

Федеральное Государственное Бюджетное Образовательное Учреждение
Высшего Образования

«Сибирский Государственный Университет
Телекоммуникаций и Информатики»

Кафедра Прикладной Математики и Кибернетики
(ПМиК)

РГР по дисциплине

«Визуальное программирование и человеко-машинное взаимодействие»

Вариант №4

Выполнил: студент 2 курса
группы ИП-015
Гулиев Э.Г.

Проверил: Милешко А.В.

Новосибирск, 2022

Содержание:

1. Задание на РГР	3
2. Ход работы.....	4
3. Исследование предметной области и создание ER диаграммы	5
4. Перевод ER диаграммы в реляционную модель, создание и заполнение БД	8
5. Проработка визуального интерфейса приложения	10
6. Создание диаграммы классов приложения	12
7. Реализация основного окна приложения.....	13
8. Реализация менеджера запросов	14

1. Задание на РГР

Создать ПО для отображения и обработки статистических данных для определённого вида спорта. ПО должно включать 2 основных окна: окно, отображающее таблицы БД со статистической информацией и результаты запросов к БД, переключение таблиц и результатов должно быть реализовано через вкладки; и окно для менеджера запросов к БД.

Первое окно должно давать возможность просматривать и изменять все таблицы БД, а также просматривать результаты запросов к БД. Должна иметься возможность удалить вкладки с результатами запросов, но не вкладки с таблицами. Также должна иметься возможность перейти к окну менеджера запросов.

Окно менеджера запросов должно предоставлять интерфейс для создания, сохранения, удаления, редактирования запросов. Созданные запросы должны отображаться в виде списка с названиями запросов, в который можно добавлять новые запросы, удалять, просматривать существующие. Для создания и редактирования запросов должен предоставляться визуальный интерфейс, а не язык запросов. Редактор запросов должен поддерживать операции выборки, соединения, группирования, подзапросы (в качестве подзапроса используются ранее сохранённые запросы).

В качестве базы данных должна использоваться SQLite. Для работы с БД можно использовать SQLite Studio. Все таблицы должны находиться в третьей нормальной форме.

Приложение и отчёт по работе должны быть размещены на GitHub, ссылка на репозиторий отправляется в качестве ответа на задание.

2. Ход работы

1. Исследование предметной области и создание ER диаграммы;
2. Перевод ER диаграммы в реляционную модель, создание и заполнение БД;
3. Проработка визуального интерфейса приложения;
4. Создание диаграммы классов приложения;
5. Реализация основного окна приложения;
6. Реализация менеджера запросов;
7. Тестирование и отладка.

3. Исследование предметной области и создание ER диаграммы

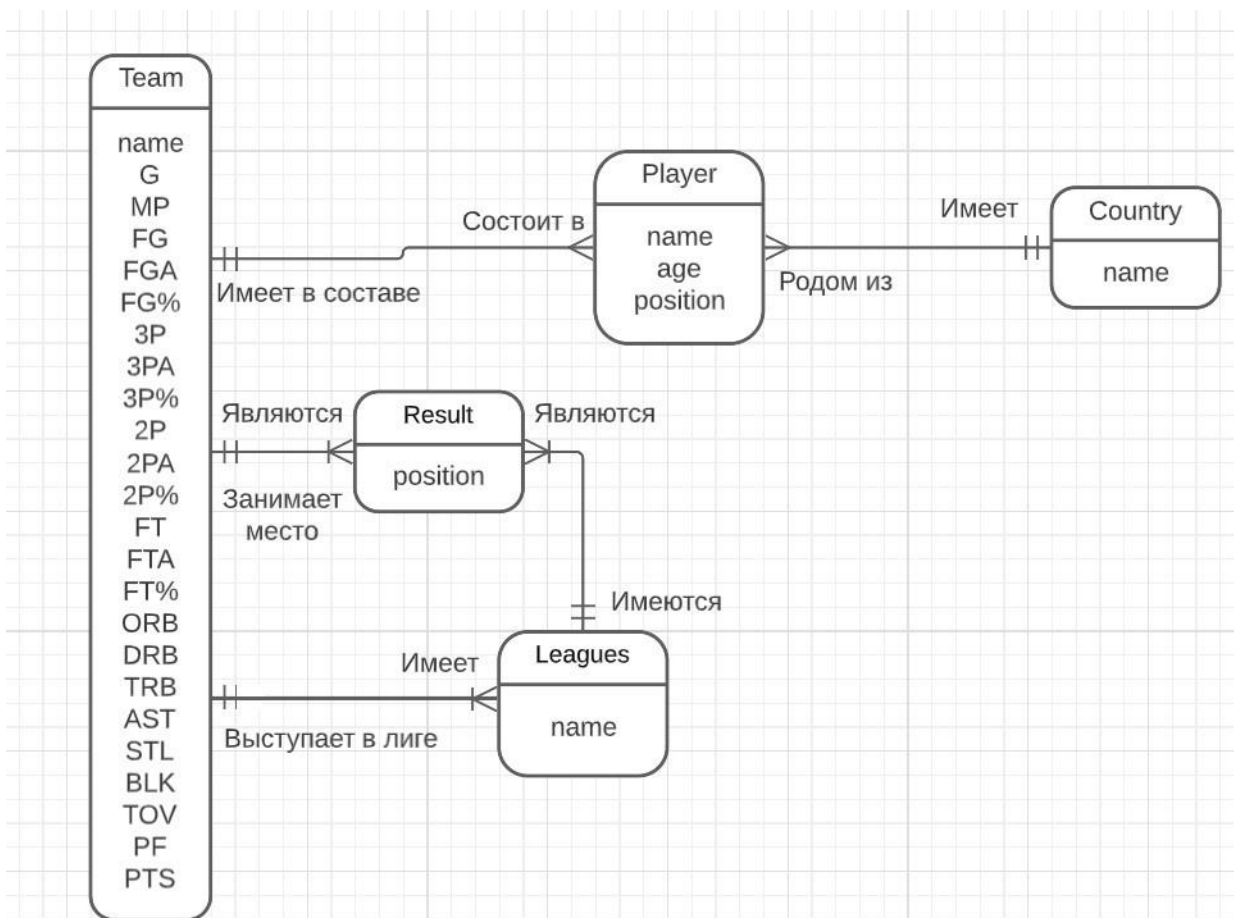


Рисунок 1. ER диаграмма

Имеем 5 сущностей:

1. **Player**, состоящая из полей:

name	ФИО игрока
age	Возраст игрока
position	Позиция игрока на площадке

Таблица 1. Поля сущности Player

2. **Country**, состоящая из полей:

name	Название страны
------	-----------------

Таблица 2. Поля сущности Country

3. **Team**, состоящая из полей:

name	Название команды
G	Количество игр
MP	Количество игрового времени
FG	Количество бросков, попавших в корзину
FGA	Количество всех выполненных бросков
FG%	Процентное соотношение между результативными бросками и общим количеством выполненных бросков
3P	Количество результативных бросков из трёхочковой зоны
3PA	Количество всех выполненных бросков из трёхочковой зоны
3P%	Процентное соотношение между результативными трёхочковыми бросками и общим количеством выполненных трёхочковых бросков
2P	Количество результативных двухочковых бросков
2PA	Количество выполненных двухочковых бросков
2P%	Процентное соотношение между результативными двухочковыми бросками и общим количеством выполненных двухочковых бросков
FT	Количество результативных штрафных бросков
FTA	Общее количество штрафных бросков
FT%	Процентное соотношение между результативными штрафными бросками и общим количеством штрафных бросков
ORB	Количество выигранных подборов под корзиной соперника
DRB	Количество выигранных подборов под своей корзиной
TRB	Полное количество выигранных подборов
AST	Количество передач, после которых был выполнен результативный бросок
STL	Количество удачных перехватов мяча

BLK	Количество удачно выполненных блоков
TOV	Количество потерянных мячей
PF	Количество персональных замечаний
PTS	Количество набранных очков

Таблица 3. Поля сущности Team

4. **Leagues**, состоящая из полей:

name	Название лиги
------	---------------

Таблица 4. Поля сущности Leagues

5. **Result**, состоящая из полей:

position	Позиция в лиге
----------	----------------

Таблица 5. Поля сущности Result

Взаимодействие сущностей между собой:

- Игрок родом только из одной страны. Страна является родиной многих игроков;
- Команда может иметь в своём составе множество игроков. Игрок может состоять только в одной команде;
- В лиге может быть множество результатов. Результаты для каждой лиги только одни;
- Команда занимает определённое место. Результат для команды только один.
- Команда выступает множестве лиг. В лиге только одна такая команда.

4. Перевод ER диаграммы в реляционную модель, создание и заполнение БД

Для перевода из ER диаграммы в реляционную модель достаточно преобразовать все сущности в таблицы.

Каждый простой тип сущности превращается в таблицу. Имя сущности становится именем таблицы. Каждый атрибут становится столбцом таблицы с тем же именем. Выставляются первичный и внешние ключи.

Player	
id_player	INTEGER первичный ключ
name	STRING
age	INTEGER
position	STRING
id_county	INTEGER внешний ключ
id_team	INTEGER внешний ключ

Таблица 6. Таблица БД «Player»

County	
id_county	INTEGER первичный ключ
name	STRING

Таблица 7. Таблица БД «County»

Team	
id_team	INTEGER первичный ключ
name	STRING
G	INTEGER
MP	INTEGER
FG	DOUBLE
FGA	DOUBLE
FG%	DOUBLE
3P	DOUBLE
3PA	DOUBLE
3P%	DOUBLE
2P	DOUBLE

2PA	DOUBLE
2P%	DOUBLE
FT	DOUBLE
FTA	DOUBLE
FT%	DOUBLE
ORB	DOUBLE
DRB	DOUBLE
TRB	DOUBLE
AST	DOUBLE
STL	DOUBLE
BLK	DOUBLE
TOV	DOUBLE
PF	DOUBLE
PTS	DOUBLE

Таблица 8. Таблица БД «Team»

League	
id_league	INTEGER первичный ключ
name	STRING

Таблица 9. Таблица БД «League»

Result	
id_league	INTEGER внешний ключ
id_team	INTEGER внешний ключ
place	INTEGER

Таблица 10. Таблица БД «Result»

Получившаяся БД находится в 3-ей нормальной форме.

5. Проработка визуального интерфейса приложения

Главное окно приложения содержит в себе основное меню:

File:

- Save (сохранение БД);
- Load (загрузка БД);
- Exit (выход из приложения).

About: кто сделал и как пользоваться.

Request: открывает окно с менеджером запросов.

Table: редактирование БД, удаление и добавление новых строк.

Ниже под меню находятся вкладки с таблицами и результатами запросов.

При смене вкладок меняется содержимое, отображаемое в таблице ниже.

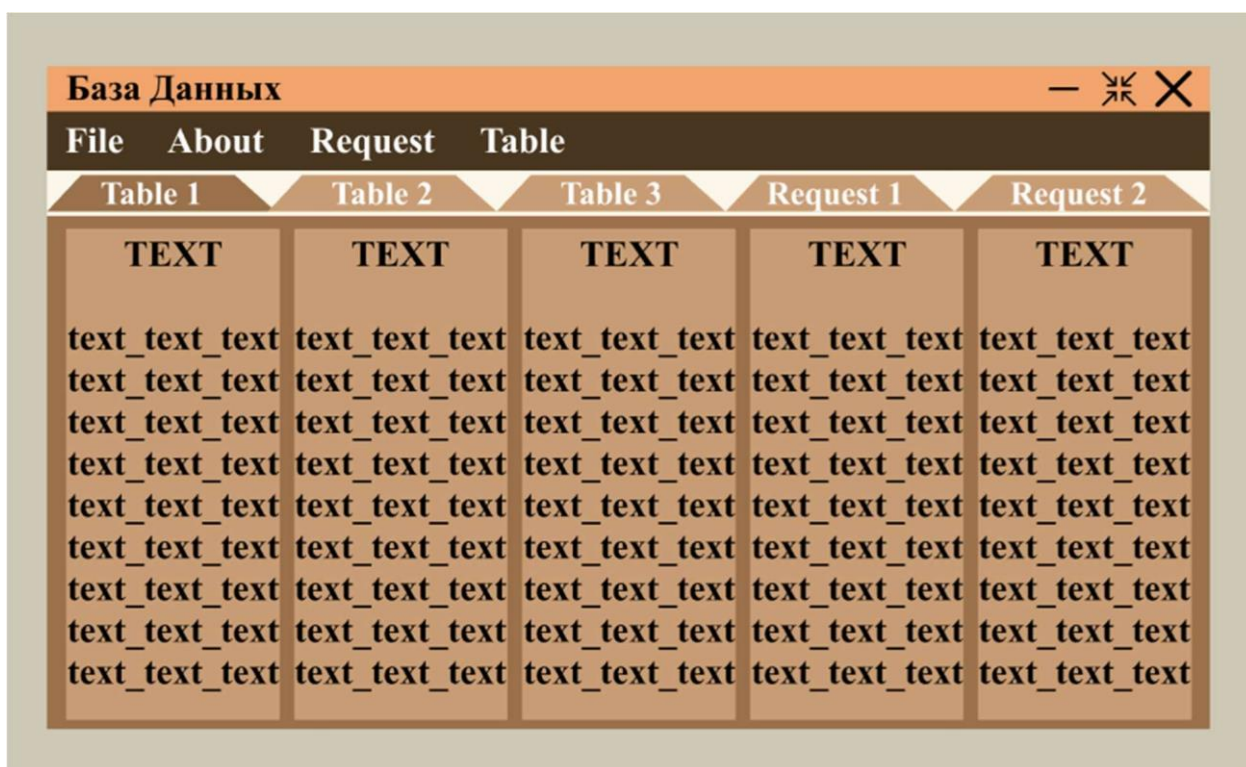


Рисунок 2. Главное окно

В менеджере запросов слева хранится список запросов. Можно создать новый или удалить. Также можно запустить выполнение запроса.

Справа находится конструктор запроса. Можно ввести имя запроса. Выбрать нужные столбцы таблиц а также выполнить запросы SELECT, JOIN, GROUP BY, WHERE, нажав на соответствующие кнопки.

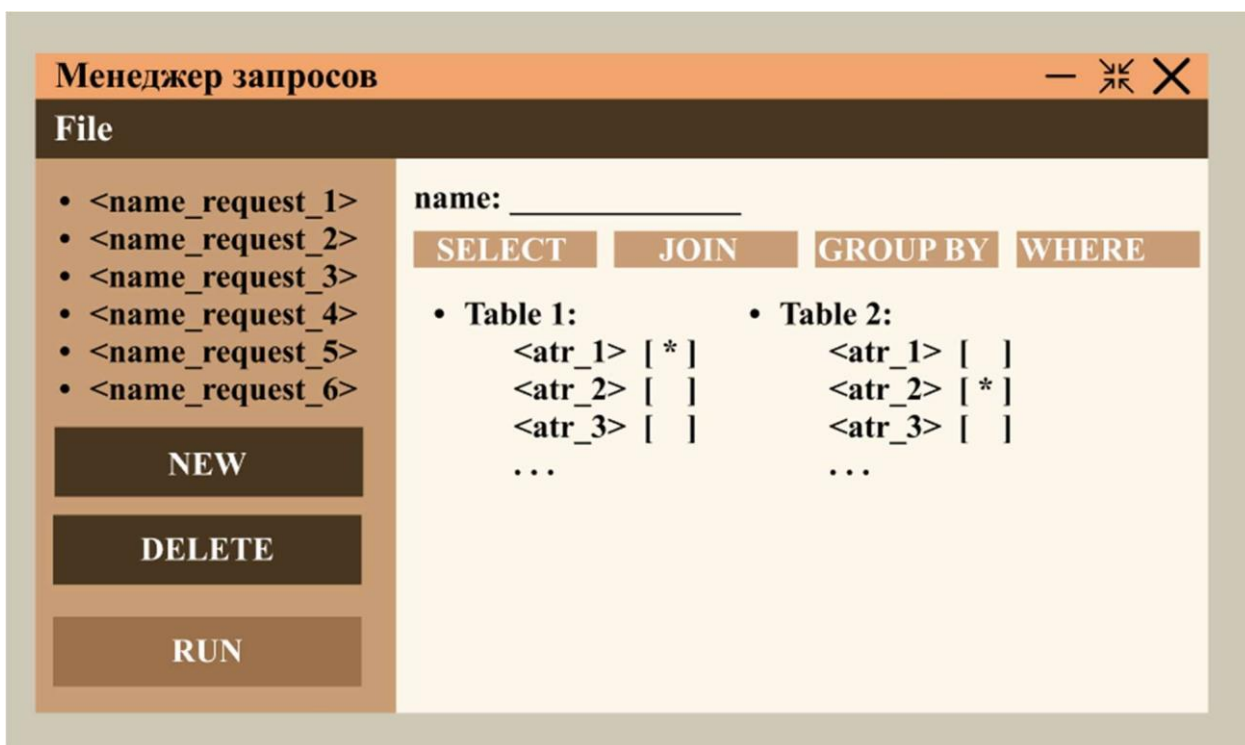


Рисунок 3. Менеджер запросов

В окошке WHERE editor можно написать условие для работы остальных запросов.

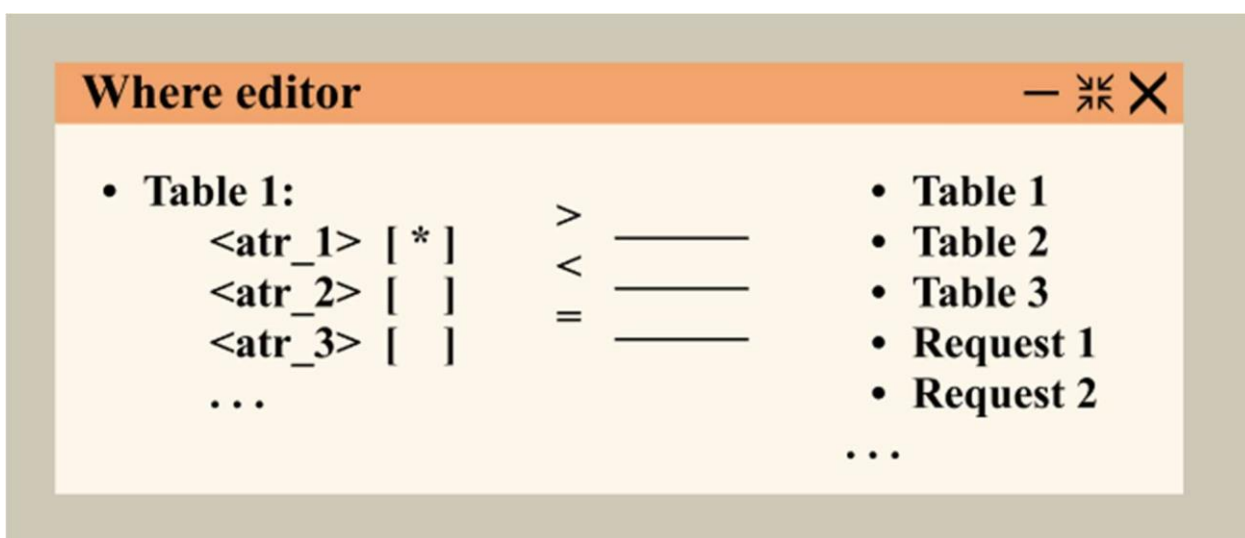


Рисунок 4. Where editor

6. Создание диаграммы классов приложения

В своей основе классы из слоя Models повторяют ER диаграмму и составленную базу данных. Также добавлен класс Request, описывающий запрос к БД, созданный в конструкторе.

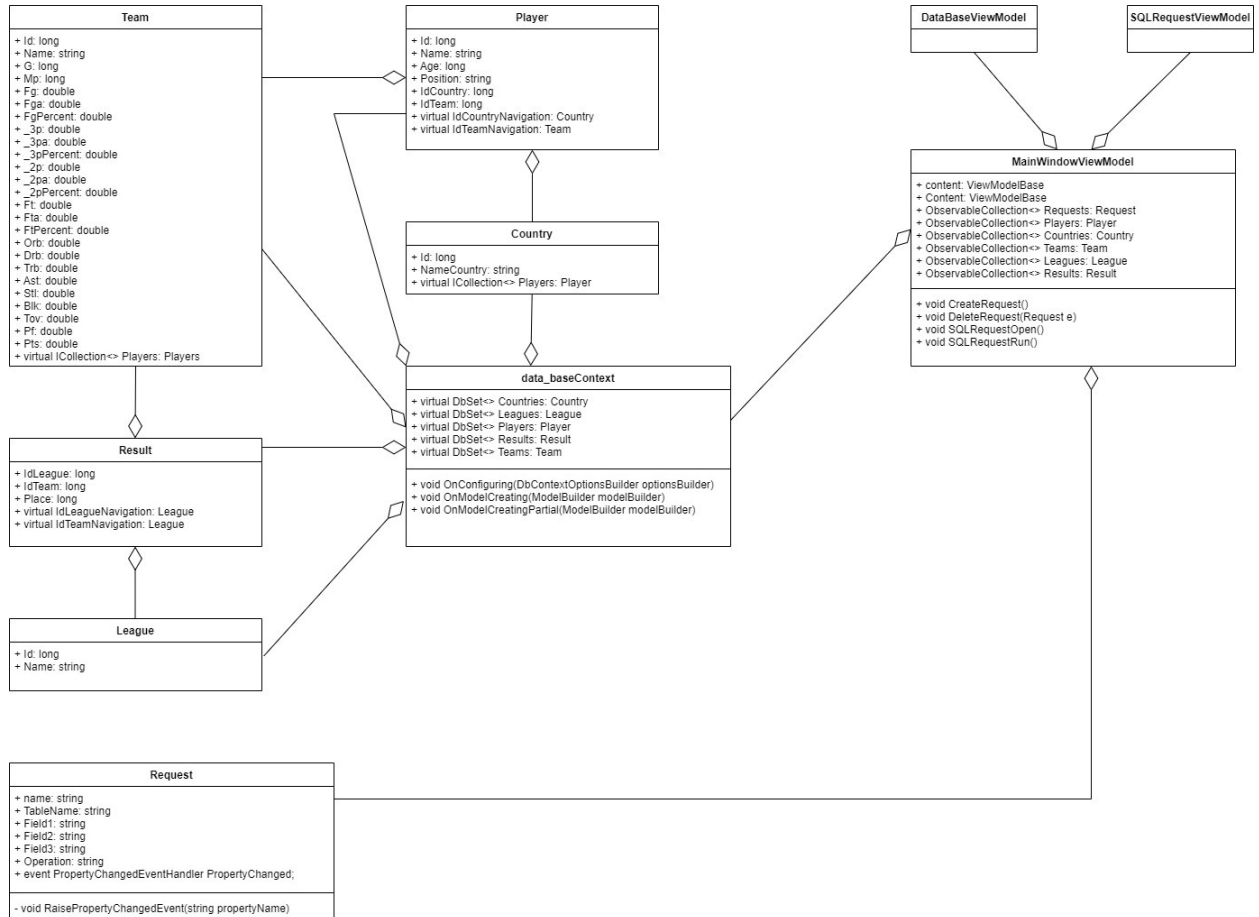


Рисунок 5. Диаграмма классов

7. Реализация основного окна приложения

RGR

Country	League	Player	Result	Team	
ID	name	age	position	id_country	id_team
2	Tyler Dorsey	30	Forward	5	1
3	Kostas Sloukas	21	Guard	3	1
4	Moustapha Fall	24	Forward	1	1
5	Shaquille McKissic	37	Guard	3	1
6	Kostas Papanikolaou	30	Forward	1	1
7	Thomas Walkup	20	Guard	5	1
8	Hassan Martin	22	Guard	2	1
9	Giannoulis Larentzakis	31	Guard	3	1
10	Giorgos Printezis	23	Guard	5	1
11	Livio Jean Charles	20	Guard	7	1
12	Michael Lountzis	20	Guard	6	1
13	Quincy Acy	22	Guard	4	1
14	Darryl Jackson	31	Forward	6	2
15	Nemanja Nedovic	29	Forward	7	2
16	Georgios Papagiannis	29	Guard	4	2
17	Ioannis Papapetrou	38	Forward	2	2
18	Okaro White	24	Guard	6	2
19	Howard Sant-Roos	21	Forward	6	2

Add Row

Delete Row

Save

Request Manager

RGR

</

8. Реализация менеджера запросов

