

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«КАБАРДИНО – БАЛКАРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМ. Х.М. БЕРБЕКОВА»**

**ИНСТИТУТ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И ЦИФРОВЫХ ТЕХНОЛОГИЙ
КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ
И ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ**

**Отчет по практической работе №1
«Построение ER-модели для заданной предметной области»**

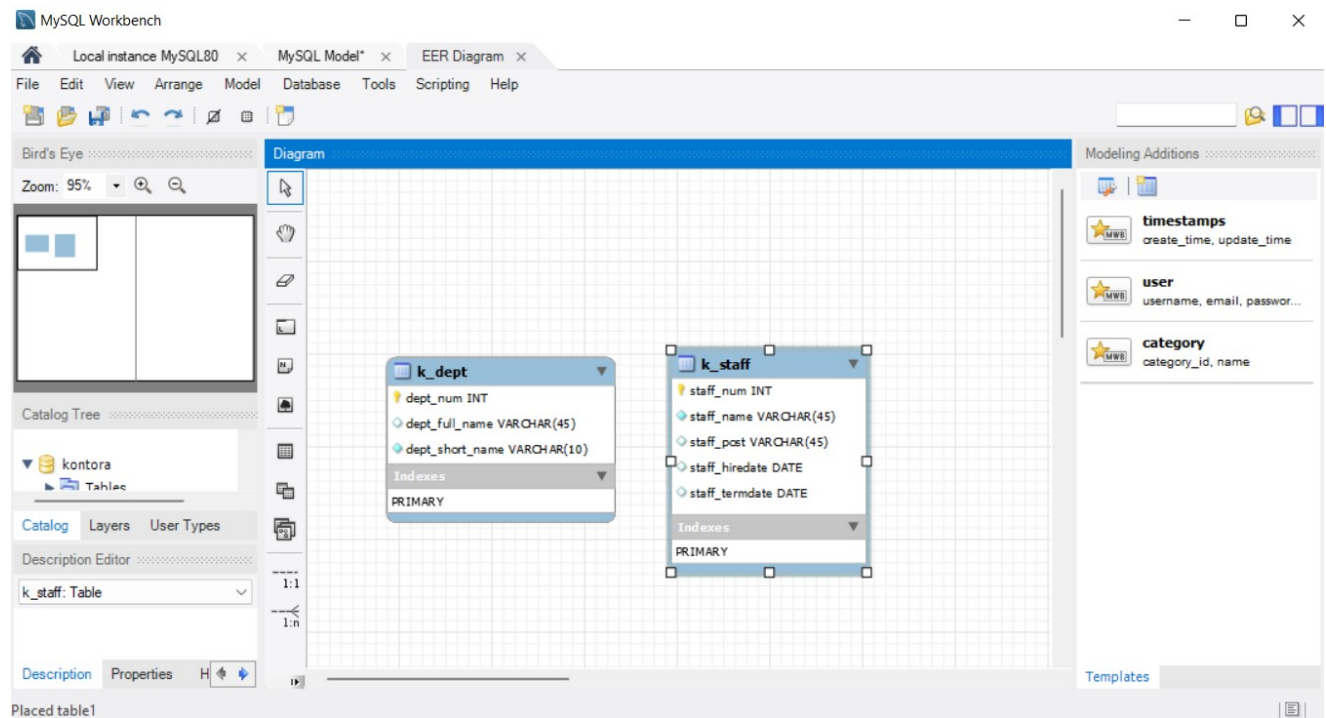
Выполнил: студент 1 г.о. ИВТ Афаунова В.А

Преподаватель: кандидат физ.-мат. наук, доцент
кафедры КТиИБ ИИИиЦТ КБГУ Калмыков Р.М.

Нальчик 2023

Цель работы: проектирование информационно-логической модели базы данных при помощи case-средства MySQL Workbench

Процесс создания и конечный вариант модели БД представлен на снимках ниже:



MySQL Workbench

Local instance MySQL80 x MySQL Model* x EER Diagram x

File Edit View Arrange Model Database Tools Scripting Help

Bird's Eye Diagram

Zoom: 95%

Catalog Tree

- kontora
 - Tables
 - Views
 - Routine Groups

Modeling Additions

- timestamps
 - create_time, update_time
- user
 - username, email, passwor...
- category
 - category_id, name

Diagram

k_dept

- dept_num INT
- 3 more...
- Indexes
 - PRIMARY
 - fk_k_dept_k_staff1_idx

k_staff

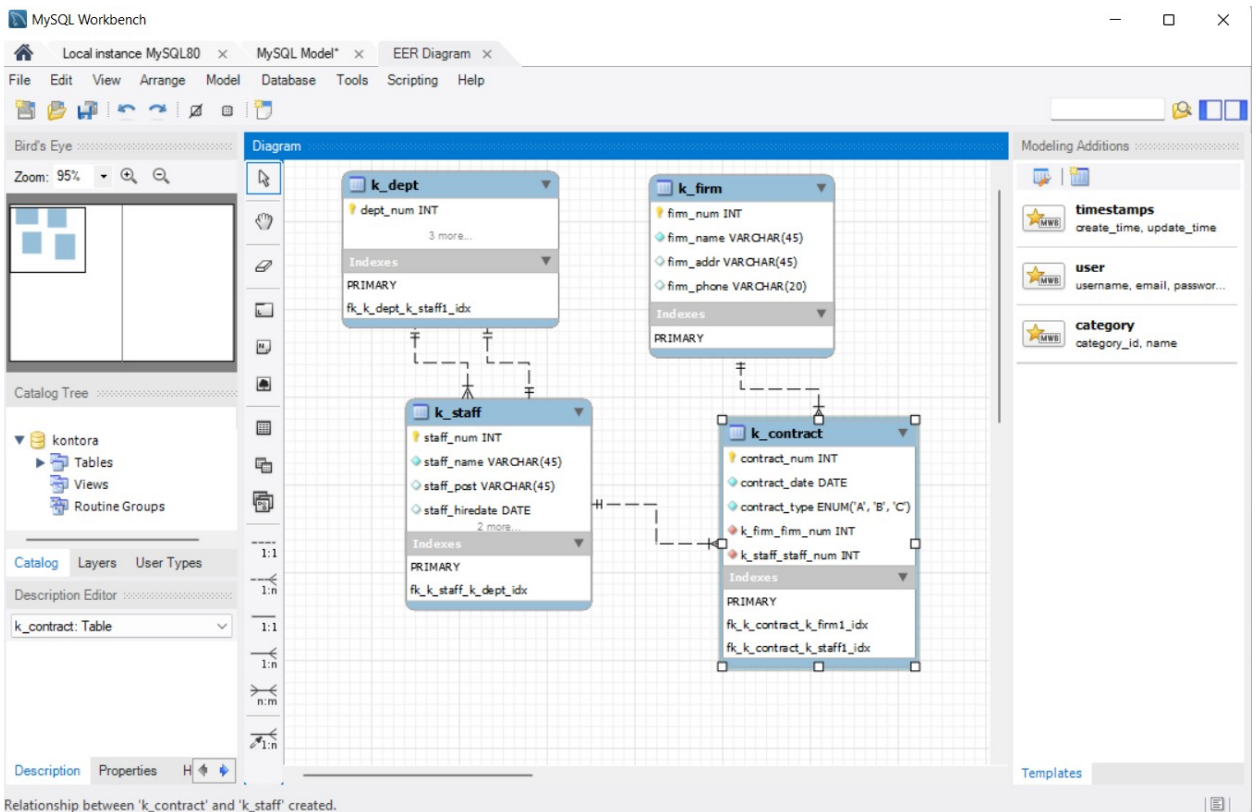
- staff_num INT
- staff_name VARCHAR(45)
- staff_post VARCHAR(45)
- staff_hiredate DATE
- 2 more...
- Indexes
 - PRIMARY
 - fk_k_staff_k_dept_idx

Relationships:

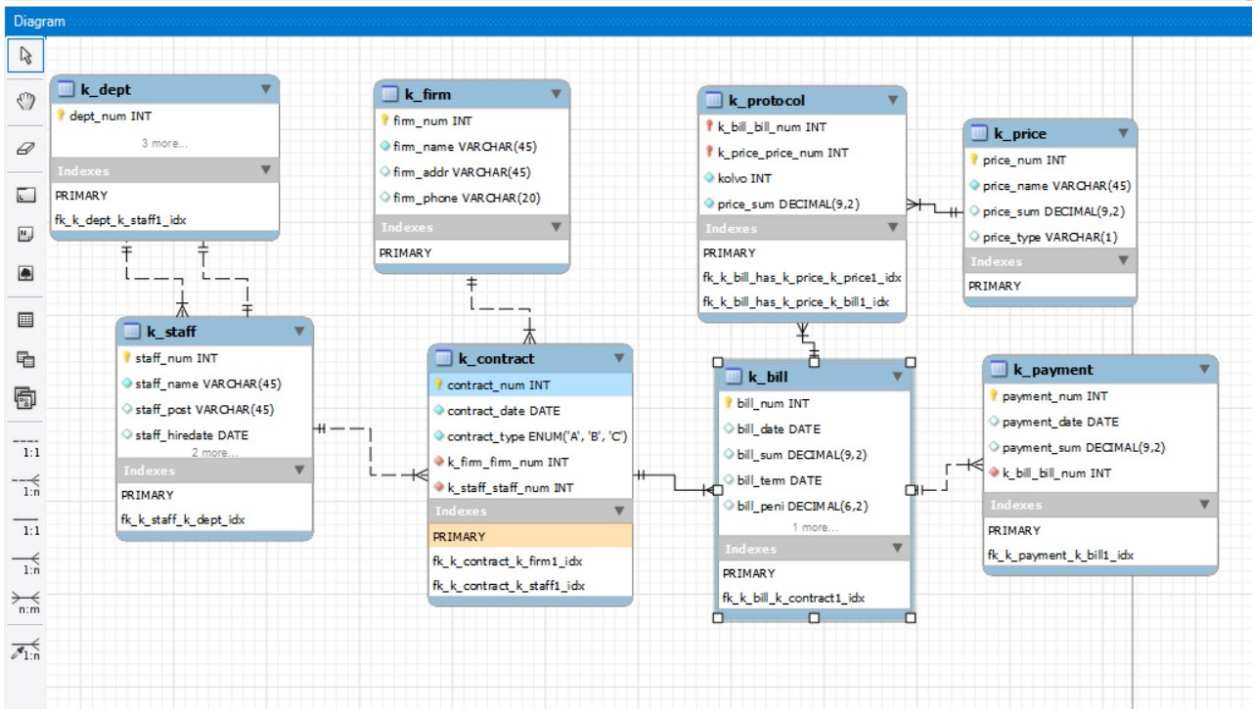
- k_dept (1) to k_staff (n)
- k_dept (1) to k_staff (1)
- k_dept (1) to k_staff (n)
- k_dept (1) to k_staff (1)
- k_dept (n) to k_staff (m)
- k_dept (1) to k_staff (n)

Description Properties H

Relationship between 'k_dept' and 'k_staff' created.



Relationship between 'k_contract' and 'k_staff' created.



Скриптинг:

-- 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 kontora
```

```
-----
```

```
-----
```

```
-- Schema kontora
```

```
-----
```

```
CREATE SCHEMA IF NOT EXISTS `kontora` DEFAULT CHARACTER SET utf8 ;
```

```
USE `kontora` ;
```

```
-----
```

```
-- Table `kontora`.`k_staff`
```

```
-----
```

```
CREATE TABLE IF NOT EXISTS `kontora`.`k_staff` (
```

```
  `staff_num` INT NOT NULL AUTO_INCREMENT,
```

```
  `staff_name` VARCHAR(45) NOT NULL,
```

```
  `staff_post` VARCHAR(45) NULL,
```

```
  `staff_hiredate` DATE NULL,
```

```
  `staff_termdate` DATE NULL,
```

```
  `k_dept_dept_num` INT NOT NULL,
```

```
  PRIMARY KEY (`staff_num`),
```

```
  INDEX `fk_k_staff_k_dept_idx` (`k_dept_dept_num` ASC) VISIBLE,
```

```
  CONSTRAINT `fk_k_staff_k_dept`
```

```
    FOREIGN KEY (`k_dept_dept_num`)
```

```
    REFERENCES `kontora`.`k_dept` (`dept_num`)
```

```
    ON DELETE NO ACTION
```

```
    ON UPDATE NO ACTION)
```

```
ENGINE = InnoDB;
```

```
-----
```

```
-- Table `kontora`.`k_dept`
```

```

-----
CREATE TABLE IF NOT EXISTS `kontora`.`k_dept` (
  `dept_num` INT NOT NULL AUTO_INCREMENT,
  `dept_full_name` VARCHAR(45) NULL,
  `dept_short_name` VARCHAR(10) NOT NULL,
  `k_staff_staff_num` INT NOT NULL,
  PRIMARY KEY (`dept_num`),
  INDEX `fk_k_dept_k_staff1_idx` (`k_staff_staff_num` ASC) VISIBLE,
  CONSTRAINT `fk_k_dept_k_staff1`
  FOREIGN KEY (`k_staff_staff_num`)
  REFERENCES `kontora`.`k_staff` (`staff_num`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `kontora`.`k_firm`

```

```

-----
CREATE TABLE IF NOT EXISTS `kontora`.`k_firm` (
  `firm_num` INT NOT NULL AUTO_INCREMENT,
  `firm_name` VARCHAR(45) NOT NULL,
  `firm_addr` VARCHAR(45) NULL,
  `firm_phone` VARCHAR(20) NULL,
  PRIMARY KEY (`firm_num`))
ENGINE = InnoDB;

```

```

-----
-- Table `kontora`.`k_contract`

```

```

-----
CREATE TABLE IF NOT EXISTS `kontora`.`k_contract` (
  `contract_num` INT NOT NULL AUTO_INCREMENT,
  `contract_date` DATE NOT NULL,

```

```

`contract_type` ENUM('A', 'B', 'C') NOT NULL,
`k_firm_firm_num` INT NOT NULL,
`k_staff_staff_num` INT NOT NULL,
PRIMARY KEY (`contract_num`),
INDEX `fk_k_contract_k_firm1_idx` (`k_firm_firm_num` ASC) VISIBLE,
INDEX `fk_k_contract_k_staff1_idx` (`k_staff_staff_num` ASC) VISIBLE,
CONSTRAINT `fk_k_contract_k_firm1`
  FOREIGN KEY (`k_firm_firm_num`)
  REFERENCES `kontora`.`k_firm` (`firm_num`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_k_contract_k_staff1`
  FOREIGN KEY (`k_staff_staff_num`)
  REFERENCES `kontora`.`k_staff` (`staff_num`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `kontora`.`k_bill`
-----

CREATE TABLE IF NOT EXISTS `kontora`.`k_bill` (
  `bill_num` INT NOT NULL AUTO_INCREMENT,
  `bill_date` DATE NULL,
  `bill_sum` DECIMAL(9,2) NULL,
  `bill_term` DATE NULL,
  `bill_peni` DECIMAL(6,2) NULL,
  `k_contract_contract_num` INT NOT NULL,
  PRIMARY KEY (`bill_num`, `k_contract_contract_num`),
  INDEX `fk_k_bill_k_contract1_idx` (`k_contract_contract_num` ASC) VISIBLE,
  CONSTRAINT `fk_k_bill_k_contract1`

```

```

FOREIGN KEY (`k_contract_contract_num`)
REFERENCES `kontora`.`k_contract` (`contract_num`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----

-- Table `kontora`.`k_payment`
-----

CREATE TABLE IF NOT EXISTS `kontora`.`k_payment` (
`payment_num` INT NOT NULL AUTO_INCREMENT,
`payment_date` DATE NULL,
`payment_sum` DECIMAL(9,2) NULL,
`k_bill_bill_num` INT NOT NULL,
PRIMARY KEY (`payment_num`),
INDEX `fk_k_payment_k_bill1_idx` (`k_bill_bill_num` ASC) VISIBLE,
CONSTRAINT `fk_k_payment_k_bill1`
FOREIGN KEY (`k_bill_bill_num`)
REFERENCES `kontora`.`k_bill` (`bill_num`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----

-- Table `kontora`.`k_price`
-----

CREATE TABLE IF NOT EXISTS `kontora`.`k_price` (
`price_num` INT NOT NULL AUTO_INCREMENT,
`price_name` VARCHAR(45) NOT NULL,
`price_sum` DECIMAL(9,2) NULL,
`price_type` VARCHAR(1) NULL,
PRIMARY KEY (`price_num`))

```


ENGINE = InnoDB;

-- Table `kontora`.`k_protocol`

```
CREATE TABLE IF NOT EXISTS `kontora`.`k_protocol` (  
  `k_bill_bill_num` INT NOT NULL,  
  `k_price_price_num` INT NOT NULL,  
  `kolvo` INT NOT NULL,  
  `price_sum` DECIMAL(9,2) NOT NULL,  
  PRIMARY KEY (`k_bill_bill_num`, `k_price_price_num`),  
  INDEX `fk_k_bill_has_k_price_k_price1_idx` (`k_price_price_num` ASC) VISIBLE,  
  INDEX `fk_k_bill_has_k_price_k_bill1_idx` (`k_bill_bill_num` ASC) VISIBLE,  
  CONSTRAINT `fk_k_bill_has_k_price_k_bill1`  
    FOREIGN KEY (`k_bill_bill_num`)  
    REFERENCES `kontora`.`k_bill` (`bill_num`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_k_bill_has_k_price_k_price1`  
    FOREIGN KEY (`k_price_price_num`)  
    REFERENCES `kontora`.`k_price` (`price_num`)  
    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;

Вывод: научились проектировать информационно-логическую модель базы данных при помощи case-средства `mysql Workbench`