Bazy Danych 1 - Sklep Budowlany

Wydział elektroniki	**Kierunek: informatyka techniczna**
Grupa zajęciowa: Pt 13:15	Semestr: 2020/2021 Lato
Prowadzący:	mgr inż. Szymon Wojciechowski

Autorzy
Byczko Maciej
Dębowski Jakub
Rzymyszkiewicz Tomasz

1. Spis treści

2. Wstęp

2.1. Opis systemu

Tematem zrealizowanego projektu jest system obsługi sieci sklepów budowlanych. Celem bazy danych jest zapewnienie podstawowych funkcjonalności potrzebnych do obsługi i zarządzania sieciową firmą budowlaną.

Nazwy tabel odzwierciedlają dane, które są w nich przechowywane dlatego nie ma potrzeby omawiania każdej z nich.

W systemie umieściliśmy niezbędne informacje z punktu widzenia różnych użytkowników bazy. Baza danych została zaprojektowana zgodnie z zasadami normalizacji.

System przechowuje następujące informacje:

- Pracownicy
    - Imię
    - Nazwisko
    - Płaca
    - Data zatrudnienia
    - Typ zatrudnienia
    - Identyfikator Przełożonego
    - Identyfikator sklepu, w którym pracuje
    - Zamówienia do których został przydzielony
- Produkty
    - Nazwa
    - Data produkcji
    - Cena kupna fabryczna
    - Cena kupna sklepowa
    - Identyfikator dostawcy
- Sklepy
    - Powierzchnia
    - Adres sklepu
        - Nazwa miasta
        - Ulica
        - Kod pocztowy
    - Identyfikator magazynu
- Klienci
    - Adres
        - Nazwa miasta
        - Ulica
        - Kod pocztowy
        - Rodzaj miejsca (Budowa, sklep etc.)
    - Numer telefonu
    - Login
    - Hasło
    - NIP firmy
    - Zamówienia
- Zamówienia
    - Identyfikator klienta
    - Lista produktów
    - Identyfikator sklepu
    - Data zamówienia
    - Status zamówienia

- Dostawca
    - Nazwa
    - Identyfikator produktu dostarczanego
- Magazyn
    - Adres
        - Miasto
        - Ulica
        - Kod pocztowy

2.2. Wymagania użytkownika

2.2.1. Szef

- Dodaje sklepy
- Zarządza managerami
    - tworzy nową umowę (ustala zarobki)
    - przydzielenie do konkretnego sklepu
- Zarządza środkami
    - przeglądanie środków (przeznacza więcej na dany sklep lub buduje nowy sklep)
- Posiada podgląd na wszystkich pracowników
- Generuje raporty finansowe

2.2.2. Manager

- Zarządza pracownikami
    - dodaje pracowników
    - ustala pensje
- Przydziela zamówienia
    - zmienia status zamówienia z *złożone* -> *przyjęte*
- Rozpatruje wnioski uzupełnienia magazynu
    - składanie zamówień
    - utworzenie kosztorysów
- Tworzenie raportów finansowych
- Przeglądanie pracowników
    - ich zarobki, umowę [tylko pracowników którzy są w tym samym sklepie co on]

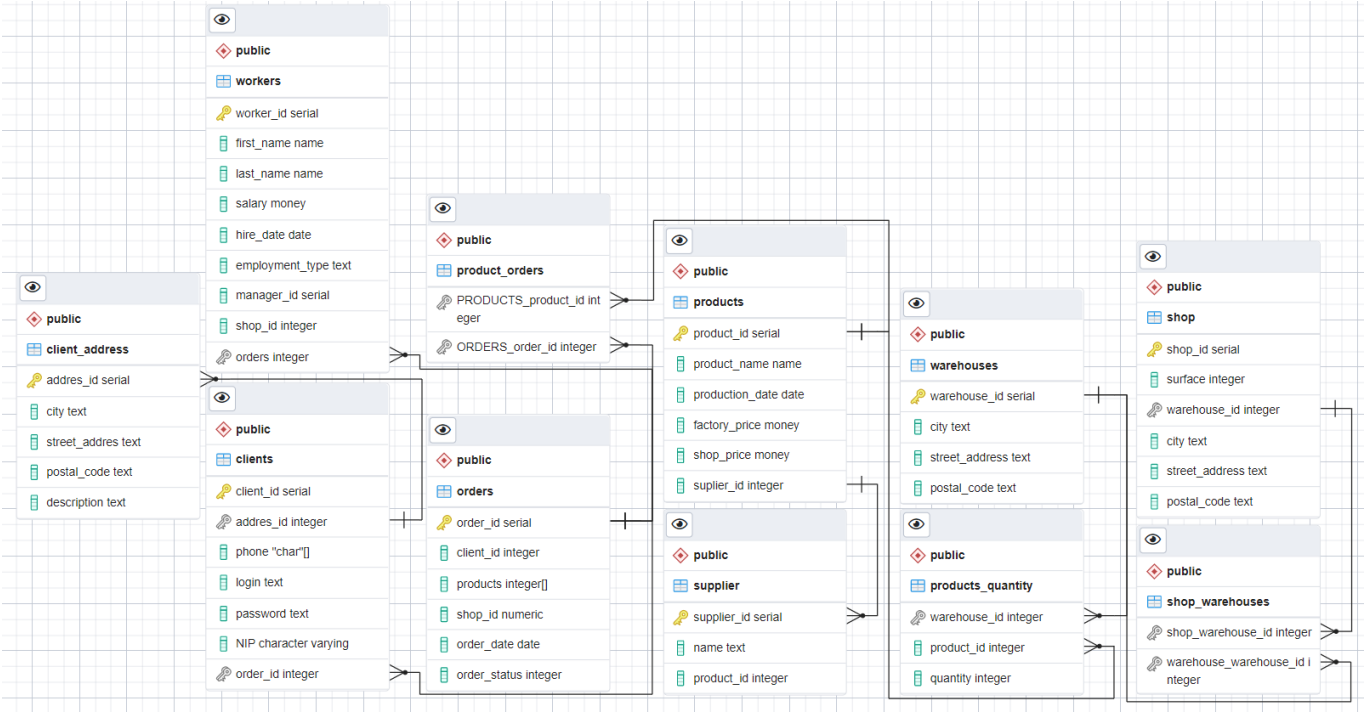
2.2.3. Pracownik

- Wykonuje zamówienia
- składa wniosek o uzupełnienie magazynu o dane towary
- Składa wniosek o podwyżkę
- Przyjmowanie uzupełnień magazynu
    - wpisuje na stan
- Przeglądanie stanu magazynu, sklepu
- zmienia status zamówień
    - *przyjęte* -> *w trakcie*
    - *w trakcie* -> *skompletowane*

2.2.4. Klient

- Składa zamówienie
    - do zamówienia wybiera sposób dokumentacji [Faktura lub Paragon]
- Anuluje zamówienia
    - tylko gdy zamówienie nie zostało przyjęte przez pracownika
- Przeglądanie zamówień

2.3. Model bazy danych



3. Implementacja bazy danych

Jako system zarządzający wybraliśmy `PostgreSQL`, a jako narzędzie wspomagające administrację bazy wybraliśmy `pgAdmin 4`. Za pomocą `pgAdmin 4` wygenerowaliśmy kod SQL tworzący bazę danych na podstawie wcześniej stworzonego w tym programie diagramu.

3.1. Przykładowe skrypty

3.1.1. Tworzenie przykładowych tabel

3.1.1.1. Tworzenie tabeli produktów

```
CREATE TABLE public.products
(
    product_id serial NOT NULL,
    product_name name NOT NULL,
    production_date date NOT NULL,
    factory_price money NOT NULL,
    shop_price money NOT NULL,
    suplier_id integer NOT NULL,
```

```
    PRIMARY KEY (product_id),
    UNIQUE(suplier_id)
);
```

3.1.1.2. Tworzenie tabeli klientów

```
CREATE TABLE public.clients
(
    client_id serial NOT NULL,
    address_id integer NOT NULL,
    phone integer NOT NULL,
    login text NOT NULL,
    password text NOT NULL,
    nip integer NOT NULL,
    order_id integer,
    PRIMARY KEY (client_id),
    UNIQUE (address_id,order_id)
);
```

3.1.1.3. Tworzenie tabeli pracowników

```
CREATE TABLE public.workers
(
    worker_id serial NOT NULL,
    first_name name NOT NULL,
    last_name name NOT NULL,
    salary money NOT NULL,
    hire_date date NOT NULL,
    employment_type text NOT NULL,
    manager_id serial NOT NULL,
    shop_id integer NOT NULL,
    orders integer,
    PRIMARY KEY (worker_id)

);
```

3.1.2. Przykładowe transakcje (Przypadki użycia)

3.1.2.1. Generacja raportu finansowego

```
SELECT SUM(products.shop_price)
FROM orders, product_orders, products
WHERE orders.order_id = 1;
```

Przykładowy rezultat:

sum
$2,203,266.00

3.1.2.2. Przeglądanie pracowników przypisanych do managera

```
SELECT first_name, last_name, salary, employment_type
FROM workers
WHERE manager_id = 1;
```

first_name	last_name	salary	employment_type
Amadeusz	Kwiatkowski	$1,532.55	sprzątaczka
Alina	Nowakowski	$5,656.11	sprzedawca
Alberta	Dąbrowski	$9,096.50	księgowy
Albert	Kwiatkowski	$4,658.55	sprzątaczka
Aleksander	Nowakowski	$3,258.44	dozorca
Ada	Wiśniewski	$4,038.82	księgowy
Adam	Zieliński	$2,186.20	magazynier
Alfred	Kwiatkowski	$6,761.83	magazynier
Agnieszka	Piotrowski	$1,939.54	dozorca
Apolinary	Kamiński	$2,774.83	dozorca
Angelina	Piotrowski	$6,896.26	sprzątaczka

4. Podsumowanie

Udało nam się utworzyć z naszych pomysłów bazę danych. Posiada ona sporą większość zakładanych przez nas podczas etapu planowania funkcjonalności. Dzięki temu że zanim zrobiliśmy praktyczną część projektu, dobrze zaplanowaliśmy jak baza powinna wyglądać, udało nam się w głównej mierze uniknąć problemów. Zdaliśmy sobie sprawę że posiadanie kopii zapasowej całej bazy, albo skryptów tworzących tabele, dane etc. jest bardzo ważne. Bez nich, gdyby wystąpił poważny błąd, moglibyśmy stracić mnóstwo czasu na odtwarzanie bazy oraz danych.

5. Literatura

Dokumentacja PostgreSQL

Dokumentacja pgAdmin4

Narzędzie do wygenerowania danych

ERD

Szczegółowy opis postaci znormalizowanych