



# Podstawy baz danych

dzień i godz zajęć: śr 11:30

nr zespołu: 4

**Autorzy:** Filip Rutkowski, Dominik Wójcik, Weronika Latos

## 1. Wymagania i funkcje systemu

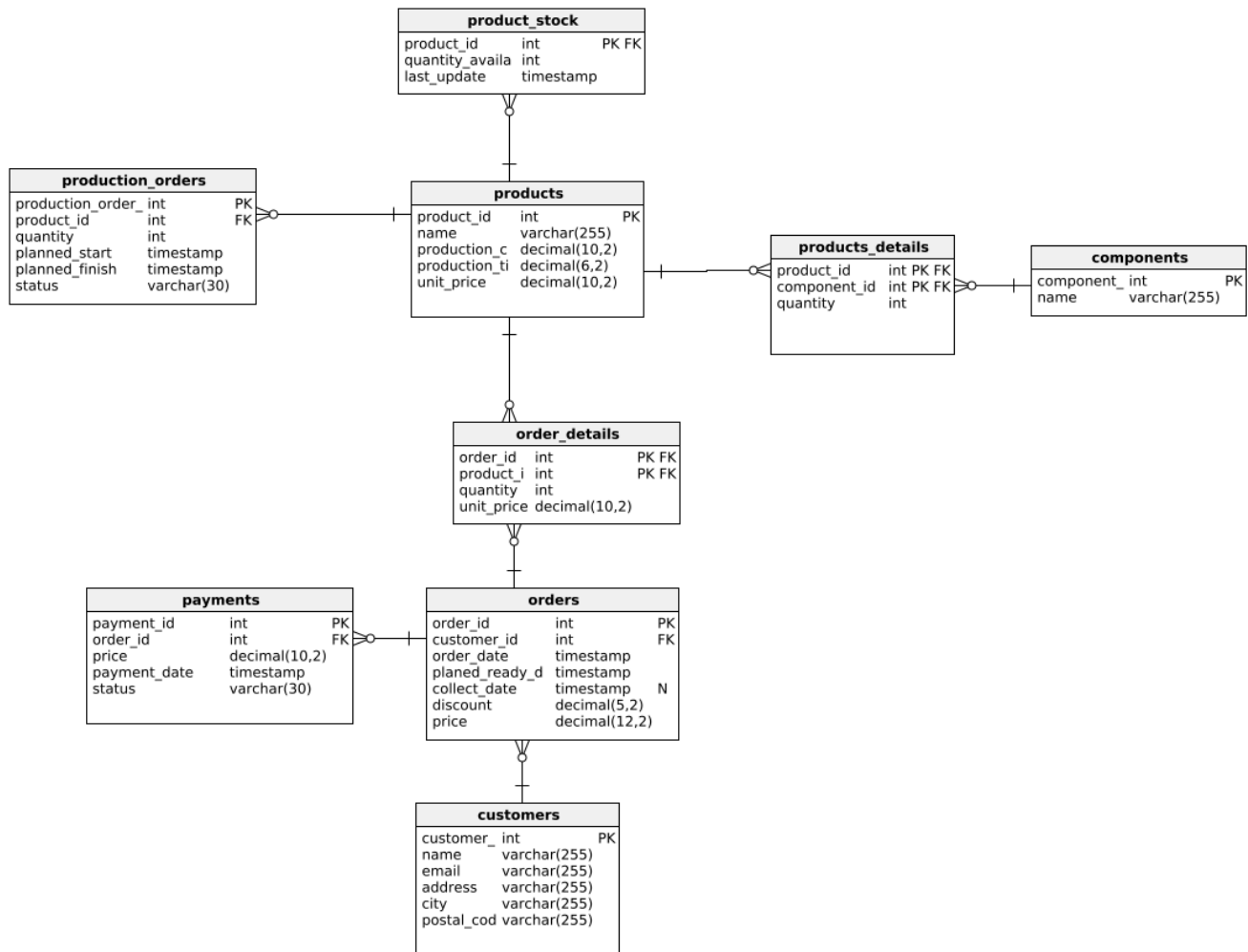
System służy do obsługi działalności firmy produkcyjno-usługowej zajmującej się wytwarzaniem oraz sprzedażą mebli, które są wyposażeniem pomieszczeń z urządzeniami komputerowymi (m. in. krzesła, biurka, biurka gamingowe, stoły, fotele biurowe, fotele gamingowe, ruchome stojaki na projektory oraz tablice interaktywne). System umożliwia monitorowanie procesu sprzedaży, stanów magazynowych, planowanie produkcji oraz obsługę zamówień klientów.

### funkcje

- dodawanie i obsługa zamówień
- obsługa płatności
- produkcja nowych mebli
- aktualizowanie stanu magazynu

## 2. Baza danych

### Schemat bazy danych



### Opis poszczególnych tabel

Z racji na to że docelowa baza będzie w SQL Server, w zapytaniach zostało użyte DATETIME, a nie TIMESTAMP

## tabela components

```
CREATE TABLE components (  
    component_id INT NOT NULL,  
    name VARCHAR(255) NOT NULL,  
    CONSTRAINT components_pk PRIMARY KEY (component_id)  
);
```

Nazwa atrybutu	Typ	Opis/Uwagi
component_id	Integer	id części
name	Varchar	nazwa części (metal, tworzywa,elementy łączące, robocizna)

## tabela products

```
CREATE TABLE products (  
    product_id INT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    production_cost DECIMAL(10, 2) NOT NULL,  
    production_time DECIMAL(6,2) NOT NULL,  
    unit_price DECIMAL(10, 2) NOT NULL  
);
```

Nazwa atrybutu	Typ	Opis/Uwagi
product_id	Integer	Klucz Główny (PK). Unikalny identyfikator produktu.
name	Varchar(255)	Nazwa produktu (np. Krzesło biurowe).
production_cost	Decimal(10, 2)	Koszt wytworzenia jednostkowego produktu

Nazwa atrybutu	Typ	Opis/Uwagi
product_time	Decimal(6,2)	Czas potrzebny na wytworzenie produktu, wyrażony w godzinach
unit_price	Decimal(10,2)	Cena sprzedaży jednostkowego produktu

## tabela **product\_details**

```
CREATE TABLE products_details (
  product_id int NOT NULL,
  component_id int NOT NULL,
  quantity int NOT NULL,
  CONSTRAINT products_details_pk PRIMARY KEY (product_id, component_id)
);
```

- klucze obce

```
ALTER TABLE products_details ADD CONSTRAINT FK_products_details_products
  FOREIGN KEY (product_id)
  REFERENCES products (product_id)
  ON DELETE CASCADE;
```

```
ALTER TABLE products_details ADD CONSTRAINT FK_products_details_components
  FOREIGN KEY (component_id)
  REFERENCES components (component_id)
  ON DELETE CASCADE;
```

- index

```
CREATE INDEX products_details_component_idx ON products_details (component_id ASC);
CREATE INDEX products_details_product_idx ON products_details (product_id ASC);
```

Nazwa atrybutu	Typ	Opis/Uwagi
product_id	Integer	Klucz Złożony (PK), Klucz Obcy (FK) do products
component_id	Integer	Klucz Złożony (PK), Klucz Obcy (FK) do components
quantity	Integer	Ilość danej części potrzebna do wytworzenia jednej jednostki produktu

## tabela product\_stock

```
product_id INT PRIMARY KEY,  
quantity_available INT NOT NULL,  
last_update DATETIME NOT NULL
```

- klucze obce

```
ALTER TABLE product_stock ADD CONSTRAINT FK_product_stock_products  
FOREIGN KEY (product_id)  
REFERENCES products (product_id);
```

Nazwa atrybutu	Typ	Opis/Uwagi
product_id	Integer	Klucz Główny (PK), Klucz Obcy (FK) do products
quantity_available	Integer	Bieżący zapas produktu w magazynie
last_update	Datetime	Data i czas ostatniej zmiany/aktualizacji stanu magazynowego

## tabela `production_orders`

```
CREATE TABLE production_orders (  
  production_order_id int NOT NULL,  
  product_id int NOT NULL,  
  quantity int NOT NULL,  
  planned_start DATETIME NOT NULL,  
  planned_finish DATETIME NOT NULL,  
  status varchar(30) NOT NULL,  
  CONSTRAINT production_order_pk PRIMARY KEY (production_order_id)  
);
```

- klucze obce

```
ALTER TABLE production_orders ADD CONSTRAINT FK_production_orders_products  
  FOREIGN KEY (product_id)  
  REFERENCES products (product_id);
```

- index

```
CREATE INDEX production_order_product_idx ON production_order (product_id ASC);
```

Nazwa atrybutu	Typ	Opis/Uwagi
production_order_id	Integer	Klucz Główny (PK). Identyfikator zlecenia produkcyjnego.
product_id	Integer	Klucz Obcy (FK) do <code>products</code> . Produkt, który ma zostać wytworzony.
quantity	Integer	Ilość produktu do wyprodukowania
planned_start	DATETIME	Planowana data rozpoczęcia prac
planned_finish	DATETIME	Planowana data zakończenia prac (na podstawie <code>products.production_time</code> )
status	Varchar(30)	Status zlecenia ('Planowane', 'W realizacji',

Nazwa atrybutu	Typ	Opis/Uwagi
		'Zakończone').

tabela customers

```
CREATE TABLE customers (  
  customer_id int NOT NULL,  
  name varchar(255) NOT NULL,  
  email varchar(255) NOT NULL,  
  address varchar(255) NOT NULL,  
  city varchar(255) NOT NULL,  
  postal_code varchar(255) NOT NULL,  
  CONSTRAINT customers_pk PRIMARY KEY (customer_id)  
);
```

Nazwa atrybutu	Typ	Opis/Uwagi
customer_id	Integer	Klucz Główny (PK). Unikalny identyfikator klienta.
name	Integer	Imię i Nazwisko klienta lub nazwa firmy.
email	Ilość produktu do wyprodukowania	Adres e-mail klienta.
address	DATETIME	Adres ulicy i numer.
city	DATETIME	Miejscowość.
postal_code	Varchar(30)	Kod pocztowy.

## tabela orders

```
CREATE TABLE dbo.orders (  
    order_id INT NOT NULL,  
    customer_id INT NOT NULL,  
    order_date DATETIME2 NOT NULL,  
    planed_ready_date DATETIME2 NOT NULL,  
    collect_date DATETIME2 NULL,  
    discount DECIMAL(5,2) NOT NULL,  
    price DECIMAL(12,2) NOT NULL,  
    CONSTRAINT orders_pk PRIMARY KEY (order_id)  
);
```

- klucze obce

```
ALTER TABLE orders ADD CONSTRAINT FK_order_customers  
    FOREIGN KEY (customer_id)  
    REFERENCES customers (customer_id)  
    ON DELETE CASCADE;
```

- index

```
CREATE INDEX order_customers_idx ON orders (customer_id ASC);
```

Nazwa atrybutu	Typ	Opis/Uwagi
order_id	Integer	Klucz Główny (PK). Unikalny identyfikator zamówienia.
customer_id	Integer	Klucz Obcy (FK) do <code>customers</code> . Klient składający zamówienie.
order_date	DATETIME	Data złożenia zamówienia.
planed_ready_date	DATETIME	Oczekiwana data realizacji zamówienia.
collect_date	DATETIME / NULL	Data odbioru zamówienia.

Nazwa atrybutu	Typ	Opis/Uwagi
discount	Decimal(5,2)	Rabat jednostkowy przydzielony do zamówienia (wartość procentowa, np. 0.15).
price	Decimal(12,2)	Całkowita cena zamówienia po rabacie.

## tabela `order_details`

```
CREATE TABLE order_details (
  order_id int NOT NULL,
  product_id int NOT NULL,
  quantity int NOT NULL,
  unit_price decimal(10,2) NOT NULL,
  CONSTRAINT order_details_pk PRIMARY KEY (order_id,products_id)
);
```

- klucze obce

```
ALTER TABLE order_details ADD CONSTRAINT FK_order_details_order
  FOREIGN KEY (order_id)
  REFERENCES orders (order_id)
  ON DELETE CASCADE;
```

```
ALTER TABLE order_details ADD CONSTRAINT FK_order_details_products
  FOREIGN KEY (products_id)
  REFERENCES products (product_id)
  ON DELETE CASCADE;
```

- index

```
CREATE INDEX order_details_order_idx ON order_details (order_id ASC);
CREATE INDEX order_details_product_idx ON order_details (products_id ASC);
```

Nazwa atrybutu	Typ	Opis/Uwagi
order_id	Integer	Klucz Złożony (PK), Klucz Obcy (FK) do <code>orders</code> .
products_id	Integer	Klucz Złożony (PK), Klucz Obcy (FK) do <code>products</code> .
quantity	Integer	Ilość zamówionego produktu.
unit_price	Decimal(10,2)	Cena jednostkowa produktu w chwili złożenia zamówienia.

## tabela `payments`

```
CREATE TABLE payments (
  payment_id int NOT NULL,
  order_id int NOT NULL,
  price decimal(10,2) NOT NULL,
  payment_date timestamp NOT NULL,
  status varchar(30) NOT NULL,
  CONSTRAINT payment_pk PRIMARY KEY (payment_id)
);
```

- klucze obce

```
ALTER TABLE payment ADD CONSTRAINT FK_payment_order
  FOREIGN KEY (order_id)
  REFERENCES orders (order_id);
```

Nazwa atrybutu	Typ	Opis/Uwagi
payment_id	Integer	Klucz Główny (PK). Identyfikator płatności.
order_id	Integer	Klucz Obcy (FK) do <code>orders</code> . Zamówienie, którego dotyczy płatność.

Nazwa atrybutu	Typ	Opis/Uwagi
price	Decimal(10,2)	Zapłacona kwota.
payment_date	DATETIME	Dokładny czas realizacji płatności.
status	Varchar(30)	Status płatności ('Zrealizowana', 'Anulowana').