



МФТИ

БАЗЫ ДАННЫХ

КУРСОВАЯ РАБОТА

Experiment DataBase

Автор:

Ляликова Ирина

Группа:

911

2020/2021

1 Введение

Данное описание прилагается к проекту, который представляет из себя базу данных, реализованную с использованием PostgreSQL, которая моделирует школьную физическую лабораторию. База данных содержит такие сущности, как школьник, ассистент, оборудование и другие, а также соответствующие им параметры. Основные задачи базы данных — отслеживать проведение работ и выставление оценок за них, а также состав и целостность оборудования.

В описании представлены этапы моделирования (концептуальная, логическая и физическая модели) с подробные пояснениями, а так же примеры использования и некоторые результаты. *Все вещи, описанные ниже основаны на реальных событиях, тем не менее, все совпадения случайны, а школьники вымышлены.*



2 Проектирование

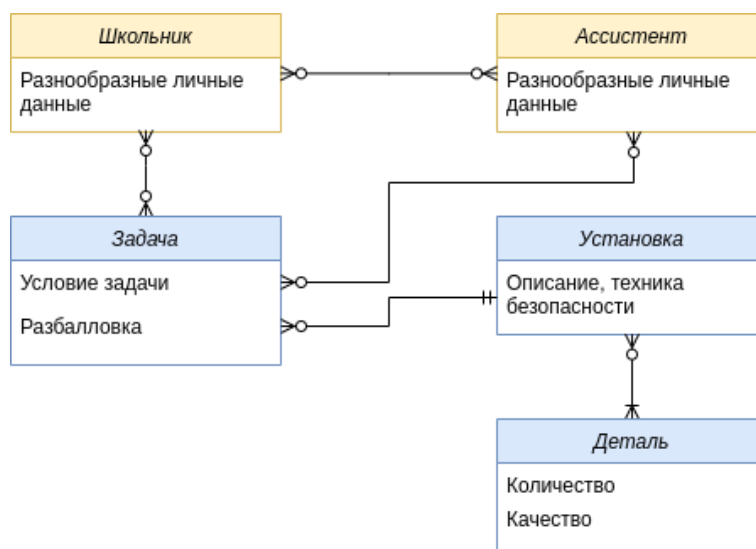
2.1 Концептуальная модель

В базу данных входит пять сущностей:

- *школьник*, который выполняет задания и получает оценки;
- *ассистент* (преподаватель было бы громко сказано), который проводит и проверяет работы за умеренную плату;
- *задача* (или работа), которую необходимо выполнить школьнику;
- *установка*, на которой работа выполняется;
- *деталь*, входящая в состав установки.

Далее представлена диаграмма концептуальной модели. Кроме сущностей, в ней так же присутствуют связи, при этом можно выделить связь *урок*, соединяющую школьника, задачу и ассистента, которой к тому же можно задать свойство даты и оценки, и *состав установки*, определяющую сколько и каких деталей нужно для сборки.

Итак, схема:

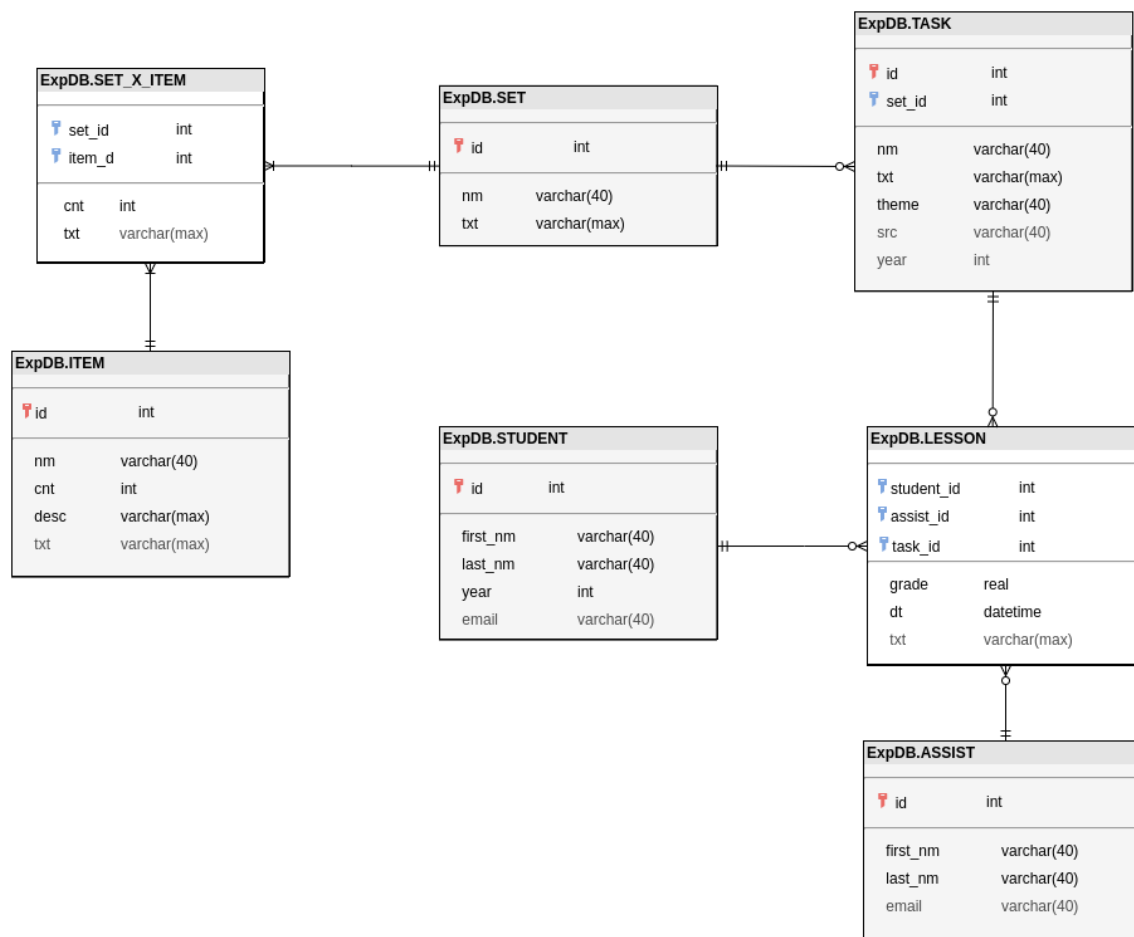


Школьник может выполнять разные работы, работы могут проводить разные ассистенты, у одной задачи одна и только одна установка, зато одна установка может пригодиться для разных работ. То же самое и с деталями: в одну установку их входит много (но не нулевое количество), и каждая деталь может использоваться в разных установках, а может не использоваться вообще нигде, но всё равно оставаться очень важным элементом системы, как например скотч. Так что на этой схеме практически все зависимости типа много-много, так что при проектировании таблиц будет интересно.

2.2 Логическая модель

Теперь нужно отразить упомянутые связи в таблицах. И тут помимо пяти таблиц для сущностей добавятся ещё две для связей. Таблицы построены во второй нормальной форме, так как это более всего подходит для ситуации. Действительно, все поля в смысловых таблицах зависят от ключа, и возможно, друг от друга, но для приведения к третьей нормальной форме потребовалось бы много таблиц, что бы усложнило логику архитектуры и не принесло значимых преимуществ.

Вот что получилось (первичные ключи обозначены красным, а внешние синим):



В репозитории лежат все схемы в форматах PNG и SVG.

2.3 Физическая модель

Здесь приведено подробное описание полей каждой таблицы.

STUDENT/Школьник			
Название	Описание	Тип данных	Ограничение
id	Идентификатор ученика	INTEGER	NOT NULL PRIMARY KEY
first_nm	Имя	VARCHAR(40)	NOT NULL
second_nm	Фамилия	VARCHAR(40)	NOT NULL

year	Класс, в котором школьник учится	INTEGER	NOT NULL PRIMARY KEY
email	Электронная почта	VARCHAR(40)	

ASSIST/Ассистент			
Название	Описание	Тип данных	Ограничение
id	Идентификатор ассистента	INTEGER	NOT NULL PRIMARY KEY
first_nm	Имя	VARCHAR(40)	NOT NULL
last_nm	Фамилия	VARCHAR(40)	NOT NULL
email	Электронная почта	VARCHAR(40)	

TASK/Задача			
Название	Описание	Тип данных	Ограничение
id	Идентификатор задачи	INTEGER	NOT NULL PRIMARY KEY
set_id	Идентификатор установки	INTEGER	NOT NULL FOREIGN KEY
nm	Название	VARCHAR(40)	NOT NULL
desc	Условие задачи	VARCHAR(MAX)	NOT NULL
theme	Раздел физики	VARCHAR(40)	NOT NULL
src	Источник	VARCHAR(40)	
year	Для какого класса задача	INTEGER	

SET/Установка			
Название	Описание	Тип данных	Ограничение
id	Идентификатор установки	INTEGER	NOT NULL PRIMARY KEY
nm	Название	VARCHAR(40)	NOT NULL
desc	Описание и техника безопасности	VARCHAR(MAX)	NOT NULL

ITEM/Деталь			
Название	Описание	Тип данных	Ограничение
id	Идентификатор детали	INTEGER	NOT NULL PRIMARY KEY
nm	Название	VARCHAR(40)	NOT NULL
cnt	Доступное количество	INTEGER	NOT NULL
desc	Описание	VARCHAR(MAX)	NOT NULL
txt	Некоторые комментарии	VARCHAR(MAX)	

LESSON/Занятие			
Название	Описание	Тип данных	Ограничение
student_id	Идентификатор студента	INTEGER	NOT NULL FOREIGN KEY
assist_id	Идентификатор ассистента	INTEGER	NOT NULL FOREIGN KEY
task_id	Идентификатор задачи	INTEGER	NOT NULL FOREIGN KEY
grade	Оценка за задачу	INTEGER	NOT NULL
dt	Дата проведения	TIMESTAMP(0)	NOT NULL
txt	Некоторые комментарии	VARCHAR(MAX)	

SET_X_ITEM/Состав установки			
Название	Описание	Тип данных	Ограничение
set_id	Идентификатор установки	INTEGER	NOT NULL FOREIGN KEY
item_id	Идентификатор детали	INTEGER	NOT NULL FOREIGN KEY
cnt	Сколько таких деталей нужно для такой установки	INTEGER	NOT NULL
txt	Некоторые комментарии	VARCHAR(MAX)	

3 Скрипты

В корневой папке репозитория находятся файлы с рабочими скриптами. В файле `create.sql` содержится код для создания пустой базы данных, а в `insert.sql` данные для заполнения таблиц (90 записей). В `requests.sql` написаны несколько смысловых запросов с использованием подзапросов, объединения таблиц, оконных функций и других радостей, которые как минимум на имеющихся данных работают правильно. В файле `crud.sql` находятся CRUD-запросы к таблицам `Student` и `Assist`. Стоит отметить, что реализация устроена так, что при удалении школьника или ассистента из базы данных уроки, в которых они принимали участие не удаляются каскадно, а заполняются нулевыми значениями. В файле `view.sql` лежат 5 представлений для каждой таблицы-сущности и 2 представления для сложных запросов. В файле `function.cpp` находится хранимая функция, позволяющая найти количество недоступных из-за нехватки оборудования работ, и простой пример её использования.