**Задание 1.**

1. Вывести количество преподавателей кафедры "Software Development”.

**select** **count**(**distinct** teachers.TeacherId)

**from** teachers, lectures, groupslectures, groupsacademy, departments

**where** teachers.TeacherId = lectures.TeacherId

**and** lectures.LectureId = groupslectures.LectureId

**and** groupslectures.GroupId = groupsacademy.GroupId

**and** groupsacademy.departmentid = departments.DepartmentId

**and** departments.Name = 'Software Development';

2. Вывести количество лекций, которые читает преподаватель “Dave McQueen”.

**select** **count**(**distinct** LectureId) **from** lectures, teachers

**where** lectures.TeacherId = teachers.TeacherId

**and** teachers.Name = 'Dave'

**and** teachers.Surname = 'McQueen';

3. Вывести количество занятий, проводимых в аудитории “D201”.

**select** **count**(\*) **from** lectures **where** LecturesRoom = 'D201';

4. Вывести названия аудиторий и количество лекций, проводимых в них.

**select** LecturesRoom, **count**(\*) **from** lectures **group by** LecturesRoom;

5. Вывести количество студентов, посещающих лекции преподавателя “Jack Underhill”.

**select** **count**(**distinct** groupsstudents.StudentId)

**from** groupsstudents, groupsacademy, groupslectures, lectures, teachers

**where** groupsstudents.GroupId = groupsacademy.GroupId

**and** groupsacademy.GroupId = groupslectures.GroupId

**and** groupslectures.LectureId = lectures.LectureId

**and** lectures.TeacherId = teachers.TeacherId

**and** teachers.Name = 'Jack' **and** teachers.Surname = 'Underhill';

6. Вывести среднюю ставку преподавателей факультета “Computer Science”.

**select** **avg**(teachers.Salary)

**from** teachers, lectures, groupslectures, groupsacademy, departments, faculties

**where** teachers.TeacherId = lectures.TeacherId

**and** lectures.LectureId = groupslectures.LectureId

**and** groupslectures.GroupId = groupsacademy.GroupId

**and** groupsacademy.departmentid = departments.DepartmentId

**and** departments.FacultyId = faculties.FacultieId

**and** faculties.Name = 'Computer Science';

7. Вывести минимальное и максимальное количество студентов среди всех групп.

**select** **count**(StudentId) **as** *CountStudent* **from** groupsstudents **group** **by** GroupId **order** **by** *CountStudent* **asc** **limit** 1

**union**

(**select** **count**(StudentId) **as** *CountStudent* **from** groupsstudents **group** **by** GroupId **order** **by** *CountStudent* **desc** **limit** 1);

8. Вывести средний фонд финансирования кафедр.

**select** **avg**(Financing) from departments;

9. Вывести полные имена преподавателей и количество читаемых ими дисциплин.

**select** teachers.Name, teachers.Surname, **count**(**distinct** lectures.SubjectId)

**from** teachers, lectures

**where** teachers.TeacherId = lectures.TeacherId

**group** **by** teachers.Name, teachers.Surname;

10. Вывести количество лекций в каждый день недели.

**select** **dayofweek**(DateLecture), **count**(\*) **from** lectures

**group** **by** **dayofweek**(DateLecture);

11. Вывести номера аудиторий и количество кафедр, чьи лекции в них читаются.

**select** lectures.LectureRoom, **count**(**distinct** departments.DepartmentId)

**from** lectures, groupslectures, groupsacademy, departments

**where** lectures.LectureId = groupslectures.LectureId

**and** groupslectures.GroupId = groupsacademy.GroupId

**and** groupsacademy.DepartmentId = departments.DepartmentId

**group** **by** lectures.LectureRoom;

12. Вывести названия факультетов и количество дисциплин, которые на них читаются.

**select** faculties.Name, **count**(**distinct** lectures.SubjectId)

**from** lectures, groupslectures, groupsacademy, departments, faculties

**where** lectures.LectureId = groupslectures.LectureId

**and** groupslectures.GroupId = groupsacademy.GroupId

**and** groupsacademy.DepartmentId = departments.DepartmentId

**and** departments.FacultyId = faculties.FacultieId

**group** **by** faculties.Name;

13. Вывести количество лекций для каждой пары преподаватель-аудитория.

**select** teachers.Name, teachers.Surname, lectures.LectureRoom, **count**(\*)

**from** lectures, teachers

**where** lectures.TeacherId = teachers.TeacherId

**group** **by** teachers.Name, teachers.Surname, lectures.LectureRoom;

**Задание 2**

1. Вывести номера корпусов, если суммарный фонд финансирования расположенных в них кафедр превышает 100000.

**select** Building **from** departments

**group** **by** Building

**having** **sum**(Financing) > 100000;

2. Вывести названия групп 5-ого курса кафедры "Software Development”, которые имеют более 10 пар в первую неделю.

**select** groupsacademy.Name **from** groupsacademy, departments, groupslectures, lectures

**where** groupsacademy.YearGroup = 5

**and** groupsacademy.departmentid = departments.DepartmentId

**and** departments.Name = 'Software Development'

**and** groupsacademy.GroupId = groupslectures.GroupId

**and** groupslectures.LectureId = lectures.LectureId

**and** **day**(lectures.DateLecture) <= 7

**group** **by** groupsacademy.Name

**having** **count**(groupslectures.GroupsLectureId) > 10;

3. Вывести названия групп, имеющих рейтинг(средний рейтинг всех студентов группы) больше, чем рейтинг группы “D221”.

**select** groupsacademy.Name, **sum**(students.Rating) **from** groupsacademy, groupsstudents, students

**where** groupsacademy.GroupId = groupsstudents.GroupId

**and** groupsstudents.StudentId = students.StudentId

**group** **by** groupsacademy.Name

**having** **sum**(students.Rating) > (

**select** **count**(students.Rating) **from** groupsacademy, groupsstudents, students

**where** groupsacademy.GroupId = groupsstudents.GroupId

**and** groupsstudents.StudentId = students.StudentId

**and** groupsacademy.Name = 'D221'

**group** **by** groupsacademy.Name);

4. Вывести фамилии и имена преподавателей, ставка которых выше средней ставки профессоров.

**select** Surname, Name, Salary **from** teachers

**where** Salary > (**select** **avg**(Salary) **from** teachers);

5. Вывести названия групп, у которых больше одного куратора.

**select** groupsacademy.Name, **count**(groupscurators.GroupId) **from** groupsacademy

**join** groupscurators **on** groupsacademy.GroupId = groupscurators.GroupId

**group** **by** groupsacademy.Name

**having** **count**(groupscurators.GroupId) > 1;

6. Вывести названия групп, имеющих рейтинг(средний рейтинг всех студентов группы) меньше, чем минимальный рейтинг групп 5-го курса.

**select** groupsacademy.Name, **sum**(students.Rating) **from** groupsacademy, groupsstudents, students

**where** groupsacademy.GroupId = groupsstudents.GroupId

**and** groupsstudents.StudentId = students.StudentId

**group** **by** groupsacademy.Name

**having** **sum**(students.Rating) > (

**select** **sum**(students.Rating) **from** groupsacademy, groupsstudents, students

**where** groupsacademy.GroupId = groupsstudents.GroupId

**and** groupsstudents.StudentId = students.StudentId

**and** groupsacademy.YearGroup = 5

**group** **by** groupsacademy.Name

**order** **by** **sum**(students.Rating) **desc**

**limit** 1);

7. Вывести названия факультетов, суммарный фонд финансирования кафедр которых больше суммарного фонда финансирования кафедр факультета “Computer Science”.

**select** faculties.Name, **sum**(departments.Financing) **from** faculties

**join** departments

**on** faculties.FacultieId = departments.FacultyId

**group** **by** faculties.Name

**having** **sum**(departments.Financing) > (

**select** faculties.Name, **sum**(departments.Financing) **from** faculties, departments

**where** faculties.FacultieId = departments.FacultyId

**and** faculties.Name = 'Computer Science'

**group** **by** faculties.Name);

8. Вывести названия дисциплин и полные имена преподавателей, читающих наибольшее количество лекций по ним.

**select** subjects.Name, teachers.Name, teachers.Surname

**from** lectures, subjects, teachers

**where** lectures.SubjectId = subjects.SubjectId

**and** lectures.TeacherId = teachers.TeacherId

**group** **by** subjects.SubjectId, teachers.TeacherId

**order** **by** **count**(lectures.TeacherId) **asc**

**limit** 1;

9. Вывести название дисциплины, по которому читается меньше всего лекций.

**select** subjects.Name, **count**(lectures.SubjectId)

**from** lectures **join** subjects

**on** lectures.SubjectId = subjects.SubjectId

**group** **by** subjects.Name

**order** **by** **count**(lectures.SubjectId) **desc**

**limit** 1;

10. Вывести количество студентов и читаемых дисциплин на кафедре "Software Development”.

**select** **count**(lectures.SubjectId), **count**(groupsstudents.StudentId)

**from** lectures, groupslectures, groupsstudents, groupsacademy, departments

**where** lectures.LectureId = groupslectures.LectureId

**and** groupslectures.GroupId = groupsacademy.GroupId

**and** groupsacademy.departmentid = departments.DepartmentId

**and** departments.Name = 'Software Development'

**group** **by** groupsacademy.GroupId;

**Кураторы (Curators)**

* Идентификатор (Id). Уникальный идентификатор куратора.
* Тип данных – int.
* Авто приращение.
* Не может содержать null-значения.
* Первичный ключ.
* Название (Name). Имя куратора.
* Тип данных – nvarchar(max).
* Не может содержать null-значения.
* Не может быть пустым.
* Фамилия (Surname). Фамилия куратора.
* Тип данных – nvarchar(max).
* Не может содержать null-значения.
* Не может быть пустым.

**create** **table** Curators

(

CuratorId **int** **not** **null** **auto\_increment**,

Name **varchar**(300) **not** **null**,

Surname **varchar**(300) **not** **null**,

**primary** **key** (CuratorId)

);

**Кафедры (Departments)**

* Идентификатор (Id). Уникальный идентификатор кафедры.
* Тип данных – int.
* Авто приращение.
* Не может содержать null-значения.
* Первичный ключ.
* Корпус (Building). Номер корпуса, в котором располагается кафедра.
* Тип данных – int.
* Не может содержать null-значения.
* Должно быть в диапазоне от 1 до 5.
* Финансирование (Financing). Фонд финансирования кафедры.
* Тип данных – money.
* Не может содержать null-значения.
* Не может быть меньше 0.
* Значение по умолчанию - 0.
* Название (Name). Название кафедры.
* Тип данных – nvarchar(100).
* Не может содержать null-значения.
* Не может быть пустым.
* Должно быть уникальным.
* Идентификатор факультета (FacultyId). Факультет, в состав которого входит кафедра.
* Тип данных – int.
* Не может содержать null-значения.
* Внешний ключ.

**create** **table** Departments

(

DepartmentId **int** **not** **null** **auto\_increment**,

Building **int** **check**(Building >= 1) **check**(Building <= 5) **not** **null**,

Financing **decimal**(10,2) **check**(Financing >= 0) **not** **null** **default** 0,

Name **varchar**(100) **not** **null** **unique**,

FacultyId **int** **not** **null**,

**primary** **key** (DepartmentId),

**foreign** **key** (FacultyId) **references** faculties(FacultieId)

);

**Факультеты (Faculties)**

* Идентификатор (Id). Уникальный идентификатор факультета.
* Тип данных – int.
* Авто приращение.
* Не может содержать null-значения.
* Первичный ключ.
* Название (Name). Название факультета.
* Тип данных – nvarchar(100).
* Не может содержать null-значения.
* Не может быть пустым.
* Должно быть уникальным.

**create table** Faculties

(

FacultieId **int** **not** **null** **auto\_increment**,

Name **varchar**(100) **not** **null unique**,

**primary** **key** (FacultieIId)

);

**Группы (Groups)**

* Идентификатор (Id). Уникальный идентификатор группы.
* Тип данных – int.
* Авто приращение.
* Не может содержать null-значения.
* Первичный ключ.
* Название (Name). Название группы.
* Тип данных – nvarchar(10).
* Не может содержать null-значения.
* Не может быть пустым.
* Должно быть уникальным.
* Курс (Year). Курс (год) на котором обучается группа
* Тип данных – int.
* Не может содержать null-значения.
* Должно быть в диапазоне от 1 до 5.
* Идентификатор кафедры (DepartmentId). Кафедра, в состав которой входит группа.
* Тип данных – int.
* Не может содержать null-значения.
* Внешний ключ.

**create** **table** GroupsAcademy

(

GroupId **int** **not** **null** **auto\_increment**,

Name **varchar**(10) **not** **null** **unique**,

YearGroup **int** **not** **null** **check**(YearGroup <= 1 **and** YearGroup >= 5),

DepartmentId **int** **not** **null**,

**primary** **key** (GroupId),

**foreign** **key** (DepartmentId) **references** departments(DepartmentId)

);

**ALTER** **TABLE** academy.groupsacademy **add** **foreign** **key** (DepartmentId) **references** departments(DepartmentId);

**Группы и кураторы (GroupsCurators)**

* Идентификатор (Id). Уникальный идентификатор группы и куратора.
* Тип данных – int.
* Авто приращение.
* Не может содержать null-значения.
* Первичный ключ.
* Идентификатор куратора (CuratorId). Куратор.
* Тип данных – int.
* Не может содержать null-значения.
* Внешний ключ.
* Идентификатор группы (GroupId). Группа.
* Тип данных – int.
* Не может содержать null-значения.
* Внешний ключ.

**create** **table** GroupsCurators

(

GroupCuratorId **int** **not** **null** **auto\_increment**,

CuratorId **int** **not** **null**,

GroupId **int** **not** **null**,

**primary** **key** (GroupCuratorId),

**foreign** **key** (CuratorId) **references** curators(CuratorId),

**foreign** **key** (GroupId) **references** groupsacademy(GroupId)

);

**Группы и лекции (GroupsLectures)**

* Идентификатор (Id). Уникальный идентификатор группы и лекции.
* Тип данных – int.
* Авто приращение.
* Не может содержать null-значения.
* Первичный ключ.
* Идентификатор группы (GroupId). Группа.
* Тип данных – int.
* Не может содержать null-значения.
* Внешний ключ.
* Идентификатор лекции (LectureId). Лекция.
* Тип данных – int.
* Не может содержать null-значения.
* Внешний ключ.

**create** **table** GroupsLectures

(

GroupsLectureId **int** **not** **null** **auto\_increment**,

GroupId **int** **not** **null**,

LectureId **int** **not** **null**,

**primary** **key** (GroupsLectureId),

**foreign** **key** (GroupId) **references** groupsacademy(GroupId),

**foreign** **key** (LectureId) **references** lectures(LectureId)

);

**Группы и студенты (GroupsStudents)**

* Идентификатор (Id). Уникальный идентификатор группы и студенты.
* Тип данных – int.
* Авто приращение.
* Не может содержать null-значения.
* Первичный ключ.
* Идентификатор группы (GroupId). Группа.
* Тип данных – int.
* Не может содержать null-значения.
* Внешний ключ.
* Идентификатор студента (StudentId). Студент.
* Тип данных – int.
* Не может содержать null-значения.
* Внешний ключ.

**create** **table** GroupsStudents

(

GroupStudentId **int** **not** **null** **auto\_increment**,

GroupId **int** **not** **null**,

StudentId **int** **not** **null**,

**primary** **key** (GroupStudentId),

**foreign** **key** (GroupId) **references** groupsacademy(GroupId),

**foreign** **key** (StudentId) **references** students(StudentId)

);

**Лекции (Lectures)**

* Идентификатор (Id). Уникальный идентификатор лекции.
* Тип данных – int.
* Авто приращение.
* Не может содержать null-значения.
* Первичный ключ.
* Дата проведения (Date). Дата проведения лекции.
* Тип данных – date.
* Не может содержать null-значения.
* Не может быть больше текущей даты.
* Идентификатор дисциплины (SubjectId). Дисциплина, по которой читается лекция.
* Тип данных – int.
* Не может содержать null-значения.
* Внешний ключ.
* Идентификатор преподавателя (TeacherId). Преподаватель, который читает лекцию.
* Тип данных – int.
* Не может содержать null-значения.
* Внешний ключ.

**create** **table** Lectures

(

LectureId **int** **not** **null** **auto\_increment**,

DateLecture **date** **check**(DateLecture <= now()) **not** **null**,

SubjectId **int** **not** **null**,

TeacherId **int** **not** **null**,

**primary** **key** (LectureId),

**foreign** **key** (SubjectId) **references** subjects(SubjectId),

**foreign** **key** (TeacherId) **references** teachers(TeacherId)

);

**Студенты (Students)**

* Идентификатор (Id). Уникальный идентификатор студенты.
* Тип данных – int.
* Авто приращение.
* Не может содержать null-значения.
* Первичный ключ.
* Название (Name). Имя студента.
* Тип данных – nvarchar(max).
* Не может содержать null-значения.
* Не может быть пустым.
* Рейтинг (Rating). Рейтинг студента.
* Тип данных – int.
* Не может содержать null-значения.
* Должно быть в диапазоне от 0 до 5.
* Фамилия (Surname). Фамилия студента.
* Тип данных – nvarchar(max).
* Не может содержать null-значения.
* Не может быть пустым.

**create** **table** Students

(

StudentId **int** **not** **null** **auto\_increment**,

Name **varchar**(300) **not** **null**,

Rating **int** **check**(Rating <= 0 **and** Rating >= 5) **not** **null**,

Surname **varchar**(300) **not** **null**,

**primary** **key** (StudentId)

);

**Дисциплины (Subjects)**

* Идентификатор (Id). Уникальный идентификатор дисциплины.
* Тип данных – int.
* Авто приращение.
* Не может содержать null-значения.
* Первичный ключ.
* Название (Name). Название дисциплины.
* Тип данных – nvarchar(max).
* Не может содержать null-значения.
* Не может быть пустым.
* Должно быть уникальным.

**create** **table** Subjects

(

SubjectId **int** **not** **null** **auto\_increment**,

Name **varchar**(300) **not** **null** **unique**,

**primary** **key** (SubjectId)

);

**Преподаватели (Teachers)**

* Идентификатор (Id). Уникальный идентификатор преподавателя.
* Тип данных – int.
* Авто приращение.
* Не может содержать null-значения.
* Первичный ключ.
* Профессор (IsProfessor). Является ли преподаватель профессором.
* Тип данных – bit.
* Не может содержать null-значения.
* Значение по умолчанию - 0.
* Название (Name). Имя преподавателя.
* Тип данных – nvarchar(max).
* Не может содержать null-значения.
* Не может быть пустым.
* Ставка (Salary). Ставка преподавателя.
* Тип данных – money.
* Не может содержать null-значения.
* Не может быть меньше либо равно 0.
* Фамилия (Surname). Фамилия преподавателя.
* Тип данных – nvarchar(max).
* Не может содержать null-значения.
* Не может быть пустым.

**create** **table** Teachers

(

TeacherId **int** **not** **null** **auto\_increment**,

IsProfessor **bit** **not** **null** **default** 0,

Name **varchar**(300) **not** **null**,

Salary **decimal**(10,2) **check**(Salary > 0) **not** **null**,

Surname **varchar**(300) **not** **null**,

**primary** **key** (TeacherId)

);

