

**Курсовой проект по курсу
«Основы реляционных баз данных. MySQL»**

Выполнил

Кузнецов В.А.

2021

ОГЛАВЛЕНИЕ

1 ВВЕДЕНИЕ	3
2 ОПИСАНИЕ	3
3 СОЗДАНИЕ БД И ТАБЛИЦ	4
3.1 Dump БД.....	4
3.2 Наполнение таблиц данными	10
3.2.1 Наполнение таблиц с помощью сервиса filldb на примере таблицы country.....	16
3.2.2 Наполнение таблиц на примере таблицы programms.....	16
4 ХАРАКТЕРНЫЕ ВЫБОРКИ	17
4.1 Запрос на подсчет количества пользователей из разных стран	17
4.2 Запрос на подсчет списка пользователей, купивших программы	18
5 ПРЕДСТАВЛЕНИЯ	19
6 ХРАНИМЫЕ ПРОЦЕДУРЫ И ТРИГГЕРЫ	21

1 ВВЕДЕНИЕ

Пояснительная записка составлена по результатам разработки базы данных сайта Movavi (<https://www.movavi.ru>).

Минимальные требования к курсовому проекту:

1. Общее текстовое описание БД и решаемых ею задач;
2. Минимальное количество таблиц – более 10 (с первичными ключами, индексами, внешними ключами);
3. Скрипты наполнения БД данными;
4. Скрипты характерных выборок (включающие группировки, JOIN'ы, вложенные таблицы);
5. Представления (минимум 2);
6. Хранимые процедуры и триггеры;
7. ER-диаграмма.

2 ОПИСАНИЕ

База данных сайта Movavi необходима для структурирования предоставляемых фирмой продуктов своим клиентам. Все продукты MES структурированы по типам, а те в свою очередь по темам (при необходимости). Помимо продуктов БД хранит информацию о пользователях, подписках и предоставляемых скидках.

3 СОЗДАНИЕ БД И ТАБЛИЦ

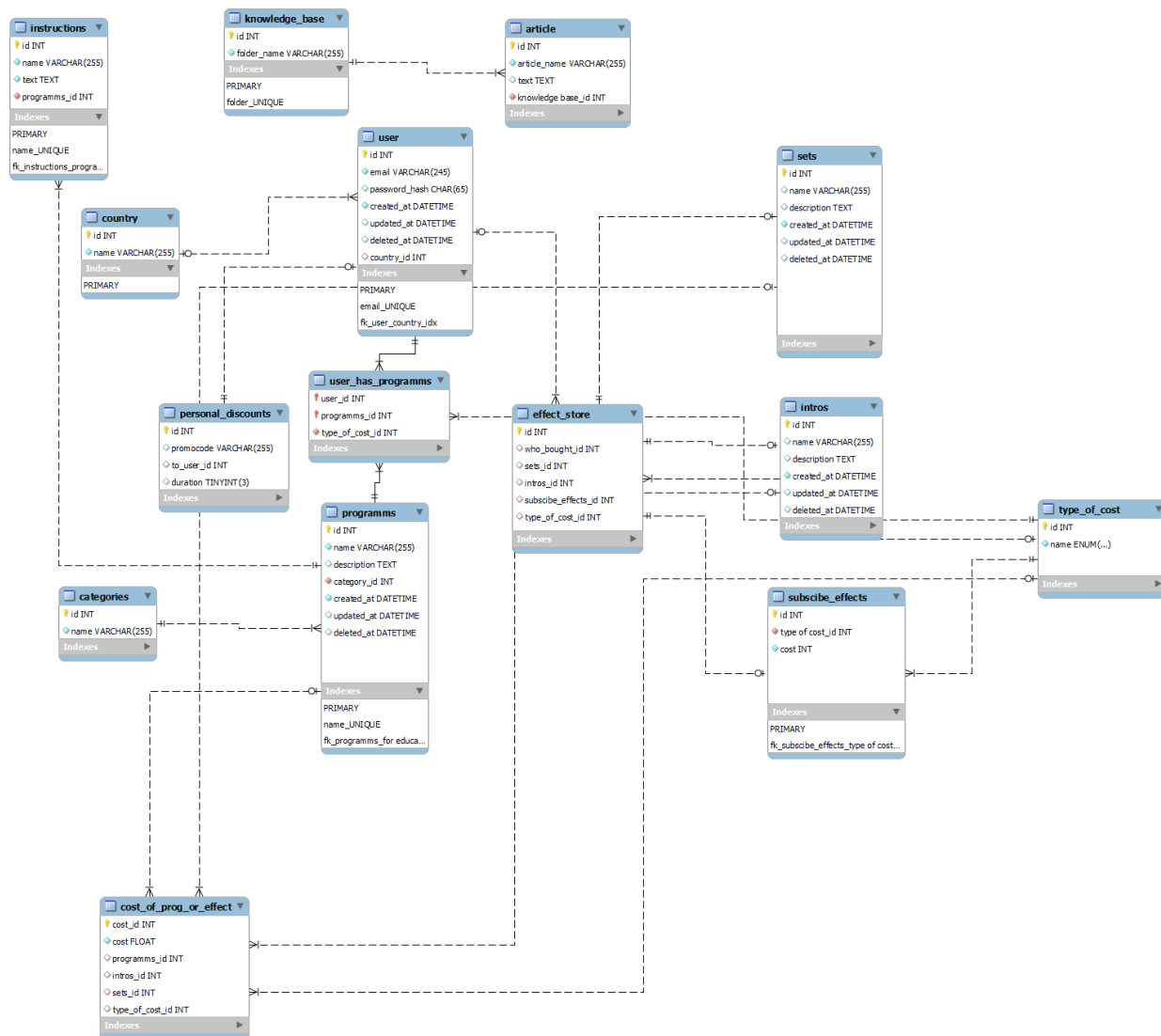


Рисунок 1 – EER диаграмма создания таблиц

3.1 Dump БД

-- MySQL Script generated by MySQL Workbench
 -- Sat Aug 21 18:45:34 2021
 -- Model: New Model Version: 1.0
 -- 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,NO_ENGINE_SUBSTITUTION';
```

-- Schema movavi_course

-- Schema movavi_course

CREATE SCHEMA IF NOT EXISTS `movavi_course` DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_0900_ai_ci ;
USE `movavi_course` ;

-- Table `movavi_course`.`country`

CREATE TABLE IF NOT EXISTS `movavi_course`.`country` (
 `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
 `name` VARCHAR(255) NOT NULL DEFAULT 'Россия',
 PRIMARY KEY (`id`))
ENGINE = InnoDB;

-- Table `movavi_course`.`user`

CREATE TABLE IF NOT EXISTS `movavi_course`.`user` (
 `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
 `email` VARCHAR(245) NOT NULL,
 `password_hash` CHAR(65) NULL DEFAULT NULL,
 `created_at` DATETIME NOT NULL DEFAULT NOW(),
 `updated_at` DATETIME NULL DEFAULT NULL ON UPDATE NOW(),
 `deleted_at` DATETIME NULL DEFAULT NULL,
 `country_id` INT UNSIGNED NULL,
 PRIMARY KEY (`id`),
 UNIQUE INDEX `email_UNIQUE` (`email` ASC) VISIBLE,
 INDEX `fk_user_country_idx` (`country_id` ASC) VISIBLE,
 CONSTRAINT `fk_user_country`
 FOREIGN KEY (`country_id`)
 REFERENCES `movavi_course`.`country` (`id`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- Table `movavi_course`.`personal_discounts`

CREATE TABLE IF NOT EXISTS `movavi_course`.`personal_discounts` (
 `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
 `promocode` VARCHAR(255) NULL,
 `to_user_id` INT UNSIGNED NULL COMMENT 'Кому предоставлена скидка',
 `duration` TINYINT(3) NULL DEFAULT 30 COMMENT 'Продолжительность акции в днях',
 PRIMARY KEY (`id`),
 INDEX `fk_discounts_user1_idx` (`to_user_id` ASC) VISIBLE,
 CONSTRAINT `fk_discounts_user1`
 FOREIGN KEY (`to_user_id`)
 REFERENCES `movavi_course`.`user` (`id`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- Table `movavi_course`.`categories`

```
CREATE TABLE IF NOT EXISTS `movavi_course`.`categories` (  
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `name` VARCHAR(255) NOT NULL COMMENT 'Категории программ',  
  PRIMARY KEY (`id`),  
  UNIQUE INDEX `name_UNIQUE` (`name` ASC) VISIBLE)  
ENGINE = InnoDB;
```

-- Table `movavi_course`.`programms`

```
CREATE TABLE IF NOT EXISTS `movavi_course`.`programms` (  
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `name` VARCHAR(255) NOT NULL,  
  `description` TEXT NULL,  
  `category_id` INT UNSIGNED NOT NULL,  
  `created_at` DATETIME NOT NULL DEFAULT NOW(),  
  `updated_at` DATETIME NULL DEFAULT NULL ON UPDATE NOW(),  
  `deleted_at` DATETIME NULL DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE INDEX `name_UNIQUE` (`name` ASC) VISIBLE,  
  INDEX `fk_programms_for education1_idx` (`category_id` ASC) VISIBLE,  
  CONSTRAINT `fk_programms_for education1`  
    FOREIGN KEY (`category_id`)  
      REFERENCES `movavi_course`.`categories` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `movavi_course`.`intros`

```
CREATE TABLE IF NOT EXISTS `movavi_course`.`intros` (  
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `name` VARCHAR(255) NULL,  
  `description` TEXT NULL,  
  `created_at` DATETIME NOT NULL DEFAULT NOW(),  
  `updated_at` DATETIME NULL DEFAULT NULL ON UPDATE NOW(),  
  `deleted_at` DATETIME NULL DEFAULT NULL,  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

-- Table `movavi_course`.`sets`

```
CREATE TABLE IF NOT EXISTS `movavi_course`.`sets` (  
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
```

```

`name` VARCHAR(255) NULL,
`description` TEXT NULL,
`created_at` DATETIME NOT NULL DEFAULT NOW(),
`updated_at` DATETIME NULL DEFAULT NULL ON UPDATE NOW(),
`deleted_at` DATETIME NULL DEFAULT NULL,
PRIMARY KEY (`id`))
ENGINE = InnoDB;

```

```

-----
-- Table `movavi_course`.`type_of_cost`
-----

```

```

CREATE TABLE IF NOT EXISTS `movavi_course`.`type_of_cost` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` ENUM('personal', 'business') NOT NULL DEFAULT 'personal',
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

```

```

-----
-- Table `movavi_course`.`subscribe_effects`
-----

```

```

CREATE TABLE IF NOT EXISTS `movavi_course`.`subscribe_effects` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `type of cost_id` INT UNSIGNED NOT NULL,
  `cost` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_subscribe_effects_type of cost1_idx` (`type of cost_id` ASC) VISIBLE,
  CONSTRAINT `fk_subscribe_effects_type of cost1`
    FOREIGN KEY (`type of cost_id`)
      REFERENCES `movavi_course`.`type_of_cost` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `movavi_course`.`effect_store`
-----

```

```

CREATE TABLE IF NOT EXISTS `movavi_course`.`effect_store` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `who_bought_id` INT UNSIGNED NULL COMMENT 'Кто купил набор эффектов\n',
  `sets_id` INT UNSIGNED NULL,
  `intros_id` INT UNSIGNED NULL,
  `subscribe_effects_id` INT UNSIGNED NULL,
  `type_of_cost_id` INT UNSIGNED NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_effect_store_user1_idx` (`who_bought_id` ASC) VISIBLE,
  INDEX `fk_effect_store_sets1_idx` (`sets_id` ASC) VISIBLE,
  INDEX `fk_effect_store_intros1_idx` (`intros_id` ASC) VISIBLE,
  INDEX `fk_effect_store_subscribe_effects1_idx` (`subscribe_effects_id` ASC) VISIBLE,
  INDEX `fk_effect_store_type_of_cost1_idx` (`type_of_cost_id` ASC) VISIBLE,
  CONSTRAINT `fk_effect_store_user1`
    FOREIGN KEY (`who_bought_id`)
      REFERENCES `movavi_course`.`user` (`id`)

```

```

ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_effect_store_sets1`
FOREIGN KEY (`sets_id`)
REFERENCES `movavi_course`.`sets` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_effect_store_intros1`
FOREIGN KEY (`intros_id`)
REFERENCES `movavi_course`.`intros` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_effect_store_subscribe_effects1`
FOREIGN KEY (`subscribe_effects_id`)
REFERENCES `movavi_course`.`subscribe_effects` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_effect_store_type_of_cost1`
FOREIGN KEY (`type_of_cost_id`)
REFERENCES `movavi_course`.`type_of_cost` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `movavi_course`.`instructions`
-----

```

```

CREATE TABLE IF NOT EXISTS `movavi_course`.`instructions` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NOT NULL,
  `text` TEXT NOT NULL,
  `programms_id` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `name_UNIQUE` (`name` ASC) VISIBLE,
  INDEX `fk_instructions_programms1_idx` (`programms_id` ASC) VISIBLE,
  CONSTRAINT `fk_instructions_programms1`
    FOREIGN KEY (`programms_id`)
    REFERENCES `movavi_course`.`programms` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `movavi_course`.`knowledge_base`
-----

```

```

CREATE TABLE IF NOT EXISTS `movavi_course`.`knowledge_base` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `folder_name` VARCHAR(255) NOT NULL COMMENT 'Название разделов в базе знаний',
  PRIMARY KEY (`id`),
  UNIQUE INDEX `folder_UNIQUE` (`folder_name` ASC) VISIBLE)
ENGINE = InnoDB;

```



```

-----
-- Table `movavi_course`.`article`
-----
CREATE TABLE IF NOT EXISTS `movavi_course`.`article` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `article_name` VARCHAR(255) NOT NULL,
  `text` TEXT NULL,
  `knowledge base_id` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `article_name_UNIQUE` (`article_name` ASC) VISIBLE,
  INDEX `fk_article_knowledge base1_idx` (`knowledge base_id` ASC) VISIBLE,
  CONSTRAINT `fk_article_knowledge base1`
    FOREIGN KEY (`knowledge base_id`)
    REFERENCES `movavi_course`.`knowledge_base` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `movavi_course`.`cost_of_prog_or_effect`
-----
CREATE TABLE IF NOT EXISTS `movavi_course`.`cost_of_prog_or_effect` (
  `cost_id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `cost` FLOAT UNSIGNED NOT NULL,
  `programms_id` INT UNSIGNED NULL,
  `intros_id` INT UNSIGNED NULL,
  `sets_id` INT UNSIGNED NULL,
  `type_of_cost_id` INT UNSIGNED NULL,
  PRIMARY KEY (`cost_id`),
  INDEX `fk_cost_of_programm_programms1_idx` (`programms_id` ASC) VISIBLE,
  INDEX `fk_cost_of_programm_type of cost1_idx` (`type_of_cost_id` ASC) VISIBLE,
  INDEX `fk_cost_of_programm_intros1_idx` (`intros_id` ASC) VISIBLE,
  INDEX `fk_cost_of_programm_sets1_idx` (`sets_id` ASC) VISIBLE,
  CONSTRAINT `fk_cost_of_programm_programms1`
    FOREIGN KEY (`programms_id`)
    REFERENCES `movavi_course`.`programms` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_cost_of_programm_type of cost1`
    FOREIGN KEY (`type_of_cost_id`)
    REFERENCES `movavi_course`.`type_of_cost` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_cost_of_programm_intros1`
    FOREIGN KEY (`intros_id`)
    REFERENCES `movavi_course`.`intros` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_cost_of_programm_sets1`
    FOREIGN KEY (`sets_id`)
    REFERENCES `movavi_course`.`sets` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `movavi_course`.`user_has_programms`
-----
CREATE TABLE IF NOT EXISTS `movavi_course`.`user_has_programms` (
  `user_id` INT UNSIGNED NOT NULL,
  `programms_id` INT UNSIGNED NOT NULL,
  `type_of_cost_id` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`user_id`, `programms_id`),
  INDEX `fk_user_has_programms_programms1_idx` (`programms_id` ASC) VISIBLE,
  INDEX `fk_user_has_programms_user1_idx` (`user_id` ASC) VISIBLE,
  INDEX `fk_user_has_programms_type_of_cost1_idx` (`type_of_cost_id` ASC) VISIBLE,
  CONSTRAINT `fk_user_has_programms_user1`
    FOREIGN KEY (`user_id`)
      REFERENCES `movavi_course`.`user` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_user_has_programms_programms1`
    FOREIGN KEY (`programms_id`)
      REFERENCES `movavi_course`.`programms` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_user_has_programms_type_of_cost1`
    FOREIGN KEY (`type_of_cost_id`)
      REFERENCES `movavi_course`.`type_of_cost` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

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

```

3.2 Наполнение таблиц данными

```

-- MySQL Script generated by MySQL Workbench
-- Sun Aug 15 15:07:15 2021
-- Model: New Model   Version: 1.0
-- 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_ZE
RO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

```

```

-----
-- Schema movavi_course
-----

```

```

-----
-- Schema movavi_course
-----

```

```
CREATE SCHEMA IF NOT EXISTS `movavi_course` DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_0900_ai_ci ;
USE `movavi_course` ;
```

```
-----
-- Table `movavi_course`.`country`
-----
```

```
CREATE TABLE IF NOT EXISTS `movavi_course`.`country` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NOT NULL DEFAULT 'Россия',
  PRIMARY KEY (`id`))
ENGINE = InnoDB;
```

```
-----
-- Table `movavi_course`.`user`
-----
```

```
CREATE TABLE IF NOT EXISTS `movavi_course`.`user` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `email` VARCHAR(245) NOT NULL,
  `password_hash` CHAR(65) NULL DEFAULT NULL,
  `created_at` DATETIME NOT NULL DEFAULT NOW(),
  `updated_at` DATETIME NULL DEFAULT NULL ON UPDATE NOW(),
  `deleted_at` DATETIME NULL DEFAULT NULL,
  `country_id` INT UNSIGNED NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `email_UNIQUE` (`email` ASC) VISIBLE,
  INDEX `fk_user_country_idx` (`country_id` ASC) VISIBLE,
  CONSTRAINT `fk_user_country`
    FOREIGN KEY (`country_id`)
      REFERENCES `movavi_course`.`country` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-----
-- Table `movavi_course`.`personal_discounts`
-----
```

```
CREATE TABLE IF NOT EXISTS `movavi_course`.`personal_discounts` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `promocode` VARCHAR(255) NULL,
  `to_user_id` INT UNSIGNED NULL COMMENT 'Кому предоставлена скидка',
  `duration` TINYINT(3) NULL DEFAULT 30 COMMENT 'Продолжительность акции в днях',
  PRIMARY KEY (`id`),
  INDEX `fk_discounts_user1_idx` (`to_user_id` ASC) VISIBLE,
  CONSTRAINT `fk_discounts_user1`
    FOREIGN KEY (`to_user_id`)
      REFERENCES `movavi_course`.`user` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```

-- Table `movavi_course`.`categories`
-----
CREATE TABLE IF NOT EXISTS `movavi_course`.`categories` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NOT NULL COMMENT 'Категории программ',
  PRIMARY KEY (`id`),
  UNIQUE INDEX `name_UNIQUE` (`name` ASC) VISIBLE)
ENGINE = InnoDB;

-- Table `movavi_course`.`programms`
-----
CREATE TABLE IF NOT EXISTS `movavi_course`.`programms` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NOT NULL,
  `description` TEXT NULL,
  `category_id` INT UNSIGNED NOT NULL,
  `created_at` DATETIME NOT NULL DEFAULT NOW(),
  `updated_at` DATETIME NULL DEFAULT NULL ON UPDATE NOW(),
  `deleted_at` DATETIME NULL DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `name_UNIQUE` (`name` ASC) VISIBLE,
  INDEX `fk_programms_for education1_idx` (`category_id` ASC) VISIBLE,
  CONSTRAINT `fk_programms_for education1`
    FOREIGN KEY (`category_id`)
      REFERENCES `movavi_course`.`categories` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- Table `movavi_course`.`type_of_subscribe`
-----
CREATE TABLE IF NOT EXISTS `movavi_course`.`type_of_subscribe` (
  `id` TINYINT(2) UNSIGNED NOT NULL AUTO_INCREMENT,
  `period` ENUM('1 year', 'lifetime') NOT NULL DEFAULT 'lifetime',
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

-- Table `movavi_course`.`intros`
-----
CREATE TABLE IF NOT EXISTS `movavi_course`.`intros` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NULL,
  `description` TEXT NULL,
  `created_at` DATETIME NOT NULL DEFAULT NOW(),
  `updated_at` DATETIME NULL DEFAULT NULL ON UPDATE NOW(),
  `deleted_at` DATETIME NULL DEFAULT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

```

-- Table `movavi_course`.`sets`

```
CREATE TABLE IF NOT EXISTS `movavi_course`.`sets` (  
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `name` VARCHAR(255) NULL,  
  `description` TEXT NULL,  
  `created_at` DATETIME NOT NULL DEFAULT NOW(),  
  `updated_at` DATETIME NULL DEFAULT NULL ON UPDATE NOW(),  
  `deleted_at` DATETIME NULL DEFAULT NULL,  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

-- Table `movavi_course`.`type_of_cost`

```
CREATE TABLE IF NOT EXISTS `movavi_course`.`type_of_cost` (  
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `name` ENUM('personal', 'business') NOT NULL DEFAULT 'personal',  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB;
```

-- Table `movavi_course`.`subscribe_effects`

```
CREATE TABLE IF NOT EXISTS `movavi_course`.`subscribe_effects` (  
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `type of cost_id` INT UNSIGNED NOT NULL,  
  `cost` INT UNSIGNED NOT NULL,  
  PRIMARY KEY (`id`),  
  INDEX `fk_subscribe_effects_type of cost1_idx` (`type of cost_id` ASC) VISIBLE,  
  CONSTRAINT `fk_subscribe_effects_type of cost1`  
    FOREIGN KEY (`type of cost_id`)  
    REFERENCES `movavi_course`.`type_of_cost` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `movavi_course`.`effect_store`

```
CREATE TABLE IF NOT EXISTS `movavi_course`.`effect_store` (  
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `who_bought_id` INT UNSIGNED NULL COMMENT 'Кто купил набор эффектов\n',  
  `sets_id` INT UNSIGNED NULL,  
  `intros_id` INT UNSIGNED NULL,  
  `subscribe_effects_id` INT UNSIGNED NULL,  
  PRIMARY KEY (`id`),  
  INDEX `fk_effect_store_user1_idx` (`who_bought_id` ASC) VISIBLE,  
  INDEX `fk_effect_store_sets1_idx` (`sets_id` ASC) VISIBLE,  
  INDEX `fk_effect_store_intros1_idx` (`intros_id` ASC) VISIBLE,
```

```

INDEX `fk_effect_store_subscribe_effects1_idx` (`subscribe_effects_id` ASC) VISIBLE,
CONSTRAINT `fk_effect_store_user1`
  FOREIGN KEY (`who_bought_id`)
  REFERENCES `movavi_course`.`user` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_effect_store_sets1`
  FOREIGN KEY (`sets_id`)
  REFERENCES `movavi_course`.`sets` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_effect_store_intros1`
  FOREIGN KEY (`intros_id`)
  REFERENCES `movavi_course`.`intros` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_effect_store_subscribe_effects1`
  FOREIGN KEY (`subscribe_effects_id`)
  REFERENCES `movavi_course`.`subscribe_effects` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `movavi_course`.`instructions`
-----

```

```

CREATE TABLE IF NOT EXISTS `movavi_course`.`instructions` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NOT NULL,
  `text` TEXT NOT NULL,
  `programms_id` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE INDEX `name_UNIQUE` (`name` ASC) VISIBLE,
  INDEX `fk_instructions_programms1_idx` (`programms_id` ASC) VISIBLE,
  CONSTRAINT `fk_instructions_programms1`
    FOREIGN KEY (`programms_id`)
    REFERENCES `movavi_course`.`programms` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `movavi_course`.`knowledge_base`
-----

```

```

CREATE TABLE IF NOT EXISTS `movavi_course`.`knowledge_base` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `folder_name` VARCHAR(255) NOT NULL COMMENT 'Название разделов в базе знаний',
  PRIMARY KEY (`id`),
  UNIQUE INDEX `folder_UNIQUE` (`folder_name` ASC) VISIBLE)
ENGINE = InnoDB;

```

-- Table `movavi_course`.`article`

```
-----  
CREATE TABLE IF NOT EXISTS `movavi_course`.`article` (  
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `article_name` VARCHAR(255) NOT NULL,  
  `text` TEXT NULL,  
  `knowledge base_id` INT UNSIGNED NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE INDEX `article_name_UNIQUE` (`article_name` ASC) VISIBLE,  
  INDEX `fk_article_knowledge base1_idx` (`knowledge base_id` ASC) VISIBLE,  
  CONSTRAINT `fk_article_knowledge base1`  
    FOREIGN KEY (`knowledge base_id`)  
      REFERENCES `movavi_course`.`knowledge_base` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `movavi_course`.`cost_of_prog_or_effect`

```
-----  
CREATE TABLE IF NOT EXISTS `movavi_course`.`cost_of_prog_or_effect` (  
  `cost_id` INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  `cost` FLOAT UNSIGNED NOT NULL,  
  `programms_id` INT UNSIGNED NULL,  
  `intros_id` INT UNSIGNED NULL,  
  `sets_id` INT UNSIGNED NULL,  
  `type_of_cost_id` INT UNSIGNED NOT NULL,  
  `type_of_subscribe_id` TINYINT(2) UNSIGNED NULL,  
  PRIMARY KEY (`cost_id`),  
  INDEX `fk_cost_of_programm_programms1_idx` (`programms_id` ASC) VISIBLE,  
  INDEX `fk_cost_of_programm_type of cost1_idx` (`type_of_cost_id` ASC) VISIBLE,  
  INDEX `fk_cost_of_programm_type_of_subscribe1_idx` (`type_of_subscribe_id` ASC) VISIBLE,  
  INDEX `fk_cost_of_programm_intros1_idx` (`intros_id` ASC) VISIBLE,  
  INDEX `fk_cost_of_programm_sets1_idx` (`sets_id` ASC) VISIBLE,  
  CONSTRAINT `fk_cost_of_programm_programms1`  
    FOREIGN KEY (`programms_id`)  
      REFERENCES `movavi_course`.`programms` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_cost_of_programm_type of cost1`  
    FOREIGN KEY (`type_of_cost_id`)  
      REFERENCES `movavi_course`.`type_of_cost` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_cost_of_programm_type_of_subscribe1`  
    FOREIGN KEY (`type_of_subscribe_id`)  
      REFERENCES `movavi_course`.`type_of_subscribe` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `fk_cost_of_programm_intros1`  
    FOREIGN KEY (`intros_id`)  
      REFERENCES `movavi_course`.`intros` (`id`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,
```

```

CONSTRAINT `fk_cost_of_programm_sets1`
  FOREIGN KEY (`sets_id`)
  REFERENCES `movavi_course`.`sets` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `movavi_course`.`user_has_programms`
-----
CREATE TABLE IF NOT EXISTS `movavi_course`.`user_has_programms` (
  `user_id` INT UNSIGNED NOT NULL,
  `programms_id` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`user_id`, `programms_id`),
  INDEX `fk_user_has_programms_programms1_idx` (`programms_id` ASC) VISIBLE,
  INDEX `fk_user_has_programms_user1_idx` (`user_id` ASC) VISIBLE,
  CONSTRAINT `fk_user_has_programms_user1`
    FOREIGN KEY (`user_id`)
    REFERENCES `movavi_course`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_user_has_programms_programms1`
    FOREIGN KEY (`programms_id`)
    REFERENCES `movavi_course`.`programms` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

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

```

3.2.1 Наполнение таблиц с помощью сервиса filldb на примере таблицы country

```

LOCK TABLES `country` WRITE;
/*!40000 ALTER TABLE `country` DISABLE KEYS */;
INSERT INTO `country` VALUES (1,'Honduras'),(2,'Bangladesh'),(3,'Korea'),(4,'Russian
Federation'),(5,'Ireland'),(6,'Lithuania'),(7,'Philippines'),(8,'Nigeria'),(9,'Ecuador'),(10,'Niue'),(11,'Saint
Helena'),(12,'Bulgaria'),(13,'Cuba'),(14,'Guernsey'),(15,'United Arab
Emirates'),(16,'Seychelles'),(17,'Samoa'),(18,'Northern Mariana Islands'),(19,'Marshall
Islands'),(20,'Tajikistan');
/*!40000 ALTER TABLE `country` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

```

3.2.2 Наполнение таблиц на примере таблицы programms

```

INSERT INTO `movavi_course`.`programms` (`name`, `description`, `category_id`) VALUES ('Movavi
Unlimited', 'Ultimate solution that combines all Movavi programs and effects in one bundle. Create
videos, edit photos, record screencasts, work with documents, and much more.', '1');
INSERT INTO `movavi_course`.`programms` (`name`, `description`, `category_id`) VALUES ('Movavi

```


Video Suite', 'All-in-one for multimedia processing: create video clips and slideshows, convert media between 180+ formats, record screen and webcam, burn DVDs and Blu-ray discs.', '1');
 INSERT INTO `movavi_course`.`programms` (`name`, `description`, `category_id`) VALUES ('Movavi Video Converter Premium', 'Movavi Video Converter Premium offers effective and fast video conversion of files for an extensive range of devices. Use essential editing tools: merge, rotate, crop, adjust the sound, improve the quality.', '1');
 INSERT INTO `movavi_course`.`programms` (`name`, `description`, `category_id`) VALUES ('Movavi Video Editor Plus', 'A comprehensive, streamlined video editing program that has everything you need to create amazing home movies with custom soundtracks, special effects, cool titles, and transitions.', '1');
 INSERT INTO `movavi_course`.`programms` (`name`, `description`, `category_id`) VALUES ('Movavi Screen Recorder', 'Create screen videos for learning and fun: record live streams, webinars, desktop and apps, Skype calls, and more. Save recordings as videos and GIF animations.', '4');
 INSERT INTO `movavi_course`.`programms` (`name`, `description`, `category_id`) VALUES ('Movavi Screen Recorder for Business', 'Capture business webinars and online calls. Record video tutorials for colleagues and clients. Review code, design, or packaging.', '4');
 INSERT INTO `movavi_course`.`programms` (`name`, `description`, `category_id`) VALUES ('Gecata by Movavi', 'A powerful game video recorder that lets you capture HD video from PC games at up to 120 fps in YouTube-ready MP4. Record audio from microphone and speakers, add webcam footage.', '4');
 INSERT INTO `movavi_course`.`programms` (`name`, `description`, `category_id`) VALUES ('Movavi Slideshow Maker', 'Make videos from your digital photo collections. Add images and video clips, link them with animated transitions, use filters to add atmosphere, add music, and more.', '3');
 INSERT INTO `movavi_course`.`programms` (`name`, `description`, `category_id`) VALUES ('Movavi Picverse', 'Use AI-based tools to experiment with images and achieve stunning results in a snap. Erase unwanted objects from photos. Remove and change image backgrounds. Restore scanned old photos. Enhance image quality manually or automatically. Emphasize texture and get rid of blur.', '3');
 INSERT INTO `movavi_course`.`programms` (`name`, `description`, `category_id`) VALUES ('Movavi Photo Manager', 'Manage your digital photo collection with ease. Automatically create albums sorted by date or location. Add tags to pictures. Quickly find the image you need out of thousands of pictures.', '3');
 INSERT INTO `movavi_course`.`programms` (`name`, `description`, `category_id`) VALUES ('PDFChef by Movavi', 'A convenient tool for working with documents. View, edit, and organize your PDFs. Insert, crop, and resize images, including logos, signatures, and stamps. Fill out popular business and legal forms.', '2');

4 ХАРАКТЕРНЫЕ ВЫБОРКИ

4.1 Запрос на подсчет количества пользователей из разных стран

```
SELECT
COUNT(u.id) as count_of_users,
c.name as country_name
FROM
`user` as u
JOIN
`country` as c
ON
u.country_id = c.id
GROUP BY
country_name
ORDER BY country_name
```

count_of_users	country_name
41	Bangladesh
52	Bulgaria
44	Cuba
51	Ecuador
57	Guernsey
60	Honduras
54	Ireland
47	Korea
53	Lithuania
53	Marshall Islands
39	Nigeria
56	Niue
43	Northern Mariana Islands
45	Philippines
43	Russian Federation
37	Saint Helena
54	Samoa
57	Seychelles
62	Tajikistan
52	United Arab Emirates

4.2 Запрос на подсчет списка пользователей, купивших программы

```

SELECT
GROUP_CONCAT(uhp.user_id) as users_who_bought,
(SELECT name FROM `programms` where programms.id = uhp.programms_id) as programm_name,
cop.cost,
(SELECT name FROM `type_of_cost` where type_of_cost.id = cop.type_of_cost_id) as type_of_cost,
sum(cost)
FROM
user_has_programms as uhp
join
cost_of_prog_or_effect as cop
on uhp.type_of_cost_id = cop.type_of_cost_id and uhp.programms_id = cop.programms_id
GROUP BY
programm_name, type_of_cost
ORDER BY
uhp.programms_id

```

users_who_bought	programm_name	cost	type_of_cost	sum(cost)
78,452,100,122,144,188,210,232,254,27...	Movavi Unlimited	7190	business	323550
1,551,573,595,617,639,661,683,705,727...	Movavi Unlimited	3590	personal	165140
442,420,398,376,354,332,310,288,266,2...	Movavi Video Suite	4990	business	229540
387,365,343,321,299,277,255,233,211,1...	Movavi Video Suite	1990	personal	89550
938,410,388,366,344,322,300,278,256,2...	Movavi Video Converter Premium	3225	business	145125
421,399,377,355,333,311,289,267,245,2...	Movavi Video Converter Premium	1290	personal	59340
4,642,664,686,708,752,774,796,818,840...	Movavi Video Editor Plus	4190	business	192740
983,961,939,917,895,873,851,829,807,7...	Movavi Video Editor Plus	1690	personal	76050
71,93,115,137,159,181,203,225,247,269...	Movavi Screen Recorder	1190	personal	127880
61,479,457,435,413,391,369,347,325,30...	Movavi Screen Recorder for Business	1790	personal	80550
601,997,7,29,51,73,95,117,139,161,205...	Gecata by Movavi	1990	personal	127880
426,602,624,646,668,690,712,734,756,7...	Movavi Slideshow Maker	3725	business	171350
569,591,613,635,657,679,701,723,745,7...	Movavi Slideshow Maker	1490	personal	67050
31,9,999,977,955,933,911,889,867,845,...	Movavi Picverse	1690	personal	137080
362,164,1000,978,934,912,890,868,846,...	Movavi Photo Manager	3290	business	151340
351,329,307,285,263,241,219,197,175,1...	Movavi Photo Manager	1290	personal	58050
22,44,66,88,110,132,154,176,198,220,2...	PDFChef by Movavi	1990	business	201600

Добавляя перед ORDER BY команду WITH ROLL UP получаем общую сумму заработанных на продаже программ денег, равную 2403815.

users_who_bought	programm_name	cost	type_of_cost	sum(cost)
351,329,307,285,263,241,219,197,175,1...	Movavi Photo Manager	1290	personal	58050
362,164,1000,362,164,1000,978,934,912,890,868,846,824,802,780,758,736,714,692,670,648,626,604,582,560,538,516				
22,44,66,88,110,132,154,176,198,220,2...	PDFChef by Movavi	1990	business	201600
22,44,66,88,110,132,154,176,198,220,2...	PDFChef by Movavi	1990	NULL	201600
601,997,7,29,51,73,95,117,139,161,205...	NULL	1990	NULL	2403815

5 Представления

В качестве первого представления создаем запрос на подсчет количества пользователей из разных стран, созданный в п. 4.1.

```
USE `movavi_course`;
CREATE OR REPLACE VIEW `view_of_users_from_country` AS
SELECT
COUNT(u.id) as count_of_users,
c.name as country_name
FROM
`user` as u
JOIN
`country` as c
ON
u.country_id = c.id
GROUP BY
country_name
ORDER BY country_name;
```

В качестве второго представления создадим запрос, который показывает какие программы больше покупали в разных странах.

```

USE `movavi_course`;
CREATE OR REPLACE VIEW `count_of_programms_by_country` AS
SELECT
COUNT(uhp.user_id) as count_of_buy,
pr.name as programm,
c.name as country
FROM
user_has_programms as uhp
join (
programms as pr,
user as u,
country as c)
on (uhp.programms_id = pr.id and uhp.user_id = u.id and c.id = u.country_id)
GROUP BY pr.name, c.name
ORDER BY country, pr.id;

```

Пример:

```
SELECT * FROM movavi_course.count_of_programms_by_country limit 25;
```

count_of_buy	programm	country
5	Movavi Unlimited	Bangladesh
4	Movavi Video Suite	Bangladesh
7	Movavi Video Converter Premium	Bangladesh
2	Movavi Video Editor Plus	Bangladesh
4	Movavi Screen Recorder	Bangladesh
4	Movavi Screen Recorder for Business	Bangladesh
3	Gecata by Movavi	Bangladesh
2	Movavi Slideshow Maker	Bangladesh
1	Movavi Picverse	Bangladesh
3	Movavi Photo Manager	Bangladesh
6	PDFChef by Movavi	Bangladesh
4	Movavi Unlimited	Bulgaria
7	Movavi Video Suite	Bulgaria
2	Movavi Video Converter Premium	Bulgaria
4	Movavi Video Editor Plus	Bulgaria
5	Movavi Screen Recorder	Bulgaria
5	Movavi Screen Recorder for Business	Bulgaria
8	Gecata by Movavi	Bulgaria
9	Movavi Slideshow Maker	Bulgaria
2	Movavi Picverse	Bulgaria
2	Movavi Photo Manager	Bulgaria
4	PDFChef by Movavi	Bulgaria
3	Movavi Unlimited	Cuba
3	Movavi Video Suite	Cuba
5	Movavi Video Converter Premium	Cuba

В дальнейшем данной представление можно использовать для более глубокого анализа данных.

Аналогичное представление создаем для магазина эффектов: наборов, интро.

```
USE `movavi_course`;
```

```

CREATE OR REPLACE VIEW `count_of_sets_by_country` AS
SELECT
COUNT(es.who_bought_id) as count_of_buy,
sets.name as set_name,
c.name as country
FROM
`effect_store` as es
join (
sets,
user,
country as c)
on (es.sets_id = sets.id and es.who_bought_id = user.id and c.id = user.country_id)
GROUP BY sets.name, c.name
ORDER BY country;

```

```

USE `movavi_course`;
CREATE OR REPLACE VIEW `count_of_intros_by_country` AS
SELECT
COUNT(es.who_bought_id) as count_of_buy,
intros.name as intro_name,
c.name as country
FROM
`effect_store` as es
join (
intros,
user,
country as c)
on (es.intros_id = intros.id and es.who_bought_id = user.id and c.id = user.country_id)
GROUP BY intros.name, c.name
ORDER BY country;

```

6 ХРАНИМЫЕ ПРОЦЕДУРЫ И ТРИГГЕРЫ

Процедура на добавление нового пользователя:

```

USE `movavi_course`;
DROP procedure IF EXISTS `add_new_user`;

DELIMITER $$
USE `movavi_course`$$
CREATE PROCEDURE `add_new_user` (IN value VARCHAR(245))
BEGIN
    SET @email = value;
    INSERT INTO `movavi_course`.`user` (`email`) VALUES (@email);
END$$

DELIMITER ;

```

```
call movavi_course.add_new_user('kolya@example.net');
```

1002	kolya@example.net	NULL	2021-08-21 17:44:27	NULL	NULL	NULL
------	-------------------	------	---------------------	------	------	------

Триггер на добавление пользователя о необходимости добавить страну проживания:

```
DROP TRIGGER IF EXISTS `movavi_course`.`user_BEFORE_INSERT`;  
  
DELIMITER $$  
USE `movavi_course`$$  
CREATE DEFINER = CURRENT_USER TRIGGER `movavi_course`.`user_BEFORE_INSERT`  
BEFORE INSERT ON `user` FOR EACH ROW  
BEGIN  
    IF NEW.country_id is NULL then  
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'При добавлении  
пользователя нужно указать страну';  
    end if;  
END$$  
DELIMITER ;
```

Попытка вставить пользователя без id страны:

Operation failed: There was an error while applying the SQL script to the database.
Executing:

```
INSERT INTO `movavi_course`.`user` (`email`) VALUES ('frost@list.net');
```

ERROR 1644: 1644: При добавлении пользователя нужно указать страну
SQL Statement:

```
INSERT INTO `movavi_course`.`user` (`email`) VALUES ('frost@list.net')
```

Вводим id страны 6 и новый пользователь добавляется.

1005	frost@list.net	NULL	2021-08-21 18:05:49	NULL	NULL	6
------	----------------	------	---------------------	------	------	---