

# INTERNETOWY KURS **PODSTAW IT**

LEKCJA 13.

## **KWERENDY I KROTKI. WPROWADZENIE DO BAZ DANYCH.**

Witaj po raz trzynasty!

Poprzednia lekcja przybliżyła Ci idee wykorzystania frameworków w procesie wytwarzania oprogramowania. Zdecydowana większość aplikacji powstających w tym procesie do poprawnego działania, oprócz kodu źródłowego, potrzebuje też bazy danych. Celem dzisiejszej lekcji jest zapoznanie Ciebie z najważniejszymi pojęciami z tego obszaru.

### **DEFINICJA**

**Baza danych** – zbiór danych zapisanych zgodnie z określonymi regułami. W węższym znaczeniu obejmuje dane cyfrowe gromadzone zgodnie z zasadami przyjętymi dla danego programu komputerowego specjalizowanego do gromadzenia i przetwarzania tych danych.

([https://pl.wikipedia.org/wiki/Baza\\_danych](https://pl.wikipedia.org/wiki/Baza_danych))

Książka telefoniczna, rozkład jazdy autobusów oraz katalog biblioteczny to dobre przykłady zbiorów danych zapisanych zgodnie z określonymi regułami. Również w przypadku cyfrowych baz danych wykorzystywanych w świecie IT, przetrzymywane w nich informacje muszą przestrzegać określonych zasad (np. dotyczących formatu i struktury danych). Najczęściej wykorzystywanym rodzajem baz danych są **bazy relacyjne** (ang. *relational database*) i to im poświęcimy dzisiaj najwięcej uwagi.

## NAJPOPULARNIEJSZE RODZAJE BAZ DANYCH

- Bazy proste:
  - kartotekowe
  - hierarhiczne
- Bazy złożone:
  - relacyjne
  - nierelacyjne (noSQL)
  - obiektowe
  - relacyjno-obiektowe
  - strumieniowe
  - temporalne

Podstawowymi elementami relacyjnej bazy danych są rekordy (inaczej krotki) oraz tabele. Baza danych to zbiór tabel, natomiast tabela to zbiór rekordów.

Wyobraź sobie, że pracujesz nad interaktywną książką telefoniczną. W przypadku relacyjnej bazy danych wszystkie Twoje kontakty mogłyby być zapisane w jednej tabeli nazwanej np. `kontakty`. Szczegóły każdego kontaktu byłyby zapisane w pojedynczym rekordzie. Rekord to zbiór szczegółów dotyczących kontaktu, np. imię, nazwisko oraz numer telefonu:

Id	Imie	nazwisko	telefon
1	Jan	Kowalski	612 345 342
2	Ola	Nowak	604 326 498
3	Tomek	Kowalczyk	513 455 667
4	Natalia	Robak	601 325 987

Przykładowa tabela kontakty relacyjnej bazy danych

Opracowanie własne.

Wiersze tabeli to rekordy, natomiast kolumny to tzw. atrybuty – części tabeli, które przechowują pojedyncze informacje rekordu (np. nazwisko).

Jak widzisz, kontaktów (krotkom) o imieniu Jan w jednej tabeli może być wiele. Na podstawie samego imienia nie będziemy w stanie rozróżnić o którego Jana nam chodzi i podać właściwego numeru telefonu. Znasz też pewnie przynajmniej jedną parę osób, którzy nazywają się dokładnie tak samo, np. dwóch Tomków Kowalczyków. Żeby sprawnie pytać bazę o informacje, musimy powiedzieć jej precyzyjnie, czego

szukamy. Jeśli nadamy krotkom unikalne identyfikatory, sprawa będzie prosta. Na pewno zauważyłeś, że pierwszą kolumną powyższej tabeli jest kolumna `id` – jest to tzw. **klucz podstawowy**, który odgrywa bardzo istotną rolę w relacyjnych baza danych.

## DEFINICJA

**Klucz podstawowy** (ang. *primary key*) zwany też kluczem głównym to jedno lub więcej pól, których wartość jednoznacznie identyfikuje każdy rekord w tabeli. (<http://miroslawzelent.pl/wyklady/bazy-danych-podstawy-typy-danych-relacja-sql-rekord-klucz.pdf>)

Klucz podstawowy musi być unikatowy – w jednej tabeli nie może wystąpić więcej niż jeden rekord z takim samym kluczem podstawowym. Zazwyczaj wartości klucza podstawowego rozpoczynają się od 1 dla pierwszego rekordu i ustawiane są rosnąco (2, 3, 4...) dla kolejnych wierszy.

Po co nam klucz podstawowy? Rzadko kiedy zdarza się, że baza danych składa się tylko z jednej tabeli. Nie bez powodu też ten rodzaj baz danych nazywa się **relacyjnym**. Wyobraź sobie teraz, że rozszerzasz możliwości swojej interaktywnej książki telefonicznej – do każdego kontaktu można dodać adres zamieszkania składających się z pól takich jak `ulica`, `miasto` oraz `kod_pocztowy`. Te wymaganie można postrzegać jako relację pomiędzy konkretnym kontaktem a jej adresem zamieszkania. Klucze podstawowe pozwalają na tworzenie relacji pomiędzy rekordami tej samej (rzadziej) lub różnych tabel.

id	ulica	miasto	kod_pocztowy	kontakt_id
1	Armii Krajowej	Gdańsk	80-871	1
2	Niepodległości	Sopot	80-452	2
3	Czołgistów	Gdynia	80-255	3
4	Wyzwolenia	Gdynia	81-978	4

Przykładowa tabela adresy z kluczem obcym do tabeli `kontakty` (relacja jeden-do-jednego).

Opracowanie własne.

Ostatnia kolumna powyższej tabeli wiąże konkretny adres zamieszkania z konkretnym kontaktem. Taki klucz wykorzystywany do powiązywania rekordów z jednej tabeli z rekordami z innej tabeli nazywany jest **kluczem obcym**.

Móglbyś(-abyś) teraz zapytać, co w przypadku, kiedy jeden kontakt ma więcej niż jeden adres zamieszkania. Nic trudnego, relacyjne bazy danych wspierają relacje różnego typu. Przedstawiona wyżej, gdzie jeden rekord z jednej tabeli posiada relacje tylko z jednym rekordem innej tabeli to relacja jeden-do-jednego (ang. one-to-one). Dodatkowo rozróżnia się:

- relacje jeden-do-wielu (ang. *one-to-many*): np. jeden kontakt posiadający kilka adresów zamieszkania lub jeden czytelnik, który wypożyczył z biblioteki kilka książek.
- relacje wiele-do-wielu (ang. *many-to-many*): np. student posiada wiele wykładów, a wykładowiec wiele studentów. Relacje tego typu stosuje się rzadko, gdyż praca z tabelami połączonymi taką relacją jest trudna.

## WAŻNE

Analityku, kojarzy Ci się tu pewnie diagram encji ERD. To właśnie diagram przedstawiający tabele w bazie danych oraz relacje między nimi. Pamiętaj, żeby nie mylić go z diagramem klas UML, który nie przedstawia tabel w bazie danych, ale poziom abstrakcji wyżej. Logiczna organizacja informacji może nieco różnić się od tego, jak odwzoruje ją projektant bazy danych. Ważne, żeby spełnienia postawione przed nią zdania. W diagramie klas jednak również masz relacje 1-1, 1-\* i na podobnej zasadzie działają relacje w bazie danych – mówią ile elementów jednego rodzaju może być powiązanych z elementami innego rodzaju

Id	ulica	miasto	kod_pocztowy	kontakt_Id
1	Armii Krajowej	Gdańsk	80-871	1
2	Niepodległości	Sopot	80-452	1
3	Czołgistów	Gdynia	80-255	1

Przykładowa tabela adresy z kluczem obcym do tabeli kontakty (relacja jeden-do-wielu).

Opracowanie własne.

Powyższy przykład pokazuje przykład relacji jeden-do-wielu. Zauważ, że w przeciwieństwie do klucza podstawowego, klucz obcy nie musi być unikalny (może wystąpić wiele razy wpisany do danego atrybutu krotek) – inaczej nie można by

przedstawić relacji jeden-do-wielu (wiele adresów z relacją do tego samego kontaktu).

Wiesz już czym charakteryzują się oraz na co pozwalają relacyjne bazy danych. Nie możemy jednak zakończyć dzisiejszej lekcji bez wyjaśnienia czym są kwerendy oraz bez rozwinięcia skrótu SQL.

Za pomocą kwerend (inaczej zapytań) możliwa jest komunikacja z bazą danych. Zarówno aplikacje jak i np. programiści "rozmawiają" z bazą danych za pomocą kwerend. Jeżeli Twoja aplikacja interaktywnej książki telefonicznej miałaby wyświetlić listę wszystkich kontaktów wysłałaby ona zapytanie do bazy danych, które mogłoby wyglądać np tak:

### PRZYKŁADOWE KWERENDY SQL

```
SELECT * FROM kontakty;  
SELECT COUNT(*) FROM kontakty;  
SELECT ulica, miasto FROM adresy;
```

Tłumacząc pierwszą z powyższych kwerend, napisaną w języku SQL, na język polski brzmiałaby ona mniej więcej tak: *Zwróć mi wszystkie dane wszystkich rekordów z tabeli kontakty.*

### DEFINICJA

**SQL** (ang. *Structured Query Language*) – strukturalny język zapytań używany do tworzenia, modyfikowania baz danych oraz do umieszczania i pobierania danych z baz danych.

(<https://pl.wikipedia.org/wiki/SQL>)

Język ten stał się standardem w komunikacji z relacyjnymi bazami danych, a możliwości samego języka są ogromne, ale to temat wykraczający poza zakres tego kursu. Jeżeli chcesz poznać podstawy języka SQL polecam jeden z wielu dostępnych kursów online: <http://www.sqlpedia.pl/kurs-sql/>.

### WAŻNE

Istotnym elementem w przypadku baz danych są kopie zapasowe. Utrzymywanie co najmniej jednej kopii zapasowej produkcyjnej bazy danych jest

obowiązkowym elementem wdrażania każdego nowego systemu polegającym na bazie danych. Kopie zapasowe pozwalają przywrócić sprawność systemu w przypadku awarii produkcyjnej bazy danych poprzez "podłączenie" w jej miejsce kopii zapasowej.

Gdzie wpisuje się te zapytania SQL? Programiści wpisują je w kodzie aplikacji. Jednak bazę można też odpytywać o dane w inny sposób, np. poprzez narzędzie SQL Developer. Posiada ono okno do wpisywania kodu SQL, opcja jego wykonania i przeglądania wyników.

Po co Tobie dostęp do bazy danych? To genialne narzędzie do sprawdzania danych, które przetwarza aplikacja. Jeśli np. chcesz sprawdzić ilu klientów mieszka w Poznaniu, albo jaki ma status wybrane zamówienie, wystarczy, że poznasz strukturę tabel w bazie, a będziesz mógł samodzielnie napisać zapytanie do bazy i zobaczyć wyniki. Przydaje się to wtedy, kiedy aplikacja nie wyświetla danych, których potrzebujesz, a chcesz coś sprawdzić. Przydaje się też do sprawdzania czy coś jest błędem, np. w bazie mamy poprawną wartość, ale aplikacja wyświetla ją niewłaściwie. Baza to też Twoje niepodważalne źródło informacji o tym jak aplikacja jest wykorzystywana i jak wygląda biznes, np. ile zamówień w miesiącu się pojawia, który sprzedawca generuje największy obrót, który produkt jest najczęściej kupowany.

## WAŻNE

Źle zaprojektowana baza danych może być dla aplikacji prawdziwą udręką, w szczególności, jeśli aplikacja przechowuje ogromne ilości danych. Dlatego pamiętaj, aby zlecić to lub skonsultować z ekspertem od baz danych, architektem albo dobrym programistą.

Ważne, żeby przekazać ile danych poszczególnych rodzajów spowiedziewamy się przechowywać, np. ile zamówień miesięcznie, ile klientów. Nie chodzi o dokładną wartość, ale o rząd wielkości i informację jak szybko dane te będą przyrastać. Pozwoli to zespołowi na zaprojektowanie bazy danych, która będzie prawidłowo przygotowana na takie obciążenie.

## RÓWNIEŻ WAŻNE :-)

Żeby dostać się do bazy danych, potrzebujesz od kolegów z zespołu tzw. dostępu (loginu, hasła i adresu, pod którym znajdziesz bazę). Możesz poprosić kogoś na początku, aby skonfigurował Ci połączenie do bazy. Popatrz jak to robi, będzie to dla Ciebie ciekawe doświadczenie. Może zapytać Cię o program do

przeglądania baz, ewentualnie poprosić, żebyś go pobrał. Następnie wykorzystasz właśnie te dane dostępne, żeby nawiązać połączenie. Potem pokaże Ci, gdzie możesz zobaczyć strukturę tabel w bazie, wszystkie dane, a gdzie możesz wpisywać kod SQL.

Jeżeli aplikacja, nad którą pracujesz ma więcej niż jedną wersję czy więcej niż jedno środowisko (różne adresy, które wpisujesz w przeglądarkę, żeby przejść do aplikacji), wiedz, że każda z nich może mieć swoją własną bazę danych. **Jeśli chcesz coś sprawdzić, to upewnij się, że łączysz się z odpowiednią bazą dla odpowiedniej aplikacji.**

Podsumowując, zapamiętaj z dzisiejszej lekcji, że najczęściej wykorzystywanym obecnie rodzajem baz danych są relacyjnej bazy danych, których podstawowymi elementami są tabele i rekordy – wiersze tabel. Unikalny klucz podstawowy pozwala jednoznacznie wskazać konkretny rekord danej tabeli, a poprzez klucze obce definiuje się relacje pomiędzy rekordami różnych tabel. Kwerenda napisana w języku SQL pozwala na komunikowanie się z relacyjną bazą danych np. na pobieranie informacji o konkretnych rekordach, lub dodawanie nowego rekordu do istniejącej tabeli.

### ZADANIE PODSTAWOWE

Dopasuj zapytanie `SELECT * FROM kontakty;` tak aby zwróciło nie wszystkie dane rekordów z tabeli `kontakty`, ale tylko kolumny `imie` oraz `nazwisko`.

### ZADANIE DLA AMBITNYCH

Napisz kwerendę SQL, która dodałaby nowy rekord do tabeli `kontakty` z dzisiejszej lekcji.

Do usłyszenia!