

ОСНОВЫ SQL

Люба Вайгель

Ментор ASAP Education



Давайте вспомним

- Что такое база данных?
- Какие виды баз данных бывают?
- В чем особенность реляционных БД?
- Что такое отношение, атрибут, кортеж?
- Для чего используются первичные и внешние ключи?
- Что такое транзакция в БД?
- Какими свойствами должна обладать транзакция?
- Для чего нужно моделирование данных?
- Какие уровни моделирования данных существуют?
- Для чего нужна нормализация БД?

План занятия

→ Структура SQL запроса

Полная в общем виде

→ Как выводить данные

Сортировка, группировка и объединение таблиц

→ Практика

Решим несколько задач на выборку данных



Разве ИИ может написать эффективные SQL-запрос?

А ты можешь?

Структура SQL запроса

SELECT ('столбцы или * для выбора всех столбцов; обязательно')

FROM ('таблица; обязательно')

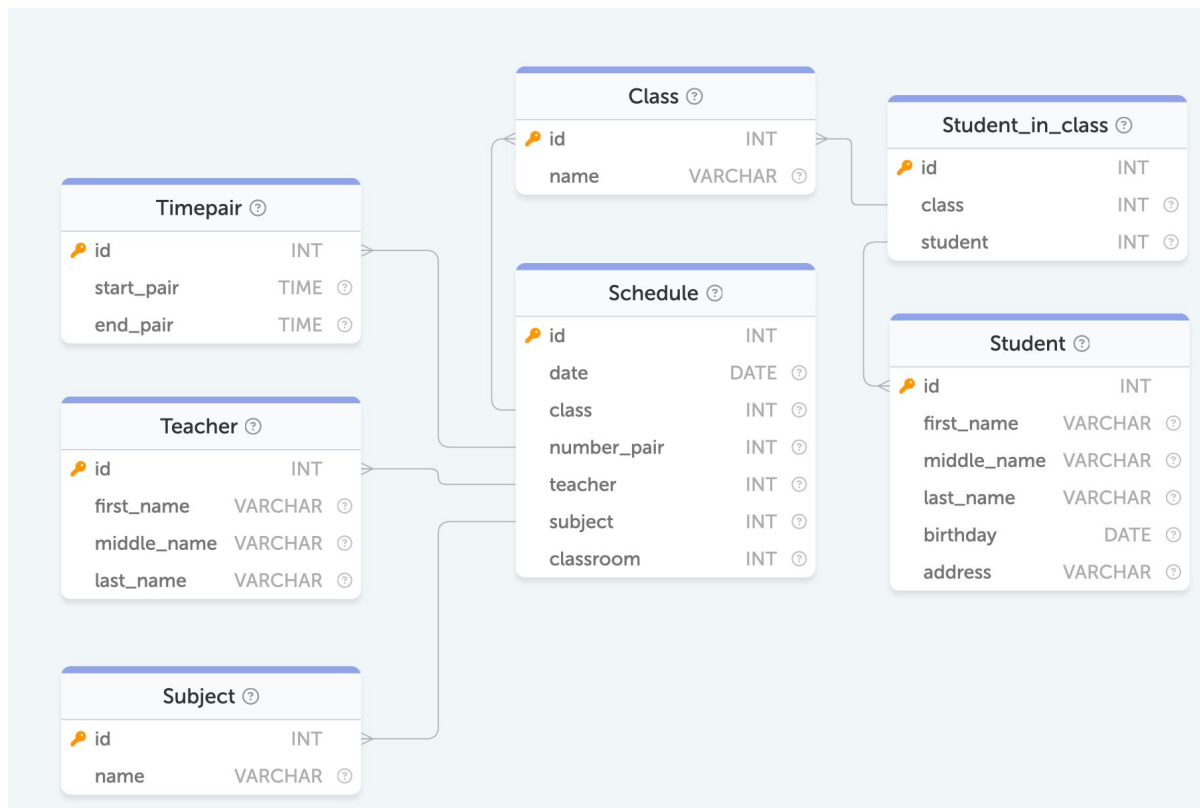
WHERE ('условие/фильтрация, например, city = 'Moscow'; необязательно')

GROUP BY ('столбец, по которому хотим сгруппировать данные; необязательно')

HAVING ('условие/фильтрация на уровне сгруппированных данных; необязательно')

ORDER BY ('столбец, по которому хотим отсортировать вывод; необязательно');

Тренировочная БД



Вывод данных из таблицы

Задача:

Получить все данные из таблицы
Student

Решение:

```
SELECT *  
  
FROM Student
```

Задача:

Получить список всех студентов (их
фамилии имена)

Решение:

```
SELECT first_name, last_name  
  
FROM Student
```

Алиасы

Если мы хотим вывести какие-то столбцы таблицы, но хотим в конечной выборке видеть их под другим названием, мы можем использовать **алиасы** (псевдонимы)

Алиасы можно использовать как для названий столбцов, так и названий таблиц

```
SELECT id, first_name AS
Name FROM Student
```

```
SELECT id, first_name Name
FROM Student
```

```
SELECT *
FROM Student AS s;
```

Условный оператор WHERE

- На случай, когда требуется сделать выборку по определенному условию, существует оператор **WHERE**, после которого следуют условия для ограничения строк
- Если запись удовлетворяет этому условию, то попадает в результат, иначе отбрасывается

```
SELECT поля_таблиц FROM
наименование_таблицы
WHERE
условие_на_ограничение_строк
[логический_оператор
другое_условие_на_ограничени
е_строк] ;
```


Условный оператор WHERE

Оператор	Обозначение	Описание
Равенство	=	Если оба значения равны, то результат будет равен 1, иначе 0
Эквивалентность	<=>	Аналогичен оператору равенства, за исключением того, что результат будет равен 1 в случае сравнения NULL с NULL и 0, когда идёт сравнение любого значения с NULL
Неравенство	<> или !=	Если оба значения не равны, то результат будет равен 1, иначе 0
Меньше	<	Если одно значение меньше другого, то результат будет равен 1, иначе 0
Меньше или равно	<=	Если одно значение меньше или равно другому, то результат будет равен 1, иначе 0
Больше	>	Если одно значение больше другого, то результат будет равен 1, иначе 0
Больше или равно	>=	Если одно значение больше или равно другому, то результат будет равен 1, иначе 0

SELECT поля_таблиц **FROM**
наименование_таблицы
WHERE

условие_на_ограничение_строк
[логический_оператор
другое_условие_на_ограничени
е_строк];

Условный оператор WHERE

Задача

Получить фамилии всех студентов,
которых зовут Иван

Решение

```
SELECT last_name
FROM Student
WHERE first_name =
"Ivan"
```

Сортировка, оператор ORDER BY

SELECT поля_таблиц **FROM**
наименование_таблицы
WHERE ...
ORDER BY столбец_1 [ASC | DESC]

- ASC - сортировка по возрастанию (по умолчанию)
- DESC - сортировка по убыванию

Тип данных	ASC (по возрастанию)	DESC (по убыванию)
Строковый тип	Лексикографический (алфавитный) порядок от «А» до «Z» (или от «А» до «Я»)	Лексикографический порядок от «Z» до «А» (или от «Я» до «А»)
	Сначала идут записи начинающиеся на «а», затем на «b» и т.д.	Сначала идут записи начинающиеся на «z», «у» и т.д.
Числовой тип	От меньшего к большему	От большего к меньшему

Тип данных	ASC (по возрастанию)	DESC (по убыванию)
Дата и время	От более ранней даты/времени к более поздней Например, сначала 01.01.2024 , затем 01.02.2024	От более поздней даты/времени к более ранней Например, сначала 01.02.2024 , затем 01.01.2024
Булевый тип	False идёт перед True	True идёт перед False

Группировка, оператор GROUP BY

Если нужно объединить
результаты выборки
по значениям одного
или нескольких столбцов,
используется оператор
GROUP BY

```
SELECT [литералы, агрегатные_функции,  
поля_группировки]  
FROM имя_таблицы  
GROUP BY поля_группировки;
```

Группировка, оператор GROUP BY

```
SELECT category_id, product_id, name
FROM product
ORDER BY category_id
```

category_id	product_id	name
3	1	Пылесос S6
3	2	Холодильник A2
5	12	Nikon D750
5	11	Lord Nikon 95
6	5	Xbox
6	3	Nintendo
6	4	PlayStation
7	6	Наушники S3

Как посчитать сколько
продуктов в каждой
из категорий?

Группировка, оператор GROUP BY

```
SELECT category_id, product_id, name  
FROM product  
ORDER BY category_id
```

category_id	product_id	name
3	1	Пылесос S6
3	2	Холодильник A2
5	12	Nikon D750
5	11	Lord Nikon 95
6	5	Xbox
6	3	Nintendo
6	4	PlayStation
7	6	Наушники S3

кол-во продуктов
в категории

```
SELECT category_id,  
       count(*) AS count_products  
FROM product  
GROUP BY category_id
```

category_id	count_products
3	2
5	2
6	3
7	1

Агрегатные функции

Агрегатная функция – это функция, которая выполняет вычисление на наборе значений и возвращает одиночное значение

```
SELECT room_id,  
        MAX(end_date) AS last_end_date  
FROM Reservations  
GROUP BY room_id
```

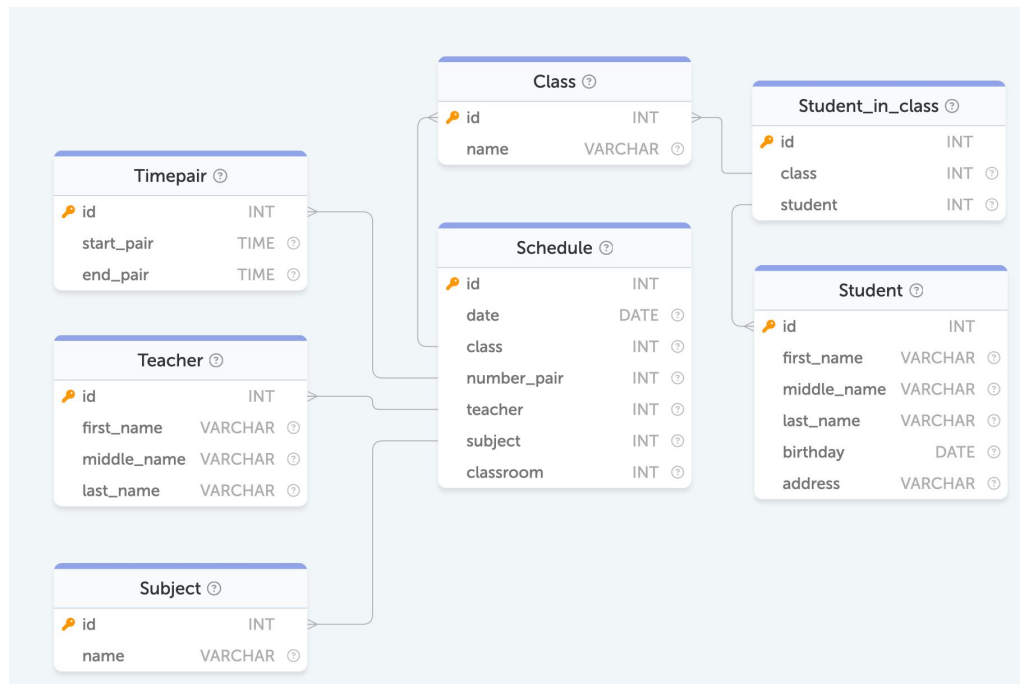
*Поиск самой поздней даты
выезда для каждого жилого
помещения*

Функция	Описание
SUM(поле_таблицы)	Возвращает сумму значений
AVG(поле_таблицы)	Возвращает среднее значение
COUNT(поле_таблицы)	Возвращает количество записей
MIN(поле_таблицы)	Возвращает минимальное значение
MAX(поле_таблицы)	Возвращает максимальное значение

В реальности очень часто
приходится делать **выборку**
из нескольких таблиц, каким-то
образом **объединяя их**

Соединение таблиц (JOIN)

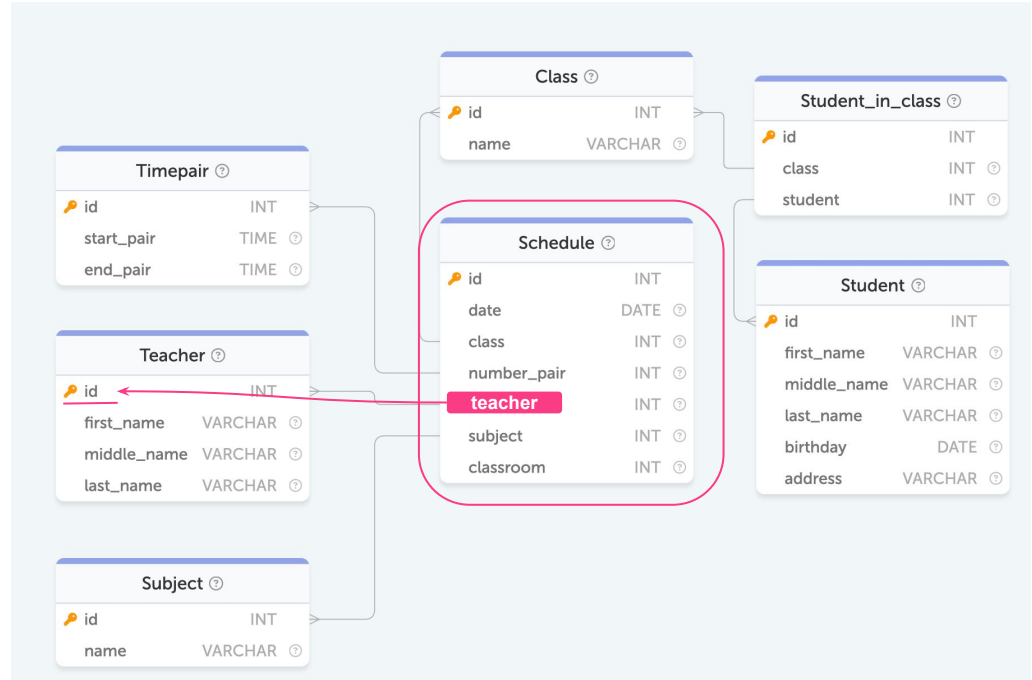
```
SELECT поля_таблиц
FROM таблица_1
[INNER] | [[LEFT | RIGHT |
FULL][OUTER]] JOIN таблица_2
ON условие_соединения
[[INNER] | [[LEFT | RIGHT |
FULL][OUTER]] JOIN таблица_n
ON условие_соединения]
```



Соединение таблиц (JOIN)

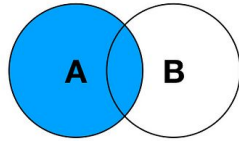
Условие соединения

**Schedule.teacher =
Teacher.id**

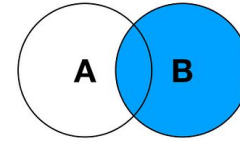


Соединение таблиц (JOIN)

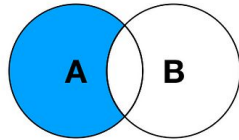
SQL JOINS



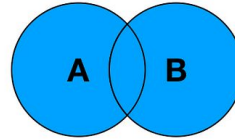
LEFT JOIN



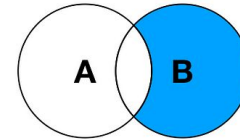
RIGHT JOIN



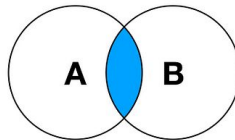
LEFT JOIN EXCLUDING
INNER JOIN



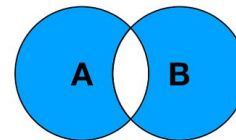
FULL OUTER JOIN



RIGHT JOIN EXCLUDING
INNER JOIN



INNER JOIN



FULL OUTER JOIN EXCLUDING
INNER JOIN

Соединение таблиц (JOIN)

Задача

Получить список студентов (их имена и фамилии), которые учатся в 9 В классе

Решение

```
select s.first_name,  
s.last_name  
  
from Students s  
  
join Student_in_class sic ON  
s.id = sic.student  
  
Join Class c ON c.id =  
sic.class  
  
where c.name = '9 В'
```

Задачи

1. Получить расписание для для 11 А класса
2. Получить список преподавателей, которые ведут занятия 2019-09-01
3. Список предметов, которые ведет учитель *Pavel Petrovich Romashkin*
4. Найти все предметы, которые проходят в аудитории 35
5. Найти число студентов в каждом классе
6. Получить список всех классов, в которых преподается предмет *Geography*
7. Выведите заполненность классов в порядке убывания