

ИИКС

Лабораторная работа №1: «Построение модели
данных»

Грущин Илья

Б21-515

2024 г

Задание

1. Выбрать предметную область для дальнейшей работы. Предметная область должна быть составлена из нескольких взаимосвязанных процессов.
2. Проанализировать выбранную предметную область и разработать систему таблиц для хранения необходимой информации в рамках реляционной базы данных. В таблице должно быть 5-8 таблиц;
3. Если модель данных не отвечает требованиям по меньшей мере третьей нормальной формы, нормализовать её. Доказать, что требования третьей нормальной формы соблюдены
4. Построить диаграмму отношений сущностей (Entity Relations Diagram; ERD) для разработанной модели данных;

Реализовать разработанную схему данных средствами языка SQL в системе управления базами данных SQLite3.

Предметная область: Видеоигры

В данной лабораторной работе рассматривается организация, занимающаяся систематизацией и управлением данными о видеоиграх, их разработчиках, издателях, жанрах, а также пользовательских рейтингах. Примерами могут выступать игровые журналисты или работники ИРИ, определяющие кому выделить денег на новую «Смуту».

Основная деятельность данной организации заключается в следующем:

- Хранение и управление информацией о видеоиграх, включая их название, год выпуска, возрастной рейтинг и пиковое количество игроков.
- Управление данными о разработчиках и издателях видеоигр.
- Классификация видеоигр по жанрам.
- Управление информацией о пользователях, включая их возраст и пол.
- Хранение и управление пользовательскими рейтингами и отзывами на видеоигры.

Спецификация таблиц

Таблица Developers

- developer_id: уникальный идентификатор разработчика.
- name: имя разработчика.
- employee_count — количество сотрудников.
- games_released — количество выпущенных игр.

Таблица Publishers

- publisher_id: уникальный идентификатор издателя.
- name: имя издателя.
- games_published — количество изданных игр.
- average_annual_revenue — среднегодовой доход.

Таблица Genres

- genre_id: уникальный идентификатор жанра.
- genre_name: название жанра.

Таблица Games

- game_id: уникальный идентификатор игры.
- title: название игры.
- release_year: год выпуска игры.
- developer_id: ссылка на разработчика игры.
- publisher_id: ссылка на издателя игры.
- age_rating: возрастной рейтинг игры.
- peak_players: пиковое количество игроков в день.

Таблица GameGenres

- game_id: ссылка на игру.
- genre_id: ссылка на жанр.
- (game_id, genre_id): составной первичный ключ, обеспечивающий уникальность комбинации игра-жанр.

Таблица Users

- user_id: уникальный идентификатор пользователя.
- username: имя пользователя.

- email: электронная почта пользователя.
- age: возраст пользователя.
- gender: пол пользователя.

Таблица Ratings

- rating_id: уникальный идентификатор рейтинга.
- game_id: ссылка на игру.
- user_id: ссылка на пользователя.
- score: оценка игры.
- review: текст отзыва.
- rating_date: дата выставления рейтинга.

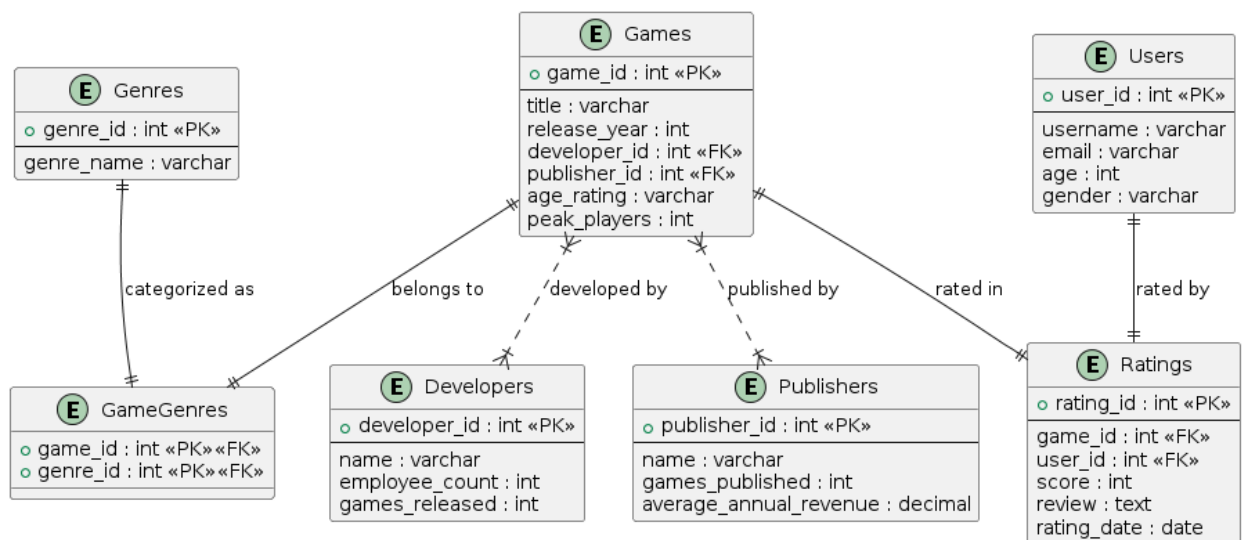
Доказательство соответствия третьей нормальной форме (3NF)

Первая нормальная форма (1NF): Все атрибуты атомарны. Все таблицы содержат только атомарные (неделимые) значения.

Вторая нормальная форма (2NF): Все неключевые атрибуты зависят от всего первичного ключа. Все таблицы либо имеют одинарный первичный ключ, либо составной ключ, как в таблице GameGenres, где атрибуты зависят от обоих ключевых атрибутов.

Третья нормальная форма (3NF): Все неключевые атрибуты не зависят транзитивно от первичного ключа. В наших таблицах нет атрибутов, которые зависят от других неключевых атрибутов.

ERD



Заключение

В ходе данной лабораторной работы была разработана реляционная база данных для хранения информации о видеоиграх, их разработчиках, издателях, жанрах и пользовательских рейтингах. Создано семь таблиц, каждая из которых соответствует требованиям третьей нормальной формы. С помощью диаграммы отношений сущностей (ERD) была визуализирована структура базы данных и взаимосвязи между таблицами. Также был разработан SQL-сценарий (см. Приложение) для создания таблиц. Результат – указанный сценарий, диаграмма отношений сущностей созданная в PostgreSQL база данных (см. рисунки ниже)

The screenshot displays the PostgreSQL Enterprise Console interface. On the left, the 'Servers' tree shows the 'db_lab' database. The 'public' schema is expanded, showing various database objects. The 'Tables (7)' section is highlighted, listing the following tables: developers, games, publishers, gamegenres, genres, ratings, and users. The 'gamegenres' table is further expanded, showing its constraints: gamegenres_game_id_fkey, gamegenres_genre_id_fkey, and gamegenres_pkey. On the right, the 'Query' tab is active, showing a SQL query: `SELECT table_name FROM information_schema.tables WHERE table_schema = 'public';`. The 'Data Output' tab shows the results of this query, listing the table names in a table format.

table_name
1 developers
2 games
3 publishers
4 gamegenres
5 genres
6 ratings
7 users

Total rows: 7 of 7 Query complete 00:00:00.049

db_lab/postgres@PostgreSQL 16

No limit

E

i.

Query Query History

1 SELECT * FROM Games;

2

3

Data Output Messages Notifications

	game_id [PK] integer	title character varying (100)	release_year integer	developer_id integer	publisher_id integer	age_rating character varying (10)	peak_players integer
--	-------------------------	----------------------------------	-------------------------	-------------------------	-------------------------	--------------------------------------	-------------------------

Приложение

Файл create.sql, реализующий создание таблицы.

```
CREATE TABLE Developers (  
    developer_id SERIAL PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    employee_count INT,  
    games_released INT  
);  
  
CREATE TABLE Publishers (  
    publisher_id SERIAL PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    games_published INT,  
    average_annual_revenue DECIMAL(15, 2)  
);  
  
CREATE TABLE Genres (  
    genre_id SERIAL PRIMARY KEY,  
    genre_name VARCHAR(100) NOT NULL  
);  
  
CREATE TABLE Games (  
    game_id SERIAL PRIMARY KEY,  
    title VARCHAR(100) NOT NULL,  
    release_year INT,  
    developer_id INT REFERENCES Developers(developer_id),  
    publisher_id INT REFERENCES Publishers(publisher_id),  
    age_rating VARCHAR(10),  
    peak_players INT  
);  
  
CREATE TABLE GameGenres (  
    game_id INT,  
    genre_id INT,  
    PRIMARY KEY (game_id, genre_id),  
    FOREIGN KEY (game_id) REFERENCES Games(game_id),  
    FOREIGN KEY (genre_id) REFERENCES Genres(genre_id)  
);  
  
CREATE TABLE Users (  
    user_id SERIAL PRIMARY KEY,  
    username VARCHAR(100) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    age INT,  
    gender VARCHAR(10)  
);  
  
CREATE TABLE Ratings (  
    rating_id SERIAL PRIMARY KEY,
```



```
game_id INT REFERENCES Games(game_id),  
user_id INT REFERENCES Users(user_id),  
score INT,  
review TEXT,  
rating_date DATE  
);
```