**Документация по структуре базы данных**

**и ее описание**

## **1. Таблица students**

### **Описание:**

Таблица хранит информацию о студентах, их контактных данных и личных данных.

**Атрибуты:**

* **student\_id** (SERIAL PRIMARY KEY) — уникальный идентификатор студента, автоматически увеличиваемый при добавлении нового студента.
* **full\_name** (VARCHAR(255)) — полное имя студента.
* **birth\_date** (DATE) — дата рождения студента.
* **contact\_info** (TEXT) — контактная информация студента (например, номер телефона, электронная почта).
* **UNIQUE (full\_name, birth\_date)** — ограничение уникальности для предотвращения дублирования студентов с одинаковым именем и датой рождения.

### **Отношения:**

* Связана с таблицей **enrollments** через внешний ключ **student\_id** (каждый студент может быть зачислен на несколько курсов).
* Связана с таблицей **grades** через таблицу **enrollments** (студент может получать оценки за курсы).

## **2. Таблица teachers**

### **Описание:**

Таблица хранит информацию о преподавателях, их контактных данных и личных данных.

### **Атрибуты:**

* **teacher id** (SERIAL PRIMARY KEY) — уникальный идентификатор преподавателя, автоматически увеличиваемый при добавлении нового преподавателя.
* **full\_name** (VARCHAR(255)) — полное имя преподавателя.
* **birth\_date** (DATE) — дата рождения преподавателя.
* **contact\_info** (TEXT) — контактная информация преподавателя (например, номер телефона, электронная почта).
* **UNIQUE (full\_name, birth\_date)** — ограничение уникальности для предотвращения дублирования преподавателей с одинаковым именем и датой рождения.

### **Отношения:**

* Связана с таблицей **courses** через внешний ключ **teacher\_id** (каждый преподаватель может преподавать несколько курсов).

## **3. Таблица courses**

### **Описание:**

Таблица хранит информацию о курсах, их названиях, описаниях и преподавателях. Каждый курс может быть привязан только к одному преподавателю.

### **Атрибуты:**

* **course\_id** (SERIAL PRIMARY KEY) — уникальный идентификатор курса, автоматически увеличиваемый при добавлении нового курса.
* **course\_name** (VARCHAR(255)) — название курса.
* **description** (TEXT) — описание курса.
* **teacher\_id** (INTEGER) — внешний ключ, ссылающийся на преподавателя, который ведет курс.
* **UNIQUE (course\_name, teacher\_id)** — ограничение уникальности для предотвращения добавления одного и того же курса, преподаваемого тем же преподавателем.

### **Отношения:**

* Связана с таблицей **teachers** через внешний ключ **teacher\_id** (каждый курс имеет одного преподавателя).
* Связана с таблицей **enrollments** через внешний ключ **course\_id** (студенты могут записываться на курс).

## **4. Таблица enrollments**

### **Описание:**

Таблица хранит информацию о зачислении студентов на курсы. Каждый студент может быть зачислен на несколько курсов.

### **Атрибуты:**

* **enrollment\_id** (SERIAL PRIMARY KEY) — уникальный идентификатор зачисления, автоматически увеличиваемый при добавлении нового зачисления.
* **student\_id** (INTEGER) — внешний ключ, ссылающийся на студента, который записался на курс.
* **course\_id** (INTEGER) — внешний ключ, ссылающийся на курс, на который записан студент.
* **UNIQUE (student\_id, course\_id)** — ограничение уникальности, предотвращающее дублирование записей для одного студента на один курс.

### **Отношения:**

* Связана с таблицей **students** через внешний ключ **student\_id** (один студент может быть зачислен на несколько курсов).
* Связана с таблицей **courses** через внешний ключ **course\_id** (каждый курс может иметь несколько студентов).
* Связана с таблицей **grades** через внешний ключ **enrollment\_id** (каждое зачисление может иметь оценку).

## **5. Таблица grades**

### **Описание:**

Таблица хранит оценки студентов по курсам, на которые они были зачислены. Каждое зачисление может иметь несколько оценок, но не более одной оценки за день.

### **Атрибуты:**

* **grade\_id** (SERIAL PRIMARY KEY) — уникальный идентификатор оценки, автоматически увеличиваемый при добавлении новой оценки.
* **enrollment\_id** (INTEGER) — внешний ключ, ссылающийся на зачисление студента на курс.
* **grade** (INTEGER) — оценка, выставленная студенту, целое число от 1 до 5.
* **grade\_date** (DATE) — дата выставления оценки. По умолчанию — текущая дата.
* **UNIQUE (enrollment\_id, grade\_date)** — ограничение уникальности, чтобы на одно зачисление на курс не могли быть выставлены оценки в один и тот же день.

### **Отношения:**

* Связана с таблицей **enrollments** через внешний ключ **enrollment\_id** (оценка привязана к конкретному зачислению студента на курс).

# **Отношения между таблицами:**

1. **students → enrollments → courses → teachers**
   * Один студент может быть зачислен на несколько курсов (многие ко многим через таблицу enrollments).
   * Один преподаватель может преподавать несколько курсов (один ко многим через таблицу courses).
2. **students и courses** связаны через таблицу **enrollments**.
3. **students и grades** связаны через таблицу **enrollments**, поскольку оценки привязаны к зачислению студентов на курсы.
4. **teachers и courses** связаны через внешний ключ **teacher\_id** в таблице courses.
5. **grades** зависит от **enrollments**, где каждая оценка относится к конкретному зачислению.

UNIQUE предотвращает дублирование записей.

# **Запросы к базе данных:**

1. возможность выводить список студентов по определённому предмету

SELECT

student\_id,

full\_name,

birth\_date,

contact\_info

FROM

students

JOIN

enrollments ON student\_id = student\_id

JOIN

courses ON course\_id = course\_id

WHERE

course\_name = 'Название предмета';

2. возможность выводить список предметов, которые преподает конкретный преподаватель

SELECT

course\_id,

course\_name,

description

FROM

courses

JOIN

teachers ON teacher\_id = teacher\_id

WHERE

full\_name = 'ФИО преподавателя';

3. возможность выводить средний балл студента по всем предметам

SELECT

student\_id,

full\_name,

ROUND(AVG(grade), 2) AS average\_grade

FROM

students

JOIN

enrollments ON student\_id = student\_id

JOIN

grades ON enrollment\_id = enrollment\_id

WHERE

full\_name = 'ФИО студента'

GROUP BY

student\_id, full\_name;

4. возможность выводить рейтинг преподавателей по средней оценке студентов

SELECT

teacher\_id,

full\_name AS teacher\_name,

ROUND(AVG(grade), 2) AS average\_student\_grade

FROM

teachers

JOIN

courses ON teacher\_id = teacher\_id

JOIN

enrollments ON course\_id = course\_id

JOIN

grades ON enrollment\_id = enrollment\_id

GROUP BY

teacher\_id, full\_name

ORDER BY

average\_student\_grade DESC;

5. возможность выводить список преподавателей, которые преподавали более 3 предметов за последний год

SELECT

teacher\_id,

full\_name,

COUNT(course\_id) AS courses\_count

FROM

teachers

JOIN

courses ON teacher\_id = teacher\_id

WHERE

creation\_date >= CURRENT\_DATE - INTERVAL '1 year'

GROUP BY

teacher\_id, full\_name

HAVING

COUNT(course\_id) > 3;

6. возможность выводить список студентов, которые имеют средний балл выше 4 по математическим предметам, но ниже 3 по гуманитарным

SELECT

student\_id,

full\_name

FROM

students

JOIN

enrollments ON student\_id = student\_id

JOIN

courses ON course\_id = course\_id

JOIN

grades ON enrollment\_id = enrollment\_id

GROUP BY

student\_id, full\_name

HAVING

AVG(CASE

WHEN course\_name LIKE '%Математика%' THEN grade

ELSE NULL

END) > 4

AND AVG(CASE

WHEN course\_name LIKE '%Гуманитарный%' THEN grade

ELSE NULL

END) < 3;

7. возможность определить предметы, по которым больше всего двоек в текущем семестре

SELECT

course\_name,

COUNT(grade) AS two\_count

FROM

grades

JOIN

enrollments ON enrollment\_id = enrollment\_id

JOIN

courses ON course\_id = course\_id

WHERE

grade = 2

AND grade\_date >= CURRENT\_DATE - INTERVAL '6 months'

GROUP BY

course\_name

ORDER BY

two\_count DESC

LIMIT 1;

8. возможность выводить студентов, которые получили высший балл по всем своим экзаменам, и преподавателей, которые вели эти предметы

SELECT

student\_id,

full\_name AS student\_name,

teacher\_id,

full\_name AS teacher\_name,

course\_name

FROM

students

JOIN

enrollments ON student\_id = student\_id

JOIN

courses ON course\_id = course\_id

JOIN

grades ON enrollment\_id = enrollment\_id

JOIN

teachers ON teacher\_id = teacher\_id

WHERE

grade = (SELECT MAX(grade) FROM grades WHERE enrollment\_id = enrollment\_id)

GROUP BY

student\_id, full\_name, teacher\_id, full\_name, course\_name;

9. возможность просматривать изменение среднего балла студента по годам обучения

SELECT

EXTRACT(YEAR FROM year) AS year,

student\_id,

full\_name,

ROUND(AVG(grade), 2) AS average\_grade

FROM

students

JOIN

enrollments ON student\_id = student\_id

JOIN

courses ON course\_id = course\_id

JOIN

grades ON enrollment\_id = enrollment\_id

WHERE

student\_id = [ID\_студента]

GROUP BY

student\_id, full\_name, EXTRACT(YEAR FROM year)

ORDER BY

year;

10. возможность определить группы, в которых средний балл выше, чем в других, по аналогичным предметам, чтобы выявить лучшие методики преподавания или особенности состава группы

SELECT

group\_id,

course\_name,

ROUND(AVG(grade), 2) AS average\_grade

FROM

enrollments

JOIN

grades ON enrollment\_id = enrollment\_id

JOIN

courses ON e.course\_id = course\_id

GROUP BY

group\_id, course\_name

ORDER BY

course\_name, average\_grade DESC;

11. вставка записи о новом студенте с его личной информацией, такой как ФИО, дата рождения, контактные данные и др.

INSERT INTO students (full\_name, birth\_date, contact\_info)

VALUES ('John Smith', '2000-05-15', 'Moscow', 'ivanov@example.com', '+7 900 123-45-67');

12. обновление контактной информации преподавателя, например, электронной почты или номера телефона, на основе его идентификационного номера или ФИО

UPDATE teachers

SET email = 'new\_email@example.com', phone = '+7 900 123-45-67'

WHERE teacher\_id = 1; -- или WHERE full\_name = 'Jane Adams';

13. удаление записи о предмете, который больше не преподают в учебном заведении. Требуется также учесть возможные зависимости, такие как оценки студентов по этому предмету

DELETE FROM courses

WHERE course\_id = 456;

14. вставка новой записи об оценке, выставленной студенту по определённому предмету, с указанием даты, преподавателя и полученной оценки

INSERT INTO grades (enrollment\_id, grade, grade\_date)

SELECT enrollment\_id, 5, '2024-12-21'

FROM enrollments

JOIN courses ON course\_id = course\_id

JOIN teachers ON teacher\_id = teacher\_id

WHERE student\_id = 123

AND course\_name = 'Math'

AND full\_name = 'Robert Green';