

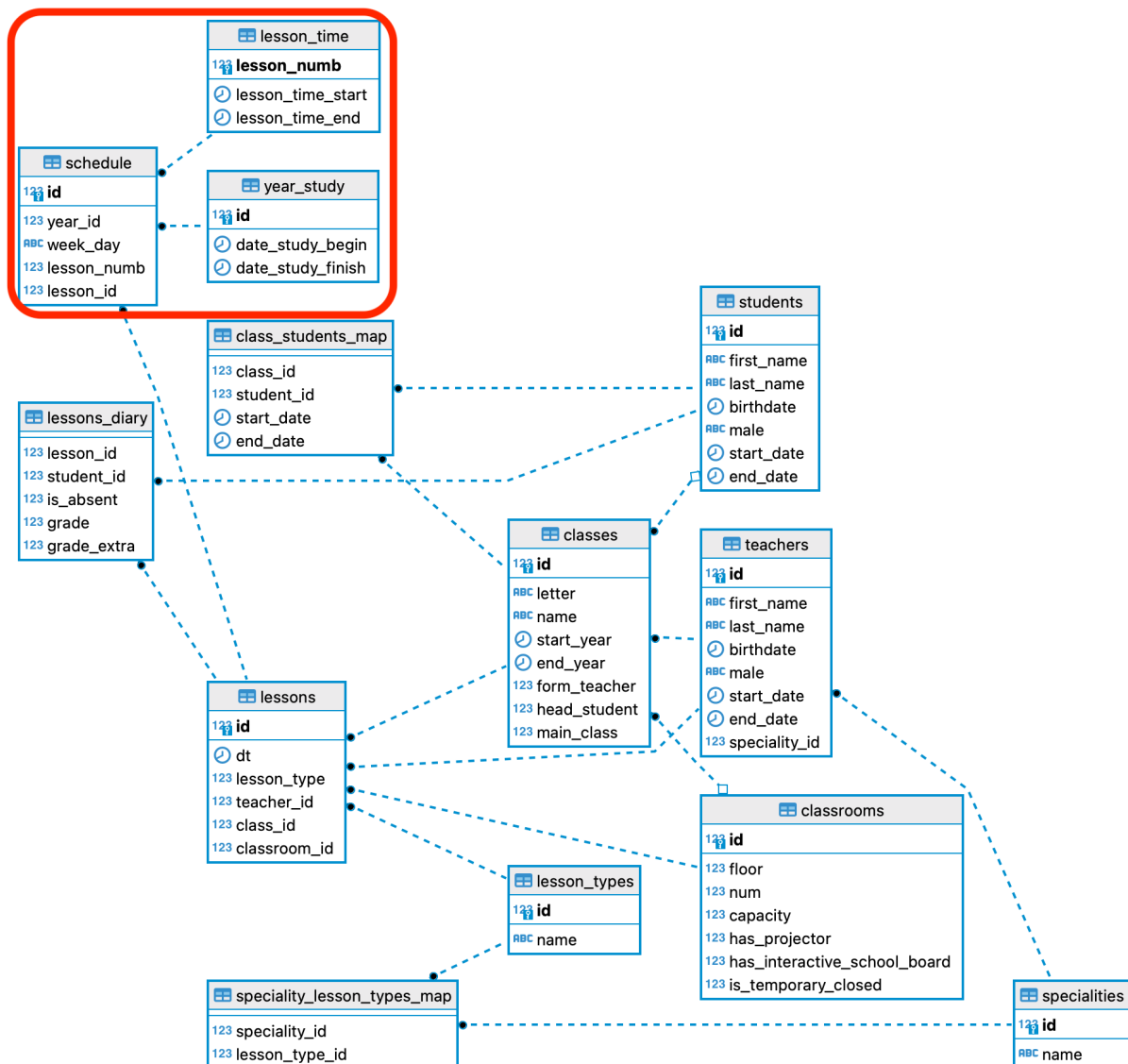
### 3.1. Требуется описать таблицу для хранения расписания уроков (schedule) и встроить ее в существующую структуру таблиц

Было встроено три таблицы.

Первая таблица schedule, которая содержит информацию о учебном дне недели, уникальном номера класса, номера урока, уникальном номере урока.

Вторая таблица содержит информацию о временном диапазоне каждого урока (время начала и окончания) по номерам (1-ый урок с 9:00 до 9:45 и тп).

Третья таблица содержит информацию id учебного года, дате начала и окончания учебного года.



#### SQL скрипт для создания таблиц:

```
-- расписание
create table schedule (
  id integer not null primary key,
  year_id integer not null,
```

```

        week_day varchar(9) not null check (week_day in ( 'monday', 'tuesday', 'wednesday',
'thursday', 'friday', 'saturday', 'sunday')),
        lesson_numb integer not null,
        lesson_id integer not null,
        constraint schdl_lesson_id_lessons_id_fkey FOREIGN KEY (lesson_id)
REFERENCES lessons(id),
        constraint schdl_lesson_numb_lesson_time_lesson_numb_fkey FOREIGN KEY
(lesson_numb) REFERENCES lesson_time(lesson_numb),
        constraint schdl_year_id_year_study_id_fkey FOREIGN KEY (year_id)
REFERENCES year_study(id)
    );

--год обучения
create table year_study (
    id integer not null primary key,
    date_study_begin date not null check (extract ('month' from date_study_begin) = 9
and extract ('day' from date_study_begin) = 1),
    date_study_finish date not null check (extract ('month' from date_study_begin) = 8
and extract ('day' from date_study_begin) = 31)
);

--время уроков
create table lesson_time (
    lesson_numb integer not null primary key check (lesson_numb < 9),
    lesson_time_start time with time zone not null,
    lesson_time_end time with time zone not null
);

```

#### **Описание столбцов таблицы schedule:**

- id (уникальный идентификатор таблицы расписание). Числовой тип, который хранит числа от -2147483648 до +2147483647. Занимает 4 байта. Не может принимать значение null. Является первичным ключом.

- week\_day (день недели). Символьный тип, который представляет строку с переменной длиной и максимальным значением 9 символа. Не может принимать значение null. Ограничение - колонка может содержать только одно из значений 'monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday'.

- lesson\_numb (уникальный идентификационный номер урока). Числовой тип, который хранит числа от -2147483648 до +2147483647. Занимает 4 байта. Не может принимать значение null.

- lesson\_id (уникальный идентификационный номер урока). Числовой тип, который хранит числа от -2147483648 до +2147483647. Занимает 4 байта. Не может принимать значение null.

Ограничения внешнего ключа:

- таблица schedule имеет внешний ключ состоящий из (classrooms\_id) ссылающийся на таблицу classrooms(id).

- таблица schedule имеет внешний ключ состоящий из (lesson\_numb) ссылающийся на таблицу lesson\_time(lesson\_numb).

- таблица schedule имеет внешний ключ состоящий из (year\_id) ссылающийся на таблицу year\_study(id).

#### **Описание столбцов таблицы lesson\_time:**

- lesson\_numb (уникальный идентификационный номер урока). Числовой тип, который хранит числа от -2147483648 до +2147483647. Занимает 4 байта. Не может принимать значение null. Имеется ограничение проверки: номер урока не может быть больше 8 (в РФ количество уроков в день не может превышать 8). Является первичным ключом.

- lesson\_time\_start (время начала урока). Тип данных для работы с временем, только время суток (с часовым поясом). Занимает 12 байт. Точность : 1 микросекунда / 14 цифр. Не может принимать значение null.

- lesson\_time\_end (время окончания урока). Тип данных для работы с временем, только время суток (с часовым поясом). Занимает 12 байт. Точность : 1 микросекунда / 14 цифр. Не может принимать значение null.

Ссылки извне: таблица schedule имеет внешний ключ состоящий из (lesson\_numb) ссылающийся на таблицу lesson\_time(lesson\_numb).

#### **Описание столбцов таблицы year\_study:**

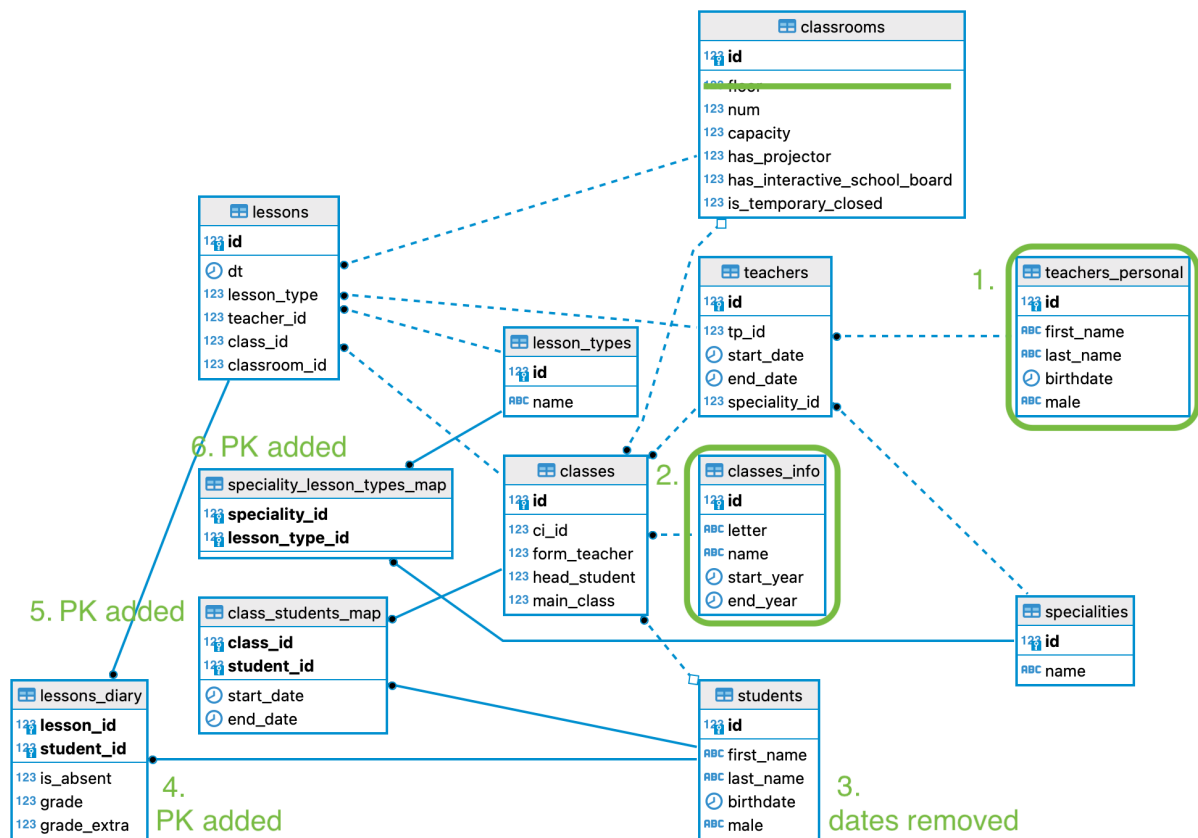
- id (уникальный идентификационный учебного года). Числовой тип, который хранит числа от -2147483648 до +2147483647. Занимает 4 байта. Не может принимать значение null. Является первичным ключом.

- date\_study\_begin (Дата начала учебного года). Тип данных для работы с датами, который хранит дату (без времени суток). Занимает 4 байта. Не может принимать значение null. Имеет ограничение проверки: каждый учебный год должен начинаться 1 сентября.

- date\_study\_finish (Дата окончания учебного года). Тип данных для работы с датами, который хранит дату (без времени суток). Занимает 4 байта. Не может принимать значение null. Имеет ограничение проверки: каждый учебный год должен заканчиваться 31 августа.

Ссылки извне: таблица schedule имеет внешний ключ состоящий из (year\_id) ссылающийся на таблицу year\_study(id).

3.2. Предоставить анализ описанной выше схемы БД, с примерами как можно было бы сделать оптимальнее в плане хранения и обращения к таблицам, необходимых индексов и т.п.



- (1) персональная информация учителей была выделена в отдельную таблицу. Данная мера обеспечивает легкость обновления персональной информации учителей, сокращения объемов памяти БД (у одного учителя может быть несколько специальностей и в таком случае персональная информация дублировалась соответствующее количество раз).

- (2) информация о классах была вынесена в отдельную таблицу. Данная мера обеспечивает легкость обновления информации о классе.

- (3) были удалены даты начала и завершения обучения учеников. Данная информация в более подробном формате представлена в таблице class\_students\_map.

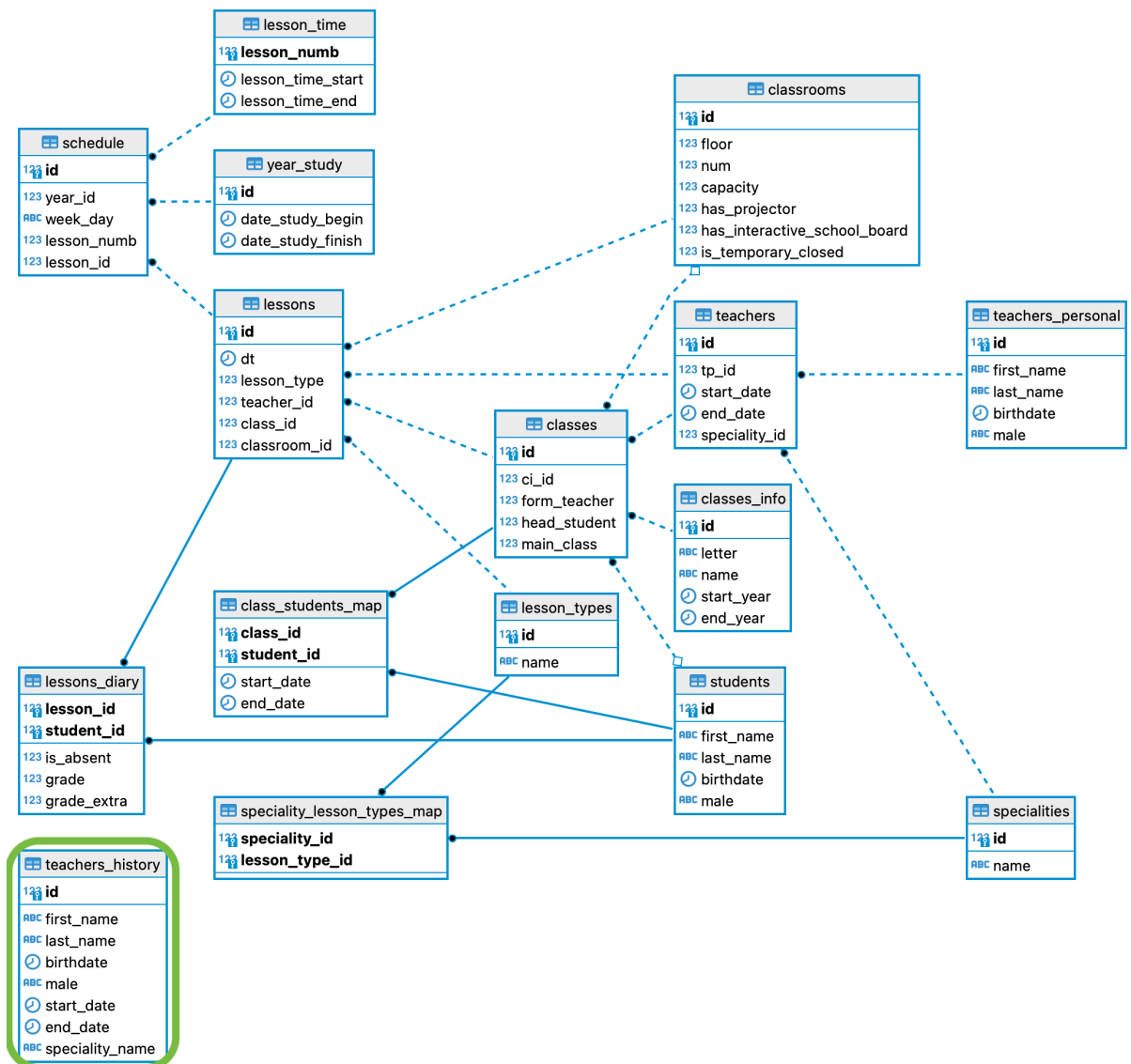
- (4-6) были добавлены составные первичные ключи. Дополнительные ограничения, которые помогают отслеживать уникальность информации.

- (7) была удалена строка этажи в таблице classrooms. Информация о этаже была интегрирована в номер кабинета (подробнее смотри в SQL скрипте Task\_3).

- (\*) были скорректированы форматы колонок для сокращения потребляемой памяти для хранения БД и добавлены ограничения уникальности, чтобы избежать появления дубликатов и избыточности информации (подробнее смотри в SQL скрипте Task\_3).

В зависимости от требований к хранению информации возможно создание **таблиц историчности в соответствии с SCD4**, которые будут содержать данные за

предыдущие учебные годы. Реализуются данные таблицы через триггерные процедуры.



Рассмотрим возможности на примере таблиц **teachers** и **teachers\_personal**. При увольнении одного из учителей (то есть при заполнении поля **end\_date**) выполняется автоматическое архивирование информации об учителе и удаление из текущих таблиц. Таким образом возможно хранение информации без влияния на загрузенность актуальных таблиц БД. Экономим вычислительные ресурсы машины при запросах, сохраняем всю необходимую информацию в соответствии с требованиями.