# Задание на лабораторную 2

1. **Нормализовать модель БД из 1 лабораторной.**

Ответ：

**1. -Таблица "Читатели" (Readers):**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID\_Читателя (Первичный ключ)** | **Фамилия читателя** | **Имя читателя** | **Отчество читателя** | **Номер читательского билета** | **Домашний адрес читателя** | **Дата рождения читателя** |
| 1 | Иванов | Иван | Иванович | 1990-01-01 | ул. Пушкина, 1 | 12345 |
| 2 | Петров | Петр | Петрович | 1985-05-15 | ул. Толстого, 2 | 67890 |

1 НФ: Удовлетворено, в каждой ячейке содержатся только атомарные значения.

- 2 НФ: Удовлетворено, нет частичных зависимостей от первичного ключа.

- 3 НФ: Удовлетворено, нет транзитивных зависимостей.

**2. Таблица "Книги" (Books):**

|  |  |  |  |
| --- | --- | --- | --- |
| Номер\_книги (Первичный ключ) | Автор книги, которую взял на руки читатель. | Название книги | Цена книг |
| 101 | А. Пушкин | Евгений Онегин | 10.99 |
| 102 | Л. Толстой | Война и мир | 25.50 |

- 1 НФ: Удовлетворено, в каждой ячейке содержатся только атомарные значения.

- 2 НФ: Удовлетворено, нет частичных зависимостей от первичного ключа.

- 3 НФ: Удовлетворено, нет транзитивных зависимостей.

1. **Таблица "Транзакции" (Transactions):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID\_Транзакции (Первичный ключ) | ,ID\_Читателя (Внешний ключ) | Номер\_книги (Внешний ключ) | Дата\_выдачи | Дата\_возврата |
| 1 | 1 | 101 | 2023-11-10 | 2023-12-05 |
| 2 | 2 | 102 | 2023-11-15 | 2023-12-05 |

- 1 НФ: Удовлетворено, в каждой ячейке содержатся только атомарные значения.

- 2 НФ: Удовлетворено, нет частичных зависимостей от составного ключа.

- 3 НФ: Удовлетворено, нет транзитивных зависимостей.

Внешние ключевые отношения:

- Транзакции.ID\_Читателя → Читатели.ID\_Читателя

- Транзакции.Номер\_книги → Книги.Номер\_книги

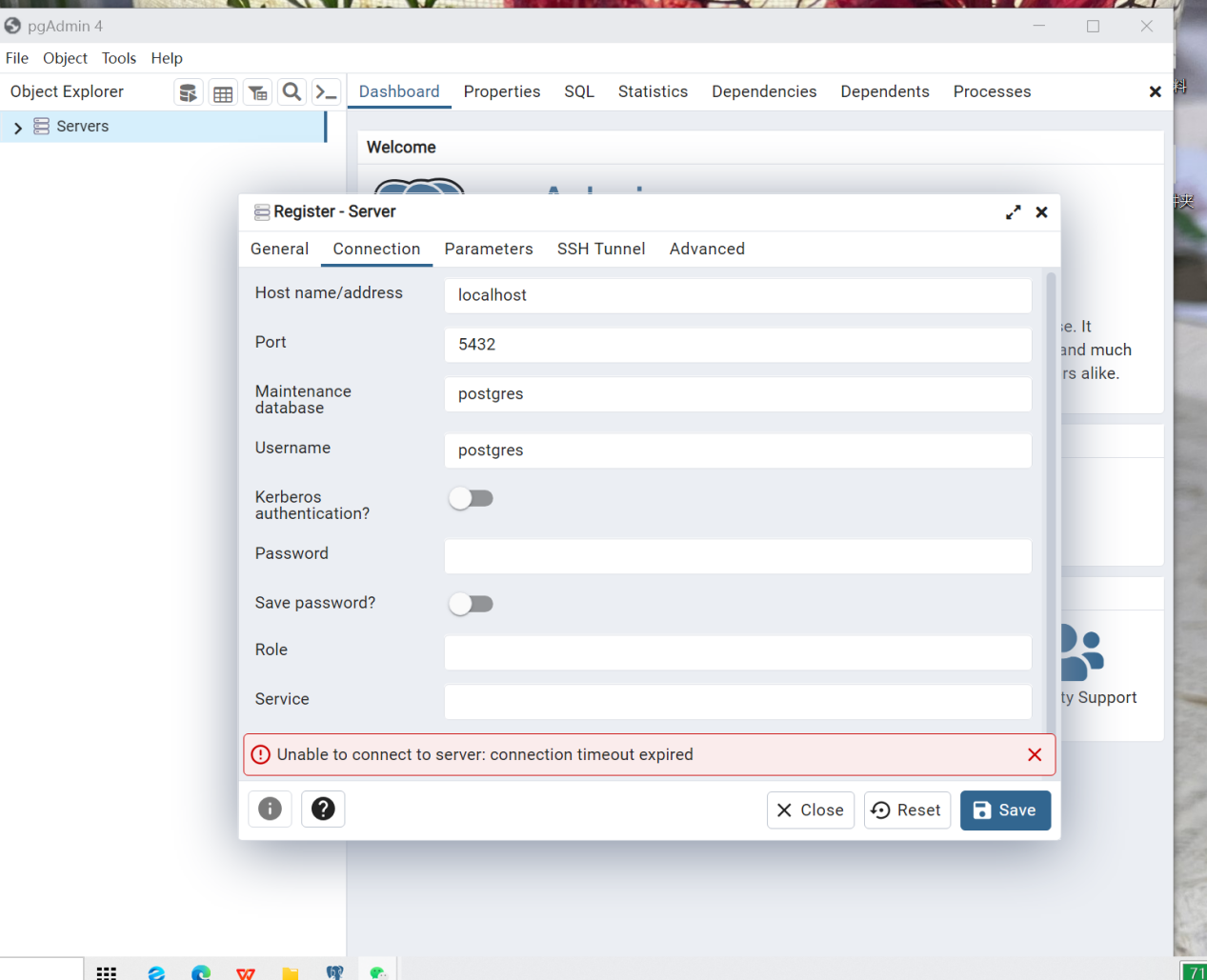
Ограничения:

- Первичный ключ (Primary Key): Каждая таблица имеет первичный ключ, используемый для уникальной идентификации каждой строки в таблице.

- Внешний ключ (Foreign Key): Используется для установления отношений между таблицами, обеспечивая целостность ссылок.

- Уникальное ограничение (Unique Constraint): Гарантирует уникальность значений в определенном столбце.

1. **Скачать и развернуть бэкап базы данных «demo».**



Здравствуйте, проблема, с которой мы сталкиваемся, показана на рисунке. Мы не можем ее создать. Мы перепробовали много дней и методов. Пожалуйста, помогите нам. Большое вам спасибо!

1. **Выполнить запросы, соответствующие вашему варианту.**

**1. Выборка всех данных из таблицы "bookings" без указания столбцов:**

SELECT \*

FROM bookings;

1. **Выборка всех данных из таблицы «flights» по столбцам: flight\_id, flight\_no, scheduled\_departure, scheduled\_arrival, departure\_airport, arrival\_airport.**

SELECT flight\_id, flight\_no, scheduled\_departure, scheduled\_arrival, departure\_airport, arrival\_airport

FROM flights;

**3. Создание любой таблицы (определить структуру новой таблицы и использовать запрос CREATE TABLE. пример для создания таблицы "example\_table"):**

CREATE TABLE example\_table (

column1 datatype1,

column2 datatype2,

column3 datatype3,

...

-- добавьте необходимые столбцы с их типами данных

);

**Замените "bookings", "flights" и "example\_table" на фактические имена таблиц в вашей базе данных, и замените `example\_table`, `column1`, `datatype1`, и так далее на фактические названия и типы столбцов для вашей новой таблицы**.