

## 0 mnie

- programista w CGM Polska (Java/AngularJS)
- aktualnie pracuję w projekcie dotyczącym systemu HIS (Hospitality Information System) dla szpitali w Europie (głównie Austria, Niemcy oraz Szwajcaria)
- 6 lat doświadczenia z bazami danych (Oracle, MySQL, MSSQL)



## Baza danych

**Baza danych** to uporządkowany zbiór danych z pewnej dziedziny tematycznej, zorganizowany w sposób umożliwiający ich wyszukiwanie według zadanych kryteriów.



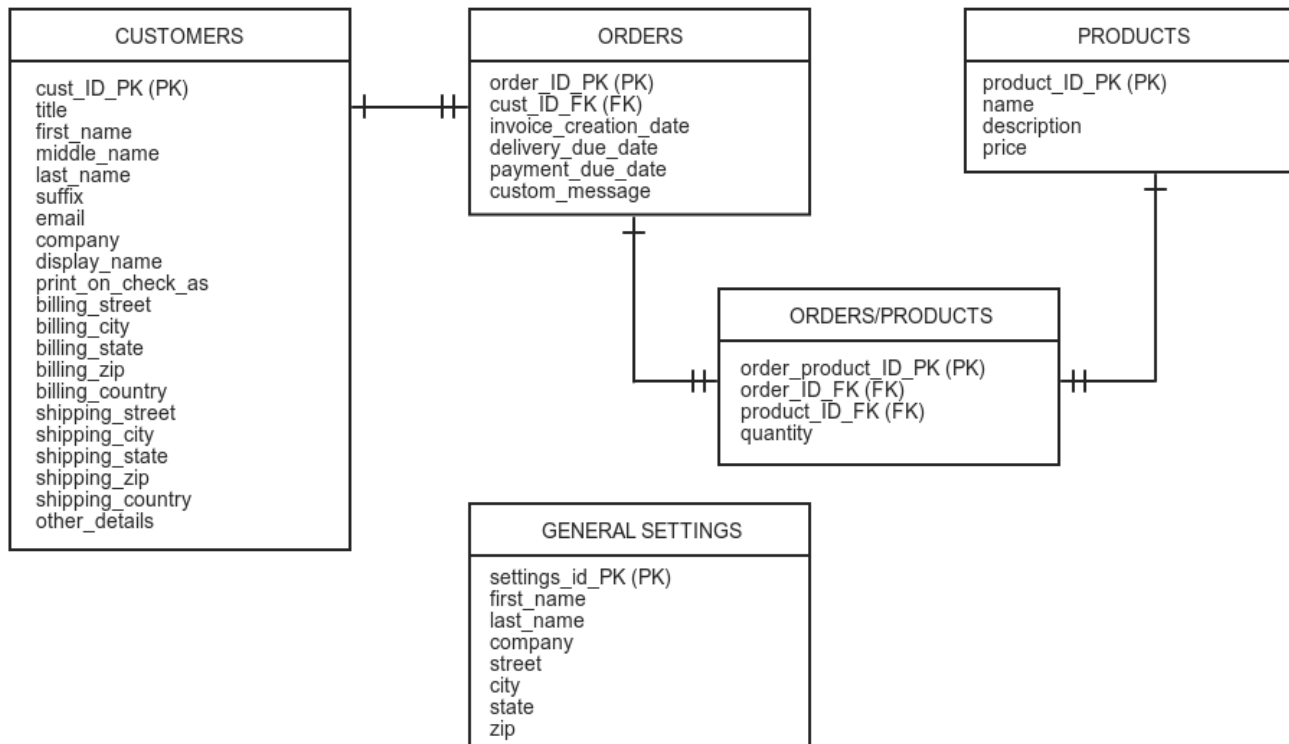
## Rodzaje baz danych

**Relacyjną bazą danych** nazywamy bazę danych w postaci tabel połączonych relacjami.

Pod pojęciem **bazy nierelacyjnej** (NoSQL database) najczęściej rozumie się przechowywanie danych w formie listy par obiektów klucz-wartość, w których nie występują powiązania relacyjne między przechowywanymi obiektami.



# Relacyjna baza danych



## Co to jest SQL?

**SQL** jest standardowym językiem do **przechowywania, manipulowania i pobierania** danych w bazach danych.



## Co to jest SQL?

- SQL - Structured Query Language
- SQL umożliwia uzyskiwanie dostępu i manipulowanie bazami danych
- SQL jest standardem ANSI (American National Standards Institute)



## Co można zrobić przy pomocy SQL?

- **wykonywać** zapytania na bazie danych
- **pobierać** dane z bazy danych
- **wstawiać** rekordy do bazy danych
- **aktualizować** rekordy w bazie danych
- **usuwać** rekordy z bazy danych



## Co można zrobić przy pomocy SQL?

- tworzyć nowe **bazy** danych
- tworzyć nowe **tabele** w bazie danych
- tworzyć **procedury** przechowywane w bazie danych
- tworzyć **widoki** w bazie danych
- ustawiać **uprawnienia** do tabel, procedur, widoków





## SQL jest standardem - ale ....

- Choć SQL jest standardem ANSI (American National Standards Institute), istnieją różne wersje języka SQL.
- Jednakże, aby być zgodnym ze standardem ANSI, wszystkie one wspierają co najmniej najważniejsze polecenia (takie jak SELECT, UPDATE, DELETE, INSERT, WHERE) w podobny sposób.



## Co oznacza RDBMS?

**RDBMS (Relational Database Management System)** - oznacza system zarządzania relacyjnymi bazami danych.

- RDBMS jest podstawą SQL i dla wszystkich nowoczesnych systemów baz danych, takich jak MS SQL Server, IBM DB2, Oracle, MySQL i Microsoft Access.



## Co oznacza RDBMS?

Dane w RDBMS są przechowywane w obiektach bazy danych zwanych tabelami.

**Tabela** to zbiór powiązanych wpisów danych i składa się z kolumn i wierszy.



## Tabele i kolumny

- Każda **tabela** jest podzielona na mniejsze jednostki nazywane **polami**.
- Pole to **kolumna** w tabeli, która ma na celu przechowywanie szczegółowych informacji o każdym rekordzie w tabeli.



## Podział SQL

- **DDL** (Data Definition Language) - wyrażenia służą do tworzenia obiektów bazy danych (tabel, kluczy powiązań...)
- **DML** (Data Manipulation Language) - służy do modyfikowania danych
- **DCL** (Data Control Language) - sterowanie dostępem do danych.



## Weź pod uwagę, że...

Słowa kluczowe SQL nie uwzględniają wielkości liter: `select` jest taki sam jak `SELECT`.

Starajcie się jednak pisać wszystkie słowa kluczowe SQL dużymi literami.



## Średnik po instrukcji SQL?

Niektóre systemy baz danych wymagają średnika na końcu każdej instrukcji SQL.

Średnik to standardowy sposób oddzielania każdej instrukcji SQL w systemach baz danych, które umożliwiają wykonywanie więcej niż jednej instrukcji SQL w tym samym połączeniu do serwera.



## SELECT

Instrukcja **SELECT** służy do wybierania danych z bazy danych.

Zwracane dane są przechowywane w tabeli wyników, zwanej zestawem wyników.





## SELECT - SKŁADNIA

```
SELECT column1, column2, ...  
FROM table_name;
```

W tym przypadku kolumna1, kolumna2, ... to nazwy pól tabeli, z których chcesz wybrać dane. Jeśli chcesz zaznaczyć wszystkie pola dostępne w tabeli, użyj następującej składni:

```
SELECT * FROM table_name;
```



## SELECT DISTINCT

Instrukcja **SELECT DISTINCT** jest używana do zwracania tylko różnych (różnych) wartości.

Wewnątrz tabeli, kolumna często zawiera wiele podwójnych wartości, a czasami chcesz tylko wymienić różne (odrębne) wartości.

```
SELECT DISTINCT column1, column2, ...  
FROM table_name;
```



## WHERE

Klauzula **WHERE** służy do filtrowania rekordów. Jest używana do wyodrębniania tylko tych rekordów, które spełniają określone warunki.

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```



## Operatory AND, OR oraz NOT

Klauzula **WHERE** może być łączona z operatorami **AND**, **OR** i **NOT**.

Operatory **AND** i **OR** są używane do **filtrowania** rekordów na podstawie więcej niż jednego warunku.

Operator **NOT** wyświetla rekord, jeśli warunek nie jest **TRUE**.



## Operatory AND, OR oraz NOT

Operatory **AND** i **OR** mogą być używane do filtrowania rekordów na podstawie więcej niż jednego warunku:

- Operator **AND** wyświetla rekord, jeśli wszystkie warunki oddzielone przez **AND** są **TRUE**.
- Operator **OR** wyświetla rekord, jeśli któryś z warunków oddzielonych przez **OR** jest **TRUE**



## Operatory AND, OR oraz NOT

*SELECT column1, column2, ...  
FROM table\_name  
WHERE condition1 AND condition2 AND condition3 ...;*

*SELECT column1, column2, ...  
FROM table\_name  
WHERE condition1 OR condition2 OR condition3 ...;*

*SELECT column1, column2, ...  
FROM table\_name  
WHERE NOT condition;*



## LIKE

Operator **LIKE** jest używany w klauzuli **WHERE** do wyszukiwania określonego wzoru w kolumnie.

W połączeniu z operatorem LIKE istnieją dwa **symbole wieloznaczne**:

% - znak procentowy oznacza zero, jeden lub wiele znaków  
\_ - podkreślenie oznacza pojedynczy znak



## LIKE

Znak procentowy i podkreślenie mogą być również używane w połączeniach! Można również łączyć dowolną liczbę warunków z operatorami AND lub OR.

```
SELECT column1, column2, ...  
FROM table_name  
WHERE column LIKE pattern;
```

Można również łączyć dowolną liczbę warunków z operatorami AND lub OR.





## IN

Operator **IN** umożliwia określenie wielu wartości w klauzuli WHERE. Operator **IN** jest skrótem dla wielu warunków OR.



## IN

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name IN (value1, value2, ...);
```

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name IN (SELECT STATEMENT);
```



## BETWEEN

Operator **BETWEEN** wybiera wartości w określonym zakresie. Wartościami mogą być liczby, tekst lub daty.

Operator **BETWEEN** do przedziału włącza początkowe i końcowe wartości.

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name BETWEEN value1 AND value2;
```



## ALIAS

**Alias**y SQL są używane do nadawania tabeli lub kolumny w tabeli, tymczasowej nazwy.

**Alias**y są często używane do nadawania bardziej czytelnych nazw kolumnom.

**Alias** istnieje tylko na czas trwania zapytania.



## ALIAS

*SELECT column\_name AS alias\_name  
FROM table\_name;*

*SELECT column\_name(s)  
FROM table\_name AS alias\_name;*



## ORDER BY

Słowo kluczowe **ORDER BY** służy do sortowania zestawu wyników w porządku rosnącym lub malejącym.

**ORDER BY** domyślnie sortuje rekordy w porządku rosnącym (**ASC**). Aby sortować rekordy w kolejności malejącej, użyj słowa kluczowego **DESC**.



## ORDER BY

```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1, column2, ... ASC/DESC;
```



## WARTOŚĆ NULL

Pole z wartością **NULL** to pole bez wartości.

Jeśli pole w tabeli jest opcjonalne, można wstawić nowy rekord lub zmodyfikować rekord bez dodawania wartości do tego pola. Następnie pole zostanie zapisane z wartością **NULL**.





## WARTOŚĆ NULL

**Uwaga:** Bardzo ważne jest, aby zrozumieć, że wartość NULL różni się od wartości zerowej lub pola zawierającego spację. Pole z wartością NULL jest puste, podczas tworzenia rekordów!



## WARTOŚĆ NULL

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NULL;
```

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NOT NULL;
```



## MAX oraz MIN

Funkcja **MIN ()** zwraca najmniejszą wartość wybranej kolumny.

Funkcja **MAX ()** zwraca największą wartość wybranej kolumny.

