

Dokumentacja

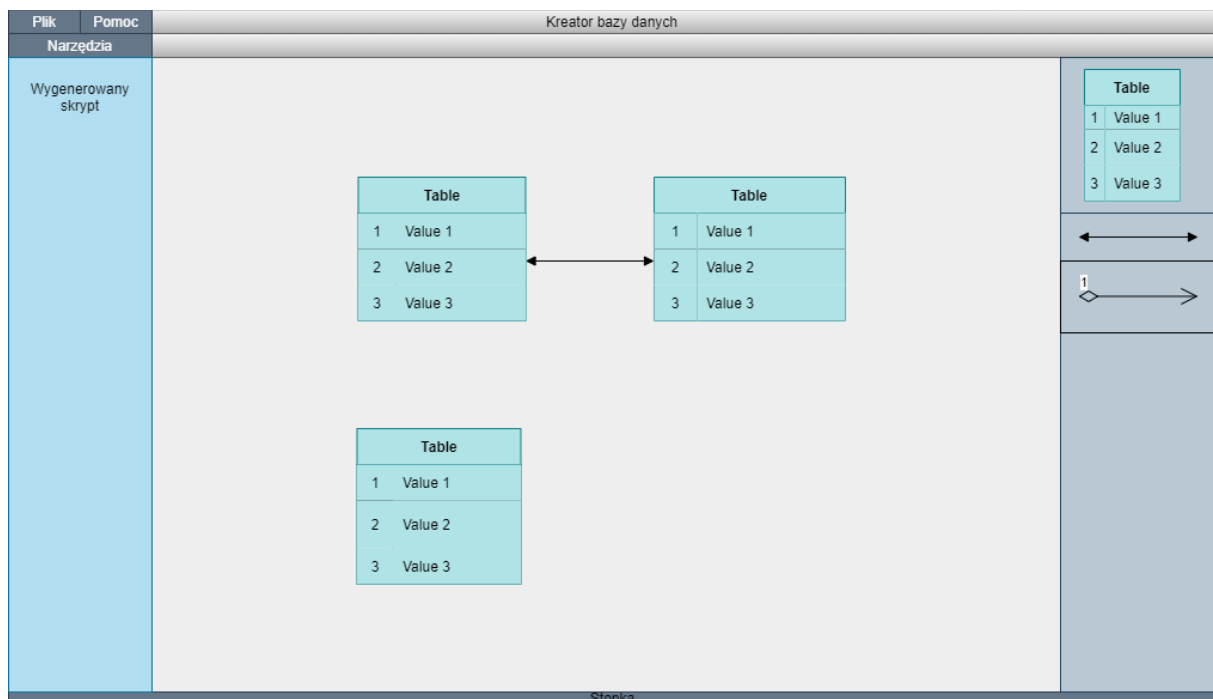
1. Temat projektu.

Tematem naszego projektu było „Modelowanie bazy danych”. Celem pracy jest opracowanie narzędzia graficznego do modelowania bazy danych. Za pomocą stworzonego przez nas programu, użytkownik ma możliwość projektowania bazy danych, a następnie wygenerowanie na podstawie stworzonego schematu graficznego, skryptu pozwalającego na stworzenie bazy danych.

2. Założenia w postaci harmonogramu oraz prototyp ekranu.

Spis funkcjonalności w kolejności chronologicznej:

- Dodawanie/usuwanie/modyfikacja tabel.
- Dodawanie/usuwanie/modyfikacja kolumn w tabelach.
- Ustawianie kluczy głównych i kluczy obcych w tabelach oraz wprowadzanie asocjacji o liczności jeden do jednego, jeden do wielu, wiele do wielu.
- Dodawanie/usuwanie/modyfikacja relacji pomiędzy tabelami.
- Zapis modelu bazy danych w pliku graficznym.
- Konwersja modelu graficznego na skrypt tworzący bazę danych.
- Możliwość wygenerowania schematu bazy danych na podstawie skryptu.



Rys. 2.1 Prototyp ekranu aplikacji.

Na rysunku 2.1 został przedstawiony prototyp ekranu tworzonej przez nas aplikacji.

3. Wykorzystywane technologie.

Do realizacji projektu zdecydowaliśmy się wykorzystać następujące technologie:

- JavaFx,
- Scene Builder.

JavaFx jest biblioteką, która umożliwia tworzenie aplikacji desktopowych w bardzo łatwy i przyjemny sposób, przy zachowaniu większej czytelności kodu. Wynika to głównie z tego, że posiada możliwość tworzenia wyglądu aplikacji w programie Scene Builder.

Biblioteka JavaFx umożliwia bardzo dobrą separację warstw aplikacji. Dzięki temu w łatwy sposób mogą pracować z kodem programiści, zajmujący się odrębnymi elementami. Ponadto technologia ta umożliwia tworzenie wyglądu aplikacji w języku XML. Jest to duże ułatwienie w przypadku późniejszego edytowania kodu czy szybkości tworzenia programu. W technologii, jaką jest JavaFx aplikacja składa się z obszaru roboczego, który definiuje przestrzeń oraz sceny, która określa co w danej przestrzeni będzie się znajdować. Scena jest często określana jako zbiornik na poszczególne elementy aplikacji. W przypadku obszaru roboczego, aplikacje mają dostęp do jednego, a dokładnie do głównego obszaru roboczego. Do niego właśnie są dodawane poszczególne sceny, które tworzy się poprzez „uzupełnianie” tych scen elementami takimi jak np. przyciski.

Narzędziem, które jest wykorzystywane do tworzenia widoku aplikacji jest Scene Builder. Jest on niezwykle wygodny, ponieważ pojawia się dzięki niemu możliwość tworzenia aplikacji za pomocą gotowych kontrolerek, przycisków, pól tekstowych itp. Natomiast po zapisaniu utworzonego widoku, program sam generuje kod XML i aktualizuje go w kodzie. Ponadto jest to wygenerowane w taki sposób, że z łatwością można go edytować z poziomu środowiska programistycznego.

4. Opis najważniejszych zagadnień wykorzystywanych w projekcie.

Relacje (związki) to zależności między tabelami umożliwiające ich logiczne powiązanie ze sobą. Typy relacji:

- **Jeden-do-jednego**, polega na tym, że jednemu rekordowi pierwszej tabeli jest przyporządkowany dokładnie jeden rekord drugiej tabeli, a jednemu rekordowi drugiej tabeli jest przyporządkowany dokładnie jeden rekord pierwszej tabeli, np. jeden przelew bankowy ma przyporządkowany jeden kod jednorazowy i odwrotnie;



Rys. 4.1 Źródło: https://fs.siteor.com/ecdl/files/RODZAJE_BAZ_DANYCH_I_ICH_BUDOWA.pdf?1289369360.

- **Jeden-do-wielu**, polega na tym, że jednemu rekordowi pierwszej tabeli jest przyporządkowanych wiele rekordów drugiej tabeli, a jednemu rekordowi drugiej tabeli jest przyporządkowany dokładnie jeden rekord pierwszej tabeli, np. jeden wychowawca ma wielu uczniów, ale uczeń ma jednego wychowawcę;

Jeden dostawca...

ID dostawcy	Nazwa firmy	Przedstawiciel
1	Exotic Liquids	Charlotte Cooper
2	New Orleans Cajun Delights	Shelly Burke
3	Grandma Kelly's Homestead	Regina Murphy
4	Tokyo Traders	Yoshi Nagase

...może dostarczyć wiele produktów.

ID produktu	Nazwa produktu	Stan magazynu	ID dostawcy
1	Chai	39	1
2	Chang	17	1
3	Aniseed Syrup	13	1
4	Chef Anton's Cajun Seasoning	53	2
5	Chef Anton's Gumbo Mix	0	2

Ale każdy produkt ma jednego dostawcę.

Rys. 4.2 Źródło: https://fs.siteor.com/ecdl/files/RODZAJE_BAZ_DANYCH_I_ICH_BUDOWA.pdf?1289369360.

- **Wiele-do-wielu**, polega na tym, że jednemu rekordowi pierwszej tabeli jest przyporządkowanych wiele rekordów drugiej tabeli, a jednemu rekordowi drugiej tabeli jest przyporządkowanych wiele rekordów pierwszej tabeli, np. jeden nauczyciel uczy wielu uczniów, a każdy uczeń ma wielu nauczycieli.

ID zamówienia	ID klienta	ID pracownika
10000	FRANS	6
10001	MEREP	8
10002	FOLKO	3

Klucz podstawowy z tabeli "Zamówienia"

Klucz podstawowy z tabeli "Produkty"

W jednym zamówieniu może być wiele produktów...

ID zamówienia	ID produktu	Cena jednostk
10001	25	14,00 zł
10001	40	12,80 zł
10001	59	38,50 zł
10002	25	14,00 zł

...i każdy produkt może się pojawić w wielu zamówieniach.

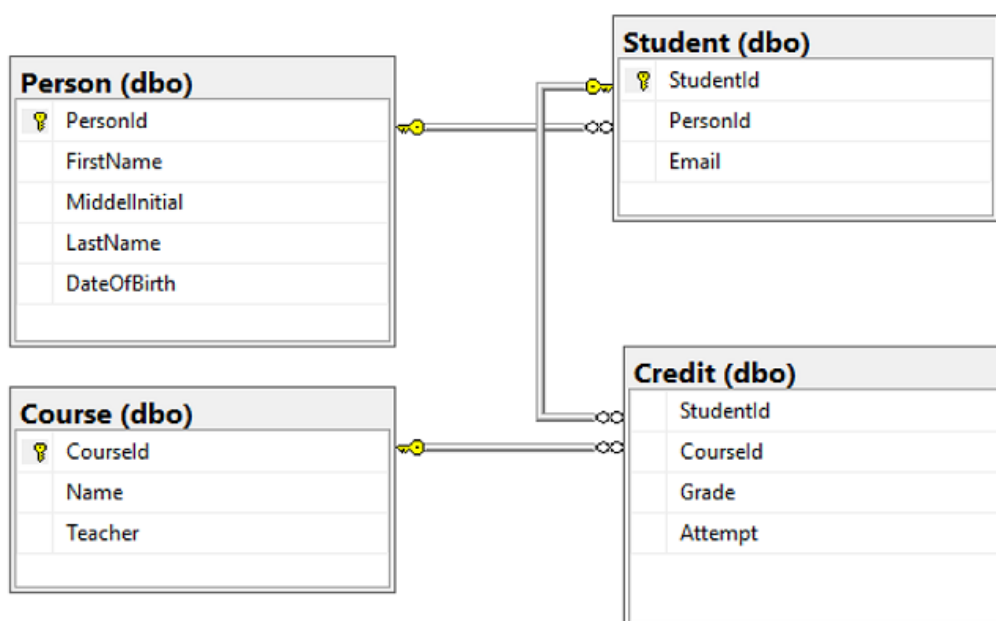
ID produktu	Nazwa produktu	Cena jednostkowa
25	NuNuCa Nuß-Nougat Creme	14,00 zł
26	Gumbär Gummibärchen	31,23 zł

Rys. 4.3 Źródło: https://fs.siteor.com/ecdl/files/RODZAJE_BAZ_DANYCH_I_ICH_BUDOWA.pdf?1289369360.

Model danych to zintegrowany zbiór zasad opisujących dane, relacje, powiązania (stosunki) pomiędzy danymi, dozwolone operacje i ograniczenia nakładane na dane i operacje. Model danych jest próbą reprezentacji świata realnego i występujących w nim obiektów, zdarzeń oraz związków zachodzących między nimi. Można go opisać jako konstrukcję składającą się z trzech komponentów:

- **części strukturalnej** – składającej się z reguł określających budowę bazy danych;
- **części manipulacyjnej** – określającej, które operacje (transakcje) aktualizacji, pobierania i zmiany struktury można wykonywać na danych;
- **części zawierającej reguły integralności** – gwarantującej stabilność działania systemu.

Jednym z modeli baz danych jest **relacyjna baza danych**. Reprezentuje ona wszystkie dane w bazie danych jako proste dwuwymiarowe tabele zwane relacjami. Każdy wiersz tabeli relacyjnej, zwany krotką, reprezentuje encję danych z kolumnami tabeli reprezentującymi atrybuty (pola). Dopuszczalne wartości dla tych atrybutów są nazywane domeną. Każdy wiersz w tabeli relacyjnej musi mieć unikalny klucz podstawowy, a także kilka kluczy pomocniczych, które odpowiadają kluczom podstawowym w innych tabelach. Jedną z zalet relacyjnego modelu bazy danych jest to, że zapewnia elastyczność, która pozwala łatwo dostosować zmiany do struktury bazy danych. Ułatwia wiele widoków tej samej bazy danych dla różnych użytkowników. Na rysunku 4.4 został przedstawiony przykład relacyjnej bazy danych.



Rys. 4.4 Przykład relacyjnej bazy danych.

Klucz główny jest atrybutem (kolumną) zapewniającym unikalność rekordu w obrębie tabeli, w której występuje. W relacyjnej bazie danych, każda relacja (tabela) musi zawierać klucz główny.

Klucz obcy jest kolumną, która nie musi być unikatowa w obrębie tabeli, w której występuje, ale jest unikatowa w tabeli, do której odnosi się ta kolumna. Oznacza to więc, że musi istnieć tabela, w której występuje kolumna będąca kluczem głównym, do której odnosi się kolumna będąca kluczem obcym w innej tabeli. Należy zauważyć, że nie zachodzi zależność odwrotna, tzn. w tabeli z kluczem głównym, do którego odnosi się klucz obcy z innej tabeli, nie musi istnieć kolumna, będąca kluczem obcym dla innej tabeli.

Dziedziczenie

Systemy relacyjnych baz danych nie obsługują dziedziczenia. Hibernate natomiast udostępnia kilka strategii, za pomocą których jesteśmy w stanie wykorzystać tę cechę:

1. **MappedSuperclass.**
Podstawowym pojęciem przy dziedziczeniu encji jest mapped superclass. Jest to klasa oznaczona adnotacją *@MappedSuperclass*. Oznaczenie nią encji spowoduje, że jej własności i relacje zostaną odzwierciedlone w encjach dziedziczących po tej klasie.
2. **Single table.**
Zgodnie z nazwą dziedziczenie jest realizowane w oparciu o jedną tabelę w bazie danych. Automatycznie implikuje to istnienie w niej kolumn, które będą w stanie przechować stan wszystkich encji, które realizują dziedziczenie w oparciu o tę metodę.
3. **Joined table.**
4. **Table per class.**