



PODSTAWY SQL

WYKŁAD 3

MGR INŻ. BARTŁOMIEJ DETTLAFF

SENIOR JAVA DEVELOPER

AGENDA

- Funkcje agregujące
- Indeksy
- Klauzula group by
- Klauzula order by
- Klauzula having
- Klauzula limit

Funkcje agregujące - czym są funkcje agregujące?

Funkcje agregujące w SQL służą do wykonywania obliczeń na wielu wierszach zbioru wynikowego i zwracania jednego skumulowanego wyniku. Są one często używane w połączeniu z instrukcjami GROUP BY, aby grupować dane według określonych kolumn i obliczać funkcje agregujące dla każdej grupy.

Funkcje agregujące - count

Przykład:

```
SELECT COUNT(*)  
FROM Products  
WHERE release_date > 2021
```

Funkcje agregujące - sum

Przykład:

```
SELECT SUM(price)
```

```
FROM Products
```

Funkcje agregujące - avg

Przykład:

```
SELECT AVG(age)  
FROM Clients
```

Funkcje agregujące - max

Przykład:

```
SELECT MAX(price)
```

```
FROM Products
```

Funkcje agregujące - min

Przykład:

```
SELECT MIN(price)
```

```
FROM Products
```


Indeksy - co to jest?

Indeksy w SQL to specjalne struktury, które mogą być tworzone w bazach danych w celu zwiększenia wydajności operacji wyszukiwania. Gdy mówimy o indeksach w kontekście baz danych, można myśleć o nich analogicznie do indeksu w książce: zamiast przeszukiwać całą książkę, aby znaleźć określone informacje, można spojrzeć na indeks, aby szybko zidentyfikować, gdzie znajdują się te informacje.

Indeksy - kilka informacji cz. 1

- **Wydajność zapytań:** Głównym celem indeksów jest zwiększenie szybkości operacji wyszukiwania w tabelach. Bez indeksu, baza danych musi przeszukać każdy wiersz tabeli, aby znaleźć interesujące nas dane (tzw. pełne skanowanie tabeli). Indeks pozwala bazie danych na szybsze zlokalizowanie wierszy.

Indeksy - kilka informacji cz. 2

- **Struktura:** Większość indeksów w bazach danych jest tworzona na podstawie struktur drzewa (np. B-drzewa). Pozwala to na bardzo szybkie operacje wyszukiwania, dodawania i usuwania.
- **Unikalność:** Indeksy mogą być używane do zapewnienia unikalności wartości w określonej kolumnie (np. unikalny identyfikator). Jeśli próbujesz wstawić dane, które naruszają tę unikalność, operacja zostanie odrzucona.

Indeksy - kilka informacji cz. 3

- **Klucze obce:** Indeksy są często tworzone automatycznie dla kolumn, które są kluczami obcymi, co pomaga w zachowaniu integralności odniesienia między tabelami.
- **Typy indeksów:** Wiele systemów baz danych oferuje różne typy indeksów, takie jak indeksy klastrujące, nieklastrujące, pełnotekstowe i przestrzenne, każdy z nich optymalizowany do obsługi określonych rodzajów zapytań.

Indeksy - kilka informacji cz. 4

- **Koszty:** Chociaż indeksy przyspieszają operacje odczytu, mają pewne koszty. Zajmują dodatkową przestrzeń dyskową, a operacje takie jak wstawianie, aktualizacja lub usuwanie rekordów mogą być wolniejsze, ponieważ struktury indeksu również muszą być aktualizowane.

Indeksy - kilka informacji cz. 4

- **Koszty:** Chociaż indeksy przyspieszają operacje odczytu, mają pewne koszty. Zajmują dodatkową przestrzeń dyskową, a operacje takie jak wstawianie, aktualizacja lub usuwanie rekordów mogą być wolniejsze, ponieważ struktury indeksu również muszą być aktualizowane.

Indeksy - tworzenie indeksu

Przykład:

```
CREATE INDEX idx_surname ON employee (surname);  
CREATE UNIQUE INDEX idx_email ON employee (email);
```


Indeksy - usuwanie indeksu

Przykład:

```
DROP INDEX idx_surname ON employee;
```


Indeksy - wyświetlanie indeksów

Przykład:

```
SHOW INDEX FROM employee;
```

Klauzula group by

Klauzula GROUP BY w SQL jest używana do grupowania wierszy, które mają te same wartości w określonych kolumnach, w celu przeprowadzenia operacji agregujących, takich jak liczenie, sumowanie czy obliczanie Średniej.

Klauzula group by - przykład

Przykład: Załóżmy, że masz tabelę Employees z kolumnami id, name, surname i department.

Jeśli chcesz dowiedzieć się, ile osób pracuje w każdym dziale:

```
SELECT department, COUNT(id) AS number_of_employees  
FROM Employees  
GROUP BY department;
```

Klauzula group by - przykład tabeli

id	name	surname	department
1	Jan	Kowalski	IT
2	Anna	Nowak	HR
3	Piotr	Zieliński	IT
4	Marta	Wiśniewska	Phinance
5	Marcin	Wójcik	HR

Klauzula group by - wynik zapytania

department	number_of_employees
IT	2
HR	2
Phinance	1

Klauzula order by

Klauzula ORDER BY w SQL służy do sortowania wyników zapytania według jednej lub więcej kolumn. Możesz określić kierunek sortowania dla każdej kolumny osobno, używając słów kluczowych ASC (rosnąco, jest to domyślny kierunek) lub DESC (malejąco).

Klauzula order by - przykład

Przykład: Załóżmy, że masz tabelę Employees z kolumnami id, name, surname i department.

Jeśli chcesz posortować wyniki według nazwiska pracowników w kolejności alfabetycznej:

```
SELECT name, surname  
FROM Employees  
ORDER BY surname;
```

Klauzula order by - przykład tabeli

id	name	surname	department
1	Jan	Kowalski	IT
2	Anna	Nowak	HR
3	Piotr	Zieliński	IT
4	Marta	Wiśniewska	Phinance
5	Marcin	Wójcik	HR

Klauzula order by - wynik zapytania

id	name	surname	department
1	Jan	Kowalski	IT
2	Anna	Nowak	HR
4	Marta	Wiśniewska	Phinance
5	Marcin	Wójcik	HR
3	Piotr	Zieliński	IT

Klauzula having

Klauzula **HAVING** w SQL jest używana do filtrowania wyników zapytania po przeprowadzeniu operacji agregujących. W praktyce, **HAVING** jest często używane w połączeniu z **GROUP BY**, chociaż nie jest to wymagane.

Jeśli porówna się to z innymi klauzulami: **WHERE** filtruje wiersze przed przeprowadzeniem operacji agregujących, podczas gdy **HAVING** filtruje wiersze po przeprowadzeniu tych operacji.

Klauzula having - przykład

Przykład: Załóżmy, że masz tabelę Employees z kolumnami id, name, surname i department.

Jeśli chcesz dowiedzieć się, w których działach pracuje więcej niż 5 osób, możesz użyć następującego zapytania:

```
SELECT department, COUNT(id) as number_of_employees  
FROM Employees  
GROUP BY department  
HAVING number_of_employees > 5;
```

Klauzula having - przykład tabeli

id	name	surname	department
1	Jan	Kowalski	IT
2	Anna	Nowak	HR
3	Piotr	Zieliński	IT
4	Marta	Wiśniewska	Phinance
5	Marcin	Wójcik	HR

Klauzula having - wynik zapytania

department	number_of_employees
IT	2
HR	2

Klauzula limit

W SQL klauzula LIMIT jest używana do ograniczenia liczby zwracanych wierszy w wyniku zapytania. Jest to szczególnie przydatne w scenariuszach, gdzie interesuje nas tylko pewna liczba wierszy z dużego zbioru danych, na przykład przy paginacji wyników w aplikacjach internetowych.

Klauzula limit - przykład

Przykład: Jeśli chcesz zwrócić tylko pierwsze 10 wierszy:

```
SELECT * FROM Employees LIMIT 10;
```