ASTANA IT
UNIVERSITY

INFORMATION AND COMMUNICATION TECHNOLOGIES
Team 6
Project name: School management system

Student name: Temir Mendigali
Group name: SE-2019
Tutor name: Gulnara Abitova
Date: 27.11.2020

Nur-Sultan 2020

Table of contents

Abstracts

- **Project field**
  - Education
- **Project name**
  - School management system
- **Company**
  - Karagandy “Bilim-Innovation” lyceum #1
- **Project objectives**
  - Learn core principles of database architecture
  - Get better at writing SQL queries
  - Understand what is database normalization
  - Practice at designing entity-relationship diagrams
  - Create complete database

Introduction

As we all know, today the state program Digital Kazakhstan is being implemented, the purpose of which is to improve the quality of life of the population through the use of digital technologies. One of the areas of this program is the digitalization of education, in particular the digitalization of educational institutions. For this reason, I decide to choose this topic.

I decided to implement this global project locally, in the territory of my former school. When I was in school, I often faced various problems that I tried to solve in this project. One such problem was the paperwork, namely sorting documents by year and subject. The school kept the achievements of students at various Olympiads and competitions. In addition, I was a robotics Olympiad and I needed to keep track of everyone who participated in this club. So, the goal of my project is to create a database which will store information about school, its students, teachers and alumnees.

The Database Management System is practical because users can access it from various platforms at any time. Also, a digital database eliminates the need to store a huge amount of paper documentation, and more importantly, speeds up the processing of information at times.

Purpose of the project

My goal is to create database for Karagandy “Bilim-Innovation” lyceum #1. Teachers at the school have been collecting student achievements in paper version for many years. Using the data in this form is not very convenient and for this reason, later they have begun transferring these papers into digital format, in the form of Microsoft Excel and Word files. But this option also has its drawbacks, because in order to access the data you need to transfer it from one computer to another in various ways, for example, via a USB flash drive. So therefore, I believe that transferring all this information to a database management system will be a very correct, relevant and logical solution of this problem.

The school keeps track of students, teachers, graduates, tutors and other staff. All these users will be able to register a personal account. The school has around 250 students, 40 teachers and administrators, 10 tutors. There are nearly 30 classrooms, 10 classes, each class has its own curator, tutor, classroom teacher, 25 subjects, 10 clubs. The database will need to contain their information, such as name, surname, birthday, class, his contacts, the clubs that the student is a member of, the subject in which he participates in the Olympiads, and it is also necessary to save the achievements of students at these Olympiads and competitions. Olympiads will contain following attributes name of the Olympiad/competition, a result (medal, certificate), place and date of the event. At the same time, the teachers will keep the subject they teach, their experience, qualifications, education, and more. Moreover, there will be other features such as: competitions calendar, lessons schedule, clubs schedule, tracking of books from library.

Project creation outline

To complete this project, I decided to create an action plan that I will rely on during the creation process. I have identified the following steps for myself:

- Determine all entities contained in the project.
- List the business rules to be followed in the project.
- Design ER diagram including all the attributes for each entity and relationship.
- Create all entities, relationships and attributes in the database using SQL.

Entities

The first step in creating this project was to define the entities that will be stored in the database. I have defined 14 entities for this project.

1. User
2. Lesson
3. Club
4. Club role
5. Class
6. Classroom
7. Book
8. Competition
9. Teacher
10. Address
11. Lesson schedule
12. Club schedule
13. Country
14. City

Business rules

After that, I began to draw up business rules that will guide me during the creation of the project and when designing the database. Below is a list of my business rules.

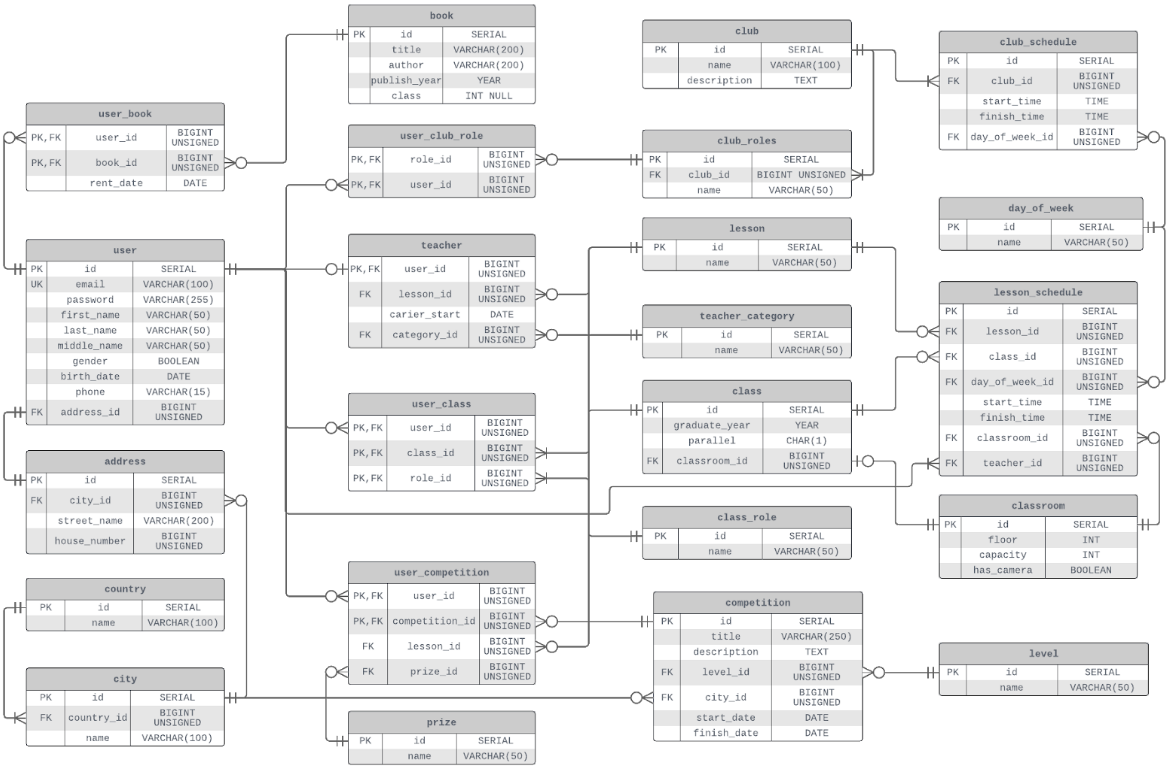
- A user can be associated with a class multiple times and have different roles in that class.
- A club can have many roles.
- Each teacher can teach one lesson.
- A student can join several clubs.
- A student can get an unlimited number of achievements in different competitions.
- A student can take only one prize place within one competition.
- At one competition, a student can participate in only one lesson.
- The student can borrow several books from the library.
- Each class has its own classroom.
- The club may have several scheduled classes.
- The user can only specify one address.
- Each class can attend several lessons.

- Each lesson can be in the lesson schedule several times.
- Different lessons from the lesson schedule can take place in a classroom.
- Each city can only be in one country.
- Each address can be linked to one city.
- The competition can only be held in one city.
- Teacher can have zero or one category.

Attributes

- User can have full name, date of birth, gender, email, password, phone number, and address.
- Address consists of city, street name, and house number.
- Country has name.
- City has name and unique number of country where it is located.
- Teacher has lessons that he teaches, start date of his career, and his category.
- Class has a graduation year, parallel, classroom, and users who have particular roles in the class.
- Classroom has capacity of students, floor, number, and labeled if it has camera or not.
- Competition has title, description, level, location city, and date.
- Book has title, author, publish year, and if it is classbook, then it has a class number for which it is intended.
- Club has name, description and roles.
- Lesson has name.
- Lesson schedule consists of class, lesson, classroom, time of lesson, day of the week.
- Club schedule consists of club, time of meeting, and day of the week.

ER diagram



user_book

PK,FK	user_id	BIGINT UNSIGNED
PK,FK	book_id	BIGINT UNSIGNED
	rent_date	DATE

user

PK	id	SERIAL
UK	email	VARCHAR(100)
	password	VARCHAR(255)
	first_name	VARCHAR(50)
	last_name	VARCHAR(50)
	middle_name	VARCHAR(50)
	gender	BOOLEAN
	birth_date	DATE
	phone	VARCHAR(15)
FK	address_id	BIGINT UNSIGNED

address

PK	id	SERIAL
FK	city_id	BIGINT UNSIGNED
	street_name	VARCHAR(200)
	house_number	BIGINT UNSIGNED

country

PK	id	SERIAL
	name	VARCHAR(100)

city

PK	id	SERIAL
FK	country_id	BIGINT UNSIGNED
	name	VARCHAR(100)

book

PK	id	SERIAL
	title	VARCHAR(200)
	author	VARCHAR(200)
	publish_year	YEAR
	class	INT NULL

user_club_role

PK,FK	role_id	BIGINT UNSIGNED
PK,FK	user_id	BIGINT UNSIGNED

teacher

PK,FK	user_id	BIGINT UNSIGNED
FK	lesson_id	BIGINT UNSIGNED
	carier_start	DATE
FK	category_id	BIGINT UNSIGNED

user_class

PK,FK	user_id	BIGINT UNSIGNED
PK,FK	class_id	BIGINT UNSIGNED
PK,FK	role_id	BIGINT UNSIGNED

user_competition

PK,FK	user_id	BIGINT UNSIGNED
PK,FK	competition_id	BIGINT UNSIGNED
FK	lesson_id	BIGINT UNSIGNED
FK	prize_id	BIGINT UNSIGNED

prize

PK	id	SERIAL
	name	VARCHAR(50)

club

PK	id	SERIAL
	name	VARCHAR(100)
	description	TEXT

club_roles

PK	id	SERIAL
FK	club_id	BIGINT UNSIGNED
	name	VARCHAR(50)

lesson

PK	id	SERIAL
	name	VARCHAR(50)

teacher_category

PK	id	SERIAL
	name	VARCHAR(50)

class

PK	id	SERIAL
	graduate_year	YEAR
	parallel	CHAR(1)
FK	classroom_id	BIGINT UNSIGNED

class_role

PK	id	SERIAL
	name	VARCHAR(50)

competition

PK	id	SERIAL
	title	VARCHAR(250)
	description	TEXT
FK	level_id	BIGINT UNSIGNED
FK	city_id	BIGINT UNSIGNED
	start_date	DATE
	finish_date	DATE

club_schedule

PK	id	SERIAL
FK	club_id	BIGINT UNSIGNED
	start_time	TIME
	finish_time	TIME
FK	day_of_week_id	BIGINT UNSIGNED

day_of_week

PK	id	SERIAL
	name	VARCHAR(50)

lesson_schedule

PK	id	SERIAL
FK	lesson_id	BIGINT UNSIGNED
FK	class_id	BIGINT UNSIGNED
FK	day_of_week_id	BIGINT UNSIGNED
	start_time	TIME
	finish_time	TIME
FK	classroom_id	BIGINT UNSIGNED
FK	teacher_id	BIGINT UNSIGNED

classroom

PK	id	SERIAL
	floor	INT
	capacity	INT
	has_camera	BOOLEAN

level

PK	id	SERIAL
	name	VARCHAR(50)

Database normalization

First Normal Form (1NF)

For a table to be in the First Normal Form, it should follow the following rules:

- Attributes should only have atomic values.
- All attributes in a table should have unique names.
- Values stored in a column should be of the same domain.

Second Normal Form (2NF)

For a table to be in the Second Normal Form, it should follow the following rules:

- It is in the 1NF.
- It does not have Partial Dependency.

Third Normal Form (3NF)

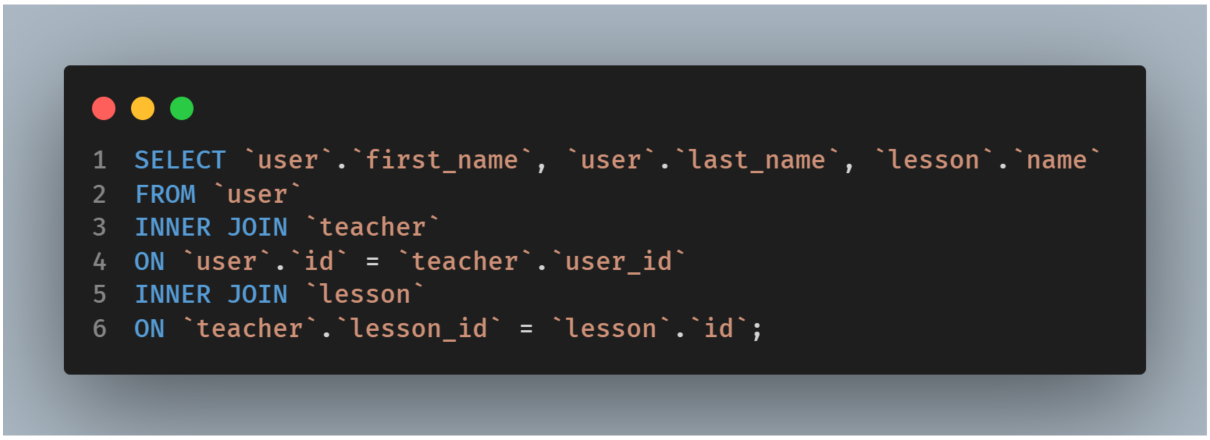
For a table to be in the Third Normal Form, it should follow the following rules:

- It is in the 2NF.
- It does not have Transitive Dependency.

When considering and studying the above ER diagram, it can be argued that all tables here are in 3NF, since they meet all the requirements. Each table has a primary key. The diagram shows all foreign keys and the relationship between them.

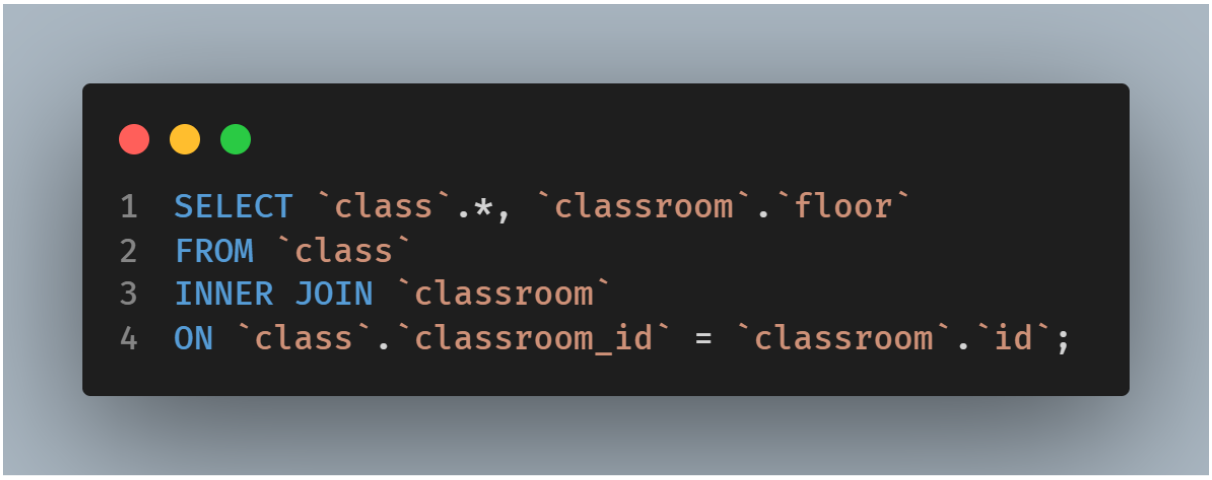
Example queries

1. Show list of the teachers and lesson that each of them teach



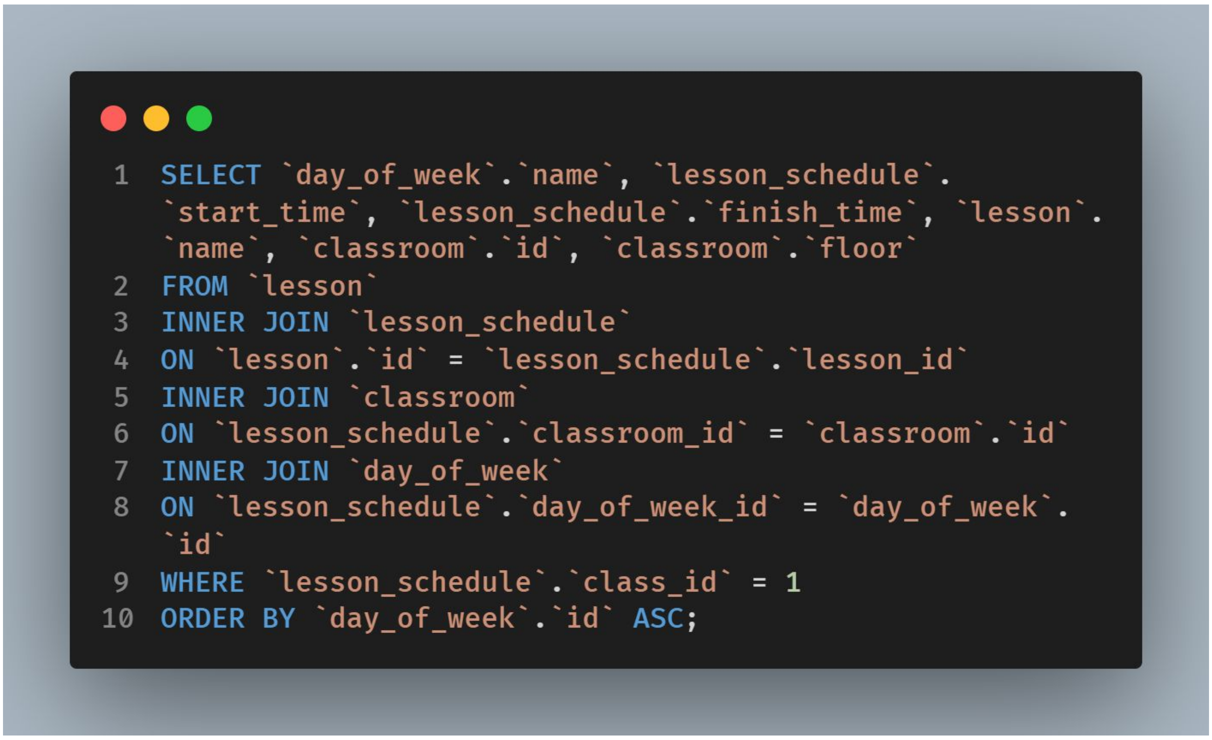
```sql
SELECT `user`.`first_name`, `user`.`last_name`, `lesson`.`name`
FROM `user`
INNER JOIN `teacher`
ON `user`.`id` = `teacher`.`user_id`
INNER JOIN `lesson`
ON `teacher`.`lesson_id` = `lesson`.`id`;
```

2. Show list of classes and their classrooms



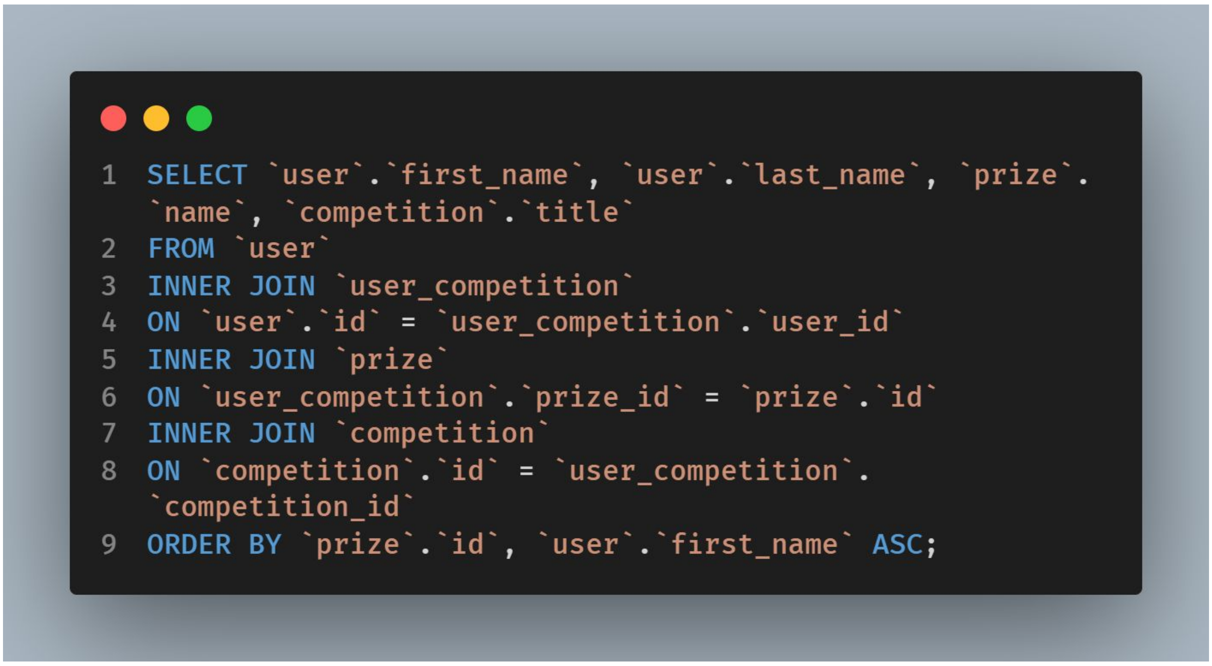
```sql
SELECT `class`.*, `classroom`.`floor`
FROM `class`
INNER JOIN `classroom`
ON `class`.`classroom_id` = `classroom`.`id`;
```

3. Show lesson schedule for class with id equal 1



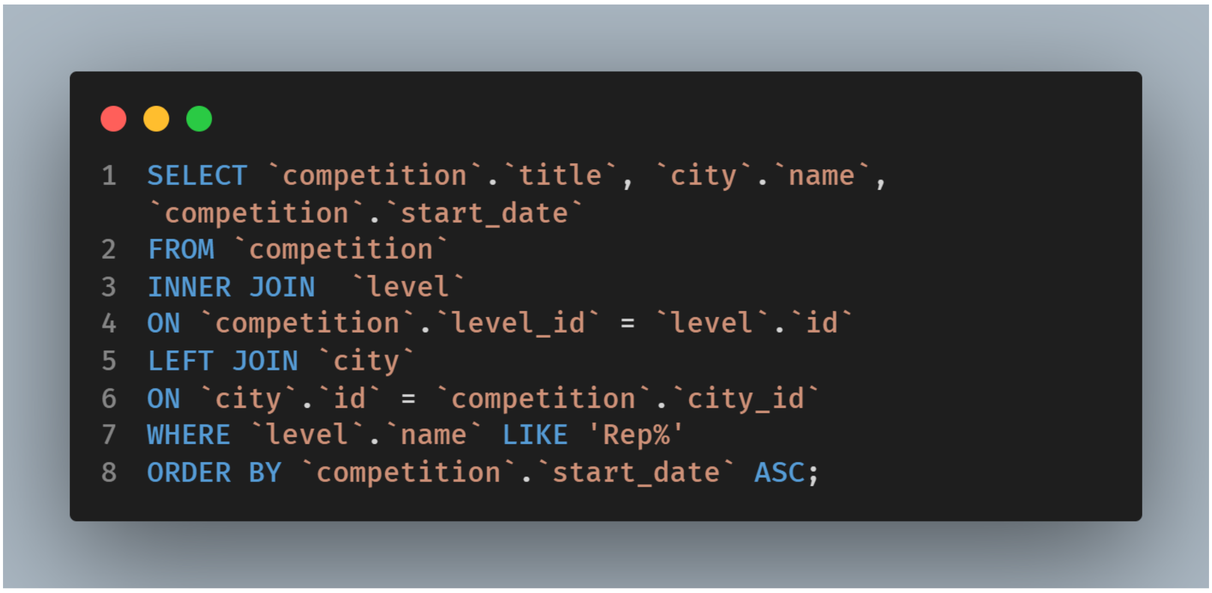
```sql
SELECT `day_of_week`.`name`, `lesson_schedule`.`start_time`, `lesson_schedule`.`finish_time`, `lesson`.`name`, `classroom`.`id`, `classroom`.`floor`
FROM `lesson`
INNER JOIN `lesson_schedule`
ON `lesson`.`id` = `lesson_schedule`.`lesson_id`
INNER JOIN `classroom`
ON `lesson_schedule`.`classroom_id` = `classroom`.`id`
INNER JOIN `day_of_week`
ON `lesson_schedule`.`day_of_week_id` = `day_of_week`.`id`
WHERE `lesson_schedule`.`class_id` = 1
ORDER BY `day_of_week`.`id` ASC;
```

4. Show achievements of every student on any competition



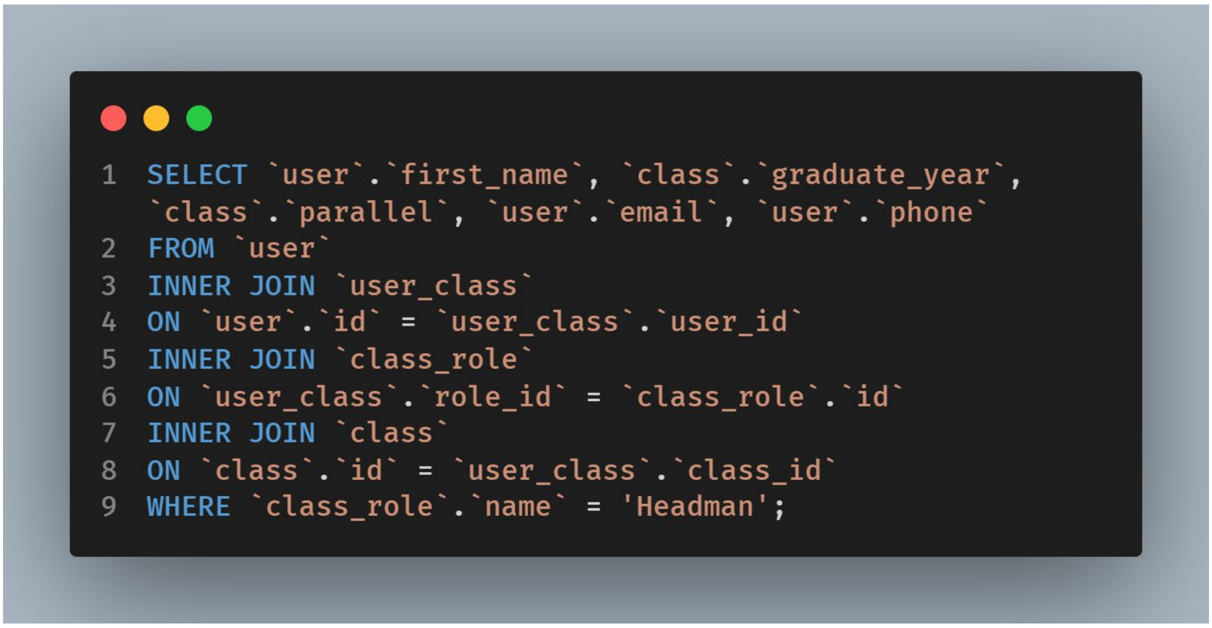
```sql
SELECT `user`.`first_name`, `user`.`last_name`, `prize`.`name`, `competition`.`title`
FROM `user`
INNER JOIN `user_competition`
ON `user`.`id` = `user_competition`.`user_id`
INNER JOIN `prize`
ON `user_competition`.`prize_id` = `prize`.`id`
INNER JOIN `competition`
ON `competition`.`id` = `user_competition`.`competition_id`
ORDER BY `prize`.`id`, `user`.`first_name` ASC;
```

5. Show competitions where level of competition starts with "Rep"



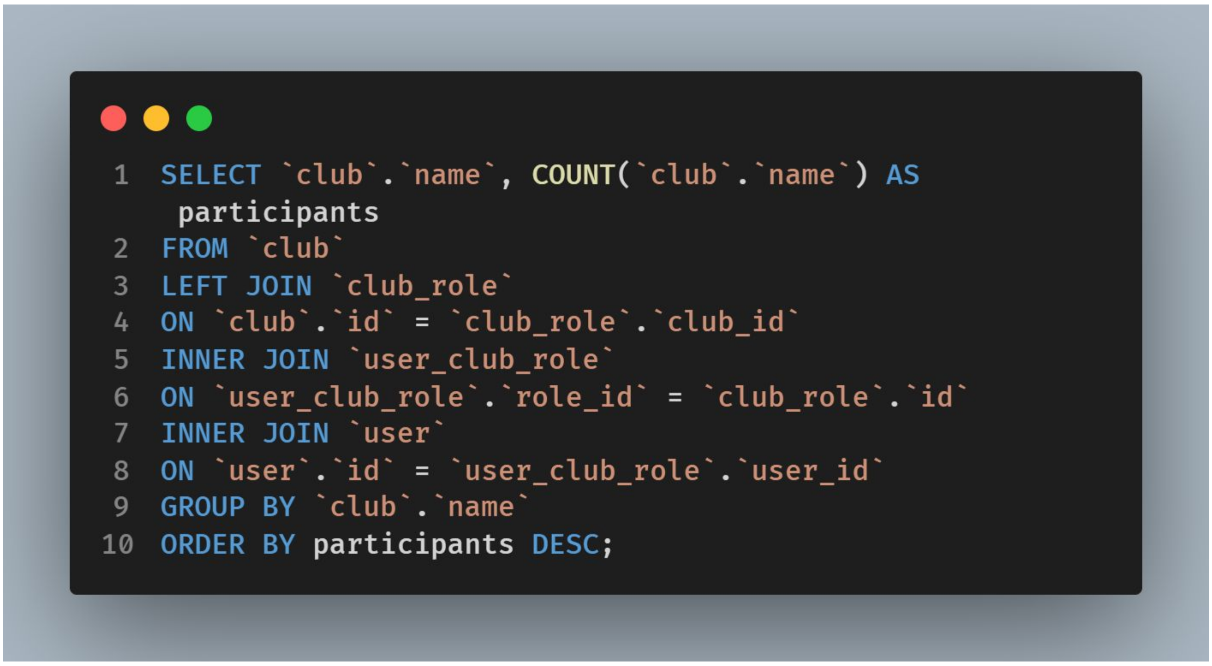
```sql
SELECT `competition`.`title`, `city`.`name`, `competition`.`start_date`
FROM `competition`
INNER JOIN  `level`
ON `competition`.`level_id` = `level`.`id`
LEFT JOIN `city`
ON `city`.`id` = `competition`.`city_id`
WHERE `level`.`name` LIKE 'Rep%'
ORDER BY `competition`.`start_date` ASC;
```

6. Show information about each headman in school



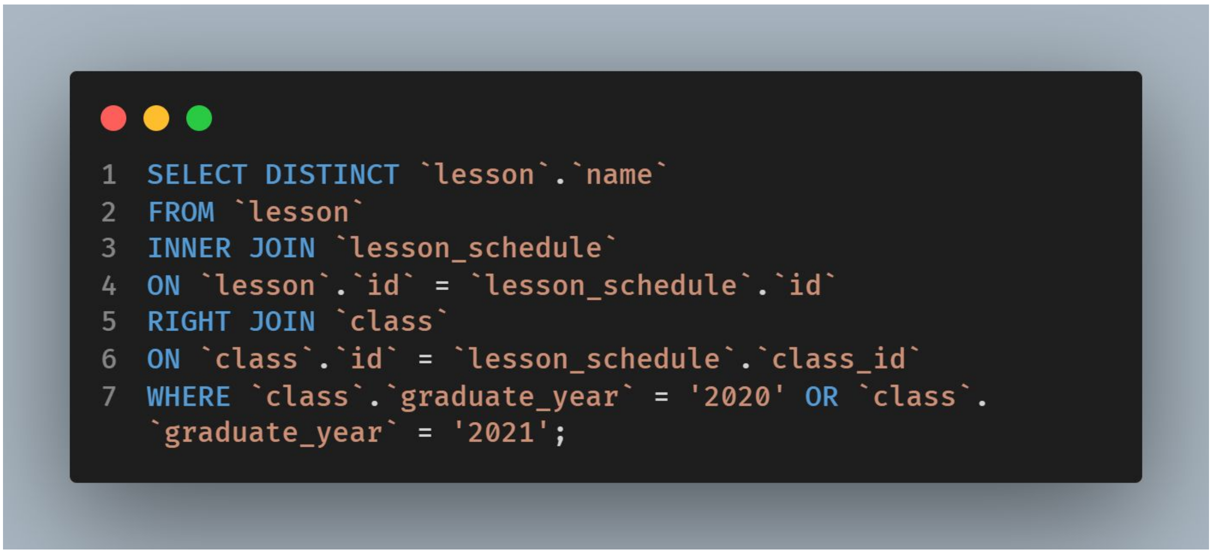
```sql
SELECT `user`.`first_name`, `class`.`graduate_year`, `class`.`parallel`, `user`.`email`, `user`.`phone`
FROM `user`
INNER JOIN `user_class`
ON `user`.`id` = `user_class`.`user_id`
INNER JOIN `class_role`
ON `user_class`.`role_id` = `class_role`.`id`
INNER JOIN `class`
ON `class`.`id` = `user_class`.`class_id`
WHERE `class_role`.`name` = 'Headman';
```

7. Show number of participants for each club



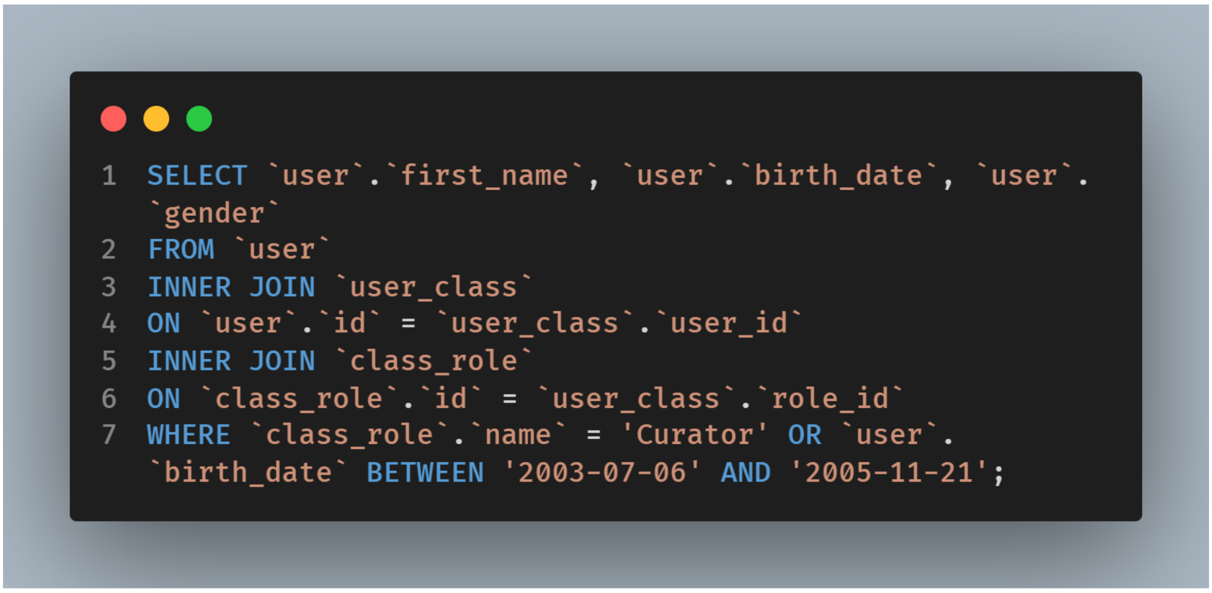
```sql
SELECT `club`.`name`, COUNT(`club`.`name`) AS participants
FROM `club`
LEFT JOIN `club_role`
ON `club`.`id` = `club_role`.`club_id`
INNER JOIN `user_club_role`
ON `user_club_role`.`role_id` = `club_role`.`id`
INNER JOIN `user`
ON `user`.`id` = `user_club_role`.`user_id`
GROUP BY `club`.`name`
ORDER BY participants DESC;
```

8. Show different lessons that are attended by students who will graduate in 2020 or in 2021



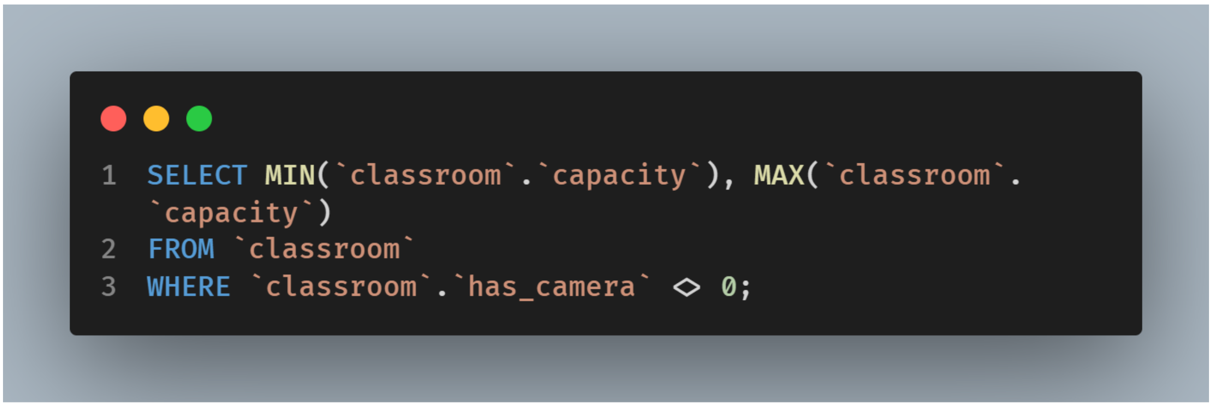
```sql
SELECT DISTINCT `lesson`.`name`
FROM `lesson`
INNER JOIN `lesson_schedule`
ON `lesson`.`id` = `lesson_schedule`.`id`
RIGHT JOIN `class`
ON `class`.`id` = `lesson_schedule`.`class_id`
WHERE `class`.`graduate_year` = '2020' OR `class`.`graduate_year` = '2021';
```

9. Show only curators and user with birthday between 6 july 2003 and 21 november 2005



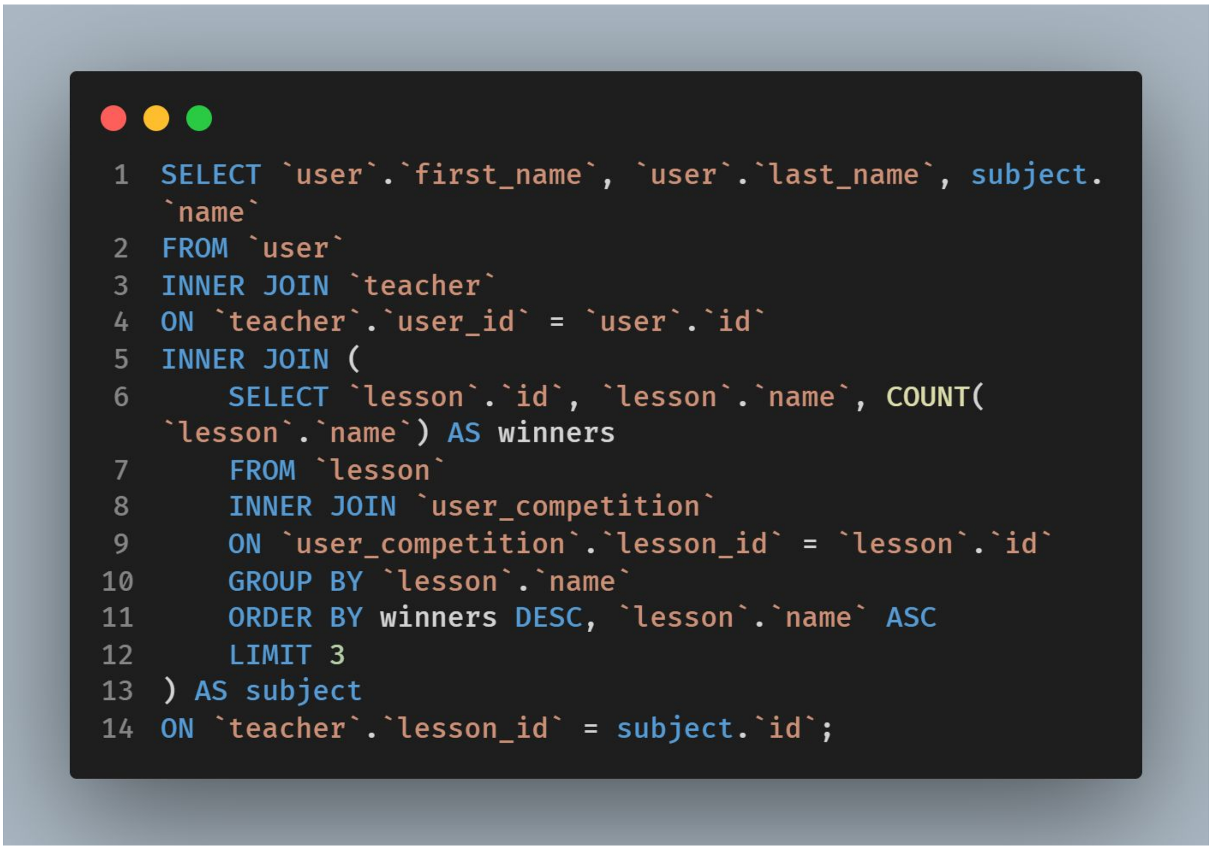
```sql
SELECT `user`.`first_name`, `user`.`birth_date`, `user`.`gender`
FROM `user`
INNER JOIN `user_class`
ON `user`.`id` = `user_class`.`user_id`
INNER JOIN `class_role`
ON `class_role`.`id` = `user_class`.`role_id`
WHERE `class_role`.`name` = 'Curator' OR `user`.`birth_date` BETWEEN '2003-07-06' AND '2005-11-21';
```

10. Show minimum and maximum capacity of students that can fit in one classroom which has no camera inside



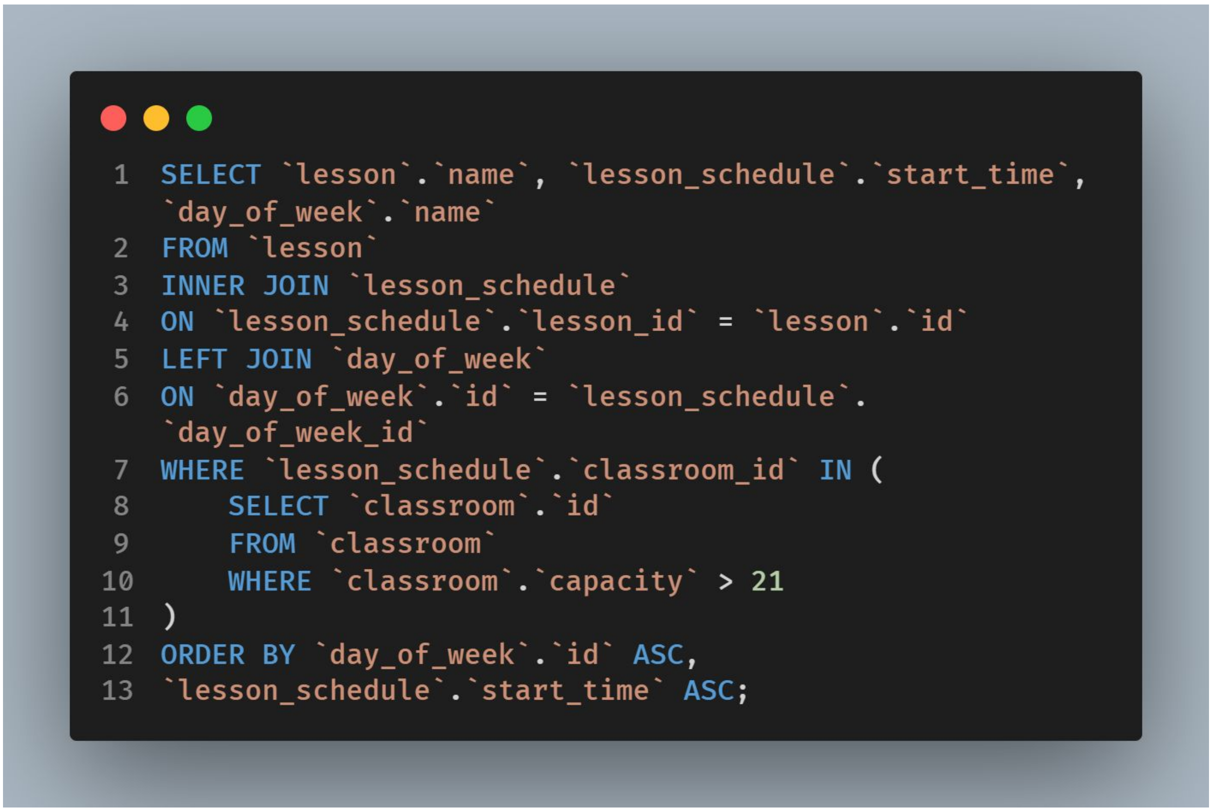
```sql
SELECT MIN(`classroom`.`capacity`), MAX(`classroom`.`capacity`)
FROM `classroom`
WHERE `classroom`.`has_camera` <> 0;
```

11. Show list of teachers who teach 3 lessons in which students win the most prizes in competition



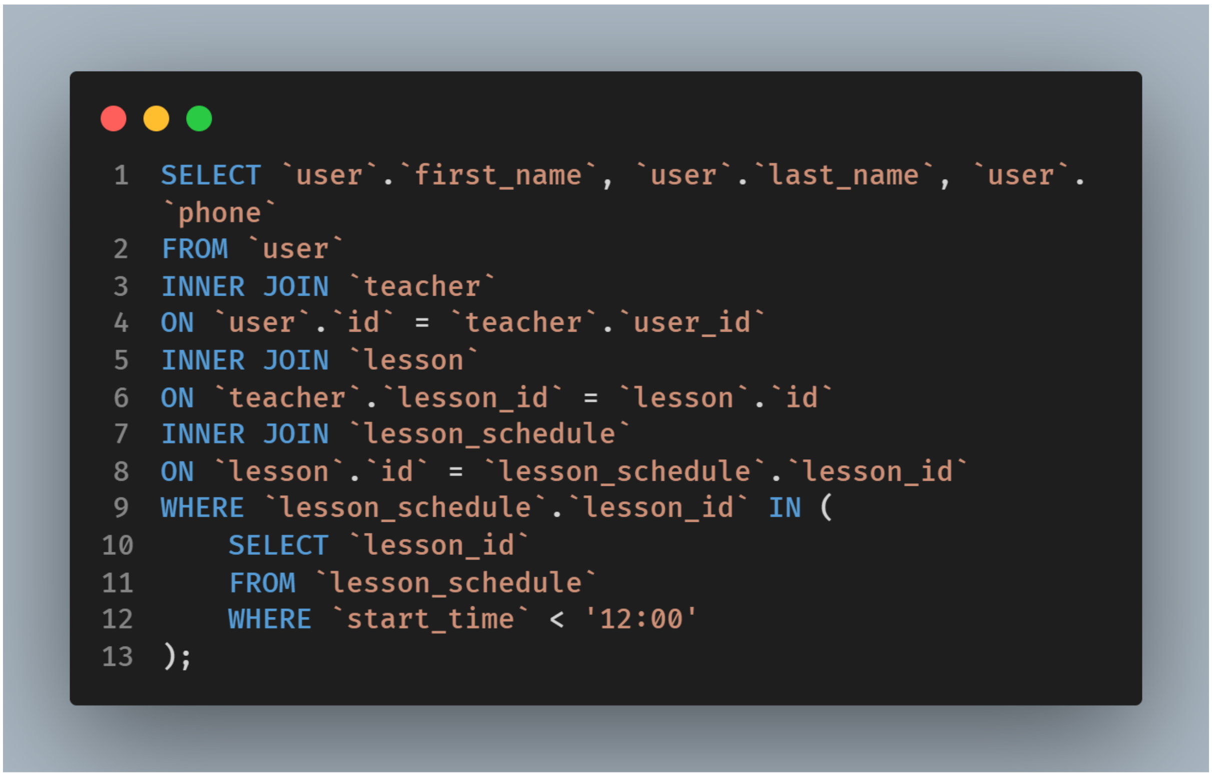
```
SELECT `user`.`first_name`, `user`.`last_name`, subject.`name`
FROM `user`
INNER JOIN `teacher`
ON `teacher`.`user_id` = `user`.`id`
INNER JOIN (
    SELECT `lesson`.`id`, `lesson`.`name`, COUNT(`lesson`.`name`) AS winners
    FROM `lesson`
    INNER JOIN `user_competition`
    ON `user_competition`.`lesson_id` = `lesson`.`id`
    GROUP BY `lesson`.`name`
    ORDER BY winners DESC, `lesson`.`name` ASC
    LIMIT 3
) AS subject
ON `teacher`.`lesson_id` = subject.`id`;
```

12. Show lessons that take place in classrooms with a capacity of more than 21 people



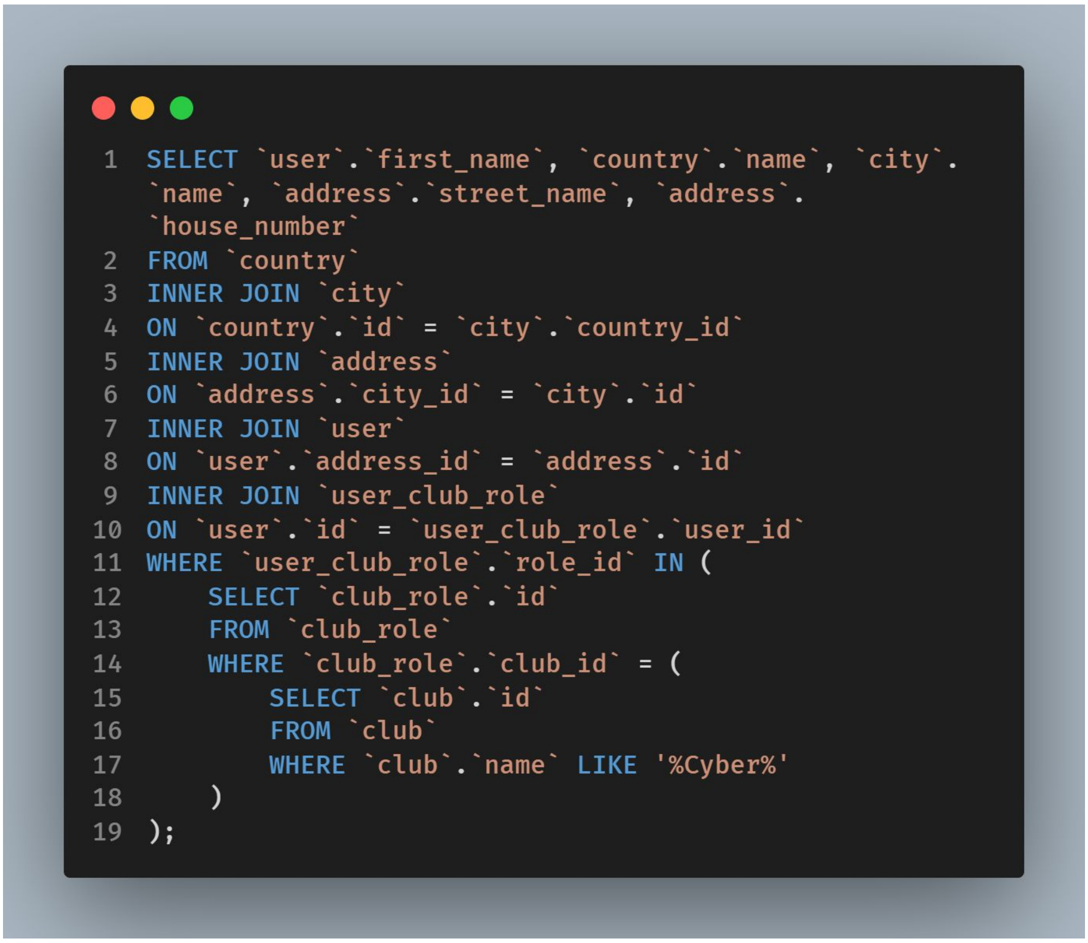
```
SELECT `lesson`.`name`, `lesson_schedule`.`start_time`, `day_of_week`.`name`
FROM `lesson`
INNER JOIN `lesson_schedule`
ON `lesson_schedule`.`lesson_id` = `lesson`.`id`
LEFT JOIN `day_of_week`
ON `day_of_week`.`id` = `lesson_schedule`.`day_of_week_id`
WHERE `lesson_schedule`.`classroom_id` IN (
    SELECT `classroom`.`id`
    FROM `classroom`
    WHERE `classroom`.`capacity` > 21
)
ORDER BY `day_of_week`.`id` ASC,
`lesson_schedule`.`start_time` ASC;
```

13. Show details about teachers who teach lessons earlier than at 12:00



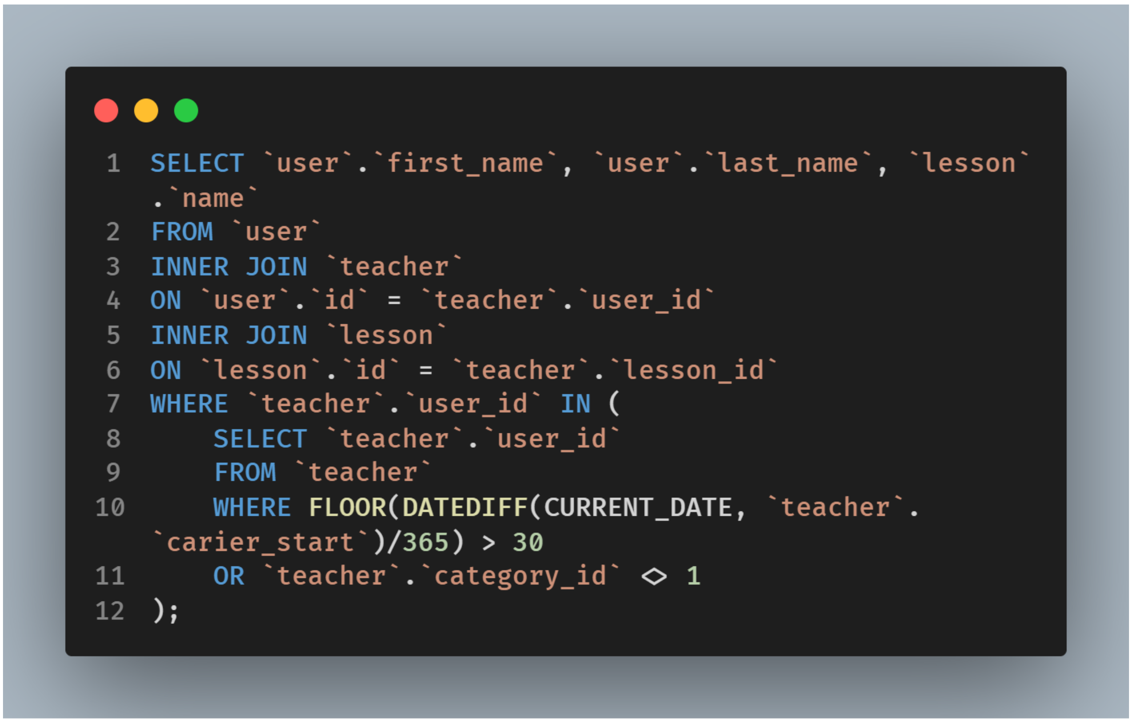
```sql
SELECT `user`.`first_name`, `user`.`last_name`, `user`.`phone`
FROM `user`
INNER JOIN `teacher`
ON `user`.`id` = `teacher`.`user_id`
INNER JOIN `lesson`
ON `teacher`.`lesson_id` = `lesson`.`id`
INNER JOIN `lesson_schedule`
ON `lesson`.`id` = `lesson_schedule`.`lesson_id`
WHERE `lesson_schedule`.`lesson_id` IN (
    SELECT `lesson_id`
    FROM `lesson_schedule`
    WHERE `start_time` < '12:00'
);
```

14. Show first name, country, city, address of users who participate in cybersport club



```sql
SELECT `user`.`first_name`, `country`.`name`, `city`.`name`, `address`.`street_name`, `address`.`house_number`
FROM `country`
INNER JOIN `city`
ON `country`.`id` = `city`.`country_id`
INNER JOIN `address`
ON `address`.`city_id` = `city`.`id`
INNER JOIN `user`
ON `user`.`address_id` = `address`.`id`
INNER JOIN `user_club_role`
ON `user`.`id` = `user_club_role`.`user_id`
WHERE `user_club_role`.`role_id` IN (
    SELECT `club_role`.`id`
    FROM `club_role`
    WHERE `club_role`.`club_id` = (
        SELECT `club`.`id`
        FROM `club`
        WHERE `club`.`name` LIKE '%Cyber%'
    )
);
```

15. Show the most experienced teachers (those who have working experience > 30 years or any teacher category excluding "Without category")



```sql
SELECT `user`.`first_name`, `user`.`last_name`, `lesson`.`name`
FROM `user`
INNER JOIN `teacher`
ON `user`.`id` = `teacher`.`user_id`
INNER JOIN `lesson`
ON `lesson`.`id` = `teacher`.`lesson_id`
WHERE `teacher`.`user_id` IN (
    SELECT `teacher`.`user_id`
    FROM `teacher`
    WHERE FLOOR(DATEDIFF(CURRENT_DATE, `teacher`.`carier_start`)/365) > 30
    OR `teacher`.`category_id` <> 1
);
```

Potential future upgrades

This project can already be used, but like any other project, it can be improved. I think that in the future it will be advisable to add support for other languages, which will allow storing information in several languages. In addition, you can add the ability to create blog posts in which you can write about interesting events that happened at the school. It will also be possible to add the ability to create a calendar of upcoming events, which will be very useful. It will also be possible to realize the opportunity for incoming sixth graders to apply to study at the school. In general, this project can still grow and improve a lot.

Conclusion

It was a very interesting and, above all, useful project. As a result of the work done, we have a database that provides the school with the ability to register a personal account for each user. Each user can add information about themselves, such as name, date of birth, gender, place of residence, cell phone number, etc. In addition, the database allows you to store student achievements and get immediate access to them, with the ability to filter and sort the data. For each competition, its name, description, city and date of the event, as well as the scale of the competition, for example, international or republican, etc., are stored. Thanks to this, teachers will have less workload and, accordingly, they will be able to devote more time to teaching students than to paperwork. Now students can follow the schedules of lessons and club meetings, and see the members of each of the clubs. Students have the opportunity to view the books available in the library and, if they wish, take it for a while to read. It is also worth noting that teachers can add information that applies only to them, for example, the date of the beginning of their working career, the subject they teach, their teaching category.

Also, I personally learned a lot of important and valuable experience from this project. I improved my query writing skills and got a better understanding of how to build a database from scratch. I also got acquainted with database normalization and applied 3 normal forms in this project. Of course this is not the limit and there are other normal forms besides the three and I plan to learn how to apply the remaining forms in the future. For this project, I also learned how to create an entity relationship diagram using crow's foot notation. At first glance it seems difficult, but with a little practice it becomes much easier.

List of used literature and resources

Learn SQL (using MySQL) in One Day and Learn It Well. SQL for Beginners with Hands-on Project. The only book you need to start coding in SQL immediately. - By Jamie Chan.

Learning SQL Generate, Manipulate, and Retrieve Data Third Edition. - By Alan Beaulieu.

https://mariadb.com/kb/en/documentation/

https://www.w3schools.com/sql/

https://dev.mysql.com/doc/