Задание подготовлено в рамках проекта АНО «Лаборатория модернизации образовательных ресурсов» «Кадровый и учебно-методический ресурс формирования общих компетенций обучающихся по программам СПО», который реализуется с использованием гранта Президента Российской Федерации на развитие гражданского общества, предоставленного Фондом президентских грантов.

**Разработчик**

Ахтямов Ришат Равильевич, ГАПОУ «Самарский государственный колледж»

**Назначение задания**

Оценка результата \ продукта деятельности. Уровень II

МДК.11.01 Технология разработки и защиты баз данных

Тема: Разработка ERD

**Комментарии**

Задание используется на этапе изучения нового материала. Обучающиеся впервые знакомятся с содержанием. Преподаватель может снять последний критерий, если считает необходимым обсудить пример, полученный обучающимися при выполнении задания до того, как они приступят к самостоятельному определению требуемых сущностей.

Задание является переходным от требований уровня I к требованиям уровня II. Преподаватель может воспользоваться более сложной версией задания в рамках этой же темы (см. задание уровня II).

Компания, в которой вы работаете, получила заказ от образовательной организации на разработку системы учета успеваемости студентов. Чтобы другим разработчикам было проще внедрить свои модули в систему, ваш коллега описал структуру базы данных и попросил на всякий случай проверить результат его работы.

Ознакомьтесь с техническим заданием, поступившим от заказчика (источник 1). Внимательно изучите структуру базы данных, разработанную коллегой (источник 2). Ознакомьтесь с правилами составления ER-диаграммы (источник 3).

**Оцените, в какой мере структура позволит сформировать систему учета, отвечающую требованиям заказчика.**

**Запишите результаты оценки в бланк. Если по какому-либо критерию вы даете отрицательную оценку, объясните причину в графе «Комментарии».**

*Бланк*

| Критерий | Оценка, да \ нет | Комментарии |
| --- | --- | --- |
| Сущность «Студенты» определена верно: имеется вся необходимая информация, тип атрибута соответствует стандарту |  |  |
| Сущность «Отделения» определена верно: имеется вся необходимая информация, тип атрибута соответствует стандарту |  |  |
| Сущность «группы»  определена верно: имеется вся необходимая информация, тип атрибута соответствует стандарту |  |  |
| Сущность «Преподаватели» определена верно: имеется вся необходимая информация, тип атрибута соответствует стандарту |  |  |
| Сущность «Предметы» определена верно: имеется вся необходимая информация, тип атрибута соответствует стандарту |  |  |
| Сущность «Оценки» определена верно: имеется вся необходимая информация, тип атрибута соответствует стандарту |  |  |
| Выделены все необходимые по ТЗ сущности |  |  |

***Источник 1***

**Требования заказчика к проектируемой системе учета успеваемости**

Разрабатываемая система должна позволять работать преподавателям в отделениях, используя единую базу с данными, а также внешние сервисы, доступные только при прохождении определенных этапов авторизации.

Система должна не только хранить и показывать информацию, а также позволять анализировать, прогнозировать и планировать действия на основе существующих данных.

Часть данных может содержать изображения. К некоторым типам данным нужно будет добавлять схемы, которые будут создаваться непосредственно в приложении.

*Описание предметной области*

В образовательной организации каждый студент обучается в группе по 25 или 15 человек, каждая группа закреплена за своим отделением. Каждое отделение находится в своем корпусе. Каждое отделение возглавляет сотрудник колледжа, он также может быть преподавателем. За каждой группой закреплен куратор из состава преподавателей.

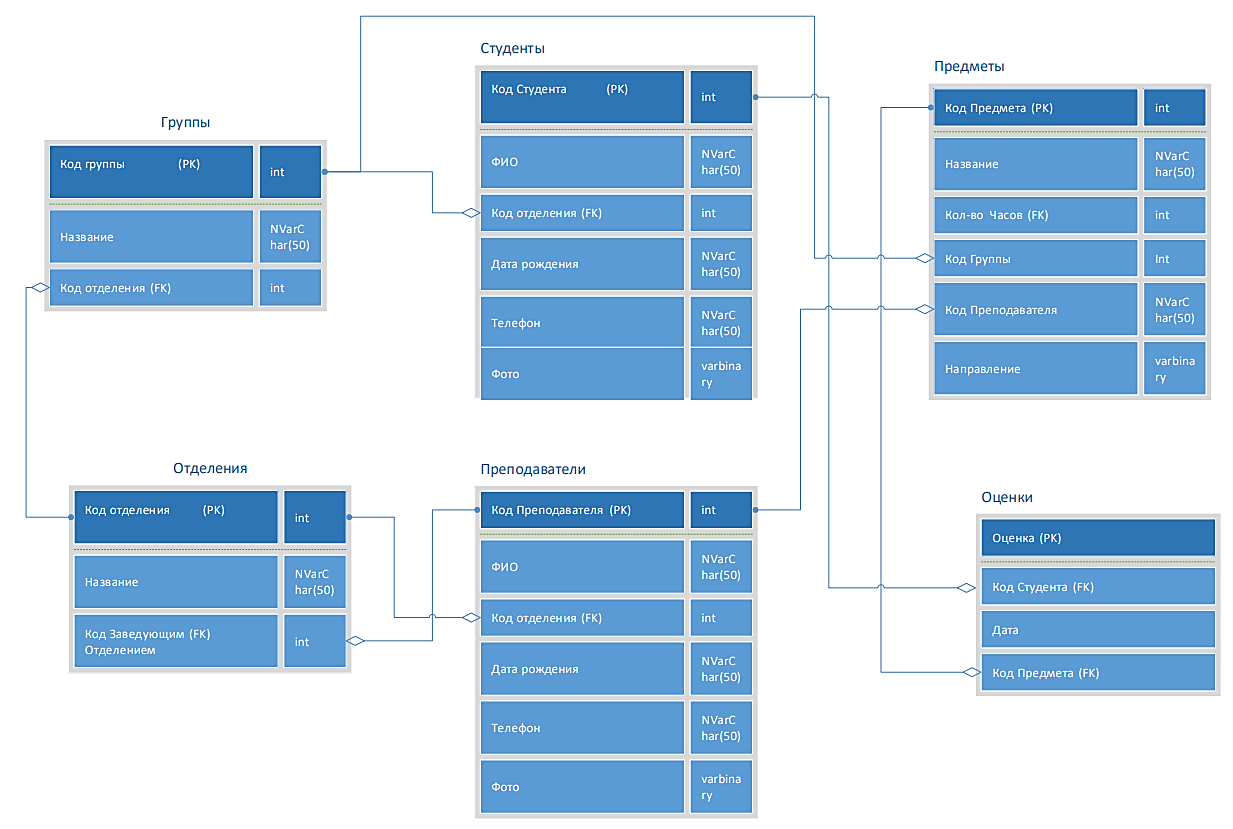
Студент может, получать оценки по предмету. Если студент не присутствовал на занятии, то преподаватель должен отметить это в системе.

Один предмет могут вести как один преподаватель, так и несколько.

На каждый семестр в группе назначается определенное количество предметов. Предметы делятся по категориям, например, общеобразовательный цикл.

Каждый преподаватель работает на определённом отделении. Заведующий отделения должен иметь возможность получить список своих студентов и сотрудников, а так же получать отчет по успеваемости своих студентов.

***Источник 2***



***Источник 3***

**Составление ER-диаграммы**

ER-диаграмма графически представляет сущности (entities) предметной области, свойства (attributes) сущностей и связи (relationship) между ними. В основе ER-диаграмм лежит принцип «рисунок нагляднее текста».

ER-диаграммы делятся на концептуальные и физические. В концептуальных ER-диаграммах не учитываются особенности конкретной базы данных. Впоследствии сущности концептуальных ER-диаграмм становятся таблицами, атрибуты - колонками, а связи реализуются путем миграции ключевых атрибутов родительских сущностей и создания внешних ключей.

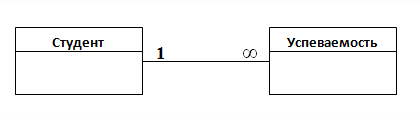
Предположим, что в учебном заведении ведется учет успеваемости студентов. Отсюда следует что:

* студент - явная сущность;
* успеваемость - явная сущность.

На первоначальном этапе моделирования данных информационной системы явно выделены две основные сущности: Студент и Успеваемость.

Критерием успеваемости является наличие отметки о сдачи экзаменов.

Сразу возникает очевидная связь между сущностями: «студент сдаёт несколько экзаменов» и «экзамены сдаются каждым студентом». Явная связь Один-ко-многим. Первый вариант диаграммы выглядит так:



*Примеры связей*

Один-к-одному: план тренировки должен быть составлен по одной заявке / по заявке может быть составлен один план тренировки. Данный тип следует использовать исключительно для связывания различных сущностей (разные сущности должны иметь разные атрибуты).

Один-ко-многим: план тренировки может включать много индивидуальных занятий / индивидуальное занятие должно относиться к одному плану тренировки. Наиболее часто используемый тип связи

Многие-ко-многим: тренер может пройти несколько курсов обучения / курс обучения может быть пройден многими тренерами. Используется исключительно в качестве временного типа. При дальнейшей разработке данная связь заменяется на две связи типа «один-ко-многим» путем добавления промежуточной сущности.

При создании таблицы для всех ее столбцов необходимо указать определенный тип данных. Тип данных определяет, какие значения могут храниться в столбце, сколько они будут занимать места в памяти. Язык T-SQL предоставляет множество различных типов. В зависимости от характера значений все их можно разделить на показанные ниже группы.

*Числовые типы данных*

BIT: хранит значение 0 или 1. Фактически является аналогом булевого типа в языках программирования. Занимает 1 байт.

TINYINT: хранит числа от 0 до 255. Занимает 1 байт. Хорошо подходит для хранения небольших чисел.

SMALLINT: хранит числа от –32 768 до 32 767. Занимает 2 байта

INT: хранит числа от –2 147 483 648 до 2 147 483 647. Занимает 4 байта. Наиболее используемый тип для хранения чисел.

BIGINT: хранит очень большие числа от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807, которые занимают в памяти 8 байт.

DECIMAL: хранит числа c фиксированной точностью. Занимает от 5 до 17 байт в зависимости от количества чисел после запятой. Данный тип может принимать два параметра precision и scale: DECIMAL(precision, scale). Параметр precision представляет максимальное количество цифр, которые может хранить число. Это значение должно находиться в диапазоне от 1 до 38. По умолчанию оно равно 18. Параметр scale представляет максимальное количество цифр, которые может содержать число после запятой. Это значение должно находиться в диапазоне от 0 до значения параметра precision. По умолчанию оно равно 0.

NUMERIC: данный тип аналогичен типу DECIMAL.

SMALLMONEY: хранит дробные значения от -214 748.3648 до 214 748.3647. Предназначено для хранения денежных величин. Занимает 4 байта. Эквивалентен типу DECIMAL(10,4).

MONEY: хранит дробные значения от -922 337 203 685 477.5808 до 922 337 203 685 477.5807. Представляет денежные величины и занимает 8 байт. Эквивалентен типу DECIMAL(19,4).

FLOAT: хранит числа от –1.79E+308 до 1.79E+308. Занимает от 4 до 8 байт в зависимости от дробной части.

REAL: хранит числа от –340E+38 to 3.40E+38. Занимает 4 байта. Эквивалентен типу FLOAT(24).

*Типы данных, представляющие дату и время*

DATE: хранит даты от 0001-01-01 (1 января 0001 года) до 9999-12-31 (31 декабря 9999 года). Занимает 3 байта.

TIME: хранит время в диапазоне от 00:00:00.0000000 до 23:59:59.9999999. Занимает от 3 до 5 байт.

DATETIME: хранит даты и время от 01/01/1753 до 31/12/9999. Занимает 8 байт.

DATETIME2: хранит даты и время в диапазоне от 01/01/0001 00:00:00.0000000 до 31/12/9999 23:59:59.9999999. Занимает от 6 до 8 байт в зависимости от точности времени.

Может иметь форму DATETIME2(n), где n представляет количество цифр от 0 до 7 в дробной части секунд.

SMALLDATETIME: хранит даты и время в диапазоне от 01/01/1900 до 06/06/2079, то есть ближайшие даты. Занимает от 4 байта.

DATETIMEOFFSET: хранит даты и время в диапазоне от 0001-01-01 до 9999-12-31. Сохраняет детальную информацию о времени с точностью до 100 наносекунд. Занимает 10 бай

*Строковые типы данных*

CHAR: хранит строку длиной от 1 до 8 000 символов. На каждый символ выделяет по 1 байту. Не подходит для многих языков, так как хранит символы не в кодировке Unicode.

Количество символов, которое может хранить столбец, передается в скобках. Например, для столбца с типом CHAR(10) будет выделено 10 байт. И если мы сохраним в столбце строку менее 10 символов, то она будет дополнена пробелами.

VARCHAR: хранит строку. На каждый символ выделяется 1 байт. Можно указать конкретную длину для столбца - от 1 до 8 000 символов, например, VARCHAR(10). Если строка должна иметь больше 8000 символов, то задается размер MAX, а на хранение строки может выделяться до 2 Гб: VARCHAR(MAX). Не подходит для многих языков, так как хранит символы не в кодировке Unicode. В отличие от типа CHAR если в столбец с типом VARCHAR(10) будет сохранена строка в 5 символов, то в столце будет сохранено именно пять символов.

NCHAR: хранит строку в кодировке Unicode длиной от 1 до 4 000 символов. На каждый символ выделяется 2 байта. Например, NCHAR(15).

NVARCHAR: хранит строку в кодировке Unicode. На каждый символ выделяется 2 байта. Можно задать конкретный размер от 1 до 4 000 символов. Если строка должна иметь больше 4000 символов, то задается размер MAX, а на хранение строки может выделяться до 2 Гб.

Еще два типа TEXT и NTEXT являются устаревшими и поэтому их не рекомендуется использовать. Вместо них применяются VARCHAR и NVARCHAR соответственно.

Примеры определения строковых столбцов:

1

2

Email VARCHAR(30),

Comment NVARCHAR(MAX)

*Бинарные типы данных*

BINARY: хранит бинарные данные в виде последовательности от 1 до 8 000 байт.

VARBINARY: хранит бинарные данные в виде последовательности от 1 до 8 000 байт, либо до 2^31–1 байт при использовании значения MAX (VARBINARY(MAX)).

Еще один бинарный тип - тип IMAGE является устаревшим, и вместо него рекомендуется применять тип VARBINARY.

*Остальные типы данных*

UNIQUEIDENTIFIER: уникальный идентификатор GUID (по сути строка с уникальным значением), который занимает 16 байт.

TIMESTAMP: некоторое число, которое хранит номер версии строки в таблице. Занимает 8 байт.

CURSOR: представляет набор строк.

HIERARCHYID: представляет позицию в иерархии.

SQL\_VARIANT: может хранить данные любого другого типа данных T-SQL.

XML: хранит документы XML или фрагменты документов XML. Занимает в памяти до 2 Гб.

TABLE: представляет определение таблицы.

GEOGRAPHY: хранит географические данные, такие как широта и долгота.

GEOMETRY: хранит координаты местонахождения на плоскости.

*Использованы материалы источников:* [*https://metanit.com/sql/sqlserver/3.3.php*](https://metanit.com/sql/sqlserver/3.3.php)*;* [*http://inf-teh-lotos.ru/sozdanie-er-diagramm*](http://inf-teh-lotos.ru/sozdanie-er-diagramm)*;*

[*https://nationalteam.worldskills.ru/skills/proektirovanie-er-diagrammy/*](https://nationalteam.worldskills.ru/skills/proektirovanie-er-diagrammy/)*.*

Инструмент проверки

| Критерий | Оценка, да \ нет | Комментарии |
| --- | --- | --- |
| Сущность «Студенты» определена верно: имеется вся необходимая информация, тип атрибута соответствует стандарту | Нет | Ошибка в названии FK (1)  Неверно выбран атрибут Дата (2) |
| Сущность «Отделения» определена верно: имеется вся необходимая информация, тип атрибута соответствует стандарту | Да |  |
| Сущность «группы»  определена верно: имеется вся необходимая информация, тип атрибута соответствует стандарту | Нет | Отсутствует информация о кураторе (3)  Отсутствует информация о количестве студентов (4) |
| Сущность «Преподаватели» определена верно: имеется вся необходимая информация, тип атрибута соответствует стандарту | Нет | Неверно выбран атрибут Дата (5) |
| Сущность «Предметы» определена верно: имеется вся необходимая информация, тип атрибута соответствует стандарту | Нет | Атрибут направление выбран неверно (6).  Нельзя назначить двух и более преподавателей (7) |
| Сущность «Оценки» определена верно: имеется вся необходимая информация, тип атрибута соответствует стандарту | Да |  |
| Выделены все необходимые по ТЗ сущности | Нет | Отсутствует сущность Корпуса (8)  Следует включить сущности (9):  Кабинеты  Специальности  Направления  Список дисциплин  Должности  Расписание звонков  Семестры  Расписание  Экзамены |

*Подсчет баллов*

|  |  |
| --- | --- |
| За каждую верную оценку | 1 балл |
| *Максимально* | *7 баллов* |
| За каждый верный комментарий 1-8 | 1 балл |
| *Максимально* | *8 баллов* |
| За полный и верный комментарий 9 при отсутствии избыточных сущностей | 3 балла |
| *За комментарий 9, содержащий один пропуск или одну избыточную сущность* | *2 балла* |
| *За комментарий 9, включающий 5-7 верных сущностей и не более одной избыточной сущности* | *1 балл* |
| ***Максимальный балл*** | ***18 баллов*** |