

МИНОБРНАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
«Санкт-Петербургский политехнический
университет Петра Великого»

Высшая школа прикладной математики и вычислительной физики
Секция «Телематика» (при ЦНИИ РТК)

Отчет о курсовой работе

Дисциплина: «Теоретические основы баз данных»

База данных шахматных партий

Сдал: _____ Черников С. Г. группа 3630201/80101
Принял: _____ к.т.н. доц. Попов С.Г.

Санкт-Петербург, 2021

Содержание

1	Аналитика	3
1.1	Описание предметной области	3
1.2	Проблемы предметной области и функции системы	5
1.3	Описание сущностей и атрибутов	6
1.4	ER-диаграмма	7
1.5	Диаграмма объектов	10
2	Синтез базы данных	11
2.1	Схема базы данных	11
2.2	Таблицы базы данных	12
2.3	Создание базы данных	16
3	Запросы к базе данных	17
3.1	Запрос 1	17
3.2	Запрос 2	20
3.3	Запрос 3	21
3.4	Запрос 4	23
3.5	Запрос 5	27
3.6	Запрос 6	29
3.7	Запрос 7	31
3.8	Запрос 8	33
	Заключение	36
	Список литературы	37
	Приложение А. Создание базы данных	38
	Приложение Б. Генерация данных для заполнения базы данных	42

1 Аналитика

1.1 Описание предметной области

Шахматы – настольная логическая игра со специальными фигурами на 64-клеточной доске для двух соперников, сочетающая в себе элементы искусства, науки и спорта.

Далее под термином "шахматы" будем понимать вид спорта "Шахматы" (правила утверждены приказом Министерства спорта РФ от 17 июля 2017 г. N 654). Существует набор единых стандартов в различных аспектах шахмат; эти стандарты приняты ФИДЕ – международной организацией шахмат; Правила вида спорта "Шахматы" не противоречат и дополняют их.

Спортивная деятельность шахматиста состоит в участии в соревнованиях – шахматных турнирах. Шахматный турнир – мероприятие, включающее в себя несколько туров, в каждом из которых осуществляется жеребьевка. В соответствии с жеребьевкой участники турнира делятся на пары (игрок белыми – игрок черными), каждая пара играет партию. При проведении турнира важную роль с организационной стороны выполняет главный арбитр, осуществляющий руководство линейными судьями (далее – арбитрами). В ходе партии арбитр следит за соблюдением правил, за тем, чтобы игрокам не мешали, контролирует ход соревнования и выполняет иные функции в соответствии с Правилами вида спорта "Шахматы".

В рамках шахматного турнира может быть предусмотрен временной контроль – ограничение по времени на всю партию для каждого игрока или ограничение времени на ход; в первом случае может быть предусмотрено добавление времени на партию или задержка времени. Каждая шахматная партия состоит из нескольких ходов (минимальное количество может быть установлено правилами соревнования). Ходы могут записываться в соответствии с Правилами. Партия оканчивается некоторым результатом (победой одного из игроков или ничьей), результат партии и оставшееся у каждого из игроков время фиксируются арбитром.

Каждый из вариантов (в некоторых случаях – семейство вариантов) начала партии (дебюта) имеет специальную кодовую запись в соответствии с распространенной классификацией, основанной на Энциклопедии шахматных дебютов – издании сербского издательства «Шахматный информатор». Каждый код представляет собой заглавную латинскую букву от А до Е и две цифры.

В ходе игры партия может достичь этапа эндшпиль – заключитель-

ной части шахматной партии, возникающей после того как было проведено сокращение большинства сил. Однако провести границу, отделяющую середину шахматной партии (миттельшпиль) от конца (эндшпиля), возможно не всегда. Обычно игра переходит в эндшпиль, когда разменено большинство фигур и нет характерных для середины игры угроз королям. Отсутствие на доске ферзей не является обязательным признаком эндшпиля. Позиции, соответствующие этапу эндшпиля, могут быть классифицированы как:

- пешечные окончания;
- коневые окончания;
- слоновые окончания;
- смешанные окончания;
- слон против коня;
- ладья против лёгкой фигуры;
- ферзь против ладьи;
- ферзь против лёгкой фигуры;
- тяжелофигурные окончания;
- ладейные окончания.

Каждый профессиональный шахматист имеет международный рейтинг ФИДЕ (рейтинг ЭЛО) – числовой коэффициент, вычисляемый по завершении каждого шахматного турнира, в котором спортсмен участвовал. Рейтинг ФИДЕ является числовой характеристикой успешности спортивной деятельности шахматиста – в основе системы рейтингов ЭЛО лежит допущение, что сила каждого шахматиста может быть представлена как вероятностная переменная, подчиняющаяся нормальному распределению. Расчёт рейтинга конкретного игрока по результатам какого-либо турнира основан на сравнении количества набранных им очков с ожидаемым, предсказанным на основе его рейтинга, количеством очков. Если по итогам турнира количество набранных очков оказывается больше, чем предсказанное значение, то рейтинг данного игрока возрастает. Если по итогам турнира количество набранных очков оказывается меньше, чем предсказанное значение, то рейтинг данного игрока уменьшается.

В зависимости от рейтинга ФИДЕ шахматисту присваивается звание ФИДЕ. Для перехода к следующему спортивному званию необходимо набрать определенное количество пунктов рейтинга.

В зависимости от среднего рейтинга шахматистов, участвующих в турнире, ФИДЕ относит турниры к различным категориям шахматных турниров.

1.2 Проблемы предметной области и функции системы

В данной предметной области значительный интерес представляет анализ имеющегося опыта сыгранных ранее партий (в том числе другими игроками). Такой анализ позволяет решить различные проблемы:

- изучение дебютной теории. Дебютная теория – область знаний, рассматривающая различные стратегии игры на первой стадии шахматной партии – в дебюте. Изучение имеющегося опыта игры позволяет уточнить имеющиеся сведения об оптимальных стратегиях игры в дебюте – может быть разработан новый дебютный вариант или новая стратегия игры после перехода из изучаемого дебюта в миттельшпиль (этап основной части шахматной партии, следующий после дебюта);
- изучение стиля игры потенциальных противников на соревнованиях высокого класса. При подготовке к соревнованиям шахматист может изучать уровень игры своих потенциальных оппонентов в различных ситуациях (в конкретных дебютах, типах эндшпилей и т.п.), а так же стиль игры в целом (склонность к нападению или защите, "жертвам" материала и т.п.);
- изучение шахматистом собственных сильных и слабых сторон;
- изучение шахматных партий с иными целями: например, изучение партий конкетного шахматиста с целью написания книги о его шахматных партиях и парадигмах игры в целом.

Данная курсовая работа посвящена проектированию и разработке базы данных, содержащей информацию о сыгранных ранее шахматных партиях различными шахматистами. Для каждой внесенной партии база данных должна содержать информацию об игроках, турнире, на котором была сыграна партия, и непосредственно о содержании партии – последовательности ходов, сделанных игроками.

К функциям разрабатываемой системы относится:

- поиск шахматных партий, в которых был разыгран дебют, соответствующий требуемому коду;
- поиск шахматных партий, в которых на доске возникала определенная позиция;
- поиск шахматных партий, в которых возникал эндшпиль определенного класса;
- поиск шахматных партий, сыгранных конкретным шахматистом (шахматистами) или результат которых фиксировался конкретным арбитром или партий сыгранных на конкретном турнире (или на турнире, судимом конкретным главным арбитром).

1.3 Описание сущностей и атрибутов

В рамках данной задачи могут быть введены следующие сущности:

1. Шахматист – отдельный спортсмен. Каждый шахматист имеет свой уникальный номер – id, получаемый при первом участии в турнире (на рейтинг).

Атрибуты:

- id;
- фамилия;
- имя;
- отчество (необязательно);
- год рождения.

2. Турнир – соревнования по шахматам, удовлетворяющие требованиям ФИДЕ.

Атрибуты:

- название;
- дата проведения (временной интервал);
- формат проведения (система, в соответствии с которой осуществляется жеребьевка перед каждым туром);
- контроль времени (правила, по которым осуществляется контроль времени);
- категория ФИДЕ.

3. Тур – один из этапов турнира, в рамках которого разбитые на пары игроки играют между собой.

Атрибуты:

- дата;
- номер тура.

4. Последовательность ходов – ходы, которые были сделаны каждым из игроков в ходе игры между ними.

Атрибуты:

- строковая запись (в соответствии с правилами записи шахматных партий).

5. Время на часах – время, оставшееся у каждого из игроков на шахматных часах в процессе их игры.

Атрибуты:

- минуты белых;
- секунды белых;
- минуты черных;
- секунды черных.

6. Арбитр, Главный арбитр.

Атрибуты:

- фамилия;
- имя;
- отчество (необязательно);
- квалификационная категория.

1.4 ER-диаграмма

Между введенными сущностями определим отношения:

Шахматист *участвует* в турнире, имея в этот период времени определенный рейтинг и звание.

Главный арбитр *судит* турнир.

Турнир *включает в себя* некоторое количество туров.

Шахматист *играет белыми* или *играет черными* в партии.

Арбитр *фиксирует* результат и время на часах для некоторых партий в определенных турах.

Каждый тур для каждой пары *состоит* в выполненной на доске последовательности ходов, завершившейся некоторым результатом.

На рис. 1 приведена ER-диаграмма базы данных шахматных партий.

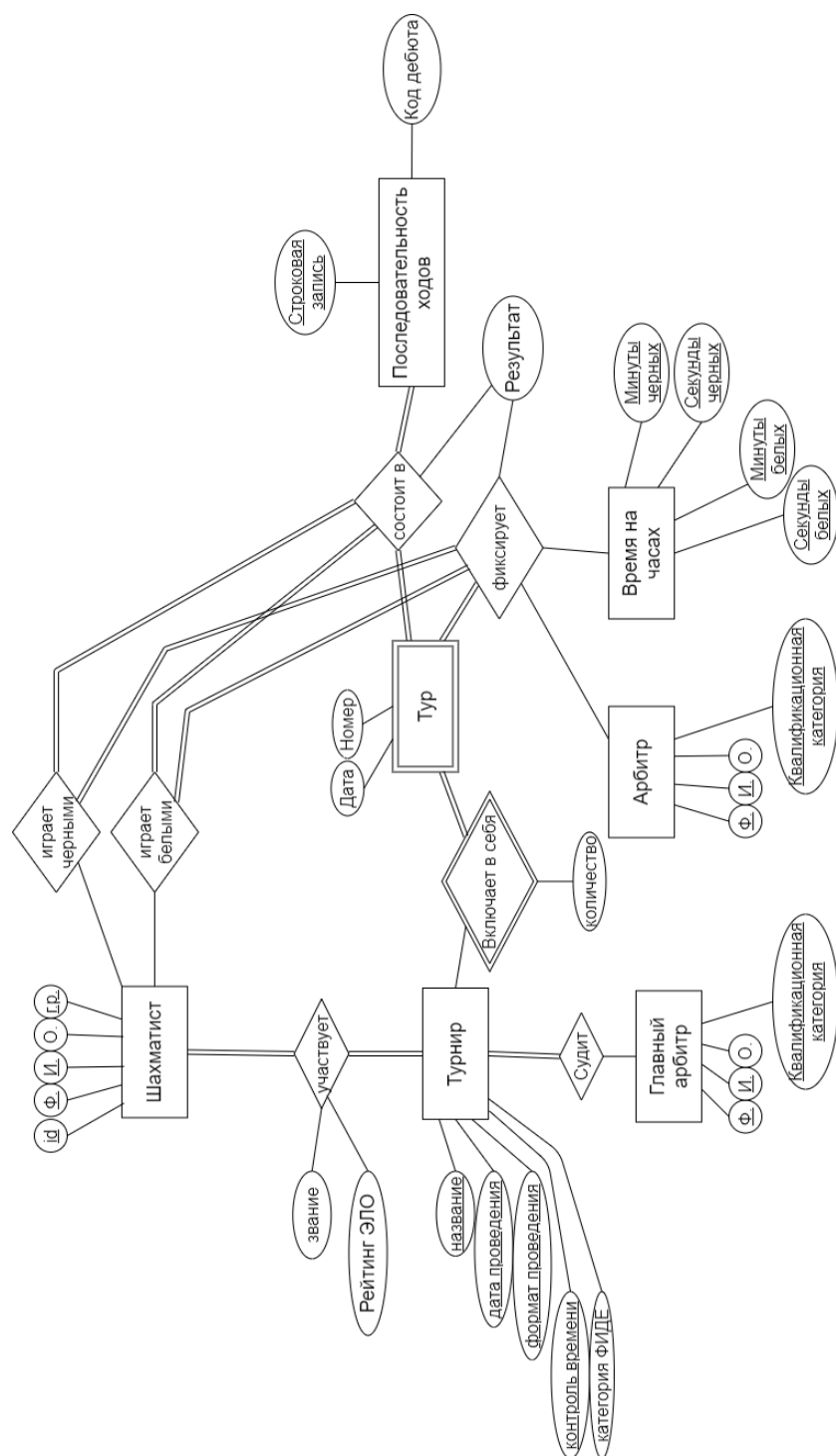


Рис. 1: ER-диаграмма базы данных

1.5 Диаграмма объектов

На рис. 2 приведена диаграмма объектов базы данных шахматных партий.

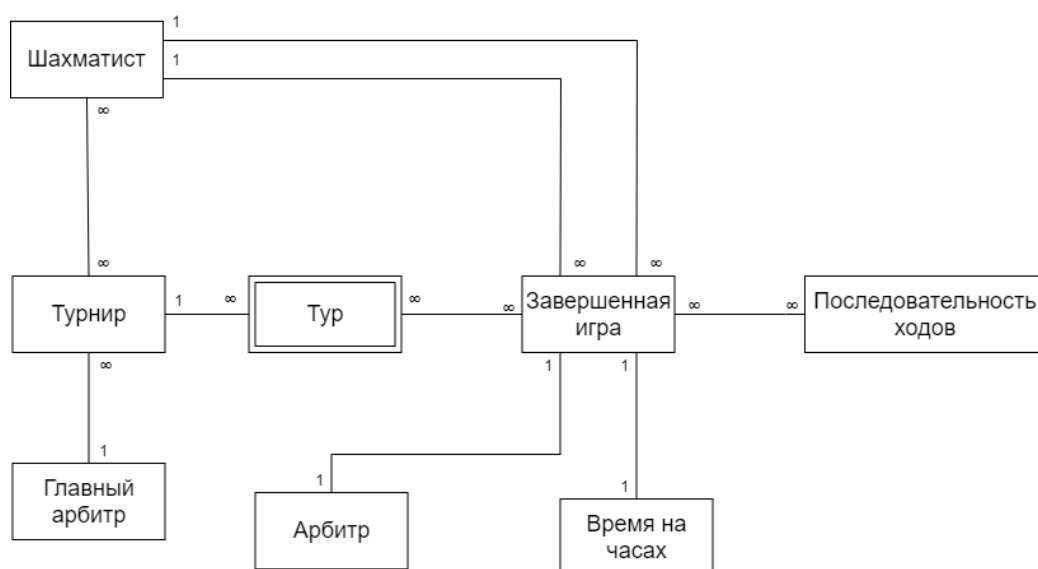


Рис. 2: Диаграмма объектов

2 Синтез базы данных

2.1 Схема базы данных

На рис. 3 приведена схема базы данных шахматных партий.

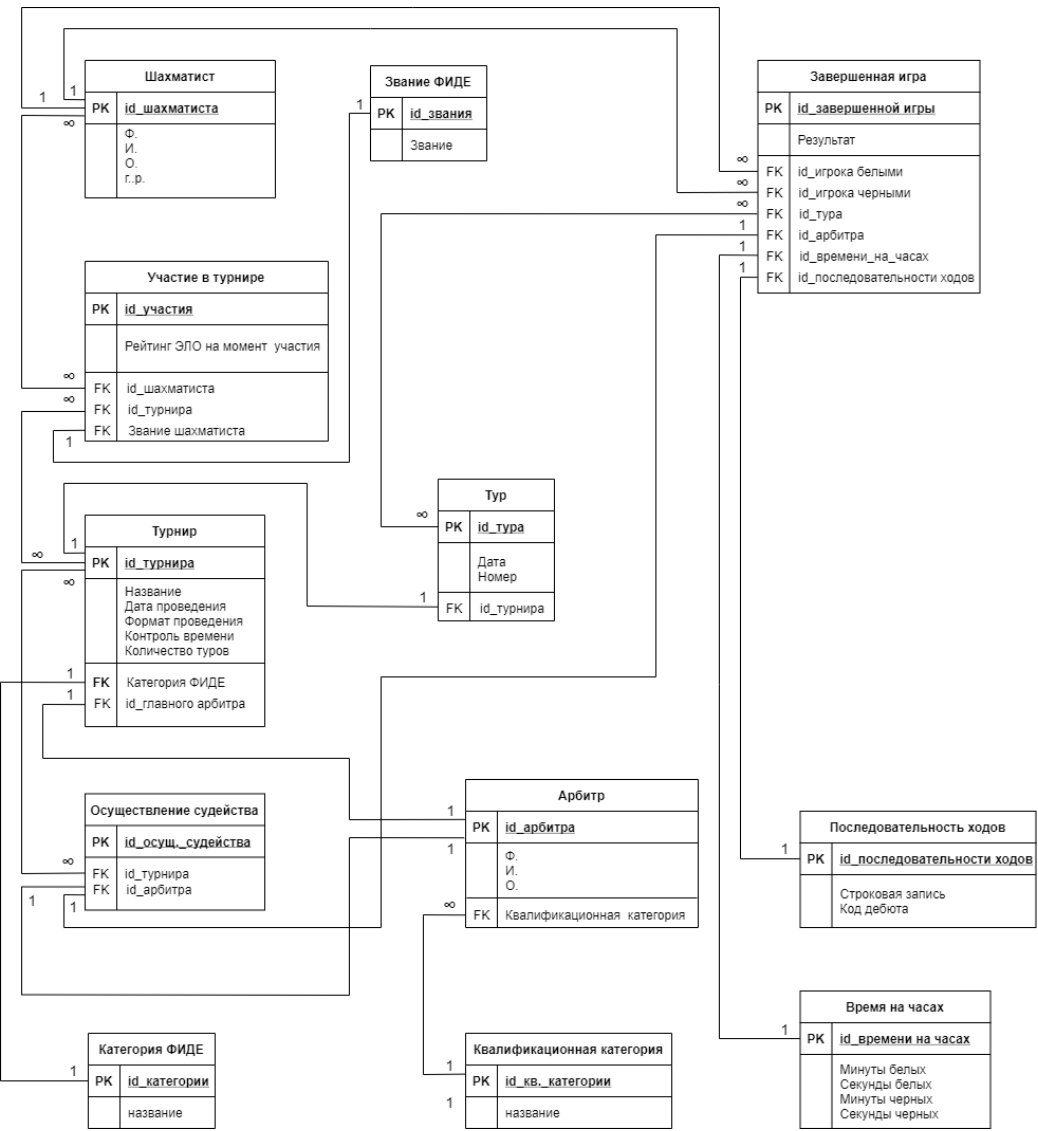


Рис. 3: Схема базы данных на русском языке

На рис. 4 приведена схема базы данных шахматных партий на английском языке.

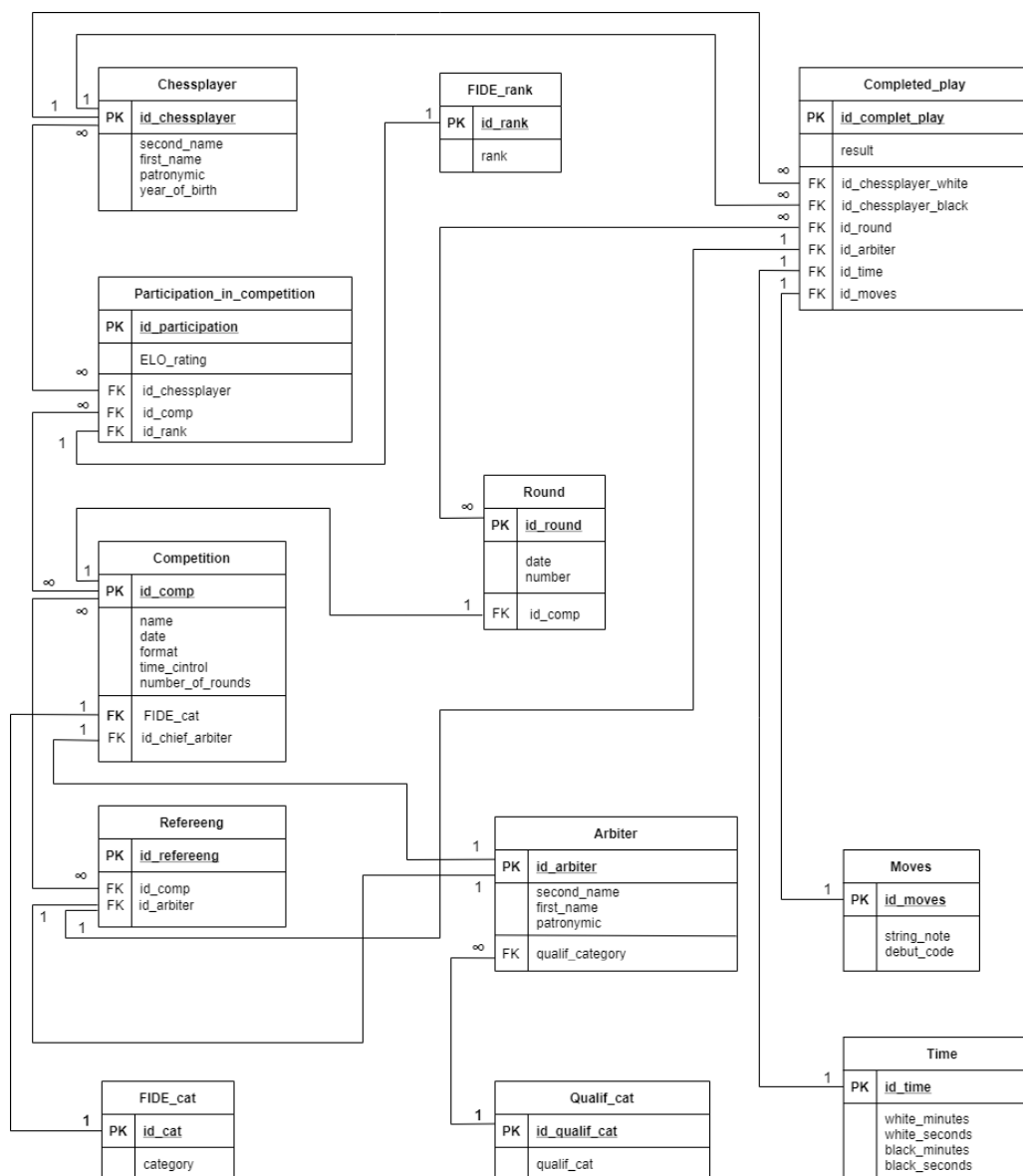


Рис. 4: Схема базы данных на английском языке

2.2 Таблицы базы данных

№	название атр.	назв атр. на англ.	тип	тип ключа	ссылка
1	id_шахматиста	id_chessplayer	BIGINT UNSIGNED	PK	-
2	Ф.	second_name	VARCHAR(30)	-	-

3	И.	first_name	VARCHAR(30)	-	-
4	О.	patronymic	VARCHAR(30)	-	-
5	г.р.	year_of_birth	YEAR	-	-

Таблица 1: Таблица "Шахматист" ("Chessplayer")

№	название атр.	назв атр. на англ.	тип	тип ключа	ссылка
1	id_участия	id_participation	INTEGER	PK	-
2	Звание шахматиста	FIDE_rank	TINYINT	FK	FIDE_rank: id_rank
3	Рейтинг ЭЛО на момент участия	ELO_rating	INTEGER UNSIGNED	-	-
4	id_шахматиста	id_chessplayer	BIGINT UNSIGNED	FK	Chessplayer: id_chessplayer
5	id_турнира	id_comp	INTEGER	FK	Competition: id_comp

Таблица 2: Таблица "Участие в турнире" ("Participation_in_competition")

№	название атр.	назв атр. на англ.	тип	тип ключа	ссылка
1	id_турнира	id_comp	INTEGER	PK	-
2	Название	_name	VARCHAR(30)	-	-
3	Дата проведения	_date	DATE	-	-
4	Формат проведения	_format	VARCHAR(30)	-	-
5	Контроль времени	time_control	VARCHAR(30)	-	-
6	Количество туров	number_of_rounds	TINYINT UNSIGNED	-	-
7	Категория ФИДЕ	FIDE_category	TINYINT	FK	FIDE_cat: id_cat
8	id главного арбитра	id_chief_arbiter	INTEGER	FK	Arbiter: id_arbiter

Таблица 3: Таблица "Турнир" ("Competition")

№	название атр.	назв атр. на англ.	тип	тип ключа	ссылка
1	id_осуш._судейства	id_refereeng	INTEGER	PK	-
2	id_турнира	id_comp	INTEGER	FK	Competition: id_comp

3	id_арбитра	id_arbiter	INTEGER	FK	Arbiter: id_arbiter
---	------------	------------	---------	----	------------------------

Таблица 4: Таблица "Осуществление судейства" ("Refereeng")

№	название атр.	назв атр. на англ.	тип	тип ключа	ссылка
1	id_категории	id_cat	TINYINT	PK	-
2	название	category	ENUM("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24")	-	-

Таблица 5: Таблица "Категории ФИДЕ" ("FIDE_cat")

№	название атр.	назв атр. на англ.	тип	тип ключа	ссылка
1	id_звания	FIDE_rank	TINYINT	PK	-
2	Звание	_rank	ENUM("GM", "IM", "FM", "CM", "WGM", "WIM", "WFM",)	-	-

Таблица 6: Таблица "Звание ФИДЕ" ("FIDE_rank")

№	название атр.	назв атр. на англ.	тип	тип ключа	ссылка
1	id_тура	id_round	INTEGER	PK	-
2	Дата	date	DATE	-	-
3	Номер	number	INTEGER	-	-
4	id_турнира	id_comp	INTEGER	FK	Competition: id_comp

Таблица 7: Таблица "Тур" ("Round")

№	название атр.	назв атр. на англ.	тип	тип ключа	ссылка
1	id_арбитра	id_arbiter	INTEGER	PK	-
2	Ф.	second_name	VARCHAR(30)	-	-
3	И.	first_name	VARCHAR(30)	-	-
4	О.	patronymic	VARCHAR(30)	-	-

5	Квалификационная категория	qualif_cat	TINYINT	FK	Qualif_cat: id_qualif_cat
---	----------------------------	------------	---------	----	------------------------------

Таблица 8: Таблица "Арбитр" ("Arbiter")

№	название атр.	назв атр. на англ.	тип	тип ключа	ссылка
1	id_кв._категории	id_qualif_cat	TINYINT	PK	-
2	название	qualif_cat	ENUM("A" "B" "C" "D")	-	-

Таблица 9: Таблица "Квалификационная категория" ("Qualif_cat")

№	название атр.	назв атр. на англ.	тип	тип ключа	ссылка
1	id_посл_ходов	id_moves	INTEGER	PK	-
2	Строковая запись	string_note	VARCHAR(65530)	-	-
3	Код дебюта	debut_code	CHAR(3)	-	-

Таблица 10: Таблица "Последовательность ходов" ("Moves")

№	название атр.	назв атр. на англ.	тип	тип ключа	ссылка
1	id_арбитра	id_time	INTEGER	PK	-
2	Минуты белых	white_minutes	INTEGER	-	-
3	Секунды белых	white_seconds	INTEGER	-	-
4	Минуты черных	black_minutes	INTEGER	-	-
5	Секунды черных	black_seconds	INTEGER	-	-

Таблица 11: Таблица "Время на часах" ("_Time")

№	название атр.	назв атр. на англ.	тип	тип ключа	ссылка
1	id_турнира	id_complet_play	INTEGER	PK	-
2	Результат	result	VARCHAR (30)	-	-
3	id_игрока белыми	id_chessplayer_ white	INTEGER	FK	Chessplayer: id_chessplayer
4	id_игрока черными	id_chessplayer_ black	INTEGER	FK	Chessplayer: id_chessplayer
5	id_тура	id_round	INTEGER	FK	_Round: id_round

6	id_арбитра	id_arbiter	INTEGER	FK	Refereeng: id_arbiter
7	id_времени_на_часах	id_time	INTEGER	FK	Time: id_time
8	id_ последовательности ходов	id_moves	INTEGER	FK	Moves: id_moves

Таблица 12: Таблица "Завершенная игра" ("Completed_play")

2.3 Создание базы данных

Для создания базы данных использовался скрипт, написанный на языке MySQL 8.0. Скрипт приведен в Приложении А. В таблице 13 указан порядок заполнения базы данных.

Название таблицы	Порядковый номер при заполнении	кол-во записей
FIDE_rank	1	8
FIDE_cat	1	25
Qualif_cat	1	4
Time	1	30000
Moves	1	30000
Chessplayer	1	5000
Arbiter	2	1500
Competition	3	100
Refereeng	4	1000
Round	4	1000
Participation_in_competition	4	6000
Completed_play	5	30000

Таблица 13: Порядок заполнения базы данных

Всего в базе данных содержится порядка 104000 записей. Алгоритм генерации данных приведен в Приложении Б.

3 Запросы к базе данных

В рамках данной работы требуется реализовать 8 запросов к базе данных в соответствии со следующим списком:

1. а) Вывести всех шахматистов, игры которых судил судья А и которые выиграли свою игру.
б) Вывести всех шахматистов, игры которых судил судья А и которые выиграли свою игру в пятом туре;
2. Посчитать количество турниров, в которых играл шахматист А, с категорией Б.
3. а) Найти арбитров, которые судили максимальное число игр;
б) Найти арбитров, которые судили минимальное число игр.
4. а) Найти шахматистов, которые сыграли больше партий, чем шахматист А;
б) Найти шахматистов, которые выиграли больше партий, чем шахматист А.
5. а) Посчитать число игр в каждом турнире;
б) Посчитать число игр в каждом туре турнира А.
6. Посчитать количество шахматистов с одинаковым числом побед.
7. Найти шахматистов, которые не играли в турнире А.
8. Посчитать число игр для каждой квалификационной категории судей по каждой категории ФИДЕ.

Для запросов 5, 6, 8 требуется построить диаграммы.

3.1 Запрос 1

Текст запроса: а) Вывести всех шахматистов, игры которых судил судья А и которые выиграли свою игру.

Код запроса:

```
1 SELECT Chessplayer.id_chessplayer AS id ,  
2   Chessplayer.second_name AS second_name ,  
3   Chessplayer.first_name AS first_name ,  
4   Chessplayer.pathronymic AS pathronymic ,
```

```

5   Arbiter.second_name AS arbiter
6 FROM Chessplayer
7 JOIN Completed_play
8   ON (Chessplayer.id_chessplayer = Completed_play.
   id_chessplayer_white AND Completed_play.result = "1-0"
   OR
9     Chessplayer.id_chessplayer = Completed_play.
   id_chessplayer_black AND Completed_play.result = "0-1")
10 JOIN Arbiter ON Arbiter.id_arbiter = Completed_play.
   id_arbiter
11 WHERE Completed_play.id_arbiter = 1;

```

На рис. 5 приведен результат выполнения запроса. Время выполнения – 0.000 сек – величина не вмещается в 3 разряда после запятой для вывода времени выполнения в среде MySQL Workbench. На рис. 6 приведен Explain запроса 1a).

	id	second_name	first_name	pathronymic	arbiter_second_name
▶	6288569	Фомин	Арсений	Антонович	Никитин
	1457605	Медведев	Леонид	Васильевич	Никитин
	2292294	Лосев	Яков	Семенович	Никитин
	9039568	Гусев	Сергей	Семенович	Никитин
	5207434	Глазунов	Иван	Аронович	Никитин
	8321479	Потапов	Владимир	Аронович	Никитин
	6170573	Иванов	Павел	Антонович	Никитин
	5272865	Зеликман	Борис	Владимирович	Никитин
	9315485	Глазунов	Валерий	Николаевич	Никитин
	2497988	Иванов	Петр	Семенович	Никитин
	5547198	Никонов	Леонид	Владимирович	Никитин
	1717983	Иванов	Геннадий	Андреевич	Никитин
	1786160	Волков	Никон	Геннадьевич	Никитин
	8274697	Гусев	Борис	Аронович	Никитин
	4620010	Никонов	Сергей	Алексеевич	Никитин
	9498995	Фомин	Никита	Андреевич	Никитин
	4841933	Денисов	Сергей	Николаевич	Никитин
	7745220	Лосев	Алексей	Геннадьевич	Никитин
	7439552	Медведев	Леонид	Владимирович	Никитин
	4927411	Гусев	Павел	Семенович	Никитин
	8526969	Щукин	Александр	Павлович	Никитин
	2688805	Курочкин	Павел	Алексеевич	Никитин
	7995156	Петров	Иван	Сергеевич	Никитин
	8518173	Щукин	Никита	Геннадьевич	Никитин
	3382770	Смирнов	Александр	Семенович	Никитин
	9339667	Потапов	Алексей	Иванович	Никитин

Рис. 5: Результат выполнения запроса 1a)

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	PRIMARY	Completed_play	none	ref	id_chessplayer_white,id_chessplayer_black,id_arbiter	id_arbi...	5	const	33	19.00	Using where
	1	PRIMARY	Chessplayer	none	ALL	PRIMARY				5070	0.04	Range checked for each record (index map: 0x1)
	2	SUBQUERY	Arbiter	none	const	PRIMARY	PRIMA...	4	const	1	100.00	

Рис. 6: Explain запроса 1а)

Для выполнения запроса необходимо связать таблицы Шахматист и Завершенная игра, чтобы получить таблицу с шахматистами, выигравшими свою партию. Из полученного соединения следует извлекать строки лишь в тех случаях, когда в таблице Завершенная игра указан требуемый арбитр, судивший матч.

По выводу Explain можно видеть, что в данной реализации базы данных операция определения выигранных шахматистом партий является тяжелой, поскольку требует прохода по всей таблице шахматистов.

Текст запроса: б) Вывести всех шахматистов, игры которых судил судья А и которые выиграли свою игру.

Код запроса:

```

1 SELECT Chessplayer.id_chessplayer AS id ,
2     Chessplayer.second_name AS second_name ,
3     Chessplayer.first_name AS first_name ,
4     Chessplayer.pathronymic AS pathronymic ,
5     Arbiter.second_name AS arbiter
6 FROM Chessplayer
7 JOIN (Completed_play
8 JOIN _Round ON Completed_play.id_round = _Round.id_round
9     AND _Round._number = 5
10 )
11     ON (Chessplayer.id_chessplayer = Completed_play.
12         id_chessplayer_white AND Completed_play.result = "1-0"
13     OR
14         Chessplayer.id_chessplayer = Completed_play.
15         id_chessplayer_black AND Completed_play.result = "0-1")
16 JOIN Arbiter ON Arbiter.id_arbiter = Completed_play.
17     id_arbiter
18 WHERE Completed_play.id_arbiter = 1;
```

На рис. 7 приведен результат выполнения запроса. Время выполнения – также 0.000 сек. На рис. 8 приведен Explain запроса 1б).

	id	second_name	first_name	pathronymic	arbiter_second_name
▶	9039568	Гусев	Сергей	Семенович	Никитин
	5272865	Зеликман	Борис	Владимирович	Никитин
	1786160	Волков	Никон	Геннадьевич	Никитин
	4841933	Денисов	Сергей	Николаевич	Никитин
	7745220	Лосев	Алексей	Геннадьевич	Никитин

Рис. 7: Результат выполнения запроса 1б)

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	PRIMARY	Completed_play	const	ref	id_chessplayer_white,id_chessplayer_black,id_round,id_arbiter	id_arbi...	5	const	33	19.00	Using where
	1	PRIMARY	_Round	const	eq_ref	PRIMARY	PRIMA...	4	chessbase.Completed_play.id...	1	10.00	Using where
	1	PRIMARY	Chessplayer	const	ALL	PRIMARY				5070	0.04	Range checked for each record (index map: 0x1)
	2	SUBQUERY	Arbiter	const	const	PRIMARY	PRIMA...	4	const	1	100.00	

Рис. 8: Explain запроса 1б)

Запрос во многом аналогичен запросу из пункта а), однако отличается от него тем, что выполняется соединение таблицы Завершенная игра с таблицей Тур с целью фильтрации первой по туру №5, после этого аналогично пункту а) выполняется соединение с таблицей Шахматист.

3.2 Запрос 2

Текст запроса: Посчитать количество турниров, в которых играл шахматист А, с категорией Б.

Код запроса:

```
1 SELECT COUNT(*) AS num
2 FROM Participation_in_competition
3 JOIN Competition ON Competition.id_comp =
   Participation_in_competition.id_comp
4 WHERE Participation_in_competition.id_chessplayer =
   1006497 AND Competition.FIDE_category = 1;
```

На рис. 9 приведен результат выполнения запроса. Время выполнения – 0.01 сек. На рис. 10 приведен Explain запроса 2.

	num
▶	1

Рис. 9: Результат выполнения запроса 2

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
►	1	SIMPLE	Participation_in_competition	const	ref	id_chessplayer,id_comp	id_che...	9	const	1	100.00	Using where
	1	SIMPLE	Competition	const	eq_ref	PRIMARY,FIDE_category	PRIMA...	4	chessbase.Participation_in_competition.id...	1	5.00	Using where

Рис. 10: Explain запроса 2

Запрос состоит в подсчете строк у сцепления таблицы Участие в турнире с таблицей Турнир по условию: id шахматиста соответствует требуемому, запись таблицы Участие в турнире соответствует записи таблицы Турнир, и в записи Турнир указана требуемая категория.

3.3 Запрос 3

Текст запроса: а) Найти арбитров, которые судили максимальное число игр.

Код запроса:

```

1 SELECT Arbiter.id_arbiter AS id ,
2     Arbiter.second_name AS second_name ,
3     COUNT(*) AS num
4 FROM Arbiter
5     JOIN Completed_play ON Arbiter.id_arbiter =
6         Completed_play.id_arbiter
7 GROUP BY Arbiter.id_arbiter
8 HAVING num = (
9     SELECT COUNT(*) AS max_num
10    FROM Arbiter
11        JOIN Completed_play ON Arbiter.id_arbiter =
12            Completed_play.id_arbiter
13    GROUP BY Arbiter.id_arbiter
14    ORDER BY max_num DESC
15    LIMIT 1
16 );

```

На рис. 11 приведен результат выполнения запроса. Время выполнения – 0.09 сек. На рис. 12 приведен Explain запроса 3а).

	id	second_name	num
►	153	Волков	83

Рис. 11: Результат выполнения запроса 3а)

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
	1	PRIMARY	Completed_play	none	index	id_arbiter	id_arbi...	5	none	26459	100.00	Using where; Using index; Using temporary
	1	PRIMARY	Arbiter	none	eq_ref	PRIMARY,qualif_cat	PRIMA...	4	chessbase.Completed_play.id_arbiter	1	100.00	none
	2	SUBQUERY	Arbiter	none	index	PRIMARY,qualif_cat	PRIMA...	4	none	1500	100.00	Using index; Using temporary; Using filesort
	2	SUBQUERY	Completed_play	none	ref	id_arbiter	id_arbi...	5	chessbase.Arbiter.id_arbiter	26	100.00	Using index

Рис. 12: Explain запроса 3а)

Запрос состоит в создании таблицы арбитров с полями id, фамилия и количество партий, которые судил данный арбитр. Для этого связываются таблицы Арбитр и Завершенная игра, выполняется группировка по id арбитра и подсчитывается количество соответствующих каждому арбитру партий. К полученной таблице следует применить групповой фильтр HAVING, оставив лишь те строки, которые соответствуют максимальному количеству партий.

Для вычисления максимального количества партий выполняется запрос, который создает такую же таблицу, какая была описана выше, упорядочивает записи по убыванию количества партий и оставляет первую запись. Таким образом, вычисляется максимальное значение количества партий, которому должно равняться количество партий в записях итоговой таблицы.

Текст запроса: б) Найти арбитров, которые судили минимальное число игр.

Код запроса:

```

1 SELECT Arbiter.id_arbiter AS id ,
2     Arbiter.second_name AS second_name ,
3     COUNT(*) AS num
4 FROM Arbiter
5     JOIN Completed_play ON Arbiter.id_arbiter =
6         Completed_play.id_arbiter
7 GROUP BY Arbiter.id_arbiter
8 HAVING num = (
9     SELECT COUNT(*) AS min_num
10    FROM Arbiter JOIN Completed_play ON Arbiter.id_arbiter =
11        Completed_play.id_arbiter
12    GROUP BY Arbiter.id_arbiter
13    ORDER BY min_num
14    LIMIT 1
15 );

```

На рис. 13 приведен результат выполнения запроса. Время выполнения – 0.09 сек. На рис. 14 приведен Explain запроса 3б).

	id	second_name	num
▶	184	Медведев	6

Рис. 13: Результат выполнения запроса 3б)

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	PRIMARY	Completed_play	none	index	id_arbiter	id_arbi...	5	none	26459	100.00	Using where; Using index; Using temporary
	1	PRIMARY	Arbiter	none	eq_ref	PRIMARY,qualif_cat	PRIMA...	4	chessbase.Completed_play.id_arbiter	1	100.00	none
	2	SUBQUERY	Arbiter	none	index	PRIMARY,qualif_cat	PRIMA...	4	none	1500	100.00	Using index; Using temporary; Using filesort
	2	SUBQUERY	Completed_play	none	ref	id_arbiter	id_arbi...	5	chessbase.Arbiter.id_arbiter	26	100.00	Using index

Рис. 14: Explain запроса 3б)

Выполнение запроса аналогично пункту а), различие состоит лишь в групповой фильтрации, при которой происходит сравнение с минимальным значением количества партий. Это значение вычисляется аналогично предыдущему пункту, однако упорядочение записей происходит по возрастанию, а не по убыванию.

3.4 Запрос 4

Текст запроса: а) Найти шахматистов, которые сыграли больше партий, чем шахматист А.

Код запроса:

```

1 SELECT Chessplayer.id_chessplayer AS id ,
2    Chessplayer.second_name AS second_name , Chessplayer.
3    first_name AS first_name ,
4    Chessplayer.pathronymic AS pathronymic ,
5    COUNT(*) AS num ,
6    (
7        SELECT COUNT(*)
8        FROM Chessplayer
9        JOIN Completed_play ON Chessplayer.id_chessplayer =
10       Completed_play.id_chessplayer_white OR Chessplayer.
11       id_chessplayer = Completed_play.id_chessplayer_black
12       WHERE Chessplayer.id_chessplayer = 6395208
13    ) as games_of_a
14 FROM Chessplayer
15 JOIN Completed_play ON Chessplayer.id_chessplayer =
16    Completed_play.id_chessplayer_white OR Chessplayer.
17    id_chessplayer = Completed_play.id_chessplayer_black
18 GROUP BY Chessplayer.id_chessplayer
19 HAVING num > (

```

```

15 SELECT COUNT(*)
16 FROM Chessplayer
17 JOIN Completed_play ON Chessplayer.id_chessplayer =
    Completed_play.id_chessplayer_white OR Chessplayer.
    id_chessplayer = Completed_play.id_chessplayer_black
18 WHERE Chessplayer.id_chessplayer = 6395208
19 );

```

На рис. 15 приведен результат выполнения запроса. Время выполнения – 0.83 сек. На рис. 16 приведен Explain запроса 4а).

	id	second_name	first_name	pathronymic	num	games_of_a
▶	1087352	Иванов	Никон	Александрович	17	16
	1106408	Никонов	Петр	Михайлович	20	16
	1195147	Никонов	Арсен	Васильевич	18	16
	1244949	Медведев	Никита	Алексеевич	17	16
	1306427	Курочкин	Андрей	Павлович	17	16
	1310425	Лосев	Алексей	Иванович	17	16
	1311373	Денисов	Николай	Александрович	17	16
	1401436	Волбуев	Арсен	Геннадьевич	17	16
	1411579	Денисов	Самуил	Семенович	17	16
	1428857	Вист	Арсен	Сергеевич	18	16
	1436717	Кузнецов	Василий	Аронович	21	16
	1529612	Зеликман	Яков	Сергеевич	20	16
	1530607	Денисов	Сергей	Александрович	18	16
	1545322	Блинов	Иван	Иванович	17	16
	1554991	Никонов	Валерий	Николаевич	17	16
	1563078	Глазунов	Арсен	Александрович	22	16
	1582450	Вист	Михаил	Андреевич	19	16
	1639681	Зайцев	Никита	Владимирович	19	16
	1649738	Зеликман	Петр	Валерьевич	17	16
	1700727	Кузнецов	Александр	Петрович	17	16
	1717983	Иванов	Геннадий	Андреевич	19	16
	1843811	Щукин	Михаил	Валерьевич	20	16
	1877114	Денисов	Леонид	Андреевич	18	16
	1911732	Волбуев	Арсений	Алексеевич	17	16
	1936481	Денисов	Владимир	Иванович	17	16
	1936889	Щукин	Арсен	Семенович	18	16

Рис. 15: Результат выполнения запроса 4а) (начало вывода)

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶ 1	PRIMARY	Chessplayer		index	PRIMARY	PRIMARY	8		5070	100.00	
1	PRIMARY	Completed_play		ALL	id_chessplayer_white,id_ch...				26459	0.04	Range checked for each record (index map: 0x6)
3	SUBQUERY	Chessplayer		const	PRIMARY	PRIMARY	8	const	1	100.00	Using index
3	SUBQUERY	Completed_play		inde...	id_chessplayer_white,id_ch...	id_chessplayer_white,id_ch...	9,9		16	100.00	Using union(id_chessplayer_white,id_chessplay...
2	SUBQUERY	Chessplayer		const	PRIMARY	PRIMARY	8	const	1	100.00	Using index
2	SUBQUERY	Completed_play		inde...	id_chessplayer_white,id_ch...	id_chessplayer_white,id_ch...	9,9		16	100.00	Using union(id_chessplayer_white,id_chessplay...

Рис. 16: Explain запроса 4а)

Запрос состоит в построении таблицы шахматистов с указанием id,

ФИО, количества сыгранных партий и столбца с одним (одинаковым для всех строк) значением – количество партий, сыгранных игроком А. Таблица строится соединением с таблицей Завершенная игра, после чего выполняется фильтрация записей: остаются лишь те записи, которым соответствуют игроки, сыгравшие больше партий, чем игрок А (происходит групповая фильтрация).

Текст запроса: б) Найти шахматистов, которые выиграли больше партий, чем шахматист А.

Код запроса:

```
1 SELECT Chessplayer.id_chessplayer AS id ,
2     Chessplayer.second_name AS second_name , Chessplayer.
3     first_name AS first_name ,
4     Chessplayer.pathronymic AS pathronymic ,
5     @cnt := COUNT(*) AS num
6 FROM Chessplayer
7 JOIN Completed_play ON Chessplayer.id_chessplayer =
8     Completed_play.id_chessplayer_white AND Completed_play.
9     result = '1-0'
10 OR Chessplayer.id_chessplayer = Completed_play.
11 id_chessplayer_black AND Completed_play.result = '0-1'
12 WHERE Chessplayer.id_chessplayer = 1649738
13 UNION ALL
14 SELECT Chessplayer.id_chessplayer AS id ,
15     Chessplayer.second_name AS second_name , Chessplayer.
16     first_name AS first_name ,
17     Chessplayer.pathronymic AS pathronymic ,
18     COUNT(*) AS num
19 FROM Chessplayer
20 JOIN Completed_play ON Chessplayer.id_chessplayer =
21     Completed_play.id_chessplayer_white AND Completed_play.
22     result = '1-0'
23 OR Chessplayer.id_chessplayer = Completed_play.
24 id_chessplayer_black AND Completed_play.result = '0-1'
25 GROUP BY Chessplayer.id_chessplayer
26 HAVING num > @cnt;
```

На рис. 17 приведен результат выполнения запроса. Время выполнения – 32.8 сек. На рис. 18 приведен Explain запроса 4б).

id	second_name	first_name	pathronymic	num
1649738	Зеликман	Петр	Валерьевич	7
1026765	Блинов	Николай	Павлович	9
1091483	Гусев	Яков	Валерьевич	8
1116136	Волков	Яков	Александрович	8
1214138	Медведев	Антон	Иванович	9
1225947	Курочкин	Владимир	Иванович	8
1244949	Медведев	Никита	Алексеевич	8
1266857	Зайцев	Андрей	Васильевич	8
1310425	Лосев	Алексей	Иванович	10
1517680	Петров	Александр	Валерьевич	8
1525451	Никитин	Леонид	Антонович	10
1532212	Медведев	Иван	Аронович	8
1560107	Блинов	Семен	Аронович	8
1563078	Глазунов	Арсен	Александрович	8
1739610	Зеликман	Иван	Александрович	8
1850367	Иванов	Алексей	Иванович	8
1877114	Денисов	Леонид	Андреевич	8
1887684	Медведев	Алексей	Александрович	8
1957541	Зайцев	Борис	Геннадьевич	8
1971234	Никонов	Андрей	Васильевич	8
2005025	Волобуев	Геннадий	Владимирович	9
2009567	Шукин	Никита	Михайлович	8
2245766	Никонов	Марк	Васильевич	8
2257051	Гусев	Антон	Сергеевич	8
2319866	Денисов	Андрей	Васильевич	8
2325808	Лосев	Петр	Антонович	8
2333309	Шукин	Владимир	Семенович	8
2623667	Глазунов	Леонид	Семенович	9
2628979	Лосев	Никон	Владимирович	8
2664037	Гусев	Марк	Михайлович	8
2714912	Лосев	Леонид	Антонович	10
2721532	Никитин	Леонид	Николаевич	8
2763023	Фомин	Антон	Геннадьевич	8
2769184	Потапов	Семен	Петрович	8
2810086	Волков	Александр	Геннадьевич	8
2819843	Блинов	Антон	Александрович	8
2842279	Шукин	Василий	Семенович	9
2865323	Зеликман	Марк	Михайлович	9
2866317	Денисов	Михаил	Алексеевич	8
2884560	Вист	Яков	Владимирович	8
2890883	Медведев	Валерий	Алексеевич	8
3048669	Блинов	Марк	Алексеевич	8
3091257	Денисов	Антон	Семенович	9
3134841	Курочкин	Петр	Андреевич	8
3161792	Фомин	Николай	Геннадьевич	8
3186181	Курочкин	Марк	Александрович	8
3208489	Блинов	Валерий	Валерьевич	8

Рис. 17: Результат выполнения запроса 4б) (начало вывода)

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	Chessplayer		const	PRIMARY	PRIMARY	8	const	1	100.00	
1	PRIMARY	Completed_play		inde...	id_chessplayer_white,id_ch...	id_chessplayer_white,id_ch...	9,9		17	19.00	Using union(id_chessplayer_white,id_chessplay...
2	UNION	Completed_play		ALL	id_chessplayer_white,id_ch...				26459	19.00	Using where; Using temporary
2	UNION	Chessplayer		ALL	PRIMARY				5070	0.04	Using where; Using join buffer (hash join)

Рис. 18: Explain запроса 4б)

Запрос коренным образом отличается от представленного в пункте а); запрос состоит в слиянии (UNION) двух таблиц, в первой подсчитывается количество партий, выигранных шахматистом А, а во второй – количество партий для каждого из шахматистов в базе данных. После подсчета выполняется фильтрация: количество выигранных партий должно быть больше, чем у шахматиста А (число выигранных им партий записано в переменной cnt).

Несмотря на внесенные улучшения (запрос, аналогичный пункту а), работает вдвое дольше) нельзя не отметить, что запрос выполняется исключительно долго – 32,8 сек. Это обусловлено тем, что построение таблицы выигранных игроком партий – крайне тяжелая операция в рам-

ках данной реализации базы данных, что можно видеть еще в таблице Explain для запроса 1.

3.5 Запрос 5

Текст запроса: а) Посчитать число игр в каждом турнире.

Код запроса:

```

1 SELECT Competition.id_comp AS id_comp, Competition._name as
   _name, COUNT(*) as number_of_plays
2 FROM Competition
3 JOIN _Round ON Competition.id_comp = _Round.id_comp
4 JOIN Completed_play ON Completed_play.id_round = _Round.
   id_round
5 GROUP BY Competition.id_comp;
```

На рис. 19 приведен результат выполнения запроса. Время выполнения – 0.03 сек. На рис. 20 приведен Explain запроса 5а).

	id_comp	_name	number_of_rounds
▶	0	Турнир города Москва	251
	1	Приз весенних каникул	282
	2	Чемпионат станции Бологое	336
	3	Приз города Москва	266
	4	Встреча весенних каникул	280
	5	Турнир весенних каникул	297
	6	Встреча осенних каникул	303
	7	Приз города Васюки	293
	8	Встреча весенних каникул	281
	9	Встреча весенних каникул	225
	10	Приз осенних каникул	241
	11	Кубок станции Бологое	254
	12	Приз осенних каникул	289
	13	Кубок весенних каникул	281
	14	Кубок станции Бологое	271
	15	Приз осенних каникул	304
	16	Приз города Васюки	266
	17	Турнир весенних каникул	296
	18	Турнир осенних каникул	293
	19	Чемпионат города Москва	239
	20	Встреча осенних каникул	252
	21	Чемпионат осенних каникул	272
	22	Турнир станции Бологое	273
	23	Встреча весенних каникул	316
	24	Чемпионат весенних каникул	243
	25	Турнир осенних каникул	302

Рис. 19: Результат выполнения запроса 5а) (начало вывода)

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	Competition	1	index	PRIMARY,FIDE_category,id...	PRIMARY	4		100	100.00	
1	SIMPLE	_Round	1	ref	PRIMARY,id_comp	id_comp	5	chessbase.Competition.id_comp	9	100.00	Using index
1	SIMPLE	Completed_play	1	ref	id_round	id_round	5	chessbase._Round.id_round	29	100.00	Using index

Рис. 20: Explain запроса 5а)

Запрос состоит в соединении таблиц Турнир, Тур и Завершенная игра, группировке по полю id Турнира и подсчете количества игр, соответствующих каждому турниру.

На рис. 21 приведена диаграмма, построенная для результата выполнения запроса.

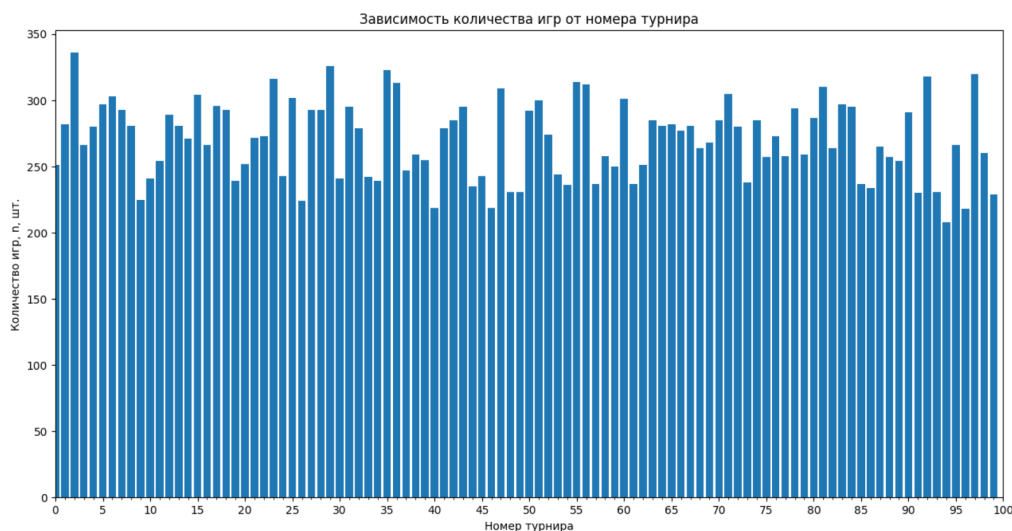


Рис. 21: Диаграмма запроса 5а)

Текст запроса: б) Посчитать число игр в каждом туре турнира А.

Код запроса:

```

1 SELECT _Round.id_round AS id_round, _Round._number AS
   _round_number, COUNT(*) AS number_of_plays
2 FROM Competition
3   JOIN _Round ON Competition.id_comp = _Round.id_comp
4   JOIN Completed_play ON Completed_play.id_round = _Round.
   id_round
5 WHERE Competition.id_comp = 1
6 GROUP BY _Round.id_round;
```

На рис. 22 приведен результат выполнения запроса. Время выполнения – 0.000 сек. На рис. 23 приведен Explain запроса 5б).

	id_round	_round_number	number_of_plays
▶	8	0	34
	9	1	31
	10	2	28
	11	3	30
	12	4	39
	13	5	27
	14	6	31
	15	7	34
	16	8	28

Рис. 22: Результат выполнения запроса 5б)

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	Competition	NULL	const	PRIMARY	PRIMARY	4	const	1	100.00	Using index
	1	SIMPLE	_Round	NULL	ref	PRIMARY_id_comp	id_comp	5	const	9	100.00	Using index
	1	SIMPLE	Completed_play	NULL	ref	id_round	id_round	5	chessbase._Round.id_round	29	100.00	Using index

Рис. 23: Explain запроса 5б)

Запрос во многом аналогичен пункту а), однако при слиянии с таблицей Тур также указывается условие соответствия id турнира требуемому.

На рис. 24 приведена диаграмма, построенная для результата выполнения запроса.

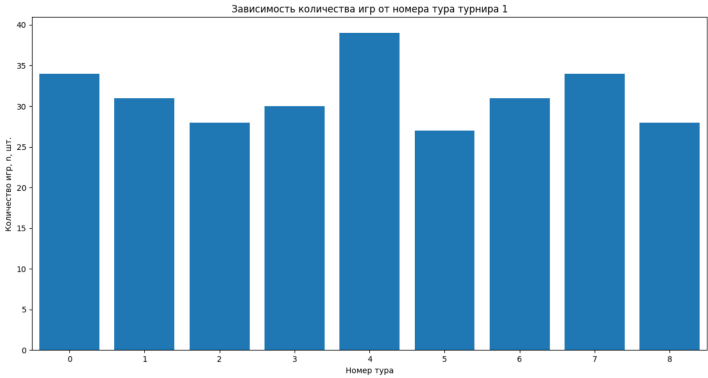


Рис. 24: Диаграмма запроса 5б)

3.6 Запрос 6

Текст запроса: Посчитать количество шахматистов с одинаковым числом побед.

Код запроса:

```
1 SELECT num_of_wins , COUNT(*)
2 FROM ((
```

```
3 SELECT Chessplayer.id_chessplayer AS id , COUNT(
4   Completed_play.id_complet_play) as num_of_wins
5 FROM Chessplayer
6   LEFT JOIN Completed_play ON Chessplayer.id_chessplayer
7   = Completed_play.id_chessplayer_white AND Completed_play
8   .result = "1-0"
9 GROUP BY Chessplayer.id_chessplayer
10 ) UNION ALL (
11 SELECT Chessplayer.id_chessplayer AS id , COUNT(
12   Completed_play.id_complet_play) as num_of_wins
13 FROM Chessplayer
14   LEFT JOIN Completed_play ON Chessplayer.id_chessplayer
15   = Completed_play.id_chessplayer_black AND Completed_play
16   .result = "0-1"
17 GROUP BY Chessplayer.id_chessplayer ) ) AS T
18 GROUP BY num_of_wins
19 ORDER BY num_of_wins;
```

На рис. 25 приведен результат выполнения запроса. Время выполнения – 0.50 сек. На рис. 26 приведен Explain запроса 6.

	num_of_wins	COUNT(*)
▶ 1	525	
2	856	
3	1051	
4	964	
5	684	
6	414	
7	205	
8	111	
9	30	
10	19	
11	3	
13	2	

Рис. 25: Результат выполнения запроса 6

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶ 1	PRIMARY	<derived2>		ALL					5367	100.00	Using temporary
2	DERIVED	Chessplayer		index	PRIMARY	PRIMARY	8		5070	100.00	Using index
2	DERIVED	Completed_play		ALL	id_chessplayer_white_id_ch...				26459	0.00	Range checked for each record (index map: 0x6)

Рис. 26: Explain запроса 6

Запрос представляет собой создание промежуточной таблицы **wins**, в которой для каждого шахматиста подсчитывается его число побед, по-

сле чего строится таблица, в которой для каждого количества побед определяется количество шахматистов, имеющих такое число побед.

На рис. 27 приведена диаграмма, построенная для результата выполнения запроса.

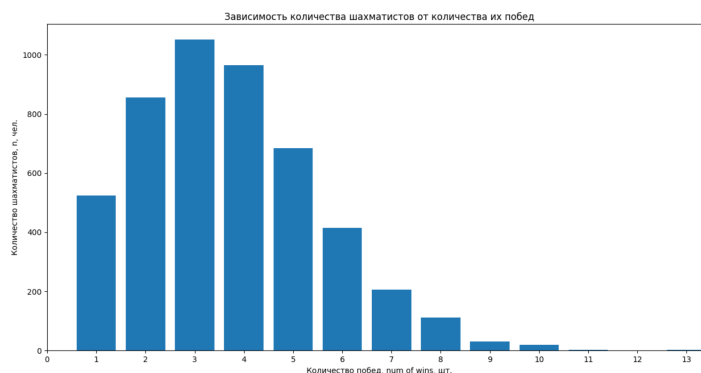


Рис. 27: Диаграмма запроса 6

3.7 Запрос 7

Текст запроса: Найти шахматистов, которые не играли в турнире А.

Код запроса:

```

1 SELECT Chessplayer.id_chessplayer AS id ,
2   Chessplayer.second_name AS second_name ,
3   Chessplayer.first_name AS first_name ,
4   Chessplayer.pathronymic AS pathronymic
5 FROM Competition
6   JOIN Participation_in_competition ON Competition.id_comp
   = 10 AND Competition.id_comp =
   Participation_in_competition.id_comp
7   RIGHT JOIN Chessplayer ON Chessplayer.id_chessplayer =
   Participation_in_competition.id_chessplayer
8 WHERE Competition.id_comp IS NULL;
```

На рис. 28 приведен результат выполнения запроса. Время выполнения – 0.07 сек, выведено 4940 записей. На рис. 29 приведен Explain запроса 7.

	id	second_name	first_name	pathronymic
▶	1003305	Курочкин	Павел	Геннадьевич
	1004990	Медведев	Леонид	Владимирович
	1006497	Глазунов	Никон	Антонович
	1006531	Потапов	Антон	Сергеевич
	1007148	Вист	Никон	Васильевич
	1007321	Петров	Владимир	Антонович
	1008654	Смирнов	Борис	Владимирович
	1010716	Петров	Никита	Сергеевич
	1013908	Никонов	Марк	Андреевич
	1014048	Блинов	Валерий	Семенович
	1016266	Никонов	Арсен	Павлович
	1020355	Никонов	Геннадий	Николаевич
	1023424	Волков	Марк	Александрович
	1023577	Лосев	Петр	Иванович
	1024443	Фомин	Самуил	Сергеевич
	1026765	Блинов	Николай	Павлович
	1032003	Лосев	Михаил	Николаевич
	1033372	Денисов	Николай	Васильевич
	1038543	Глазунов	Николай	Антонович
	1038756	Медведев	Николай	Алексеевич
	1041555	Никонов	Яков	Владимирович
	1047041	Богданов	Андрей	Алексеевич
	1050410	Никонов	Борис	Павлович
	1050647	Вист	Михаил	Николаевич
	1051926	Зеликман	Василий	Аронович
	1053316	Богданов	Петр	Геннадьевич

Рис. 28: Результат выполнения запроса 7 (начало вывода)

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	Chessplayer	ALL						5070	100.00	
	1	SIMPLE	Competition	const	PRIMARY	PRIMARY	PRIMARY	4	const	1	100.00	Using where; Using index
	1	SIMPLE	Participation_in_competition	ref		id_chessplayer,id_comp	id_chessplayer	9	chessbase.Chessplayer.id_chesspla...	1	100.00	Using where

Рис. 29: Explain запроса 7

В запросе выполняется соединение таблиц Турнир и Участие в турнире (с указанием условия – id турнира A), после чего выполняется RIGHT JOIN с таблицей Шахматист по условию – id шахматиста соответствует id участия. Таким образом, построена таблица, в которой указаны все шахматисты, которым в случае их участия в турнире A соответствует id этого турнира (и id участия в турнире); если же шахматист в турнире не участвовал, то на месте id турнира стоит NULL. Это значит, что достаточно из получившейся таблицы извлечь всех шахматистов, у которых на указанной позиции стоит NULL.

3.8 Запрос 8

Текст запроса: Посчитать число игр для каждой квалификационной категории судей по каждой категории ФИДЕ.

Код запроса:

```

1 SELECT Qualif_cat.qualif_cat AS Qualification_category ,
2   FIDE_cat.category AS FIDE_category ,
3   COUNT(Completed_play.id_complet_play) AS num
4 FROM FIDE_cat
5   JOIN Qualif_cat
6   LEFT JOIN Arbiter ON Arbiter.qualif_cat = Qualif_cat.
   id_qualif_cat
7   LEFT JOIN Completed_play ON Completed_play.id_arbiter =
   Arbiter.id_arbiter
8 GROUP BY Qualif_cat.id_qualif_cat , FIDE_cat.id_cat
9 ORDER BY Qualif_cat.id_qualif_cat , FIDE_cat.id_cat;
```

На рис. 30 приведен результат выполнения запроса. Время выполнения – 0.09 сек. На рис. 31 приведен Explain запроса 8.

	Qualification_category	FIDE_category	num
▶	A		134
	A	1	299
	A	2	582
	A	3	410
	A	4	457
	A	5	664
	A	6	205
	A	7	98
	A	8	102
	A	9	403
	A	10	230
	A	11	196
	A	12	212
	A	13	337
	A	14	158
	A	15	166
	A	16	232
	A	17	93
	A	18	84
	A	19	209
	A	20	301
	A	21	379
	A	22	494
	A	23	387
	B		106
	B	1	234

Рис. 30: Результат выполнения запроса 8 (начало вывода)

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
►	1	SIMPLE	Qualif_cat	10000	ALL	qualif_cat	qualif_cat	10000	qualif_cat	5	100.00	Using temporary; Using filesort
	1	SIMPLE	FIDE_cat	10000	ALL	qualif_cat	qualif_cat	10000	qualif_cat	24	100.00	Using join buffer (hash join)
	1	SIMPLE	Arbiter	10000	ref	qualif_cat	qualif_cat	2	chessbase.Qualif_cat.id_qualif_cat	375	100.00	Using index
	1	SIMPLE	Completed_play	10000	ref	id_arbiter	id_arbiter	5	chessbase.Arbiter.id_arbiter	27	100.00	Using index

Рис. 31: Explain запроса 8

В запросе необходимо получить все сочетания квалификационных категорий судей и категорий ФИДЕ турниров. Для этого выполняется соединение таблицы Категория ФИДЕ с таблицей Квалификационная категория, которая соединяется последовательно с помощью LEFT JOIN с таблицами Арбитр и Завершенная игра. В результате для каждой категории ФИДЕ и квалификационной категории подсчитывается количество завершенных игр (ненулевых id).

На рис. 32 приведена диаграмма, построенная для результата выполнения запроса.

Зависимость количества игр от категории ФИДЕ турнира
и квалификационной категории арбитра

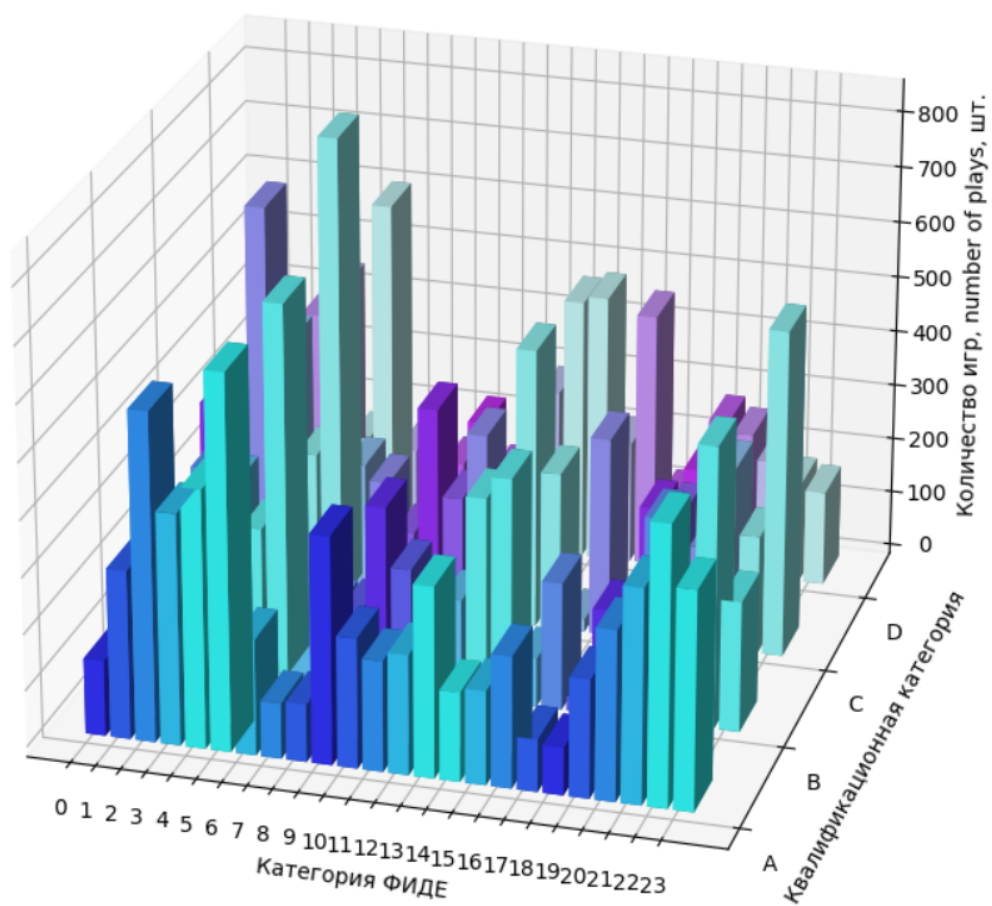


Рис. 32: Диаграмма запроса 8

Заключение

В ходе данной работы были получены навыки работы с MySQL 8.0, изучен его синтаксис и особенности. Также были получены навыки работы в среде MySQL Workbench, построения ER-диаграмм и диаграмм объектов, навыки проектирования базы данных, навыки анализа вывода команды EXPLAIN.

В рамках разработки базы данных шахматных партий была разработана ER-диаграмма, описывающая процессы, протекающие в предметной области, а также схема объектов.

На основе построенных диаграмм была спроектирована и реализована база данных шахматных партий. База содержит 12 таблиц – 6 словарей первого уровня, 1 словарь второго уровня, 1 словарь третьего и 1 словарь четвертого уровня. База была заполнена случайными данными, сгенерированными программой на языке программирования Python 3.8 (программа приведена в приложении Б). Заполненная база данных содержит порядка 104000 записей.

Были реализованы 8 запросов к данной базе данных. К нескольким из них также были построены диаграммы. К каждому из запросов приложен вывод EXPLAIN для него. Также указано время выполнения для каждого из запросов.

На основе выводов EXPLAIN и измеренного времени выполнения запросов можно сделать вывод о том, что к недостаткам реализации базы данных относится сложность запросов, связанных с определением выигранных шахматистом партий.

Список литературы

- [1] FIDE HANDBOOK. – URL: <https://handbook.fide.com>. — (дата обращения: 04.04.2021).
- [2] MySQL Documentaion. // MySQL 8.0 Reference Manual. — URL: <https://dev.mysql.com/doc/>. — (дата обращения: 07.05.2021).
- [3] Руководство по MySQL. – URL: <https://metanit.com/sql/mysql/>. — (дата обращения: 07.05.2021).

Приложение А. Создание базы данных

```
1 CREATE DATABASE chessbase;
2
3 USE chessbase;
4
5 CREATE TABLE FIDE_rank (
6     id_rank TINYINT PRIMARY KEY,
7     _rank ENUM("GM", "IM", "FM", "CM", "WGM", "WIM", "WFM", "
8     ")
9 );
10
11 CREATE TABLE Qualif_cat (
12     id_qualif_cat TINYINT PRIMARY KEY,
13     qualif_cat ENUM("A", "B", "C", "D")
14 );
15
16 CREATE TABLE FIDE_cat (
17     id_cat TINYINT PRIMARY KEY,
18     category ENUM("", "1", "2", "3", "4", "5", "6", "7", "8",
19     "9", "10", "11", "12", "13", "14", "15", "16", "17",
20     "18", "19", "20", "21", "22", "23", "24")
21 );
22
23 CREATE TABLE Chessplayer (
24     id_chessplayer BIGINT UNSIGNED PRIMARY KEY,
25     second_name VARCHAR(30),
26     first_name VARCHAR(30),
27     pathronymic VARCHAR(30),
28     year_of_birth YEAR
29 );
30
31 CREATE TABLE Moves (
32     id_moves INTEGER PRIMARY KEY,
33     string_note TEXT,
34     debut_code CHAR(3)
35 );
36
37 CREATE TABLE _Time (
38     id_time INTEGER PRIMARY KEY,
39     white_minutes INTEGER,
40     white_seconds INTEGER,
41     black_minutes INTEGER,
```

```
40  black_seconds INTEGER
41 );
42
43 CREATE TABLE Arbiter (
44  id_arbiter INTEGER PRIMARY KEY,
45  second_name VARCHAR(30),
46  first_name VARCHAR(30),
47  pathronimic VARCHAR(30),
48  qualif_cat TINYINT,
49
50  FOREIGN KEY (qualif_cat) REFERENCES Qualif_cat(
51    id_qualif_cat)
52 );
53
54 CREATE TABLE Competition (
55  id_comp INTEGER PRIMARY KEY,
56  _name VARCHAR(30),
57  _date DATE,
58  _format VARCHAR(30),
59  time_control VARCHAR(30),
60  number_of_rounds INTEGER,
61  FIDE_category TINYINT,
62  id_chief_arbiter INTEGER,
63
64  FOREIGN KEY (FIDE_category) REFERENCES FIDE_cat(id_cat),
65  FOREIGN KEY (id_chief_arbiter) REFERENCES Arbiter(
66    id_arbiter)
67 );
68
69 CREATE TABLE _Round (
70  id_round INTEGER PRIMARY KEY,
71  _date DATE,
72  _number TINYINT UNSIGNED,
73  id_comp INTEGER,
74
75  FOREIGN KEY (id_comp) REFERENCES Competition(id_comp)
76 );
77
78 CREATE TABLE Participation_in_competition (
79  id_participation INTEGER PRIMARY KEY,
80  ELO_rating INTEGER UNSIGNED,
81  id_chessplayer BIGINT UNSIGNED,
82  id_comp INTEGER,
```

```
81 FIDE_rank TINYINT ,
82
83 FOREIGN KEY (FIDE_rank) REFERENCES FIDE_rank(id_rank) ,
84 FOREIGN KEY (id_chessplayer) REFERENCES Chessplayer(
    id_chessplayer) ,
85 FOREIGN KEY (id_comp) REFERENCES Competition(id_comp)
86 );
87
88 CREATE TABLE Refereeing (
89     id_refereeing INTEGER PRIMARY KEY ,
90     id_comp INTEGER ,
91     id_arbiter INTEGER ,
92
93     FOREIGN KEY (id_comp) REFERENCES Competition(id_comp) ,
94     FOREIGN KEY (id_arbiter) REFERENCES Arbiter(id_arbiter)
95 );
96
97 CREATE TABLE Includ_round (
98     id_includ_round INTEGER PRIMARY KEY ,
99     number_of_rounds INTEGER ,
100    id_round INTEGER ,
101    id_comp INTEGER ,
102
103    FOREIGN KEY (id_round) REFERENCES _Round(id_round) ,
104    FOREIGN KEY (id_comp) REFERENCES Competition(id_comp)
105 );
106
107 CREATE TABLE Completed_play (
108     id_complet_play INTEGER PRIMARY KEY ,
109     result VARCHAR(30) ,
110     id_chessplayer_white BIGINT UNSIGNED ,
111     id_chessplayer_black BIGINT UNSIGNED ,
112     id_round INTEGER ,
113     id_arbiter INTEGER ,
114     id_time INTEGER ,
115     id_moves INTEGER ,
116
117     FOREIGN KEY (id_chessplayer_white) REFERENCES Chessplayer
        (id_chessplayer) ,
118     FOREIGN KEY (id_chessplayer_black) REFERENCES Chessplayer
        (id_chessplayer) ,
119     FOREIGN KEY (id_round) REFERENCES _Round(id_round) ,
120     FOREIGN KEY (id_arbiter) REFERENCES Arbiter(id_arbiter) ,
```



```
121 FOREIGN KEY (id_time) REFERENCES _Time(id_time),  
122 FOREIGN KEY (id_moves) REFERENCES Moves(id_moves)  
123 );
```

Приложение Б. Генерация данных для заполнения базы данных

```
1 from random import *
2 from chess import *
3
4 FIDE_CATS = [str(i) for i in range(1, 26)]
5
6 def genNames(n: int) -> List[Tuple[str, str, str]]:
7     snames = ["Смирнов", "Иванов", "Кузнецов", "Волобуев", "
8         Медведев", "Волков", "Зайцев", "Зеликман", "Блинов", "
9         Денисов",
10        "Потапов", "Фомин", "Гусев", "Щукин", "Никонов", "Глазунов",
11        "Петров", "Вист", "Лосев", "Никитин", "Курочкин", "Богданов"]
12    fnames = ["Василий", "Владимир", "Иван", "Михаил", "Павел",
13        "Петр", "Яков", "Марк", "Арсений", "Никита", "Семен", "
14        Валерий",
15        "Андрей", "Антон", "Алексей", "Самуил", "Борис", "Сергей", "
16        Арсен", "Леонид", "Никон", "Александр", "Геннадий", "Николай"
17    ]
18    patrs = ["Васильевич", "Владимирович", "Иванович", "
19        Михайлович", "Павлович", "Петрович", "Аронович", "Семенович",
20        "Валерьевич",
21        "Андреевич", "Антонович", "Алексеевич", "Сергеевич", "
22        Геннадьевич", "Николаевич", "Александрович"]
23
24    return [(choice(snames), choice(fnames), choice(patrs))
25            for _ in range(n)]
26
27 def genIds(n: int, min: int, max: int) -> List[int]:
28     res = []
29     while len(res) < n:
30         new = randint(min, max)
31         if new not in res:
32             res.append(new)
33
34     return res
35
36 def genDate() -> str:
37     return str(genYear(1900, 2020)) + '-' + str(randint(1,
38         12)) + '-' + str(randint(1, 28))
39
```

```

28 def genYear(min, max) -> int:
29     return randint(min, max)
30
31 def genPlayers(n: int):
32     names = genNames(n)
33     ids = [1503014] + genIds(n-1, 1000000, 9500000)
34     ys = [genYear(1940, 2005) for _ in range(n)]
35
36     return [(str(ids[i]), names[i][0], names[i][1], names[i]
37             ][2], str(ys[i])) for i in range(n)]
38
39 def genArbiters(n: int):
40     names = genNames(n)
41     qcs = [randint(0, 3) for _ in range(n)]
42
43     return [(str(names[i][0]), str(names[i][1]), str(names[i]
44             ][2]), str(qcs[i])) for i in range(n)]
45
46 def genCompetition(n: int, arbiters):
47     fns = ["Кубок", "Приз", "Чемпионат", "Турнир", "Встреча"]
48     sns = ["города Москва", "весенних каникул", "осенних каникул", "
49             города Васюки", "станции Бологое"]
50
51     names = [choice(fns) + ' ' + choice(sns) for _ in range(n)
52             ]
53     dates = [genDate() for _ in range(n)]
54     formats = [choice(["round-robin", "swiss"]) for _ in
55                range(n)]
56     times = [choice(['90+30', '60+30', '15+10', '10+0', '2+0'
57                     ]) for _ in range(n)]
58     nrounds = [10 + randint(-2, 2) for _ in range(n)]
59
60     return [(names[i], dates[i], formats[i], times[i], str(
61             nrounds[i]), str(randint(0, 23)), str(choice(arbiters)))
62             for i in range(n)]
63
64 def genRounds(n, id):
65     return [(genDate(), str(i), str(id)) for i in range(n)]
66
67 def genTime(n):
68     return [[str(i), str(randint(1, 30)), str(randint(0, 59))
69             , str(randint(1, 30)), str(randint(0, 59))] for i in
70             range(n)]

```

```

61
62 def genMoves(n, whiteWin):
63     def _randMoves():
64         b = Board()
65         b.reset()
66
67         n_moves = 35 + randint(-10, 10)
68         result = ''
69
70         for i in range(1, n_moves + 1):
71             if b.legal_moves.count() == 0:
72                 break
73             m = choice(list(b.legal_moves))
74             result += str(i) + '. ' + str(m) + ' '
75             b.push(m)
76
77             if i == n_moves and whiteWin:
78                 break
79
80             if b.legal_moves.count() == 0:
81                 break
82
83             m = choice(list(b.legal_moves))
84             result += str(m) + ' '
85             b.push(m)
86
87         return result, choice('ABCDE') + str(choice([i for i in
88             range(10, 100)]))
89
90     return [_randMoves() for _ in range(n)]
91
92 def createPlayers(n):
93     ps = genPlayers(n)
94     with open('players.txt', 'w') as f:
95         f.writelines(['\t'.join(p) + '\n' for p in ps])
96
97     return ps
98
99 def createArbs(n):
100     ars = genArbiters(n)
101     with open('arbs.txt', 'w') as f:
102         f.writelines([str(p) + '\t' + '\t'.join(ars[p]) + '\n'
103             for p in range(len(ars))])

```

```

102
103 def createComps(n, arbiters):
104     cs = genCompetition(n, arbiters)
105     with open('comps.txt', 'w') as f:
106         f.writelines([str(c) + '\t' + '\t'.join(cs[c]) + '\n'
107                        for c in range(len(cs))])
108
109 def createTimes(n):
110     times = genTime(n)
111     with open('times.txt', 'w') as f:
112         f.writelines(['\t'.join(times[p]) + '\n' for p in range
113                        (len(times))])
114
115 def createMoves(n):
116     white = randint(0, n/2)
117     with open('moves.txt', 'w') as f:
118         w1 = genMoves(n/2, white/2)
119         w2 = genMoves(n/2, (n-white)/2)
120         f.writelines([str(p) + '\t' + '\t'.join(w1[p]) + '\n'
121                        for p in range(len(w1))] + ['\t'.join(w2[p]) + '\n' for
122                        p in range(len(w2))])
123
124 def createRanks():
125     ranks = ["GM", "IM", "FM", "CM", "WGM", "WIM", "WFM", ""]
126     with open('ranks.txt', 'w') as f:
127         f.writelines([str(p) + '\t' + ranks[p] + '\n' for p in
128                        range(8)])
129
130 def create_Qualifs():
131     cats = ["A", "B", "C", "D"]
132     with open('qualif_cats.txt', 'w') as f:
133         f.writelines([str(p) + '\t' + cats[p] + '\n' for p in
134                        range(4)])
135
136 def create_Fides():
137     cats = [""] + [str(i) for i in range(25)]
138     with open('FIDE_cats.txt', 'w') as f:
139         f.writelines([str(p) + '\t' + cats[p] + '\n' for p in
140                        range(25)])
141
142 def createRounds(ids):
143     # ids — колво— турниров
144     roundIdx = 0

```

```

138 ns = [randint(8, 10) for i in range(ids)]
139 rounds = []
140 with open('rounds.txt', 'w') as f:
141     for i in range(ids):
142         for p in genRounds(ns[i], i):
143             f.write(str(roundldx) + '\t' + '\t'.join(p) + '\n')
144             rounds.append((roundldx, p[2]))
145             roundldx += 1
146
147     return roundldx + 1, rounds
148
149
150 def createParticipations(n, players_ids):
151     parts = [[str(i), str(randint(1800, 2100)), str(choice(
152         players_ids)[0]),
153         str(randint(0, 99)), str(randint(0, 7))]] for i in range(n
154         )]
155     with open('partic.txt', 'w') as f:
156         f.writelines(['\t'.join(p) + '\n' for p in parts])
157
158
159 def createRefereeing(n):
160     arbsld = [i for i in range(1500)]
161     refs = [[str(i), str(randint(0, 99)), str(arbsld.pop(
162         randint(0, len(arbsld) - 1)))] for i in range(n)]
163     with open('refs.txt', 'w') as f:
164         f.writelines(['\t'.join(p) + '\n' for p in refs])
165
166     return refs
167
168
169 def createCompletedPlays(n, pids, arbs, rounds):
170
171     def getArb(round):
172         compld = None
173         for r in rounds:
174             if r[0] == round:
175                 compld = r[1]
176                 break
177         return [i[2] for i in arbs if i[1] == compld]
178
179     plays = []
180     for i in range(n):
181         round = randint(1, n-1) % len(rounds)

```

```

178     plays.append([str(i), choice(['1-0', '0-1', '1/2-1/2'])
179                  , choice(pids)[0], choice(pids)[0], str(round),
180                  str(choice(getArb(round))), str(randint(0, 29999)), str
181                  (randint(0, 29999))])
182
183 with open('plays.txt', 'w') as f:
184     f.writelines(['\t'.join(p) + '\n' for p in plays])
185
186 if __name__ == '__main__':
187     createTimes(30000)
188     pids = createPlayers(5000)
189     createMoves(30000)
190     create_Qualifs()
191     createRanks()
192     create_Fides()
193
194     createArbs(1500)
195     createComps(100, [i for i in range(1500)])
196
197     createParticipations(6000, pids)
198     arbs = createRefereeing(1000)
199     nrounds, rounds = createRounds(100)
200     createCompletedPlays(nrounds * 30, pids, arbs, rounds)

```