Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное учреждение высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

Кафедра «информационных технологий и систем»

**КУРСОВАЯ РАБОТА**

по дисциплине: «Базы данных»

на тему: «БД «Курсовые работы»»

Разработал:

Студент группы 8091

Шаклеин В. В.\_\_\_\_\_\_

«\_\_».\_\_\_\_\_.2020г.

Проверил преподаватель:

Телина И. С.\_\_\_\_\_\_

«\_\_».\_\_\_\_\_.2020г.

**Великий Новгород**

**2020**

Содержание

[Введение **Error! Bookmark not defined.**](#_Toc59500699)

[ГЛАВА 1. Структурный анализ 4](#_Toc59500700)

[Анализ предметной области и требований к информационной системе 4](#_Toc59500701)

[Инфологическая модель 6](#_Toc59500702)

[ГЛАВА 2. Физическая реализация реляционной БД 7](#_Toc59500703)

[Разработка таблиц БД и обеспечение целостности данных 7](#_Toc59500704)

[Разработка запросов 9](#_Toc59500705)

[ГЛАВА 3. Разработка основных программных объектов базы данных 9](#_Toc59500706)

[Разработка триггеров 9](#_Toc59500707)

[Разработка представлений 10](#_Toc59500708)

[ЗАКЛЮЧЕНИЕ 11](#_Toc59500709)

# ВВЕДЕНИЕ

Целью данной работы является создание базы данных на тему “Курсовые работы”: студенты разных групп выполняют курсовые работы на разные темы по различным дисциплинам под руководством разных преподавателей.

В ходе проделанной работы были реализованы запросы (6 простых и 2 сложных), 1 триггер(для сохранения всех добавленных тем курсовой в архив), 2 представления(для объединения данных относящихся к дисциплине преподавателя и курсовой работе). Кроме этого, была разработана структура базы данных, а также таблицы были заполнены данным.

В качестве СУБД была выбрана “SQLite”, а для разработки клиентского приложения был использован Javascript, с использованием библиотеки sqlite3 для осуществления запросов. В качестве среды разработки использовалась Visual Studio Code.

# ГЛАВА 1. Структурный анализ

## Анализ предметной области и требований к информационной системе

|  |  |  |
| --- | --- | --- |
| **Название сущности** | **Идентификатор сущности** | **Назначение сущности** |
| Студент | student | Содержит информацию о студентах |
| Преподаватель | teacher | Содержит информацию о преподавателях |
| Дисциплина | discipline | Содержит информацию о дисциплинах |
| Дисциплина преподавателя | course | Связывает преподавателя и дисциплину |
| Курсовая работа | coursework | Содержит информацию о курсовых работых |
| Архив тем курсовых работ | coursework\_themes\_archive | Сохраняет все уникальных темах курсовых работ |

Таблица 1 Таблица сущностей

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя столбца | Содержательное описание | Тип данных | Размерность | Область допустимых значений | Возможность значения NULL | Пример |
| student\_id | Идентификатор студента | INTEGER | - | 0-… | False | 1 |
| student\_name | Имя студента | VARCHAR | 50 | ‘А-я’ | False | Всеволод |
| student\_surname | Фамилия студента | VARCHAR | 50 | ‘А-я’ | False | Шаклеин |
| student\_patronymic | Отчество студента | VARCHAR | 50 | ‘А-я’ | True | Владиславович |
| student\_id | Идентификатор преподавателя | INTEGER | - | 0-… | False | 1 |
| student\_name | Имя преподавателя | VARCHAR | 50 | ‘А-я’ | False | Ирина |
| student\_surname | Фамилия преподавателя | VARCHAR | 50 | ‘А-я’ | False | Телина |
| student\_patronymic | Отчество преподавателя | VARCHAR | 50 | ‘А-я’ | True | Сергеевна |
| discipline\_id | Идентификатор дисциплины | INTEGER | - | 0-… | False | 1 |
| discipline\_name | Название дисциплины | VARCHAR | 50 | ‘А-я’ | False | Базы данных |
| course\_id | Идентификатор дисциплины преподавателя | INTEGER |  | 0-… | False | 1 |
| coursework\_theme | Тема курсовой работы | VARCHAR | 500 | Любой текст | False | Почему параллельна ложная цитата? |

Таблица 2 Поля сущностей

## Инфологическая модель

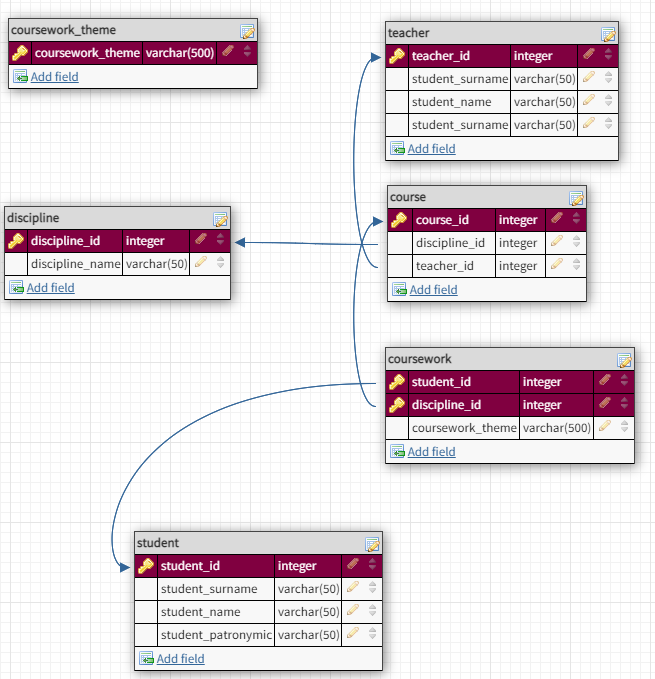


Рисунок 1 Инфологическая модель

Предлагаемая структура описывает реализацию структуры базы данных, в которой хранится информация о курсовых работах, студентах, которые их выполняют и преподавателях, которые их принимают.

# ГЛАВА 2. Физическая реализация реляционной БД

## Разработка таблиц БД и обеспечение целостности данных

Создание БД произведено с помощью DB Designer и ручного изменения запросов в VS Code.

SQL запросы, создающие таблицы в БД:

CREATE TABLE teacher (

    teacher\_id integer PRIMARY KEY AUTOINCREMENT,

    teacher\_surname varchar(50) NOT NULL,

    teacher\_name varchar(50) NOT NULL,

    teacher\_patronymic varchar(50)

  );

CREATE TABLE student (

    student\_id integer PRIMARY KEY AUTOINCREMENT,

    student\_surname varchar(50) NOT NULL,

    student\_name varchar(50) NOT NULL,

    student\_patronymic varchar(50)

  );

CREATE TABLE discipline (

    discipline\_id integer PRIMARY KEY AUTOINCREMENT,

    discipline\_name varchar(50) NOT NULL UNIQUE

  );

CREATE TABLE course (

    course\_id integer PRIMARY KEY AUTOINCREMENT,

    discipline\_id INTEGER REFERENCES discipline(discipline\_id) NOT NULL,

    teacher\_id INTEGER REFERENCES teacher(teacher\_id) NOT NULL

  );

CREATE TABLE coursework (

    student\_id INTEGER REFERENCES student(student\_id) NOT NULL,

    course\_id INTEGER REFERENCES course(course\_id) NOT NULL,

    coursework\_theme varchar(500) NOT NULL,

    PRIMARY KEY (student\_id, course\_id)

  );

CREATE TABLE coursework\_themes\_archive (

    coursework\_theme varchar(500) PRIMARY KEY

  );

SQL запросы, заполняющие таблицы исходными данными:

INSERT INTO

  discipline (

    discipline\_name

  )

  VALUES

('Экономика'),

('Теория языков программирования и методы трансляции'),

('Базы данных'),

…

INSERT INTO

  teacher (

    teacher\_name,

    teacher\_surname,

    teacher\_patronymic

  )

VALUES

('Анна', 'Макаревич', 'Николаевна'),

('Владимир', 'Макаров', 'Алексеевич'),

('Ирина', 'Телина', 'Сергеевна'),

…

INSERT INTO

  course (

    discipline\_id,

    teacher\_id

)

VALUES

  (1, 1),

(2, 2),

(3, 3),

…

INSERT INTO

  student (

    student\_name,

    student\_surname,

    student\_patronymic

)

VALUES

  ('Баранов', 'Алексей', 'Сергеевич'),

('Бочкарев', 'Борис', 'Анатольевич'),

('Васильев', 'Иван', 'Владимирович'),

…

INSERT INTO

  coursework (

    student\_id,

    course\_id,

    coursework\_theme

)

VALUES

  (1, 3, 'БД «Проектное бюро»: сотрудники разных отделов участвуют в различных проектах фирмы.'),

(2, 3, 'БД «Клиника»: пациенты из разных районов города лечатся в одной поликлинике у разных врачей по разным направлениям.'),

(3, 3, 'БД «Медицинский холдинг»: пациенты из разных районов города лечатся в нескольких медучреждениях медицинского холдинга у разных врачей.'),

…

## Разработка запросов

Далее приведены сложные запросы

Подсчёт количества дисциплин, которые ведёт преподаватель:

SELECT teacher.teacher\_id, teacher\_name, teacher\_surname, teacher\_patronymic, COUNT(\*) as count

      FROM teacher

      LEFT JOIN course

      ON course.teacher\_id = teacher.teacher\_id

      GROUP BY teacher.teacher\_id

Получение всех студентов и преподавателей:

SELECT teacher\_name as name, teacher\_surname as surname, teacher\_patronymic as patronymic, "Преподаватель" as role

      FROM teacher

      UNION

      SELECT student\_name as name, student\_surname as surname, student\_patronymic as patronymic, "Студент" as role

      FROM student

# ГЛАВА 3. Разработка основных программных объектов базы данных

## Разработка триггеров

Триггер, добавляющий тему курсовой в архив тем при её создании:

CREATE TRIGGER add\_theme\_to\_archive AFTER INSERT ON coursework

  BEGIN

    INSERT INTO coursework\_themes\_archive(coursework\_theme) VALUES (NEW.coursework\_theme);

  END

## Разработка представлений

Представление данных о дисциплине преподавателя:

CREATE VIEW v\_course

  AS SELECT

    course\_id,

    teacher.teacher\_id,

    teacher\_name,

    teacher\_surname,

    teacher\_patronymic,

    discipline.discipline\_id,

    discipline\_name

  FROM

    course

  LEFT JOIN teacher ON teacher.teacher\_id = course.teacher\_id

  LEFT JOIN discipline ON discipline.discipline\_id = course.discipline\_id

Представление данных о курсовой работе:

CREATE VIEW v\_coursework

  AS SELECT

    coursework\_theme,

    coursework.student\_id,

    coursework.course\_id,

    student\_name,

    student\_surname,

    student\_patronymic,

    teacher\_name,

    teacher\_surname,

    teacher\_patronymic,

    discipline\_name

  FROM

    coursework

  LEFT JOIN course ON coursework.course\_id = course.course\_id

  LEFT JOIN teacher ON teacher.teacher\_id = course.teacher\_id

  LEFT JOIN discipline ON discipline.discipline\_id = course.discipline\_id

  LEFT JOIN student ON student.student\_id = coursework.student\_id

# ЗАКЛЮЧЕНИЕ

В ходе выполнения задачи по курсовой работе, была создана база данных, а также клиентское приложение, позволяющие хранить, отображать и добавлять данные о курсовых работах. Была разработана инфологическая модель, создана БД с таблицами и заполненными данными, созданы триггеры, представления и индексы, а также написаны запросы.

