

Схема данных GraphQL (<https://graphql.org/learn/schema/>)

Схема для курса

```
type Course {  
  id: ID!  
  title: String!  
  description: String!  
  cost: Int!  
  enrolledStudentsCount: Int # Бонусное поле, показывающее количество зарегистрированных студентов  
}
```

Схема для студента

```
type Student {  
  id: ID!  
  firstName: String!  
  lastName: String!  
  age: Int!  
  courses: [Course!]!  
}
```

Схема для запросов

```
type Query {  
  # Получение всех курсов с возможностью фильтрации по стоимости  
  courses(costLessThan: Int): [Course!]!  
  
  # Получение всех студентов с возможностью пагинации  
  students(offset: Int, limit: Int): [Student!]!  
  
  # Получение информации о курсе по его ID  
  course(id: ID!): Course  
  
  # Получение информации о студенте по его ID  
  student(id: ID!): Student  
}
```

```
# Схема для мутаций
type Mutation {
  # Добавление нового курса
  createCourse(title: String!, description: String!, cost: Int!): Course!

  # Редактирование существующего курса
  updateCourse(id: ID!, title: String, description: String, cost: Int): Course

  # Удаление курса
  deleteCourse(id: ID!): Boolean

  # Добавление нового студента
  createStudent(firstName: String!, lastName: String!, age: Int!): Student!

  # Редактирование существующего студента
  updateStudent(id: ID!, firstName: String, lastName: String, age: Int): Student

  # Удаление студента
  deleteStudent(id: ID!): Boolean

  # Регистрация студента на курс
  enrollStudent(studentId: ID!, courseId: ID!): Student!

  # Отмена регистрации студента с курса
  unenrollStudent(studentId: ID!, courseId: ID!): Student!
}
```