МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»

Основная профессиональная образовательная программа
Направление подготовки 09.03.01 Информатика и вычислительная техника
Направленность (профиль) «Технологии разработки программного обеспечения»
форма обучения – очная

ОТЧЁТ

по реализации проекта для дисциплины «Базы данных»
по направлению “09.03.01 – Информатика и вычислительная техника”
(профиль: “Технологии разработки программного обеспечения ”)

Преподаватель: к.ф-м.н., доцент кафедры ИТиЭО

_____
(Жуков Н. Н.)

Преподаватель: ассистент кафедры ИТиЭО

_____
(Иванова Е. А.)

Студенты 2 курса:
Стецук М.Н. _____
Каргаполов Д.А. _____

Санкт-Петербург
2023

Оглавление

Ответственные

Перед началом рассмотрения обязанностей каждого из участников данного проекта, стоит отметить совместное прорабатывание каждого процесса разработки представленной базы данных. Благодаря идейному дополнению каждого аспекта данный проект был выполнен успешно.

Стецук М. Н – является идейным лидером команды. В обязанности Стецука М.Н. входили: определение предметной области данной базы данных, проектирование схемы и процесс нормализации базы данных, отладка и тестирование проекта.

Каргаполов Д.А. – разработчик проекта. В обязанности Каргаполова Д.А. входили: создание сущностей, наполнение и выстраивание логики атрибутов для них, заполнение базы данных, адаптирование создаваемой архитектуры.

Оба участника составляли текстовую отчётность и проектировали её логическую последовательность.

Предметная область

Наша образовательная организация является творческим многопрофильным учреждением дополнительного образования детей. Все возможные занятия развивают знания учеников в определённом направлении. А также они делятся по направлениям, ориентированным на развитие навыков в одной определённой области знаний.

За каждым направлением закреплён куратор, который способствует организации деятельности по различным дисциплинам данного направления. Каждую дисциплину ведёт высококвалифицированный специалист.

Ученик волен выбирать направления и дисциплины на своё усмотрение, тем самым составляя свою рабочую нагрузку.

Наше учебное заведение также организует различные мероприятия для учеников. Со временем их будет становиться только больше!

Для контроля над образовательным процессом и ведения ведомости наша образовательная организация использует самостоятельно разработанную базу данных.

Основная информация, хранимая нашей организацией:

●Информация о наших учениках;

●Информация о родителях или опекунах наших учеников (данная информация имеет важное значение, нам всегда необходмо оставаться на связи, в случае отсутствия ученика на занятиях или возникновения чрезвычайных ситуаций);

●Информация о наших преподавателях (наша организация располагает всей необходмой информацией, которуая демонстрирует квалифицированность и качества образования, предоставляемого нашими преподавателями);

●Информация о направлениях подготовки (для удобства выбора желаемого направления обучения, мы предоставляем краткое описание, информацию о кураторе и рекомендуемый возраст);

●Информация о дисциплинах, проводимых в составе направления (для полного отображения узкой направленности выбранной дисциплины, мы предоставляем такие сведения как описание, используемое оборудование, а также, для удобства выбора, указываем время и дни занятий);

●Информация о нагрузке ученика (совокупность необходимых, уже использованных данных, для более удобного отображения нагрузки ученика по каждому направлению и дисциплине с указанным количеством часов и дней);

●Информация о проведении внеурочных мероприятий различного масштаба и характера (Мы заранее предоставляем данную информацию нашим ученикам, чтобы они могли позвать родителей или согласовать с ними своё посещение).

Наша учебная организация также имеет онлайн портал на котором публикуется различная информация о мероприятиях, направлениях подготовки и многом другом. Любой может создать собственный аккаунт на нашем портале и ознакомиться с этой информацией, однако для наших учеников и преподавателей мы предоставляем возможность указания своего персонального идентификатора при регистрации, чтобы получить доступ к информации о предстоящих датах занятий, результатах обучения и многом другом (создание аккаунта является необязательным, а также на каждый персональный идентификатор может быть зарегистрирован только один аккаунт).

Ход выполнения нормализации

После выделения конкретных сущностей и определения функционала выделенных сущностей, был сформирован список атрибутов каждой сущности:

STUDENTS

Каждый ученик имеет ФИО, дату рождения, возраст, пол, документ удостоверяющий личность, серию/номер документа. Первичным ключом данной сущности является атрибут id_students. Все атрибуты данной сущности, кроме возраста и пола являются обязательными к заполнению.

PARENT

Каждый родитель имеет ФИО, телефон, почту, идентификатор ученика (внешний ключ). Первичным ключом данной сущности является атрибут fio. Все атрибуты кроме почты являются обязательными к заполнению.

EMPLOYEES

Каждый преподаватель имеет ФИО, дату рождения, возраст, пол, документ удостоверяющий личность, серию/номер документа, должность, почту, опыт работы, персональные достижения. Первичным ключом данной сущности является атрибут id_employees. Все атрибуты данной сущности, кроме возраста, пола, опыта работы, персональных достижений являются обязательными к заполнению.

EVENTS

Каждое внеаудиторное мероприятие имеет название, место проведения, дату и время проведения, организующего/ответственного преподавателя (внешний ключ), описание. Первичным ключом данной сущности является атрибут id_events. Все атрибуты кроме описания являются обязательными к заполнению.

DEPARTMENTS

Каждое направление имеет название, главу направления, возрастные рекомендации и описание. Первичным ключом данной сущности является атрибут id_departments. Все атрибуты являются обязательными к заполнению.

CLASSES

Каждая дисциплина имеет название, идентификатор направления (внешний ключ), идентификатор преподавателя (внешний ключ), кабинет, дни занятий, время занятий, необходимое оборудование, рекомендуемый возраст, описание. Первичным ключом данной сущности является атрибут id_classes. Все атрибуты кроме оборудования и рекомендованного возраста являются обязательными к заполнению.

WORKLOAD

Учебная нагрузка имеет идентификатор ученика (внешний ключ), идентификатор направления (внешний ключ), идентификатор дисциплины (внешний ключ), идентификатор преподавателя (внешний ключ), выбранные дни занятий, комментарий преподавателя. Первичным ключом данной сущности является атрибут id_workload. Все атрибуты кроме комментария преподавателя являются обязательными к заполнению.

USERS

Каждый пользователь имеет логин, пароль, идентификатор ученика (внешний ключ), идентификатор преподавателя (внешний ключ). Первичным ключом данной сущности является атрибут Login. Атрибуты логин и пароль являются обязательными к заполнению. Один из атрибутов идентификаторов заполняется пользователем в случае регистрации, как ученик или преподаватель.

Особенности некоторых сущностей

Для сущностей: students, employees, departments, classes - мы задали уникальное именование атрибутов id, для более удобного ориентирования и упрощения ведения таблицы загрузки учеников (workload).

Объяснение идентификаторов:

Знак '_' - означает наличие некоторый цифры на его месте.

- Сущность students: идентификатор id_students имеет вид: "S___"
- Сущность employees: идентификатор id_employees имеет вид: "E___"
- Сущность departments: идентификатор id_departments имеет вид: "D__"
- Сущность classes: идентификатор id_classes имеет вид: "C__"

Задание id с параметром AUTO_INCREMENT существенно осложняло работу с созданной базой данных, т.к. тогда при заполнении информации в зависимых сущностях (особенно сущности Workload) могла возникнуть проблема для идентификации сущности, из которой взято соответствующее значение.

Также для сущностей students и employees необходимо использование уникальных идентификаторов, без AUTO_INCREMENT в связи с их использованием при регистрации на онлайн ресурсе. Т.к. связь students,employees -> users является связью 1:1, то при появлении одинаковых идентификаторов в students и employees возникнет ошибка при их использовании в users (т.к. на один уникальный идентификатор может быть зарегистрирован только один аккаунт).

Объяснение выбранной СУБД

В данных условиях мы используем реляционную модель базы данных по причине необходимости взаимосвязи определённых сущностей с другими сущностями путём создания главных и внешних ключей на основе атрибутов сущностей. Для сохранения целостности базы данных необходимо наличие связей между сущностями (так как записи в определённых сущностях напрямую зависят от записей в других сущностях).

Разработанная база данных имеет 8 сущностей, каждая из которых содержит в себе некоторое количество атрибутов, т.к. реляционная модель позволяет это сделать. Однако если бы мы выбрали нереляционную модель, то (т.к. отсутствует привычная для реляционных баз данных схема из строк и столбцов, а используется схема состоящая из пар ключ:значение) количество сущностей бы значительно возрасло.

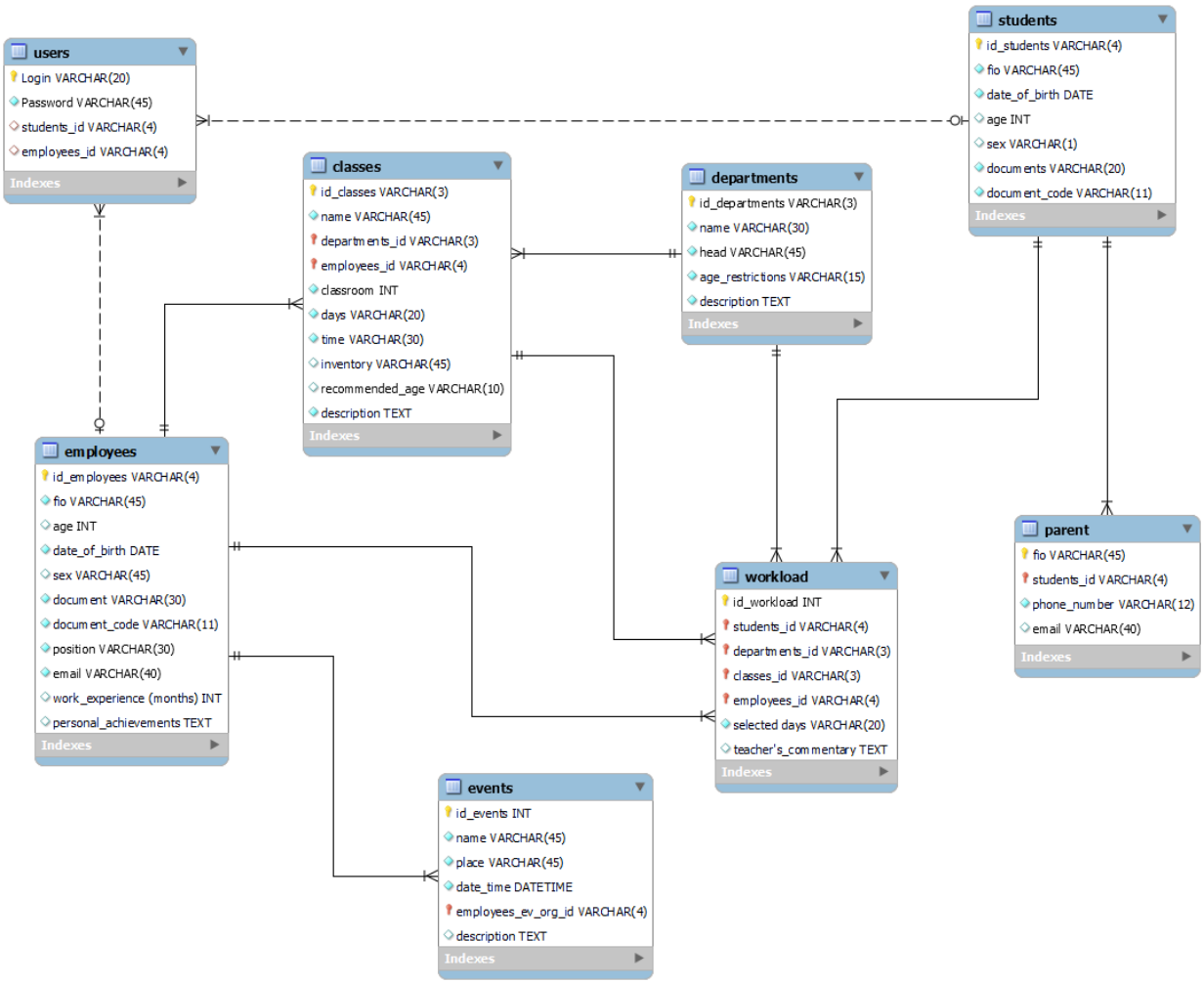
Нереляционная модель удобна для хранения больших объёмов однотипной информации, однако в нашем случае информация является разноплановой, в связи с чем необходимо иметь гибкую структуру, простоту извлечения/изменения даннных, а также иметь возможность простого обращения к этим данным. Именно поэтому для разработки базы данных использовалась реляционная модель.

База данных была разработана нами самостоятельно, в связи, с чем использовалась изученная нами СУБД: "MySQL".

Причины выбора данной СУБД:

- Низкий порог входа для начала разработки;
- Доступный функционал;
- Возможность создания и представления как графическим методом, так и методом команд;
- Возможность работы с БД с помощью запросов на языке SQL.

ER – диаграмма



Объяснение связей между сущностями

Все связи между сущностями в разработанной базе данных являются идентифицирующими, кроме связей с сущностью users. Аргументацией такого выбора является зависимость одних сущностей от других и не возможность существования записей в зависимых сущностях без указания конкретных записей из главной(ых) сущности(ей), кроме сущности users.

Аргументация для каждой связи:

- students → parent: В таблице parent не может существовать записи без указания конкретного ученика;

- employees → events: Не может существовать запись о некотором событии, без указания ответственного преподавателя (наблюдающего);

- departments, employees → classes: Запись про каждую дисциплину должна содержать информацию о том, к какому направению она относится и какой преподаватель её ведёт (дисциплина не может быть добавлена без указания этих данных);

- students, employees, departments, classes → workload: Каждая запись содержит в себе идентификаторы перечисленных сущностей, так как каждая запись в сущности workload не может быть добавлена без студента и без дисциплины, которая в свою очередь не может существовать без преподавателя и направления;

- students, employees → users: т.к. аккаунт может создать любой пользователь портала, то запись в users может существовать без указания некоторой записи из сущностей students и employees, а значит быть независимой, поэтому связь является неидентифицирующей.

Мощности связей:

- students → parent: Для каждого ученика может быть указано нексколько родителей/опекунов (т.к. данная информация необходима для связи в случае возникновения непредвиденных обстоятельств, могут быть указаны также бабушки/дедушки), поэтому связь 1:n;

- employees → events: Каждый преподаватель может являться организатором для нескольких мероприятий (и единственным), поэтому связь 1:n;

- departments → classes: Каждое направление может содержать несколько дисциплин, но дисциплина не может относиться к нескольким направлениям, поэтому связь 1:n;

- employees → classes: Каждую дисциплину ведёт только один определённый преподаватель и каждый преподаватель может вести несколько дисциплин, поэтому связь 1:n;

- students → workload: Каждый ученик может заниматься несколькими дисциплинами, а значит несколько раз быть указаным в таблице с нагрузкой, поэтому связь 1:n;

- employees → workload: Преподаватель ведёт у групп учеников, поэтому может встречаться в таблице с нагрузкой несколько раз, поэтому связь 1:n;

- departments → workload: Каждое направление может встречаться в таблице нагрузки учеников несколько раз (относится к ученикам), поэтому связь 1:n;

- classes → workload: Каждая дисциплина может встречаться в таблице нагрузки учеников несколько раз (разные ученики могли выбрать одну дисциплину), поэтому связь 1:n;

- students → users: На каждый персональный идентификатор ученика может быть зарегистрирован ровно один аккаунт, поэтому связь 1:1;

- employees → users: На каждый персональный идентификатор преподавателя может быть зарегистрирован ровно один аккаунт, поэтому связь 1:1.

Исходный текст запросов

По созданию таблиц

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,N
O_ENGINE_SUBSTITUTION';

-- -----------------------------------------------------
-- Schema project
-- -----------------------------------------------------

-- -----------------------------------------------------
-- Schema project
-- -----------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `project` DEFAULT CHARACTER SET utf8 ;
USE `project` ;

-- -----------------------------------------------------
-- Table `project`.`students`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `project`.`students` (
  `id_students` VARCHAR(4) NOT NULL,
  `fio` VARCHAR(45) NOT NULL,
  `date_of_birth` DATE NOT NULL CHECK (YEAR(`date_of_birth`) <= 2017),
  `age` INT NULL,
  `sex` VARCHAR(1) NULL,
  `documents` VARCHAR(20) NOT NULL,
  `document_code` VARCHAR(11) NOT NULL,
  PRIMARY KEY (`id_students`),
  UNIQUE INDEX `idstudents_UNIQUE` (`id_students` ASC) VISIBLE,
  UNIQUE INDEX `fio_UNIQUE` (`fio` ASC) VISIBLE,
  UNIQUE INDEX `document_code_UNIQUE` (`document_code` ASC) VISIBLE)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `project`.`parent`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `project`.`parent` (
  `fio` VARCHAR(45) NOT NULL,
  `students_id` VARCHAR(4) NOT NULL,
  `phone_number` VARCHAR(12) NOT NULL,
  `email` VARCHAR(40) NULL,
  PRIMARY KEY (`fio`, `students_id`),
  INDEX `parent_to_student_idx` (`students_id` ASC) VISIBLE,
  CONSTRAINT `parent_to_student`
    FOREIGN KEY (`students_id`)
    REFERENCES `project`.`students` (`id_students`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `project`.`employees`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `project`.`employees` (
  `id_employees` VARCHAR(4) NOT NULL,
  `fio` VARCHAR(45) NOT NULL,
  `age` INT NULL,
  `date_of_birth` DATE NOT NULL,
  `sex` VARCHAR(45) NULL,
  `document` VARCHAR(30) NOT NULL,
  `document_code` VARCHAR(11) NOT NULL,
  `position` VARCHAR(30) NOT NULL,
  `email` VARCHAR(40) NOT NULL,
  `work_experience (months)` INT NULL,
```

```
  `personal_achievements` TEXT NULL,
  PRIMARY KEY (`id_employees`),
  UNIQUE INDEX `id_employees_UNIQUE` (`id_employees` ASC) VISIBLE,
  UNIQUE INDEX `fio_UNIQUE` (`fio` ASC) VISIBLE,
  UNIQUE INDEX `document_code_UNIQUE` (`document_code` ASC) VISIBLE)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `project`.`departments`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `project`.`departments` (
  `id_departments` VARCHAR(3) NOT NULL,
  `name` VARCHAR(30) NOT NULL,
  `head` VARCHAR(45) NOT NULL,
  `age_restrictions` VARCHAR(15) NOT NULL,
  `description` TEXT NOT NULL,
  PRIMARY KEY (`id_departments`),
  UNIQUE INDEX `id_departments_UNIQUE` (`id_departments` ASC) VISIBLE,
  UNIQUE INDEX `name_UNIQUE` (`name` ASC) VISIBLE)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `project`.`classes`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `project`.`classes` (
  `id_classes` VARCHAR(3) NOT NULL,
  `name` VARCHAR(45) NOT NULL,
  `departments_id` VARCHAR(3) NOT NULL,
  `employees_id` VARCHAR(4) NOT NULL,
  `classroom` INT NOT NULL,
  `days` VARCHAR(20) NOT NULL,
  `time` VARCHAR(30) NOT NULL,
  `inventory` VARCHAR(45) NULL,
  `recommended_age` VARCHAR(10) NULL,
  `description` TEXT NOT NULL,
  PRIMARY KEY (`id_classes`, `employees_id`, `departments_id`),
  UNIQUE INDEX `id_classes_UNIQUE` (`id_classes` ASC) VISIBLE,
  INDEX `classes_to_dep_idx` (`departments_id` ASC) VISIBLE,
  INDEX `classes_to_emp_idx` (`employees_id` ASC) VISIBLE,
  UNIQUE INDEX `name_UNIQUE` (`name` ASC) VISIBLE,
  CONSTRAINT `classes_to_dep`
    FOREIGN KEY (`departments_id`)
    REFERENCES `project`.`departments` (`id_departments`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `classes_to_emp`
    FOREIGN KEY (`employees_id`)
    REFERENCES `project`.`employees` (`id_employees`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `project`.`workload`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `project`.`workload` (
  `id_workload` INT NOT NULL AUTO_INCREMENT,
  `students_id` VARCHAR(4) NOT NULL,
  `departments_id` VARCHAR(3) NOT NULL,
  `classes_id` VARCHAR(3) NOT NULL,
  `employees_id` VARCHAR(4) NOT NULL,
  `selected days` VARCHAR(20) NOT NULL,
  `teacher's_commentary` TEXT NULL,
  PRIMARY KEY (`id_workload`, `students_id`, `departments_id`, `classes_id`, `employees_id`),
  UNIQUE INDEX `id_student's_plan_UNIQUE` (`id_workload` ASC) VISIBLE,
  INDEX `st_p_to_student_idx` (`students_id` ASC) VISIBLE,
  INDEX `st_p_to_department_idx` (`departments_id` ASC) VISIBLE,
  INDEX `st_p_to_classes_idx` (`classes_id` ASC) VISIBLE,
  INDEX `st_p_to_employee_idx` (`employees_id` ASC) VISIBLE,
  CONSTRAINT `st_p_to_student`
    FOREIGN KEY (`students_id`)
    REFERENCES `project`.`students` (`id_students`)
    ON DELETE CASCADE
```

```
    ON UPDATE CASCADE,
  CONSTRAINT `st_p_to_department`
    FOREIGN KEY (`departments_id`)
    REFERENCES `project`.`departments` (`id_departments`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `st_p_to_classes`
    FOREIGN KEY (`classes_id`)
    REFERENCES `project`.`classes` (`id_classes`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `st_p_to_employee`
    FOREIGN KEY (`employees_id`)
    REFERENCES `project`.`employees` (`id_employees`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `project`.`events`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `project`.`events` (
  `id_events` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `place` VARCHAR(45) NOT NULL,
  `date_time` DATETIME NOT NULL,
  `employees_ev_org_id` VARCHAR(4) NOT NULL,
  `description` TEXT NULL,
  PRIMARY KEY (`id_events`, `employees_ev_org_id`),
  UNIQUE INDEX `idevents_UNIQUE` (`id_events` ASC) VISIBLE,
  INDEX `event_to_employee_idx` (`employees_ev_org_id` ASC) VISIBLE,
  CONSTRAINT `event_to_employee`
    FOREIGN KEY (`employees_ev_org_id`)
    REFERENCES `project`.`employees` (`id_employees`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
-- -----------------------------------------------------
-- Table `project`.`users`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `project`.`users` (
  `Login` VARCHAR(20) NOT NULL,
  `Password` VARCHAR(45) NOT NULL CHECK (`Password` LIKE '________%'),
  `students_id` VARCHAR(4) NULL,
  `employees_id` VARCHAR(4) NULL,
  PRIMARY KEY (`Login`),
  UNIQUE INDEX `Login_UNIQUE` (`Login` ASC) VISIBLE,
  INDEX `users_to_employees_idx` (`employees_id` ASC) VISIBLE,
  UNIQUE INDEX `students_id_UNIQUE` (`students_id` ASC) VISIBLE,
  UNIQUE INDEX `employees_id_UNIQUE` (`employees_id` ASC) VISIBLE,
  CONSTRAINT `users_to_students`
    FOREIGN KEY (`students_id`)
    REFERENCES `project`.`students` (`id_students`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `users_to_employees`
    FOREIGN KEY (`employees_id`)
    REFERENCES `project`.`employees` (`id_employees`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

По наполнению таблиц

```
INSERT INTO `project`.`students`
(`id_students`,`fio`,`date_of_birth`,`age`,`sex`,`documents`,`document_code`) VALUES
('S001','Kargapolov Denis Andreevich','2005-04-07',18,'m','passport','4617539198'),
('S002','Stecuk Maxim Nikolaevich','2006-03-28',17,'m','passport','4617539199'),
('S003','Zuhir Amira Saidovna','2008-04-18',16,'f','passport','4617148828'),
('S004','Kruchkova Anastasia Sergeevna','2010-01-01',13,'f','svidetelstvo','1337666444'),
('S005','Volozhanin Vladik Kirillovich','2006-08-08',17,'m','passport','4618687310'),
('S006','Lotuga Danila Dapkunatevich','2008-05-13',16,'m','passport','1842003333'),
('S007','Klementiev Lesha Ibragimovich','2008-05-14',16,'m','passport','6667778880'),
('S008','Maximova Angelina Vyacheslavovna','2013-06-06',10,'f','svidetelstvo','8800228777'),
('S009','Sobchak Ksenia Anatolyevna','2013-07-08',10,'f','svidetelstvo','1234567890'),
('S010','Gorshok Grisha Grigoryev','2005-03-06',18,'m','passport','4718678123'),
('S011','Adushkina Katya Marzhelovna','2008-09-09',16,'f','passport','4819432030'),
('S012','Arogundade Zak Franclin','2010-09-04',13,'m','svidetelstvo','9757589056'),
('S013','Reichwald Benjamin Igrevich ','2009-08-03',14,'m','passport','8631598709'),
('S014','Tumor Yves Anatolievich','2006-10-10',17,'m','passport','1352867089'),
('S015','Miros Vlad Nikolaevich','2007-05-19',16,'m','passport','8213967513'),
('S016','Rudkovskiy Gnom Gnomich','2013-08-06',10,'m','svidetelstvo','3128891221'),
('S017','Pupkin Petr Vovochkin','2006-11-19',17,'m','passport','8689120767'),
('S018','Blinovskaya Elena Viktorovna','2008-06-15',16,'f','passport','6231877812'),
('S019','Dubrovina Arina Romanovna','2009-07-03',14,'f','passport','6132967126'),
('S020','Dubok Vasily Nikolaevich','2010-03-06',13,'m','svidetelstvo','8213697679');

INSERT INTO `project`.`parent` (`fio`, `students_id`, `phone_number`, `email`) VALUES
('Stecuk Nikolay Kirillovuch', 'S002', '+78908097878', 'sergay@mail.ru'),
('Zuhir Said Saidovich', 'S003', '+78320751673', 'saidsaid@gmail.com'),
('Kruchkov Sergey Anatolievich', 'S004', '+76713248983', 'kruchkov@mail.ru'),
('Volozhanin Kiriill Vyacheslavovich', 'S005', '+73284727362', 'volozh@mail.ru'),
('Lotuga Dapkunayte lvovich', 'S006', '+72347286823', 'dabdab@gmail.com'),
('Klementyev Vyacheslav Igorevich', 'S007', '+74732734679', 'klimsanich@mail.ru'),
('Maximov Maxim Maximich', 'S008', '+77342622839', 'slaymyangel@mail.ru'),
('Sobchak Anatoly Andreevich', 'S009', '+72734629342', 'sobsob@mail.ru'),
('Gorshok Grigory Grigoryevich', 'S010', '+78324282389', 'pankihoy@mail.ru'),
('Adushkin Marzhela Nikolaevich', 'S011', '+73242868688', 'ayavdush@mail.ru'),
('Arogundade Franclin Ferdinand', 'S012', '+73847279469', 'notmymail@mail.ru'),
('Reichwal Igor Igorevich', 'S013', '+72364729879', 'reichreich@mail.ru'),
('Tumor Anoly Andreevich', 'S014', '+77126319796', 'tumooor@gmail.com'),
('Miros Vladlena Andreevna', 'S015', '+78347237696', 'ohimtired@mail.ru'),
('Rudkovskaya Yana Igorevna', 'S016', '+76429469293', 'yayanarudkovskaya@mail.ru'),
('Pupkina Marina Andreevna', 'S017', '+77237187679', 'pupkina1939@mail.ru'),
('Blinovskia Klim Sanich', 'S018', '+72376723480', 'blinblin@mail.ru'),
('Dubrovin Roman Antonovich', 'S019', '+76342274920', 'dubdub@mail.ru'),
('Dubok Nikolay Nikolayevich', 'S020', '+71732108798', 'dubokvdube@mail.ru');

INSERT INTO `project`.`employees` (`id_employees`, `fio`, `age`, `date_of_birth`, `sex`, `document`,
`document_code`, `position`, `email`, `work_experience (months)`) VALUES
('E001', 'Zhukov Nikolay Nikolaevich', '30', '1993-04-23', 'm', 'passport', '6847534578', 'teacher',
'VLIUZNN@mail.ru', '36'),
('E002', 'Margancova Katerina Ivanova', '20', '2003-07-15', 'f', 'passport', '6213261873', 'teacher',
'hikate@mail.ru', '10'),
('E003', 'Svetlov Igor Vladimirovich', '30', '1993-08-16', 'm', 'passport', '8327261769', 'teacher',
'svetlov@mail.ru', '43'),
('E004', 'Vlasov Dmitry Antonovich', '35', '1988-09-03', 'm', 'passport', '8213797989', 'teacher',
'vlasovzz@mail.ru', '53'),
('E005', 'Awakura Lain Li', '28', '1995-07-07', 'f', 'passport', '8923172178', 'teacher',
'lastprotocol@mail.ru', '27');

INSERT INTO `project`.`events` (`name`, `place`, `date_time`, `employees_ev_org_id`, `description`) VALUES
('school jubiley', 'school hall', '2023-05-05 13:00:00', 'E002', 'Hooray nashey school celih nol let!!!'),
('day of kofeman', 'school bufet', '2023-06-08 17:00:00', 'E003', 'COFFEE!!!');

INSERT INTO `project`.`departments` (`id_departments`, `name`, `head`, `age_restrictions`, `description`)
VALUES
('D01', 'Maths', 'Petr Petrov', '7-18', 'Solving different tasks'),
('D02', 'Programming', 'Lain Awakura', '10-18', 'web development processes'),
('D03', 'Design', 'Katerina Margancova', '10-18', 'design practice and theory'),
('D04', 'Robotics', 'Igor Svetlov', '13-18', 'all styles of making & coding robots');
```

```
INSERT INTO `project`.`classes` (`id_classes`,`name`, `departments_id`, `employees_id`, `classroom`,
`days`, `time`, `description`) VALUES
('C01', 'OlympMaths', 'D01', 'E001', '237', 'mon thy sat', '18:30 - 20:00', 'Very Intersting'),
('C02', 'C++ coding', 'D02', 'E004', '217', 'mon tue wed', '18:30 - 20:00', 'coding!!!'),
('C03', 'Logical operations', 'D02', 'E005', '218', 'thu fri sat', '16:00 - 18:00', 'logica'),
('C04', 'History Design', 'D03', 'E002', '307', 'fri sat', '15:30 - 17:00', 'Very Intersting!'),
('C05', 'Controllers', 'D04', 'E003', '308', 'mon wen fri', '17:30 - 19:30', 'Prog and Create controllers
for robots;)'),
('C06', 'SchoolMaths', 'D01', 'E001', '237', 'mon thy sat', '17:00 - 18:30', 'Extra School Lessons');

INSERT INTO `project`.`workload` (`students_id`, `departments_id`, `classes_id`, `employees_id`, `selected
days`,`teacher's_commentary`) VALUES
('S001', 'D02', 'C03', 'E005', 'thu fri', NULL),
('S002', 'D01', 'C01', 'E001', 'mon sat', NULL),
('S010', 'D03', 'C04', 'E002', 'sat', NULL),
('S002', 'D04', 'C05', 'E003', 'mon wen', NULL),
('S004', 'D04', 'C05', 'E003', 'mon', NULL),
('S005', 'D01', 'C01', 'E001', 'mon thu sat', NULL),
('S005', 'D01', 'C06', 'E001', 'mon thu sat', NULL),
('S006', 'D02', 'C02', 'E004', 'mon tue wed', NULL),
('S006', 'D02', 'C03', 'E005', 'thu fri sat', NULL),
('S007', 'D03', 'C04', 'E002', 'fri sat', 'do not seat with other students'),
('S008', 'D04', 'C05', 'E003', 'mon wen fri', NULL),
('S009', 'D01', 'C01', 'E001', 'mon sat', NULL),
('S009', 'D02', 'C02', 'E004', 'mon', NULL),
('S011', 'D03', 'C04', 'E002', 'sat', NULL),
('S012', 'D04', 'C05', 'E003', 'mon wen', NULL),
('S013', 'D02', 'C02', 'E004', 'mon tue', 'do not make effort with student'),
('S014', 'D01', 'C01', 'E001', 'mon', NULL),
('S014', 'D01', 'C06', 'E001', 'mon', NULL),
('S015', 'D03', 'C04', 'E002', 'sat', 'not payed enough'),
('S016', 'D01', 'C01', 'E001', 'sat', NULL),
('S016', 'D01', 'C06', 'E001', 'sat', NULL),
('S017', 'D02', 'C02', 'E004', 'mon tue', NULL),
('S018', 'D04', 'C05', 'E003', 'wen', NULL),
('S019', 'D03', 'C04', 'E002', 'fri sat', 'seat closer to teacher'),
('S020', 'D03', 'C04', 'E002', 'fri ', NULL);

INSERT INTO `project`.`users` (`Login`, `Password`, `students_id`, `employees_id`) VALUES
('tulenchik666', 'kukarekyyyyy', 'S002', Null),
('amiranagibator228', 'zaychik1999', 'S003', Null),
('nastyamirnaya228', 'bitcheslocal1333', 'S004', Null),
('vladospapiros777', 'ogurec2003', 'S005', Null),
('dant4ick2203', 'yalubluselenygomez', 'S006', Null),
('leshaXgod', 'secretsecret3', 'S007', Null),
('maximova2007', 'optimusprime666', 'S008', Null),
('SobchakSobchak', 'ahahahahahahahah', 'S009', Null),
('gorshochekschemto', 'witchersdoll', 'S010', Null),
('psihushkaa4', 'lovenhante333', 'S011', Null),
('aloecoder', 'gtbsg0000', 'S012', Null),
('benjaminSek', 'aoaoaoaoaoaooaoa', 'S013', Null),
('tumortumor', 'inspiteofwar2007', 'S014', Null),
('vladosmiros', 'nuaktogood', 'S015', Null),
('rudik777', 'gnomiiiiik', 'S016', Null),
('zhukovN', 'zhuknevivozhuk', Null, 'E001'),
('margancova2008', 'ihkatyakatya', Null, 'E002'),
('svetlovv', 'parol228775', Null, 'E003'),
('linuxlover1993', '44444444', Null, 'E004'),
('lainlainlain', 'lastprotocol', Null, 'E005');
```