Дарья Яковлева, M3439

**Домашнее задание 3**

Отношение с атрибутами

StudentId, StudentName, GroupId, GroupName, CourseId, CourseName, LecturerId, LecturerName, Mark.

1. Функциональные зависимости в данном отношении:

StudentId → StudentName

GroupId → GroupName

CourseId → CourseName

LecturerId → LecturerName

StudentId CourseId LecturerId → Mark

GroupId CourseId → LecturerId

StudentId → GroupId

1. Ключи данного отношения:

{StudentId, CourseId}

1. Неприводимое множество функциональных зависимостей для данного отношения:

StudentId → StudentName

GroupId → GroupName

CourseId → CourseName

LecturerId → LecturerName

StudentId CourseId → Mark

StudentId CourseId → LecturerId

StudentId → GroupId

**Домашнее задание 4**

Дано отношение с атрибутами StudentId, StudentName, GroupId, GroupName, CourseId, CourseName, LecturerId, LecturerName, Mark.

1. Инкрементальное приведение отношения в 5 нормальную форму:

**1НФ**: StudentId, StudentName, GroupId, GroupName, CourseId, CourseName, LecturerId, LecturerName, Mark

**2НФ**:

StudentId, GroupId, GroupName, StudentName;

CourseId, CourseName;

StudentId, CourseId, LecturerId, LecturerName, Mark

**3НФ**:

StudentId, StudentName;

StudentId, GroupId;

GroupId, GroupName;

CourseId, CourseName;

StudentId, CourseId, LecturerId, Mark;

LecturerId, LecturerName

**4НФ, 5НФ:**

StudentId, StudentName;

StudentId, GroupId;

GroupId, GroupName;

CourseId, CourseName;

GroupId, CourseId, LecturerId;

StudentId, CourseId, Mark;

LecturerId, LecturerName

DROP DATABASE deanery;

CREATE DATABASE deanery;

DROP TABLE marks;

DROP TABLE lecturer;

DROP TABLE courses;

DROP TABLE lecturers;

DROP TABLE students;

DROP TABLE groups;

CREATE TABLE IF NOT EXISTS groups (

group\_id int PRIMARY KEY,

group\_name varchar(100)

);

CREATE TABLE IF NOT EXISTS students (

student\_id int PRIMARY KEY,

student\_name varchar(100),

group\_id int REFERENCES groups (group\_id)

);

CREATE TABLE IF NOT EXISTS lecturers (

lecturer\_id int PRIMARY KEY,

lecturer\_name varchar(50)

);

CREATE TABLE IF NOT EXISTS courses (

course\_id int PRIMARY KEY,

course\_name varchar(50)

);

CREATE TABLE IF NOT EXISTS marks (

mark\_value int,

course\_id int not null REFERENCES courses (course\_id),

student\_id int not null REFERENCES students (student\_id),

PRIMARY KEY (course\_id, student\_id)

);

CREATE TABLE IF NOT EXISTS lecturer (

lecturer\_id int,

course\_id int not null REFERENCES courses (course\_id),

group\_id int not null REFERENCES groups (group\_id),

PRIMARY KEY (course\_id, group\_id)

);

INSERT INTO groups (group\_id, group\_name)

VALUES (1, 'M3439'),

(2, 'M3339'),

(3, 'M3239'),

(4, 'M3139'),

(5, 'M3438');

INSERT INTO students (student\_id, student\_name, group\_id)

VALUES (1, 'Ivan Belonogov', 1),

(2, 'Evgeny Nemchenko', 5),

(3, 'Michail Putilin', 4),

(4, 'Appolinaria Romanova', 3),

(5, 'Anna Rodionova', 2),

(6, 'Vyacheslav Moklev', 1);

INSERT INTO lecturers (lecturer\_id, lecturer\_name)

VALUES (1, 'Georgiy Korneev'),

(2, 'Andrew Stankevich'),

(3, 'Oksana Pavlova');

INSERT INTO courses (course\_id, course\_name)

VALUES (1, 'Java'),

(2, 'Algorithm and Data Structures'),

(3, 'English'),

(4, 'Databases'),

(5, 'DMath');

INSERT INTO marks (mark\_value, course\_id, student\_id)

VALUES (100, 2, 1),

(100, 2, 2),

(100, 2, 3),

(100, 2, 4),

(100, 2, 5);

INSERT INTO lecturer (lecturer\_id, course\_id, group\_id)

VALUES (1, 1, 1),

(1, 4, 2),

(2, 2, 3),

(2, 5, 4),

(3, 3, 5);