**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Саратовский государственный технический университет**

**имени Гагарина Ю.А.»**

Институт прикладных информационных технологий и коммуникаций

Направление «Информационные системы и технологии»

Кафедра «Прикладные информационные технологии»

**Отчет по лабораторной работе №2**

Выполнил студент Б1-ИФСТ-31,

Песчанов Арсений Андреевич

Проверил преподаватель

Кузьмин Алексей Константинович

Саратов 2023

# Выберете для себя предметную область, которая покажется более интересной. При этом можно использовать предлагаемый ниже список предметных областей.

Автобусные перевозки

# Продумайте бизнес-процесс или часть бизнес-процесса в выбранной предметной области, для которого необходимо хранение и управление данными.

В предметной области, одним из бизнес-процессов, для которого необходимо хранение и управление данными, может быть процесс управления рейсами. Этот процесс включает в себя следующие шаги:

Регистрация рейса: в этом шаге необходимо собрать информацию о рейсе, такую как номер рейса, дата и время выезда, место назначения и другие детали.

Управление пассажирами: необходимо управлять информацией о пассажирах, которые совершают поездку. Это может включать в себя регистрацию пассажиров, выдачу посадочных талонов

Мониторинг и отчетность: в этом шаге необходимо мониторить выполнение рейсов, собирать данные о задержках, отменах, загруженности рейсов и других показателях производительности.

# **Продумайте и спроектируйте модель данных, состоящую из 3-4 связанных между собой таблиц. При этом модель должна соответствовать 3 нормальной форме.**

Таблица «Рейсы» («Flights»):

flight\_id – id рейса,

city\_out – город отбытия,

city\_in – город прибытия,

flight\_time – время рейса,

time\_out – время отбытия,

time\_in – время прибытия,

date\_out – дата отбытия,

date\_in – дата прибытия,

price – цена поездки,

bus\_name – наименование автобуса,

bus\_amount\_seats – кол-во мест в автобусе,

bus\_number – автомобильный номер автобуса,

driver\_employee\_id– внешний id сотрудника водителя (вторичный ключ);

Таблица «Билеты» («Tickets»):

ticket\_id – id билета,

ticket\_type – тип билета (child / adult),

firstname – имя пассажира,

lastname – фамилия пассажира,

patronymic – отчество пассажира,

phone – номер телефона пассажира,

document\_type – тип документа (паспорт, свидетельство о рождении и т.д.),

document\_number – номер документа,

date\_birth – дата рождения,

seat\_number – номер сидения,

flight\_id – id рейса (вторичный ключ),

seller\_id – внешний id сотрудника который продал билет;

Таблица «Сотрудники» («Employee»)

employer\_id – id сотрудника,

firstname – имя сотрудника,

lastname – фамилия сотрудника,

patronymic – отчество сотрудника,

phone – номер телефона сотрудника,

passport – серия и номер паспорта сотрудника,

Inn – ИНН сотрудника,

date\_birth – дата рождения,

place\_residence – место жительства,

place\_registration – место регистрации.

job – должность

salary - зарплата

# Максимально продумайте ограничения, которые необходимо задействовать в спроектированных таблицах. При этом ожидается, что будут задействованы хотя-бы по 1 разу все рассмотренные на занятии типы ограничений и определение default. В итоге должна поддерживаться сущностная, ссылочная и доменная целостность.

**Таблица "Рейсы" (Flights):**

***Ограничение первичного ключа***: flight\_id - уникальный идентификатор рейса.

***Ограничение NOT NULL***: flight\_id, time\_out, time\_in, date\_out, date\_in, city\_out, city\_in - эти поля должны быть заполнены для каждого рейса.

***Ограничение CHECK***: можно задать ограничение на дату и время вылета, чтобы убедиться, что они находятся в будущем.

***Ограничение DEFAULT***: можно задать значение по умолчанию для некоторых полей, например, date\_out, time\_out может иметь текущую дату и время по умолчанию.

**Таблица "Билеты" (Tickets):**

***Ограничение первичного ключа***: ticket\_id - уникальный идентификатор рейса.

***Ограничение NOT NULL***: flight\_id, firstname, lastname, date\_birth, flight\_id, document\_number, document\_type, ticket\_type

***Ограничение CHECK***: можно задать ограничение на seat\_number > 0, document\_number, date\_birth

***Ограничение DEFAULT***: можно задать значение по умолчанию для некоторых полей, например, ticket\_type.

**Таблица "Сотрудники" (Employee):**

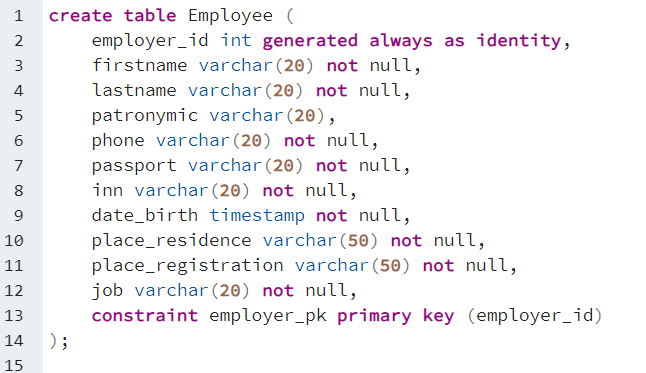
***Ограничение первичного ключа***: employer\_id - уникальный идентификатор рейса.

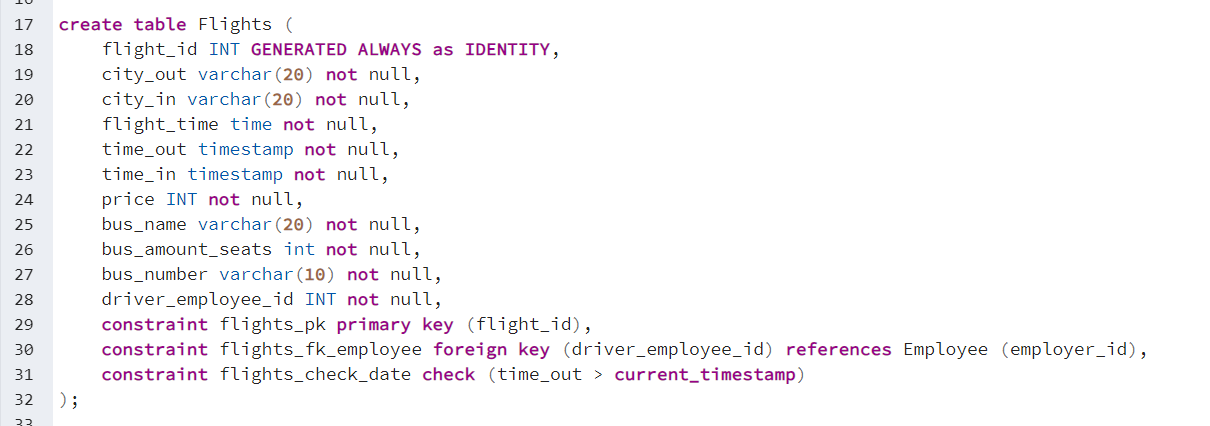
***Ограничение NOT NULL***: employer\_id, firstname, lastname, date\_birth, passport, Inn, place\_residence, place\_registration, position

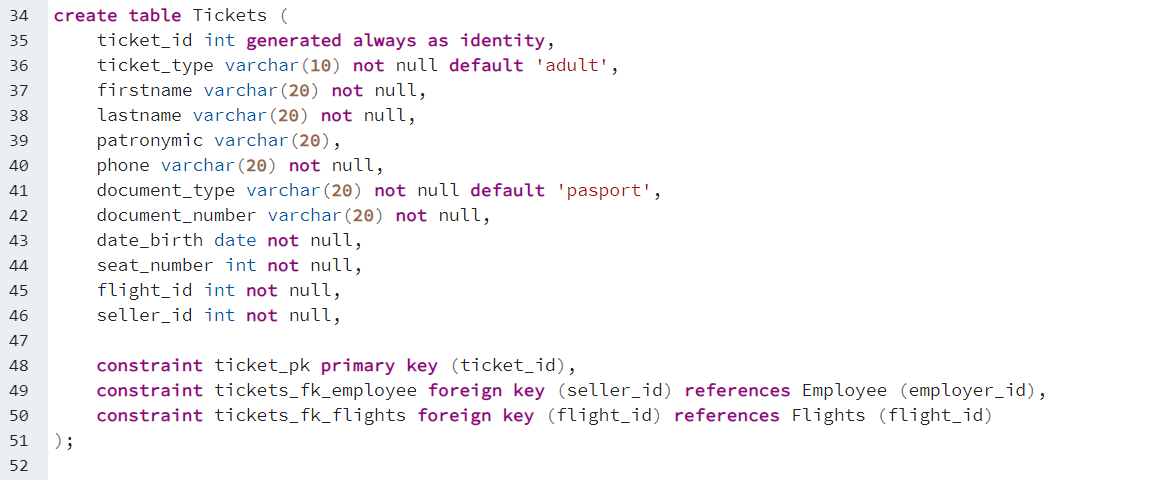
***Ограничение CHECK***: можно задать ограничение на passport, Inn, place\_residence, place\_registration.

***Ограничение DEFAULT***: -

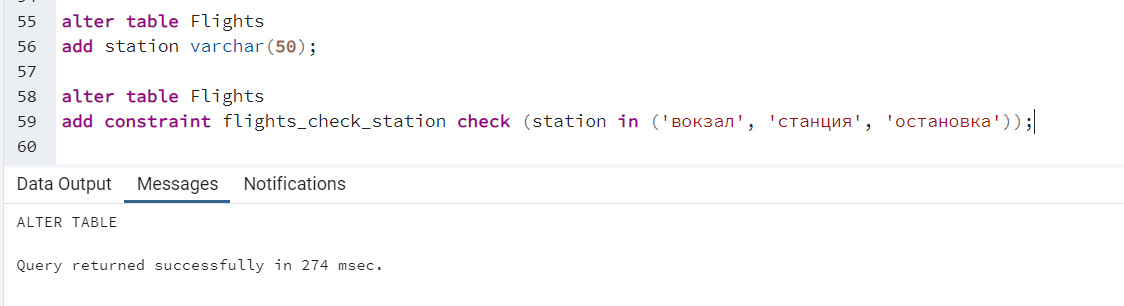
# Реализуйте спроектированные таблицы при помощи команд DDL. При этом предусмотрите счётчики для столбцов, являющимися суррогатными первичными ключами. Ограничения объявляйте как в inline, так и в out-of-line форме.

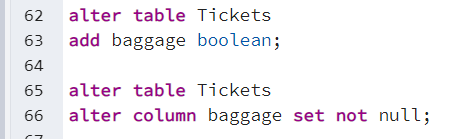


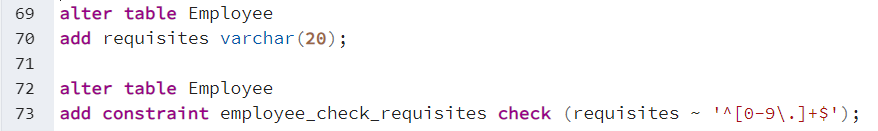




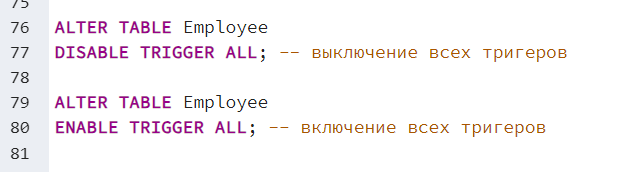
# Добавьте в созданные таблицы дополнительные столбцы при помощи ALTER TABLE. Также добавьте и дополнительные ограничения.



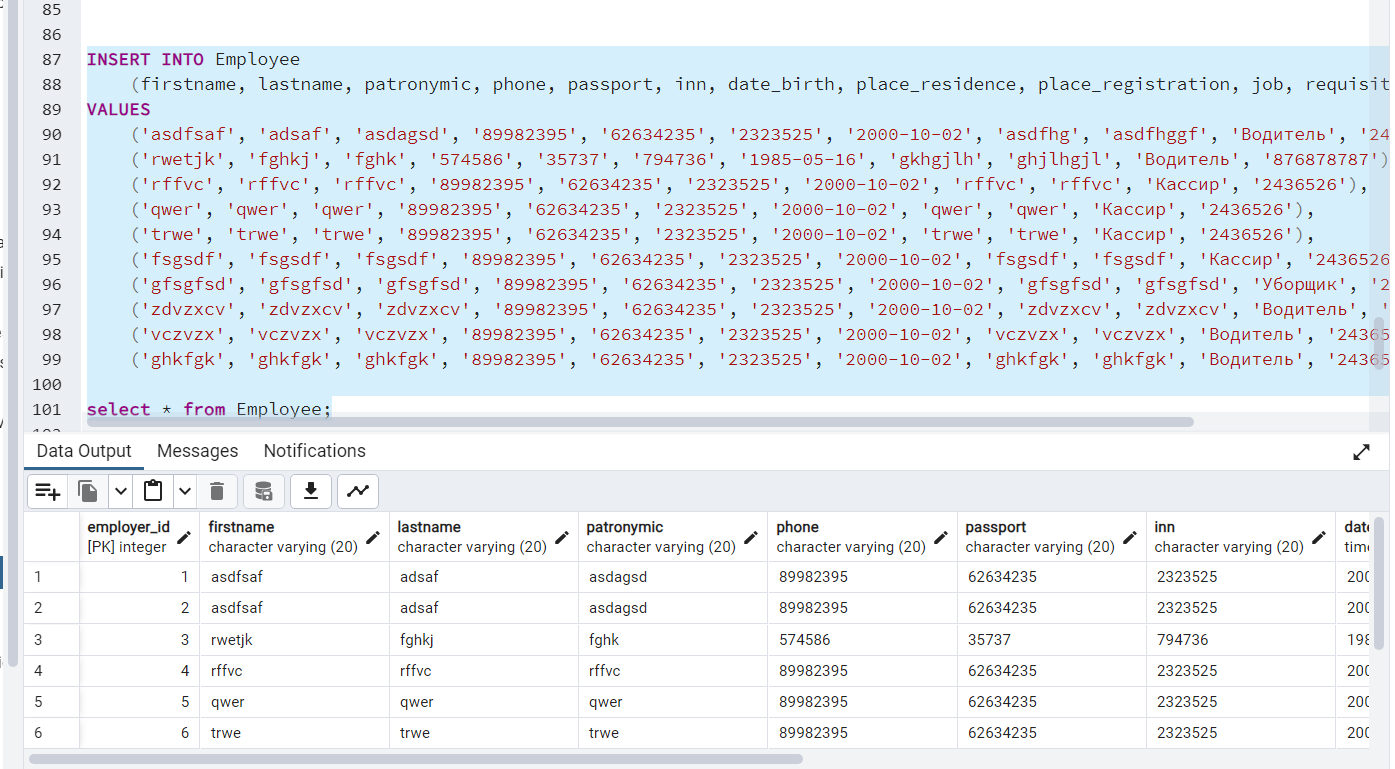


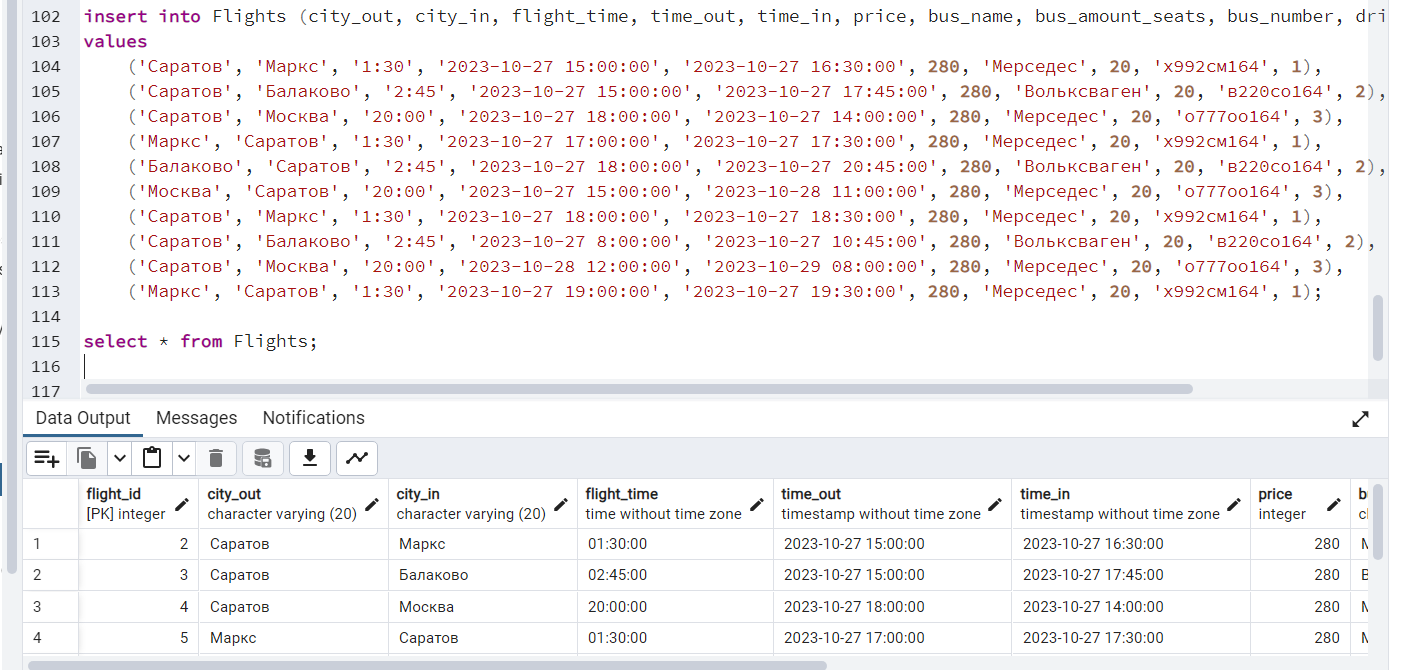


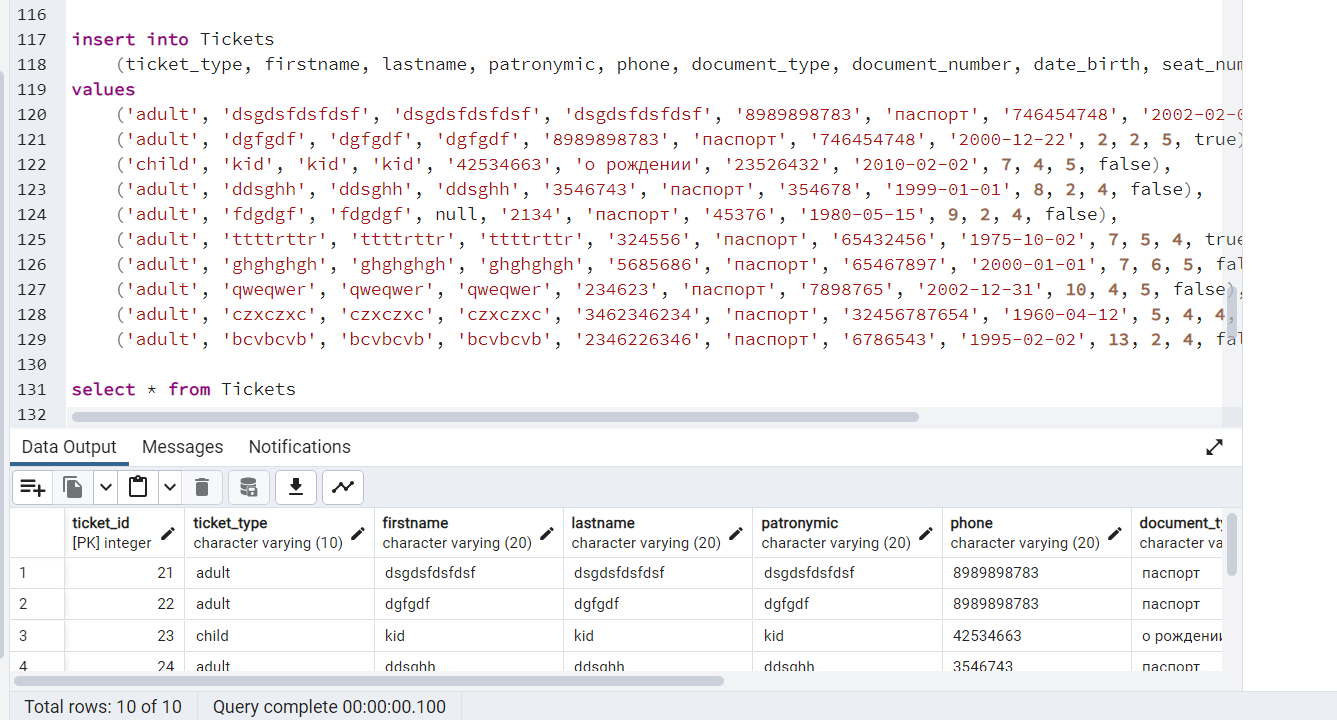
# Напишите код, который выключает какое-либо ограничение и включает его. При включении существующие записи не должны проверяться на соответствие данному ограничению.



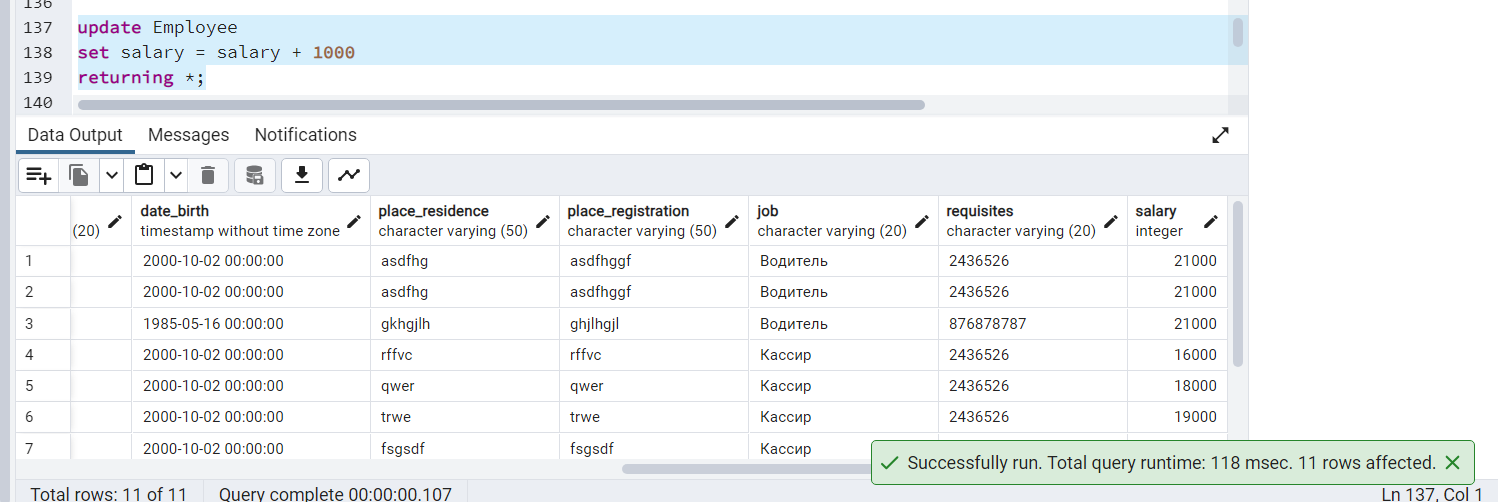
# Напишите инструкции DML, которые наполнят созданные ранее таблицы данными. Достаточно будет по 10-15 записей в каждую таблицу.







# Обновите некоторый числовой столбец для всех записей таблицы. При этом одновременно с update просчитайте статистическую информацию по проведённым обновлениям (используйте для этого RETURNING).



# Сделайте новую доп. таблицу и напишите код, который выполнит слияние вновь созданной таблицы и существующей ранее таблицы (используйте MERGE или аналог).  При этом выполняемая операция не должна носить чисто формальный характер, а должна интерпретироваться и обосновываться с точки зрения бизнес-процесса.

