Министерство науки и высшего образования РФ

ФГАОУ ВПО

Национальный исследовательский технологический университет «МИСиС»

Контрольное домашнее задание

На тему

«Регистратура хирургической больницы»

Выполнил:

студент группы БИВТ-23-2

Тимонин.В.В.

Проверила:

Валова А.А.

Москва, 2024

**Оглавление**

1. Постановка задачи

2. Описание структуры БД

2.1. Вербальная модель

2.2. Реляционная модель

2.3. Анализ функциональных зависимостей

3. Заполнение БД информацией

4. Описание представлений

5. Описание функций

6. Описание хранимых процедур

7. Описание триггеров

8. Примеры работы с БД с использованием созданных объектов

9. Список литературы

**Постановка задачи**

Целью данной работы является создание базы данных для управления информацией о пациентах, врачах, отделениях, палатах, лечении и хирургических операциях в медицинском учреждении. База данных должна обеспечивать:

* Учет пациентов и их личных данных.
* Учет врачей, их специализаций и опыта работы.
* Учет отделений и палат в больнице.
* Учет лечения пациентов, включая диагнозы, даты начала и окончания лечения.
* Учет хирургических операций, их результатов и стоимости.
* Автоматический расчет общей стоимости лечения на основе проведенных операций.
* Управление счетами за лечение и их статусами оплаты.
* Предоставление информации о свободных палатах и пациентах, находящихся на лечении.

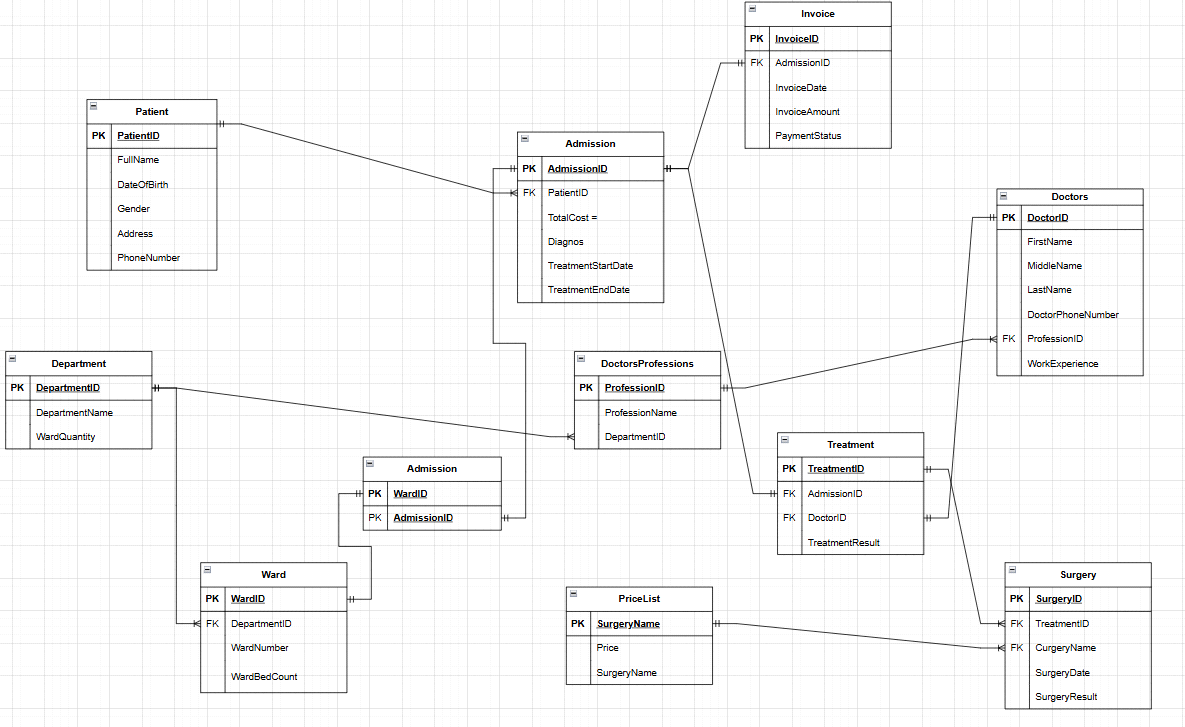
**Описание структуры БД**

**Вербальная модель**

База данных состоит из следующих сущностей:

* **Patient**: Информация о пациентах (ФИО, пол, дата рождения, телефон, адрес).
* **Department**: Отделения больницы (название, количество палат).
* **Doctors**: Врачи (ФИО, телефон, стаж работы, специализация).
* **DoctorsProfessions**: Специализации врачей (название, связь с отделением).
* **SurgeryPriceList**: Прайс-лист хирургических операций (название операции, стоимость).
* **Surgery**: Хирургические операции (связь с лечением, дата операции, результат).
* **Treatment**: Лечение пациентов (связь с заездом, врачом, результат лечения).
* **Admission**: Заезд пациента на лечение (диагноз, даты начала и окончания лечения, общая стоимость).
* **Ward**: Палаты (номер, количество мест, связь с отделением).
* **AdmissionWard**: Связь между заездом и палатой.
* **Invoice**: Счета за лечение (дата, сумма, статус оплаты).

**Реляционнная модель**



**Анализ функциональных зависимостей**

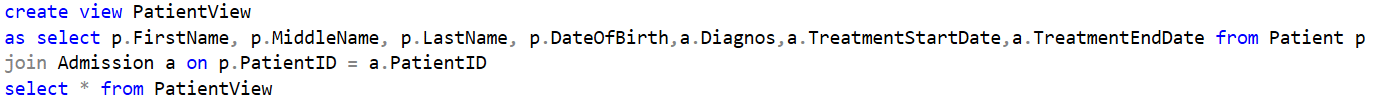
* **Patient**: PatientID → FirstName, MiddleName, LastName, Gender, DateOfBirth, PhoneNumber, Address
* **Department**: DepartmentID → DepartmentName, WardQuantity
* **Doctors**: DoctorID → DoctorFirstName, DoctorMiddleName, DoctorLastName, DoctorPhoneNumber, WorkExperienceYears, ProfessionID
* **DoctorsProfessions**: ProfessionID → ProfessionName, DepartmentID
* **SurgeryPriceList**: SurgeryName → Price
* **Surgery**: SurgeryID → TreatmentID, SurgeryName, SurgeryDate, SurgeryResult
* **Treatment**: TreatmentID → AdmissionID, DoctorID, TreatmentResult
* **Admission**: AdmissionID → PatientID, Diagnos, TreatmentStartDate, TreatmentEndDate, TotalCost
* **Ward**: WardID → DepartmentID, WardNumber, WardBedCount
* **AdmissionWard**: (AdmissionID, WardID) → (AdmissionID, WardID)
* **Invoice**: InvoiceID → AdmissionID, InvoiceDate, InvoiceAmount, PaymentStatus

**Пример заполнение БД информацией**

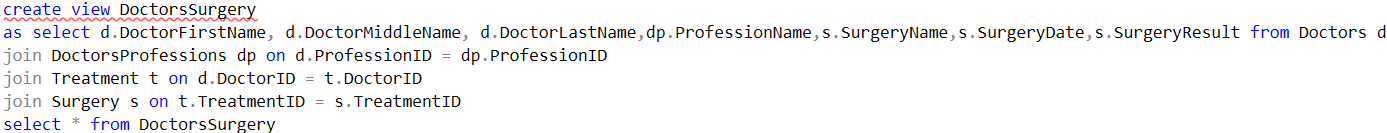


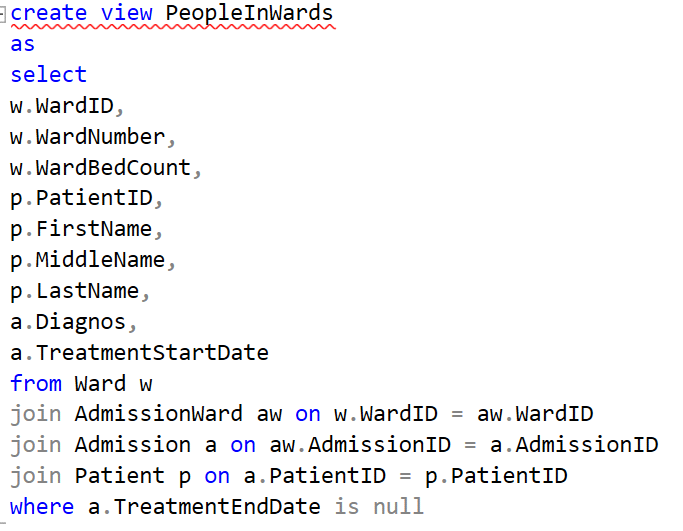
**Описание представлений**

1) Представление PatientView. Представление пациентов, всех их диагнозов и дат лечения



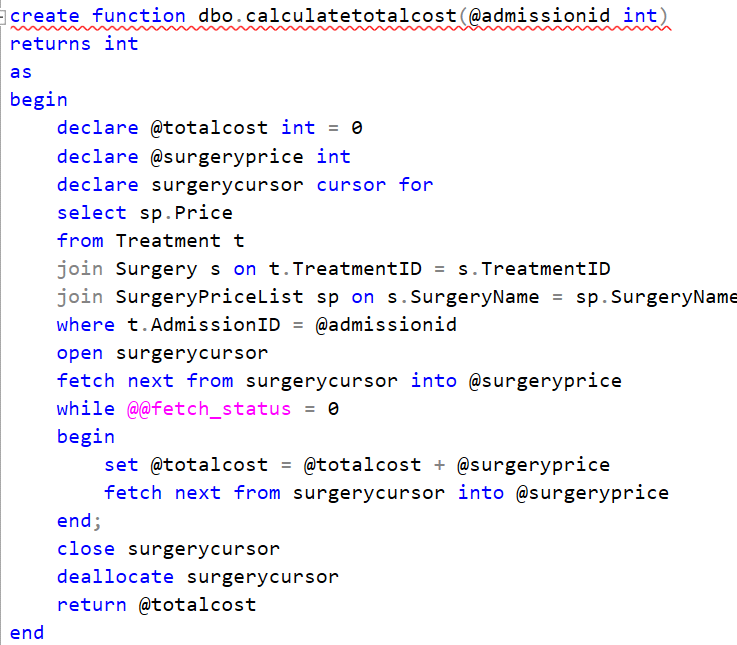
2) Представление DoctorsSurgery. Представление о врачах и всех их операциях

3) Представление PeopleInWards. Представление о палатах и людях, которые находятся в них в данный момент.

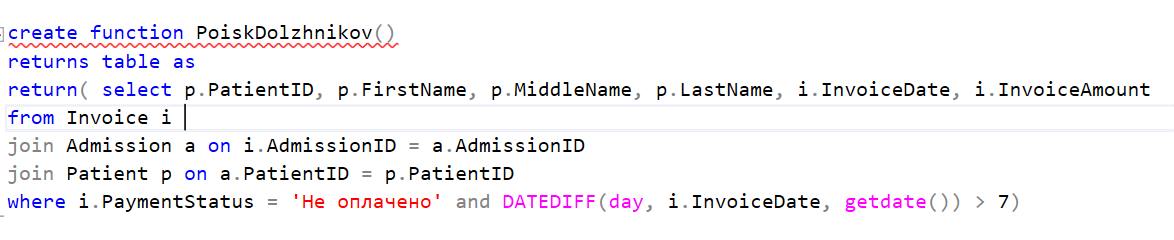


**Описание функций**

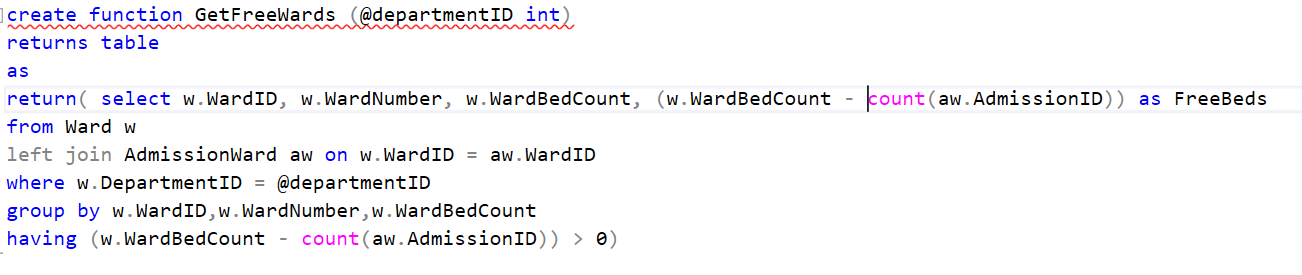
1) Функция CalculateTotalCost. Рассчитывает общую стоимость лечения на основе проведенных операций. На вход принимает @AdmissionID. Использует курсор surgerycursor. Эта функция используется в триггере.



2) Функция PoiskDolzhnikov. По политике больницы на оплату счета за лечение дается неделя. Функция выводит таблицу с информацией о людях, у которых счет за лечение не закрыт более 7 дней. Использует datediff для вычисления разницы между датой выставления счета и сегодняшней датой.

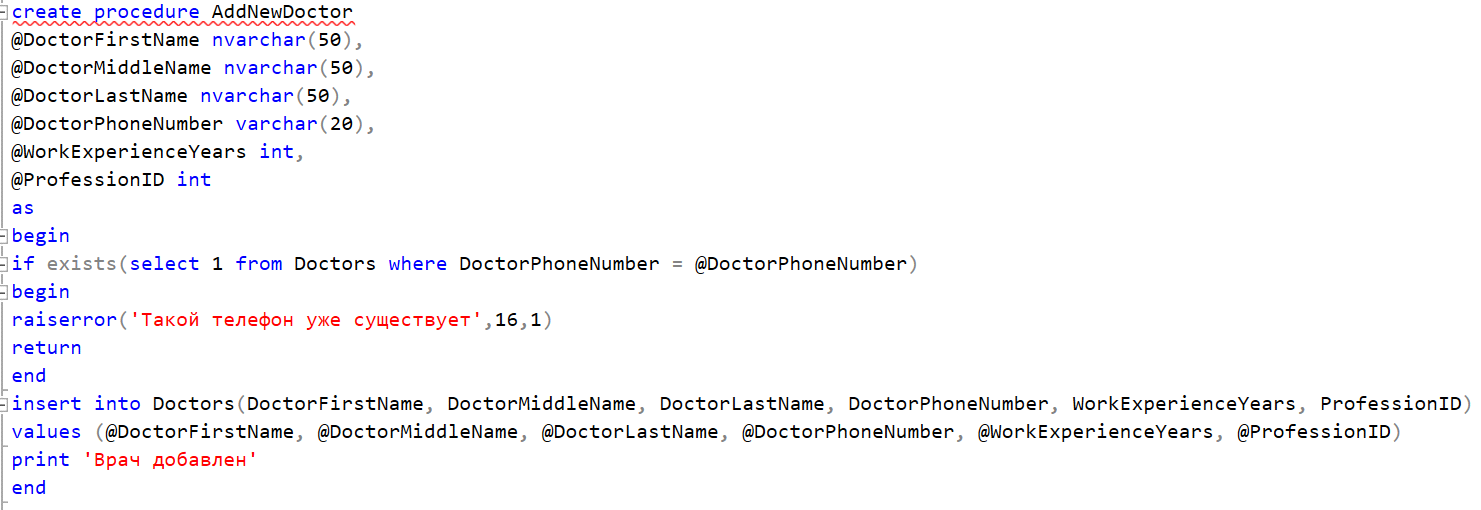


3) Функция GetFreeWards. На вход принимает @DepartmentID. Возвращает таблицу с палатами из выбранного отделения, в которых есть хотя бы 1 свободное место. Подсчет свободных мест происходит за счет таблицы AdmissionWard.



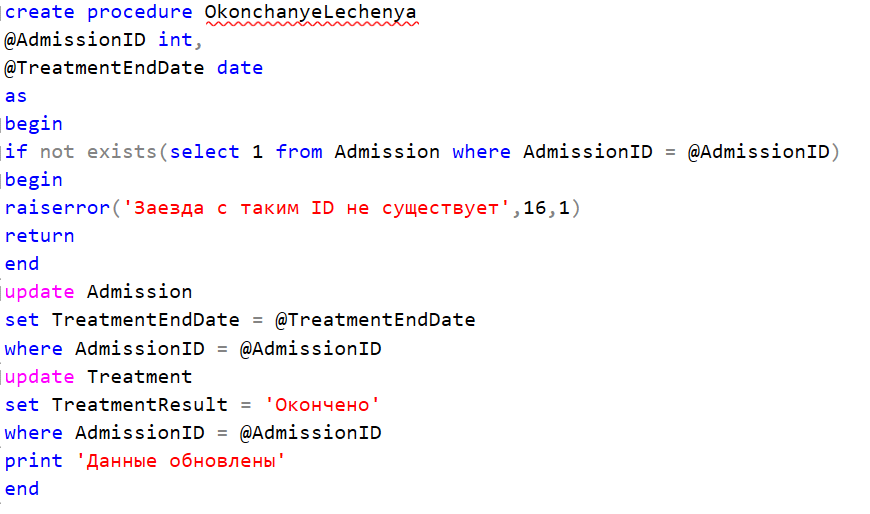
**Описание хранимых процедур**

1) Процедура AddNewDoctor. Добавляет нового врача в базу данных.

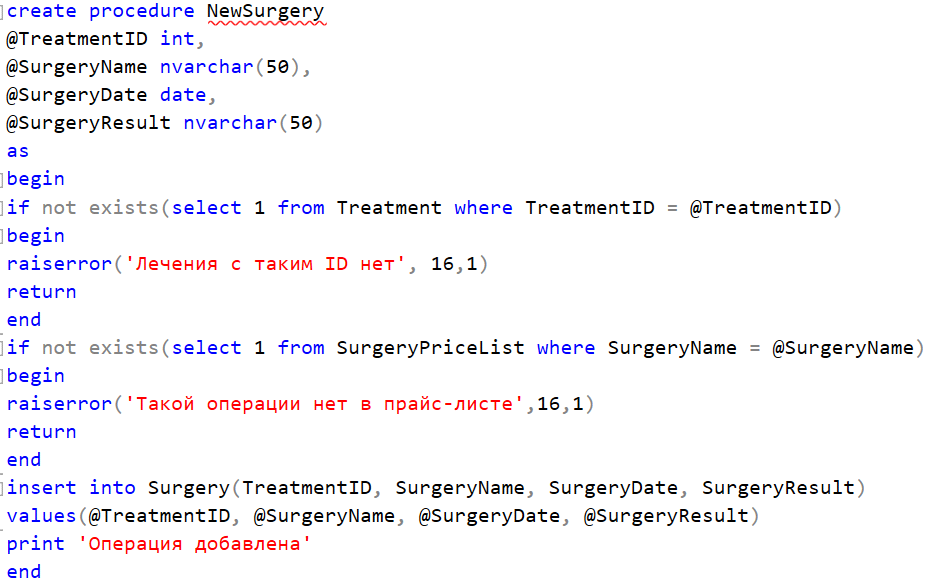


Принимает на вход личные данные, опыт работы и профессию доктора, после чего проверяет номер телефона, если доктор с таким телефоном уже есть в бд, то выбрасывает ошибку, если все хорошо, то добавляет доктора и оповещает об этом сообщением.

2) Процедура OkonchanyeLechenya. Обновляет данные в бд, заканчивая лечение в больнице.

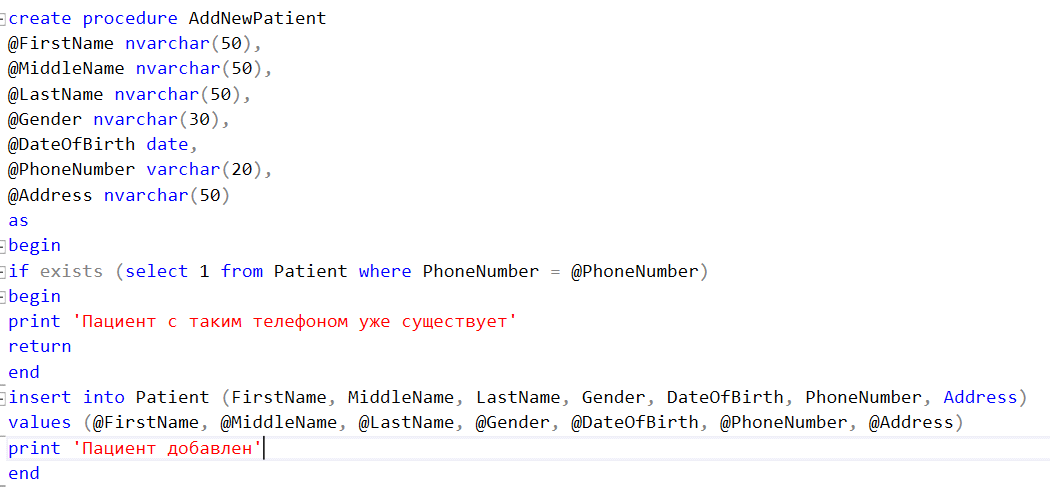
Принимает на вход ID заезда в больницу и дату окончания лечения. Делает проверку на корректность ID заезда в больницу. Если все хорошо, то в таблицу Admission выставляет дату окончания лечения, а в таблице Treatment изменяет результат лечения с ’В процессе’ на ’Окончено’. Выводит сообщение об успешном обновлении данных.

3) Процедура NewSurgery. Добавляет в бд проведенную операцию.



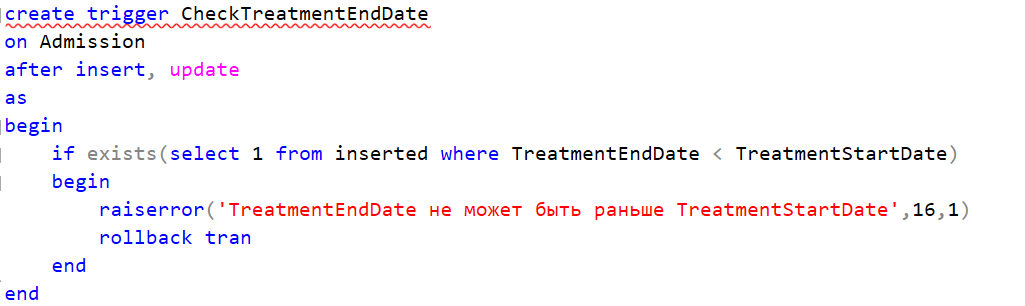
На вход принимает ID лечения, название проведенной операции, дату и результат операции. Проверяет на корректность ID лечения и названия операции. Если все хорошо, то добавляет данные и оповещает об этом в сообщении.

4) Процедура AddNewPatient. Добавляет личные данные нового пациента



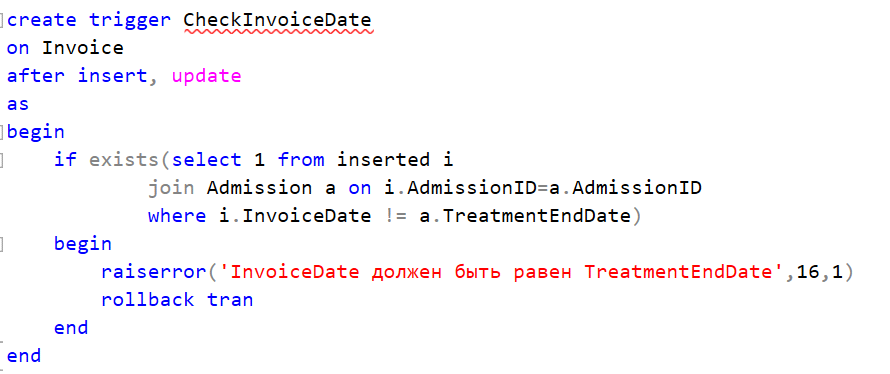
**Описание триггеров**

1) Триггер CheckTreatmentEndDate. Проверяет, что дата окончания лечения идет после даты начала лечения.

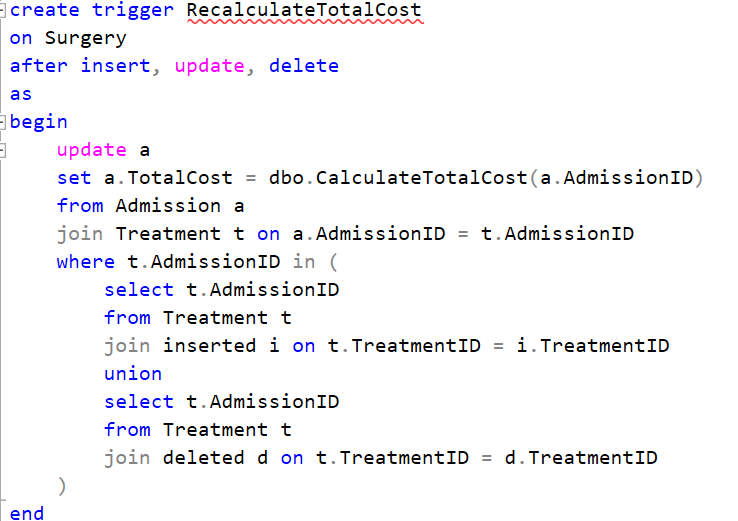


Если дата окончания до даты начала, то выбрасывает ошибку и откатывает изменения.

2) Триггер CheckInvoiceDate. Проверяет, что дата выставления счета совпадает с датой выезда из больницы.

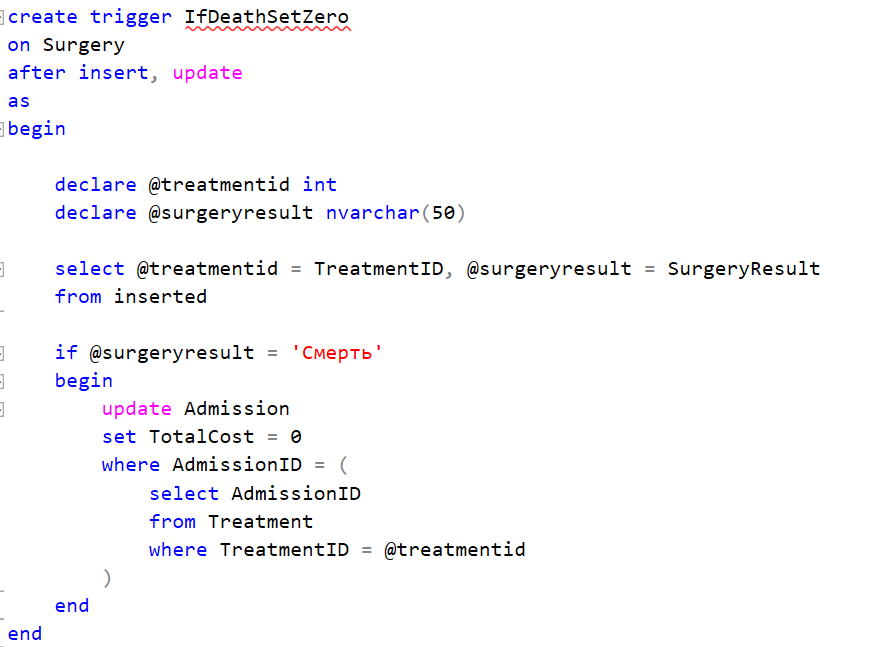
Если дата не совпадает, то выбрасывает ошибку и отменяет изменения.

3) Триггер RecalculateTotalCost. Пересчитывает TotalCost, при добавлении, изменении, удалении в таблице Surgery.



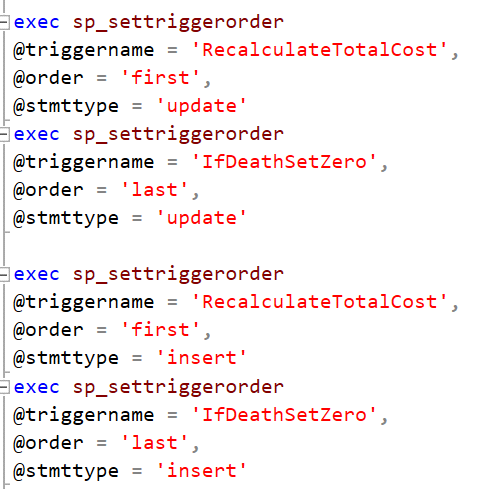
Использует уже написанную функцию CalculateTotalCost. Из inserted и deleted выбираются AdmissionID связанные с изменениями и по ним пересчитывается TotalCost.

4) Триггер IfDeathZero. По политике больницы, если пациент умер во время операции, то счет его семье не выставляется. Устанавливает TotalCost = 0, если добавленная операция была окончена смертью.

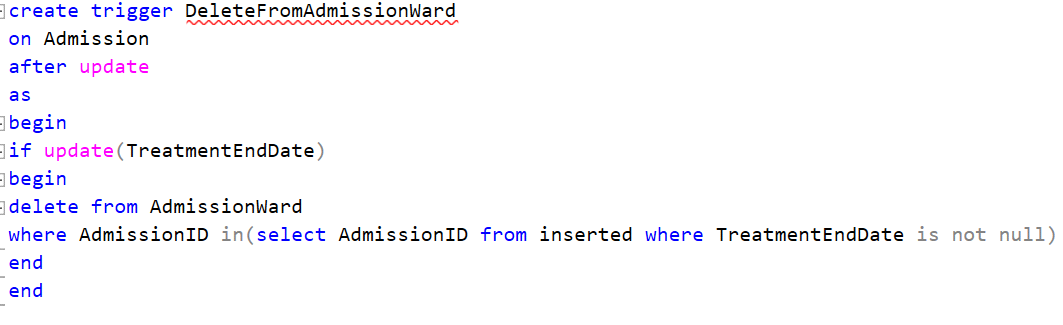


Использует таблицу inserted.

У меня 2 триггера на таблицу Surgery, поэтому я выставляю их очередность

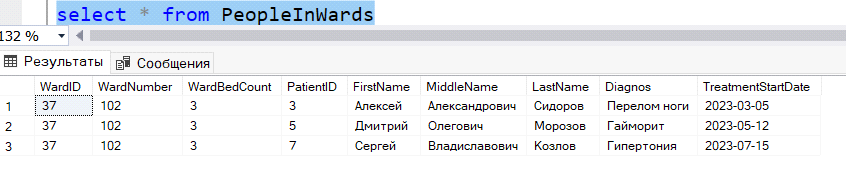


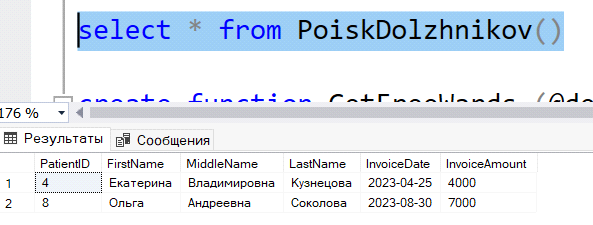
5) Триггер DeleteFromAdmissionWard. Удаляет запись о нахождении человека в палате при окончании лечения.

Когда в таблице Admission происходит обновление, а точнее, если выставляется дата окончания лечения, это означает, что человек выехал и запись о его нахождении в палате надо удалить.

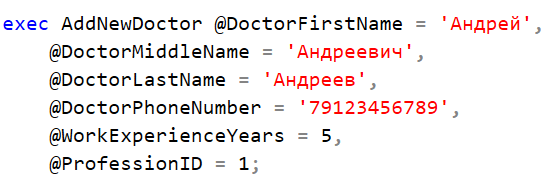
**Примеры работы с БД с использованием созданных объектов**

Представление PeopleInWards. Из представления видно, что сейчас в больнице лежит 3 человека, все в 102 палате.

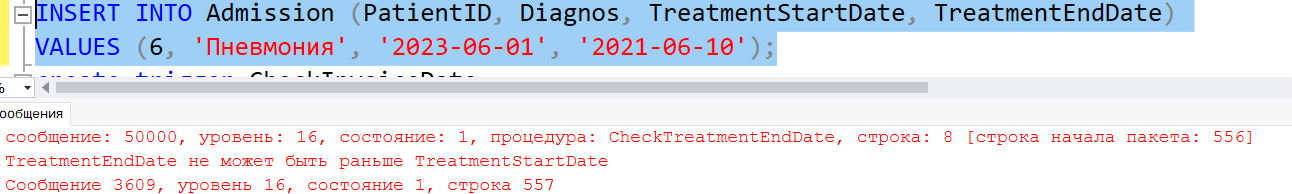
Функция PoiskDolzhnikov. При вызове функция выводит таблицу с 2 людьми, которые просрочили оплату



Процедура AddNewDoctor.



Триггер CheckTreatmentEndDate



**Список литературы**

1) Курс лекций по предмету “Базы данных”

2) Документация по Microsoft SQL - SQL Server | Microsoft Learn

3) MS SQL Server 2022 и T-SQL - METANIT.COM