**Разработка на SQL**

**Вариант 1. Проектирование баз данных**

Таблица **courses:**

|  |  |  |
| --- | --- | --- |
| Наименование | Тип | Описание |
| **Id[[1]](#footnote-1)** | integer | Идентификатор - Primary Key |
| Name | text[[2]](#footnote-2) | Наименование курса |
| *Description[[3]](#footnote-3)* | text | Описание курса (может быть пустым) |

**SQL:**

CREATE TABLE courses (

id integer GENERATED BY DEFAULT AS IDENTITY,

name text NOT NULL UNIQUE,

description text NULL,

CONSTRAINT pk\_course PRIMARY KEY (id)

);

Таблица **students:**

|  |  |  |
| --- | --- | --- |
| Наименование | Тип | Описание |
| **Id** | integer | Идентификатор - Primary Key |
| FirstName | text | Имя студента |
| LastName | text | Фамилия студента |

**SQL:**

CREATE TABLE students (

id integer GENERATED BY DEFAULT AS IDENTITY,

first\_name text NOT NULL,

last\_name text NOT NULL,

CONSTRAINT pk\_student PRIMARY KEY (id)

);

Таблица **schedules** является основной таблицей для учета проводимых курсов обучения, реализует связь один ко многим с таблицей курсов.

Выполняет требования задания:

⦁ Один и тот же курс может проводиться несколько раз.

⦁ Количество мест на курсах обучения определяется при добавлении курса в расписание.

|  |  |  |
| --- | --- | --- |
| Наименование | Тип | Описание |
| **Id** | integer | Идентификатор - Primary Key |
| CourseId | integer | Идентификатор курса - является внешним ключом используется для связи с таблицей Courses |
| SeatsNumber | integer | Количество свободных мест |
| StartDate | date | Дата начала курса |
| EndDate | date | Дата окончания курса |

**SQL:**

CREATE TABLE schedules (

id integer GENERATED BY DEFAULT AS IDENTITY,

course\_id integer NOT NULL,

seats\_number integer NOT NULL CHECK (seats\_number > 0),

start\_date date NOT NULL CHECK (start\_date <= end\_date),

end\_date date NOT NULL,

CONSTRAINT pk\_schedule PRIMARY KEY (id),

CONSTRAINT uk\_schedule\_id\_course\_id UNIQUE (id, course\_id),

CONSTRAINT fk\_schedule\_course\_course\_id FOREIGN KEY (course\_id) REFERENCES course (id)

);

CREATE INDEX ix\_schedule\_course\_id ON schedule (course\_id);

Таблица **courses\_students** является связующей таблицей для курсов и студентов, реализует связь многие ко многим.

Выполняет требования задания:

⦁ Один и тот же обучаемый может обучаться на разных курсах обучения.

⦁ Один и тот же обучаемый не может проходить один и тот же курс дважды.

|  |  |  |
| --- | --- | --- |
| Наименование | Тип | Описание |
| **CourseId** | integer | Идентификатор - составной ключ |
| **StudentId** | integer |
| ScheduleId | integer | Используется для идентификации курса в расписании |
| IsCoursePassed | boolean | Обозначает прохождение курса студентом |

**SQL:**

CREATE TABLE courses\_students (

course\_id integer NOT NULL,

student\_id integer NOT NULL,

schedule\_id integer NOT NULL,

is\_course\_passed boolean NOT NULL,

CONSTRAINT pk\_course\_student PRIMARY KEY (course\_id, student\_id),

CONSTRAINT fk\_courses\_students\_courses\_course\_id FOREIGN KEY (course\_id) REFERENCES course (id),

CONSTRAINT fk\_course\_student\_schedule\_schedule\_id FOREIGN KEY (schedule\_id, course\_id) REFERENCES schedule (id, course\_id),

CONSTRAINT fk\_courses\_students\_students\_student\_id FOREIGN KEY (student\_id) REFERENCES student (id)

);

CREATE INDEX ix\_courses\_students\_student\_id ON courses\_students (student\_id);

1. Жирным шрифтом выделены ключевые поля [↑](#footnote-ref-1)
2. Тип можно изменить в зависимости от требований на VARCHAR() или CHAR() [↑](#footnote-ref-2)
3. Курсивом выделены поля необязательные для заполнения [↑](#footnote-ref-3)