

Hurtownie danych – Spr. 1. (dwa etapy)

PWr. WIiT, Data: 13.03.2025 r.

| | | |
|----------|----------------------------------|-------|
| Student | Email: 272677@student.pwr.edu.pl | Ocena |
| Indeks | <u>272677</u> | |
| Imię | <u>Michał</u> | |
| Nazwisko | <u>Dudniczenko</u> | |

Zestaw składa się z 1 zadania realizowanego w dwóch etapach oraz dodatkowych zadań z podstaw DML. Jeżeli nie potrafisz rozwiązać zadania, to próbuj podać, chociaż częściowe rozwiązanie lub uzasadnienie przyczyny braku rozwiązania. Pamiętaj o podaniu nr. indeksu oraz imienia i nazwiska.

SZBD: **MS SQL 2022/2019**

Baza danych: **Szpital**

Zad. 1.

Proszę utworzyć bazę danych, która umożliwi gromadzenie i przetwarzanie danych niezbędnych dla funkcjonowania szpitala. Rozpatrywany wycinek rzeczywistości (obszar dziedzinowy) obejmuje następujące określenia:

1. Szpital
2. Oddział
3. Lekarz
4. Pacjent
5. Choroba
6. Diagnoza

Etap I

Pierwszy etap procesu opracowania bazy danych dotyczy analizy obszaru dziedzinowego w zakresie wyżej wymienionych pojęć i obejmuje:

1. Charakterystykę **dziedziny** (słowny opis kluczowych aspektów dziedziny)
2. Definicję **podstawowych pojęć** (określenie semantyki obiektów), które mają być reprezentowane w bazie danych
3. Identyfikację **klas obiektów** dziedziny oraz ich własności
4. Identyfikację i specyfikację **reguł dziedzinowych** (biznesowych)
5. Definicję **słownika danych** dla lekarza, pacjenta, choroby
6. **Model konceptualny danych** oraz jego weryfikacja względem reguł biznesowych
7. Wykaz źródeł (strony Internetowe, książki, dokumentacje techniczne, itd.)

Poszczególne elementy mogą być opracowane (co najmniej jedna pozycja) z wykorzystaniem sztucznej inteligencji (LLM)

Zasady specyfikacji składowych dokumentacji:

2. Definicja podstawowych pojęć (słownik dziedzinowy)

Tabela 1. Słownik pojęć dziedzinowych

| Lp. | Nazwa | Definicja (semantyka) |
|-----|-------|-----------------------|
| 1. | | |
| 2. | | |

| | | |
|-----|--|--|
| ... | | |
|-----|--|--|

3. Specyfikacja własności klas obiektów

Tabela 2. Specyfikacja klas obiektów

| Lp. | Nazwa klasy | Zbiór własności klasy | Klucze kandydujące |
|-----|-------------|-----------------------|--------------------|
| 1. | | | |
| 2. | | | |
| ... | | | |

4. Specyfikacja reguł biznesowych

Tabela 3. Reguły biznesowe

| Lp. | Reguła |
|--------|--------|
| Reg/01 | |
| Reg/02 | |
| ... | |

Przykład:

Reg/01 – Szpital składa się z co najmniej jednego oddziału

Reg/02 – Oddział ...

5. Słownik danych

Tabela 4. Słownik atrybutów klas

| 1. | Nazwa klasy | | | |
|---------|-------------|------------|-----------|--------------------------|
| Lp. | Atrybut | Typ danych | Znaczenie | Ograniczenia dziedzinowe |
| 1. 1 | | | | |
| 1. 2 | | | | |
| ... | | | | |
| 2. | Nazwa klasy | | | |
| Lp. | Atrybut | Typ danych | Znaczenie | Ograniczenia dziedzinowe |
| 2. 1 | | | | |
| 2. 2 | | | | |
| ... | | | | |

6. Model konceptualny danych – diagram klas UML

Uwaga:

- Proszę nie modyfikować i nie usuwać definicji sprawozdania

- Dyskusja i ocena realizacji zadań pierwszego etapu odbędzie się na zajęciach lab. w dniu 20.03.2025 lub 21.03.2025

Rozwiązanie:

1. Charakterystykę dziedziny (słowny opis kluczowych aspektów dziedziny)

Szpital jest instytucją medyczną, posiada swój unikalny numer, nazwę oraz adres. Adres określony jest przez kraj, kod pocztowy, miasto, ulicę, numer domu i opcjonalnie numer lokalu. Szpital składa się z co najmniej jednego oddziału. Oddział posiada swój unikalny numer, nazwę, przypisany numer telefonu kontaktowego oraz całkowitą liczbę dostępnych łóżek dla pacjentów. Szpital zatrudnia lekarzy, którzy mogą pracować na dowolnej liczbie oddziałów. Oddział może mieć przypisanego ordynatora, czyli jednego z zatrudnionych lekarzy, który będzie zarządzał tym oddziałem. Lekarz może ordynować dowolną liczbę oddziałów i pracować w dowolnej liczbie szpitali. Lekarza identyfikuje numer prawa do wykonywania zawodu, posiada również określoną specjalizację i tytuł naukowy. Do szpitala zgłaszają się pacjenci, którzy podobnie jak lekarze są identyfikowani poprzez dane osobowe, na które składają się: numer PESEL, płeć, imię, nazwisko, data urodzenia i opcjonalnie numer telefonu i adres e-mail. Pacjentem jest osoba, która zgłosiła się z ważnym skierowaniem, została przywieziona przez zespół ratowników medycznych lub zgłosiła się z problemem medycznym wymagającym pilnej interwencji. Lekarze spotykają się z pacjentami na wizytach, podczas których diagnozują ich stan zdrowia. Lekarz może przyjmować dowolną liczbę pacjentów. Wizytę określają jej data i godzina, informacja o gabinecie, w którym się odbywa, opis wizyty oraz zalecenia, które lekarz przekazuje pacjentowi. Jeżeli wizyta była przeprowadzona po wcześniejszej rejestracji pacjenta z ważnym skierowaniem to również kod tego skierowania. Diagnoza lekarska może zidentyfikować jedną lub więcej chorób. Chorobę określają unikalny kod choroby, jej nazwa oraz informacja o tym, czy jest zakaźna. Pacjenci mogą zostać przyjęci na jeden z oddziałów szpitala. W tym samym czasie na oddziale może przebywać maksymalnie tylu pacjentów ile wynosi całkowita liczba łóżek na tym oddziale. Pacjent może w tym samym czasie być przyjęty tylko na jednym z oddziałów.

2. Definicja podstawowych pojęć (słownik dziedziny)

Tabela 5. Słownik pojęć dziedziny

| Lp. | Nazwa | Definicja (semantyka) |
|-----|-----------|---|
| 1. | Szpital | jednostka medyczna, składająca się z oddziałów, w których pracują lekarze i leczeni są pacjenci |
| 2. | Oddział | część szpitala specjalizująca się w leczeniu określonych schorzeń, np. kardiologia, neurologia |
| 3. | Adres | zbiór danych jednoznacznie identyfikujący lokalizację budynku |
| 4. | Lekarz | specjalista zatrudniony w szpitalu, przyjmuje pacjentów na wizytach, na których diagnozuje ich stan zdrowia |
| 5. | Pacjent | osoba, która zgłosiła się do szpitala. Może zostać zdiagnozowany przez lekarza w ramach wizyty i/lub zostać przyjęty na oddział |
| 6. | Choroba | niepożądany stan zdrowia, może zostać rozpoznany u pacjenta przez lekarza jako część wystawionej diagnozy podczas przeprowadzonej wizyty. Określona jest przez unikalny kod, nazwę oraz informację o tym, czy jest zakaźna. |
| 7. | Wizyta | oficjalne spotkanie pomiędzy lekarzem a pacjentem, podczas którego lekarz diagnozuje stan zdrowia pacjenta |
| 8. | Ordynator | jeden z zatrudnionych lekarzy, zarządza oddziałem szpitala |

3. Specyfikacja własności klas obiektów

Tabela 6. Specyfikacja klas obiektów

| Lp. | Nazwa klasy | Zbiór własności klasy | Klucze kandydujące |
|-----|-------------|--|--|
| 1. | Osoba | PESEL, plec, imię, nazwisko, dataUrodzenia, nrTel, email | PESEL |
| 2. | Lekarz | PESEL, plec, imię, nazwisko, dataUrodzenia, nrTel, email, pwz, specjalizacja, tytułNaukowy | PESEL, pwz |
| 3. | Adres | kraj, kodPocztowy, miasto, ulica, nrDomu, nrLokalu | (kraj, kodPocztowy, miasto, ulica, nrDomu, nrLokalu) |
| 4. | Szpital | nrSzpitala, nazwa | nrSzpitala |
| 5. | Oddział | nrOddziału, nazwa, nrTel, liczbaŁozek, nrSzpitala | (nrSzpitala, nrOddziału) |
| 6. | Choroba | kodChoroby, nazwa, czyZakazna | kodChoroby |

uwaga: atrybut nrSzpitala pojawia się z powodu kompozycji, oddział musi istnieć w ramach szpitala, numer szpitala jest niezbędny do identyfikacji instancji oddziału

4. Specyfikacja reguł biznesowych

Tabela 7. Reguły biznesowe

| Lp. | Reguła |
|--------|---|
| Reg/01 | Pacjent może w tym samym czasie znajdować się na maksymalnie jednym oddziale. |
| Reg/02 | Pacjent może mieć wizyty u dowolnie wielu lekarzy. |
| Reg/03 | Osoba musi podać jeden adres zamieszkania. |
| Reg/04 | Lekarz może przyjmować na wizyty dowolną ilość pacjentów. |
| Reg/05 | Lekarz może pracować na dowolnej liczbie oddziałów. |
| Reg/06 | Lekarz może pracować w dowolnej liczbie szpitali. |
| Reg/07 | Lekarz może ordynować dowolnie wiele oddziałów. |
| Reg/08 | Szpital musi mieć określony jeden adres. |
| Reg/09 | Pod danym adresem może mieszkać jeden lub więcej pacjentów. |
| Reg/10 | W szpitalu może pracować dowolnie wielu lekarzy. |
| Reg/11 | Szpital musi posiadać jeden lub więcej oddziałów. |
| Reg/12 | Oddział musi należeć do jednego szpitala. |
| Reg/13 | Oddział może być ordynowany przez maksymalnie jednego lekarza. |
| Reg/14 | Na oddziale może znajdować się maksymalnie tylu pacjentów, ile wynosi liczba łóżek na oddziale. |
| Reg/15 | Na oddziale może pracować dowolnie wielu lekarzy. |
| Reg/16 | Choroba może zostać zdiagnozowana w ramach dowolnie wielu wizyt. |

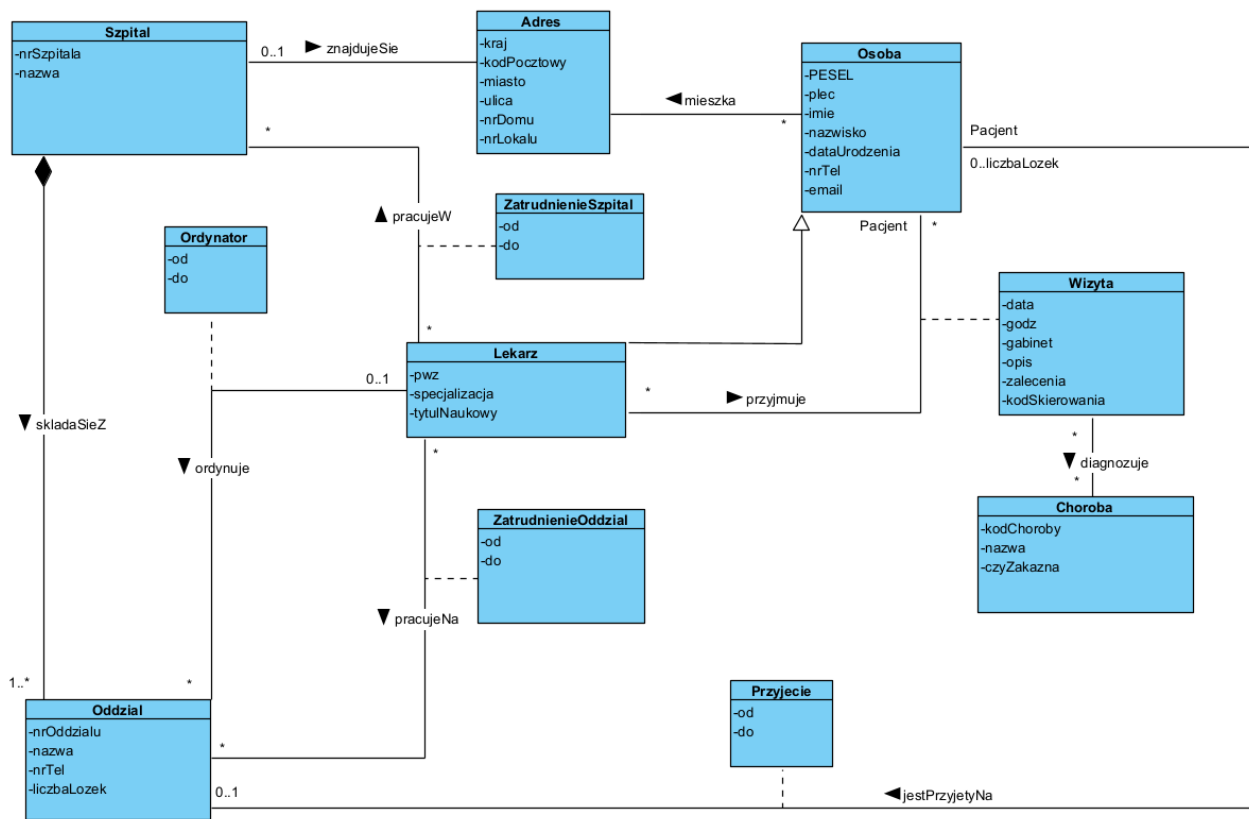
5. Słownik danych

Tabela 8. Słownik atrybutów klas

| | | | | |
|-----|---------------|---------------|--|--------------------------|
| 1. | Osoba | | | |
| Lp. | Atrybut | Typ danych | Znaczenie | Ograniczenia dziedzinowe |
| 1.1 | PESEL | CHAR(11) | numer PESEL osoby | NOT NULL |
| 1.2 | plec | CHAR | płeć osoby | 'm' OR 'k' |
| 1.3 | imie | NVARCHAR(100) | imię osoby | NOT NULL |
| 1.4 | nazwisko | NVARCHAR(100) | nazwisko osoby | NOT NULL |
| 1.5 | dataUrodzenia | DATE | data urodzenia osoby | NOT NULL |
| 1.6 | nrTel | VARCHAR(20) | numer telefonu kontaktowego osoby | |
| 1.7 | email | VARCHAR(100) | adres e-mail osoby | |
| 2. | Lekarz | | | |
| Lp. | Atrybut | Typ danych | Znaczenie | Ograniczenia dziedzinowe |
| 2.1 | pwz | VARCHAR(10) | unikalny numer prawa do wykonywania zawodu lekarza | NOT NULL |
| 2.2 | specjalizacja | NVARCHAR(100) | specjalizacja lekarza, np. otolaryngolog | |
| 2.3 | tytulNaukowy | NVARCHAR(100) | tytuł naukowy lekarza | NOT NULL |
| 3. | Adres | | | |
| Lp. | Atrybut | Typ danych | Znaczenie | Ograniczenia dziedzinowe |
| 3.1 | kraj | NVARCHAR(100) | nazwa kraju | NOT NULL |
| 3.2 | kodPocztowy | VARCHAR(10) | kod pocztowy kraju | NOT NULL |
| 3.3 | miasto | NVARCHAR(100) | nazwa miasta | NOT NULL |
| 3.4 | ulica | NVARCHAR(100) | nazwa ulicy | NOT NULL |
| 3.5 | nrDomu | NVARCHAR(10) | numer budynku | NOT NULL |
| 3.6 | nrLokalu | NVARCHAR(10) | opcjonalny numer lokalu w budynku | |
| 4. | Szpital | | | |
| Lp. | Atrybut | Typ danych | Znaczenie | Ograniczenia dziedzinowe |
| 4.1 | nrSzpitala | INT | unikalny numer szpitala | NOT NULL, >0 |
| 4.2 | nazwa | NVARCHAR(100) | nazwa szpitala | NOT NULL |
| 5. | Oddział | | | |
| Lp. | Atrybut | Typ danych | Znaczenie | Ograniczenia dziedzinowe |
| 5.1 | nrOddzialu | INT | unikalny numer oddziału w obrębie danego szpitala | NOT NULL, >0 |
| 5.2 | nazwa | NVARCHAR(100) | nazwa oddziału | NOT NULL |
| 5.3 | nrTel | VARCHAR(20) | numer telefonu kontaktowego do oddziału | NOT NULL |

| | | | | |
|-----|-------------|---------------|---|--------------------------|
| 5.4 | liczbaLozek | INT | całkowita liczba dostępnych łóżek dla pacjentów na oddziale | NOT NULL, >0 |
| 6. | Choroba | | | |
| Lp. | Atrybut | Typ danych | Znaczenie | Ograniczenia dziedziczne |
| 6.1 | kodChoroby | VARCHAR(20) | unikalny kod choroby ICD-10 | NOT NULL |
| 6.2 | nazwa | NVARCHAR(100) | nazwa choroby | NOT NULL |
| 6.3 | czyZakazna | BIT | wartość prawda/fałsz określająca czy choroba jest zakaźna | NOT NULL, 0 OR 1 |

6. Model konceptualny danych – diagram klas UML



7. Wykaz źródeł (strony Internetowe, książki, dokumentacje techniczne, itd.)

ChatGPT, <https://remedium.md/icd10>

Etap II

Etap ten obejmuje następujące zadania:

1. Implementację conceptualnego modelu danych w systemie MS SQL 2019 w postaci skryptu SQL (definicje tabel wraz z ograniczeniami)
2. Przygotowanie danych testowych, które pozwolą zweryfikować poprawność zdefiniowania ograniczeń dziedzinowych. Wykonanie testów (bez testów wydajnościowych) poprzez wykonanie próby wprowadzenia kilku rekordów do każdej tabeli, sprawdzając w ten sposób poprawność implementacji (zarówno poprawne dane, jak i niezgodne z obowiązującymi regułami – komentując i wyjaśniając uzyskane komunikaty z systemu SZBD). Dane testowe należy przygotować w postaci skryptu SQL (zbiór poleceń INSERT INTO ...)

Rozwiązanie:

1. Skrypt SQL zawierający definicje tabel wraz z ograniczeniami:

```
CREATE TABLE Szpital (  
    id INT IDENTITY(1,1),  
    nazwa NVARCHAR(100) NOT NULL,  
    kraj NVARCHAR(100) NOT NULL,  
    kodPocztowy VARCHAR(10) NOT NULL,  
    miasto NVARCHAR(100) NOT NULL,  
    ulica NVARCHAR(100) NOT NULL,  
    nrDomu NVARCHAR(10) NOT NULL,  
    nrLokalu NVARCHAR(10),  
  
    CONSTRAINT pk_szpital PRIMARY KEY (id),  
);  
  
CREATE TABLE Oddzial (  
    id INT IDENTITY(1,1),  
    nr INT NOT NULL,  
    nazwa NVARCHAR(100) NOT NULL,  
    liczbaLozek INT NOT NULL,  
    idSzpitala INT NOT NULL,  
  
    CONSTRAINT pk_oddzial PRIMARY KEY (id),  
    CONSTRAINT fk_oddzial_szpital FOREIGN KEY (idSzpitala) REFERENCES Szpital(id),  
    CONSTRAINT ck_oddzial_nr CHECK (nr > 0),  
    CONSTRAINT ck_oddzial_liczbaLozek CHECK (liczbaLozek > 0),  
    CONSTRAINT uq_oddzial_nr_szpital UNIQUE (nr, idSzpitala),  
);
```

```
CREATE TABLE Lekarz (  
    id INT IDENTITY(1,1),  
    PESEL CHAR(11) NOT NULL,  
    plec CHAR(1) NOT NULL,  
    imie NVARCHAR(100) NOT NULL,  
    nazwisko NVARCHAR(100) NOT NULL,  
    dataUrodzenia DATE NOT NULL,  
    nrTel VARCHAR(20) NOT NULL,  
    email VARCHAR(100) NOT NULL,  
    kraj NVARCHAR(100) NOT NULL,  
    kodPocztowy VARCHAR(10) NOT NULL,  
    miasto NVARCHAR(100) NOT NULL,  
    ulica NVARCHAR(100) NOT NULL,  
    nrDomu NVARCHAR(10) NOT NULL,  
    nrLokalu NVARCHAR(10),  
    pwz VARCHAR(10) NOT NULL,  
    tytulNaukowy NVARCHAR(100) NOT NULL,  
    specjalizacja NVARCHAR(100),  
  
    CONSTRAINT pk_lekarz PRIMARY KEY (id),  
    CONSTRAINT ck_lekarz_plec CHECK (plec IN ('k', 'm')),  
    CONSTRAINT ck_lekarz_PESEL CHECK (PESEL LIKE '[0-9]{11}'),  
    CONSTRAINT uq_lekarz_PESEL UNIQUE (PESEL),  
    CONSTRAINT uq_lekarz_pwz UNIQUE (pwz),  
);
```



```
CREATE TABLE Pacjent (  
    id INT IDENTITY(1,1),  
    PESEL CHAR(11) NOT NULL,  
    plec CHAR(1) NOT NULL,  
    imie NVARCHAR(100) NOT NULL,  
    nazwisko NVARCHAR(100) NOT NULL,  
    dataUrodzenia DATE NOT NULL,  
    nrTel VARCHAR(20),  
    email VARCHAR(100),  
    kraj NVARCHAR(100) NOT NULL,  
    kodPocztowy VARCHAR(10) NOT NULL,  
    miasto NVARCHAR(100) NOT NULL,  
    ulica NVARCHAR(100) NOT NULL,  
    nrDomu NVARCHAR(10) NOT NULL,  
    nrLokalu NVARCHAR(10),  
  
    CONSTRAINT pk_pacjent PRIMARY KEY (id),  
    CONSTRAINT ck_pacjent_plec CHECK (plec IN ('k', 'm')),  
    CONSTRAINT ck_pacjent_PESEL CHECK (PESEL LIKE '[0-9]{11}'),  
    CONSTRAINT uq_pacjent_PESEL UNIQUE (PESEL),  
);
```

```
CREATE TABLE Choroba (  
    id INT IDENTITY(1,1),  
    kodChoroby VARCHAR(20) NOT NULL,  
    nazwa NVARCHAR(100) NOT NULL,  
    czyZakazna BIT NOT NULL,  
  
    CONSTRAINT pk_choroba PRIMARY KEY (id),  
    CONSTRAINT uq_choroba_kodChoroby UNIQUE (kodChoroby),  
);
```

```
CREATE TABLE Ordynator (  
    id INT IDENTITY(1,1),  
    od DATE NOT NULL DEFAULT GETDATE(),  
    do DATE,  
    idOddzialu INT NOT NULL,  
    idLekarza INT NOT NULL,  
  
    CONSTRAINT pk_ordynator PRIMARY KEY (id),  
    CONSTRAINT fk_ordynator_oddzial FOREIGN KEY (idOddzialu) REFERENCES Oddzial(id),  
    CONSTRAINT fk_ordynator_lekarz FOREIGN KEY (idLekarza) REFERENCES Lekarz(id),  
    CONSTRAINT ck_ordynator_od_do CHECK (do IS NULL OR do > od),  
);
```

```

CREATE TABLE ZatrudnienieSzpital (
    id INT IDENTITY(1,1),
    od DATE NOT NULL DEFAULT GETDATE(),
    do DATE,
    idSzpitala INT NOT NULL,
    idLekarza INT NOT NULL,

    CONSTRAINT pk_zatrudnienieSzpital PRIMARY KEY (id),
    CONSTRAINT fk_zatrudnienieSzpital_szpital FOREIGN KEY (idSzpitala) REFERENCES Szpital(id),
    CONSTRAINT fk_zatrudnienieSzpital_lekarz FOREIGN KEY (idLekarza) REFERENCES Lekarz(id),
    CONSTRAINT ck_zatrudnienieSzpital_od_do CHECK (do IS NULL OR do > od),
);

CREATE TABLE ZatrudnienieOddzial (
    id INT IDENTITY(1,1),
    od DATE NOT NULL DEFAULT GETDATE(),
    do DATE,
    idOddzialu INT NOT NULL,
    idLekarza INT NOT NULL,

    CONSTRAINT pk_zatrudnienieOddzial PRIMARY KEY (id),
    CONSTRAINT fk_zatrudnienieOddzial_oddzial FOREIGN KEY (idOddzialu) REFERENCES Oddzial(id),
    CONSTRAINT fk_zatrudnienieOddzial_lekarz FOREIGN KEY (idLekarza) REFERENCES Lekarz(id),
    CONSTRAINT ck_zatrudnienieOddzial_od_do CHECK (do IS NULL OR do > od),
);

CREATE TABLE Przyjecie (
    id INT IDENTITY(1,1),
    od DATE NOT NULL DEFAULT GETDATE(),
    do DATE,
    idOddzialu INT NOT NULL,
    idPacjenta INT NOT NULL,

    CONSTRAINT pk_przyjecie PRIMARY KEY (id),
    CONSTRAINT fk_przyjecie_oddzial FOREIGN KEY (idOddzialu) REFERENCES Oddzial(id),
    CONSTRAINT fk_przyjecie_pacjent FOREIGN KEY (idPacjenta) REFERENCES Pacjent(id),
    CONSTRAINT ck_przyjecie_od_do CHECK (do IS NULL OR do > od),
);

CREATE TABLE Wizyta (
    id INT IDENTITY(1,1),
    dataWizyty DATE NOT NULL,
    godzWizyty TIME NOT NULL,
    gabinet VARCHAR(10) NOT NULL,
    opis NVARCHAR(max) NOT NULL,
    zalecenie NVARCHAR(max) NOT NULL,
    kodSkierowania VARCHAR(100),
    idLekarza INT NOT NULL,
    idPacjenta INT NOT NULL,

    CONSTRAINT pk_wizyta PRIMARY KEY (id),
    CONSTRAINT fk_wizyta_lekarz FOREIGN KEY (idLekarza) REFERENCES Lekarz(id),
    CONSTRAINT fk_wizyta_pacjent FOREIGN KEY (idPacjenta) REFERENCES Pacjent(id),
);

```

```
CREATE TABLE Diagnoza (  
    id INT IDENTITY(1,1),  
    idWizyty INT NOT NULL,  
    idChoroby INT NOT NULL,  
  
    CONSTRAINT pk_diagnoza PRIMARY KEY (id),  
    CONSTRAINT fk_diagnoza_wizyta FOREIGN KEY (idWizyty) REFERENCES Wizyta(id),  
    CONSTRAINT fk_diagnoza_choroba FOREIGN KEY (idChoroby) REFERENCES Choroba(id),  
);
```

Po uruchomieniu skryptu otrzymałem poniższy wynik, wszystko dobrze.

```
Commands completed successfully.
```

```
Completion time: 2025-03-24T18:03:12.0478081+01:00
```

2. Skrypt SQL wstawiający prawidłowe dane:

```
-- wstawienie prawidłowych danych do tabel

INSERT INTO Szpital (nazwa, kraj, kodPocztowy, miasto, ulica, nrDomu, nrLokalu) VALUES
('Szpital Centralny', 'Polska', '00-001', 'Warszawa', 'Marszałkowska', '10', '2'),
('Szpital Miejski', 'Polska', '30-002', 'Kraków', 'Długa', '20', NULL),
('Szpital Wojewódzki', 'Polska', '50-003', 'Wrocław', 'Świdnicka', '5', 'B');

INSERT INTO Oddzial (nr, nazwa, liczbaLozek, idSzpitala) VALUES
(1, 'Kardiologia', 50, 1),
(2, 'Neurologia', 40, 1),
(3, 'Ortopedia', 35, 1),
(4, 'Chirurgia', 45, 1),
(5, 'Onkologia', 30, 1),
(6, 'Pediatria', 60, 1),
(7, 'Geriatrya', 25, 1),
(1, 'Urologia', 20, 2),
(2, 'Kardiologia', 40, 2),
(1, 'Pulmonologia', 30, 3);

INSERT INTO Lekarz (PESEL, plec, imie, nazwisko, dataUrodzenia, nrTel, email, kraj, kodPocztowy, miasto, ulica, nrDomu, nrLokalu, pwz, tytu
('90010112345', 'm', 'Jan', 'Kowalski', '1990-01-01', '600123456', 'jan.kowalski@example.com', 'Polska', '00-001', 'Warszawa', 'Marszałkows
('85051256789', 'k', 'Anna', 'Nowak', '1985-05-12', '601987654', 'anna.nowak@example.com', 'Polska', '30-002', 'Kraków', 'Długa', '15', '5'
('78092311122', 'm', 'Piotr', 'Zieliński', '1978-09-23', '602345678', 'piotr.zielinski@example.com', 'Polska', '40-003', 'Katowice', 'Sokol
('92030433344', 'k', 'Maria', 'Wiśniewska', '1992-03-04', '603456789', 'maria.wisniewska@example.com', 'Polska', '50-004', 'Wrocław', 'Legn
('81071555566', 'm', 'Tomasz', 'Lewandowski', '1981-07-15', '604567890', 'tomasz.lewandowski@example.com', 'Polska', '60-005', 'Poznań', 'G
('95020177788', 'k', 'Karolina', 'Dąbrowska', '1995-02-01', '605678901', 'karolina.dabrowska@example.com', 'Polska', '70-006', 'Szczecin',
('87080899900', 'm', 'Andrzej', 'Wójcik', '1987-08-08', '606789012', 'andrzej.wojcik@example.com', 'Polska', '80-007', 'Gdańsk', 'Długa',
('93061222233', 'k', 'Joanna', 'Kamińska', '1993-06-12', '607890123', 'joanna.kaminska@example.com', 'Polska', '90-008', 'Łódź', 'Piotrkows
('76042944455', 'm', 'Marek', 'Szymański', '1976-04-29', '608901234', 'marek.szymanski@example.com', 'Polska', '20-009', 'Lublin', 'Lipowa'
('89031666677', 'k', 'Barbara', 'Jankowska', '1989-03-16', '609012345', 'barbara.jankowska@example.com', 'Polska', '10-010', 'Białystok',

INSERT INTO Pacjent (PESEL, plec, imie, nazwisko, dataUrodzenia, nrTel, email, kraj, kodPocztowy, miasto, ulica, nrDomu, nrLokalu) VALUES
('90010112345', 'm', 'Jan', 'Kowalski', '1990-01-01', '123456789', 'jan.kowalski@example.com', 'Polska', '00-001', 'Warszawa', 'Miodowa',
('85050554321', 'k', 'Anna', 'Nowak', '1985-05-05', '987654321', 'anna.nowak@example.com', 'Polska', '30-001', 'Kraków', 'Długa', '20', '5'
('92030423456', 'm', 'Piotr', 'Wiśniewski', '1992-03-04', '111222333', 'piotr.w@example.com', 'Polska', '80-001', 'Gdańsk', 'Grunwaldzka',
('89080887654', 'k', 'Maria', 'Lewandowska', '1989-08-08', '444555666', 'maria.l@example.com', 'Polska', '40-001', 'Katowice', 'Kościuszki'
('95020109876', 'm', 'Tomasz', 'Dąbrowski', '1995-02-01', '777888999', 'tomasz.d@example.com', 'Polska', '20-001', 'Lublin', 'Lipowa', '8',
('81090934567', 'k', 'Karolina', 'Wójcik', '1981-09-09', '222333444', 'karolina.w@example.com', 'Polska', '60-001', 'Poznań', 'Święty Marci
('87070776543', 'm', 'Michał', 'Zieliński', '1987-07-07', '555666777', 'michal.z@example.com', 'Polska', '90-001', 'Łódź', 'Piotrkowska',
('93050556789', 'k', 'Alicja', 'Kamińska', '1993-05-05', '888999000', 'alicja.k@example.com', 'Polska', '50-001', 'Wrocław', 'Kazimierza Wi
('94030365432', 'm', 'Paweł', 'Jankowski', '1994-03-03', '999000111', 'pawel.j@example.com', 'Polska', '70-001', 'Szczecin', 'Jagiellońska'
('96060643210', 'k', 'Ewa', 'Mazur', '1996-06-06', '000111222', 'ewa.m@example.com', 'Polska', '10-001', 'Olsztyn', 'Dworcowa', '5', NULL);
```

```
INSERT INTO Choroba (kodChoroby, nazwa, czyZakazna) VALUES
('C001', 'Grypa', 1),
('C002', 'Zapalenie płuc', 1),
('C003', 'Nowotwór płuc', 0),
('C004', 'Choroba wieńcowa', 0),
('C005', 'Cukrzyca', 0),
('C006', 'Gruźlica', 1),
('C007', 'Asthma', 0),
('C008', 'Nadciśnienie', 0),
('C009', 'Padaczka', 0),
('C010', 'Alzheimer', 0);
```

```
INSERT INTO Ordynator (od, do, idOddzialu, idLekarza) VALUES
('2020-01-01', NULL, 1, 1),
('2019-06-15', '2023-06-30', 2, 2),
('2021-03-01', NULL, 3, 3),
('2018-09-10', '2022-09-09', 4, 6),
('2022-02-20', NULL, 5, 7);
```

```
INSERT INTO ZatrudnienieSzpital (od, do, idSzpitala, idLekarza) VALUES
('2015-05-10', NULL, 1, 3),
('2017-08-12', NULL, 2, 4),
('2016-11-20', '2023-11-19', 1, 5),
('2023-11-23', NULL, 2, 5),
('2019-01-01', NULL, 3, 6),
('2018-03-15', NULL, 1, 8),
('2021-09-01', NULL, 1, 9),
('2017-12-05', '2022-12-04', 2, 10),
('2016-06-30', NULL, 1, 1),
('2019-04-22', NULL, 1, 1);
```

```
INSERT INTO ZatrudnienieOddzial (od, do, idOddzialu, idLekarza) VALUES
('2009-09-17', NULL, 1, 3),
('2011-12-01', NULL, 1, 4),
('2002-01-27', '2005-05-14', 3, 3),
('1994-10-09', NULL, 4, 3),
('1997-05-24', NULL, 1, 4),
('2004-12-24', '2010-10-06', 5, 5),
('2006-07-06', NULL, 6, 6),
('1995-09-25', NULL, 1, 5),
('1994-08-06', '2003-02-13', 2, 8),
('2016-06-30', NULL, 3, 9);
```

```

INSERT INTO ZatrudnienieOddzial (od, do, idOddzialu, idLekarza) VALUES
('2009-09-17', NULL, 1, 3),
('2011-12-01', NULL, 1, 4),
('2002-01-27', '2005-05-14', 3, 3),
('1994-10-09', NULL, 4, 3),
('1997-05-24', NULL, 1, 4),
('2004-12-24', '2010-10-06', 5, 5),
('2006-07-06', NULL, 6, 6),
('1995-09-25', NULL, 1, 5),
('1994-08-06', '2003-02-13', 2, 8),
('2016-06-30', NULL, 3, 9);

INSERT INTO Przyjecie (od, do, idOddzialu, idPacjenta) VALUES
('1998-03-31', '2007-12-12', 1, 1),
('2014-10-10', NULL, 1, 2),
('2016-05-27', '2017-12-16', 2, 3),
('2017-08-02', NULL, 3, 3),
('1994-07-04', NULL, 4, 3),
('2003-01-21', NULL, 5, 4),
('2005-03-30', '2007-12-01', 1, 1),
('2019-12-15', NULL, 2, 2),
('1997-01-21', NULL, 1, 3),
('2009-11-18', NULL, 1, 1);

INSERT INTO Wizyta (dataWizyty, godzWizyty, gabinet, opis, zalecenie, kodSkierowania, idLekarza, idPacjenta) VALUES
('2012-04-04', '12:40', '119', 'Przykładowy opis wizyty: pacjent z bólem głowy, potrzeba dalszej diagnostyki.', 'Zalecenie: odpoczynek, uniki', '309f', '1', 1),
('2020-09-19', '07:50', '309f', 'Przykładowy opis wizyty: pacjent z bólem głowy, potrzeba dalszej diagnostyki.', 'Zalecenie: odpoczynek, uniki', '309f', '1', 1),
('2023-03-01', '16:40', '256', 'Przykładowy opis wizyty: pacjent z bólem głowy, potrzeba dalszej diagnostyki.', 'Zalecenie: odpoczynek, uniki', '256', '1', 1),
('2022-10-15', '16:10', '145a', 'Przykładowy opis wizyty: pacjent z bólem głowy, potrzeba dalszej diagnostyki.', 'Zalecenie: odpoczynek, uniki', '145a', '1', 1),
('1997-04-28', '12:30', '342', 'Przykładowy opis wizyty: pacjent z bólem głowy, potrzeba dalszej diagnostyki.', 'Zalecenie: odpoczynek, uniki', '342', '1', 1),
('2010-08-23', '08:20', '100', 'Przykładowy opis wizyty: pacjent z bólem głowy, potrzeba dalszej diagnostyki.', 'Zalecenie: odpoczynek, uniki', '100', '1', 1),
('1992-01-03', '07:00', '38', 'Przykładowy opis wizyty: pacjent z bólem głowy, potrzeba dalszej diagnostyki.', 'Zalecenie: odpoczynek, uniki', '38', '1', 1),
('2018-07-09', '14:20', '292', 'Przykładowy opis wizyty: pacjent z bólem głowy, potrzeba dalszej diagnostyki.', 'Zalecenie: odpoczynek, uniki', '292', '1', 1),
('2001-06-07', '13:00', '108b', 'Przykładowy opis wizyty: pacjent z bólem głowy, potrzeba dalszej diagnostyki.', 'Zalecenie: odpoczynek, uniki', '108b', '1', 1),
('2014-06-13', '15:30', '324a', 'Przykładowy opis wizyty: pacjent z bólem głowy, potrzeba dalszej diagnostyki.', 'Zalecenie: odpoczynek, uniki', '324a', '1', 1);

INSERT INTO Diagnoza (idWizyty, idChoroby) VALUES
(1, 1),
(2, 1),
(2, 2),
(2, 3),
(4, 2),
(5, 3),
(5, 1),
(2, 5),
(3, 2),
(1, 9);

```

Po uruchomieniu otrzymałem wiele odpowiedzi świadczących o pomyślnym wstawieniu danych, treść jednej z wiadomości:

```
(10 rows affected)
```

```
Completion time: 2025-03-24T18:16:49.3583789+01:00
```


Testowanie ograniczeń:

```
-- próba wstawienia numeru <= 0
INSERT INTO Oddzial (nr, nazwa, liczbaLozek, idSzpitala) VALUES
(-1, 'Kardiologia', 50, 1);
```

The INSERT statement conflicted with the CHECK constraint "ck_oddzial_nr". The conflict occurred in database "Szpital", table "dbo.Oddzial", column 'nr'.

```
-- próba wstawienia oddziału o parze (nr, idSzpitala), która już istnieje w bazie
INSERT INTO Oddzial (nr, nazwa, liczbaLozek, idSzpitala) VALUES
(1, 'Kardiologia', 50, 1);
```

Violation of UNIQUE KEY constraint 'uq_oddzial_nr_szpital'. Cannot insert duplicate key in object 'dbo.Oddzial'. The duplicate key value is (1, 1).

```
-- próba wstawienia oddziału o liczbie łóżek <= 0
INSERT INTO Oddzial (nr, nazwa, liczbaLozek, idSzpitala) VALUES
(1, 'Kardiologia', 0, 10);
```

The INSERT statement conflicted with the CHECK constraint "ck_oddzial_liczbaLozek". The conflict occurred in database "Szpital", table "dbo.Oddzial", column 'liczbaLozek'.

```
-- próba wstawienia -powtarza się PESEL
INSERT INTO Lekarz (PESEL, plec, imie, nazwisko, dataUrodzenia, nrTel, email, kraj, kodPocztowy, miasto, ulica, nrDomu, nrLokalu, pwz, tytuł)
VALUES ('90010112345', 'm', 'Jan', 'Kowalski', '1990-01-01', '600123456', 'jan.kowalski@example.com', 'Polska', '00-001', 'Warszawa', 'Marszałkowska', '12345', '6789', '123456', 'Lekarz');
```

Violation of UNIQUE KEY constraint 'uq_lekarz_PESEL'. Cannot insert duplicate key in object 'dbo.Lekarz'. The duplicate key value is (90010112345).

```
-- próba wstawienia -powtarza się pwz
INSERT INTO Lekarz (PESEL, plec, imie, nazwisko, dataUrodzenia, nrTel, email, kraj, kodPocztowy, miasto, ulica, nrDomu, nrLokalu, pwz, tytuł)
VALUES ('90010912345', 'm', 'Jan', 'Kowalski', '1990-01-01', '600123456', 'jan.kowalski@example.com', 'Polska', '00-001', 'Warszawa', 'Marszałkowska', '12345', '6789', '123456', 'Lekarz');
```

Violation of UNIQUE KEY constraint 'uq_lekarz_pwz'. Cannot insert duplicate key in object 'dbo.Lekarz'. The duplicate key value is (123456).

```
-- próba wstawienia PESEL w złym formacie
INSERT INTO Lekarz (PESEL, plec, imie, nazwisko, dataUrodzenia, nrTel, email, kraj, kodPocztowy, miasto, ulica, nrDomu, nrLokalu, pwz, tytuł)
VALUES ('123456789', 'm', 'Jan', 'Kowalski', '1990-01-01', '600123456', 'jan.kowalski@example.com', 'Polska', '00-001', 'Warszawa', 'Marszałkowska', '12345', '6789', '123456', 'Lekarz');
```

The INSERT statement conflicted with the CHECK constraint "ck_lekarz_PESEL". The conflict occurred in database "Szpital", table "dbo.Lekarz", column 'PESEL'.

```
-- próba wstawienia płeć w złym formacie
INSERT INTO Lekarz (PESEL, plec, imie, nazwisko, dataUrodzenia, nrTel, email, kraj, kodPocztowy, miasto, ulica, nrDomu, nrLokalu, pwz, tytuł)
VALUES ('12345678900', 'x', 'Jan', 'Kowalski', '1990-01-01', '600123456', 'jan.kowalski@example.com', 'Polska', '00-001', 'Warszawa', 'Marszałkowska', '12345', '6789', '123456', 'Lekarz');
```

The INSERT statement conflicted with the CHECK constraint "ck_lekarz_plec". The conflict occurred in database "Szpital", table "dbo.Lekarz", column 'plec'.

```
-- próba wstawienia - kod choroby się powtarza
INSERT INTO Choroba (kodChoroby, nazwa, czyZakazna) VALUES
('C001', 'Grypa', 1),
```

Violation of UNIQUE KEY constraint 'uq_choroba_kodChoroby'. Cannot insert duplicate key in object 'dbo.Choroba'. The duplicate key value is (C001).

```
-- próba dodania - data do wcześniejsza niż data od
INSERT INTO Ordynator (od, do, idOddzialu, idLekarza) VALUES
('2020-01-01', '2019-09-08', 1, 1);
```

The INSERT statement conflicted with the CHECK constraint "ck_ordynator_od_do". The conflict occurred in database "Szpital", table "dbo.Ordynator".

```
-- próba wstawienia drugi raz tej samej choroby w ramach tej samej wizyty
INSERT INTO Diagnoza (idWizyty, idChoroby) VALUES
(1, 1);
```

Violation of UNIQUE KEY constraint 'uq_diagnoza_wizyta_choroba'. Cannot insert duplicate key in object 'dbo.Diagnoza'. The duplicate key value is (1, 1).

Komentarz: wszystkie ograniczenia dziedziczne działają prawidłowo, dostajemy dokładną informację zwrotną, że to właśnie te ustawione ograniczenie zostało naruszone.

Etap III

Podstawy SQL

Źródło danych:

Baza danych: AdventureWorks - SalesOrderHeader

Zad. 1.

Proszę utworzyć wykaz zamówień złożonych w pierwszym roku rejestracji zamówień (Identyfikator, Rok, Kwota zamówienia) . Przykładowy wynik zapytania przedstawiony jest poniżej w tabeli 5.

Zapytanie SQL + fragment wyniku (4 rekordy z ?)

Tabela 5. Fragment wyników zapytania zad. 1.

Tabela 9. Wynik zapytania zad. 1.

| Identyfikator | Rok | Kwota |
|---------------|------|----------|
| 45266 | 2012 | 27605.63 |
| 45267 | 2012 | 3899.68 |
| 45268 | 2012 | 944.62 |
| 45269 | 2012 | 2280.14 |
| ... | ... | ... |

Rek.: 4/?

Rozwiązanie:

```
SELECT
    SalesOrderID AS "Identyfikator",
    YEAR(OrderDate) AS "Rok",
    ROUND(TotalDue, 2) AS "Kwota"
FROM Sales.SalesOrderHeader
WHERE OrderDate < DATEADD(YEAR, 1, (SELECT TOP 1 OrderDate FROM Sales.SalesOrderHeader ORDER BY 1 ASC));
```

Wyniki zapytania z zad. 1

| Identyfikator | Rok | Kwota |
|---------------|------|----------|
| 43659 | 2011 | 23153,23 |
| 43660 | 2011 | 1457,33 |
| 43661 | 2011 | 36865,80 |
| 43662 | 2011 | 32474,93 |

Rek.: 4/3025

Zad. 2.

Proszę wyznaczyć liczbę klientów, którzy mają liczbę zamówień:

1. mniejszą niż 10
2. w przedziale 10 - 19
3. co najmniej 20

Przykładowy wynik zapytania przedstawiony jest poniżej w tabeli 6.:

Tabela 6. Fragment wyników zapytania zad. 2. – wariant A

| Grupa | Liczba klientów |
|--------|-----------------|
| 0-9 | 19002 |
| 10-19 | 103 |
| 20 ... | 14 |

Rek.: 3/3

lub

Tabela 7. Fragment wyników zapytania zad. 2. – wariant B

| Grupa | 0-9 | 10-19 | 20 ... |
|-------|-------|-------|--------|
| | 19002 | 103 | 14 |

Rek.: 1

Rozwiązanie: Zapytanie SQL + wynik zapytania

Wariant A:

```
WITH Zamowienia AS (
    SELECT
        CustomerID,
        COUNT(*) AS LiczbaZamowien
    FROM Sales.SalesOrderHeader
    GROUP BY CustomerID
)
SELECT
    CASE
        WHEN LiczbaZamowien < 10 THEN '0-9'
        WHEN LiczbaZamowien < 20 THEN '10-19'
        ELSE '20 ...'
    END AS Grupa,
    COUNT(*) AS "Liczba Klientów"
FROM Zamowienia
GROUP BY
    CASE
        WHEN LiczbaZamowien < 10 THEN '0-9'
        WHEN LiczbaZamowien < 20 THEN '10-19'
        ELSE '20 ...'
    END
END
;
```

Wyniki zapytania z zad.2, wariant A

| Grupa | Liczba klientów |
|--------|-----------------|
| 0-9 | 19002 |
| 10-19 | 103 |
| 20 ... | 14 |

Wariant B:

```
WITH Zamowienia AS (  
    SELECT  
        CustomerID,  
        COUNT(*) AS LiczbaZamowien  
    FROM Sales.SalesOrderHeader  
    GROUP BY CustomerID  
)  
SELECT  
    '' AS Grupa,  
    SUM(CASE WHEN LiczbaZamowien < 10 THEN 1 ELSE 0 END) AS [0-9],  
    SUM(CASE WHEN LiczbaZamowien BETWEEN 10 AND 19 THEN 1 ELSE 0 END) AS [10-19],  
    SUM(CASE WHEN LiczbaZamowien >= 20 THEN 1 ELSE 0 END) AS [20 ...]  
FROM Zamowienia;
```

Wyniki zapytania z zad.2, wariant B

| Grupa | 0-9 | 10-19 | 20 ... |
|-------|-------|-------|--------|
| | 19002 | 103 | 14 |

Zad. 3.

Ustalić, jakie czynniki mają wpływ na liczbę dokonanych zakupów. Przykładowy wynik zapytania przedstawiony jest poniżej w tabeli 2.3.

Źródło danych: SalesOrderHeaderSalesReason, ?

Tabela 2.3. Fragment wyników zapytania zad. 3

| Czynnik | Dotyczy |
|--------------|---------|
| Price | 17473 |
| On Promotion | 3515 |
| Manufacturer | 1746 |
| ... | ... |

Rek.: 3/?

Rozwiązanie: Zapytanie SQL + fragment wyniku (4 rekordy z ?)

```
SELECT
    r.Name AS Czynnik,
    COUNT(*) AS Dotyczy
FROM Sales.SalesOrderHeaderSalesReason hr
JOIN Sales.SalesReason r ON r.SalesReasonID = hr.SalesReasonID
GROUP BY r.Name
ORDER BY 2 DESC;
```

Wyniki zapytania z zad.3

| Czynnik | Dotyczy |
|--------------|---------|
| Price | 17473 |
| On Promotion | 3515 |
| Manufacturer | 1746 |
| Quality | 1551 |

Rek.: 4/7

Wnioski:

Wykonałem zadania wchodzące w skład etapu 1. Osobiście nie lubię tworzenia projektu konceptualnego, ale rozumiem, że przy projektowaniu o wiele bardziej złożonego i skomplikowanego systemu, lepsze poznanie obszaru dziedzinowego mogłoby mieć bezpośrednie przełożenie na jakość końcowego produktu. Sztuczna inteligencja była pomocna, szczególnie ogólne pytania związane z dziedziną projektu, np. "Czy lekarz może pracować na kilku oddziałach?", "W jakim formacie jest pzw lekarza w Polsce?", "Jakie atrybuty może mieć klasa reprezentująca oddział?". Myślę, że pozwoliła mi zaoszczędzić trochę czasu i zaproponowała trochę rzeczy, na które sam nie wpadłem. Wykonałem zadanie 1 etapu 2, czyli transformację modelu konceptualnego do faktycznej implementacji w systemie SQL Server. Wprowadziłem trochę zmian względem projektu konceptualnego. Do każdej klasy dodałem atrybut id jako klucz główny, uważam, że jest to dobra praktyka programistyczna, dzięki temu w powiązanych tabelach możemy przechowywać INT zajmujący mniej miejsca niż np. NVARCHAR. Do tego wydaje mi się to bardziej naturalne, intuicyjne i upraszcza późniejsze tworzenie zapytań (szczególnie JOINy). W klasie szpitala id zastąpiło numer, ponieważ ich semantyka była taka sama. Zdecydowałem się utworzyć osobne tabele lekarz i pacjent, bez abstrakcyjnej osoby, ponieważ uważam, że wiele czytelniej jest rozbić te dane na 2 osobne tabele, ponieważ lekarz i pacjent, w kontekście szpitala są całkowicie różnymi od siebie bytami. Sytuacja, w której lekarz jest też zarejestrowany jako pacjent szpitala jest rzadka i moim zdaniem czytelność struktury bazy jest w tym przypadku dużo ważniejsza, dodatkowo, wtedy w każdym rekordzie Lekarza, musielibyśmy przechowywać referencję na Osobę, więc ostatecznie takie rozwiązanie mogłoby nawet skutkować utratą miejsca, zamiast jego zaoszczędzeniem. Zrezygnowałem również z osobnej klasy Adres, ponieważ i tak dla każdego lekarza, pacjenta i szpitala musimy przechowywać osobny rekord z danymi adresowymi, ponieważ nawet, jeżeli np. dwie osoby będą mieszkały pod tym samym adresem to jedna z nich może zmienić adres, a druga nie. Uważam zatem, że dane adresu bezpośrednio w tabelach Szpital, Pacjent i Lekarz jest lepszym rozwiązaniem, czytelniejszym i upraszczającym późniejsze zapytania, szczególnie, że nie przewidujemy możliwości zmiany zbioru atrybutów określających adres. Tutaj praktycznie nie korzystałem z AI, jedynie dokumentacja T-SQL z listą dostępnych typów i składnią poleceń. Część druga etapu 2 zajęła mi chyba najwięcej czasu. Darmowa wersja ChataGPT generowała dane słabej jakości, dużo musiałem ręcznie poprawiać, myślę, że następnym razem lepszym pomysłem będzie napisanie ręcznie prostego skryptu w pythonie. Etap 3 poszedł mi dość sprawnie, jedynie z zadaniem drugim miałem trochę problemów. Opierałem się głównie na materiałach wykładowych.

Uwaga:

- **Sprawozdanie bez wniosków końcowych nie będzie sprawdzane i tym samym ocena jest negatywna!**
- Plik ze sprawozdaniem powinien mieć nazwę **Spr01HD-index-Nazwisko - 2025** i format **pdf**
- **Kompletne sprawozdanie należy przekazać w terminie do 26.03.2025 godz. 20:00**
- Oprogramowanie MS SQL 2022/2019
- Model konceptualny danych należy przygotować wykorzystując narzędzie wspierające język UML np. Visual Paradigm
- Wnioski powinny zawierać ocenę możliwości wykorzystania sztucznej inteligencji