

SQL Składnia



Agenda

- **SQL**
- Typy danych i operatory
- Podział języka SQL
- Funkcje agregujące
- Zapytania: ORDER, WHERE, GROUP, HAVING





Agenda

- SQL
- **Typy danych i operatory**
- Podział języka SQL
- Funkcje agregujące
- Zapytania: ORDER, WHERE, GROUP, HAVING





Agenda

- SQL
- Typy danych i operatory
- **Podział języka SQL**
- Funkcje agregujące
- Zapytania: ORDER, WHERE, GROUP, HAVING





Agenda

- SQL
- Typy danych i operatory
- Podział języka SQL
- **Funkcje agregujące**
- Zapytania: ORDER, WHERE, GROUP, HAVING





Agenda

- SQL
- Typy danych i operatory
- Podział języka SQL
- Funkcje agregujące
- **Zapytania: ORDER, WHERE, GROUP, HAVING**





Czym jest SQL?

SQL (**S**tructured **Q**uery **L**anguage) jest to język zapytań, który zapewnia komunikację między użytkownikiem lub aplikacją, a relacyjną bazą danych.





Historia

- Lata 60.
- 1974 r.
- 1978 r.
- Ewolucja Standardów
- Rozwój systemów Zarządzania Bazą Danych
- SQL w Open Source
- SQL w Analizie Danych i biznesie





Dlaczego SQL?

- Efektywne zarządzanie danymi
- Prostota użycia
- Uniwersalność
- Jednolitość działania
- Skalowalność
- Optymalizacja zapytań
- Zastosowanie w Analizie Danych



Jak działa SQL?

- Analiza składni
- Optymalizacja zapytania
- Plan wykonawczy
- Wykonanie planu
- Zwrócenie wyniku
- Zarządzanie transakcjami



Zasady formatowania SQL

- **Formatowanie zapytań:**

```
SELECT
    first_name,
    last_name
FROM
    users
WHERE
    age > 25;
```





Zasady formatowania SQL

- **Używanie małych liter:**

```
SELECT
    first_name,
    last_name
FROM
    users
WHERE
    age > 25;
```





Zasady formatowania SQL

- **Nazwy Tabel i Kolumn:**

```
SELECT
    first_name AS "First Name",
    last_name AS "Last Name"
FROM
    user_data
WHERE
    user_age > 25;
```





Zasady formatowania SQL

- **Używanie aliasów:**

```
SELECT
    first_name AS "First Name",
    last_name AS "Last Name",
    birth_date AS "Date of Birth"
FROM
    customers;
```





Zasady formatowania SQL

- Unikaj używania "*":

```
SELECT
    product_name,
    price
FROM
    products;
```





Zasady formatowania SQL

- **Komentarze:**

-- Ponizej wybieramy produkty, których ilość w magazynie przekracza 10

```
SELECT
    product_name,
    stock_quantity
FROM
    products
WHERE
    stock_quantity > 10;
```



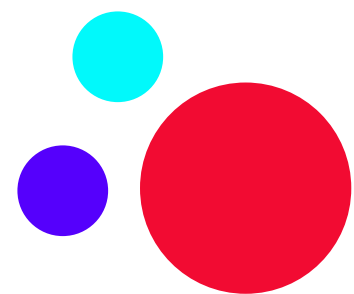
Typy danych

```
CREATE TABLE nazwa_tabeli (  
    Kolumna1 typ_kolumny,  
    Kolumna2 typ_kolumny,  
    ...,  
    KolumnaN typ_kolumny,  
    PRIMARY KEY (kolumna)  
);
```




Typy danych

- typ numeryczny dokładny
- typ numeryczny przybliżony
- typ daty i godziny
- łańcuch znaków
- łańcuch znaków w systemie Unicode
- typ binarny
- inne



SQL Składnia

Typ numeryczny dokładny

Typ	Zakres
bigint	-9,223,372,036,854,775,808 – 9,223,372,036,854,775,807
int	-2,147,483,648 – 2,147,483,647
smallint	-32,768 – 32,767
tinyint	0 – 255
bit	0 – 1
decimal, numeric	$-10^{38} + 1 - 10^{38} - 1$
money	-922,337,203,685,477.5808 – 922,337,203,685,477.5807
smallmoney	-214,748.3648 – 214,748.3647



Typ numeryczny przybliżony

Typ	Zakres
float	$-1.79E + 308 - 1.79E + 308$
real	$-3.40E + 38 - 3.40E + 38$



Typ daty i godziny

Typ	Zakres
datetime	1 Stycznia 1753 – 31 Grudnia 9999
smalldatetime	1 Stycznia 1900 – 6 Czerwca 2079
date	Domyślnym formatem jest: YYYY-MM-DD. Domyślna wartość to 1900-01-01
time	Czas dnia w formacie 15:30 P.M.





Typ łańcuch znaków

Typ	Zakres
char	1 – 8000 znaków (typ stało-znakowy)
varchar	1 – 8000 znaków (typ zmienne-znakowy)
text	Maksymalna długość wynosi 2.147.483.647 znaków





Typ binarny

Typ	Zakres
binary	Maksymalna długość to 8000 bajtów (stały strumień bajtów)
varbinary	Maksymalna długość to 8000 bajtów (zmienny strumień bajtów)
image	Maksymalna długość to 2,147,483,647 bajtów





Inne typy danych

Typ	Zakres
sql_variant	Odpowiednik var w języku C#. Pozwala na przechowywanie różnych typów obsługiwanych przez język SQL.
timestamp	Przechowuje unikalny numer/liczbę binarną dla całej bazy danych. Automatycznie aktualizowany w momencie aktualizacji danego wiersza.
uniqueidentifier	Przechowuje unikalny identyfikator, tzw. GUID.
cursor	Referencja do obiektu kursora, tj. typ danych dla parametrów wyjściowych procedury.
table	Przechowuje zestaw wyników, który może zostać wykorzystany do dalszej obróbki. Typ ten znajduje swoje zastosowanie np. w przypadku transakcji – powiemy o tym na nieco późniejszym etapie.



Operator

- Arytmetyczne
- Porównujące
- Logiczne
- Negujące





Operatory arytmetyczne

- + (Dodawanie)
- - (Odejmowanie)
- * (Mnożenie)
- / (Dzielenie)
- % (Reszta z dzielenia)





Operatory porównujące

- = (Równa się)
- != lub <> (Nie równa się)
- < (Mniejsze niż)
- > (Większe niż)
- <= (Mniejsze lub równe)
- >= (Większe lub równe)



Operatory logiczne

- AND (I logiczne)
- OR (Lub logiczne)
- NOT (Negacja logiczna)





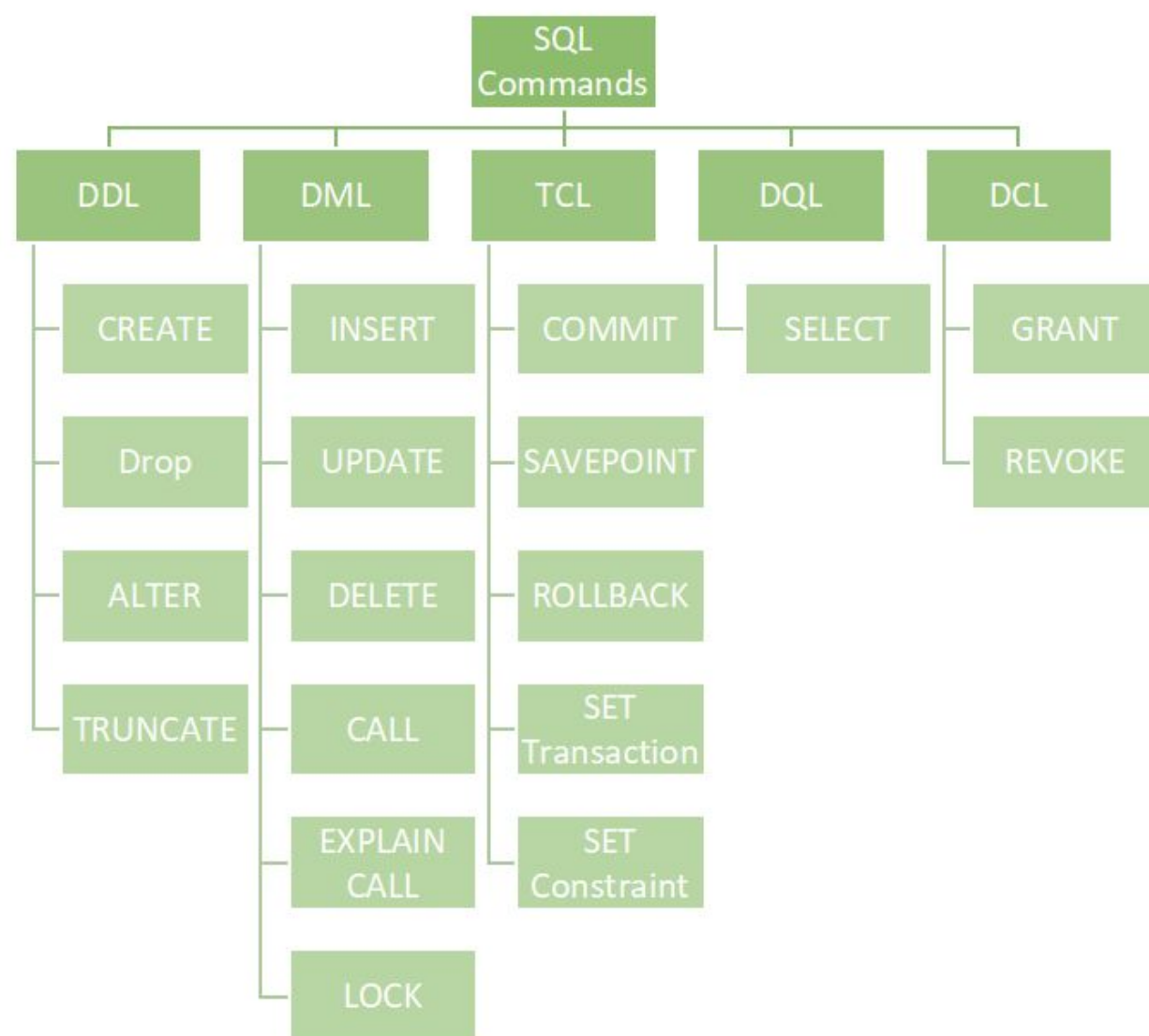
Operatory negujące

- IS NULL (Jest puste)
- IS NOT NULL (Nie jest puste)
- NOT (Negacja)





Podział języka SQL





Data Definition Language

- CREATE
- DROP
- ALTER
- TRUNCATE





Data Definition Language

```
CREATE TABLE Pracownicy (  
    ID INT PRIMARY KEY,  
    Imie VARCHAR(50),  
    Nazwisko VARCHAR(50),  
    Wiek INT  
);
```

```
DROP TABLE Pracownicy;
```

```
ALTER TABLE Pracownicy  
ADD Telefon VARCHAR(15);
```

```
TRUNCATE TABLE Pracownicy;
```




Zadanie 2.1 DDL (instrukcja)

- Stwórz tabelę o nazwie Studenci, gdzie kolumnami są kolejno ID (klucz główny tabeli), Imie (kolumna VARCHAR max 50 znaków), Nazwisko (kolumna VARCHAR max 50 znaków) oraz Wiek (kolumna Int).
- Do nowo utworzonej tabeli przy użyciu ALTER dodaj kolumnę o nazwie Email (VARCHAR max 100 znaków).
- Zmień typ kolumny Wiek na SMALLINT przy użyciu ALTER.
- Usuń wszystkie rekordy z tabeli Studenci za pomocą TRUNCATE.
- Usuń tabelę Studenci przy użyciu DROP.



Data Manipulation Language

- INSERT
- UPDATE
- DELETE
- CALL
- EXPLAIN CALL
- LOCK





Data Manipulation Language

```
INSERT INTO Pracownicy (Imie, Nazwisko, Wiek) VALUES ('Anna',  
'Kowalska', 30);
```

```
UPDATE Pracownicy SET Wiek = 31 WHERE Nazwisko = 'Kowalska';
```

```
DELETE FROM Pracownicy WHERE Nazwisko = 'Kowalska';
```

```
CALL DodajPracownika('Jan', 'Nowak', 25);
```

```
EXPLAIN CALL DodajPracownika('Jan', 'Nowak', 25);
```

```
LOCK TABLE Pracownicy IN SHARE MODE;
```





Zadanie 2.2 DML (instrukcja)

- **Stwórz tabelę Studenci za pomocą poniższej query:**

```
CREATE TABLE Studenci (  
    ID INT PRIMARY KEY,  
    Imie VARCHAR(50),  
    Nazwisko VARCHAR(50),  
    Wiek INT  
);
```

- **Dodaj do tabeli nowy rekord za pomocą funkcji INSERT:**
ID: 1, Imie: Joanna, Nazwisko: Nowak, Wiek: 20
- **Zmodyfikuj rekord z pomocą UPDATE – dla ID: 1 zmień wiek na 21 lat**
- **Usuń rekord za pomocą DELETE dla ID: 1**



Transaction Control Language

- COMMIT
- SAVEPOINT
- ROLLBACK
- SET Transaction
- SET Constraint





Data Control Language

- GRANT
- REVOKE





Data Query Language

- SELECT





Przykładowe zapytania SELECT

```
SELECT * FROM table;
```

```
SELECT column1, column2, ...columnN FROM table;
```



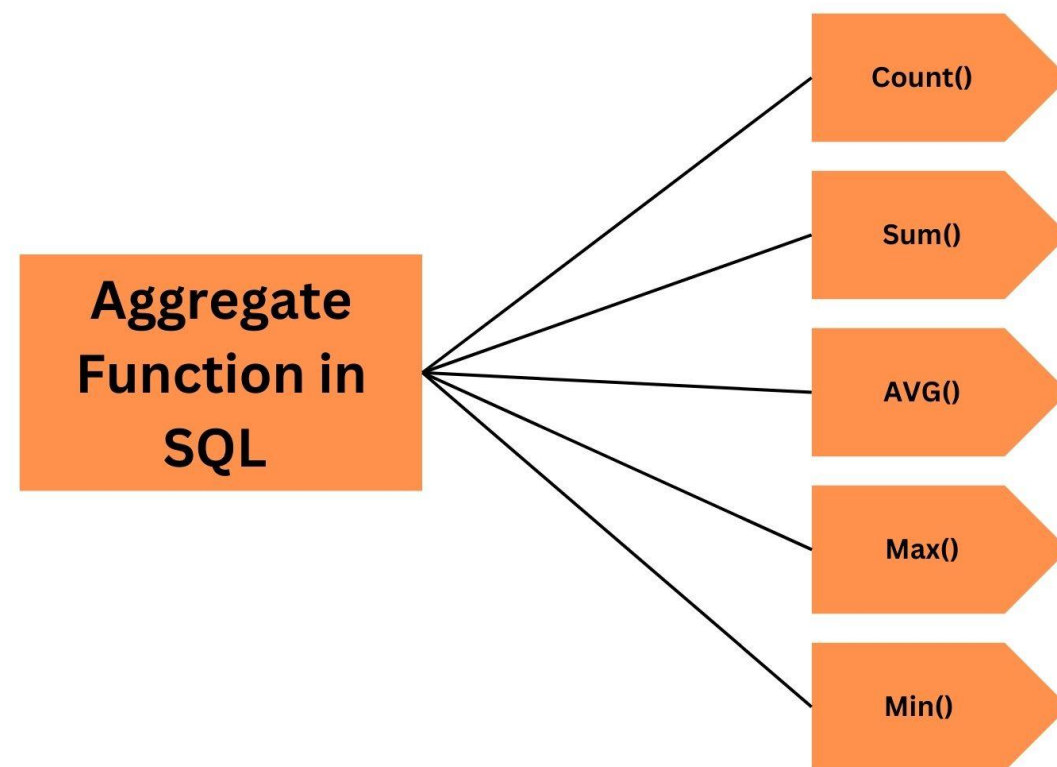


Zadanie 2.3 SELECT (instrukcja)

- Otwórz zbiór danych (all_seasons) w DBeaver o koszykarzach NBA, który zaimportowałeś podczas preworku.
- Otwórz nowy skrypt SQL.
- Przy użyciu SELECT wybierz wszystkie kolumny z tabeli all_seasons.
- W kolejnym zapytaniu przy użyciu SELECT wybierz kolumny z nazwiskami koszykarzy, ich wiek, wzrost oraz wagę.

Funkcje agregujące

- COUNT
- DISTINCT
- MAX
- MIN
- AVG



BOARD



Funkcje agregujące – COUNT

- **Podstawowe zliczanie wszystkich wierszy w tabeli:**

```
SELECT COUNT(*) FROM Customers;
```

- **Zliczanie unikalnych wartości w kolumnie:**

```
SELECT COUNT(DISTINCT Country) FROM Customers;
```



Funkcje agregujące – DISTINCT

- **Zliczanie unikalnych produktów w kategorii:**

```
SELECT Category, COUNT(DISTINCT ProductName) AS  
UniqueProducts  
FROM Products  
GROUP BY Category;
```

- **Lista unikalnych klientów z zamówieniami:**

```
SELECT DISTINCT CustomerName  
FROM Customers  
WHERE CustomerID IN (SELECT DISTINCT CustomerID FROM  
Orders);
```




Zadanie 2.4 COUNT (instrukcja)

- Dla tabeli all_seasons o koszykarzach NBA – sprawdź ile wynosi liczba wszystkich rekordów w tabeli.
- Sprawdź ile wynosi liczba koszykarzy w tabeli all_seasons.
- Sprawdź jaka jest unikalna lista krajów, z których pochodzą koszykarze.
- Sprawdź jaka jest unikalna liczba kolegów koszykarzy NBA.



Funkcje agregujące – MAX

- **Znalezienie najwyższej wartości w kolumnie "Price,,:**

```
SELECT MAX(Price) AS HighestPrice  
FROM Products;
```

- **Znalezienie najnowszego zamówienia:**

```
SELECT MAX(OrderDate) AS LatestOrderDate  
FROM Orders;
```

- **Znalezienie najwyższego wynagrodzenia pracownika:**

```
SELECT MAX(Salary) AS HighestSalary  
FROM Employees;
```



Funkcje agregujące – MIN

- **Znalezienie najniższej wartości w kolumnie "Price,,:**

```
SELECT MIN(Price) AS LowestPrice  
FROM Products;
```

- **Znalezienie najwcześniejszej daty zamówienia:**

```
SELECT MIN(OrderDate) AS EarliestOrderDate  
FROM Orders;
```

- **Znalezienie najniższego wynagrodzenia pracownika:**

```
SELECT MIN(Salary) AS LowestSalary  
FROM Employees;
```




Funkcje agregujące – AVG

- **Średnia wieku pracowników:**

```
SELECT AVG(Age) AS AverageAge  
FROM Employees;
```

- **Średni czas trwania projektów:**

```
SELECT AVG(Duration) AS AverageDuration  
FROM Projects;
```

- **Średnia ocen filmów:**

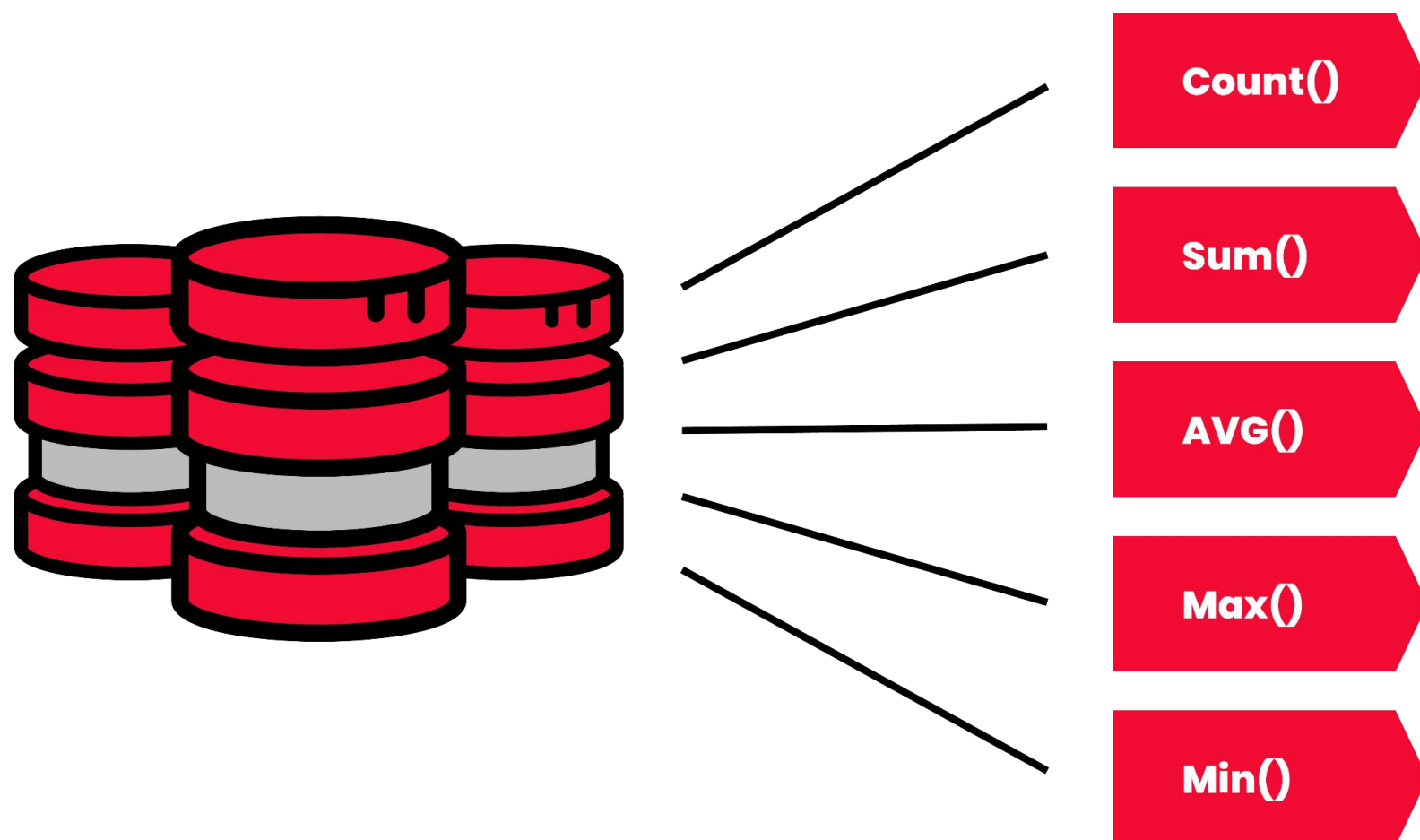
```
SELECT AVG(Rating) AS AverageRating  
FROM Movies;
```



Zadanie 2.5 AVG (instrukcja)

- Dla zbioru danych all_seasons o koszykarzach NBA sprawdź ile wynosi najmniejszy, największy oraz średni wzrost koszykarza NBA.
- Sprawdź ile wynosi najmniejsza, największa oraz średnia waga koszykarza NBA.
- Sprawdź ile wynosi najmniejsza, największa oraz średnia liczba punktów zdobytych przez koszykarzy.

Funkcje agregujące – podsumowanie





Struktura zapytań

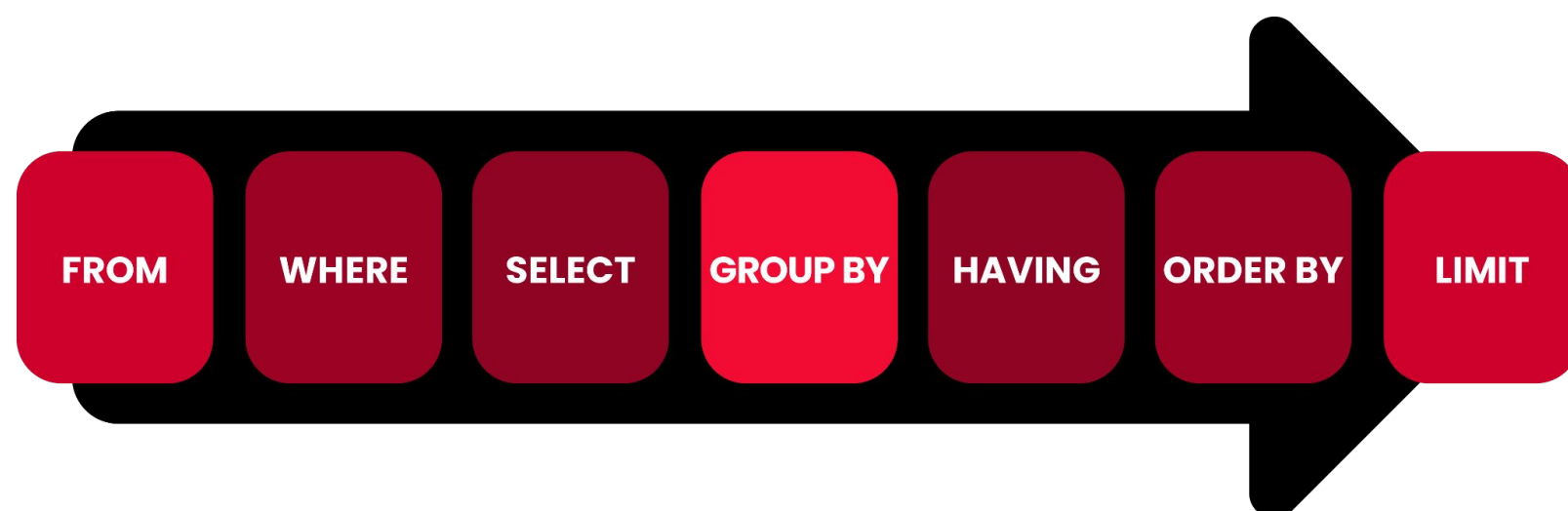
- ORDER
- WHERE
- GROUP BY
- HAVING





Struktura zapytań

1. SELECT Kategoria, AVG(Cena) as SredniaCena
2. FROM Produkty
3. WHERE Cena > 50
4. GROUP BY Kategoria
5. HAVING COUNT(*) > 2
6. ORDER BY SredniaCena DESC;





ORDER BY

- **Sortowanie po jednej kolumnie:**
SELECT Imie, Nazwisko, Wiek FROM Pracownicy ORDER BY Nazwisko ASC;
- **Sortowanie po wielu kolumnach:**
SELECT Imie, Nazwisko, Wiek FROM Pracownicy ORDER BY Dzial ASC, Wiek DESC;
- **Sortowanie tekstu nieuwzględniającego wielkości liter:**
SELECT NazwaProduktu FROM Produkty ORDER BY LOWER(NazwaProduktu) ASC;
- **Sortowanie dat:**
SELECT Tytuł, DataPremiery FROM Filmy ORDER BY DataPremiery DESC;



Zadanie 2.6 ORDER BY (instrukcja)

- Przy użyciu zbioru danych all_seasons posortuj malejąco koszykarzy według liczby zdobytych punktów w sezonie.
- Znajdź 10 najnowszych rekordów w zbiorze według sezonu.
- Posortuj koszykarzy według wzrostu, a następnie według liczby zdobytych punktów.



WHERE

- **Filtrowanie danych numerycznych:**
`SELECT * FROM Klienci WHERE Wiek > 30;`
- **Filtrowanie danych tekstowych:**
`SELECT * FROM Produkty WHERE Kategoria = 'Elektronika';`
- **Filtrowanie danych z użyciem operatorów logicznych:**
`SELECT * FROM Produkty WHERE Cena > 100 AND Dostepnosc = 'Dostępny';`
- **Filtrowanie dat:**
`SELECT * FROM Wydarzenia WHERE DataRozpoczecia >= '2023-01-01';`



WHERE – in / between

- in / between
 - in (X,Y,Z) / between X and Z
 - where liczba_pasazerow in (2, 3)
 - where liczba_pasazerow between 2 and 5

SELECT * FROM tabela WHERE kolumna IN ('wartość1', 'wartość2', 'wartość3');

SELECT * FROM tabela WHERE kolumna BETWEEN
wartość_min AND wartość_max;



WHERE – warunki like / ilike

- Warunki (wyszukiwanie tekstu):
 - like – wyszukiwanie wzorca w tekście (czułe na wielkość liter): like 'Gd%'
 - ilike – wyszukiwanie wzorca w tekście (nieczułe na wielkość liter): ilike 'Gd%'

```
SELECT * FROM tabela WHERE kolumna LIKE 'wartość%';
```

```
SELECT * FROM tabela WHERE kolumna ILIKE 'Wartość%';
```



WHERE – wildcards

- wildcards dla like to % i _
 - % – dowolna liczba znaków
 - _ – jeden znak

SELECT * FROM tabela WHERE kolumna LIKE 'wartość%';
Dowolna liczba znaków

SELECT * FROM tabela WHERE kolumna LIKE '_wartość';
Jeden znak



Zadanie 2.7 WHERE (instrukcja)

- Dla zbioru danych all_seasons:
 - Znajdź wszystkich koszykarzy, którzy zdobyli więcej niż 30 punktów.
 - Znajdź koszykarzy, których imię zaczyna się na literę „L” i kończy na „y”.
 - Znajdź koszykarzy, którzy są z drużyny LAL oraz GSW.
 - Znajdź koszykarzy, których wzrost mieści się między 200 a 220 cm.



GROUP BY

- **Grupowanie według kolumny numerycznej:**
SELECT Dzial, AVG(Wynagrodzenie) as
SrednieWynagrodzenie FROM Pracownicy GROUP BY
Dzial;
- **Grupowanie według kolumny tekstowej:**
SELECT Kategoria, COUNT(*) as IloscProduktow FROM
Produkty GROUP BY Kategoria;
- **Grupowanie według wielu kolumn:**
SELECT Kategoria, Producent, AVG(Cena) as
SredniaCena FROM Produkty GROUP BY Kategoria,
Producent;



GROUP BY

- **Grupowanie z filtrowaniem:**

```
SELECT Kategoria, Producent, COUNT(*) as  
IloscProduktow FROM Produkty WHERE Cena > 50  
GROUP BY Kategoria, Producent;
```

- **Grupowanie po funkcji agregującej:**

```
SELECT Producent, MAX(Ocena) as  
NajwyzszaOcena FROM Recenzje GROUP BY  
Producent;
```

HAVING

- **Filtrowanie grup o określonym warunku:**

```
SELECT Kategoria, AVG(Cena) as SredniaCena  
FROM Produkty GROUP BY Kategoria HAVING  
AVG(Cena) > 50;
```

- **Filtrowanie grup po liczbie elementów:**

```
SELECT Producent, COUNT(*) as IloscRecenzji  
FROM Recenzje GROUP BY Producent HAVING  
COUNT(*) >= 5;
```




HAVING

- **Filtrowanie grup po agregatach:**

```
SELECT Dzial, AVG(Wynagrodzenie) as  
SrednieWynagrodzenie FROM Pracownicy GROUP  
BY Dzial HAVING AVG(Wynagrodzenie) > 60000;
```

- **Filtrowanie grup po warunkach logicznych:**

```
SELECT Kategoria, AVG(Cena) as SredniaCena  
FROM Produkty GROUP BY Kategoria HAVING  
AVG(Cena) > 50 AND MAX(Ilosc) > 10;
```



Zadanie 2.8 GROUP BY (instrukcja)

- Dla zbioru danych all_seasons:
 - Znajdź liczbę unikalnych drużyn oraz ich liczbę występów – posortuj dane malejąco.
 - Znajdź średnią liczbę punktów zdobytych przez koszykarzy w każdym sezonie, ale wyłącznie z tych sezonów, w których średnia zdobytych punktów była większa od 20.



Zadanie 2.9 Podsumowanie (instrukcja)

Znajdź drużyny NBA, które w sezonie 2020–21 miały średnią ilość zdobytych punktów na mecz większą niż 8 oraz łączną liczbę zdobytych punktów większą niż 150.

Wyniki posortuj malejąco według średniej punktów na mecz.



Podsumowanie

