ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  |  |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1 |
| Разработка концептуальной модели предметной области |
| по дисциплине: ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4131 |  | 30.09.2023 |  | В.А.Алексеев |
|  |  |  | подпись, дата |  | инициалы, фамилия |

1. **Задание**

Создать физическую модель базы данных, находящуюся в третьей нормальной форме в соответствии с заданным вариантом. Расписать ссылочную целостность БД в таблице.

Формат описания ссылочной целостности:

-Дочерняя таблица;

-Столбцы, составляющие внешний ключ;

-Родительская таблица;

-Наименование ссылочной целостности при удалении;

-Описание действий по поддержанию ссылочной целостности при удалении;

-Наименование ссылочной целостности при обновлении;

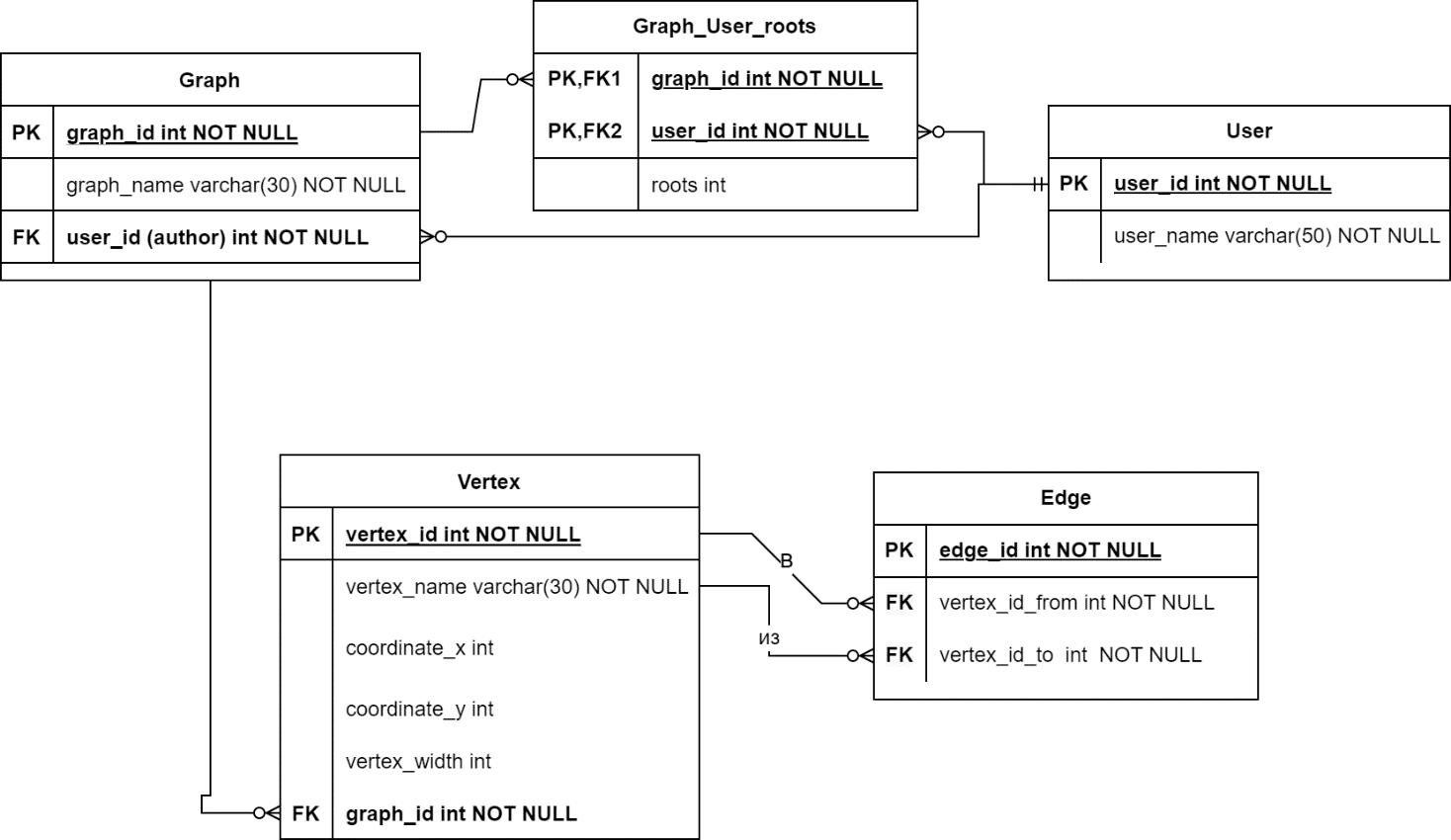
-Описание действий по поддержанию ссылочной целостности при обновлении;

-Обоснование выбора типа поддержки ссылочной целостности

1. **Таблица целостности**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Дочер няя таблица (с внешн им ключом) | Внеш ний ключ | Родите льская таблица | Как поддерживается ссылочная целостность при удалении | Описание ссылочной целостнос ти при удалении | Как поддерж ивается ссылочная целостность при обновлении | Описание ссылочной целостности при обновлении | Обоснование |
| Graph | **user\_id (author) int** | User | Каскадируется | При удалении данных из User, удалятся все связанные данные из Graph | Каскадируется | При обновлении первичного ключа User, обновится внешний ключ из Graph, который не имеет связанных с ним данных | Граф не может быть без автора и существовать сам по себе. При этом БД должна поддерживать удаление любых пользователей. |
| Graph\_User\_roots | **user\_id int** | User | Каскадируется | При удалении данных из User, удалятся все связанные данные из Graph\_User\_roots | Каскадируется | При обновлении первичного ключа User, обновится внешний ключ из Graph\_User\_roots | Без пользователя не может быть и прав по отношению к графу. При этом БД должна поддерживать удаление любых пользователей. |
| Graph\_User\_roots | **graph\_id int** | Graph | Каскадируется | При удалении данных из Graph, удалятся все связанные данные из Graph\_User\_roots | Каскадируется | При обновлении первичного ключа Graph, обновится внешний ключ из Graph\_User\_roots | Без графа не может быть и прав по отношению к нему. При этом БД должна поддерживать удаление любых графов. |
| Vertex | **graph\_id int** | Graph | Каскадируется | При удалении данных из Graph, удалятся все связанные данные из Vertex | Каскадируется | При обновлении первичного ключа Graph, обновится внешний ключ из Vertex,который не имеет связанных с ним данных | Вершина не может существовать без графа. При этом БД должна поддерживать удаление любых графов. |
| Vertex | **vertex\_id\_from int** | Edge | Каскадируется | При удалении данных из Vertex, удалятся все связанные данные из Edge | Каскадируется | При обновлении первичного ключа Vertex, обновится внешний ключ из Edge | Ребро не может существовать без вершины. При этом БД должна поддерживать удаление любых вершин. |
| Vertex | **vertex\_id\_to int** | Edge | Каскадируется | При удалении данных из Vertex, удалятся все связанные данные из Edge | Каскадируется | При обновлении первичного ключа Vertex, обновится внешний ключ из Edge | Ребро не может существовать без вершины. При этом БД должна поддерживать удаление любых вершин. |

1. **Физическая модель БД**



1. **Концептуальная схема предметной области**

