak	Andrzej	1234567	70052000234	C09	18.01.21	1234567	C09	Erytrogram	37
82012090803	Kowalski	Jan	1234567	82012090803	A14	11.02.21	1234567	A14	Leukocyty	25
70052000234	Nowak	Andrzej	1234567	70052000234	C41	25.02.21	1234567	C41	LimfocytyT	45

I tabela wirtualna II tabela wirtualna

NazwiskoP	ImięP	S.ICD9	Kategoria
Kowalski	Jan	A14	Leukocyty
Kowalski	Jan	C09	Erytrogram
Kowalski	Jan	C19	Hemoglobina
Nowak	Andrzej	C09	Erytrogram

Po selekcji wierszy i rzutowaniu kolumn



Język SQL umożliwia wykonywanie obliczeń arytmetycznych. Wyrażenie arytmetyczne może zawierać nazwy kolumn źródła danych, wartości stałe, operatory arytmetyczne i funkcje. Może być umieszczone w klauzuli `SELECT` (wyniki tworzą nową kolumnę) lub wykorzystane do skonstruowania warunku w klauzuli `WHERE`.

Operatory arytmetyczne

+	dodawanie	*	mnożenie
-	odejmowanie	/	dzielenie

Uwaga: Wyrażenie arytmetyczne wykonywane jest od lewej do prawej z zachowaniem priorytetów operatorów: po pierwsze mnożenie i dzielenie, następnie dodawanie i odejmowanie. W celu zmiany kolejności należy użyć nawiasów.

Wybrane funkcje arytmetyczne

<code>ABS (w)</code>	wartość bezwzględna
<code>CEIL (w)</code>	zaokrąglenie w górę do pełnej całkowitej
<code>FLOOR (w)</code>	zaokrąglenie w dół do pełnej całkowitej
<code>ROUND (w, n)</code>	zaokrąglenie w do n miejsc po przecinku (domyślnie zero)
<code>SQRT (w)</code>	pierwiastek kwadratowy
<code>TRUNCATE (w)</code>	obcięcie części dziesiętnej

Wyznaczenie ceny brutto badań

```
SELECT ICD9, Cena, ROUND(Cena*1.23,2)
FROM B;
```

ICD9	Cena	ROUND(Cena*1.23,2)
A14	25	30.75
A15	40	49.20
C09	37	45.51
C19	28	34.44
C41	45	55.35

Badania, których cena brutto przekracza 45PLN

```
SELECT ICD9, Cena, ROUND(Cena*1.23,2)
FROM B
WHERE ROUND(Cena*1.23,2) > 45;
```

ICD9	Cena	ROUND(Cena*1.23,2)
A15	40	49.20
C09	37	45.51
C41	45	55.35

Uwaga: W celu zwiększenia czytelności wyniku zapytania można wykorzystać operację przemianowania nadając aliasy wybranym kolumnom wyniku.

```
SELECT ICD9, Cena AS 'Cena netto'
        ROUND(Cena*1.23,2) AS 'Cena brutto'
FROM B;
```

Agregacja polega na utworzeniu grup krotek wejściowego źródła danych na podstawie wartości wybranych atrybutów i wykonaniu na tak utworzonych grupach operacji sumujących. Taka operacja pozwala na wykonanie obliczeń na wartościach pochodzących z całej kolumny (kilku wierszy) źródła danych.

Składnia

```
SELECT atrybuty, funkcje  
FROM źródła_danych  
GROUP BY atrybuty
```

Uwaga: W przypadku użycia GROUP BY klauzula SELECT może zawierać jedynie nazwy atrybutów występujących na liście GROUP BY oraz funkcje agregujące, które dotyczą pozostałych atrybutów źródeł danych.

Funkcje agregujące

- `SUM(atrybut)` – suma wartości wskazanego atrybutu,
- `AVG(atrybut)` – średnia wartość wskazanego atrybutu,
- `MAX(atrybut)` – wartość maksymalna wskazanego atrybutu,
- `MIN(atrybut)` – wartość minimalna wskazanego atrybutu,
- `COUNT (*)` – liczba elementów w grupie.

Agregacja danych – przykład I

Lista kodów wszystkich wykonanych badań

```
SELECT ICD9  
FROM S  
GROUP BY ICD9;
```

PESEL	ICD9	Data	PWZ
70052000234	A14	15.01.21	1234567
70052000234	A15	15.01.21	1234567
82012090803	A14	18.01.21	7654321
82012090803	C09	18.01.21	7654321
82012090803	C19	18.01.21	7654321
70052000234	C09	18.01.21	1234567
82012090803	A14	11.02.21	1234567
70052000234	C41	25.02.21	1234567

ICD9
A14
A15
C09
C19
C41

PESEL	ICD9	Data	PWZ
70052000234	A14	15.01.21	1234567
82012090803	A14	18.01.21	7654321
82012090803	A14	11.02.21	1234567

PESEL	ICD9	Data	PWZ
70052000234	A15	15.01.21	1234567

PESEL	ICD9	Data	PWZ
82012090803	C09	18.01.21	7654321
70052000234	C09	18.01.21	1234567

PESEL	ICD9	Data	PWZ
82012090803	C19	18.01.21	7654321

PESEL	ICD9	Data	PWZ
70052000234	C41	25.02.21	1234567

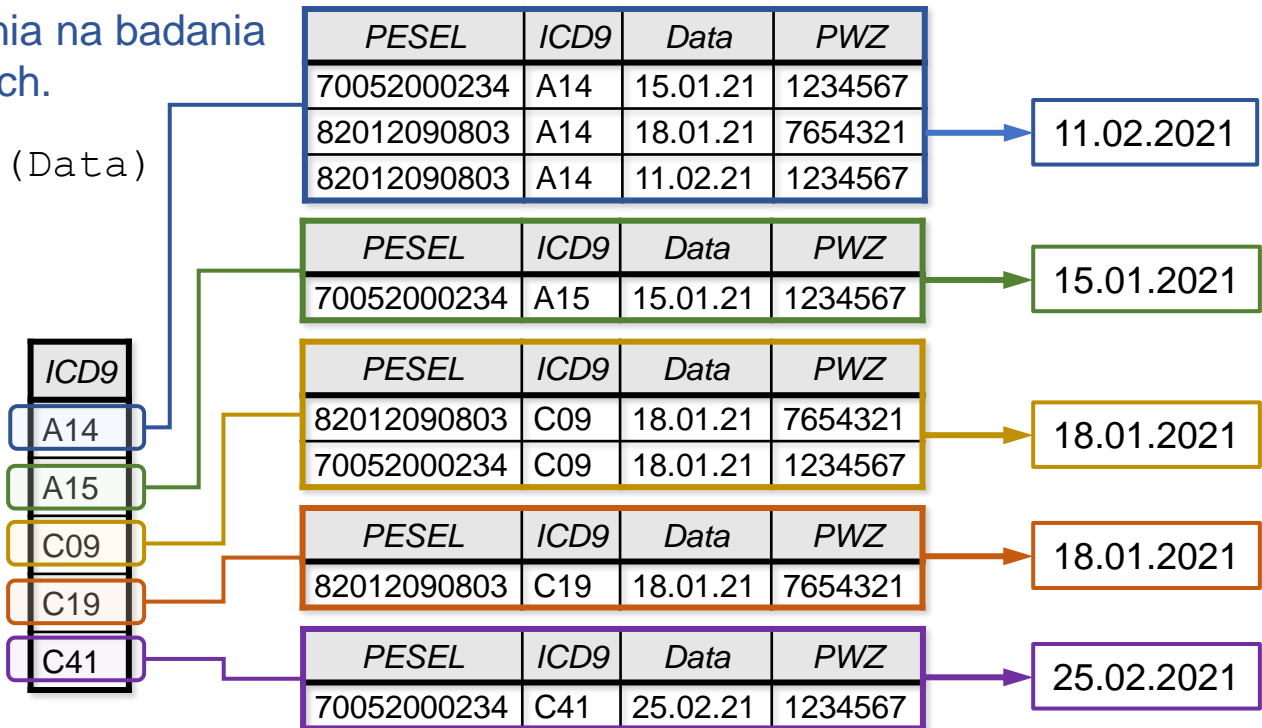


Agregacja danych – przykład II

Ostatnia data skierowania na badania o poszczególnych kodach.

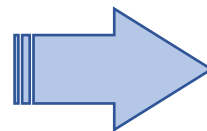
```
SELECT ICD9, MAX(Data)
FROM S
GROUP BY ICD9;
```

ICD9	MAX(Data)
A14	11.02.21
A15	15.01.21
C09	18.01.21
C19	18.01.21
C41	25.02.21



Liczba skierowań na badania o poszczególnych kodach

```
SELECT ICD9, COUNT(*)
FROM S
GROUP BY ICD9;
```



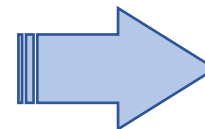
ICD9	COUNT(*)
A14	3
A15	1
C09	2
C19	1
C41	1

Agregacja danych – przykład III

Sumaryczny koszt badań o poszczególnych kodach

```
SELECT S.ICD9, SUM(Cena)
FROM S INNER JOIN B ON S.ICD9=B.ICD9
GROUP BY ICD9;
```

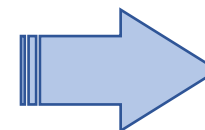
PESEL	S.ICD9	Data	PWZ	B.ICD9	Kategoria	Cena
70052000234	A14	15.01.21	1234567	A14	Leukocyty	25
70052000234	A15	15.01.21	1234567	A15	Glukoza	40
82012090803	A14	18.01.21	7654321	A14	Leukocyty	25
82012090803	C09	18.01.21	7654321	C09	Erytrogram	37
82012090803	C19	18.01.21	7654321	C19	Hemoglobina	28
70052000234	C09	18.01.21	1234567	C09	Erytrogram	37
82012090803	A14	11.02.21	1234567	A14	Leukocyty	25
70052000234	C41	25.02.21	1234567	C41	Limfocyty T	45



ICD9	SUM(Cena)
A14	75
A15	40
C09	74
C19	28
C41	45

Całkowity koszt wszystkich wykonanych badań

```
SELECT SUM(Cena)
FROM S INNER JOIN B ON S.ICD9=B.ICD9;
```



SUM(Cena)
262

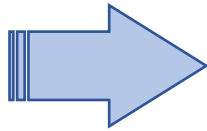
Agregacja danych – przykład IV

Liczba skierowań wystawionych w kolejnych dniach przez poszczególnych lekarzy.

```
SELECT Data, PWZ, Count (*)  
FROM S  
GROUP BY Data, PWZ;
```

Uwaga: Grupowanie na podstawie wartości dwóch kolumn: PWZ i Data. Grupę tworzą wiersze, które mają takie same wartości obydwu atrybutów.

PESEL	ICD9	Data	PWZ	
70052000234	A14	15.01.21	1234567	15.01.2021, 1234567
70052000234	A15	15.01.21	1234567	
82012090803	A14	18.01.21	7654321	18.01.2021, 7654321
82012090803	C09	18.01.21	7654321	
82012090803	C19	18.01.21	7654321	
70052000234	C09	18.01.21	1234567	18.01.2021, 1234567
82012090803	A14	11.02.21	1234567	11.02.2021, 1234567
70052000234	C41	25.02.21	1234567	25.02.2021, 1234567



Data	PWZ	COUNT(*)
15.01.21	1234567	2
18.01.21	7654321	3
18.01.21	1234567	1
11.02.21	1234567	1
25.02.21	1234567	1

Selekcja w agregowanych danych

W zapytaniach realizujących agregację selekcja może być wykonana na dwóch etapach:

- Na pierwotnym zbiorze danych (przed wykonaniem grupowania),
- Na danych zgrupowanych (wybór utworzonych grup).

Składnia

SELECT *atrybuty, funkcje*

FROM *źródła_danych*

WHERE *warunek* ← selekcja przed grupowaniem (wybór wierszy)

GROUP BY *atrybuty*

HAVING *warunek;* ← selekcja po grupowaniu (wybór grup)

Uwaga 1: Selekcja **WHERE** jest wykonywana jako pierwsza, więc grupy krotek są tworzone wyłącznie z wierszy, które spełniają kryterium **WHERE**.

Uwaga 2: W warunku umieszczonym w klauzuli **WHERE** dostępne są wszystkie kolumny źródła danych, natomiast w klauzuli **HAVING** można korzystać wyłącznie z atrybutów występujących na liście **GROUP BY** oraz rezultatów funkcji agregujących.

Selekcja w agregowanych danych – przykład

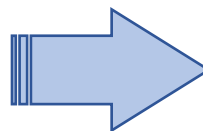
Lista dni, w których lekarz o PWZ = 1234567 wystawił więcej niż jedno skierowanie.

```
SELECT Data, Count (*)  
FROM S  
WHERE PWZ = '1234567'  
GROUP BY Data  
HAVING COUNT(*) > 1;
```

Uwaga: wybór wierszy zawierających skierowania lekarza 1234567 musi być przeprowadzony przed grupowaniem danych (PWZ nie występuje na liście GROUP BY, patrz uwaga 2 na s.21).

Selekcja WHERE

PESEL	ICD9	Data	PWZ
70052000234	A14	15.01.21	1234567
70052000234	A15	15.01.21	1234567
82012090803	A14	18.01.21	7654321
82012090803	C09	18.01.21	7654321
82012090803	C19	18.01.21	7654321
70052000234	C09	18.01.21	1234567
82012090803	A14	11.02.21	1234567
70052000234	C41	25.02.21	1234567



Grupowanie na podstawie daty

PESEL	ICD9	Data	PWZ
70052000234	A14	15.01.21	1234567
70052000234	A15	15.01.21	1234567
70052000234	C09	18.01.21	1234567
82012090803	A14	11.02.21	1234567
70052000234	C41	25.02.21	1234567

Wybór grup spełniających kryterium HAVING

Data	COUNT(*)
15.01.21	2

Składnia

```
SELECT atrybuty  
FROM źródła_danych  
WHERE warunek  
ORDER BY kryterium [ASC | DESC];
```

Klauzula `ORDER BY` wykonywana jest jako ostatni krok przetwarzania danych, operuje na w pełni uformowanym zbiorze danych sortując go zgodnie z podanym kryterium.

kryterium jest listą atrybutów lub wyrażeń których wartości określają sposób uporządkowania zbioru wynikowego. Kryterium podstawowym jest element umieszczony po lewej stronie, jeżeli kilka krotek ma tą samą wartość uwzględniane są kolejne elementy z listy.

- **ASC** – sortowanie rosnące (domyślne),
- **DESC** – sortowanie malejące.

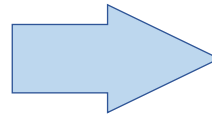
Uwaga: zgodnie ze standardem SQL-92 kryterium sortowania może zawierać tylko atrybuty występujące w klauzuli `SELECT`, standard SQL:1999 wprowadził możliwość sortowania po dowolnych kolumnach źródeł występujących w klauzuli `FROM`.

Sortowanie wyniku – przykład

Skierowania sortowane po PESEL i ICD9

```
SELECT PESEL, ICD9, Data FROM S  
ORDER BY PESEL, ICD9;
```

PESEL	ICD9	Data	PWZ
70052000234	A14	15.01.21	1234567
70052000234	A15	15.01.21	1234567
82012090803	A14	18.01.21	7654321
82012090803	C09	18.01.21	7654321
82012090803	C19	18.01.21	7654321
70052000234	C09	18.01.21	1234567
82012090803	A14	11.02.21	1234567
70052000234	C41	25.02.21	1234567



PESEL	ICD9	Data
70052000234	A14	15.01.21
70052000234	A15	15.01.21
70052000234	C09	18.01.21
70052000234	C41	25.02.21
82012090803	A14	18.01.21
82012090803	A14	11.02.21
82012090803	C09	18.01.21
82012090803	C19	18.01.21

Uwaga: dane zostały posortowane po numerze PESEL (pierwsze pole na liście ORDER BY). Grupy wierszy, w których PESEL jest taki sam są posortowane po ICD9.

Ograniczenie liczby krotek wyniku

Składnia

```
SELECT atrybuty FROM źródło_danych WHERE warunek  
LIMIT liczba [OFFSET przesunięcie];
```

Klauzula `LIMIT` ogranicza ilość zwracanych krotek do określonej przez *liczba*. Opcjonalny element `OFFSET przesunięcie` określa od której krotki rozpoczyna się zwracana lista (domyślnie zero).

Uwaga 1: możliwe jest skrócenie zapisu do `LIMIT przesunięcie, liczba`

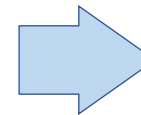
Uwaga 2: `LIMIT` może nie być dostępne w niektórych DBMS.

Przykład

Lista badań od 2 do 4 wiersza

```
SELECT * FROM B  
LIMIT 3 OFFSET 1;
```

ICD9	Kategoria	Cena
A14	Leukocyty	25
A15	Glukoza	40
C09	Erytrogram	37
C19	Hemoglobina	28
C41	Limfocyty T	45



ICD9	Kategoria	Cena
A15	Glukoza	40
C09	Erytrogram	37
C19	Hemoglobina	28

Składnia instrukcji SELECT

```
SELECT atrybuty, funkcje
FROM źródło_danych1 INNER|OUTER JOIN źródło_danych2 ON warunek1
...
INNER|OUTER JOIN źródło_danychN ON warunekN-1
WHERE warunek
GROUP BY atrybuty
HAVING warunek
ORDER BY kryterium [ASC | DESC]
LIMIT liczba [OFFSET przesunięcie];
```

Kolejność operacji

1. **FROM, JOIN** – złączenie krotek wszystkich źródeł danych,
2. **WHERE** – selekcja wierszy w tabeli wirtualnej utworzonej przez **FROM**,
3. **GROUP BY*** – grupowanie danych na podstawie wartości określonych atrybutów,
4. **HAVING** – selekcja grup,
5. **SELECT** – rzutowanie (wybór) kolumn,
6. **ORDER BY** – sortowanie wyniku,
7. **LIMIT** – ograniczenie liczby krotek.

***GROUP BY** wykonuje agregację danych, co oznacza konieczność wykonania wyrażeń w klauzuli **SELECT**. Od tego punktu stają się dostępne aliasy nadane atrybutom i wyrażeniom w klauzuli **SELECT**.