

SQL



Многотабличные базы данных

Создать БД	<code>CREATE DATABASE teststep;</code>
Изменить наименование БД	<code>ALTER DATABASE teststep MODIFY NAME = teststep2;</code>
Удалить БД	<code>DROP DATABASE teststep;</code>
Создать таблицу	<code>USE teststep CREATE TABLE students (-- PRIMARY KEY = UNIQUE + NOT NULL Id int not null unique identity(1, 1) primary key, Surname nvarchar(max) not null check (Surname <> N''), Grant money not null check (Grant >= 0) default 0.0, StartTime time not null check (StartTime between '08:00' and '18:00'), StartDate date not null check (StartDate <= getdate()) default getdate(), EndDate date not null, check (StartDate < EndDate);</code>
Изменить наименование таблицы	<code>USE teststep EXEC sp_rename 'students', 'Students';</code>
Удалить таблицу	<code>USE teststep DROP TABLE students;</code>
Добавить поле в таблицу	<code>USE teststep ALTER TABLE students ADD letter NVARCHAR(5);</code>
Изменить тип поля в таблице	<code>USE teststep ALTER TABLE students ALTER COLUMN letter NVARCHAR(10);</code>
Удалить поле из таблицы	<code>USE teststep ALTER TABLE students DROP COLUMN letter;</code>
Вставить строку с информацией в таблицу	<code>USE teststep INSERT INTO students (id, fio, groupNumber) VALUES (1, N'Иванов Иван Иванович', N'EKO-11021'), (2, N'Петров Петр Петрович', N'EKO-11022'), (3, N'Сидоров Сидор Сидорович', N'EKO-11021');</code>
Выборка данных из таблицы	<code>USE teststep SELECT * FROM students;</code>
Обновить (редактировать) данные поля в таблице по условию	<code>USE teststep UPDATE students SET groupNumber = N'EKO-11023' WHERE id = 3;</code>
Удалить строку в таблице по условию	<code>USE teststep DELETE FROM students WHERE id = 3;</code>

SQL

Очистить данные таблицы	<pre>USE teststep TRUNCATE TABLE students;</pre>
Выборка данных из таблицы	<pre>USE teststep SELECT * FROM students;</pre>
Выборка данных из таблицы с псевдонимами	<pre>USE teststep SELECT id AS ИД, fio AS ФИО FROM students;</pre>
Выборка уникальных данных из таблицы	<pre>USE teststep SELECT DISTINCT * FROM students;</pre>
Выборка ограниченного количества записей (строк) из таблицы	<pre>USE teststep SELECT TOP 2 * FROM students;</pre>
Выборка данных из таблицы по условию	<pre>USE teststep SELECT * FROM students WHERE id = 1;</pre>
Выборка данных из таблицы по составному условию	<pre>USE teststep SELECT * FROM students WHERE id > 1 AND id < 3; -- AND OR NOT = != > < <= >=</pre>
Выборка данных из таблицы по условию вхождения в множество	<pre>USE teststep SELECT * FROM students WHERE id IN (1); -- IN NOT IN</pre>
Выборка данных из таблицы по условию вхождения в диапазон	<pre>USE teststep SELECT * FROM students WHERE id BETWEEN 1 AND 3;</pre>
Выборка данных из таблицы по условию вхождения в фильтр	<pre>USE teststep SELECT * FROM students WHERE fio LIKE N'%Иван%'; -- % _</pre>
Выборка данных из таблицы по условию вхождения в фильтр регулярного выражения	<pre>USE teststep SELECT * FROM students WHERE id LIKE '[1-2]'; -- [abc] [!abc] [начало-конец]</pre>
Выборка данных из таблицы с сортировкой	<pre>USE teststep SELECT * FROM students ORDER BY id DESC; -- ASC DESC</pre>

SQL

При использовании однотабличных баз данных имеют место три специфические проблемы, связанные с обновлением, удалением и вставкой данных. Осуществление указанных операций с таблицами базы данных может привести к противоречивости хранящихся в них данных, что в целом отрицательно скажется на работе с этой базой данных. Такого рода операции называются аномалиями, то есть тем, что является отклонением от нормы.

Name	BirthDate	Department	Phone	Group	Subject
Sophia Nelson	1984-12-08	Software development	32-12	31PPS11	C#
Emma Kirk	1973-05-12	Mathematics	55-34	32PR31	Discrete Math
Henry MacAlister	1975-02-17	Software development	32-12	30PR11	SQL Server
Michael Cooper	1978-11-23	Software development	32-12	29PR21	ADO.NET
Daniel Williams	1979-07-30	Cybersecurity	37-65	32PPS11	ITE1
Sophia Nelson	1984-12-08	Software development	32-12	30PR11	JavaScript
Daniel Williams	1979-07-30	Cybersecurity	37-65	32PPS11	WIN10

Как вы заметили, в данной таблице хранятся сведения о кафедрах, преподавателях, предметах и группах, в которых преподаются указанные предметы. Аномалия обновления связана с избыточность данных, хранимых в соответствующей таблице. Избыточность данных является причиной того, что в процессе обновления информации в таблице часть дублируемых данных не будут изменены, что приведет к противоречивости хранимой информации. Существует два вида избыточности: явная и неявная. Явная избыточность выражается в **дублировании одинаковой информации**, например, в таблице Teachers некоторые данные о преподавателе Sophia Nelson повторяются несколько раз. В том случае если у нее изменится фамилия на Davies (она выйдет замуж), то для того чтобы избежать противоречивости данных в таблице, необходимо будет вносить изменения в каждую запись об этом преподавателе.

Name	BirthDate	Department	Phone	Group	Subject
<u>Sophia Davies</u>	1984-12-08	Software development	32-12	31PPS11	C#
Emma Kirk	1973-05-12	Mathematics	55-34	32PR31	Discrete Math
Henry MacAlister	1975-02-17	Software development	32-12	30PR11	SQL Server
Michael Cooper	1978-11-23	Software development	32-12	29PR21	ADO.NET
Daniel Williams	1979-07-30	Cybersecurity	37-65	32PPS11	ITE1
<u>Sophia Nelson</u>	1984-12-08	Software development	32-12	30PR11	JavaScript
Daniel Williams	1979-07-30	Cybersecurity	37-65	32PPS11	WIN10

Неявная избыточность проявляется в виде неявной зависимости между различными записями одной таблицы, например, номер телефона кафедры Software development повторяется в сведениях о нескольких преподавателях. Допустим, если в случае изменения на кафедре номера телефона (на 48–22) мы внесем изменения только в запись о преподавателе Henry MacAlister, то при последующей работе с таблицей Teachers мы, фактически, не сможем определить правильный номер телефона кафедры, так как в записях о преподавателях Sophia Nelson и Michael Cooper он останется прежним.

SQL

Name	BirthDate	Department	Phone	Group	Subject
Sophia Nelson	1984-12-08	Software development	32-12	31PPS11	C#
Emma Kirk	1973-05-12	Mathematics	55-34	32PR31	Discrete Math
Henry MacAlister	1975-02-17	Software development	48-22	30PR11	SQL Server
Michael Cooper	1978-11-23	Software development	32-12	29PR21	ADO.NET
Daniel Williams	1979-07-30	Cybersecurity	37-65	32PPS11	ITE1
Sophia Nelson	1984-12-08	Software development	32-12	30PR11	JavaScript
Daniel Williams	1979-07-30	Cybersecurity	37-65	32PPS11	WIN10

Аномалия вставки проявляется в тех случаях, когда существует необходимость поместить в таблицу запись, у которой **отсутствует часть информации**. Отсутствие некоторых данных в таблице, в дальнейшем, может послужить причиной получения неверных результатов при выполнении запросов к базе данных. Это не является критичным если поле с отсутствующими данными может иметь неопределенное значение (NULL-значения), однако для полей, обязательных к заполнению, такая ситуация приведет к возникновению ошибки. В нашем примере добавление в таблицу Teachers преподавателя, о котором неизвестны группа и читаемый им предмет невозможно, поэтому мы не сможем «принять на работу» человека пока не выясним эту информацию.

```
SQLQuery1.sql - DE...ESKTOP\Yuriy (54))* X
INSERT INTO Teachers ([Name],BirthDate,Department,Phone,[Group],[Subject])
VALUES ('John Doe', '1974-03-22', 'Mathematics', '55-34', NULL, NULL);
```

110 %

Messages

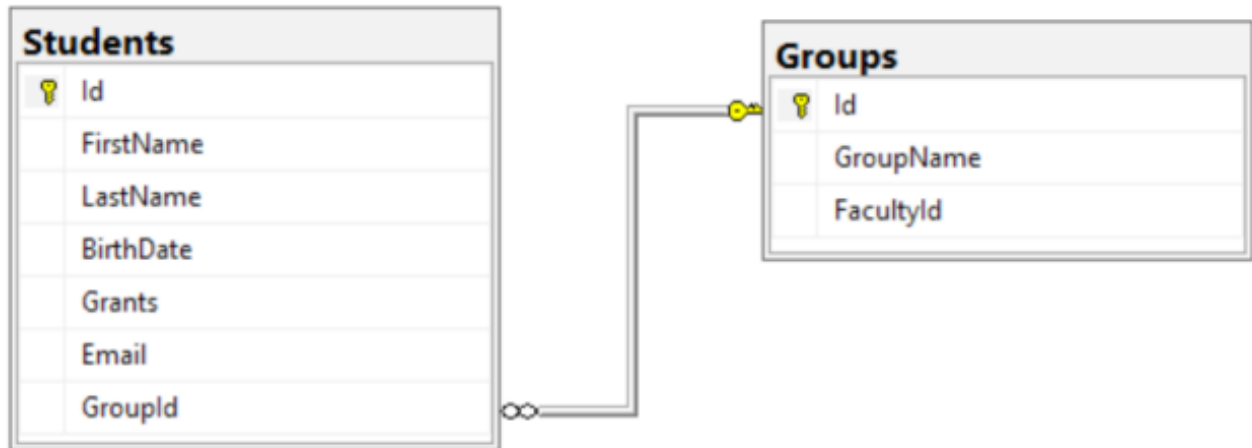
Msg 515, Level 16, State 2, Line 1
Cannot insert the value NULL into column 'Group', table 'University.dbo.Teachers';
column does not allow nulls. INSERT fails.
The statement has been terminated.

Аномалия удаления связана с **уникальностью информации** в определенных записях, при удалении которых эти данные будут утеряны. Например, в нашем случае если преподаватель Emma Kirk уволиться с работы, и мы удалим из таблицы Teachers запись содержащую информацию о ней, то это приведет к потере данных о предмете Discrete Math, кафедре Mathematics и группе 32PR31, потому что информация о них содержится только в этой записи.

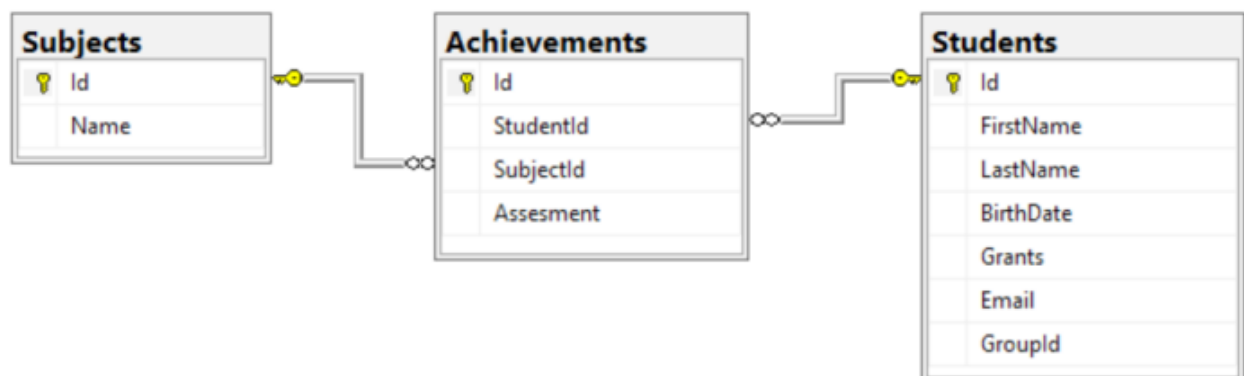
Name	BirthDate	Department	Phone	Group	Subject
Sophia Nelson	1984-12-08	Software development	32-12	31PPS11	C#
Emma Kirk	1973-05-12	Mathematics	55-34	32PR31	Discrete Math
Henry MacAlister	1975-02-17	Software development	32-12	30PR11	SQL Server
Michael Cooper	1978-11-23	Software development	32-12	29PR21	ADO.NET
Daniel Williams	1979-07-30	Cybersecurity	37-65	32PPS11	ITE1
Sophia Nelson	1984-12-08	Software development	32-12	30PR11	JavaScript
Daniel Williams	1979-07-30	Cybersecurity	37-65	32PPS11	WIN10

SQL

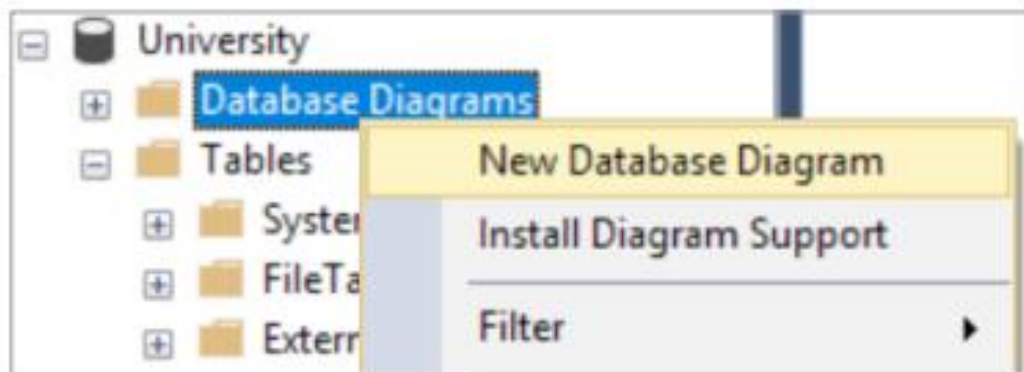
Внешний ключ — это поле таблицы, в котором содержится значение первичного ключа другой таблицы. Именно благодаря соотношению первичного ключа одной таблицы и внешнего ключа другой и формируются связи между таблицами в многотабличной базе данных. В качестве примера приведем часть диаграммы базы данных, на которой в таблицах Students и Groups первичными ключами являются столбцы Id, что визуально отмечено изображениями ключа.



Важно отметить, что в одной таблице допускается наличие нескольких столбцов, являющихся внешними ключами, каждый из которых будет хранить значения первичных ключей различных таблиц текущей базы данных. В качестве примера приведем таблицу Achievements, которая содержит информацию об успеваемости студентов по различным предметам.

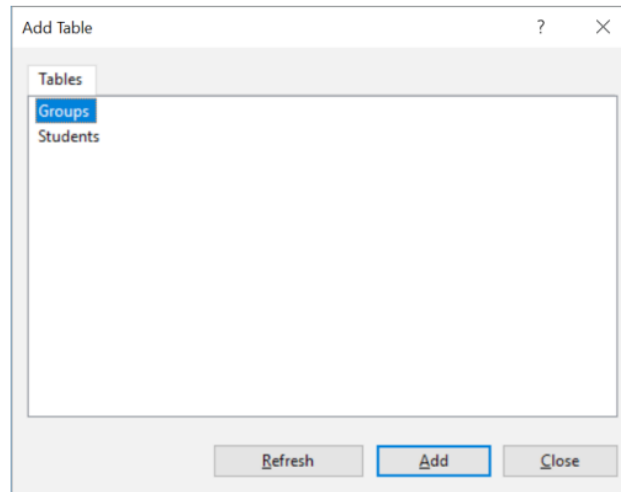


Для создания связей удобно использовать **диаграммы**.

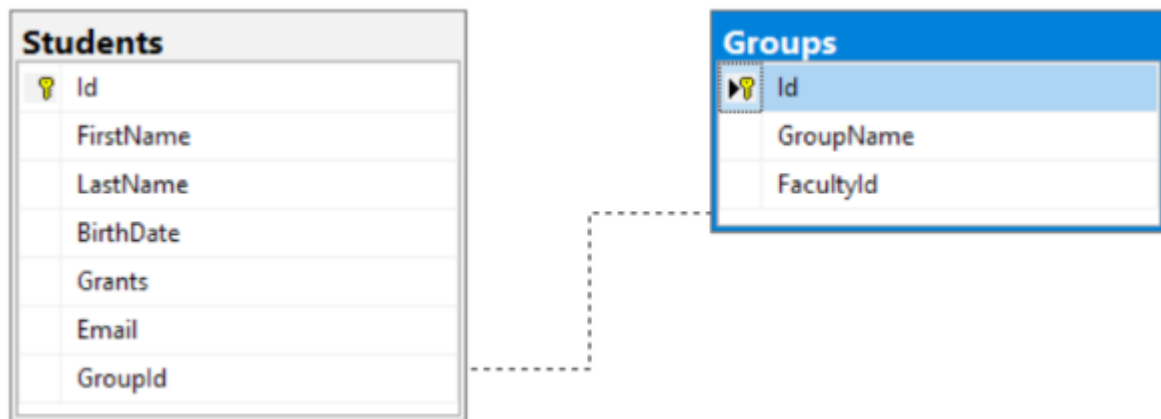


После этого появится форма **Add Table** со списком всех таблиц, которые существуют в текущей базе данных. Для того чтобы добавить их на диаграмму необходимо выбрать в списке нужные таблицы и нажать кнопку **Add**.

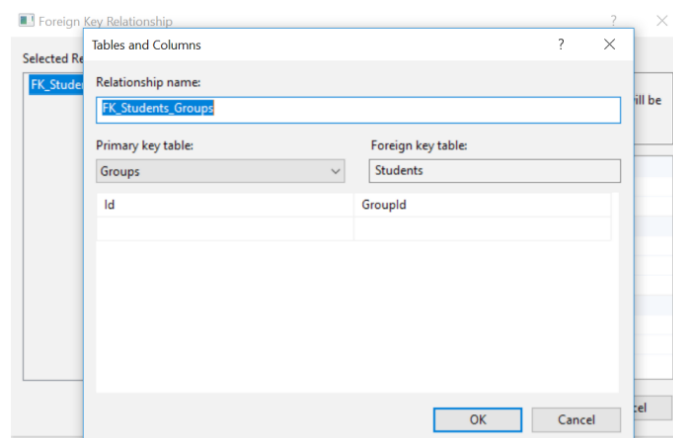
SQL



В результате выполнения этих действий вы получите диаграмму требуемой базы данных. Нам осталось установить связи между таблицами, для того чтобы это сделать необходимо кликнуть левой клавишей мыши по изображению ключа в поле первичного ключа Id таблицы Groups и не отпуская клавиши протянуть мышь на поле внешнего ключа GroupId таблицы Students и отпустить клавишу.



Сразу же после этого вы увидите две формы Tables and Columns и Foreign Key Relationship.



Нормальная форма таблицы - каждое значение в записи должно быть атомарным (неделимым), то есть любой столбец в таблице должен содержать только одно значение для каждой строки; все записи в таблице должны быть разными, даже если в нескольких записях содержится одинаковая информация, то вся запись в целом должна быть уникальной для таблицы.

SQL

Name	BirthDate	Department	Phone	Group	Subject
Sophia Nelson	1984-12-08	Software development	32-12	31PPS11	C#
Emma Kirk	1973-05-12	Mathematics	55-34	32PR31	Discrete Math
Henry MacAlister	1975-02-17	Software development	32-12	30PR11	SQL Server
Michael Cooper	1978-11-23	Software development	32-12	29PR21	ADO.NET
Daniel Williams	1979-07-30	Cybersecurity	37-65	32PPS11	ITE1
Sophia Nelson	1984-12-08	Software development	32-12	30PR11	JavaScript
Daniel Williams	1979-07-30	Cybersecurity	37-65	32PPS11	WIN10

Id	LastName	FirstName	BirthDate	Department	Phone	Group	Subject
1	Nelson	Sophia	1984-12-08	Software development	32-12	31PPS11	C#
2	Kirk	Emma	1973-05-12	Mathematics	55-34	32PR31	Discrete Math
3	MacAlister	Henry	1975-02-17	Software development	32-12	30PR11	SQL Server
4	Cooper	Michael	1978-11-23	Software development	32-12	29PR21	ADO.NET
5	Williams	Daniel	1979-07-30	Cybersecurity	37-65	32PPS11	ITE1
6	Nelson	Sophia	1984-12-08	Software development	32-12	30PR11	JavaScript
7	Williams	Daniel	1979-07-30	Cybersecurity	37-65	32PPS11	WIN10

Teachers

Id	LastName	FirstName	BirthDate	Department	Phone
1	Nelson	Sophia	1984-12-08	Software development	32-12
2	Kirk	Emma	1973-05-12	Mathematics	55-34
3	MacAlister	Henry	1975-02-17	Software development	32-12
4	Cooper	Michael	1978-11-23	Software development	32-12
5	Williams	Daniel	1979-07-30	Cybersecurity	37-65
6	Nelson	Sophia	1984-12-08	Software development	32-12
7	Williams	Daniel	1979-07-30	Cybersecurity	37-65

GroupsSubjects

Id	Group	Subject
1	31PPS11	C#
2	32PR31	Discrete Math
3	30PR11	SQL Server
4	29PR21	ADO.NET
5	32PPS11	ITE1
6	30PR11	JavaScript
7	32PPS11	WIN10

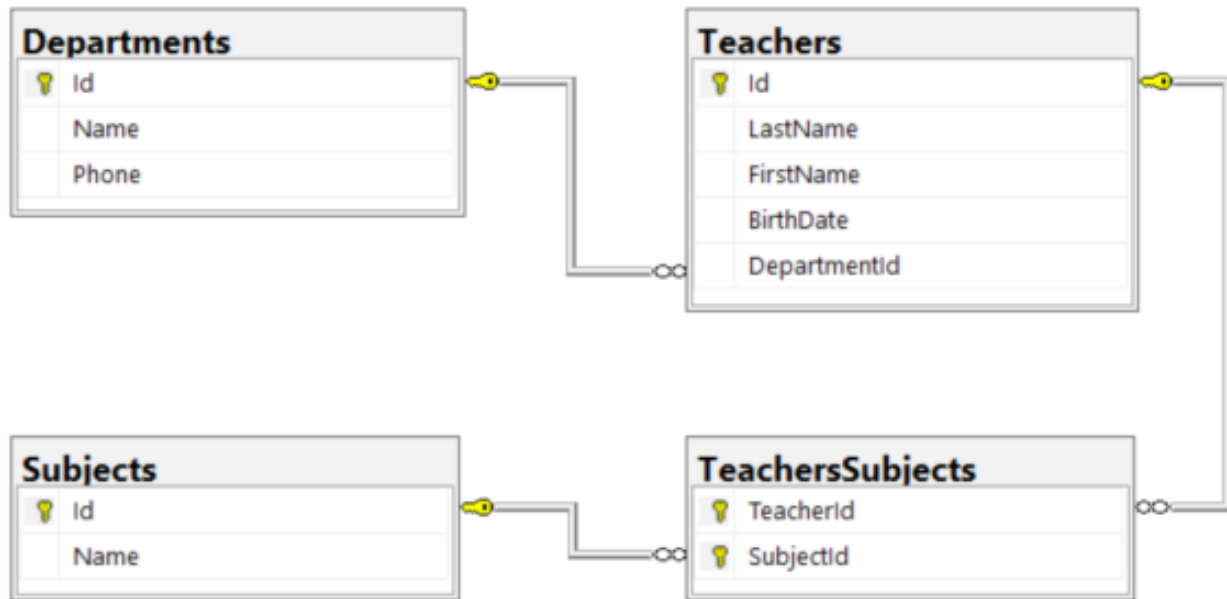
Teachers

Id	LastName	FirstName	BirthDate
1	Nelson	Sophia	1984-12-08
2	Kirk	Emma	1973-05-12
3	MacAlister	Henry	1975-02-17
4	Cooper	Michael	1978-11-23
5	Williams	Daniel	1979-07-30

Departments

Id	Department	Phone
1	Software development	32-12
2	Mathematics	55-34
3	Cybersecurity	37-65

SQL



```
SELECT FirstName + ' ' + LastName AS FullName, Name
FROM Departments, Teachers
WHERE Teachers.DepartmentId = Departments.Id;
```

FullName	Name
Sophia Nelson	Software development
Emma Kirk	Mathematics
Henry MacAlister	Software development
Michael Cooper	Software development
Daniel Williams	Cybersecurity

```
SELECT FirstName + ' ' + LastName AS FullName,
       Name AS SubjectName
FROM Teachers AS T, Subjects AS S,
       TeachersSubjects AS TS
WHERE T.Id=TS.TeacherId AND S.Id=TS.SubjectId;
```

FullName	SubjectName
Sophia Nelson	C#
Sophia Nelson	JavaScript
Emma Kirk	Discrete Math
Henry MacAlister	SQL Server
Michael Cooper	ADO.NET
Daniel Williams	ITE1
Daniel Williams	WIN10

SQL

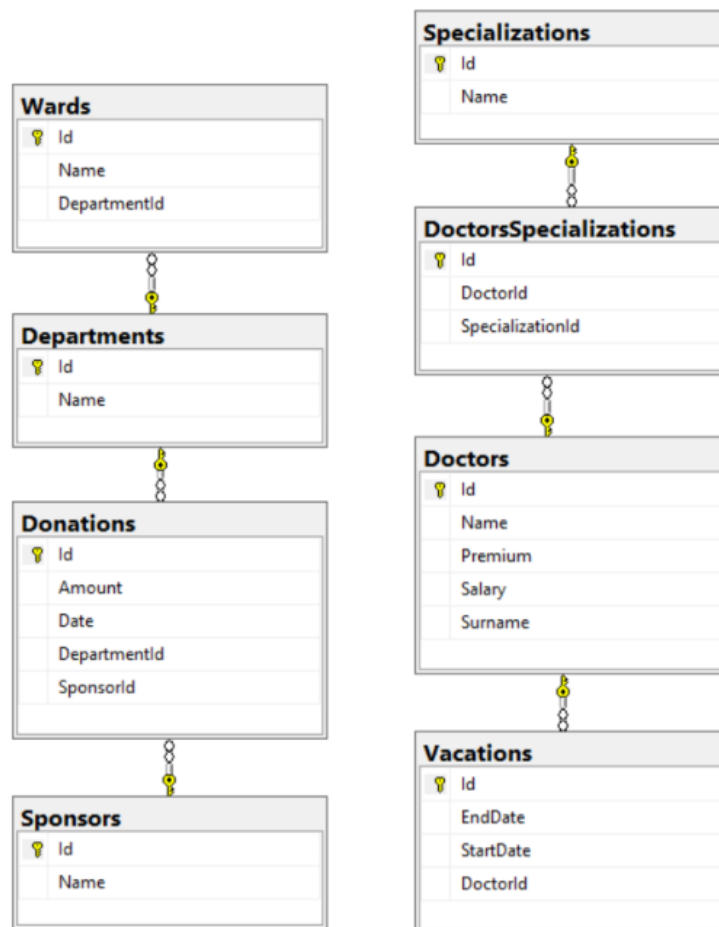
```
SELECT D.Name AS DeptName, S.Name  
      AS SubjectName  
FROM Departments AS D, Teachers AS T, Subjects  
      AS S, TeachersSubjects AS TS  
WHERE D.Id = T.DepartmentId AND T.Id=TS.  
      TeacherId AND S.Id=TS.SubjectId;
```

DeptName	SubjectName
Software development	C#
Software development	JavaScript
Mathematics	Discrete Math
Software development	SQL Server
Software development	ADO.NET
Cybersecurity	ITE1
Cybersecurity	WIN10

Связать таблицы	<code>alter table DoctorsSpecializations add foreign key (DoctorId) references Doctors(Id);</code>
Получить общие данные из нескольких таблиц	<code>SELECT B.Name, W.Name FROM Buildings AS B, Wards AS W WHERE B.Id = W.BuildingId</code>



Практическая работа



1. Отделения (Departments).

■ Идентификатор (Id). Уникальный идентификатор отделения.

- ▷ Тип данных — int.
- ▷ Авто приращение.
- ▷ Не может содержать null-значения.
- ▷ Первичный ключ.
- Название (Name). Название отделения.
- ▷ Тип данных — nvarchar(100).
- ▷ Не может содержать null-значения.
- ▷ Не может быть пустым.
- ▷ Должно быть уникальным.

2. Врачи (Doctors).

■ Идентификатор (Id). Уникальный идентификатор врача.

- ▷ Тип данных — int.
- ▷ Авто приращение.
- ▷ Не может содержать null-значения.
- ▷ Первичный ключ.
- Имя (Name). Имя врача.
- ▷ Тип данных — nvarchar(max).

- ▷ Не может содержать null-значения.
- ▷ Не может быть пустым.
 - Надбавка (Premium). Надбавка врача.
- ▷ Тип данных — money.
- ▷ Не может содержать null-значения.
- ▷ Не может быть меньше 0.
- ▷ Значение по умолчанию — 0.
 - Ставка (Salary). Ставка врача.
- ▷ Тип данных — money.
- ▷ Не может содержать null-значения.
- ▷ Не может быть меньше либо равно 0.
 - Фамилия (Surname). Фамилия врача.
- ▷ Тип данных — nvarchar(max).
- ▷ Не может содержать null-значения.
- ▷ Не может быть пустым.
- 3. Врачи и специализации (DoctorsSpecializations).
 - Идентификатор (Id). Уникальный идентификатор врача и специализации.
- ▷ Тип данных — int.
- ▷ Авто приращение.
- ▷ Не может содержать null-значения.
- ▷ Первичный ключ.
 - Идентификатор врача (DoctorId). Врач.
- ▷ Тип данных — int.
- ▷ Не может содержать null-значения.
- ▷ Внешний ключ.
 - Идентификатор специализации (SpecializationId). Специализация.
- ▷ Тип данных — int.
- ▷ Не может содержать null-значения.
- ▷ Внешний ключ.
- 4. Пожертвования (Donations).
 - Идентификатор (Id). Уникальный идентификатор пожертвования.
- ▷ Тип данных — int.
- ▷ Авто приращение.
- ▷ Не может содержать null-значения.
- ▷ Первичный ключ.
 - Сумма (Amount). Сумма пожертвования.
- ▷ Тип данных — money.
- ▷ Не может содержать null-значения.
- ▷ Не может быть меньше либо равно 0.
 - Дата (Date). Дата пожертвования.
- ▷ Тип данных — date.
- ▷ Не может содержать null-значения.
- ▷ Не может быть больше текущей даты.
- ▷ Значение по умолчанию — текущая дата.
 - Идентификатор отделения (DepartmentId). Отделение, которому было совершено пожертвование.
- ▷ Тип данных — int.
- ▷ Не может содержать null-значения.

- ▷ Внешний ключ.
 - Идентификатор спонсора (SponsorId). Спонсор, совершивший пожертвование.
- ▷ Тип данных — int.
- ▷ Не может содержать null-значения.
- ▷ Внешний ключ.
- 5. Специализации (Specializations).
 - Идентификатор (Id). Уникальный идентификатор специализации.
- ▷ Тип данных — int.
- ▷ Авто приращение.
- ▷ Не может содержать null-значения.
- ▷ Первичный ключ.
 - Название (Name). Название специализации.
- ▷ Тип данных — nvarchar(100).
- ▷ Не может содержать null-значения.
- ▷ Не может быть пустым.
- ▷ Должно быть уникальным.
- 6. Спонсоры (Sponsors).
 - Идентификатор (Id). Уникальный идентификатор спонсора.
- ▷ Тип данных — int.
- ▷ Авто приращение.
- ▷ Не может содержать null-значения.
- ▷ Первичный ключ.
 - Название (Name). Название спонсора.
- ▷ Тип данных — nvarchar(100).
- ▷ Не может содержать null-значения.
- ▷ Не может быть пустым.
- ▷ Должно быть уникальным.
- 7. Отпуска (Vacations).
 - Идентификатор (Id). Уникальный идентификатор отпуска.
- ▷ Тип данных — int.
- ▷ Авто приращение.
- ▷ Не может содержать null-значения.
- ▷ Первичный ключ.
 - Дата завершения (EndDate). Время завершения отпуска.
- ▷ Тип данных — date.
- ▷ Не может содержать null-значения.
- ▷ Должно быть больше даты начала отпуска.
 - Дата начала (StartDate). Время начала отпуска.
- ▷ Тип данных — date.
- ▷ Не может содержать null-значения.
 - Идентификатор врача (DoctorId). Врач, отправившийся в отпуск.
- ▷ Тип данных — int.
- ▷ Не может содержать null-значения.
- ▷ Внешний ключ.
- 8. Палаты (Wards).
 - Идентификатор (Id). Уникальный идентификатор палаты.
- ▷ Тип данных — int.
- ▷ Авто приращение.

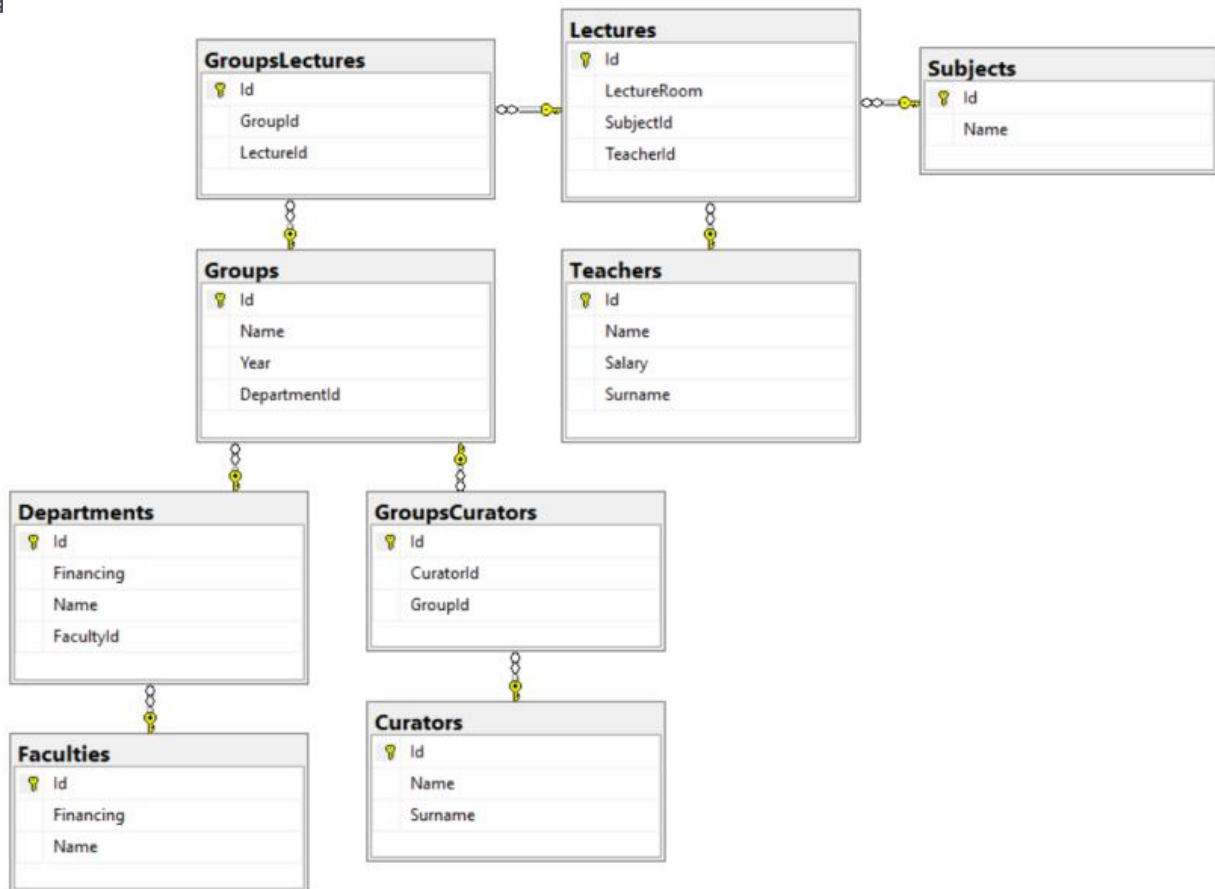
- ▷ Не может содержать null-значения.
- ▷ Первичный ключ.
 - Название (Name). Название палаты.
- ▷ Тип данных — `nvarchar(20)`.
- ▷ Не может содержать null-значения.
- ▷ Не может быть пустым.
- ▷ Должно быть уникальным.
- Идентификатор отделения (DepartmentId). Отделение, в котором располагается палата.
- ▷ Тип данных — `int`.
- ▷ Не может содержать null-значения.
- ▷ Внешний ключ.

Запросы

1. Вывести ФИО врачей (в одном поле с псевдонимом) и их специализации.
2. Вывести фамилии и зарплаты (сумма ставки и надбавки) врачей, которые не находятся в отпуске.
3. Вывести названия палат, которые находятся в отделении "Intensive Treatment".
4. Вывести названия отделений без повторений, которые спонсируются компанией "Umbrella Corporation".
5. Вывести все пожертвования за последний месяц в виде: отделение, спонсор, сумма пожертвования, дата пожертвования.
6. Вывести фамилии врачей с указанием отделений, в которых они проводят обследования. Необходимо учитывать обследования, проводимые только в будние дни.
7. Вывести названия палат и корпуса отделений, в которых проводит обследования врач "Helen Williams".
8. Вывести названия отделений, которые получали пожертвование в размере больше 100000, с указанием их врачей.
9. Вывести названия отделений, в которых есть врачи не получающие надбавки.
10. Вывести названия специализаций, которые используются для лечения заболеваний со степенью тяжести выше 2.
11. Вывести названия отделений и заболеваний, обследования по которым они проводили за последние полгода.
12. Вывести названия отделений и палат, в которых проводились обследования по серьезным заболеваниям.



Домашняя работа



1. Кураторы (Curators).

■ Идентификатор (Id). Уникальный идентификатор куратора.

▷ Тип данных — int.

▷ Авто приращение.

▷ Не может содержать null-значения.

▷ Первичный ключ.

■ Имя (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.

2. Кафедры (Departments).

■ Идентификатор (Id). Уникальный идентификатор кафедры.

▷ Тип данных — int.

▷ Авто приращение.

▷ Не может содержать null-значения.

▷ Первичный ключ.

- 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.

3. Факультеты (Faculties)

- Идентификатор (Id). Уникальный идентификатор факультета.

- ▷ Тип данных — int.
- ▷ Авто приращение.
- ▷ Не может содержать null-значения.
- ▷ Первичный ключ.

- Финансирование (Financing). Фонд финансирования факультета.

- ▷ Тип данных — money.
- ▷ Не может содержать null-значения.
- ▷ Не может быть меньше 0.
- ▷ Значение по умолчанию — 0.

- Название (Name). Название факультета.

- ▷ Тип данных — nvarchar(100).
- ▷ Не может содержать null-значения.
- ▷ Не может быть пустым.
- ▷ Должно быть уникальным.

4. 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.
- 5. 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.
- 6. 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.
- 7. Лекции (Lectures)
 - Идентификатор (Id). Уникальный идентификатор лекции.
 - ▷ Тип данных — int.
 - ▷ Авто приращение.
 - ▷ Не может содержать null-значения.
 - ▷ Первичный ключ.
 - Аудитория (LectureRoom). Аудитория в которой читается лекция.
 - ▷ Тип данных — nvarchar(max).
 - ▷ Не может содержать null-значения.
 - ▷ Не может быть пустым.
 - 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.
- 8. 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(100).
 - ▷ Не может содержать null-значения.
 - ▷ Не может быть пустым.
 - ▷ Должно быть уникальным.
- 9. Преподаватели (Teachers)
 - Идентификатор (Id). Уникальный идентификатор преподавателя.
 - ▷ Тип данных — int.
 - ▷ Авто приращение.
 - ▷ Не может содержать null-значения.
 - ▷ Первичный ключ.
 - Имя (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.
- 1. Вывести все возможные пары строк преподавателей и групп.
- 2. Вывести названия факультетов, фонд финансирования кафедр которых превышает фонд финансирования факультета.
- 3. Вывести фамилии кураторов групп и названия групп, которые они курируют.
- 4. Вывести имена и фамилии преподавателей, которые читают лекции у группы "P107".
- 5. Вывести фамилии преподавателей и названия факультетов, на которых они читают лекции.
- 6. Вывести названия кафедр и названия групп, которые к ним относятся.
- 7. Вывести названия дисциплин, которые читает преподаватель "Samantha Adams".
- 8. Вывести названия кафедр, на которых читается дисциплина "Database Theory".
- 9. Вывести названия групп, которые относятся к факультету "Computer Science".
- 10. Вывести названия групп 5-го курса, а также название факультетов, к которым они относятся.
- 11. Вывести полные имена преподавателей и лекции, которые они читают (названия дисциплин и групп), причем отобразить только те лекции, которые читаются в аудитории "B103".