Aplikacja do zarządzania firmą ze sklepami z figurkami

Autorzy:

Bartłomiej Dec

Jakub Hapunik

Szymon Niewiński

Spis treści

1. Wstęp

Tematem projektu jest aplikacja służąca do zarządzania firmą obsługującą sklepy z figurkami i grami. Użytkownik-administrator korzystający z aplikacji może dowolnie przeglądać, zmieniać i usuwać dane związane ze sklepami, produktami, pracownikami, dostawcami i klientami.

2. Analiza wymagań systemu

2.1 Wymagania funkcjonalne

Użytkownik korzystający z aplikacji powinien mieć dostęp do listy oraz wyszukiwarki produktów. Wyszukiwarka ma umożliwiać filtrowanie produktów na podstawie wybranych opcji (np na podstawie ich typu). Ponadto użytkownik powinien móc edytować dane produktu, oraz dodawać i usuwać produkty z listy produktów w przypadku jeżeli zaszłaby taka potrzeba.

Użytkownik korzystający z aplikacji powinien mieć wgląd w listę klientów i dostęp do wyszukiwarki klientów. Wyszukiwarka ma umożliwiać filtrowanie klientów na podstawie wybranych opcji. Ponadto użytkownik powinien móc edytować dane, oraz dodawać i usuwać klientów z listy w przypadku gdy zajdzie taka potrzeba.

Użytkownik korzystający z aplikacji powinien mieć wgląd w listę pracowników. Przy pomocy listy powinien móc dowolnie usuwać, dodawać lub edytować dane pracowników wewnątrz systemu w przypadku zatrudniania, zwalniania lub zmiany danych osobowych pracownika. Wyszukiwarka pracowników będzie umożliwiać filtrowanie wypisywanych na ekranie pracowników na podstawie wybranych opcji (np po imieniu lub płci).

Użytkownik korzystający z aplikacji powinien mieć wgląd w listę umów. Przy pomocy listy powinien móc dowolnie usuwać, dodawać lub edytować dane umów konkretnych pracowników wewnątrz systemu. Wyszukiwarka będzie umożliwiać filtrowanie wypisywanych na ekranie umów na podstawie wybranych opcji (np po dacie rozpoczęcia umowy).

Użytkownik korzystający z aplikacji powinien mieć wgląd w listę sklepów i magazynów. Przy pomocy listy będzie mógł w razie potrzeby dowolnie usuwać, dodawać lub edytować dane sklepów i magazynów. Dzięki wyszukiwarce będzie mógł filtrować wyświetlane miejsca na podstawie wybranych opcji (np po mieście-lokalizacji lub po limicie pracowników)

Użytkownik korzystający z aplikacji powinien mieć wgląd w listę dostawców. W przypadku zajścia odpowiednich zmian, przy pomocy listy będzie mógł dowolnie usuwać, dodawać lub edytować dane konkretnych dostawców. Dzięki wyszukiwarce będzie mógł odnaleźć dostawców na podstawie m. in. daty rozpoczęcia współpracy.

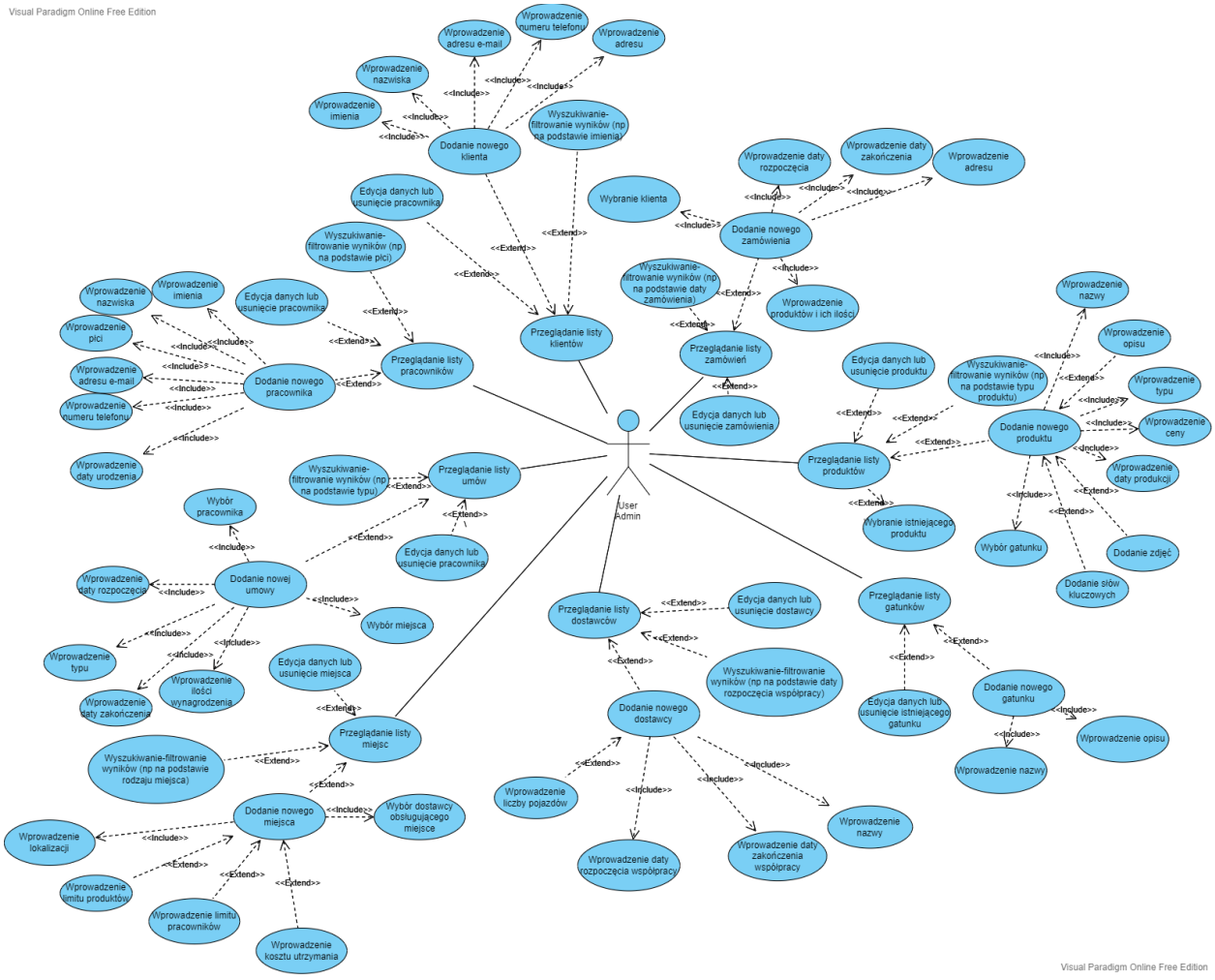
Użytkownik korzystający z aplikacji powinien mieć dostęp do spisu gatunków produktów. Przy jej pomocy będzie mógł usunąć lub edytować istniejące gatunki lub dodać nowe. Będą one służyły łatwiejszemu grupowaniu produktów wśród listy produktów.

Użytkownik korzystający z aplikacji powinien mieć dostęp do listy zamówień. Przy jej pomocy będzie mógł dowolnie przeglądać, dodawać, edytować i anulować zamówienia. W tym celu będzie wykorzystywał istniejące w systemie produkty oraz dane znajdujących się w systemie klientów.

2.2 Wymagania niefunkcjonalne

Na komputerach administratorów należy zainstalować aplikację i skonfigurować połączenie z bazą danych. Baza obsługuje do 10 połączeń jednocześnie.

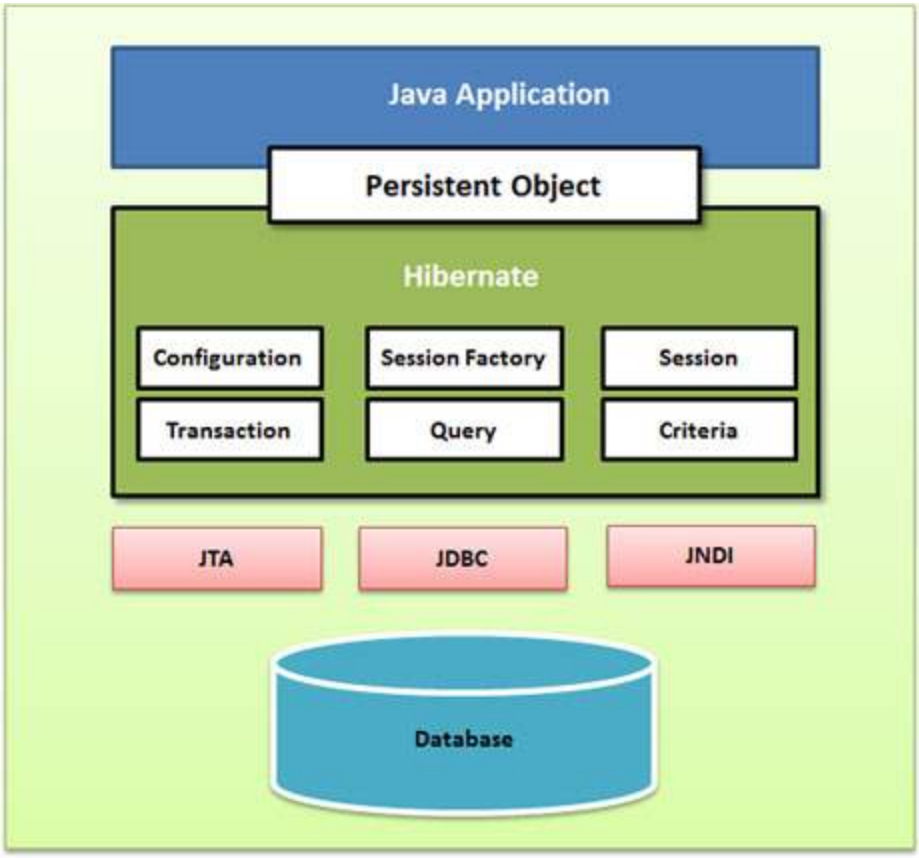
2.3 Diagram przypadków użycia



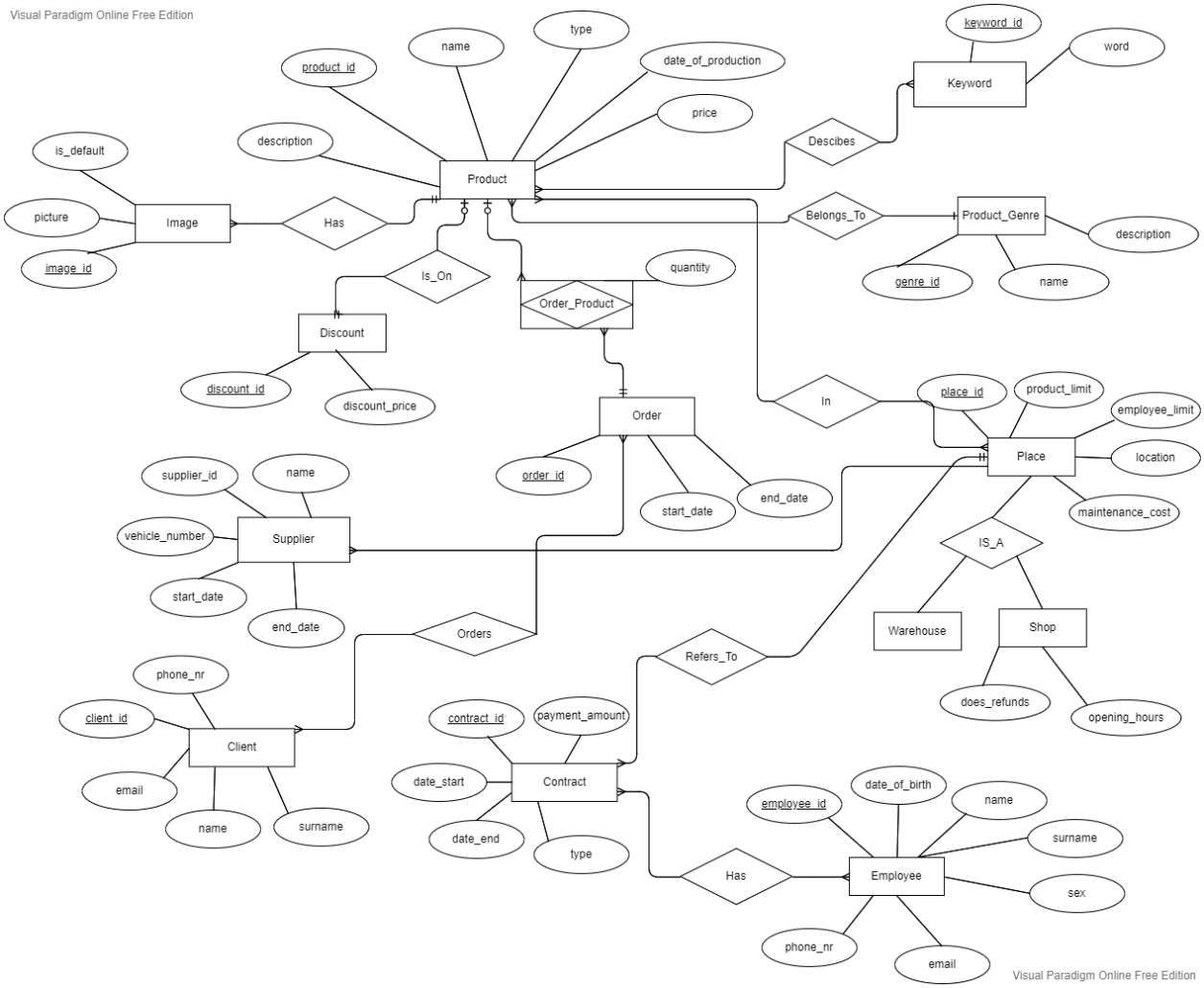
3. Wykorzystane technologie

- Hibernate
- Java Swing

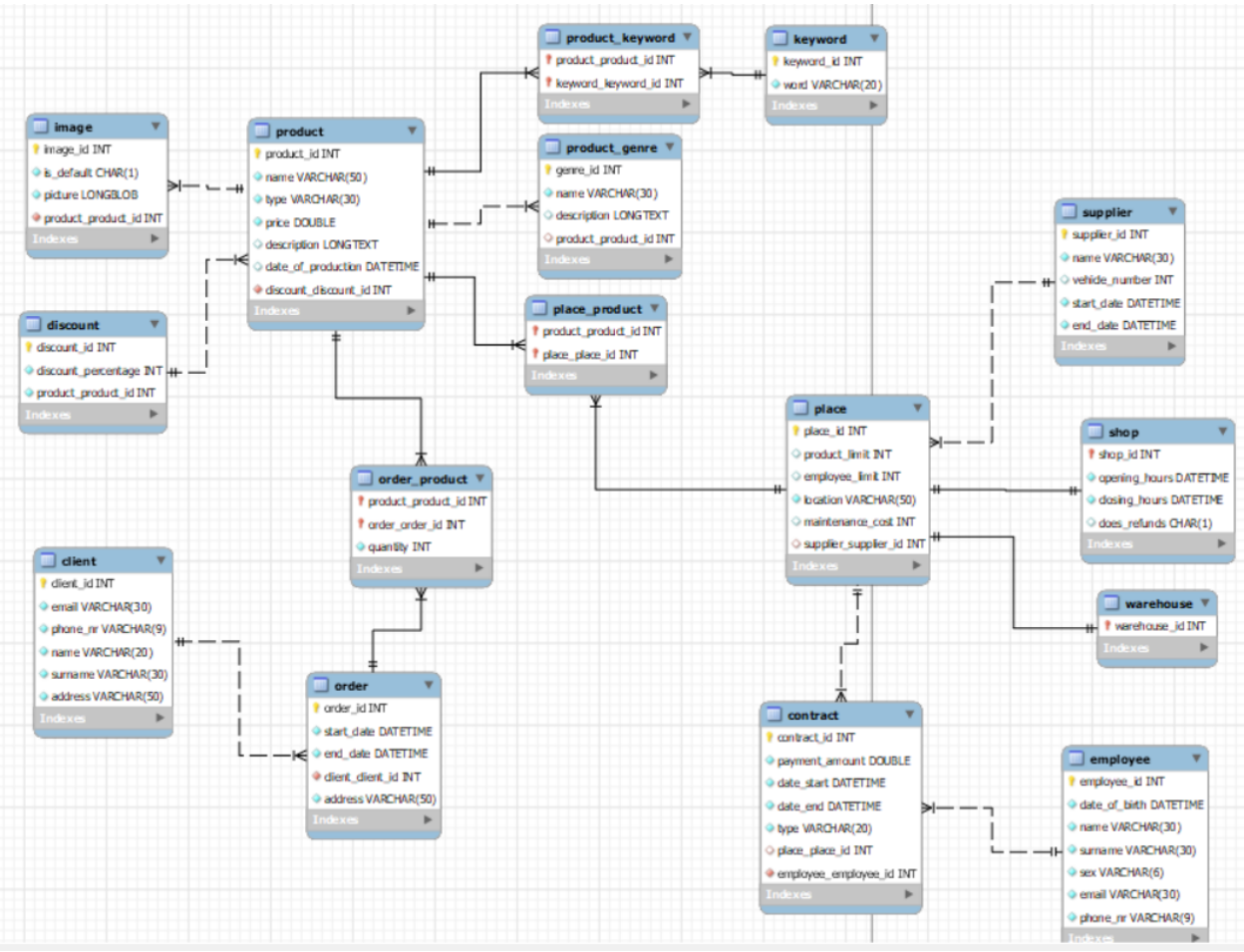
4. Projekt aplikacji



4.1 Projekt koncepcyjny bazy danych



4.2 Projekt schematu relacyjnego



4.3 Mapowanie klas na tabele bazodanowe

W załączonych plikach.

4.4 Opis poszczególnych tabel

- Tabela ,,Product” przechowuje informacje o produktach:
  - id - uniwersalny identyfikator
  - name - nazwa produktu
  - type - typ produktu (np figurka, gra, akcesorium, narzędzie)
  - date_of_production - data produkcji
  - price - cena produktu
  - description - krótki opis produktu
- Tabela ,,Supplier” opisuje dane związane z konkretnym dostawcą:
  - id - uniwersalny identyfikator

    - name - nazwa dostawcy
    - vehicles_number - ilość dostępnych pojazdów
    - start_date - data rozpoczęcia współpracy
    - end_date - data zakończenia współpracy
- Tabela ,,Order” zawiera dane zamówienia:
    - id - uniwersalny identyfikator
    - start_date - data zamówienia
    - end_date - przewidywany czas realizacji
- Tabela ,,Image” opisuje zdjęcia produktów:
    - id - uniwersalny identyfikator
    - is_default - określa, czy zdjęcie jest zdjęciem domyślnym
    - picture - określa plik zdjęcia
    - product_id - identyfikator produktu
- Tabela ,,Employee” opisuje konkretnych pracowników:
    - id - uniwersalny identyfikator
    - name - imię pracownika
    - surename - ilość czasu wymaganego przez przepis
    - date_of_birth - data urodzenia pracownika
    - sex - płeć pracownika
- Tabela ,,Contract” zawiera dane wszystkich umów:
    - id - uniwersalny identyfikator
    - employee_id - identyfikator pracownika
    - place_id - identyfikator miejsca pracy
    - payment_amount - ilość wynagrodzenia
    - date_start - data rozpoczęcia umowy
    - date_end - data zakończenia umowy
    - type - typ umowy
- Tabela Places zawiera dane sklepów/magazynów:
    - id - uniwersalny identyfikator
    - product_limit - limit ilości produktów
    - employee_limit - limit ilości pracowników
    - location - miasto, w którym sklep się znajduje
    - maintenance_cost - koszt utrzymania miejsca
- Tabela ,,Shops” zawiera dane sklepów:
    - id - uniwersalny identyfikator
    - opening_hours - godziny sprzedaży sklepu
    - closing_hours - godziny zamknięcia
    - does_refunds - czy jest możliwość otrzymania zwrotów lub wymiany produktów
- Tabela ,,Warehouses” zawiera dane magazynów:
    - id - id magazynu
- Tabela ,,Product_Genre” opisuje gatunek produktów:
    - id - uniwersalny identyfikator
    - name - nazwa gatunku

    - description - opis gatunku
- Tabela ,,Discount” opisuje zniżki konkretnych produktów:
    - id - uniwersalny identyfikator
    - product_id - identyfikator obniżanego produktu
    - discount_percentage - zniżka procentowa

5. Funkcjonalność aplikacji

1. Zarządzanie produktami:

- Użytkownik ma możliwość wyszukiwania, dodawania i usuwania produktów z listy,
- Użytkownik ma możliwość edycji danych produktów - dodawanie i usuwanie zdjęć, zmiana nazwy, kategorii, gatunku, ceny, słów kluczowych, daty produkcji i opisu przedmiotu.

2. Zarządzanie pracownikami:

- Użytkownik ma możliwość wyszukiwania, dodawania i usuwania pracowników z listy,
- Użytkownik ma możliwość edycji danych pracowników - zmiana imienia, nazwiska, adresu email, daty urodzenia oraz płci,
- Użytkownik ma możliwość wyszukiwania, dodawania i usuwania umów pracowników.

3. Zarządzanie sklepami i magazynami

- Użytkownik ma możliwość wyszukiwania, dodawania i usuwania placówek z listy sklepów i magazynów,
- Użytkownik ma możliwość edycji danych sklepów i magazynów - lokalizacji, godzin otwarcia, kosztów utrzymania, limitów pracowników i produktów
- Użytkownik ma możliwość wprowadzenia zniżki obowiązującej dla danego produktu w danym sklepie

4. Zarządzanie dostawcami i dostawami

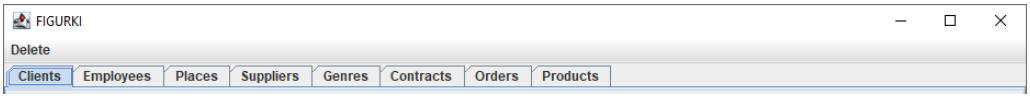
- Użytkownik ma możliwość wyszukiwania, dodawania i usuwania dostawców z listy.
- Użytkownik ma możliwość edycji danych dostawcy - zmiana nazwy, ilości pojazdów, daty rozpoczęcia i zakończenia współpracy
- Użytkownik ma możliwość wyszukiwania, dodawania i usuwania aktualnych dostaw z listy.
- Użytkownik ma możliwość edycji danych dostawy - zmiana przydzielonego dostawcy, ilości i rodzaju produktów, daty rozpoczęcia i zakończenia dostawy

5. Zarządzanie klientami

- Użytkownik ma możliwość wyszukania, dodawania i usuwania klientów z listy.
- Użytkownik ma możliwość edycji danych klientów - zmiany imienia, nazwiska, numeru telefonu, adresu i adresu e-mail klienta

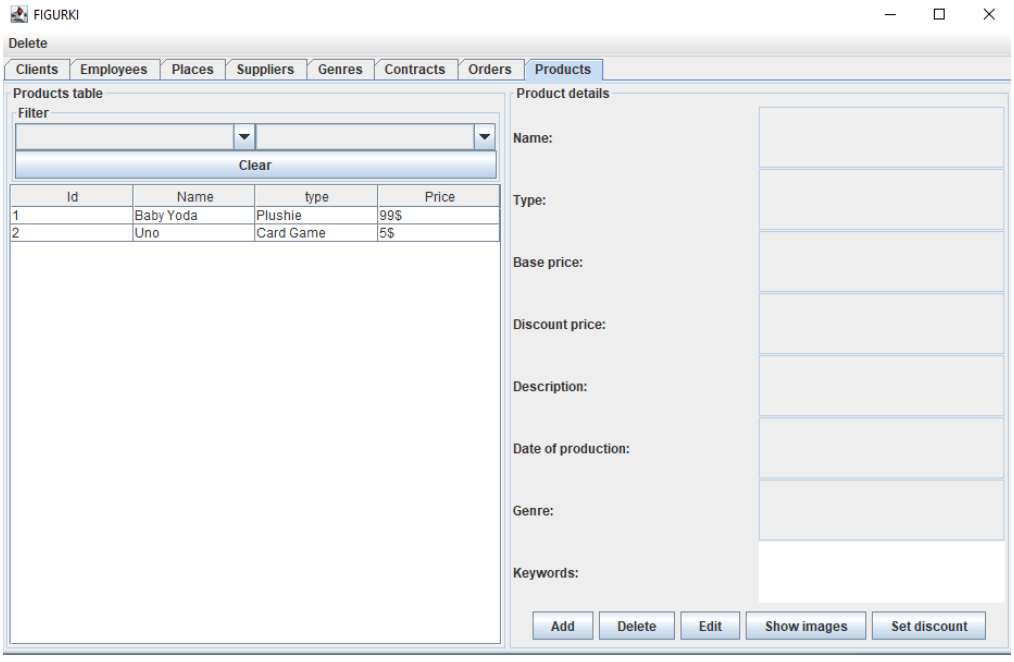
6. Interfejs użytkownika

- Pasek nawigacji - umożliwia nawigowanie po aplikacji. Znajduje się na nim 8 przycisków dostępnych z każdego widoku. Po wciśnięciu jednego z nich użytkownik jest odsyłany do odpowiedniego ekranu. Nad nim znajduje się menu Delete, gdzie można wybrać opcję usunięcia obrazka lub słowa kluczowego z bazy.

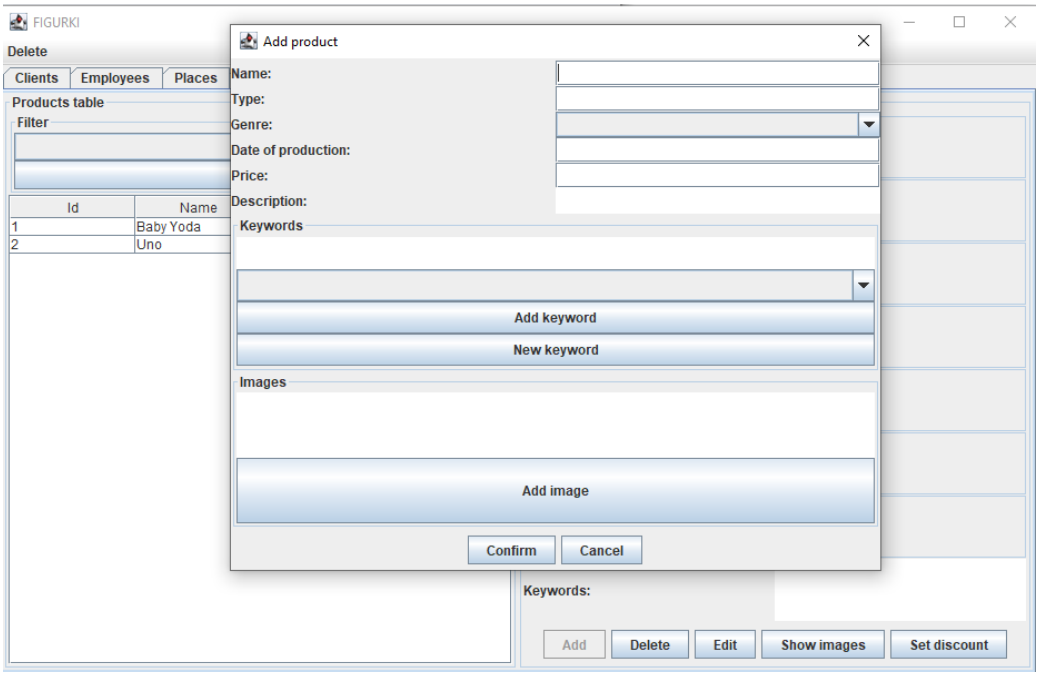


Pasek nawigacji po uruchomieniu programu

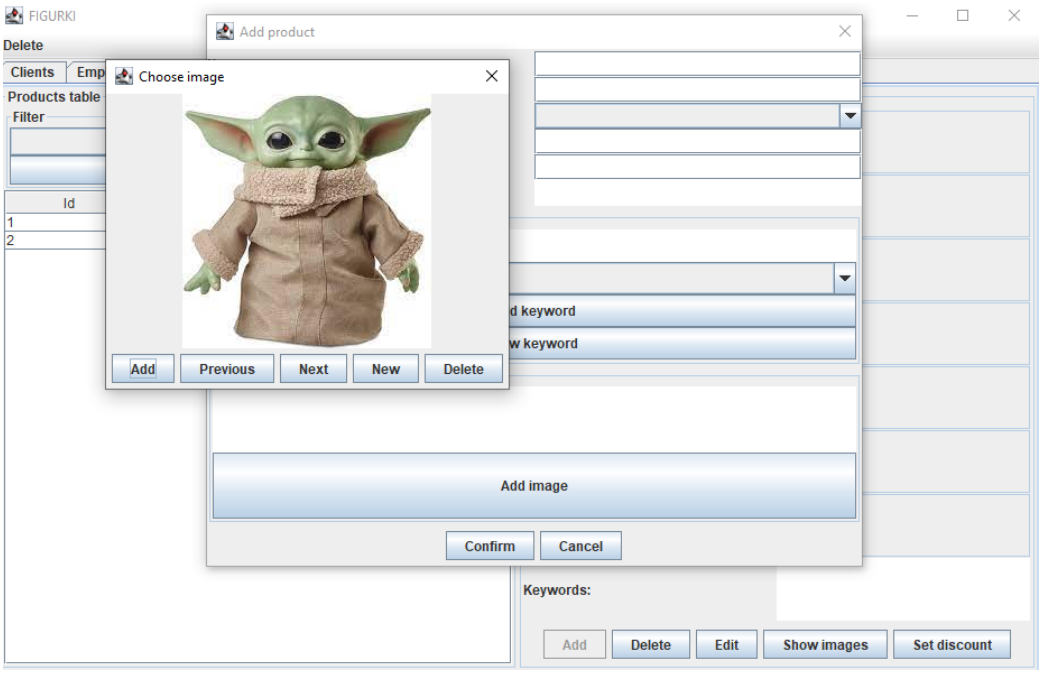
- Ekran zarządzania produktami - umożliwia wyszukiwanie, przeglądanie, dodawanie, usuwanie i edycję produktów. Po lewej stronie widnieje lista produktów. Nad listą znajduje się wyszukiwarka-filtr, umożliwiający wyświetlenie jedynie interesujących użytkownika produktów, np typu ,,Plushie" (pluszak). Niżej znajduje się przycisk wyłączający aktualny filtr. Po prawej stronie widnieją wszystkie informacje związane z produktem. Pod nimi znajduje się 5 przycisków - Add dodaje nowy produkt, Delete usuwa zaznaczony produkt, Edit pozwala na edycję aktualnie wybranego produktu, Show images wyświetla zdjęcia, Set discount ustanawia aktualną zniżkę ceny.



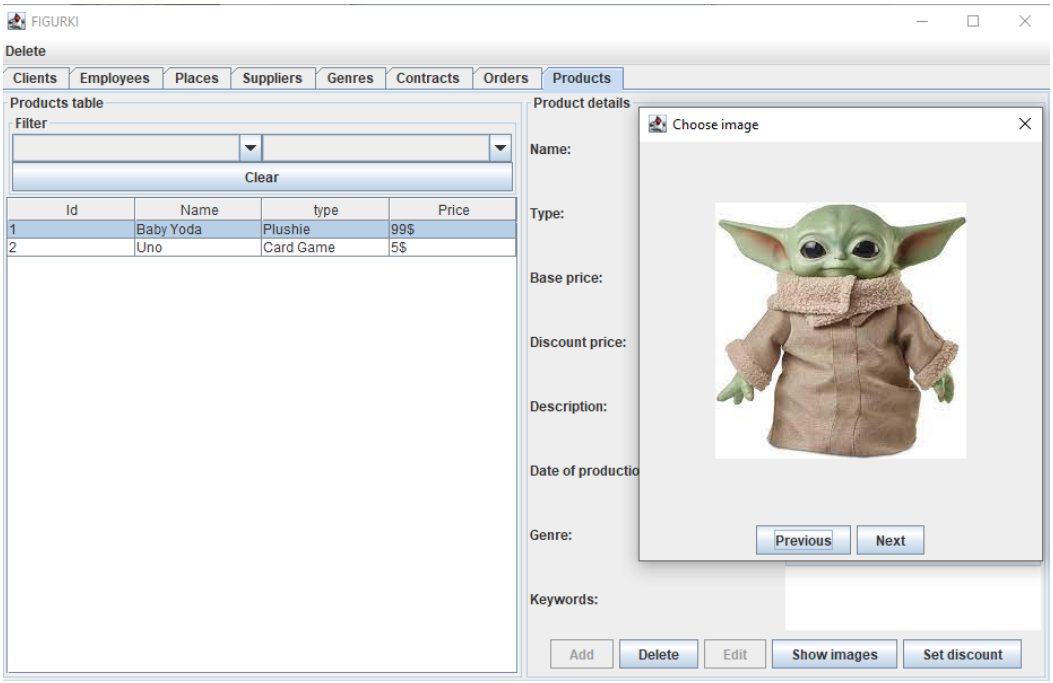
Ekran wyszukiwania produktów



Po wybraniu opcji ,,Add" wyskakuje okienko, które należy wypełnić informacjami o produkcie

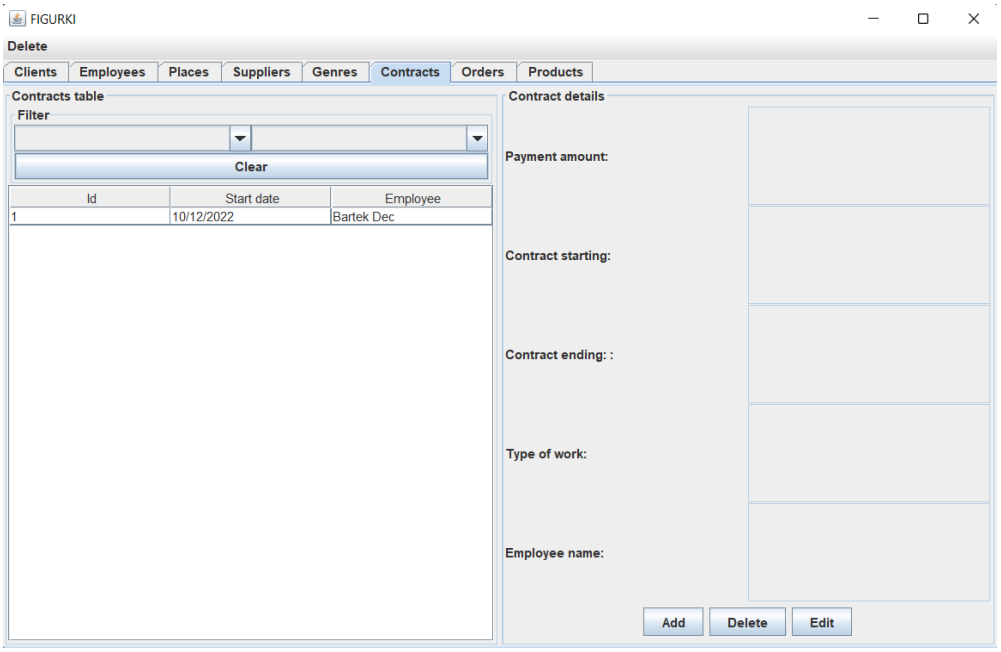


Po wybraniu opcji ,,Add image" aplikacja umożliwia wybór obrazka

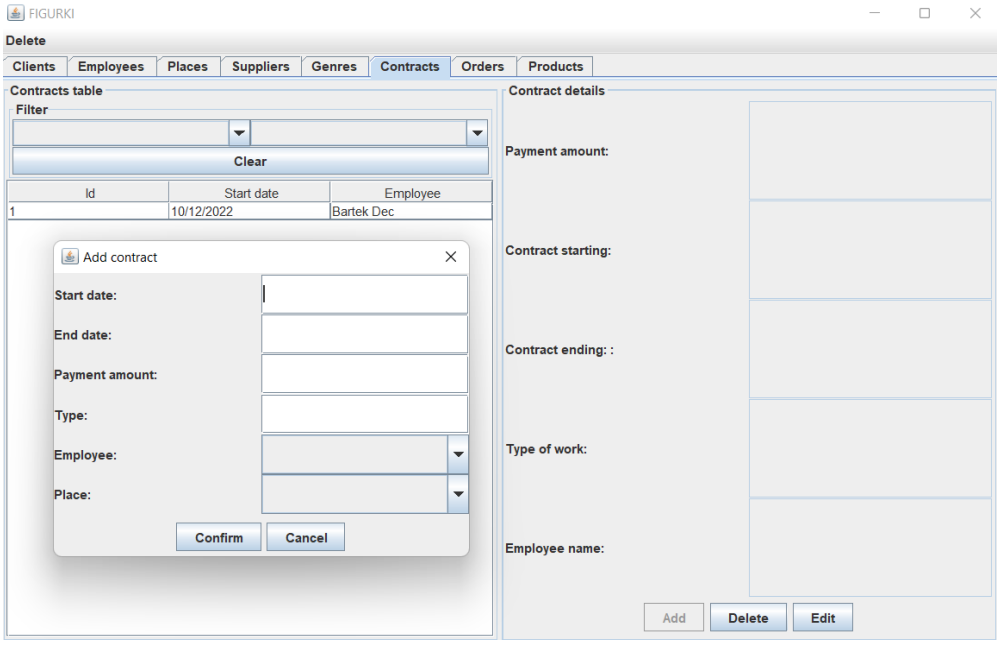


Po wybraniu konkretnego produktu i wciśnięciu przycisku „Show images" wyświetlany zostaje w nowym okienku wybrany do produktu obrazek

- Ekran zarządzania pracownikami - umożliwia wyszukiwanie, przeglądanie, dodawanie, usuwanie i edycję danych pracowników. Po lewej stronie widnieje lista pracowników. Nad listą znajduje się wyszukiwarka-filtr, umożliwiający wyświetlenie jedynie interesujących użytkownika pracowników, np mężczyzn lub kobiet. Niżej znajduje się przycisk wyłączający aktualny filtr. Po prawej stronie widnieją wszystkie informacje związane z aktualnie wybranym pracownikiem. Pod nimi znajdują się 3 przyciski - Add dodaje nowego pracownika, Delete usuwa zaznaczonego pracownika, Edit pozwala na edycję danych aktualnie wybranego pracownika



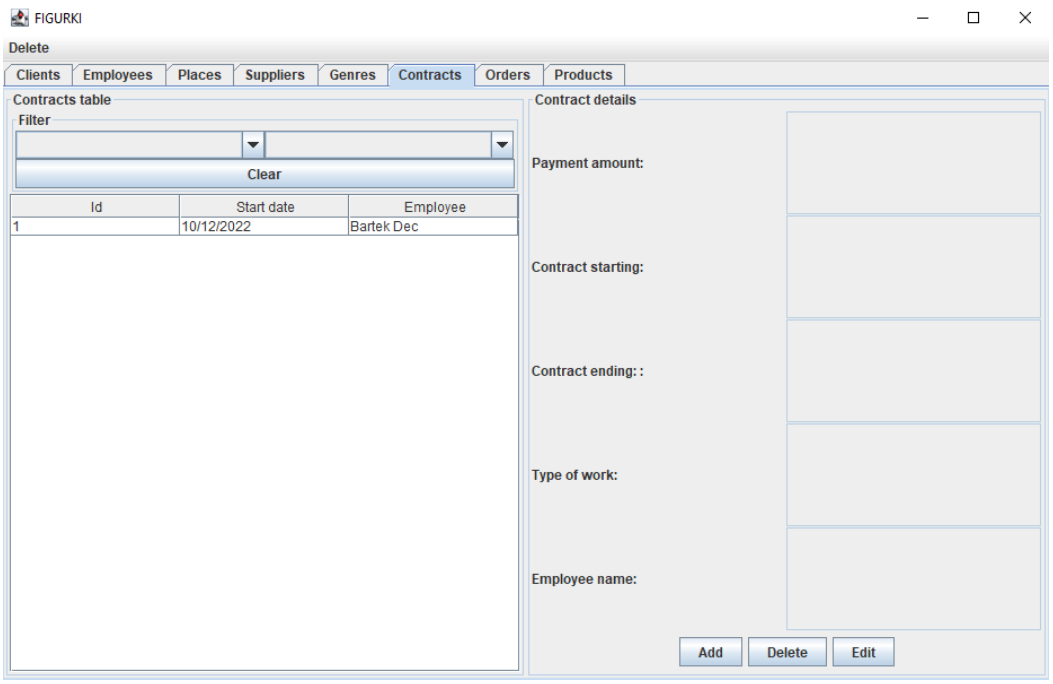
Ekran zarządzania pracownikami



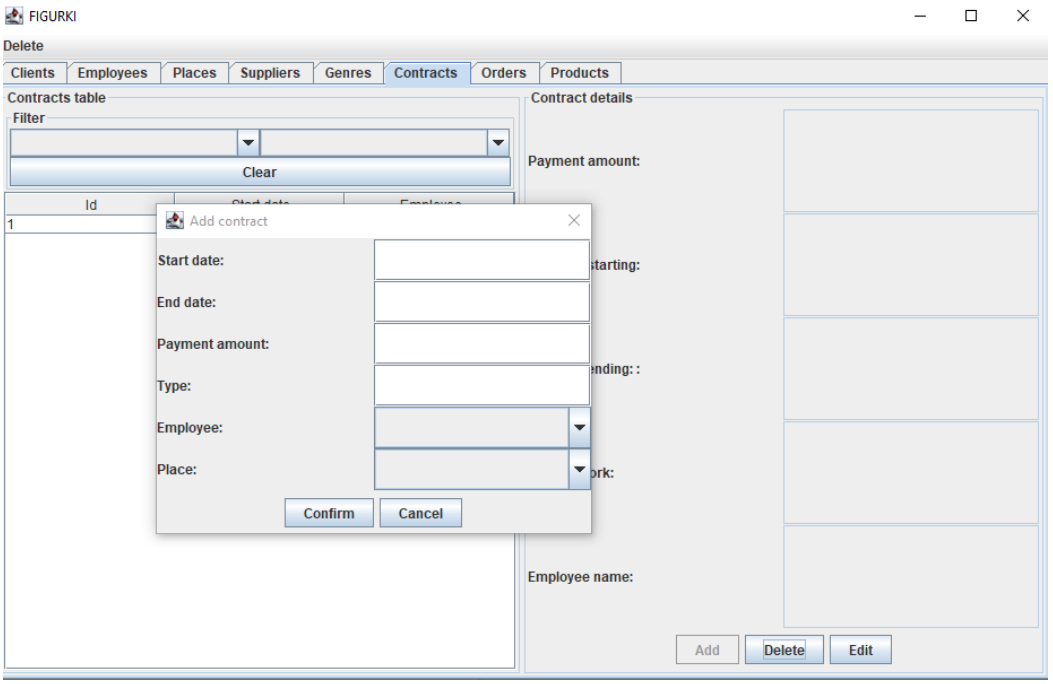
Po wybraniu opcji ,,Add" wyskakuje okienko, które należy wypełnić informacjami o pracowniku

- Ekran zarządzania umowami - umożliwia wyszukiwanie, przeglądanie, dodawanie, usuwanie i edycję danych umów. Po lewej stronie widnieje lista umów. Nad listą znajduje się wyszukiwarka-filtr, umożliwiający wyświetlenie jedynie interesujących użytkownika umów, np o konkretnej dacie podpisania. Niżej znajduje się przycisk wyłączający aktualny filtr. Po prawej stronie widnieją wszystkie informacje związane z

aktualnie wybraną umową. Pod nimi znajdują się 3 przyciski - Add dodaje nową umowę, Delete usuwa zaznaczoną umowę, Edit pozwala na edycję danych aktualnie wybranej umowy

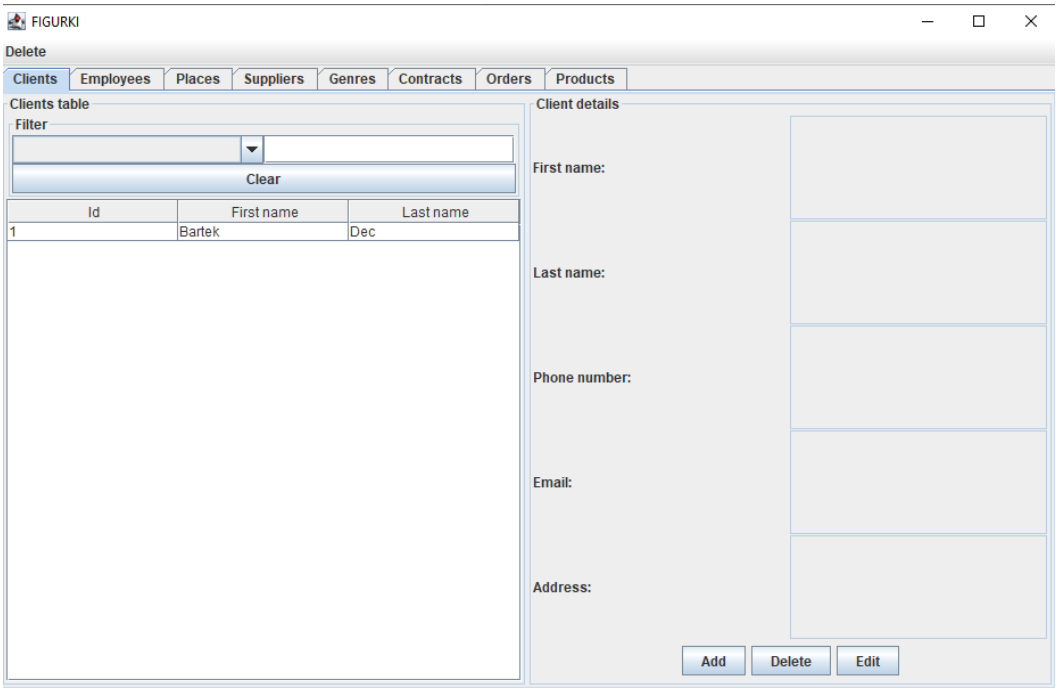


Ekran zarządzania umowami

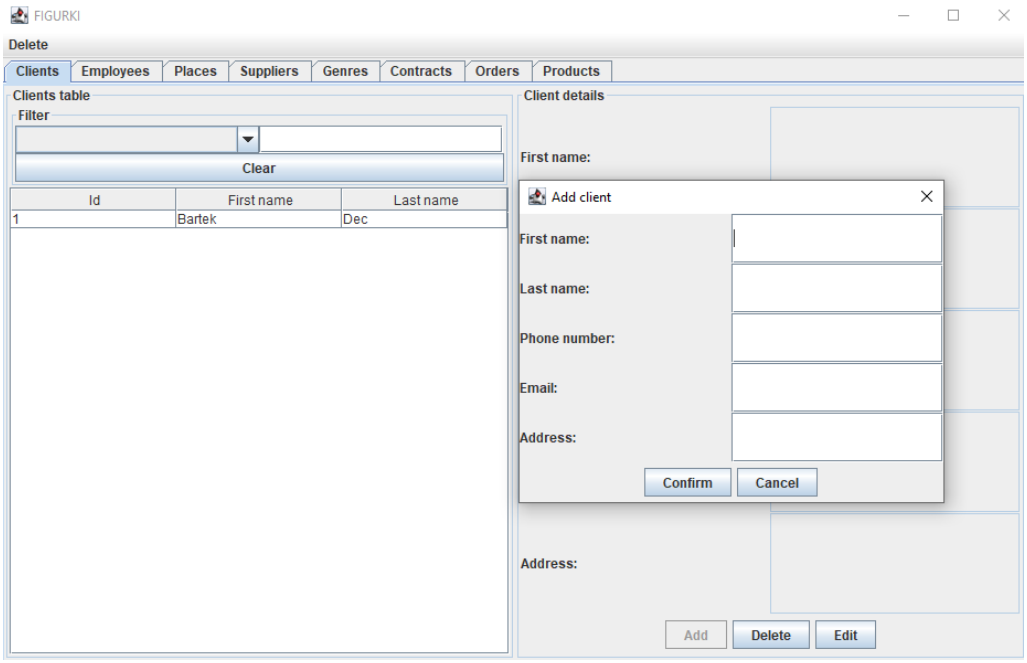


Po wybraniu opcji ,,Add" wyskakuje okienko, które należy wypełnić informacjami o umowie

- Ekran zarządzania klientami - umożliwia wyszukiwanie, przeglądanie, dodawanie, usuwanie i edycję danych klientów. Po lewej stronie widnieje lista klientów. Nad listą znajduje się wyszukiwarka-filtr, umożliwiający wyświetlenie jedynie interesujących użytkownika klientów, np o konkretnym imieniu. Niżej znajduje się przycisk wyłączający aktualny filtr. Po prawej stronie widnieją wszystkie informacje związane z aktualnie wybranym klientem. Pod nimi znajdują się 3 przyciski - Add dodaje nowego klienta, Delete usuwa zaznaczonego klienta, Edit pozwala na edycję danych aktualnie wybranego klienta.

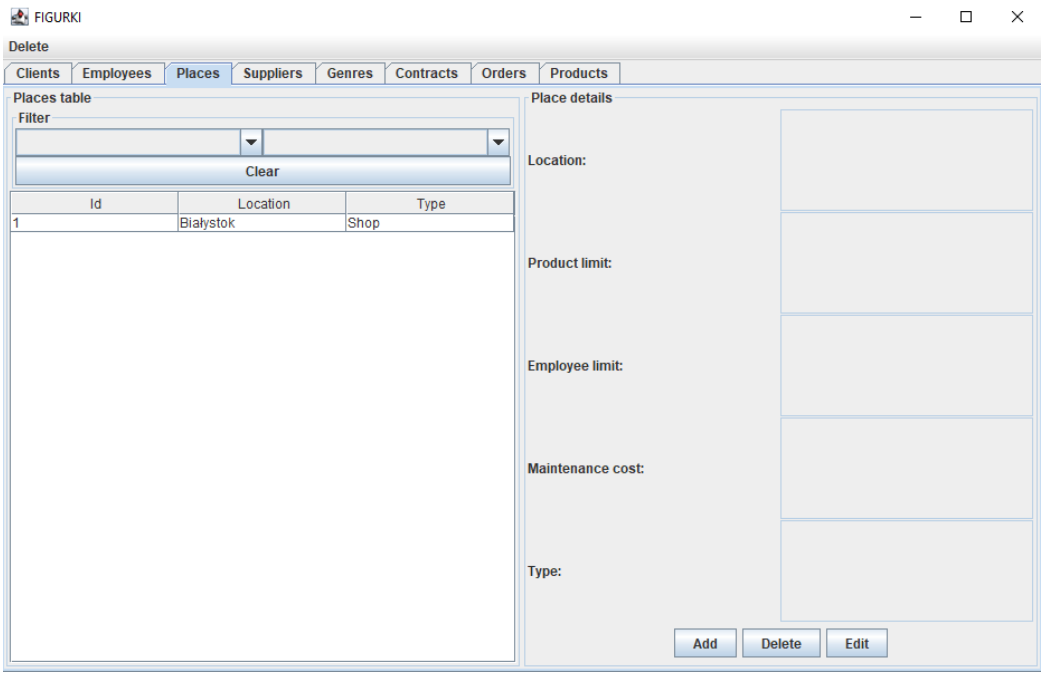


Ekran zarządzania klientami

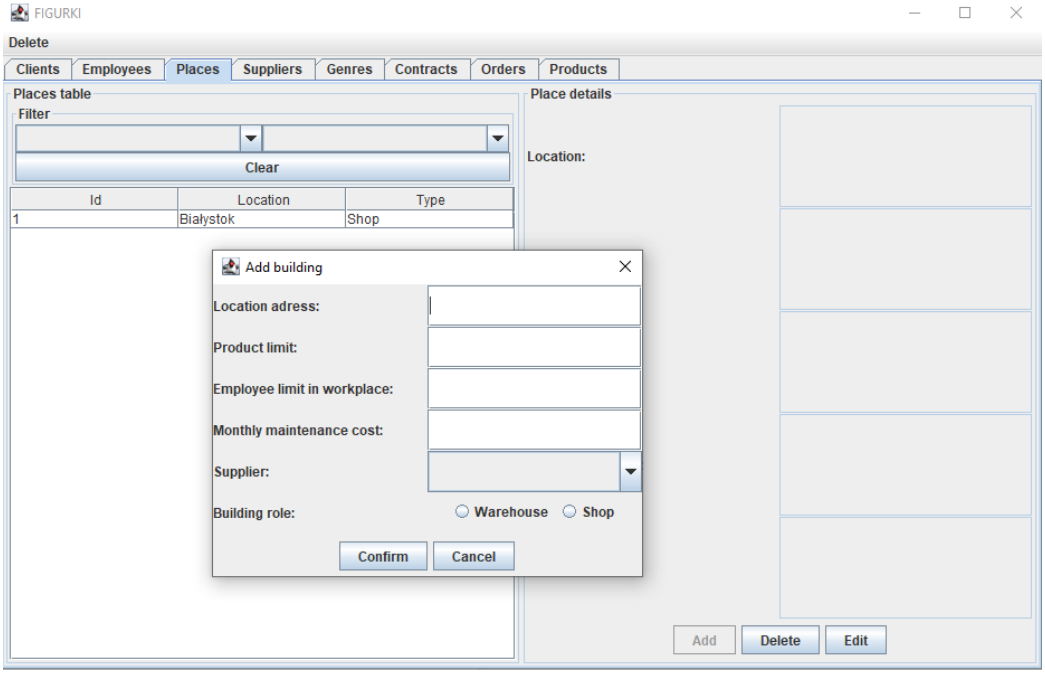


Po wybraniu opcji ,,Add" wyskakuje okienko, które należy wypełnić informacjami o kliencie

- Ekran zarządzania miejscami - umożliwia wyszukiwanie, przeglądanie, dodawanie, usuwanie i edycję danych miejsc (sklepów i magazynów). Po lewej stronie widnieje lista miejsc. Nad listą znajduje się wyszukiwarka-filtr, umożliwiający wyświetlenie jedynie interesujących użytkownika miejsc, np o typie ,,Shop" (sklep). Niżej znajduje się przycisk wyłączający aktualny filtr. Po prawej stronie widnieją wszystkie informacje związane z aktualnie wybranym miejscem. Pod nimi znajdują się 3 przyciski - Add dodaje nowe miejsce, Delete usuwa zaznaczone miejsce, Edit pozwala na edycję danych aktualnie wybranego miejsca.



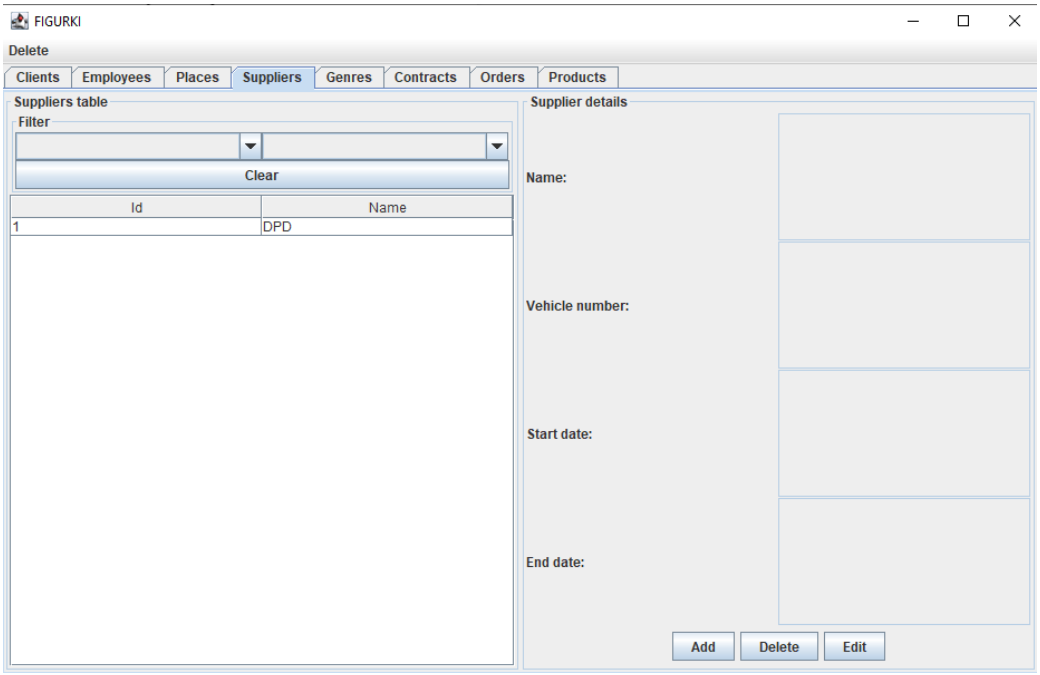
Ekran zarządzania miejscami



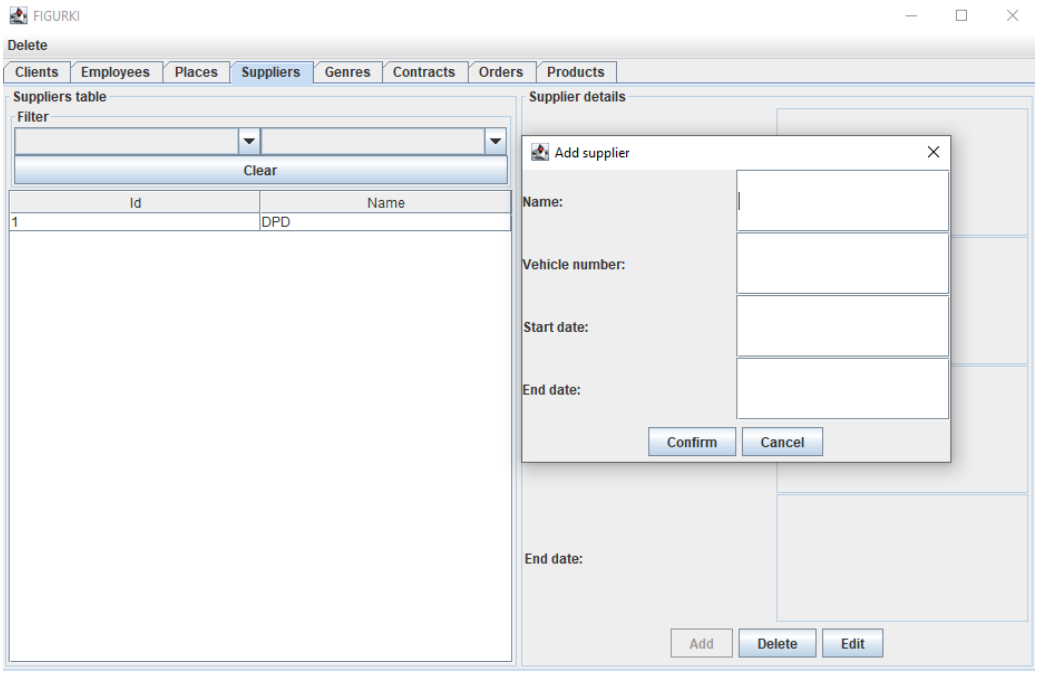
Po wybraniu opcji ,,Add" wyskakuje okienko, które należy wypełnić informacjami o miejscu

- Ekran zarządzania dostawcami - umożliwia wyszukiwanie, przeglądanie, dodawanie, usuwanie i edycję dostawców. Po lewej stronie widnieje lista dostawców. Nad listą znajduje się wyszukiwarka-filtr, umożliwiający wyświetlenie jedynie interesujących użytkownika dostawców, np na podstawie ilości dostępnych pojazdów. Niżej znajduje się przycisk wyłączający aktualny filtr. Po prawej stronie widnieją wszystkie informacje

związane z aktualnie wybranym dostawcą. Pod nimi znajdują się 3 przyciski - Add dodaje nowego dostawcę, Delete usuwa zaznaczonego dostawcę, Edit pozwala na edycję danych aktualnie wybranego dostawcy.

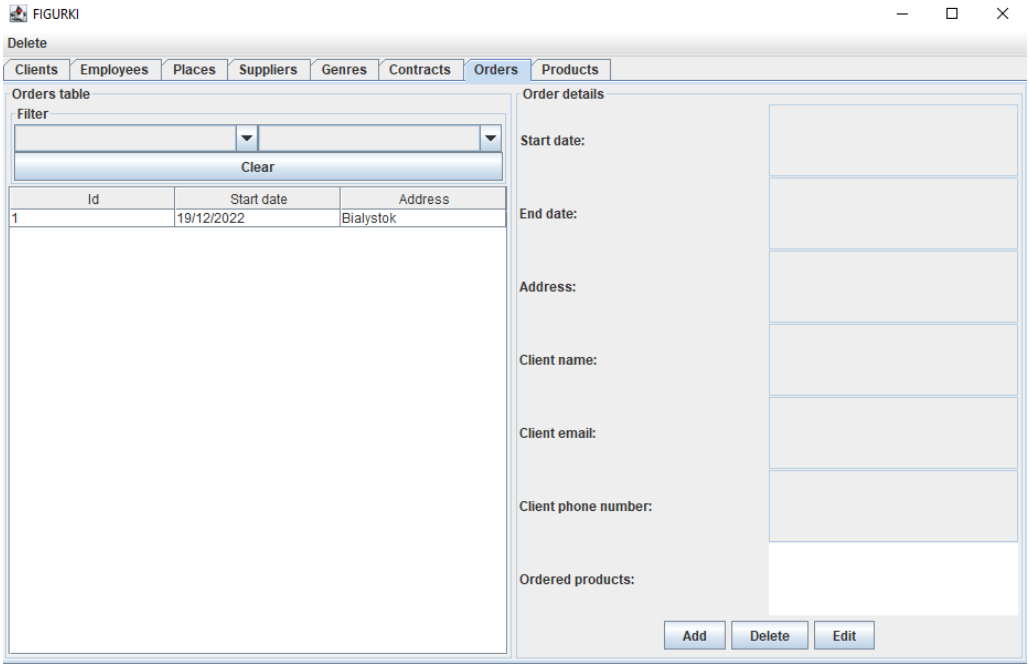


Ekran zarządzania dostawcami

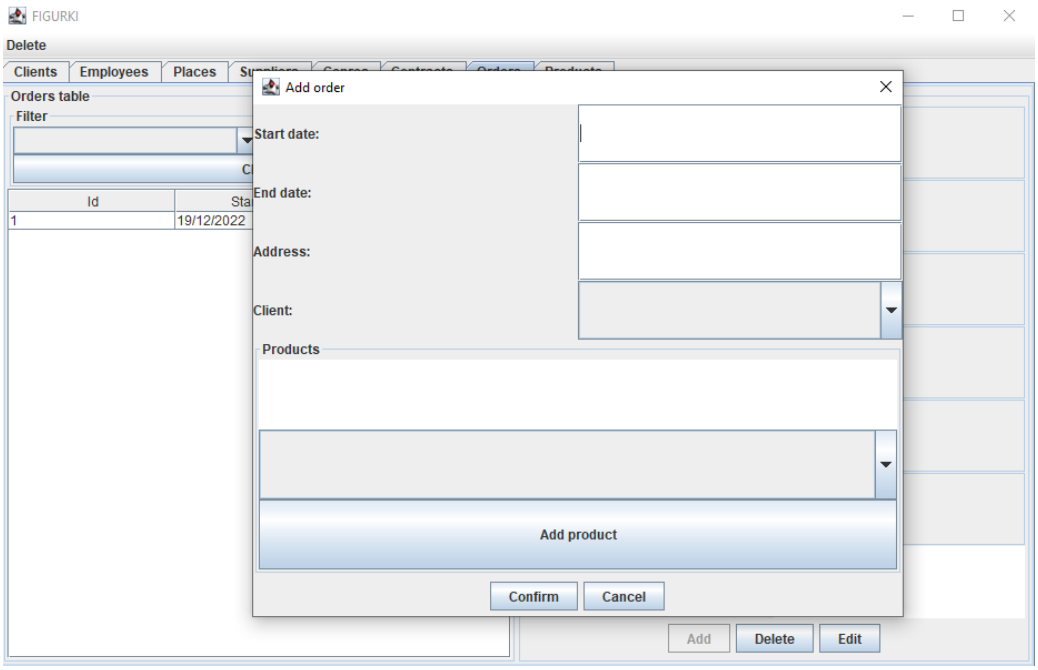


Po wybraniu opcji ,,Add" wyskakuje okienko, które należy wypełnić informacjami o dostawcy

- Ekran zarządzania zamówieniami - umożliwia wyszukiwanie, przeglądanie, dodawanie, usuwanie i edycję danych zamówień. Po lewej stronie widnieje lista zamówień. Nad listą znajduje się wyszukiwarka-filtr, umożliwiający wyświetlenie jedynie interesujących użytkownika zamówień, np o określonym adresie. Niżej znajduje się przycisk wyłączający aktualny filtr. Po prawej stronie widnieją wszystkie informacje związane z aktualnie wybranym zamówieniem. Pod nimi znajdują się 3 przyciski - Add dodaje nowe zamówienie, Delete usuwa zaznaczone zamówienie, Edit pozwala na edycję danych aktualnie wybranego zamówienia.



Ekran zarządzania zamówieniami



Po wybraniu opcji ,,Add" wyskakuje okienko, które należy wypełnić informacjami o zamówieniu

7. Podsumowanie

Aplikacja umożliwia proste zarządzanie sklepami, pracownikami, produktami i dostawami. Korzystanie z niej nie jest skomplikowane, wszak w dowolnym momencie widnieje 8 przycisków umożliwiających wybór dowolnego ekranu do zarządzania, a samo korzystanie z aplikacji jest intuicyjne. Wszechobecne wyszukiwania na podstawie konkretnych opcji umożliwiają szybkie i trafne przefiltrowanie oraz wyszukanie konkretnego rekordu.

Dodatek A: Skrypty tworzące obiekty baz danych

```
-- MySQL Workbench Forward Engineering


SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERR
OR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
-- -----------------------------------------------------
-- Schema mydb
-- -----------------------------------------------------
-- -----------------------------------------------------
-- Schema sbd
-- -----------------------------------------------------


-- -----------------------------------------------------
-- Schema sbd
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `sbd` DEFAULT CHARACTER SET utf8mb4 COLLATE
utf8mb4_0900_ai_ci ;
USE `sbd` ;


-- -----------------------------------------------------
-- Table `sbd`.`client`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `sbd`.`client` (
  `client_id` INT NOT NULL AUTO_INCREMENT,
  `email` VARCHAR(30) NOT NULL,
  `phone_nr` VARCHAR(9) NOT NULL,
  `name` VARCHAR(20) NOT NULL,
  `surname` VARCHAR(30) NOT NULL,
  `address` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`client_id`))
ENGINE = InnoDB
```

```
AUTO_INCREMENT = 3
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;



-- -----------------------------------------------------
-- Table `sbd`.`employee`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `sbd`.`employee` (
  `employee_id` INT NOT NULL AUTO_INCREMENT,
  `date_of_birth` DATETIME NOT NULL,
  `name` VARCHAR(30) NOT NULL,
  `surname` VARCHAR(30) NOT NULL,
  `sex` VARCHAR(6) NOT NULL,
  `email` VARCHAR(30) NOT NULL,
  `phone_nr` VARCHAR(9) NOT NULL,
  PRIMARY KEY (`employee_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;



-- -----------------------------------------------------
-- Table `sbd`.`supplier`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `sbd`.`supplier` (
```

```
  `supplier_id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(30) NOT NULL,
  `vehicle_number` INT NULL DEFAULT NULL,
  `start_date` DATETIME NOT NULL,
  `end_date` DATETIME NOT NULL,
  PRIMARY KEY (`supplier_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `sbd`.`place`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `sbd`.`place` (
  `place_id` INT NOT NULL AUTO_INCREMENT,
  `product_limit` INT NULL DEFAULT NULL,
  `employee_limit` INT NULL DEFAULT NULL,
  `location` VARCHAR(50) NOT NULL,
  `maintenance_cost` INT NULL DEFAULT NULL,
  `supplier_supplier_id` INT NULL DEFAULT NULL,
  PRIMARY KEY (`place_id`),
  INDEX `place_supplier_fk` (`supplier_supplier_id` ASC) VISIBLE,
  CONSTRAINT `place_supplier_fk`
    FOREIGN KEY (`supplier_supplier_id`)
    REFERENCES `sbd`.`supplier` (`supplier_id`))
```

```
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;



-- -----------------------------------------------------
-- Table `sbd`.`contract`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `sbd`.`contract` (
  `contract_id` INT NOT NULL AUTO_INCREMENT,
  `payment_amount` DOUBLE NOT NULL,
  `date_start` DATETIME NOT NULL,
  `date_end` DATETIME NOT NULL,
  `type` VARCHAR(20) NOT NULL,
  `place_place_id` INT NULL DEFAULT NULL,
  `employee_employee_id` INT NOT NULL,
  PRIMARY KEY (`contract_id`),
  INDEX `contract_employee_fk` (`employee_employee_id` ASC) VISIBLE,
  INDEX `contract_place_fk` (`place_place_id` ASC) VISIBLE,
  CONSTRAINT `contract_employee_fk`
    FOREIGN KEY (`employee_employee_id`)
    REFERENCES `sbd`.`employee` (`employee_id`),
  CONSTRAINT `contract_place_fk`
    FOREIGN KEY (`place_place_id`)
    REFERENCES `sbd`.`place` (`place_id`))
ENGINE = InnoDB
```

```
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;



-- -----------------------------------------------------
-- Table `sbd`.`discount`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `sbd`.`discount` (
  `discount_id` INT NOT NULL AUTO_INCREMENT,
  `discount_percentage` INT NOT NULL,
  `product_product_id` INT NOT NULL,
  PRIMARY KEY (`discount_id`),
  UNIQUE INDEX `discount__idx` (`product_product_id` ASC) VISIBLE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;



-- -----------------------------------------------------
-- Table `sbd`.`product`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `sbd`.`product` (
  `product_id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(50) NOT NULL,
  `type` VARCHAR(30) NOT NULL,
  `price` DOUBLE NOT NULL,
```

```
  `description` LONGTEXT NULL DEFAULT NULL,
  `date_of_production` DATETIME NULL DEFAULT NULL,
  `discount_discount_id` INT NOT NULL,
  PRIMARY KEY (`product_id`),
  UNIQUE INDEX `product__idx` (`discount_discount_id` ASC) VISIBLE,
  CONSTRAINT `product_discount_fk`
    FOREIGN KEY (`discount_discount_id`)
    REFERENCES `sbd`.`discount` (`discount_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;



-- -----------------------------------------------------
-- Table `sbd`.`image`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `sbd`.`image` (
  `image_id` INT NOT NULL AUTO_INCREMENT,
  `is_default` CHAR(1) NOT NULL,
  `picture` LONGBLOB NOT NULL,
  `product_product_id` INT NOT NULL,
  PRIMARY KEY (`image_id`),
  INDEX `image_product_fk` (`product_product_id` ASC) VISIBLE,
  CONSTRAINT `image_product_fk`
    FOREIGN KEY (`product_product_id`)
    REFERENCES `sbd`.`product` (`product_id`))
```

```
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `sbd`.`keyword`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `sbd`.`keyword` (
  `keyword_id` INT NOT NULL AUTO_INCREMENT,
  `word` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`keyword_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;


-- -----------------------------------------------------
-- Table `sbd`.`order`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `sbd`.`order` (
  `order_id` INT NOT NULL AUTO_INCREMENT,
  `start_date` DATETIME NOT NULL,
  `end_date` DATETIME NOT NULL,
  `client_client_id` INT NOT NULL,
  `address` VARCHAR(50) NOT NULL,
```

```
  PRIMARY KEY (`order_id`),
  INDEX `order_client_fk` (`client_client_id` ASC) VISIBLE,
  CONSTRAINT `order_client_fk`
    FOREIGN KEY (`client_client_id`)
    REFERENCES `sbd`.`client` (`client_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;



-- -----------------------------------------------------
-- Table `sbd`.`order_product`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `sbd`.`order_product` (
  `product_product_id` INT NOT NULL,
  `order_order_id` INT NOT NULL,
  `quantity` INT NOT NULL,
  PRIMARY KEY (`product_product_id`, `order_order_id`),
  INDEX `order_product_order_fk` (`order_order_id` ASC) VISIBLE,
  CONSTRAINT `order_product_order_fk`
    FOREIGN KEY (`order_order_id`)
    REFERENCES `sbd`.`order` (`order_id`),
  CONSTRAINT `order_product_product_fk`
    FOREIGN KEY (`product_product_id`)
    REFERENCES `sbd`.`product` (`product_id`))
ENGINE = InnoDB
```

```
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;



-- -----------------------------------------------------
-- Table `sbd`.`place_product`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `sbd`.`place_product` (
  `product_product_id` INT NOT NULL,
  `place_place_id` INT NOT NULL,
  PRIMARY KEY (`product_product_id`, `place_place_id`),
  INDEX `place_product_place_fk` (`place_place_id` ASC) VISIBLE,
  CONSTRAINT `place_product_place_fk`
    FOREIGN KEY (`place_place_id`)
    REFERENCES `sbd`.`place` (`place_id`),
  CONSTRAINT `place_product_product_fk`
    FOREIGN KEY (`product_product_id`)
    REFERENCES `sbd`.`product` (`product_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;



-- -----------------------------------------------------
-- Table `sbd`.`product_genre`
-- -----------------------------------------------------
```

```
CREATE TABLE IF NOT EXISTS `sbd`.`product_genre` (
  `genre_id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(30) NOT NULL,
  `description` LONGTEXT NULL DEFAULT NULL,
  `product_product_id` INT NULL DEFAULT NULL,
  PRIMARY KEY (`genre_id`),
  INDEX `product_genre_product_fk` (`product_product_id` ASC) VISIBLE,
  CONSTRAINT `product_genre_product_fk`
    FOREIGN KEY (`product_product_id`)
    REFERENCES `sbd`.`product` (`product_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;



-- -----------------------------------------------------
-- Table `sbd`.`product_keyword`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `sbd`.`product_keyword` (
  `product_product_id` INT NOT NULL,
  `keyword_keyword_id` INT NOT NULL,
  PRIMARY KEY (`product_product_id`, `keyword_keyword_id`),
  INDEX `product_keyword_keyword_fk` (`keyword_keyword_id` ASC) VISIBLE,
  CONSTRAINT `product_keyword_keyword_fk`
    FOREIGN KEY (`keyword_keyword_id`)
    REFERENCES `sbd`.`keyword` (`keyword_id`),
```

```
  CONSTRAINT `product_keyword_product_fk`
    FOREIGN KEY (`product_product_id`)
    REFERENCES `sbd`.`product` (`product_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;



-- -----------------------------------------------------
-- Table `sbd`.`shop`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `sbd`.`shop` (
  `shop_id` INT NOT NULL,
  `opening_hours` DATETIME NOT NULL,
  `closing_hours` DATETIME NOT NULL,
  `does_refunds` CHAR(1) NULL DEFAULT NULL,
  PRIMARY KEY (`shop_id`),
  CONSTRAINT `shop_place_fk`
    FOREIGN KEY (`shop_id`)
    REFERENCES `sbd`.`place` (`place_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;



-- -----------------------------------------------------
```

```
-- Table `sbd`.`warehouse`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `sbd`.`warehouse` (
  `warehouse_id` INT NOT NULL,
  PRIMARY KEY (`warehouse_id`),
  CONSTRAINT `warehouse_place_fk`
    FOREIGN KEY (`warehouse_id`)
    REFERENCES `sbd`.`place` (`place_id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;



SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```