

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ**

Факультет информационных технологий и программирования

Кафедра компьютерных технологий

ОТЧЁТ

по курсовой работе

по дисциплине «Базы данных»

Тема: «Электронный дневник для летней математической
школы «Спектр»

Выполнил:

студент группы М3437

А. И. Плотников _____
(подпись)

Проверил:

преподаватель

С. В. Савосин _____
(подпись)

Оценка: _____

1. Описание предметной области

База данных разрабатывается для приложения, которое будет использоваться в летних математических школах (ЛМШ) и математических кружках (МК) в качестве электронного дневника. Основная причина разработки – упрощение обработки данных, связанных с рабочим процессом (и не только), в виду того, что на текущий момент данные хранятся в бумажном виде, либо в слабо структурированных таблицах, и требуют много времени для обработки. Поэтому одним из решений является автоматизация хранения и запросов к данным.

Учебный процесс имеет определённый временной период (ВП). По согласованному расписанию ученики приходят на занятия и делают попытки сдать решение задач, при этом они не ограничены в количестве подходов по одной задаче. У каждой задачи в составе одной темы есть дедлайн, после которого её стоимость обнуляется. Так же задача может быть разобрана в классе, после чего она уже не может приниматься.

Темой занятия называется некоторый набор задач, который сформирован на раздаточном листочке с условиями. В этом списке могут быть задачи, которые обязательно будут разобраны на занятии (они изначально не имеют стоимости), а остальные считаются зачётными (и имеют фиксированную стоимость).

Тема задания сама по себе является частью учебного плана, который утверждается для учебной группы старшим преподавателем. Учебная группа – множество учеников и преподавателей, которые непосредственно участвуют в учебном процессе. При этом преподаватель может вести занятия в нескольких учебных группах. У каждой группы выбирается один преподаватель, который считается старшим, он имеет дополнительные обязанности, но их учёт не требуется в базе данных.

Учебные группы, расписание и учебные планы формируются для одного временного периода, в течение которого происходит обучение. После окончания периода обучения - группы, расписание и планы хранятся в архиве.



2. Техническое задание

База данных должна соответствовать следующим требованиям:

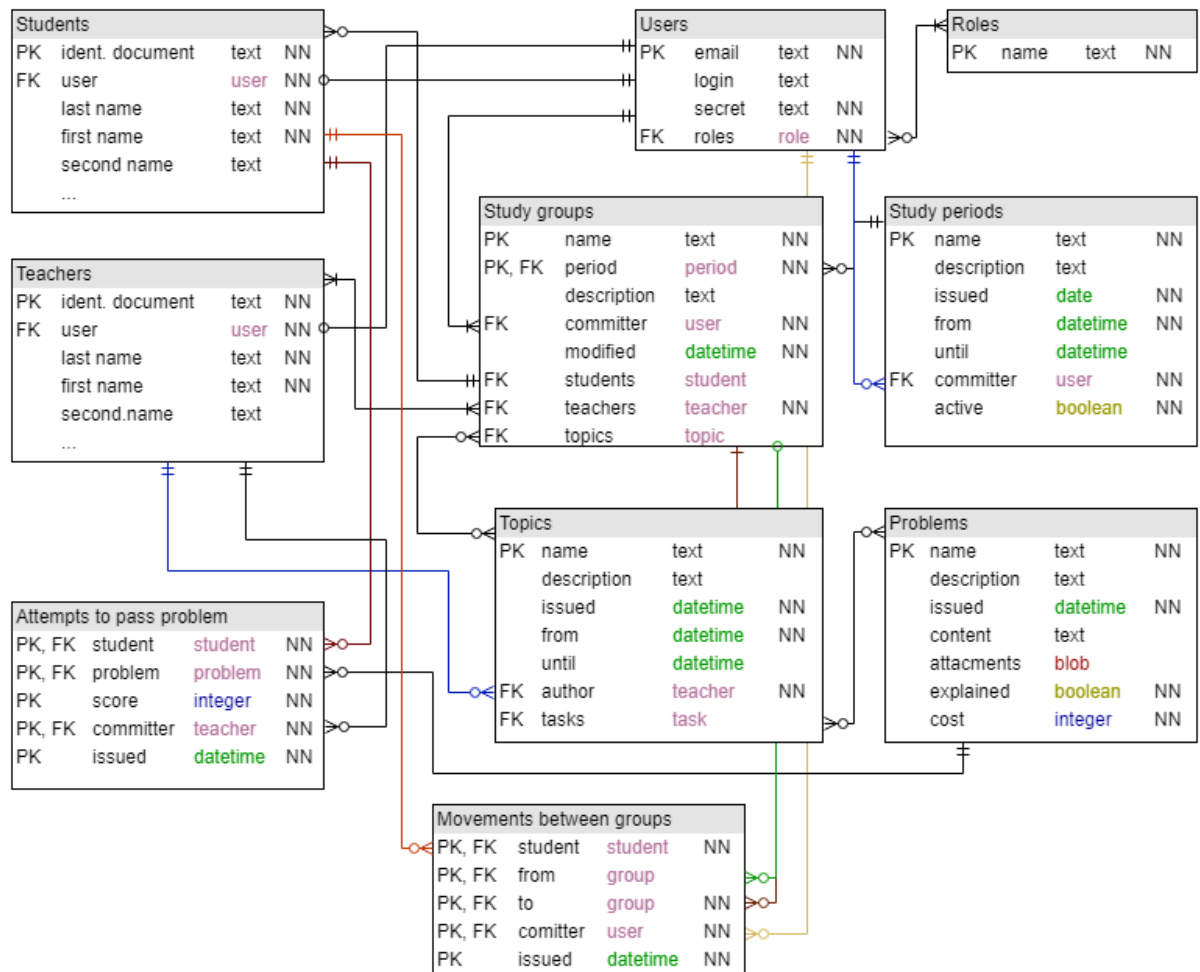
- Хранение данных, позволяющих авторизовать и идентифицировать пользователей приложения (например: email, логин, пароль, ...), а также их права доступа к отдельным частям приложения (просмотр результатов, выставление результатов, изменение структуры, администрирование). Эти данные должны храниться в независимости от ВП учебного процесса.
- Хранение данных об ВП учебных процессов. Минимальный набор отслеживаемых **данных**: название периода; дата публикации; дата и время начала; дата и время окончания; автор периода; (пояснение к периоду); доступность (значение: *да* или *нет*);
- Хранение персональных данных пользователей 2 (двух) типов: **ученик** (ФИО; место регистрации; документ УЛ; сведения о родителях; ...), **преподаватель** (ФИО; место регистрации; документ УЛ; контакты; ...). Эти данные непосредственно связаны с ВП, и для каждого ВП они могут отличаться.
- Хранение учебной структуры для каждого учебного периода: список групп, список пользователей, список тем, список задач. Внутренняя организация этой структуры будет описана ниже.
- Список групп – множество записей про учебные группы. Отслеживаемые **данные** для каждой группы: учебный период;

название; описание или пояснение; кто и когда вносил изменения. К каждой группе относится учебный план (темы занятий) и список пользователей (ученики и преподаватели).

- Список учеников группы – множество записей про учеников, которые отнесены к одной учебной группе. Во время учебного процесса состав этого списка может меняться (перевод ученика в другую группу). Для каждого ученика должны отслеживаться **данные**: учебная группа; дата и время, когда был переведён; кем был переведён; причина перевода. Переводом может считаться перевод из группы в группу, а также первое назначение (в момент регистрации).
- Список преподавателей группы – (опционально) множество записей про преподавателей, которые отнесены к одной учебной группе.
- Учебный план группы – множество записей про темы, которые отнесены к одной учебной группе. Во время учебного процесса список тем может изменяться (добавление и удаление тем). Для каждой темы должны отслеживаться **данные**: группа; название; пояснение или описание; дата публикации; дата и время начала; дата и время окончания; автор. К каждой теме относится множество задач.
- Задачи из темы – множество записей про задачи, которые относятся к одной учебной теме. Во время учебного процесса список тем может изменяться (добавление и удаление задач). Для каждой темы должны отслеживаться **данные**: тема; название; пояснение или описание; условие задачи; прикреплённые файлы; была ли разобрана; стоимость.
- Хранение данных о переводах учеников между группами. Отслеживаемые **данные**: ученик; группа, из которой был переведён; группа, в которую переведён; пользователь, который перевёл; когда был совершён перевод. При первоначальном закреплении ученика в группе, поле «откуда» должно быть пустым, либо ссылаться на запись-заглушку. Каждая запись должна изменяться 1 (один) раз при создании (дальнейшее изменение недопустимо).
- Хранение данные о результатах сдач задач за каждый подход ученика. Отслеживаемые **данные**: ученик; задача, которую сдаёт; баллы; преподаватель, который проверил; когда проверил. Каждая запись должна изменяться 1 (один) раз при создании (дальнейшее изменение недопустимо).
- Гарантия того, что результат проверки может быть вставлен только преподавателем.
- Гарантия того, что переведён из группы в группу может быть только ученик.

3. Логическая модель

По заданному техническому заданию построим логическую модель:



Модель насчитывает 10 (десять) логических сущностей. Опишем подробнее взаимодействие некоторых из этих сущностей:

- Пользователи (Users) и роли (Roles): связь между этими сущностями имеет тип *oMtMi*. *oM* - обусловлено тем, что роли могут существовать отдельно от сущности пользователя, и не обязательно каждая роль должна быть распределена между пользователями. *Mi* – обусловлено тем, что каждый зарегистрированный пользователь должен иметь роль (либо роли).

Это отношение не является нормализованным, поэтому в физической модели, оно должно раскрываться через абстракцию: таблица, в которой 2 (два) атрибута, которые являются внешними ключами, на соответствующие сущности.

- Пользователи (Users) и студенты (Students) или преподаватели (Teachers): отношение между сущностями *iiOtOo*. *iiM* – обусловлено

тем, что каждый кортеж сущностей учеников или преподавателей ссылается ровно на одного пользователя системы. *Oo* – обусловлено тем, что не каждый пользователь системы является учеником или преподавателем (например, это может быть модератор системы). Такое отношение является нормализованным и не требует дополнительных изменений.

Примечание относительно персональных данных и периодов обучения: все персональные данные косвенно привязаны к одному ВП учебного процесса через учебную группу (Study groups). Ни сущность ученика, ни сущность преподавателя не может существовать самостоятельно (без группы), поэтому запись, которая ссылается на группу, автоматически привязывается к ВП учебного процесса. Аналогично для других сущностей, ссылающихся на сущность группы.

- Группы (Study groups) и ВП учебного процесса (Study periods): отношение между сущностями *oMtOii*. *oM* – обусловлено тем, что один ВП учебного процесса может быть в разных учебных группах, при этом группа с таким периодом обязана существовать (это может быть какое-то событие, не требующее групп). *Oii* – обусловлено тем, что группа обязана относиться ровно к одному ВП учебного процесса и не может существовать без него. Такое отношение является нормализованным и не требует дополнительных изменений.
- Группы (Study groups) и ученики (Students): отношение между сущностями *iiOtMo*. *iiO* – обусловлено тем, что ученик может находиться ровно в одной группе и не может существовать как сущность без группы. *Mo* – обусловлено тем, что группа может быть пустой, либо состоять из нескольких учеников. Такое отношение является нормализованным и не требует дополнительных изменений.
- Группы (Study groups) и преподаватели (Teachers): отношение между сущностями *iMtMi*. *iM* – обусловлено тем, что преподаватель не может существовать как сущность без группы. *Mi* – обусловлено тем, что в группе обязательно должен быть преподаватель, причём в группе может быть несколько преподавателей, и один преподаватель может быть в нескольких учебных группах. Такое отношение не является нормализованным и требует раскрытия через абстракцию: таблица, в которой 2 (два) атрибута, которые являются внешними ключами, на соответствующие сущности.

Остальные отношения между учебными группами (Study groups), темами (Topics) занятий и задачами из темы (Problems) строятся аналогичным образом. Единственная особенность заключается в том, что эти сущности могут существовать независимо и добавляться по мере необходимости. Их

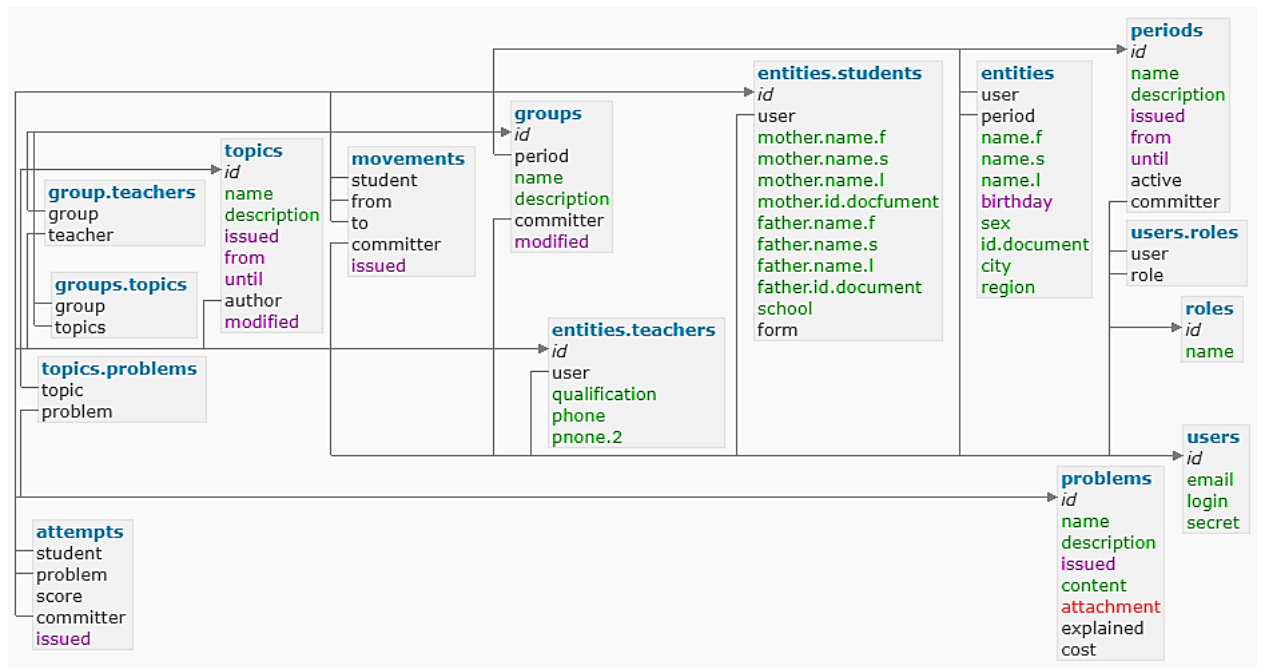
отношения не состоят в нормализованной форме, поэтому они требуют раскрытия через абстракцию, потому что считаем, что вариативность в атрибутах неллнедопустима и любое изменение должно быть зарегистрировано как новая сущность, при этом различимость сущностей достигается только за счёт названия соответствующего кортежа.

Есть ещё 2 (две) сущности, которые требуют отдельного описания: переводы между группами (Movements between groups) и попытки сдать задачу (Attempts to pass problem).

Каждая из этих сущностей идентифицируется исключительно по всем атрибутам имеет отношение *oMtOii* с другими сущностями, описанными выше. Такие отношения находятся в нормальной форме и не требуют дополнительных изменений, а сами типы обусловлены тем, что кортежи этих 2 (двух) сущностей могут содержать все возможные комбинации отношений с другими сущностями, и при этом они обязаны ссылаться на реальные данные.

4. Физическая модель

На основе логической модели построим уже физическую модель:



Обоснование решений, принятых при создании физической модели:

- Добавлена неописанная сущность (entities) для построения абстракции над персональными данными в терминах ООП. Ученики и преподаватели имеют общие атрибуты, которые могут быть вынесены в отдельную таблицу, чтобы избежать дублирования.

- В сущности ученика применена денормализация по 2 (второй) и 3 (третьей) формам, связанная с тем, что данные о родителях и их документы, удостоверяющие личность, являются зависимыми и в нормальной форме были бы вынесены с отношением *OtO* в отдельные таблицы (например, родителей – parents и документов - documents).

Денормализация обосновывается тем, что при запросе персональных данных об ученике требуется вся информация о нём (проверено на практике). Сделано для ускорения работы базы данных.

- В сущности преподавателя применена денормализация по 1 (первой) форме, связанная с тем, что контактные данные (телефонные номера) являются кортежем и в нормальной форме были бы вынесены с отношением *OtM* в отдельную таблицу (например, контактные номера - contacts).

Денормализация обосновывается тем, что преподаватель обязан иметь контактные данные, а для проверки этого предиката в нормализованной форме пришлось бы добавить триггер.

- Абстракция, которая раскрывает связь *MtM* между группами (Study groups) и учениками (Students) не применена ввиду того, что она дублируется сущностью переводами (movements) учеников между группами. При этом ослаблено условие на то, что если ученик зарегистрировался, то он приписан к какой-то группе. Это аналогично тому, что при регистрации ученик перемещается в «группу зарегистрировавшихся», которая не является учебной, потому что на практике формирование групп происходит перед самым началом учебного процесса.

- Индексы построены на всех атрибутах, которые являются внешними ключами.

Добавлен индекс по атрибуту класса (form) в сущности ученика (students), это сделано для ускорения запросов типа «список всех учеников 9 класса».

Ни одного запланированного индекса не было удалено.

- Добавлен триггер, который после регистрации нового пользователя автоматически выдаёт ему роль простого пользователя ((1) user).