Przedmiot: Bazy danych 1
Autorzy: Jan Jędrzejewski, Szymon Wysocki, Patrycja Wysocka
Zespół: Z15
Prowadzący projektu: dr inż. Piotr Maciąg
Data: 28.01.2022 r.

Projekt – Baza danych dla obsługi magazynu

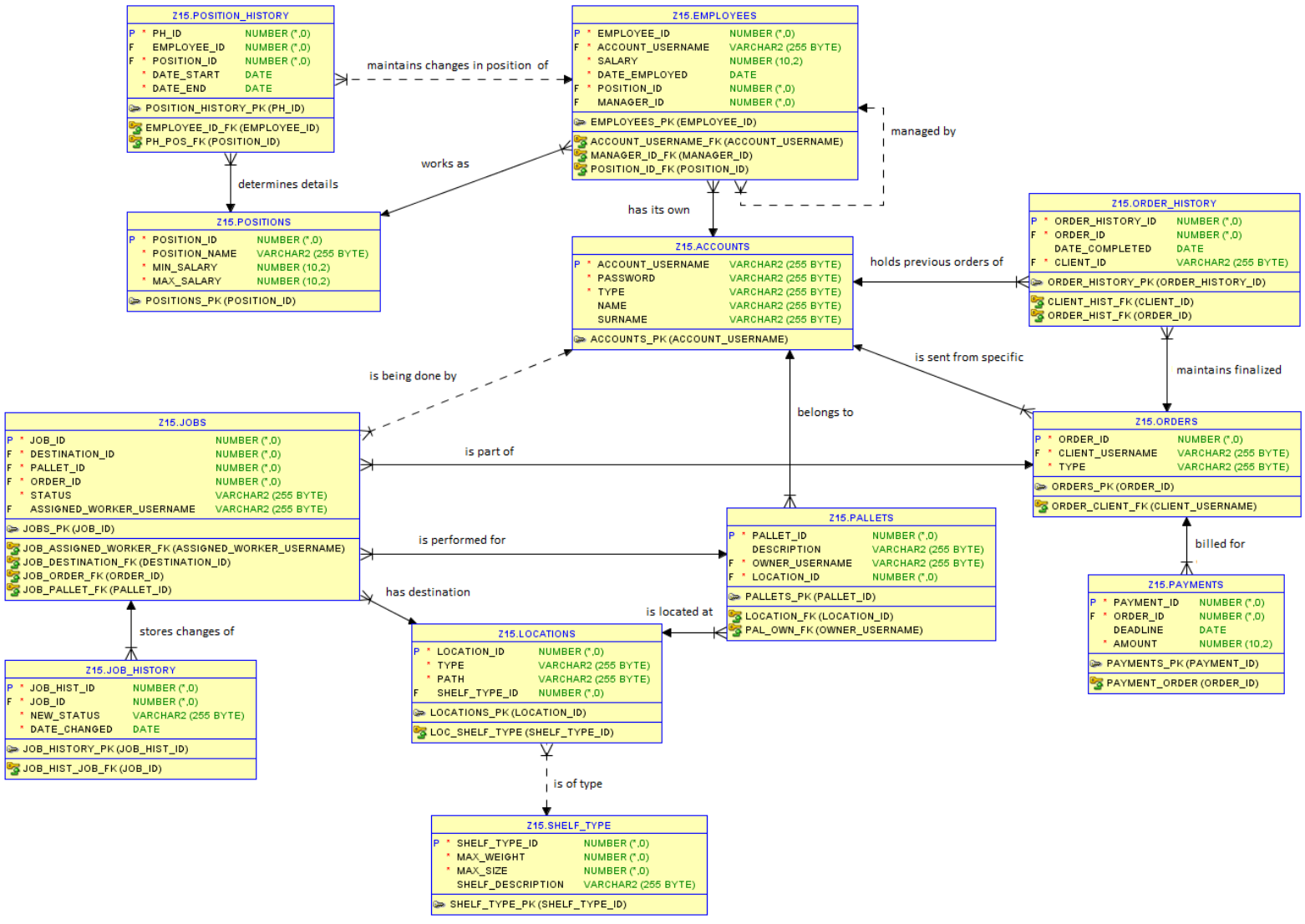
1. Cel i opis projektu

Celem projektu była implementacja bazy danych oraz aplikacji w programie Java spełniające podstawowe funkcjonalności umożliwiające obsługę magazynu.

Użyte technologie:

- Baza danych Oracle (wydziałowa)
- JPA(Hibernate) do połączenia z bazą z poziomu aplikacji w Javie
- JBcrypt - biblioteka do bezpiecznego haszowania haseł

2. Schemat E/R:



3. Opis tabel

Accounts – tabela zawierająca dane kont jakie istnieją w systemie. Pozwala na zalogowanie się do aplikacji - dla każdego typu konta wyświetlany jest inny widok spełniający inne funkcjonalności. Hasła są hashowane przed wstawieniem do tabeli.

Nazwa atrybutu	Wymagane	Typ	Opis
Account_username	tak	VARCHAR(255)	login użytkownika
Password	tak	VARCHAR(255)	hasło użytkownika (salted hash)
Type	tak	VARCHAR(255)	typ konta(Klient, Pracownik lub Manager)
Name	nie	VARCHAR(255)	imię Użytkownika
Surname	nie	VARCHAR(255)	nazwisko użytkownika

Employees – tabela przytrzymująca dodatkowe dane dotyczące tylko pracowników.

Nazwa atrybutu	Wymagane	Typ	Opis
Employee_id	tak	INTEGER	Numer identyfikacyjny pracownika
Acocunt_username	tak	VARCHAR(255)	Login użytkownika
Salary	tak	NUMBER(10, 2)	Regularnie wypłacana pensja (bez bonusów)
Date_employed	tak	DATE	Data od kiedy pracownik jest zatrudniony w firmie
Position_id	Tak	INTEGER	Identyfikator pozycji
Manager_id	Nie	INTEGER	Identyfikator managera danego pracownika (o ile go ma)

Positions – tabela pozwalająca uzyskać podstawowe dane charakterystyczne dla każdego stanowiska

Nazwa atrybutu	Wymagane	Typ	Opis
Position_id	tak	INTEGER	Identyfikator stanowiska
Position_name	tak	VARCHAR(255)	Nazwa stanowiska
Min_salary	tak	NUMBER(10, 2)	Minimalna wypłata, jaką mogą uzyskać pracownicy na tym stanowisku
Max_salary	tak	NUMBER(10, 2)	Maksymalna wypłata, jaką mogą uzyskać pracownicy na tym stanowisku

Position_history – tabela przetrzymująca historię, pracowników, którzy byli na danych stanowiskach, umożliwia śledzenie awansów. Docelowo przy każdej zmianie stanowiska powinien uruchamiać się wyzwalacz, który dodaje rekord.

Nazwa atrybutu	Wymagane	Typ	Opis
Ph_id	tak	INTEGER	Identyfikator zmiany
Employee_id	tak	INTEGER	Identyfikator pracownika, którego zmiana dotyczyła
Position_id	tak	INTEGER	Identyfikator stanowiska
Date_start	Tak	DATE	Data, w której pracownik rozpoczął pracować na tym stanowisku
Date_end	tak	DATE	Data, w której pracownik ukończył pracować na tym stanowisku

Jobs – tabela zawierająca dane każdego pojedynczego zadania (takiego, które w danej chwili może wykonywać magazynier) zawierającego przedmioty z konkretnego zamówienia.

Nazwa atrybutu	Wymagane	Typ	Opis
Job_id	tak	INTEGER	Identyfikator zadania
Destination_id	tak	INTEGER	Identyfikator lokalizacji, do której ma zostać przeniesione zadanie
Pallet_id	tak	INTEGER	Identyfikator palety, na której mieszczą się rzeczy
Order_id	tak	INTEGER	Identyfikator zamówienia, do którego należy zadanie
Status	tak	VARCHAR(255)	Status zamówienia (dostępne: PLANNED, PENDING, IN_PROGRESS, COMPLETED)
Assigned_worker_username	nie	VARCHAR(255)	Identyfikator magazyniera, któr dostal priorytet, aby wykonać dane zadanie

Job_history – tabela umożliwiająca śledzenie historii zadań (widoczne są wszystkie zmiany w statusach każdego zadania). Docelowo przy każdej zmianie stanowiska powinien uruchamiać się wyzwalacz, który dodaje rekord.

Nazwa atrybutu	Wymagane	Typ	Opis
Job_hist_id	tak	INTEGER	Identyfikator zmiany
Job_id	tak	INTEGER	Identyfikator zadania
New_status	tak	VARCHAR(255)	Nowy status zamówienia
Date_changed	tak	DATE	Data w której zmieniono status zamówienia

Locations – tabela zawierająca wszystkie lokalizacje magazynu oraz ich szczegóły – dzięki czemu nie trzeba w wielu miejscach powielać tych samych informacji

Nazwa atrybutu	Wymagane	Typ	Opis
Location_id	tak	INTEGER	Identyfikator lokalizacji
Type	tak	VARCHAR(255)	Typ lokalizacji (dostępne: SHELF, IN_RAMP, OUT_RAMP)
Path	tak	VARCHAR(255)	Dokładny opis lokalizacji w formacie ‘alejka/półka/regał’
Shelf_type_id	nie	INTEGER	Identyfikator typu półki

Shelf_type – rodzaje i właściwości półek

Nazwa atrybutu	Wymagane	Typ	Opis
Shelf_type_id	tak	INTEGER	Identyfikator typu półki
Max_weight	tak	INTEGER	Maksymalna waga przedmiotów jakie mogą być na półce
Max_size	tak	INTEGER	Maksymalny rozmiar towarów jakie mogą znajdować się na półce
Shelf_description	nie	VARCHAR(255)	opis

Pallets – tabela zawierająca dane palety, tak, żeby można było jasno określić do którego klienta należą rzeczy na danej palecie oraz łatwo odnaleźć jej lokalizację.

Nazwa atrybutu	Wymagane	Typ	Opis
Pallet_id	tak	INTEGER	Identyfikator palety
Description	nie	VARCHAR(255)	Opis towarów jakie znajdują się na palecie
Owner_username	tak	VARCHAR(255)	Identyfikator klienta, do którego należą rzeczy na tej palecie
Loaction_id	tak	INTEGER	Identyfikator lokalizacji, na której znajduje się paleta

Orders – tabela w której są zawarte wszystkie zamówienia przebywające w magazynie (na 1 zamówienie składa się jedno, bądź wiele zadań do wykonania przez pracownika – towary umieszczane są na kilku paletach, więc się rozdrabniają na kilka zadań)

Nazwa atrybutu	Wymagane	Typ	Opis
Order_id	tak	INTEGER	Identyfikator zamówienia
Client_username	tak	VARCHAR(255)	Identyfikator klienta
Type	tak	VARCHAR(255)	Typ zamówienia (dostępne: IN, OUT) (dodaj do magazynu/wyjmij z magazynu)

Order_history – tabela umożliwiająca przegląd skończonych zamówień. Przy zmianie statusu uruchamiany jest trigger, który umieszcza w niej nowy rekord.

Nazwa atrybutu	Wymagane	Typ	Opis
Order_history_id	tak	INTEGER	Identyfikator zmiany
Order_id	tak	INTEGER	Identyfikator zamówienia
Date_completed	nie	DATE	Data zakończenia wykonywania zamówienia
Client_id	tak	INTEGER	Identyfikator zlecającego

Payments – tabela w której są przetrzymywane koszta jakie klient musi ponieść w związku z konkretnym zamówieniem. Koszt zamówienia liczony jest wedle wzoru przez funkcję. Klienci mają ustalony deadline zapłaty i po jego przekroczeniu naliczane są odsetki.

Nazwa atrybutu	Wymagane	Typ	Opis
Payment_id	tak	INTEGER	Identyfikator płatności
Order_id	tak	INTEGER	Identyfikator zamówienia
Deadline	nie	DATE	Data deadline'u
Amount	tak	NUMBER(10, 2)	Kwota jaką musi zapłacić zamawiający

4. Zaimplementowane funkcje/procedury/triggery:

Funkcje:
- Obliczanie ceny zamówienia,
- obliczanie bonusu jakie otrzymuje manager.

Procedury:
- Tworzenie nowego zamówienia,
- przypisywanie magazyniera do danego zadania, wykonanie zadania,
- wypisanie dla każdego klienta sumy płatności jakie musi wykonać.

Triggery:
- Sprawdzanie czy jest jedna paleta na jednej półce,
- sprawdzanie czy konto przypisane do zadanie jest typu pracownik,
- sprawdzanie czy konto przypisane do zamówienia jest typu klient
- dodawanie nowych wierszy do tabeli order_history.

Sekwencje: Zostały użyte przy automatycznym generowaniu nowych identyfikatorów do tabel.

5. Podział aplikacji na widoki oraz zaimplementowane funkcjonalności:

Aplikacja po uruchomieniu pokazuje **ekran logowania** - jest na nim możliwość zalogowania się (oczywiście po wpisaniu odpowiednich danych uwierzytelniających), a także utworzenie nowego konta. Po zalogowaniu się na odpowiednie konto aplikacje przekierowuje na kolejny ekran, odpowiadający typowi konta:

- **Ekran managera** - w bocznym panelu pokazuje przegląd zawartości magazynu w formie drzewka alejek/regałów/półek. W głównym panelu także w formie drzewka dostępna jest lista zamówień, a w nich przypisane do nich konkretne zadania, razem z ich docelową lokalizacją. Manager zarządza przydzielaniem konkretnych zadań do swoich pracowników.

- **Ekran klienta** - składa się z dwóch głównych tabel, w których wyświetlane są informacje o aktualnych zamówieniach złożonych do zlecenia (z prawej strony) i wszystkie przedmioty jakie przechowujemy obecnie w magazynie (z lewej strony). Pod tabelami mamy możliwość zlecenie dodania nowego przedmiotu do magazynu i wyjęcia z magazynu znajdującego się tam przedmiotu.

- **Ekran pracownika -** pokazuje listę jego obecnych zadań do wykonania. Po wyborze konkretnego zadania aplikacja dynamicznie wyświetla szczegóły dotyczące danego zadania w tabeli obok. Jeżeli pracownik zdecyduje się na wykonanie zadania to przez naciśnięcie na przycisk “do” jest przekierowywany na kolejny pełnoekranowy widok:

    - Ekran zadania pracownika - pełnoekranowy widok, który pokazuje pracownikowi gdzie ma zlokalizować dane zadanie. W tym miejscu pracownik może potwierdzić wykonanie zadania, bądź zrezygnować z jego realizacji. W obu przypadkach wysyłane są odpowiednie sygnały do bazy danych.

6. Aplikacja w języku JAVA - instrukcja uruchomienia

Gotowe pliki .jar dostępne są w zakładce Releases. Wymagana wersja Javy to 11. Po uruchomieniu należy zalogować się na odpowiednie konto.
Przykładowe dane:

- klient - username: c1 / hasło: pass
- manager - username: m1 / hasło: pass
- pracownik - username: w1 / hasło: pass

Link do repozytorium:

https://gitlab-stud.elka.pw.edu.pl/swysocki/pap21z-z15/-/tree/main

Link do folderu ze skryptami sql:

https://gitlab-stud.elka.pw.edu.pl/swysocki/pap21z-z15/-/tree/main/sql