**//Проверить везде подписи цифр к картинкам**

**2. Схема базы данных**

Реляционная модель

Реляционная модель данных — это способ рассмотрения данных, то есть предписание для способа представления данных (посредством таблиц) и для способа работы с таким представлением (посредством операторов). Она связана с тремя аспектами данных: структурой (объекты), целостностью и обработкой данных (операторы).

В реляционной модели, в отличие от иерархической или сетевой, не существует физических отношений. Вся информация хранится в виде таблиц (отношений), состоящих из рядов и столбцов. А данные двух таблиц связаны общими столбцами, а не физическими ссылками или указателями. Для манипуляций с рядами данных существуют специальные операторы.

В отличие от двух других типов СУБД, в реляционных моделях данных нет необходимости просматривать все указатели, что облегчает выполнение запросов на выборку информации по сравнению с сетевыми и иерархическими СУБД. Это одна из основных причин, почему реляционная модель оказалась более удобна. В реляционной модели, как объекты, так и их отношения представлены только таблицами, и ничем более.

Распространённые реляционные СУБД: Oracle, Sybase, DB2, Ingres, Informix и MS-SQL Server.

Для создания базы данных был выбран Microsoft SQL Server 2019.

Модель базы данных — тип модели данных, которая определяет логическую структуру базы данных и принципиально определяет, каким образом данные могут быть сохранены, организованы и обработаны.

Текущая база данных состоит из одиннадцати таблиц, ее структура представлена на рисунке 2.1.

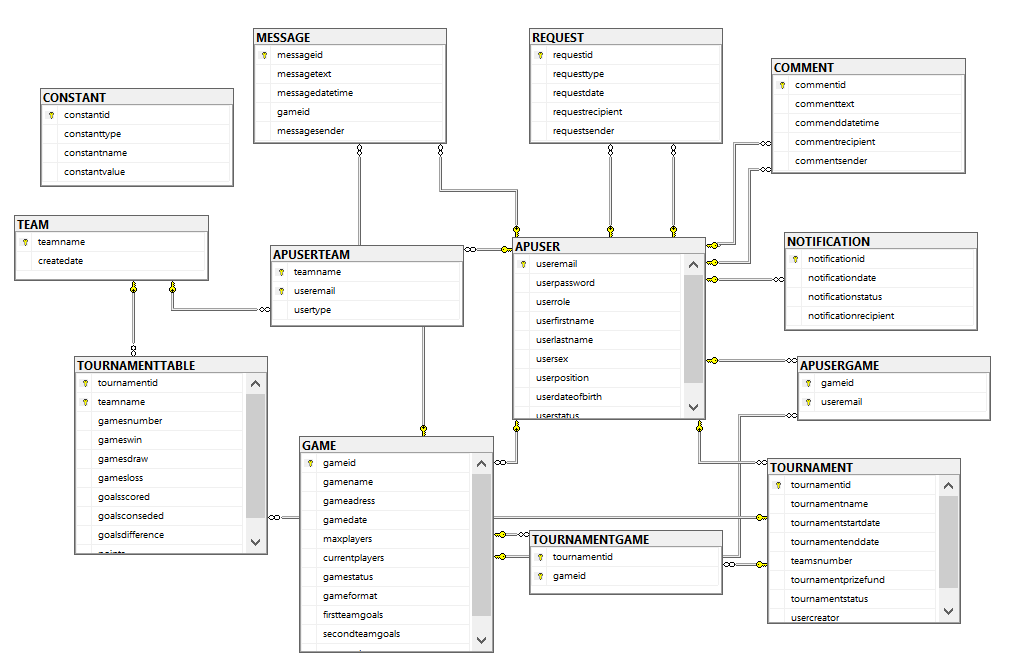


Рисунок 2.1 – Структура базы данных

На рисунке 2.2 представлена структура таблица APUSER, которая содержит информацию о пользователях приложения:

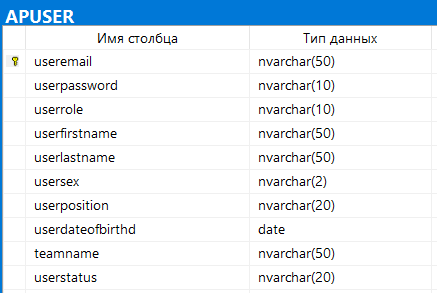


Рисунок 2.1 – Структура таблицы APUSER

Таблица содержит следующие поля:

* *USEREMAIL* – почтовый адрес пользователя – ключевое поле;
* *USERPASSWORD* – пароль от учетной записи пользователя;
* *USERROLE* – роль пользователя в приложении;
* *USERFIRSTNAME* – имя пользователя;
* *USERLASTNAME* – фамилия пользователя;
* *USERSEX* – пол пользователя;
* *USERPOSITION* – игровая позиция пользователя;
* *USERDATEOFBIRTH* – дата рождения пользователя;
* *TEAMNAME* – наименование футбольной команды пользователя;
* *USERSTATUS* – статус учетной записи пользователя;

На рисунке 2.3 представлена структура таблица CONSTANT, которая содержит константные значения, используемые в приложении:

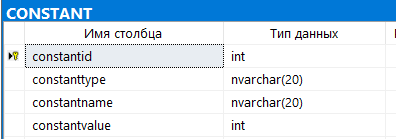


Рисунок 2.1 – Структура таблицы CONSTANT

Таблица содержит следующие поля:

* *CONSTANTID* – идентификатор константного значения – ключевое поле;
* *CONSTANTTYPE* – тип константного значения;
* *CONSTANTNAME* – строковое представление константного значения;
* *CONSTANTVALUE* – целочисленное представление;

На рисунке 2.3 представлена структура таблица MESSAGE, которая предназначена для хранения сообщений пользователей:

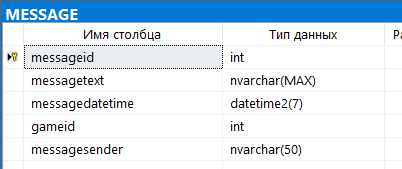


Рисунок 2.1 – Структура таблицы MESSAGE

Таблица содержит следующие поля:

* *MESSAGEID* – идентификатор сообщения – ключевое поле;
* *MESSAGETEXT* – содержимое отправляемого сообщения;
* *MESSAGEDATETIME* – время отправки сообщения;
* *GAMEID* – идентификатор чата футбольного матча в который отправляются сообщения пользователями;
* *MESSAGESENDER* – идентификатор пользователя, который является отправителем сообщения;

На рисунке 2.3 представлена структура таблица COMMENT, которая предназначена для хранения комментариев, оставляемых в профиле пользователя:

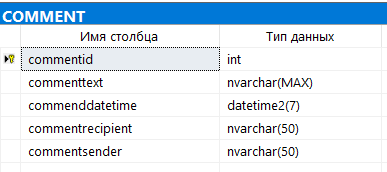


Рисунок 2.1 – Структура таблицы COMMENT

Таблица содержит следующие поля:

* *COMMENTEID* – идентификатор комментария – ключевое поле;
* *COMMENTTEXT* – содержимое отправляемого комментария;
* *COMMENTDATETIME* – время отправки комментария;
* *COMMENTRECIPIENT* – идентификатор пользователя, в профиле которого оставляют комментарий – внешний ключ к таблице APUSER;
* *COMMENTSENDER* – идентификатор пользователя, который является отправителем комментария – внешний ключ к таблице APUSER;

На рисунке 2.3 представлена структура таблица REQUEST, которая предназначена для хранения запросов/приглашений на участие в матче:

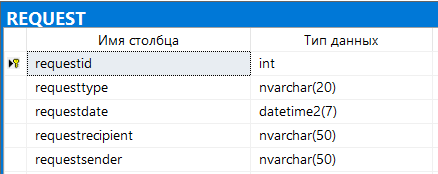


Рисунок 2.1 – Структура таблицы REQUEST

Таблица содержит следующие поля:

* *REQUESTID* – идентификатор запрос – ключевое поле;
* *REQUESTTYPE* – тип запроса;
* *REQUESTDATE*– время отправки запроса;
* *REQUESTRECIPIENT* – идентификатор пользователя, который является получателем запроса – внешний ключ к таблице APUSER;
* *REQUESTSENDER* – идентификатор пользователя, который является отправителем запроса – внешний ключ к таблице APUSER;

На рисунке 2.3 представлена структура таблица NOTIFICATION, которая предназначена для хранения уведомлений пользователя:

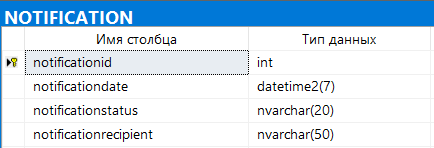


Рисунок 2.1 – Структура таблицы NOTIFICATION

Таблица содержит следующие поля:

* *NOTIFICATIONID* – идентификатор уведомления – ключевое поле;
* *NOTIFICATIONDATE* – время отправки уведомления;
* *NOTIFICATIONSTATUS* – состояние уведомления;
* *NOTIFICATIONRECIPIENT* – идентификатор пользователя, который является получателем уведомления – внешний ключ к таблице APUSER;

На рисунке 2.3 представлена структура таблица GAME, которая предназначена для хранения информации о футбольных матчах, создаваемых пользователями:

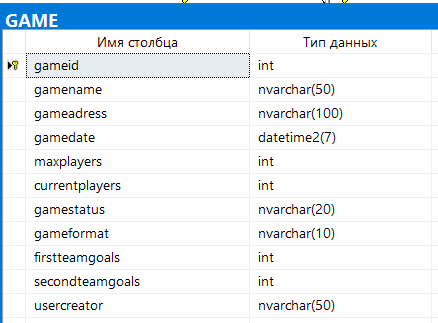


Рисунок 2.1 – Структура таблицы GAME

Таблица содержит следующие поля:

* *GAMEID* – идентификатор футбольного матча – ключевое поле;
* *GAMENAME* – наименование футбольного матча;
* *GAMEADRESS* – адрес проведения футбольного матча;
* *GAMEDATE* – дата и время проведения футбольного матча;
* *MAXPLAYERS* – максимальное количество участников футбольного матча;
* *CURRENTPLAYERS* – текущее количество участников футбольного матча;
* *GAMESTATUS* – текущий статус футбольного матча;
* *GAMEFORMAT* – формат футбольного матча;
* *FIRSTTEAMGOALS* – количество голов, забитых первой командой;
* *SECONDTEAMGOALS* – количество голов, забитых второй командой;
* *USERCREATOR* – идентификатор пользователя, который является создателем футбольного матча – внешний ключ к таблице APUSER;

На рисунке 2.3 представлена структура таблица APUSERGAME, которая предназначена для организации связи многие ко многим между таблицами APUSER и GAME:

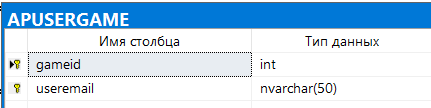


Рисунок 2.1 – Структура таблицы APUSERGAME

Таблица содержит следующие поля:

* *GAMEID* – идентификатор футбольного матча – первая часть составного ключевого поля – внешний ключ к таблице GAME;
* *USEREMAIL* – почтовый адрес пользователя вторая часть составного ключевого поля – внешний ключ к таблице APUSER;

На рисунке 2.3 представлена структура таблицы TOURNAMENT, которая предназначена для хранения информации о футбольных турнирах, создаваемых пользователями:

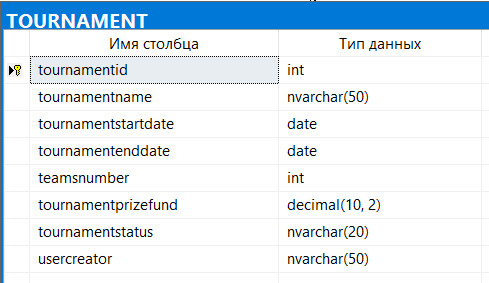


Рисунок 2.1 – Структура таблицы TOURNAMENT

Таблица содержит следующие поля:

* *TOURNAMENTID* – идентификатор футбольного турнира – ключевое поле;
* *TOURNAMENTNAME* – наименование футбольного турнира;
* *TOURNAMENTSTARTDATE* – дата и время начала футбольного турнира;
* *TOURNAMENTENDDATE* – дата и время окончания футбольного турнира;
* *TEAMSNUMBER* – количество команд участников футбольного турнира;
* *TOURNAMENTPRIZEFUND* – призовой фонд турнира;
* *TOURNAMENTSTATUS* – статус футбольного турнира;
* *USERCREATOR* – идентификатор пользователя, который является создателем футбольного турнира – внешний ключ к таблице APUSER;
* На рисунке 2.3 представлена структура таблицы TOURNAMENTGAME, которая предназначена для организации связи многие ко многим между таблицами TOURNAMENT и GAME:

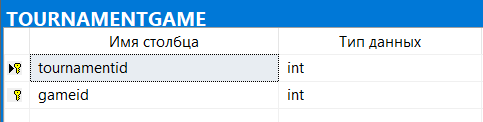


Рисунок 2.1 – Структура таблицы TOURNAMENT

Таблица содержит следующие поля:

* *TOURNAMENTID* – идентификатор футбольного турнира – вторая часть составного ключевого поля – внешний ключ к таблице GAME;
* *GAMEID* – идентификатор футбольного матча – первая часть составного ключевого поля – внешний ключ к таблице GAME;

На рисунке 2.3 представлена структура таблицы TOURNAMENTTABLE, которая предназначена для хранения информации о турнирных таблицах футбольных турниров, создаваемых пользователями:

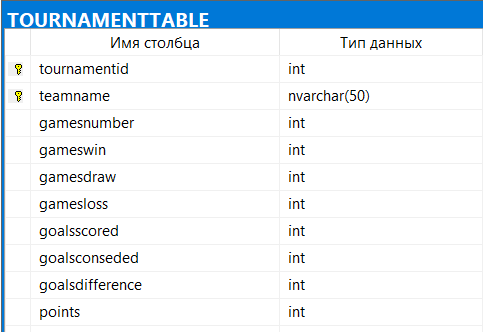


Рисунок 2.1 – Структура таблицы TOURNAMENTTABLE

Таблица содержит следующие поля:

* *TOURNAMENTID* – идентификатор футбольного турнира – ключевое поле;
* *TEAMNAME* – наименование футбольной команды – внешний ключ к таблице TEAM;
* *GAMESNUMBER* – количество игр, которое сыграла команда на турнире;
* *GAMESWIN* – количество игр, которые завершились победой команды на турнире;
* *GAMESNUMBER* – количество игр, которое завершилось ничейным результат для команды на турнире;
* *GAMESLOSS* – количество игр, которое завершилось поражением команды на турнире;
* *GOALSSCORED* – количество голов, забитых командой на турнире;
* *GOALSCONSEDED* – количество голов, пропущенных командой на турнире;
* *GOALSDIFFERENCE* – разница между пропущенными и забитыми голами.
* *POINTS* – количество очков, которое команда набрала на турнире.

На рисунке 2.3 представлена структура таблицы TEAM, которая предназначена для хранения информации о футбольных командах:

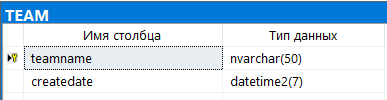


Рисунок 2.1 – Структура таблицы TEAM

Таблица содержит следующие поля:

* *TEAMNAME* – наименование футбольной команды – ключевое поле;
* *CREATEDATE* – дата создания футбольной команды;

На рисунке 2.3 представлена структура таблицы APUSERTEAM, которая предназначена для организации связи многие ко многим таблиц APUSER и TEAM:

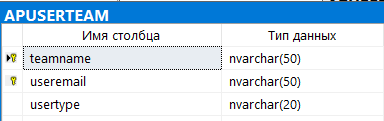


Рисунок 2.1 – Структура таблицы APUSERTEAM

Таблица содержит следующие поля:

* *TEAMNAME* – наименование футбольной команды – первая часть составного ключевого поля – внешний ключ к таблице TEAM;
* *USEREMAIL* – идентификатор пользователя – вторая часть составного ключевого поля – внешний ключ к таблице APUSER;
* *USERTYPE* – тип пользователя: создатель, участник.