Предполагаю, что под «Один и тот же обучаемый не может проходить один и тот же курс дважды» имеется в виду то, что он не может посетить курс дважды ВООБЩЕ, а не в какой-то конкретный момент. То есть если даже курс проводится еще раз в другую дату, обучающийся все равно не сможет на него записаться снова.

Хочу сразу отметить, что в таблицах используются только самые необходимые поля, которые нужны для выполнения задания. Поэтому, например, в таблице для обучающихся хранятся только их имя и фамилия, которые по факту тоже не нужны, просто иначе таблица содержала бы только id.

Также все первичные ключи являются serial (в PostgreSQL). То есть проставляются, автоматически увеличиваясь на 1.

Таблица «courses» для курсов состоит из следующих полей:

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Пояснение |
| course\_id | INT | Первичный ключ |
| name | VARCHAR (255) | Название курса |
| description | TEXT | Описание курса |

Таблица «students» для обучающихся состоит из следующих полей:

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Пояснение |
| student\_id | INT | Первичный ключ |
| name | VARCHAR (255) | Имя обучающегося |
| description | VARCHAR (255) | Фамилия обучающегося |

Таблица «conducting\_courses» хранит проведенные курсы. Если таблица для самих курсов хранит только общую информацию о них, то в этой таблице уже хранятся какие-то специфические для каждой проводимой вариации курса данные. Таблица состоит из следующих полей:

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Пояснение |
| conducting\_course\_id | INT | Первичный ключ |
| course\_id | INT | Внешний ключ на таблицу «courses». Нужен, чтобы знать какой курс проводится |
| begin\_date | DATE | Дата начала курса |
| end\_date | DATE | Дата окончания курса |
| capacity | INT | Общее количество мест на курс |

Таблица «record\_course» хранит записи обучающихся на курсы. Состоит из следующих полей:

|  |  |  |
| --- | --- | --- |
| Поле | Тип | Пояснение |
| record\_course\_id | INT | Первичный ключ |
| student\_id | INT | Внешний ключ на таблицу «students». Нужен, чтобы знать к какому обучающемуся относится запись |
| conducting\_course\_id | INT | Внешний ключ на таблицу «conducting\_courses». Нужен, чтобы знать на какую именно вариацию курса обучающийся записан. Также по полям (student\_id, conducting\_course\_id) будет создан уникальный индекс. Это нужно, чтобы гарантировать, что обучающийся не может несколько раз записаться на одну и ту же вариацию проводимого курса. |
| course\_id | INT | Внешний ключ на таблицу «courses». Вообще, мы можем получить курс обратившись к «conducting\_courses», так как мы знаем его id. Это поле было добавлено, чтобы гарантировать, что обучающийся не может пройти курс повторно. То есть будет создан уникальный индекс по полям (student\_id, course\_id).  Это поле можно не добавлять, а просто проверять в коде программы перед добавлением информации проходил ли он этот курс. Но так как это является одним из требований ИМЕННО к базе данных, то я решил реализовать это на уровне самой базы. Стоит отметить, что это приводит к денормализации базы данных |

Нужно отметить, что будет сделан составной внешний ключ (conducting\_course\_id, course\_id) на таблицу «conducting\_courses». Он гарантирует, что указанный в записи таблицы «record\_course» уникальный идентификатор проводимого курса действительно относится к указанному курсу. Он ссылается не на первичный, а на уникальный составной ключ той таблицы. Снова хочу отметить, что этого можно было бы избежать, если убрать внешний ключ на таблицу «course», но тогда на уровне базы данных не будет выполняться требования под номером 3. Поэтому мне пришлось денормализовать базу.

Также будет сделан индекс по полю «conducting\_course\_id» из таблицы «record\_course». Это нужно для быстрой проверки свободных мест на проводимый курс, потому что мы будем запрашивать количество записей с этим id и сравнивать с количеством мест в поле «capacity» таблицы «conducting\_courses».

У меня нет информации, по каким полям еще часто будет что-то искаться, поэтому индексов больше нет. В теории, можно сделать индекс по дате начала курса, если они будут часто фильтроваться по этому полю.

Еще базу можно улучшить, если добавить в нее триггер, который буде проверять наличие свободных мест при записи на курс. Грубо говоря, при добавлении записи в таблицу «record\_course» будет считаться количество записей с id вариации проводимого курса («conducting\_course\_id») в этой таблице, на который происходит запись. Затем это количество записей будет сравниваться с общим количеством мест (поле capacity в таблице «conducting\_courses»). Если общее количество мест меньше, чем количество насчитанных записей, то будет выдаваться ошибка и запись не будет добавлена.

Вот две итоговые схемы базы данных (также отдельным файлом приложен скрипт для ее создания в PostgreSQL). Это одна и та же база, просто визуализирована в разных программах.



