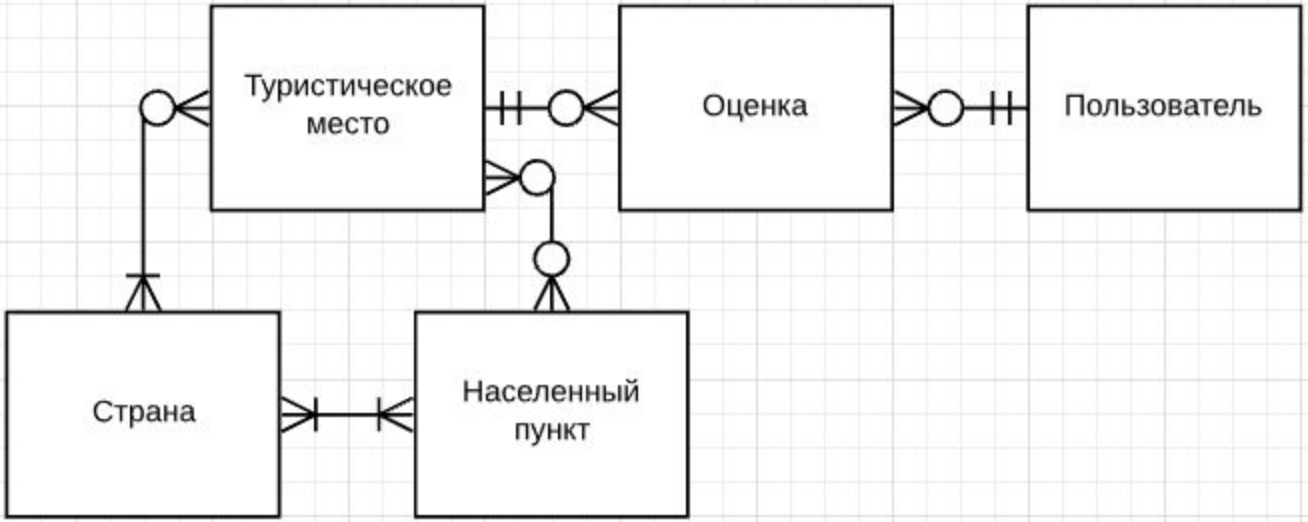
Проект по курсу “Базы данных”

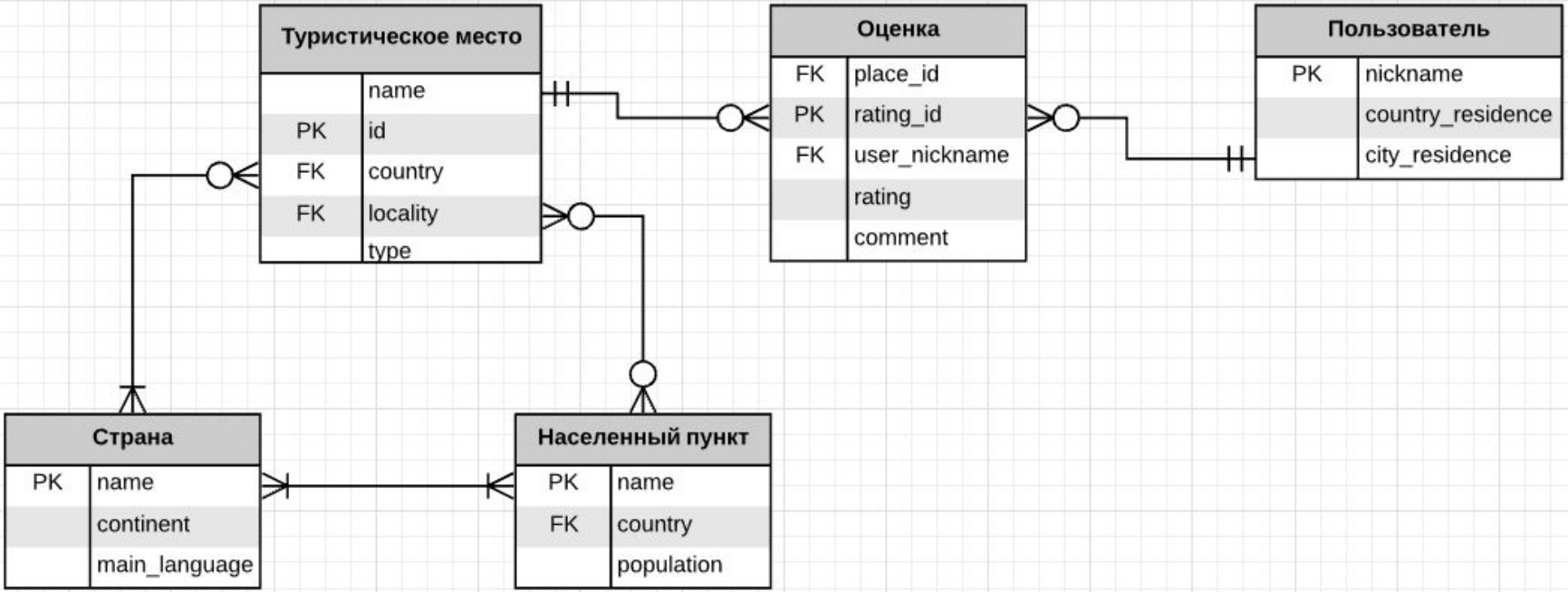
“База туристических мест с рейтингом”

Выполнил Кудрявцев Иван
Семинарист Медведева Анастасия
2019 год

Концептуальная модель



Логическая модель



	country_name	continent	main_language
1	Russia	Europe	Russian
2	The United Kingdom	Europe	English
3	The United States of America	North America	English
4	Japan	Asia	Japanese
5	Korea	Asia	Korean
6	Spain	Europe	Spanish
7	New Zealand	Australia/Oceania	English
8	Sweden	Europe	Swedish
9	Angola	Africa	Portuguese
10	Brazil	South America	Portuguese
11	Israel	Asia	Hebrew

	locality_name	country	population
1	Saint Petersburg	Russia	5350000
2	Dmitrov	Russia	61000
3	London	The United Kingdom	8500000
4	Los Angeles	The United States of America	3900000
5	Miami	The United States of America	463000
6	Kyoto	Japan	1400000
7	Barcelona	Spain	1600000
8	Christchurch	New Zealand	375000
9	Goteborg	Sweden	570000
10	Luanda	Angola	2400000
11	Brazilia	Brazil	2400000
12	Jerusalem	Israel	857000
13	Moscow	Russia	12500000
14	Tokyo	Japan	13700000
15	Sao Paulo	Brazil	12100000

	spot_name	id	country	locality	type
1	Moscow Kremlin	1	Russia	Moscow	Kremlin
2	Bolshoi Theatre	2	Russia	Moscow	Theatre
3	Sokolniki Park	3	Russia	Moscow	Park
4	The State Hermitage Museum	4	Russia	Saint Petersburg	Museum
5	The Lahta Center	5	Russia	Saint Petersburg	Skyscraper
6	Saint Isaacs Cathedral	6	Russia	Saint Petersburg	Cathedral
7	Dmitrovs Kremlin	7	Russia	Dmitrov	Kremlin
8	Tower Bridge	8	The United Kingdom	London	Bridge
9	Museum of London	9	The United Kingdom	London	Museum
10	Hollywood sign	10	The United States of America	Los Angeles	Sign
11	The Vincent Thomas Bridge	11	The United States of America	Los Angeles	Bridge
12	Miami Beach	12	The United States of America	Miami	Beach
13	Tokyo Skytree	13	Japan	Tokyo	Tower
14	Kyoto Imperial Palace	14	Japan	Kyoto	Palace
15	Kyoto Tower	15	Japan	Kyoto	Tower
16	Palau Nacional	16	Spain	Barcelona	Palace
17	ChristChurch Cathedral	17	New Zealand	Christchurch	Cathedral
18	Sao Paulo Cathedral	18	Brazil	Sao Paulo	Cathedral
19	The Garden Tomb	19	Israel	Jerusalem	Tomb

	place_id	rating_id	user_nickname	rating	comment
1	1	101	kayman233	10	Cool!
2	3	102	kayman233	8	A nice place
3	7	103	kayman233	8	I live here!!
4	1	104	user123	8	
5	13	105	user123	8	
6	19	106	user123	7	
7	15	107	user123	10	
8	15	108	user123	10	
9	16	109	kayman233	9	Amazing
10	13	110	chingchong	9	Yeee
11	14	111	chingchong	9	Yee
12	15	112	chingchong	7	Yeee
13	4	113	xxx_Vasya_xxx	5	BORING
14	5	114	xxx_Vasya_xxx	5	BORING
15	6	115	xxx_Vasya_xxx	5	BORING
16	1	116	Maria	10	WOW
17	9	118	Maria	10	OWO
18	17	119	Maria	10	oWo
19	1	120	Valter	10	Muy bonito
20	10	121	Valter	9	Me gusta
21	11	122	Valter	8	Bonito
22	7	123	Valter	9	Me encanta
23	8	117	Maria	6	UWU

	nickname	country_residence	city_residence
1	kayman233	Russia	Dmitrov
2	user123	France	Paris
3	tourist	Belarus	Gomel
4	chingchong	China	Beijing
5	xxx_Vasya_xxx	Russia	Moscow
6	Maria	Russia	Rostov-on-Don
7	Valter	Mexico	Mexico-city

Польза

1. Можно использовать для backend-а
2. Использовать в домашних целях
3. Данные можно анализировать!

Запросы

```sql
/*1. Англоговорящие страны*/
SELECT tourism.countries.country_name
FROM tourism.countries
WHERE countries.main_language = 'English';

/*2. Страны и общее суммарное население по ним*/
SELECT tourism.countries.country_name, SUM(tourism.localities.population)
FROM tourism.countries
INNER JOIN tourism.localities ON tourism.localities.country = tourism.countries.country_name
GROUP BY tourism.countries.country_name
ORDER BY tourism.countries.country_name;

/*3. Рейтинг всех достопримечательностей по убыванию*/
SELECT tourism.tourist_spots.spot_name, avg(tourism.ratings.rating)
FROM tourism.tourist_spots
INNER JOIN tourism.ratings ON tourism.ratings.place_id = tourism.tourist_spots.id
GROUP BY tourism.tourist_spots.spot_name
ORDER BY avg(tourism.ratings.rating) DESC;
```

```sql
/*4. Страны, у которых нет достопримечательностей*/
SELECT tourism.countries.country_name
FROM tourism.countries
FULL JOIN tourism.tourist_spots ON countries.country_name = tourist_spots.country
GROUP BY country_name
HAVING count(tourist_spots.spot_name) = 0;

/*5. Средниая оценка, поставленная пользователем по убыванию, а если нет оценок, то вывести 0*/
SELECT tourism.users.nickname, ROUND(coalesce(avg(tourism.ratings.rating), 0), 2) AS "average_rating"
FROM tourism.users
FULL JOIN tourism.ratings on users.nickname = ratings.user_nickname
GROUP BY nickname
ORDER BY avg(tourism.ratings.rating) DESC NULLS LAST;

/*6. Вывод населения городов по странам с нарастающим итогом*/
SELECT tourism.localities.locality_name, tourism.localities.country,
       SUM(localities.population) OVER (PARTITION BY localities.country ORDER BY localities.locality_name) AS "total_population"
FROM tourism.localities;

/*7. Вывод нарастающего среденего населения(с округлением*/
SELECT tourism.localities.locality_name, tourism.localities.country,
       ROUND(AVG(localities.population) OVER (ORDER BY localities.population), 0)AS "average_population"
FROM tourism.localities;
```

```sql
/*1) Представление по среднему рейтингу достопримечательностей города*/
CREATE OR REPLACE VIEW tourism.localities_view AS
SELECT locality,
       AVG(rating) AS rating
FROM tourism.tourist_spots
       LEFT JOIN tourism.ratings
                ON tourist_spots.id = tourism.ratings.place_id
GROUP BY locality
ORDER BY locality;

/*2) Представление для храмов*/
CREATE VIEW tourism.cathedrals_view AS
SELECT spot_name,
       locality
FROM tourism.tourist_spots
WHERE type = 'Cathedral';

/*3) Самые популярные места*/
CREATE VIEW tourism.most_popular_view AS
SELECT spot_name,
       locality,
       COUNT(rating) AS visits
FROM tourism.tourist_spots
       LEFT JOIN tourism.ratings
                ON tourist_spots.id = tourism.ratings.place_id
GROUP BY spot_name, locality
ORDER BY COUNT(rating) DESC;

/*Уменьшить население у городов с населением свыше 10 миллионов(например чтобы не учитывать часть населения живущих в пригороде)*/
CREATE OR REPLACE FUNCTION tourism.decrease_population (population_limit integer)
RETURNS TABLE(locality_name VARCHAR, population BIGINT) AS $$
  UPDATE tourism.localities
  SET population = population - 1000000
  WHERE population > population_limit;
  SELECT locality_name, population
  FROM tourism.localities
  WHERE population > population_limit;
$$ LANGUAGE SQL;
```