

Министерство образования и науки РФ
Санкт-Петербургский Политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа искусственного интеллекта
Направление 02.03.01 «Математика и компьютерные науки»
Дисциплина «Теоретические основы баз данных»

Отчёт по курсовой работе
«Создание базы данных птиц Санкт-Петербурга и Ленинградской области»

Студент:

Башарина Е.А., гр. 3530201/90101

Преподаватель:

Попов С.Г.

Санкт-Петербург — 2022

Содержание

1	Описание предметной области	2
2	Выделение сущностей и их атрибутов	3
3	ER-диаграмма	5
3.1	Чтение ER-диаграммы	6
3.2	Схема объектов	7
4	Схема базы данных	8
4.1	Атрибуты таблиц базы данных	8
5	Программирование	15
5.1	Создание базы данных	15
5.2	Заполнение таблиц базы данных	15
6	Запросы к базе данных	17
6.1	Запрос 1	17
6.1.1	Запрос 1.1	19
6.2	Запрос 2	21
6.3	Запрос 3	22
6.3.1	Запрос 3.1	23
6.4	Запрос 4	24
6.5	Запрос 5	26
6.6	Запрос 6	27
6.7	Запрос 7	30
6.7.1	Запрос 7.1	31
6.8	Запрос 8	32
	Заключение	35
	Приложение А	36
	Приложение Б	38

1 Описание предметной области

Рассматриваемая предметная область - видовое разнообразие *Aves* (птиц) Санкт-Петербурга и Ленинградской области.

Цель функционирования - определение вида птицы.

Каждый таксон птиц обладает некоторыми уникальными морфологическими особенностями, уникальным рисунком песни, и занимает строго определённый ареал. Эти характеристики, как правило, используются при определении вида или рода птицы «в поле», без доступа к какому-либо специальному оборудованию. На территории города Санкт-Петербурга и Ленинградской области в дикой природе встречаются представители более чем 22 семейств птиц, включающих в себя до сотни видов каждое. Широкое видовое разнообразие делает актуальным вопрос о возможности полевого определения особей как профессиональными орнитологами, так и любителями, и об инструментах, способных облегчить и упростить этот процесс. Для создания подобных инструментов необходимо наличие базы данных, включающей в себя информацию о наиболее часто встречающихся видах: морфологические особенности, типичная среда обитания, пример пения, время активности.

В некоторых случаях оказывается невозможным однозначно определить особь до вида, и тогда применяется определение до таксонов более высокого уровня (например, рода: запись *Sylvia sp.* будет означать, что встречена особь из рода славков). Виды объединяются в рода, рода - в семейства, для каждого из которых также существуют специфические особенности, характерные всем их представителям.

К морфологическим особенностям относятся:

- окрас оперения;
- средний размер (или диапазон размеров) особи;
- форма клюва;
- форма хвоста;
- относительная длина маховых и рулевых перьев (по отношению к размеру туловища и длине хвоста).

При наличии у описываемого вида полового диморфизма (существенных отличий внешнего вида у особей разного пола) должно присутствовать описание морфологии самцов и самок по отдельности.

Признак **среды обитания** включает в себя перечень условий, в которых наиболее часто встречаются представители данного вида (например, луг, подлесок, пойма, береговая линия крупных водоёмов, etc).

Одним из самых удобных в использовании и однозначных признаков, позволяющих определить вид птицы, является её **пение**. Рисунок песни, как правило, уникален для каждого вида, и позволяет отличить друг от друга даже близкородственные виды: так, например, морфологические различия между пеночкой-теньковкой (*Phylloscopus collybita*) и пеночкой-весничкой (*Phylloscopus trochilus*) невелики, однако их пение существенно различается. Поэтому в базе данных, предназначенной для полевого определения вида птицы, необходимо

также наличие примеров пения и крика каждого из описываемых видов.

Время активности птицы - это некоторый временной интервал, позволяющий судить о том, в какое время суток особей рассматриваемого вида наиболее вероятно встретить в дикой природе.

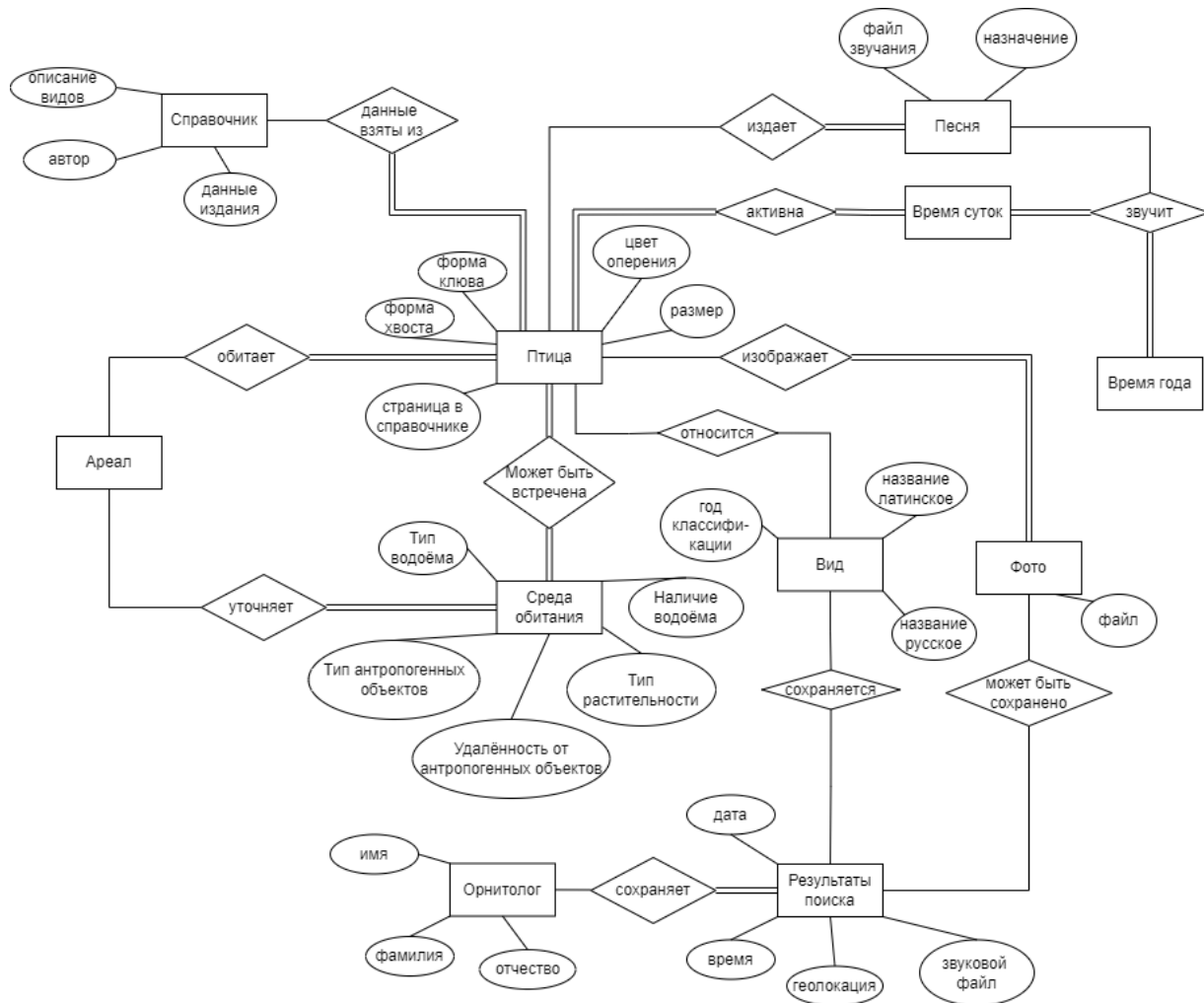
2 Выделение сущностей и их атрибутов

Выделены следующие сущности:

1. Птица - особь, животное, которое подлежит определению. Каждая птица обладает следующими атрибутами:
 - окрас оперения;
 - размер туловища;
 - страница в справочнике;
 - форма клюва;
 - форма хвоста;
2. Вид - таксон младшего уровня в классической иерархической системе, к которому относятся особи птиц. Название вида имеет следующие атрибуты:
 - латинское название;
 - русское (тривиальное) название;
 - год классификации;
3. Среда обитания - совокупность внешних факторов, характеризующих место в пространстве, на котором была встречена птица. Имеет следующие атрибуты:
 - тип растительности;
 - удаление от водоёма;
 - тип водоёма;
 - удаление от антропогенных объектов;
 - тип антропогенных объектов;
4. Ареал - географическая область, в границах которой наиболее вероятно встретить особей того или иного вида.
5. Песня - совокупность звуков (мелодия, свист, крик, etc), которые издаёт птица. Имеет следующие атрибуты:
 - файл звучания;
 - назначение;
6. Орнитолог - человек, определяющий вид птицы. Обладает следующими атрибутами:
 - фамилия;

- имя;
 - отчество;
7. Справочник - письменный источник, содержащий фиксированное формальное описание видов птиц с указанием всех характерных особенностей. Обладает следующими атрибутами:
- описание видов;
 - автор;
 - данные издания;
8. Время суток - точное время или временной интервал, в течение которого производятся наблюдения за активными птицами.
9. Время года - диапазон дат, относящийся к одному из четырех общепринятых периодов (весна, лето, осень, зима).
10. Фото - файл с фотографиями типовых представителей вида. Имеет атрибут:
- файл;
11. Результаты поиска - набор данных, предоставляемых орнитологу после выполнения поиска вида птицы. Имеют следующие атрибуты:
- звуковой файл;
 - дата;
 - время;
 - геолокация.

3 ER-диаграмма



3.1 Чтение ER-диаграммы

- Птица обитает в ареале.
- Птица относится к виду.
- Данные о птице взяты из справочника.
- Птица издаёт песню.
- Птица активна в определённое время суток.
- Песня звучит в определённое время суток.
- Песня звучит в определённое время года.
- Птица может быть встречена в среде обитания.
- Ареал уточняет среду обитания.
- Птица относится к виду.
- Вид сохраняется в результатах поиска.
- Орнитолог сохраняет результаты поиска.
- В результатах поиска может быть сохранено фото.
- Фото изображает птицу.

3.2 Схема объектов

На рис. 1 представлена схема объектов базы данных, отражающая их количественные связи.

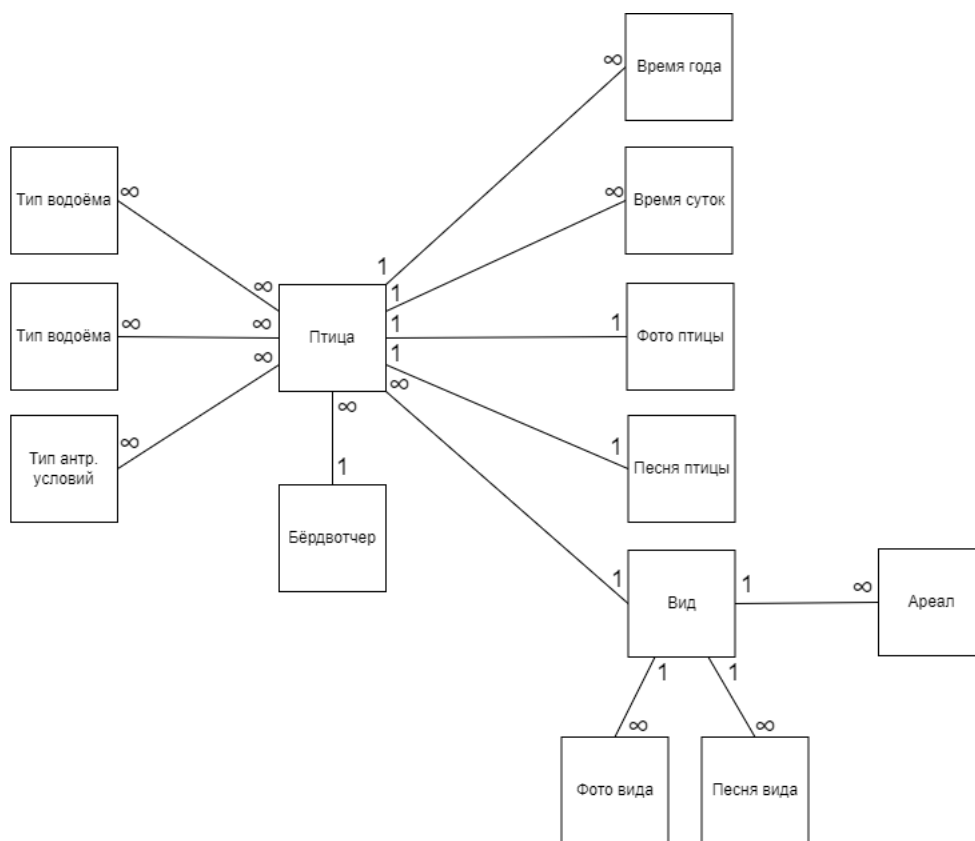


Рис. 1: Схема объектов

4 Схема базы данных

На рис. 2 изображена схема базы данных, построенная в соответствии с ER-диаграммой. Для реализации была создана упрощённая схема базы данных - без учётов результатов поиска, - на русском (рис. 3) и английском (рис. 4) языках.

4.1 Атрибуты таблиц базы данных

В Таблицах 1 - 16 представлены атрибуты каждой таблицы проектируемой базы данных с указанием типа данных каждого поля. В случае, если поле представляет собой ключ, указывается также его тип (PK – собственный ключ, FK – внешний ключ), а также таблица, на которую он ссылается.

№	Название рус.	Название англ.	Тип	Тип ключа	Ссылка
1	id_птицы	id_bird	INT	PK	-
2	Цвет оперения	Feathers color	VARCHAR(30)	-	-
3	Форма клюва	Beak shape	VARCHAR(30)	-	-
4	Размер	Size	VARCHAR(30)	-	-
5	Форма хвоста	Tail shape	VARCHAR(30)	-	-
6	Наличие водоёма	Pool	BOOL	-	-
7	Удаленность от антропогенных объектов	Anthropogenic objects distance	VARCHAR(30)	-	-
8	id_вида	id_species	INT	FK	Species: id_species
9	id_бёрдвотчера	id_birdwatcher	INT	FK	Birdwatcher: id_birdwatcher

Таблица 1: Птица - Bird

№	Название рус.	Название англ.	Тип	Тип ключа	Ссылка
1	id_вида	id_species	INT	PK	-
2	Латинское название	Latin name	VARCHAR(30)	-	-
3	Русское название	Russian name	VARCHAR(30)	-	-
4	id_справочника	id_book	INT	FK	Book: id_book

Таблица 2: Вид - Species



Рис. 2: Схема базы данных

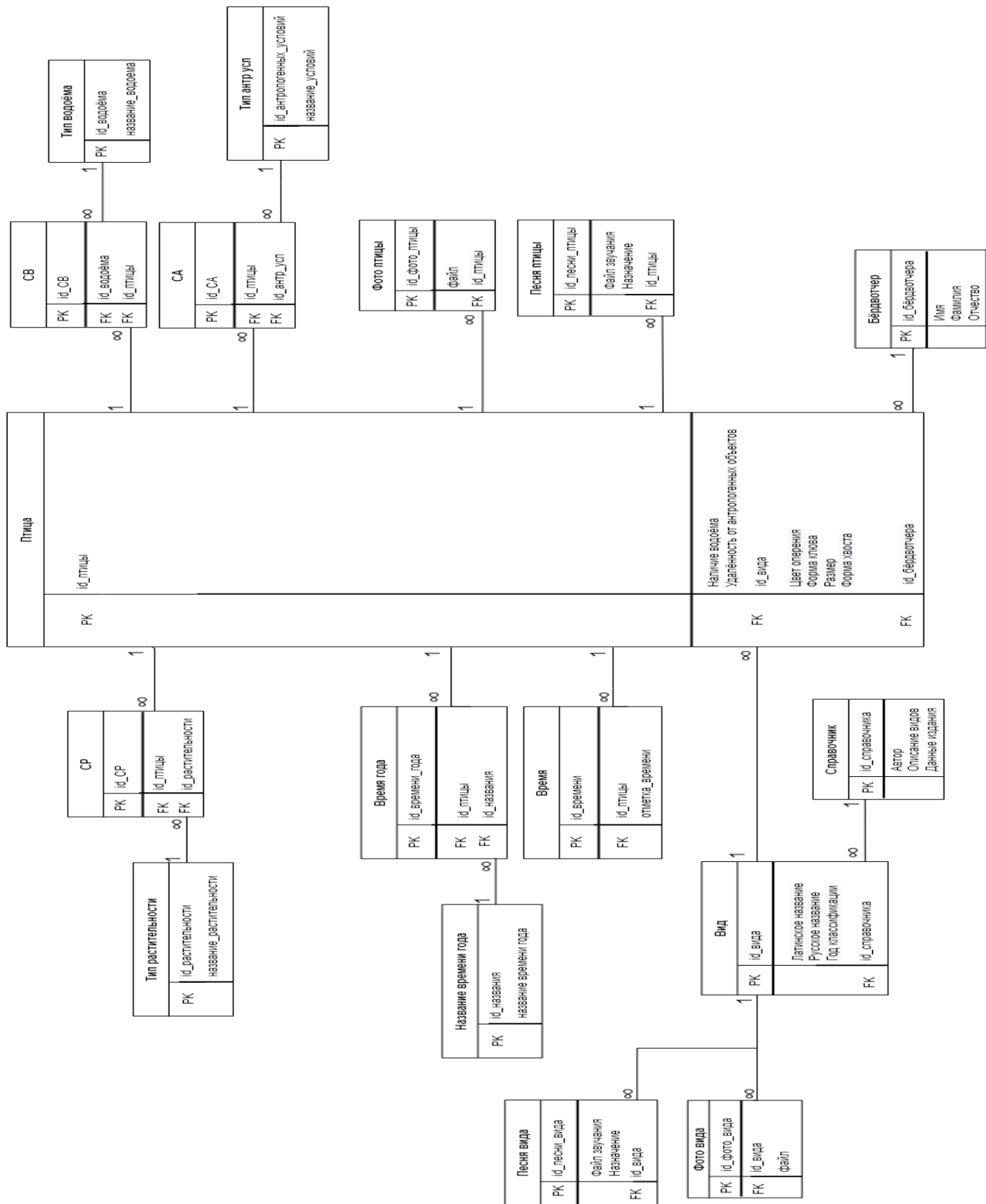


Рис. 3: Упрощенная схема базы данных на русском языке

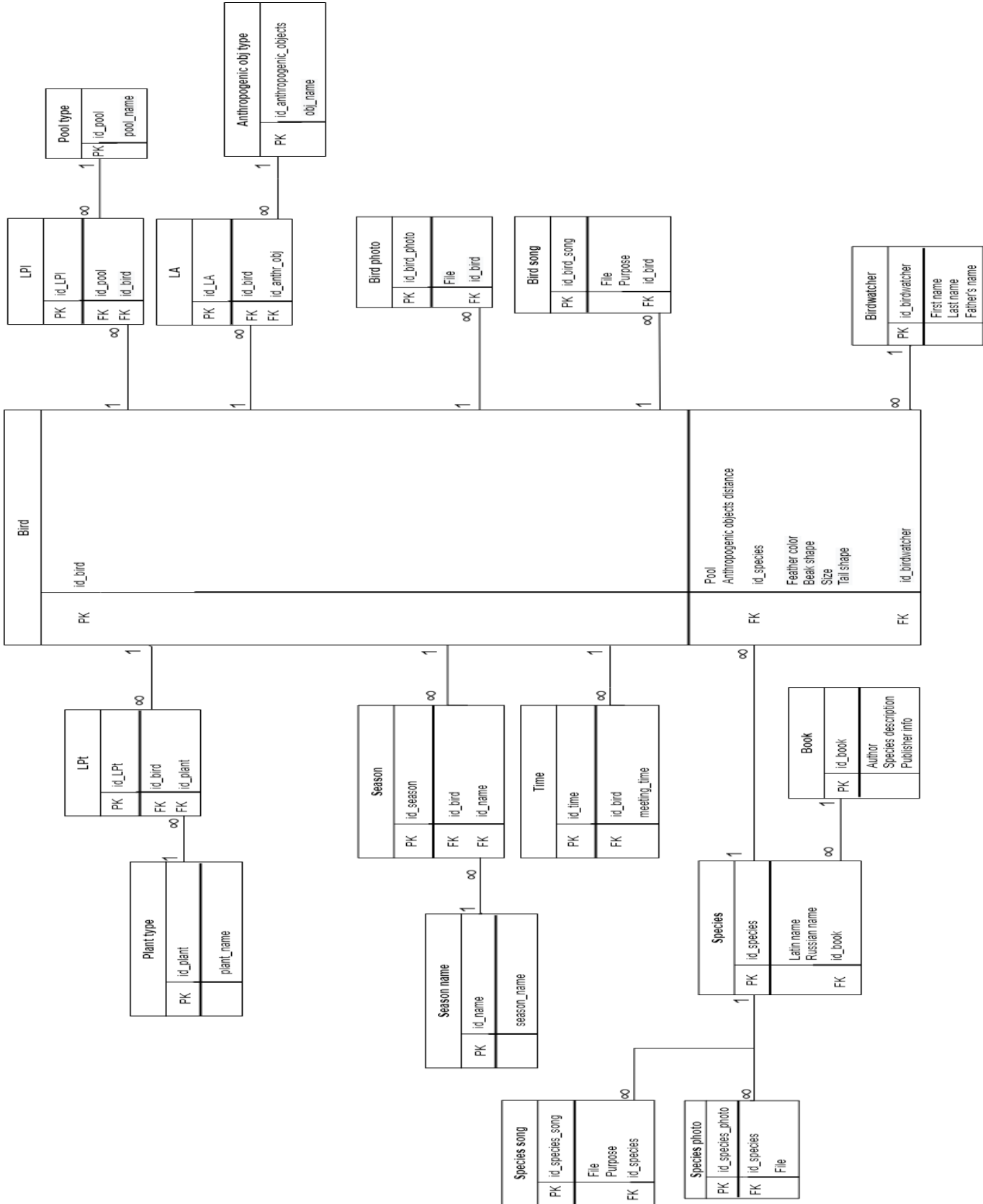


Рис. 4: Упрощенная схема базы данных на английском языке

№	Название рус.	Название англ.	Тип	Тип ключа	Ссылка
1	id_песни_вида	id_species_song	INT	PK	-
2	Файл звучания	File	TEXT	-	-
3	Назначение	Purpose	VARCHAR(30)	-	-
4	id_вида	id_species	INT	FK	Species: id_species

Таблица 3: Песня вида - Species song

№	Название рус.	Название англ.	Тип	Тип ключа	Ссылка
1	id_фото_вида	id_species_photo	INT	PK	-
2	Файл	File	TEXT	-	-
3	id_вида	id_species	INT	FK	Species: id_species

Таблица 4: Фото вида - Species photo

№	Название рус.	Название англ.	Тип	Тип ключа	Ссылка
1	id_песни_птицы	id_bird_song	INT	PK	
2	Файл звучания	File	TEXT	-	-
3	Назначение	Purpose	VARCHAR(30)	-	-
4	id_птицы	id_bird	INT	FK	Bird: id_bird

Таблица 5: Песня птицы - Bird song

№	Название рус.	Название англ.	Тип	Тип ключа	Ссылка
1	id_фото_птицы	id_bird_photo	INT	PK	-
2	Файл	File	TEXT	-	-
3	id_птицы	id_bird	INT	FK	Bird: id_bird

Таблица 6: Фото птицы - Bird photo

№	Название рус.	Название англ.	Тип	Тип ключа	Ссылка
1	id_справочника	id_book	INT	PK	-
2	Автор	Author	VARCHAR(30)	-	-
3	Описание видов	Species description	LARGETEXT	-	-
4	Данные издания	Publisher info	TEXT	-	-

Таблица 7: Справочник - Book

№	Название рус.	Название англ.	Тип	Тип ключа	Ссылка
1	id_бердвотчера	id_bird	INT	PK	-
2	Фамилия	Last name	VARCHAR(30)	-	-
3	Имя	First name	VARCHAR(30)	-	-
4	Отчество	Father's name	VARCHAR(30)	-	-

Таблица 8: Бёрдвотчер - Birdwatcher

№	Название рус.	Название англ.	Тип	Тип ключа	Ссылка
1	id_времени_года	id_season	INT	PK	-
2	id_птицы	id_bird	INT	FK	Bird: id_bird
3	название	season_name	VARCHAR(30)	-	-

Таблица 9: Время года - Season

№	Название рус.	Название англ.	Тип	Тип ключа	Ссылка
1	id_времени	id_time	INT	PK	-
2	id_птицы	id_bird	INT	FK	Bird: id_bird
3	значение	time_value	DATETIME	-	-

Таблица 10: Время - Time

№	Название рус.	Название англ.	Тип	Тип ключа	Ссылка
1	id_растительности	id_plant	INT	PK	-
2	название	plant_name	VARCHAR(30)	-	-

Таблица 11: Тип растительности - Plant type

№	Название рус.	Название англ.	Тип	Тип ключа	Ссылка
1	id_водоёма	id_pool	INT	PK	-
3	название	pool_name	VARCHAR(30)	-	-

Таблица 12: Тип водоёма - Pool type

№	Название рус.	Название англ.	Тип	Тип ключа	Ссылка
1	id_антропогенных_условий	id_anthropogenic_objects	INT	PK	-
3	название	obj_name	VARCHAR(30)	-	-

Таблица 13: Тип антропогенных условий - Anthropogenic object type

№	Название рус.	Название англ.	Тип	Тип ключа	Ссылка
1	id_СР	id_LPt	INT	PK	-
2	id_птицы	id_bird	INT	FK	-
3	id_растительности	id_plant	INT	FK	-

Таблица 14: СР (связь растительности) - LPt (link plant)

№	Название рус.	Название англ.	Тип	Тип ключа	Ссылка
1	id_СВ	id_LPl	INT	PK	-
2	id_птицы	id_bird	INT	FK	-
3	id_водоёма	id_pool	INT	FK	-

Таблица 15: СВ (связь водоёма) - LPl (link pool)

№	Название рус.	Название англ.	Тип	Тип ключа	Ссылка
1	id_СА	id_LA	INT	PK	-
2	id_птицы	id_bird	INT	FK	-
3	id_антропогенных_условий	id_anthropogenic_objects	INT	FK	-

Таблица 16: СА (связь антропогенных условий)- LA (link anthropogenic objects)

5 Программирование

5.1 Создание базы данных

Для генерации базы данных был написан скрипт в графическом клиента MySQL Workbench. Код реализации базы данных представлен в Приложении А.

Процесс создания базы данных разделялся на следующие этапы.

1. С помощью команды `CREATE DATABASE` на сервере создаётся база данных `Birds`. Для дальнейшего её использования применяется команда `USE Birds`.
2. С помощью команды `CREATE TABLE` создаются таблицы, описанные в разделе 4.1, в следующей последовательности:
 - в первую очередь создаются словари, не имеющие вторичных ключей - таблицы `Book`, `Birdwatcher`, `Pool_type`, `Plant_type`, `Season_type`, `Anthr_obj_type`;
 - затем создаются таблицы, связанные вторичными ключами только со словарями - `Species`, `Bird`;
 - в последнюю очередь создаются остальные таблицы - `Species_song`, `Species_photo`, `Bird_song`, `Bird_photo`, `Season`, `Time`, `Lpt`, `Lpl`, `LA`.

При этом в каждой из таблиц указываются первичные ключи командой `PRIMARY KEY` и вторичные ключи командами `CONSTRAINT ... FOREIGN KEY ... REFERENCES`.

3. Для первичных ключей устанавливается автоматическое увеличение значения столбца при добавлении новых записей с помощью команды `AUTO_INCREMENT`.

5.2 Заполнение таблиц базы данных

Для заполнения таблиц была реализована программа на языке Python, в которой записаны соответствующие SQL-команды. Код программы представлен в Приложении В. При этом данные разделяются на два типа:

- заранее заданные значения (например, 'Лето', 'Зима', 'Осень' и 'Весна' для словаря сезонов);
- уникальные для каждой записи значения, генерирующиеся случайным образом на основе некоторого базового правила (например, название вида птицы или имя бёрд-вотчера).

Подключение к базе данных происходит при помощи модуля **mysql.connector**, который позволяет подключиться к базе данных и отправлять SQL-запросы серверу. Данные в таблицах распределены равномерно (например, на каждую запись в таблице «Book» приходится от 1 до 10 записей в таблице «Species»). Ниже представлено количество записей в каждой из таблиц базы данных `Birds`.

Таблица	№ при заполнении	Количество записей
birdwatcher	1	50
book	1	25
season_type	1	4
plant_type	1	5
pool_type	1	5
anthr_obj_type	1	4
species	2	100
bird	2	100000
species_song	3	100
species_photo	3	100
bird_song	3	200371
bird_photo	3	149588
season	3	100000
meeting_time	3	100000
lpt	3	300137
lpl	3	299902
la	3	299841
TOTAL	-	1 550 232

6 Запросы к базе данных

В рамках данной работы требуется реализовать следующие 8 запросов к базе данных:

1. вывести всех птиц, которых видел в сезон X бёрдвотчер Y с типом водоёма Z;
 - 1.1. не менее трёх раз;
2. посчитать число птиц вида X, для которых загружены и фото, и песни;
3. найти бёрдвотчера, который наблюдал максимальное число птиц (если таких несколько, то вывести всех);
 - 3.1. найти бёрдвотчера, который наблюдал минимальное число птиц;
4. посчитать число видов с одинаковым числом птиц;
5. найти те книги, для которых не было встречено ни одной птицы;
6. для каждого сезона и каждого вида посчитать число наблюдений;
7. для всех бёрдвотчеров посчитать
 - 7.1. число птиц;
 - 7.2. число песен, которые в сумме исполнили эти птицы;
8. найти все виды птиц, число наблюдений которых больше, чем у вида X.

Для запросов 4, 6 и 7 необходимо также построить диаграммы.

6.1 Запрос 1

Формулировка запроса: вывести всех птиц, которых видел в сезон X бёрдвотчер Y с типом водоёма Z.

Код запроса на SQL:

```
SELECT species.latin_name as 'Latin_name',
species.russian_name as 'Russian_name',
meeting_time.time_name as 'Meeting_time',
bird.feath_color as 'Feather_color',
bird.size as 'Size',
bird_photo.bird_file as 'Photo',
bird_song.bird_file as 'Song'
FROM bird
INNER JOIN species on bird.id_species=species.id_species
INNER JOIN meeting_time on meeting_time.id_bird=bird.id_bird
INNER JOIN birdwatcher ON bird.id_birdwatcher=birdwatcher.id_birdwatcher
INNER JOIN season ON bird.id_bird=season.id_bird
INNER JOIN season_type ON season.id_name=season_type.id_name
INNER JOIN lpl ON lpl.id_bird=bird.id_bird
INNER JOIN pool_type ON lpl.id_pool=pool_type.id_pool
```

```

INNER JOIN bird_photo ON bird.id_bird=bird_photo.id_bird
INNER JOIN bird_song ON bird.id_bird=bird_song.id_bird
WHERE birdwatcher.last_name = 'Голубев'
AND birdwatcher.first_name = 'Ярослав'
AND birdwatcher.fathers_name = 'Георгиевич'
AND season_type.season_name = 'лето'
AND pool_type.pooltype = 'река'
GROUP BY bird.id_bird;

```

Описание запроса

Запрос состоит в связывании между собой посредством INNER JOIN следующих таблиц: птица, её песня и фото, вид, бёрдвотчер, а также словари сезонов и водоёмов. Записи результирующей таблицы группируются по ID птицы, и из них выбираются те, что сделаны летом, бёрдвотчером «Голубев Ярослав Георгиевич» у типа водоёма «река». Выводятся поля: латинское название вида, русское название вида, время встречи, цвет перьев птицы, размер птицы, а также её фотография и песня.

Время выполнения запроса

Время выполнения этого запроса составило 0.125 секунд.

На рис. 5 показан результат выполнения этого запроса, а на рис. 6 - план его выполнения.

Latin name	Russian name	Meeting time	Feather color	Size	Photo	Song
Yiqqjseurll dibtrikdhr	Жлчяьгнхлщй	1936-06-02 16:01:17	вишневый	крупный	C:/Users/Admin/bkjrwbjhrb	C:/Users/Admin/cavyndyvil
Yiqqjseurll dibtrikdhr	Жлчяьгнхлщй	1939-03-02 22:06:24	оливковый	крупный	C:/Users/Admin/fvjxfvlkz	C:/Users/Admin/qkabngasqk
Sddjfhvnmpe xnxrnwqfan	Шсцйххуцуиа	1905-01-05 20:35:46	зелёный	средний	C:/Users/Admin/evusbsracs	C:/Users/Admin/ejyofingxm
Vhleqqsaryl cokknsxwvy	Гойфйшофдей	1972-08-30 14:49:45	индиго	средний	C:/Users/Admin/oyuwaxvadt	C:/Users/Admin/okjbitekot
Vhleqqsaryl cokknsxwvy	Гойфйшофдей	1968-03-21 11:15:35	коричневый	крупный	C:/Users/Admin/poulgyatn	C:/Users/Admin/xbetluyuuq
Vhleqqsaryl cokknsxwvy	Гойфйшофдей	1922-11-24 01:08:44	розовый	средний	C:/Users/Admin/mxokzmdavq	C:/Users/Admin/jacbwlefu
Vhleqqsaryl cokknsxwvy	Гойфйшофдей	2006-10-25 01:29:11	коричневый	крупный	C:/Users/Admin/bzakyxtkr	C:/Users/Admin/cffgidgcvcq
Dxaryhfiuym yqnpzvpwzct	Мнсыйкпгчдс	1950-09-01 10:05:21	зелёный	мелкий	C:/Users/Admin/nlpnisvenr	C:/Users/Admin/mxwuyaftrf
Rqibwjgywvp gafduikqyy	Кздхтефьэт	1927-01-13 17:28:42	серый	крупный	C:/Users/Admin/qupkzndhqh	C:/Users/Admin/xqfvkqdvcr
Rqibwjgywvp gafduikqyy	Кздхтефьэт	1918-12-01 23:51:53	бордовый	мелкий	C:/Users/Admin/tvriapgcrd	C:/Users/Admin/iawrifuaeg
Zsetrjulnmf eezgknpth	Юешлйррхадь	2001-07-25 00:20:56	оранжевый	мелкий	C:/Users/Admin/knltpnqlv	C:/Users/Admin/jjgfbkygj
Zsetrjulnmf eezgknpth	Юешлйррхадь	1961-07-24 12:19:17	малиновый	мелкий	C:/Users/Admin/wklpdndlhv	C:/Users/Admin/sieaqbfazf
Jibypmrfdwt stoqlcsnni	Ювешьшиквсы	1965-08-19 16:38:09	фиолетовый	средний	C:/Users/Admin/aykdrnyyd	C:/Users/Admin/vumgfgfbca
Jibypmrfdwt stoqlcsnni	Ювешьшиквсы	1993-11-29 07:57:28	оливковый	средний	C:/Users/Admin/tmxntqpczb	C:/Users/Admin/zimsnacyji
Jibypmrfdwt stoqlcsnni	Ювешьшиквсы	1973-02-10 16:37:08	розовый	мелкий	C:/Users/Admin/vkmwmjdyt	C:/Users/Admin/nthomhulb

Рис. 5: Результат выполнения запроса 1

id	select_type	table	partitions	type	possible_keys	key
1	SIMPLE	birdwatcher		ALL	PRIMARY	
1	SIMPLE	season_type		ALL	PRIMARY	
1	SIMPLE	pool_type		ALL	PRIMARY	
1	SIMPLE	bird		ref	PRIMARY,id_species,id_birdwatcher	id_birdwatcher
1	SIMPLE	species		eq_ref	PRIMARY	PRIMARY
1	SIMPLE	meeting_time		ref	id_bird	id_bird
1	SIMPLE	season		ref	id_bird,id_name	id_bird
1	SIMPLE	lpl		ref	id_bird,id_pool	id_bird
1	SIMPLE	bird_photo		ref	id_bird	id_bird
1	SIMPLE	bird_song		ref	id_bird	id_bird

key...	ref	rows	filtered	Extra
		50	2.00	Using where; Using temporary
		4	25.00	Using where; Using join buffer (hash...
		5	20.00	Using where; Using join buffer (hash...
5	birbs.birdwatcher.id_birdwatcher	2062	100.00	Using where
4	birbs.bird.id_species	1	100.00	
5	birbs.bird.id_bird	1	100.00	
5	birbs.bird.id_bird	1	33.33	Using where
5	birbs.bird.id_bird	3	25.00	Using where
5	birbs.bird.id_bird	1	100.00	
5	birbs.bird.id_bird	2	100.00	

Рис. 6: EXPLAIN для запроса 1

6.1.1 Запрос 1.1

Формулировка запроса: вывести всех птиц, которых видел в сезон X бёрдвотчер Y с типом водоёма Z не менее трёх раз.

Код запроса на SQL:

```

select species.latin_name as "Latin_name",
species.russian_name as "Russian_name",
t as "Meeting_time",
f as "Feather_color",
size as "Size",
ph as "Photo",
song as "Song",
count(b) as cnt
from (select bird.id_bird as b,
bird.feath_color as f,
bird.size as size,
bird.id_species as s,
bird.id_birdwatcher as bw,
meeting_time.time_name as t,
bird_photo.bird_file as ph,
bird_song.bird_file as song
from bird
INNER JOIN season ON bird.id_bird=season.id_bird
INNER JOIN season_type ON season.id_name=season_type.id_name
INNER JOIN lpl ON lpl.id_bird=bird.id_bird
INNER JOIN pool_type ON lpl.id_pool=pool_type.id_pool
INNER JOIN meeting_time on meeting_time.id_bird=bird.id_bird
INNER JOIN bird_photo ON bird.id_bird=bird_photo.id_bird

```

```

INNER JOIN bird_song ON bird.id_bird=bird_song.id_bird
Where season_type.season_name = 'лето'
AND pool_type.pooltype = 'река'
group by b
) cnt_table
inner join species on cnt_table.s=species.id_species
inner join birdwatcher on cnt_table.bw=birdwatcher.id_birdwatcher
WHERE birdwatcher.last_name = 'Голубев'
AND birdwatcher.first_name = 'Ярослав'
AND birdwatcher.fathers_name = 'Георгиевич'
group by s
having cnt >= 3;

```

Описание запроса

Для этого запроса потребовалось модифицировать предыдущий запрос, отделив соединение таблиц, относящихся к птице (тип водоема, сезон, время встречи, etc), от связывания птицы с видом. В первую очередь создаётся таблица всех птиц, относящихся к сезону X и типу водоёма Z, сгруппированная по id птицы. Затем полученная таблица связывается с видом и бёрдвотчером, а результат группируется по id вида. Из итоговой таблицы выбираются те же поля, что и в запросе 1, но с добавлением поля cnt: COUNT(id_bird). Наконец, при помощи HAVING выбираются записи, у которых cnt больше либо равен трём.

Время выполнения запроса

Время выполнения этого запроса составило 1.281 секунд.

На рис. 7 показан результат выполнения этого запроса, а на рис. 8 - план его выполнения.

Latin name	Russian name	Meeting time	Feather_color	Size	Photo	Song	cnt
Vhlecsqsyrl ccknsxwvy	Пойфйшофдей	1972-08-30 14:49:45	индиго	средний	C:/Users/Admin/oyuwaxvadt	C:/Users/Admin/okjbitekot	4
Jlbypmrfdwit stoqlcsnni	Юшвешиквсы	1965-08-19 16:38:09	фиолетовый	средний	C:/Users/Admin/aykkrnyyd	C:/Users/Admin/vumgfgfbca	3
Ankdlrfgdli gaxqzcogtk	Жггншпфлй	1975-03-01 17:04:42	сизый	мелкий	C:/Users/Admin/ybydcxztig	C:/Users/Admin/jkxfamrctd	4
Wksgqzfttf gcjzlipop	Цьшшкккабьц	1928-04-03 06:56:46	зелёный	мелкий	C:/Users/Admin/ohlgedsqol	C:/Users/Admin/lhqrvxgaqx	5
Xxdavkqgth qfqumytivd	Кнйхздрбиче	1949-11-23 07:20:12	голубой	мелкий	C:/Users/Admin/wtwskhxlq	C:/Users/Admin/mmpimhibm	4
Fuijactmqc cgmirbpsye	Яхяияйтэтмх	1973-07-13 21:32:41	красный	мелкий	C:/Users/Admin/jpaeptxutp	C:/Users/Admin/ckuimpltnl	3
Gguetswewfi qfatttgkdl	Усщршкшюжзя	1990-09-14 05:57:10	тенно-оранжевый	средний	C:/Users/Admin/tlfdjuvzf	C:/Users/Admin/zqpnwyhthn	3
Cboktkoisli ivopgawvwm	Ъшацтьсгъяз	1975-12-16 19:06:57	лазурный	средний	C:/Users/Admin/nquervbsnh	C:/Users/Admin/gszswztnyb	6
Oyybziofmfo tscqgmaqv	Сукьо́йхибиц	1906-10-11 02:16:21	голубой	крупный	C:/Users/Admin/gwafxpviav	C:/Users/Admin/yneegrtstn	3
Flnyndxedup tyjrqofcsx	Чесырпвтзим	1922-06-19 12:45:57	бордовый	средний	C:/Users/Admin/aomfdexsqe	C:/Users/Admin/kkoufrvghr	3
Vdiwhrgexss fhgoueufc	Уюсызрсчжом	1900-09-10 19:47:41	сизый	крупный	C:/Users/Admin/ktnmkdxpaq	C:/Users/Admin/fgklrfqqgd	5
Unsguvpnln qvyzmri feh	Рцхядрвехщц	1978-12-13 09:29:35	оранжевый	крупный	C:/Users/Admin/wnepemdoeg	C:/Users/Admin/wmufadphn	3
Gbjhhtxbajx fgvmuovvyw	ьсбихяеоэрт	1924-03-26 00:27:48	серый	средний	C:/Users/Admin/rjswrdybv	C:/Users/Admin/loyivzemke	3
Ixnbeqxmaaf ufbfzhfcyw	Бтншчдысыэ	2004-12-08 22:16:37	тенно-оранжевый	мелкий	C:/Users/Admin/cbdzkqhfg	C:/Users/Admin/uafrrwqel	6
Xsupprehuvn lelylcznjk	Уийужуделеа	1916-06-21 19:10:10	индиго	крупный	C:/Users/Admin/pgwpmhifzh	C:/Users/Admin/dvshmtjznd	3
Dyfpzupkkgq bjoxhlzhc	Чнтэблфжюкд	1939-12-31 03:09:49	розовый	крупный	C:/Users/Admin/xrejknuhqj	C:/Users/Admin/flbxrujyst	4

Рис. 7: Результат выполнения запроса 1(1)

id	select_type	table	partitions	type	possible_keys	key
1	SIMPLE	birdwatcher		ALL	PRIMARY	
1	SIMPLE	season_type		ALL	PRIMARY	
1	SIMPLE	pool_type		ALL	PRIMARY	
1	SIMPLE	bird		ref	PRIMARY,id_species,id_birdwatcher	id_birdwatcher
1	SIMPLE	species		eq_ref	PRIMARY	PRIMARY
1	SIMPLE	meeting_time		ref	id_bird	id_bird
1	SIMPLE	season		ref	id_bird,id_name	id_bird
1	SIMPLE	lpl		ref	id_bird,id_pool	id_bird
1	SIMPLE	bird_photo		ref	id_bird	id_bird
1	SIMPLE	bird_song		ref	id_bird	id_bird

key...	ref	rows	filtered	Extra
		50	2.00	Using where; Using temporary
		4	25.00	Using where; Using join buffer (hash...
		5	20.00	Using where; Using join buffer (hash...
5	birbs.birdwatcher.id_birdwatcher	2062	100.00	Using where
4	birbs.bird.id_species	1	100.00	
5	birbs.bird.id_bird	1	100.00	
5	birbs.bird.id_bird	1	33.33	Using where
5	birbs.bird.id_bird	3	25.00	Using where
5	birbs.bird.id_bird	1	100.00	
5	birbs.bird.id_bird	2	100.00	

Рис. 8: EXPLAIN для запроса 1(1)

6.2 Запрос 2

Формулировка запроса: посчитать число птиц вида X, для которых загружены и фото, и песни.

Код запроса на SQL:

```
select count(*)
from (select bird.id_bird
      FROM bird
      INNER JOIN species ON species.id_species=bird.id_species
      INNER JOIN bird_photo ON bird.id_bird=bird_photo.id_bird
      INNER JOIN bird_song ON bird.id_bird=bird_song.id_bird
      WHERE species.latin_name='Yiqqjseurll_dibrtrkkhr'
      group by bird.id_bird) table_birds;
```

Описание запроса

В первую очередь строится таблица, в которой содержатся все птицы вида X. Результат необходимо связывать это с таблицами песен и фотографий птиц, группируя по id птицы. Поскольку применяется INNER JOIN, получившаяся таблица содержит записи только о тех птицах вида X, у которых есть фотографии и песни. Для неё выполняется COUNT(*), чтобы найти число таких птиц.

Время выполнения запроса

Время выполнения этого запроса составило 0.031 секунд.

На рис. 9 показан результат выполнения этого запроса, а на рис. 10 - план его выполнения.

	count(*)
►	1402

Рис. 9: Результат выполнения запроса 2

id	select_type	table	partitions	type	possible_keys	key
1	PRIMARY	<derived2>		ALL		
2	DERIVED	species		ALL	PRIMARY	
2	DERIVED	bird		ref	PRIMARY,id_species,id_birdwatcher	id_species
2	DERIVED	bird_photo		ref	id_bird	id_bird
2	DERIVED	bird_song		ref	id_bird	id_bird

key...	ref	rows	filtered	Extra
		34555	100.00	
		100	10.00	Using where; Using temporary
5	birbs.species.id_spedes	1164	100.00	Using index
5	birbs.bird.id_bird	1	100.00	Using index
5	birbs.bird.id_bird	2	100.00	Using index

Рис. 10: EXPLAIN для запроса 2

6.3 Запрос 3

Формулировка запроса: найти бёрдвотчера, который наблюдал максимальное число птиц (если таких несколько, то вывести всех).

Код запроса на SQL:

```
SELECT birdwatcher.last_name ,
birdwatcher.first_name ,
birdwatcher.fathers_name ,
count(bird.id_bird) as cnt
from birdwatcher
inner join bird on bird.id_birdwatcher=birdwatcher.id_birdwatcher
group by birdwatcher.id_birdwatcher
having cnt = (select count(*) as max_num
from birdwatcher
join bird on bird.id_birdwatcher=birdwatcher.id_birdwatcher
group by birdwatcher.id_birdwatcher
order by max_num desc LIMIT 1);
```

Описание запроса

Для начала необходимо посчитать максимальное число наблюдений cnt_max среди всех бёрдвотчеров. Это реализуется при помощи команды сортировки по убыванию с лимитом в одну запись ORDER BY count DESC LIMIT 1. Затем реализуется таблица с общим числом птиц cnt для каждого из бердвотчеров, в которой выводятся ФИО бёрдвотчера и cnt. Из этой таблицы выбираются те строки, в которых cnt = cnt_max.

Время выполнения запроса

Время выполнения этого запроса составило 0.141 секунд.

На рис. 11 показан результат выполнения этого запроса, а на рис. 12 - план его выполнения.

	id_birdwatcher	last_name	first_name	fathers_name	cnt
►	42	Голубев	Роман	Викторович	2108

	id_birdwatcher	last_name	first_name	fathers_name	cnt
►	42	Голубев	Роман	Викторович	4084
	50	Беляев	Александр	Дмитриевич	4084

Рис. 11: Результат выполнения запроса 3 (случай одного и нескольких бёрдвотчеров)

id	select_type	table	partitions	type	possible_keys	key
1	PRIMARY	birdwatcher		index	PRIMARY	PRIMARY
1	PRIMARY	bird		ref	id_birdwatcher	id_birdwatcher
2	SUBQUERY	birdwatcher		index	PRIMARY	PRIMARY
2	SUBQUERY	bird		ref	id_birdwatcher	id_birdwatcher

key...	ref	rows	filtered	Extra
4		50	100.00	
5	birbs.birdwatcher.id_birdwatcher	2062	100.00	Using index
4		50	100.00	Using index; Using temporary; Using...
5	birbs.birdwatcher.id_birdwatcher	2062	100.00	Using index

Рис. 12: EXPLAIN для запроса 3

6.3.1 Запрос 3.1

Формулировка запроса: найти бёрдвотчера, который наблюдал минимальное число птиц (если таких несколько, то вывести всех).

Код запроса на SQL:

```
SELECT birdwatcher.last_name ,
birdwatcher.first_name ,
birdwatcher.fathers_name ,
count(bird.id_bird) as cnt
from birdwatcher
inner join bird on bird.id_birdwatcher=birdwatcher.id_birdwatcher
group by birdwatcher.id_birdwatcher
having cnt = (select count(*) as min_num
              from birdwatcher
              join bird on bird.id_birdwatcher=birdwatcher.id_birdwatcher
              group by birdwatcher.id_birdwatcher
              order by max_num asc LIMIT 1);
```

Описание запроса

Этот запрос строится аналогично предыдущему. Разница заключается в том, что для выбора минимального числа птиц среди всех бёрдвотчеров используется сортировка в порядке возрастания, а не убывания ORDER BY count ASC LIMIT 1.

Время выполнения запроса

Время выполнения этого запроса составило 0.125 секунд.

На рис. 13 показан результат выполнения этого запроса, а на рис. 14 - план его выполнения.

	id_birdwatcher	last_name	first_name	fathers_name	cnt
►	30	Воробьев	Виктор	Александрович	1934

Рис. 13: Результат выполнения запроса 3(1)

id	select_type	table	partitions	type	possible_keys	key
1	PRIMARY	birdwatcher		index	PRIMARY	PRIMARY
1	PRIMARY	bird		ref	id_birdwatcher	id_birdwatcher
2	SUBQUERY	birdwatcher		index	PRIMARY	PRIMARY
2	SUBQUERY	bird		ref	id_birdwatcher	id_birdwatcher

key...	ref	rows	filtered	Extra
4		50	100.00	
5	birbs.birdwatcher.id_birdwatcher	2062	100.00	Using index
4		50	100.00	Using index; Using temporary; Using...
5	birbs.birdwatcher.id_birdwatcher	2062	100.00	Using index

Рис. 14: EXPLAIN для запроса 3(1)

6.4 Запрос 4

Формулировка запроса: посчитать число видов с одинаковым числом птиц.

Код запроса на SQL:

```
select count(id_species) as SpeciesCount, cnt as BirdCount
from (select species.id_species, count(bird.id_bird) as cnt
      from species
      inner join bird on species.id_species=bird.id_species
      group by species.id_species) as tmp_table
group by cnt
```

Описание запроса

В первую очередь считается число птиц для каждого вида. Для этого выбирается COUNT(id_bird) из таблицы, связывающей bird и species, в которой записи сгруппированы по id_species. Строки получившейся таблицы группируются по количеству птиц (GROUP BY cnt), после чего считается количество видов count(id_species).

Время выполнения запроса

Время выполнения этого запроса составило 0.063 секунд.

На рис. 15 показан результат выполнения этого запроса, а на рис. 16 - план его выполнения.

SpeciesCount	BirdCount
2	789
1	1813
1	1692
1	220
1	695
1	54
1	1995
1	1205
1	1195
1	916
1	1132
1	1687
1	199
1	350
1	337

Рис. 15: Результат выполнения запроса 4

id	select_type	table	partitions	type	possible_keys	key
1	PRIMARY	<derived2>		ALL		
2	DERIVED	species		index	PRIMARY,id_book	PRIMARY
2	DERIVED	bird		ref	id_species	id_species
		key...	ref	rows	filtered	Extra
				116472	100.00	Using temporary
4				100	100.00	Using index
5		birbs.species.id_spedes		1164	100.00	Using index

Рис. 16: EXPLAIN для запроса 4

На рис. 17 изображена диаграмма для запроса 4. По оси X отложено количество птиц, а по оси Y - количество видов, для которых встречено это число птиц. Поскольку распределение равномерное и случайное в большом диапазоне значений, вероятность совпадения числа встреченных птиц у двух или более видов мала. В связи с этим нет необходимости предоставлять диаграмму по всем записям. На рисунке показаны лишь первые 30 видов, для двух из которых значение cnt совпало.

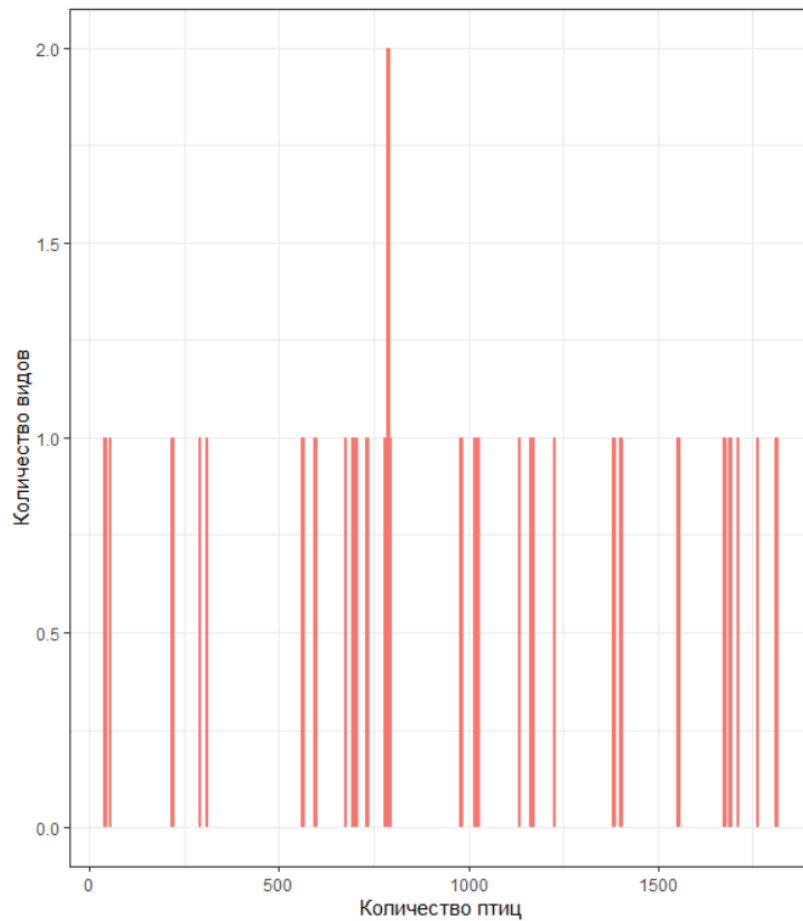


Рис. 17: Диаграмма для запроса 4

6.5 Запрос 5

Формулировка запроса: найти те книги, для которых не было встречено ни одной птицы.

Код запроса на SQL:

```
select author , pub_info , cnt_species , count(bird.id_bird) as cnt_birds
from(
    select book.author as author , book.pub_info as pub_info , species.id_spec
    from book
    inner join species on species.id_book=book.id_book
    group by book.id_book) tmp_table
inner join bird on tmp_table.id_species=bird.id_species
group by tmp_table.id_book
having cnt_birds=0;
```

Описание запроса

В первую очередь находится общее число cnt птиц для каждой книги. В получившейся таблице выбираются записи, для которых cnt = 0. Поскольку в результирующей таблице требуется также вывести количество видов для найденных книг, применяется вложенный запрос, чтобы считать суммарное количество видов отдельно от суммарного количества птиц.

Время выполнения запроса

Время выполнения этого запроса составило 0.062 секунд.

На рис. 18 показан результат выполнения этого запроса, а на рис. 20 - план его выполнения.

В силу того, что исходное распределение формировалась таким образом, чтобы на каждую книгу приходился хотя бы один вид, а на каждый вид - хотя бы одна птица, в базе данных не может существовать ни одной такой книги, для которой не было бы найдено ни одной птицы. Это видно на рис. 18: результатом выполнения запроса является пустая таблица. Для того, чтобы проверить работоспособность запроса, изначально условие было изменено следующим образом: отбирались книги, для которых было найдено менее 1000 птиц. Это потребовало минимальных изменений в коде (вставить «having cnt_birds<1000» вместо «having cnt_birds=0»). На рис. 19 представлен результат выполнения этого запроса.

	author	pub_info	cnt_species	cnt_birds
--	--------	----------	-------------	-----------

Рис. 18: Результат выполнения запроса 5

	author	pub_info	cnt_species	cnt_birds
▶	Волков Леонид Григорьевич	Alauda	4	948
	Воробьев Ярослав Александрович	Central European Journal of Biology	2	176
	Павлов Андрей Васильевич	Neotropical Birding	2	593
	Соколов Денис Геннадиевич	Ornis Fennica	1	889
	Зайцев Арсений Богданович	Avicultural Magazine	7	456
	Петров Григорий Вольфович	Ornithos	4	669
	Волков Игорь Аркадьевич	Avicultural Magazine	5	277
	Смирнов Денис Аркадьевич	Ardea	3	288
	Морозов Игорь Антонович	Bulletin of the British Ornithologists	9	906
	Новиков Леонид Вячеславович	Aquila	2	779
	Петров Арсений Александрович	Ornithologische Mitteilungen	5	339
	Тарасов Фёдор Валерьевич	Avicultural Magazine	1	37
	Лебедев Григорий Витальевич	Central European Journal of Biology	2	571
	Морозов Николай Антонович	Avian Science	6	678

Рис. 19: Результат выполнения запроса 5 с поправкой условия

id	select_type	table	partitions	type	possible_keys	key
1	PRIMARY	<derived2>		ALL		
1	PRIMARY	bird		ref	id_species	id_species
2	DERIVED	book		index	PRIMARY	PRIMARY
2	DERIVED	species		ref	id_book	id_book

key...	ref	rows	filtered	Extra
		100	100.00	Using temporary
5	tmp_table.id_spedes	1164	100.00	Using index
4		25	100.00	
5	birbs.book.id_book	4	100.00	Using index

Рис. 20: EXPLAIN для запроса 5

6.6 Запрос 6

Формулировка запроса: для каждого сезона и каждого вида посчитать число наблюдений.

Код запроса на SQL:

```
select * from
(select season_type.season_name as season ,
species.latin_name as species ,
count(bird.id_bird) as numb_birds
from bird
inner join season on season.id_bird=bird.id_bird
inner join season_type on season.id_name=season_type.id_name
inner join species on bird.id_species=species.id_species
group by season , species
UNION
select season_type.season_name as season ,
species.latin_name as species ,
count(bird.id_bird)
from season_type
left join season on season.id_name=season_type.id_name
left join bird on season.id_bird = bird.id_bird
cross join species
```

```

where season_type.season_name not in (
    select season_type.season_name as season
    from season
    inner join season_type on season.id_name=season_type.id_name
)
group by species.id_species) tbl
order by species;

```

Описание запроса

В этом запросе объединяются две таблицы. В первой связываются таблицы птиц, сезонов и видов, а результат группируется по виду и сезону посредством GROUP BY. Эта таблица содержит число птиц для всех видов и всех сезонов, для которых было хотя бы одно наблюдение, однако не учитываются сезоны, для которых не было ни одного наблюдения. Чтобы исправить этот недочет, создаётся вторая таблица, в которой рассматриваются те сезоны, для которых не было наблюдений. В результате объединения посредством UNION получается итоговая таблица, из которой выбираются название сезона, латинское название вида и количество птиц.

Время выполнения запроса

Время выполнения этого запроса составило 2.031 секунд.

Для иллюстрации корректной работы этого запроса в таблицу season_type был добавлен несуществующий сезон «летосень», к которому не привязано ни одной записи в таблицах season и bird. На рис. 21 показан результат выполнения этого запроса, а на рис. 22 - план его выполнения.

season	species	numb_birds
летосень	Awwijytxuug yyqdlezhfc	0
зима	Awwijytxuug yyqdlezhfc	135
весна	Awwijytxuug yyqdlezhfc	124
лето	Awwijytxuug yyqdlezhfc	149
осень	Awwijytxuug yyqdlezhfc	125
летосень	Beypptfcqm jjakchdphe	0
зима	Beypptfcqm jjakchdphe	468
лето	Beypptfcqm jjakchdphe	482
осень	Beypptfcqm jjakchdphe	458
весна	Beypptfcqm jjakchdphe	487
летосень	Bhaargigcta pzqjeyeoba	0
зима	Bhaargigcta pzqjeyeoba	388
весна	Bhaargigcta pzqjeyeoba	354
лето	Bhaargigcta pzqjeyeoba	373
осень	Bhaargigcta pzqjeyeoba	358
летосень	Buepyjxhcn neonydyuw	0

Рис. 21: Результат выполнения запроса 6

id	select_type	table	partitions	type	possible_keys	key
1	PRIMARY	species		ALL	PRIMARY	
1	PRIMARY	bird		ref	PRIMARY,id_species	id_species
1	PRIMARY	season		ref	id_bird,id_name	id_bird
1	PRIMARY	season_type		eq_ref	PRIMARY	PRIMARY
2	UNION	season_type		ALL		
2	UNION	species		ALL	PRIMARY,id_book	
2	UNION	season		ref	id_name	id_name
2	UNION	bird		eq_ref	PRIMARY	PRIMARY
3	SUBQUERY	season_type		ALL	PRIMARY	
3	SUBQUERY	season		ref	id_name	id_name
	UNION RESULT	<union1,2>		ALL		

key...	ref	rows	filtered	Extra
		100	100.00	Using temporary
5	birds.species.id_spedes	1088	100.00	Using index
5	birds.bird.id_bird	1	100.00	Using where
4	birds.season.id_name	1	100.00	
		5	100.00	Using where; Using temporary
		100	100.00	Using join buffer (hash join)
5	birds.season_type.id_name	33477	100.00	
4	birds.season.id_bird	1	100.00	Using index
		5	100.00	
5	birds.season_type.id_name	33477	100.00	Using index
		0		Using temporary; Using filesort

Рис. 22: EXPLAIN для запроса 6

На рис. 23 изображена диаграмма для запроса 6. По оси X отложены виды, по оси Y - сезоны, а по оси Z - суммарное количество птиц.

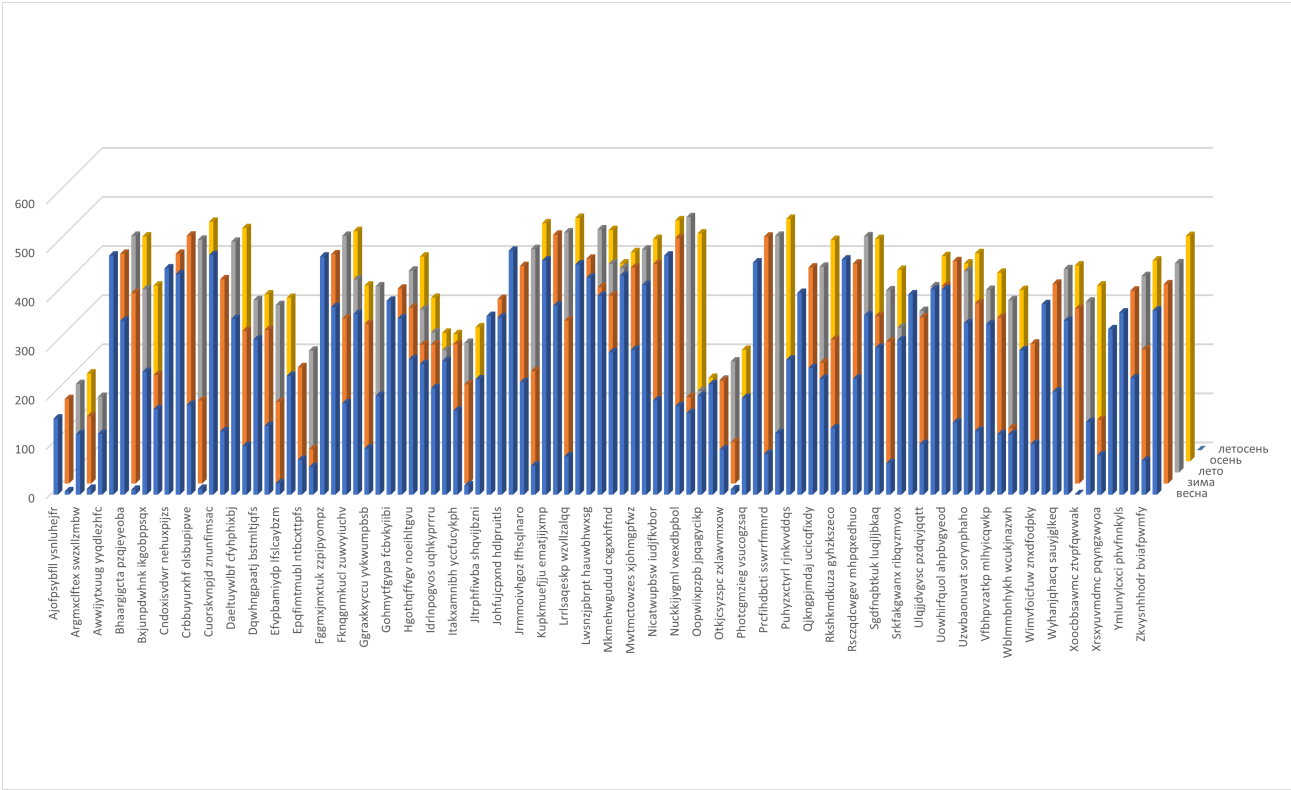


Рис. 23: Диаграмма для запроса 6

6.7 Запрос 7

Формулировка запроса: для всех бёрдвотчеров посчитать число птиц.

Код запроса на SQL:

```
SELECT birdwatcher.last_name as last_name ,
       birdwatcher.first_name as first_name ,
       birdwatcher.fathers_name as fathers_name ,
       count(bird.id_bird) as cnt_bird
from birdwatcher
inner join bird on bird.id_birdwatcher=birdwatcher.id_birdwatcher
group by birdwatcher.id_birdwatcher
```

Описание запроса

В этом запросе связываются таблицы птиц и бёрдвотчеров, причем строки результата группируются по id бёрдвотчера. Из получившейся таблицы выводятся ФИО каждого бёрдвотчера и суммарное количество птиц для него.

Время выполнения запроса

Время выполнения этого запроса составило 0.062 секунд.

На рис. 24 показан результат выполнения этого запроса, а на рис. 25 - план его выполнения.

last_name	first_name	fathers_name	cnt_bird
Соколов	Александр	Александрович	2064
Голубев	Ярослав	Георгиевич	1966
Иванов	Александр	Вячеславович	2003
Иванов	Андрей	Владиславович	2015
Морозов	Давид	Вольфович	2100
Павлов	Андрей	Вячеславович	1953
Виноградов	Степан	Александрович	2028
Волков	Степан	Борисович	1971
Васильев	Александр	Анатольевич	2015
Волков	Фёдор	Антонович	2005
Голубев	Павел	Георгиевич	1984
Воробьёв	Денис	Васильевич	2025
Виноградов	Владислав	Вячеславович	1990
Иванов	Арсений	Вячеславович	2020
Попов	Денис	Валентинович	1993
Голубев	Артёмий	Борисович	1966

Рис. 24: Результат выполнения запроса 7

id	select_type	table	partitions	type	possible_keys	key
1	SIMPLE	birdwatcher		index	PRIMARY	PRIMARY
1	SIMPLE	bird		ref	id_birdwatcher	id_birdwatcher

key...	ref	rows	filtered	Extra
4		50	100.00	
5	birds.birdwatcher.id_birdwatcher	2062	100.00	Using index

Рис. 25: EXPLAIN для запроса 7

6.7.1 Запрос 7.1

Формулировка запроса: для всех бёрдвотчеров посчитать число песен, которые в сумме исполнили птицы.

Код запроса на SQL:

```
SELECT last_name ,
first_name ,
fathers_name ,
count(id_bird_song) as cnt_song
from birdwatcher
inner join bird on bird.id_birdwatcher=birdwatcher.id_birdwatcher
inner join bird_song on bird.id_bird=bird_song.id_bird
group by birdwatcher.id_birdwatcher ;
```

Описание запроса

Для начала связываются таблицы бёрдвотчеров и птиц, результат группируется по id бёрдвотчера. В получившейся таблице подсчитывается количество песен для каждой птицы посредством вызова COUNT(id_bird_song).

Время выполнения запроса

Время выполнения этого запроса составило 0.656 секунд.

На рис. 26 показан результат выполнения этого запроса, а на рис. 27 - план его выполнения.

last_name	first_name	fathers_name	cnt_song
Соколов	Александр	Александрович	4142
Голубев	Ярослав	Георгиевич	3908
Иванов	Александр	Вячеславович	3958
Иванов	Андрей	Владиславович	4055
Морозов	Давид	Вольфович	4201
Павлов	Андрей	Вячеславович	3946
Виноградов	Степан	Александрович	4070
Волков	Степан	Борисович	3926
Васильев	Александр	Анатолевич	3972
Волков	Фёдор	Антонович	4023
Голубев	Павел	Георгиевич	3969
Воробьёв	Денис	Васильевич	4079
Виноградов	Владислав	Вячеславович	3941
Иванов	Арсений	Вячеславович	4015
Попов	Денис	Валентинович	4000
Голубев	Артений	Борисович	3883

Рис. 26: Результат выполнения запроса 7(1)

id	select_type	table	partitions	type	possible_keys	key
1	SIMPLE	birdwatcher		index	PRIMARY	PRIMARY
1	SIMPLE	bird		ref	PRIMARY,id_birdwatcher	id_birdwatcher
1	SIMPLE	bird_song		ref	id_bird	id_bird

key...	ref	rows	filtered	Extra
4		50	100.00	
5	birbs.birdwatcher.id_birdwatcher	2062	100.00	Using index
5	birbs.bird.id_bird	2	100.00	Using index

Рис. 27: EXPLAIN для запроса 7(1)

На рис. 28 изображена диаграмма для запросов 7 и 7.1. По оси X отложены бёрдвотчеры, а по оси Y - суммарное количество наблюдений, сделанных этими бёрдвотчерами. Синим цветом показано суммарное число птиц для каждого бёрдвотчера, а красным - суммарное число песен для этих птиц.

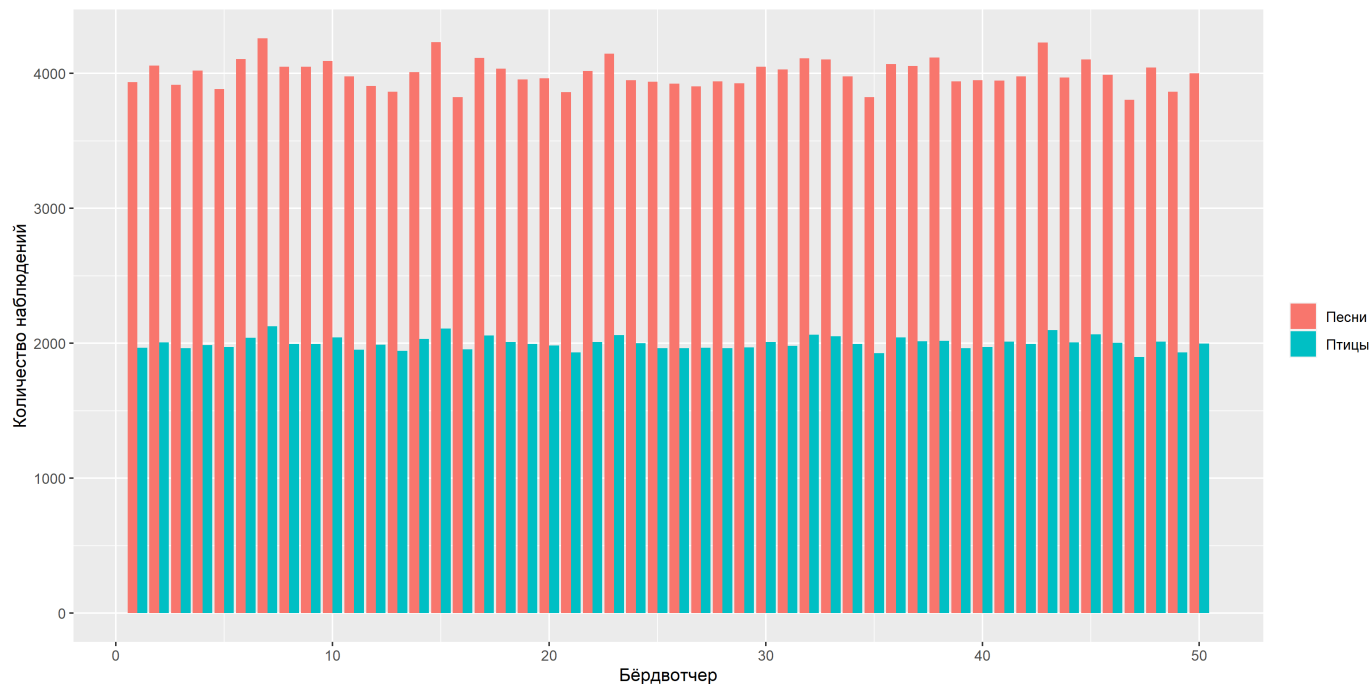


Рис. 28: Диаграмма для запроса 7

6.8 Запрос 8

Формулировка запроса: найти все виды птиц, число наблюдений которых больше, чем у вида X.

Код запроса на SQL:

```
SELECT species.latin_name as species, count(bird.id_bird) as cnt FROM bird
INNER JOIN species ON species.id_species=bird.id_species
group by species.id_species
having cnt > (SELECT count(bird.id_bird) FROM bird
INNER JOIN species ON species.id_species=bird.id_species
WHERE species.latin_name='Yiqqjseurll_dibrtrkchr');
```

Описание запроса

В первую очередь находится общее число птиц cnt_sp у вида X. Затем формируется таблица, содержащее количество птиц для каждого вида: для этого связываются таблицы bird и species с группировкой по id вида, и считается суммарное число птиц cnt посредством команды COUNT(id_bird). Из получившейся таблицы выбираются записи, в которых cnt > cnt_sp

Время выполнения запроса

Время выполнения этого запроса составило 0.062 секунд.

На рис. 29 показан результат выполнения этого запроса, а на рис. 30 - план его выполнения.

species	cnt
Sddjfhvnmpe xnxrnwqfan	1690
Rqibwjgywvp gafduikqyy	1678
Xxdavkqgth qfqumytvd	1764
Fuijacttmqc cgmirbpsye	1556
Cboktfkoisl ivopgawvwm	1712
Flnyndxedup tyjrqofcsx	1813
Vdiwhrgexss fhgoueucfr	1692
Unsguvpnlh qvzymrifeh	1995
Xsupprehuvn leylcznj	1687
Wsabwkyjdrj uwyrjfrkur	1676
Doeqiwskeho bwukdvnqwx	1907
Elhywpaier cibqzpjf	1573
Ykehqcwoxse mpyakyhji	1721
Kziyrhilvo fuxrgictvf	1728
Fhxywuinsub utrxztism	1569
Gaoeopzlfqu irupukdhkr	1920

Рис. 29: Результат выполнения запроса 8

id	select_type	table	partitions	type	possible_keys	key
1	PRIMARY	species		index	PRIMARY,id_book	PRIMARY
1	PRIMARY	bird		ref	id_species	id_species
2	SUBQUERY	species		ALL	PRIMARY	
2	SUBQUERY	bird		ref	id_species	id_species

key...	ref	rows	filtered	Extra
4		100	100.00	
5	birbs.species.id_spedes	1164	100.00	Using index
		100	10.00	Using where
5	birbs.species.id_spedes	1164	100.00	Using index

Рис. 30: EXPLAIN для 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 птиц была разработана ER-диаграмма, описывающая процессы, протекающие в предметной области, а также схема объектов.

На основе построенных диаграмм была спроектирована и реализована база данных птиц. База содержит 17 таблиц – 6 словарей первого уровня, 2 словаря второго уровня и 9 словарей третьего уровня. База была заполнена случайными данными, сгенерированными программой на языке программирования Python 3.8. 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.

Приложение А. Реализация базы данных

Реализация базы данных.

```
drop database if exists birds;
create database birds;
use birds;
create table anthr_obj_type(
    id_obj int primary key auto_increment,
    objtype varchar(30)
);
create table plant_type(
    id_plant int primary key auto_increment,
    planttype varchar(30)
);

create table pool_type(
    id_pool int primary key auto_increment,
    pooltype varchar(30)
);

create table season_type(
    id_name int primary key auto_increment,
    season_name varchar(30)
);

create table book(
    id_book int primary key auto_increment,
    author text,
    species_desc longtext,
    pub_info text
);

create table birdwatcher(
    id_birdwatcher int primary key auto_increment,
    last_name varchar(30),
    first_name longtext,
    fathers_name text
);

create table species(
    id_species int primary key auto_increment,
    latin_name varchar(30),
    russian_name varchar(30),
    area text,
    id_book int,
    foreign key (id_book) references book (id_book)
```

```

);

create table bird(
    id_bird int primary key auto_increment,
    feath_color varchar(30),
    beak_shape varchar(30),
    size varchar(30),
    tail_shape varchar(30),
    pool tinyint,
    anthr_obj_distance varchar(30),
    id_species int,
    id_birdwatcher int,
    foreign key (id_species) references species (id_species),
    foreign key (id_birdwatcher) references birdwatcher (id_birdwatcher)
);

create table bird_photo(
    id_photo int primary key auto_increment,
    bird_file text,
    id_bird int,
    foreign key (id_bird) references bird (id_bird)
);

create table bird_song(
    id_bird_song int primary key auto_increment,
    bird_file text,
    purpose varchar(30),
    id_bird int,
    foreign key (id_bird) references bird (id_bird)
);

create table species_photo(
    id_species_photo int primary key auto_increment,
    species_file text,
    id_species int,
    foreign key (id_species) references species (id_species)
);

create table species_song(
    id_species_song int primary key auto_increment,
    species_file text,
    purpose varchar(30),
    id_species int,
    foreign key (id_species) references species (id_species)
);

```

```

create table la(
    id_la int primary key auto_increment ,
    id_obj int ,
    id_bird int ,
    foreign key (id_bird) references bird (id_bird),
    foreign key (id_obj) references anthr_obj_type (id_obj)
);

create table lpl(
    id_lpl int primary key auto_increment ,
    id_pool int ,
    id_bird int ,
    foreign key (id_bird) references bird (id_bird),
    foreign key (id_pool) references pool_type (id_pool)
);

create table lpt(
    id_lpt int primary key auto_increment ,
    id_plant int ,
    id_bird int ,
    foreign key (id_bird) references bird (id_bird),
    foreign key (id_plant) references plant_type (id_plant)
);

create table season(
    id_season int primary key auto_increment ,
    id_name int ,
    id_bird int ,
    foreign key (id_bird) references bird (id_bird),
    foreign key (id_name) references season_type (id_name)
);

create table meeting_time(
    id_time int primary key auto_increment ,
    time_name datetime ,
    id_bird int ,
    foreign key (id_bird) references bird (id_bird)
);

```

Приложение Б. Код заполнения базы данных

Код заполнения базы данных на языке Python:

main.py

```
import mysql.connector as connector
from random import choice, randint
import data
import sys

try:
    con = connector.connect(user='root',
                             password='kb2002kb',
                             host='127.0.0.1',
                             database='birds'
                             )
    print( 'Connection_complete! ' )
except connector.Error as err:
    print( 'Something_went_wrong:_', err)
    sys.exit()

cur = con.cursor(buffered=True)

if data.create_data:
    # order = 1
    for i in range(data.numb_birdwatchers):
        cur.execute( 'INSERT INTO birdwatcher_(
last_name, _first_name, _fathers_name) VALUES(%s, %s, %s)',
                     (choice(data.last_names), choice(data.first_names),
choice(data.father_names)))
        for i in range(data.numb_books):
            cur.execute( 'INSERT INTO book_(author, _species_desc, _pub_info)
VALUES(%s, %s, %s)',
                         (choice(data.last_names) + '_ ' + choice(data.first_names)
+ '_ ' + choice(data.father_names),
                         data.generate_description(32), choice(data.pub_infos)))
            for i in range(data.numb_season_types):
                cur.execute( "INSERT INTO season_type_(season_name)
VALUES_( ' " + data.season_types[i] + " ' )" )
            for i in range(data.numb_pools):
                cur.execute( "INSERT INTO pool_type_(pooltype)
VALUES_( ' " + data.pool_types[i] + " ' )" )
            for i in range(data.numb_plants):
                cur.execute( "INSERT INTO plant_type_(planttype)
VALUES_( ' " + data.plant_types[i] + " ' )" )
            for i in range(data.numb_anthr_obj):
                cur.execute( "INSERT INTO anthr_obj_type_(objtype)
```



```

VALUES_('\" + data.anthr_obj_types[i] + '\')")
    print('Order_1_completed')

# order = 2
cur.execute('SELECT_*_FROM_book_WHERE_id_book_>_0')
books = cur.fetchall()
distr = data.randNums(data.numb_books, 1, 10, data.numb_species)
for i in range(data.numb_books):
    for j in range(distr[i]):
        cur.execute('INSERT INTO_species_(latin_name ,
russian_name ,_area ,_id_book)_VALUES_(%s ,_%s ,_%s ,_' +
                        str(books[i][0]) + ')',
                        (data.generate_sp_name('eng'),
data.generate_sp_name('rus'), data.generate_path()))
        print('Species_completed')
        cur.execute('SELECT_*_FROM_species')
        species = cur.fetchall()
        distr = data.randNums(data.numb_species, 1, 2000, data.numb_birds)
        print('Distr_completed')
        for i in range(data.numb_species):
            for j in range(distr[i]):
                cur.execute('INSERT INTO_bird
(feath_color ,_beak_shape ,_size ,_tail_shape ,_'
'pool ,_anthr_obj_distance ,_id_species ,_id_birdwatcher)
VALUES_(%s ,_%s ,_%s ,_%s ,_' +
str(randint(0, 1)) + ',_%s ,_' + str(species[i][0]) +
',_' + str(randint(1, data.numb_birdwatchers)) + ')',
(choice(data.feather_colors), choice(data.beak_shapes),
choice(data.sizes), choice(data.tail_shapes),
choice(data.anthr_obj_distances)))
                print('Order_2_completed')

# order = 3
for i in range(data.numb_species):
    cur.execute("INSERT INTO_species_song
(species_file ,_purpose ,_id_species)_VALUES_('\" +
data.generate_path() + '\",_%s ,_%s)", (choice(data.purposes),
str(species[i][0])))

cur.execute("INSERT INTO_species_photo_(species_file ,_id_species)_VALUES
('\" + data.generate_path() + '\",_' +
str(species[i][0]) + ')')

cur.execute('SELECT_*_FROM_bird')
distr_songs = data.randNums(data.numb_birds, 1, data.numb_bird_songs, -1)

```

```

distr_photos = data.randNums(data.numb_birds, 1, data.numb_bird_photos, -1)
distr_plant_links = data.randNums(data.numb_birds, 1, data.numb_lpt, -1)
distr_pool_links = data.randNums(data.numb_birds, 1, data.numb_lpl, -1)
distr_antr_links = data.randNums(data.numb_birds, 1, data.numb_lpa, -1)
birds = cur.fetchall()
print( 'All_distr_completed' )
for i in range(data.numb_birds):
    bird = str(birds[i][0])
    cur.execute( 'SELECT_*_FROM_season_type' )
    seasons = cur.fetchall()
    cur.execute( 'INSERT INTO_season
(id_bird, _id_name)_VALUES_( ' + bird + ",_" +
str(choice(seasons[:4])[0]) + " )" )
    cur.execute( "INSERT INTO_meeting_time
(time_name, _id_bird)_VALUES_( ' " + str(data.generate_time()) + " ',_" +
bird + ' )' )
    for j in range(distr_songs[i]):
        cur.execute( "INSERT INTO_bird_song
(bird_file, _purpose, _id_bird)_VALUES_( ' " +
data.generate_path() + " '___,_%s,_%s)", (choice(data.purposes), bird))
    for j in range(distr_photos[i]):
        cur.execute( "INSERT INTO_bird_photo
(bird_file, _id_bird)_VALUES_( ' " + data.generate_path() + " ',_" +
bird + ' )' )
    for j in range(distr_plant_links[i]):
        cur.execute( 'SELECT_*_FROM_plant_type' )
        plants = cur.fetchall()
        cur.execute( 'INSERT INTO_lpt
(id_plant, _id_bird)_VALUES_( ' +
                                str(choice(plants)[0]) + ",_" + bird + " )" )
    for j in range(distr_pool_links[i]):
        cur.execute( 'SELECT_*_FROM_pool_type' )
        pools = cur.fetchall()
        cur.execute( 'INSERT INTO_lpl
(id_pool, _id_bird)_VALUES_( ' +
                                str(choice(pools)[0]) + ",_" + bird + " )" )
    for j in range(distr_antr_links[i]):
        cur.execute( 'SELECT_*_FROM_anthr_obj_type' )
        objects = cur.fetchall()
        cur.execute( 'INSERT INTO_la
(id_obj, _id_bird)_VALUES_( ' +
                                str(choice(objects)[0]) + ",_" + bird + " )" )
print( 'Order_3_completed' )

```

```
con.commit()  
cur.close()  
con.close()
```

```

data.py

import string
import random
from datetime import datetime, timedelta
import re

def randNums(n,a,b,s):
    if s == -1:
        count = 0
        nums = []
        while count < n:
            r = random.randint(a, b)
            count += 1
            nums.append(r)
    else:
        #finds n random ints in [a,b] with sum of s
        while True:
            total, count = 0,0
            nums = []
            while total < s and count < n:
                r = random.randint(a,b)
                total += r
                count += 1
                nums.append(r)
            if total == s and count == n: break
    return nums

create_data = True

# order = 1
numb_birdwatchers = 50
numb_season_types = 5
numb_books = 25
numb_pools = 5
numb_plants = 5
numb_anthr_obj = 4

# order = 2
numb_species = 100
numb_birds = 100000

#order = 3
numb_sp_songs = 100
numb_sp_photos = 100

```

```

numb_bird_songs = 3
numb_bird_photos = 2
numb_lpt = 5
numb_lpl = 5
numb_lpa = 5

season_types = [ 'лето', 'осень', 'зима', 'весна', 'летосень' ]
feather_colors = [ 'красный', 'зелёный', 'оранжевый', 'белый', 'чёрный',
                   'коричневый', 'малиновый', 'голубой', 'фиолетовый', 'розовый',
                   'синий', 'жёлтый', 'серый', 'оливковый', 'охра', 'вишневый',
                   'темнооранжевый', 'бордовый', 'лазурный', 'индиго' ]
beak_shapes = [ 'острый', 'плоский', 'крючковидный', 'загнутый_вниз',
                 'тонкий', 'толстый', 'загнутый_вверх', 'долото',
                 'кинжал', 'шипцы', 'широкий', 'узкий', 'зазубренный' ]
sizes = [ 'крупный', 'средний', 'мелкий' ]
tail_shapes = [ 'клиновидный', 'короткий', 'длинный', 'острый',
                 'Уобразный', 'раздвоенный', 'овальный', 'плоский' ]
anthr_obj_distances = [ 'внутри_населенного_пункта', 'рядом_с_населенным_пунктом',
                        'ненаселенная_территория' ]
anthr_obj_types = [ 'крупный_город', 'деревня', 'сельская_местность', 'пгт' ]
purposes = [ 'переключка', 'сигнал_тревоги', 'подражание',
              'дуэтное', 'охранная', 'брачная' ]
# descriptions
pub_infos = [
    'Беркут', 'Орнитология',
    'The_Russian_Journal_of_Ornithology_Официальный_сайт_—_http://www.ornis.su',
    'Пернатые_хищники_и_их_охрана_—_Raptors_Conservation',
    'Acta_Ornithologica',
    'Alauda', 'Alula', 'Ardea', 'Avian_Science',
    'Avicultural_Magazine',
    'Aquila', 'Bird_Study', 'Birding_Scotland',
    'Bulletin_of_the_British_Ornithologists',
    'Central_European_Journal_of_Biology', 'Cotinga',
    'Der_Falke', 'Ibis_—_British_Ornithologists_Union',
    'Journal_of_Avian_Biology', 'Journal_of_Ornithology',
    'Neotropical_Birding', 'Ornis_Fennica',
    'Ornithologische_Mitteilungen', 'Ornithos', 'Die_Vogelwelt',
    'Dutch_Birding' ]
first_names = [ 'Александр', 'Григорий', 'Лев', 'Андрей', 'Роман',
                 'Арсений', 'Степан', 'Владислав', 'Никита', 'Глеб',
                 'Марк', 'Давид', 'Ярослав', 'Евгений', 'Матвей',
                 'Фёдор', 'Николай', 'Алексей', 'Андрей', 'Артемий',
                 'Виктор', 'Никита', 'Даниил', 'Денис', 'Егор', 'Игорь',
                 'Лев', 'Леонид', 'Павел' ]
last_names = [ 'Смирнов', 'Иванов', 'Кузнецов', 'Соколов',
                'Попов', 'Лебедев', 'Козлов', 'Новиков', 'Морозов',

```

```

'Петров', 'Волков', 'Соловьёв', 'Васильев', 'Зайцев',
'Павлов', 'Семёнов', 'Голубев', 'Виноградов', 'Богданов',
'Воробьёв', 'Фёдоров', 'Михайлов', 'Беляев', 'Тарасов']
father_names = ['Александрович', 'Алексеевич', 'Анатольевич',
'Андреевич', 'Антонович', 'Аркадьевич', 'Артемович',
'Бедросович', 'Богданович', 'Борисович', 'Валентинович',
'Валерьевич', 'Васильевич', 'Викторович', 'Витальевич',
'Владимирович', 'Владиславович', 'Вольфович', 'Вячеславович',
'Геннадиевич', 'Георгиевич', 'Григорьевич', 'Данилович',
'Денисович', 'Дмитриевич']
seasons = ['Лето', 'Зима', 'Осень', 'Весна']
pool_types = ['озеро', 'море', 'река', 'пруд', 'ручей']
plant_types = ['хвойный_лес', 'лиственный_лес', 'болото', 'пойма', 'вырубка']

```

```

def generate_description(length):
    letters = string.ascii_lowercase
    return ''.join(random.choice(letters) for i in range(length))

```

```

def generate_sp_name(lang):
    if lang == 'eng':
        letters = string.ascii_lowercase
        return random.choice(letters).upper() + \
            ''.join(random.choice(letters) for i in range(10))
    + '_' + \
        ''.join(random.choice(letters) for i in range(10))
    else:
        return rstr.xeger(r'АЯая[-][-]{10}')

```

```

def generate_path():
    return 'C:/Users/Admin/' + generate_description(10)

```

```

def generate_time(min_year=1900, max_year=datetime.now().year):
    start = datetime(min_year, 1, 1, 00, 00, 00)
    years = max_year - min_year + 1
    end = start + timedelta(days=365 * years)
    return start + (end - start) * random.random()

```