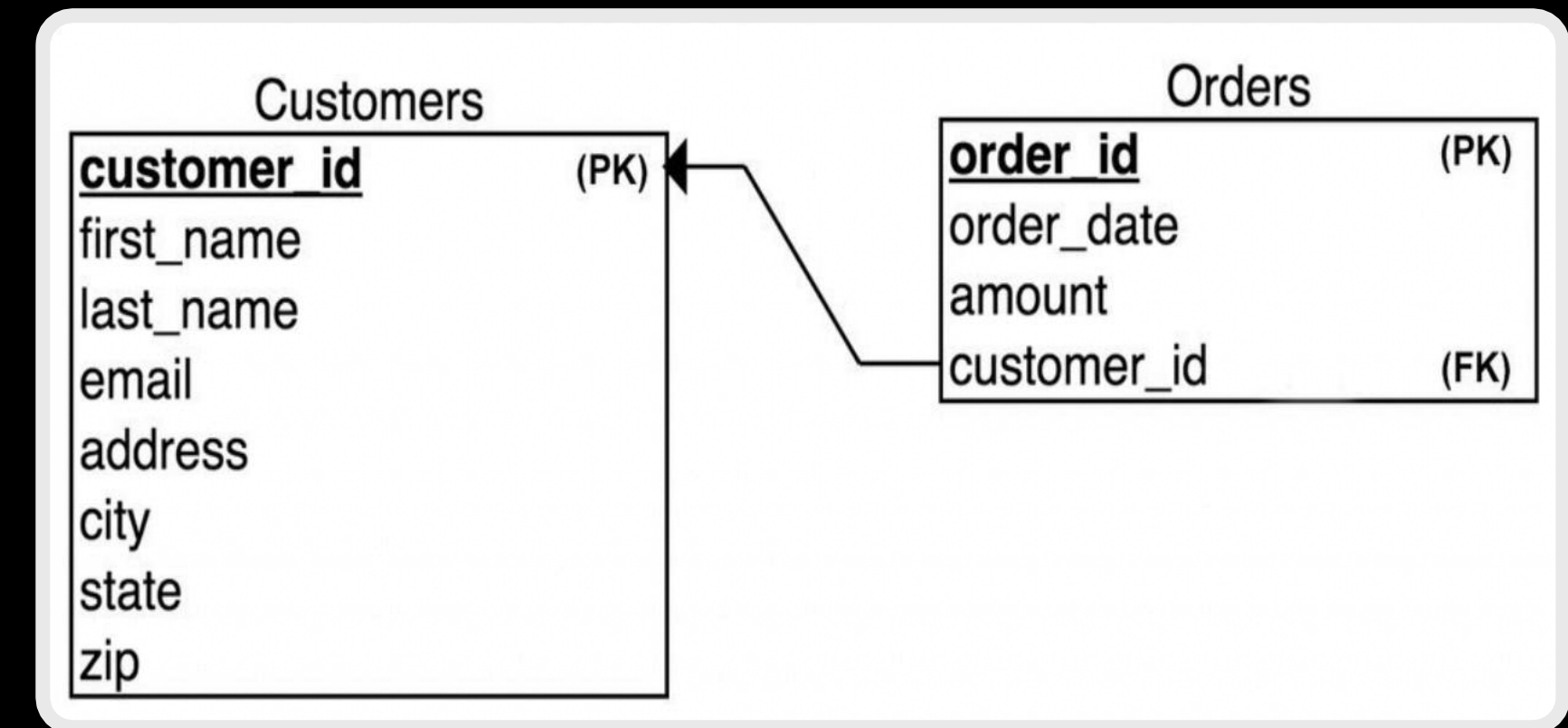


Зачем нам базы данных?

	Простой перебор	Поиск по индексу*
100 записей	100 мс	7 мс
10k записей	10 секунд	14 мс
1kk записей	11 дней	28 мс



- Структурирование/упорядочивание
- Хранение
- Систематизация
- Управление

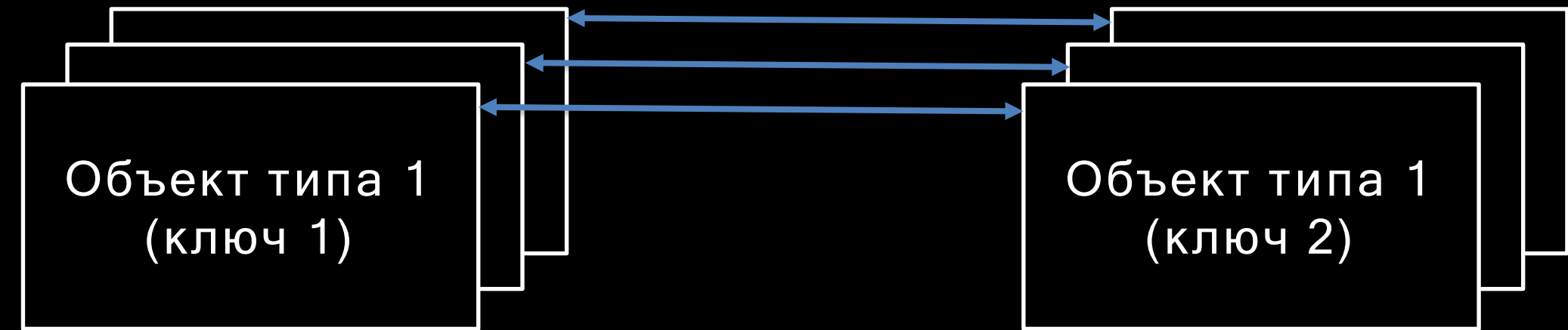
NoSQL

	Тип	Примеры
1	Ключ-значение (Key-Value)	Redis, Amazon DynamoDB
2	Документоориентированные (JSON)	MongoDB, FirestoreDB
3	Графовые	AllegroGraph, IBM Graph, Neo4j, Titan
4	Поисковые БД	ElasticSearch

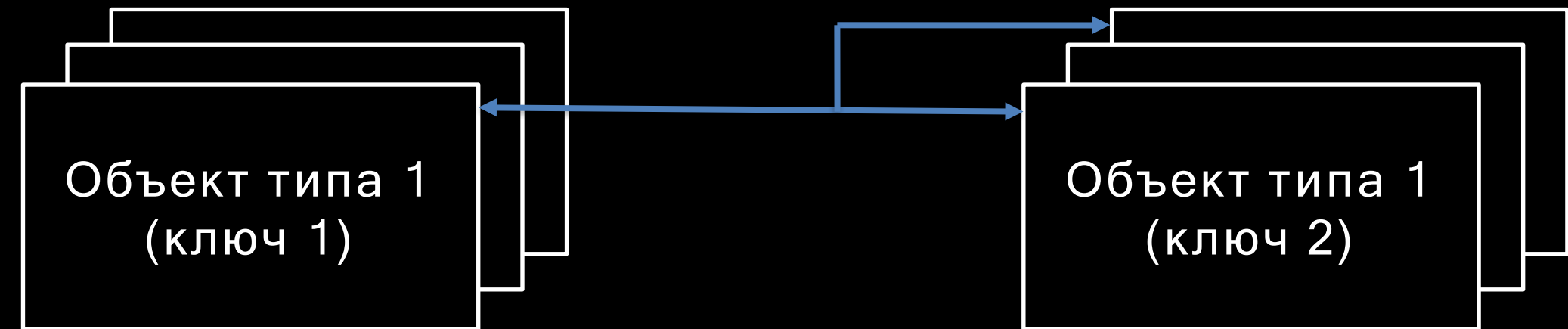


Виды связей

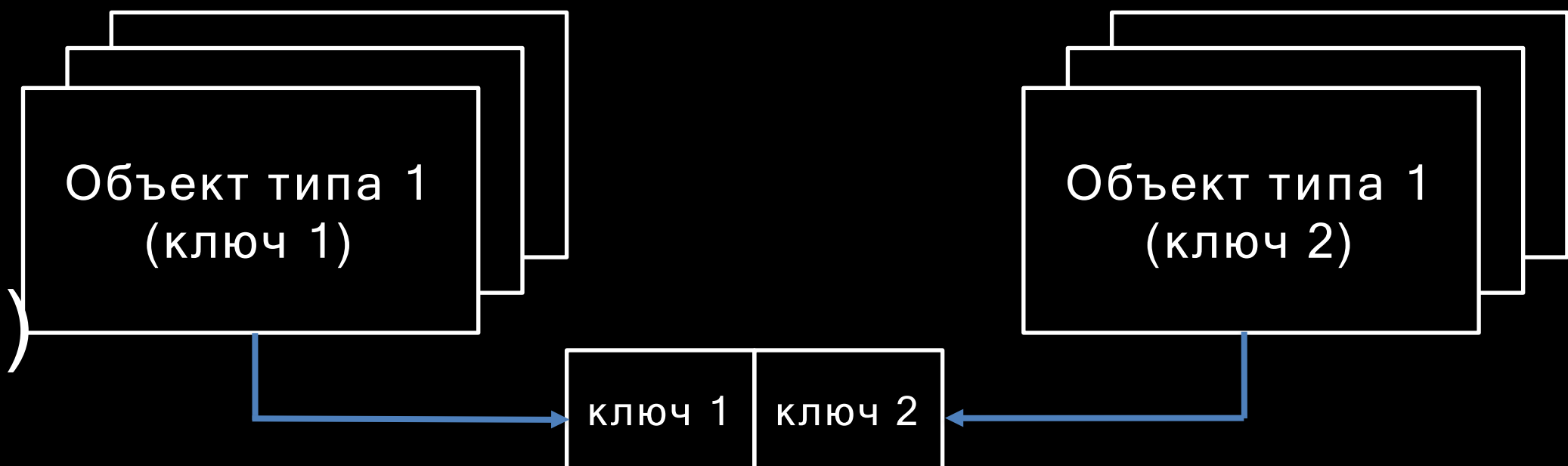
- Один ко многим (1to1)
(Студент → Группа)



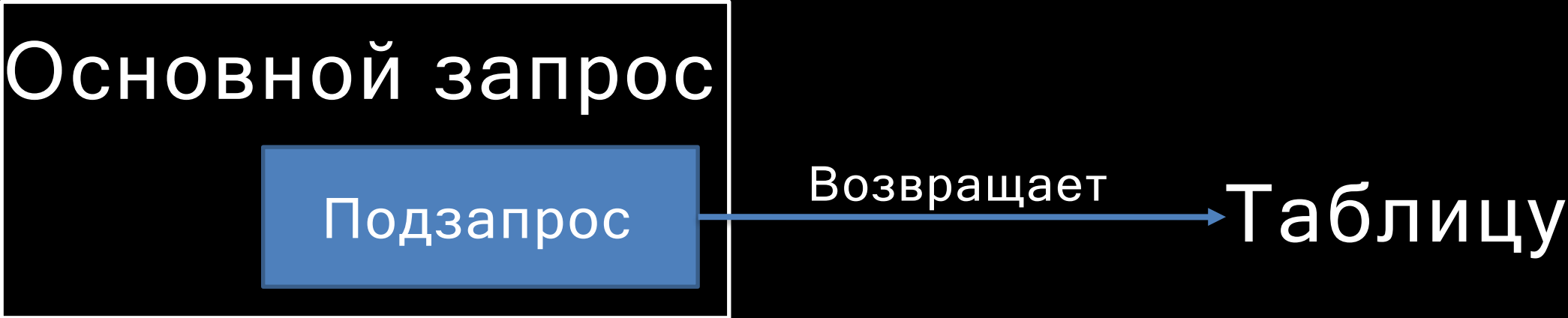
- Один ко многим (1toM)
(Студент → Экзамены)



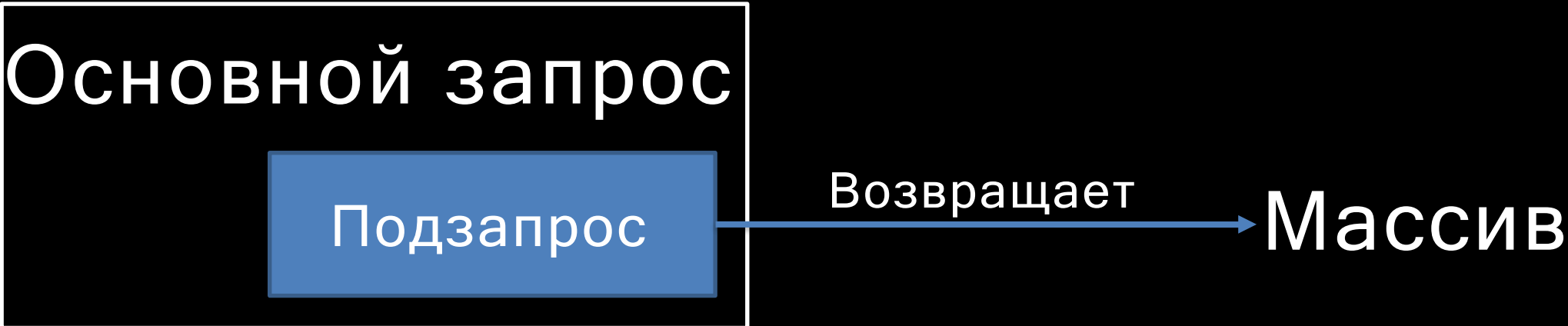
- Многие ко многим (MtoM)
(Расписание занятий)



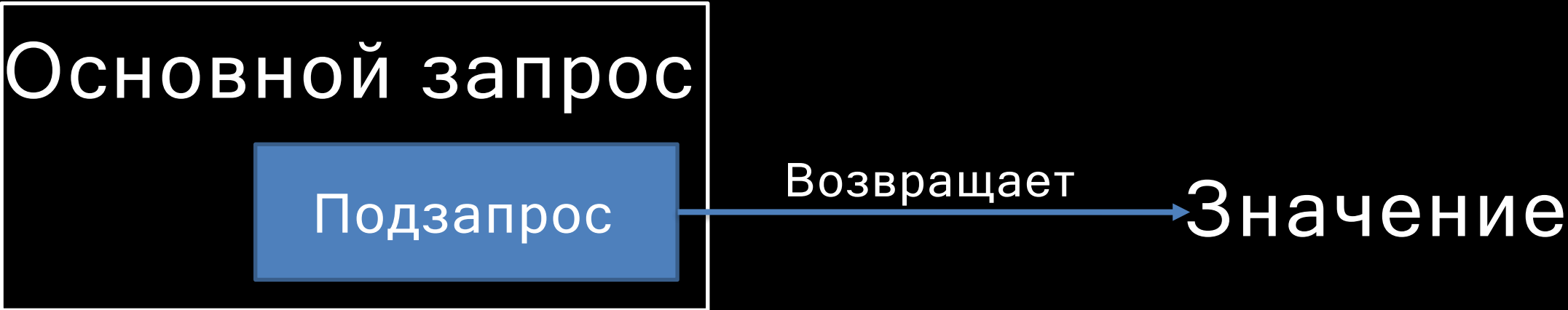
Подзапросы (вложенные запросы)



Номер сотрудника	ФИО сотрудника	Должность
325	Иванов И.И.	Программист
567	Сергеева С.С.	Администратор БД
225	Петров П.П.	Программист
976	Николаев Н.Н.	Системный администратор



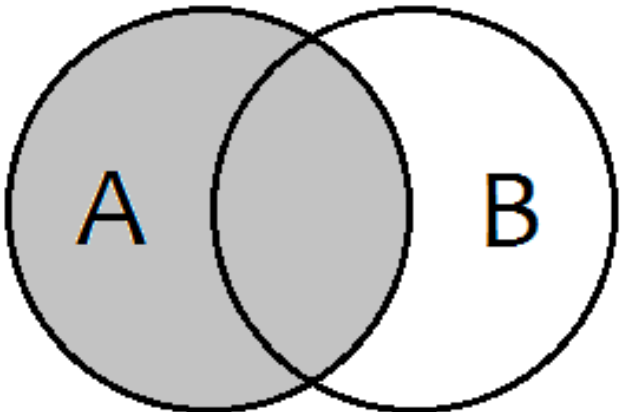
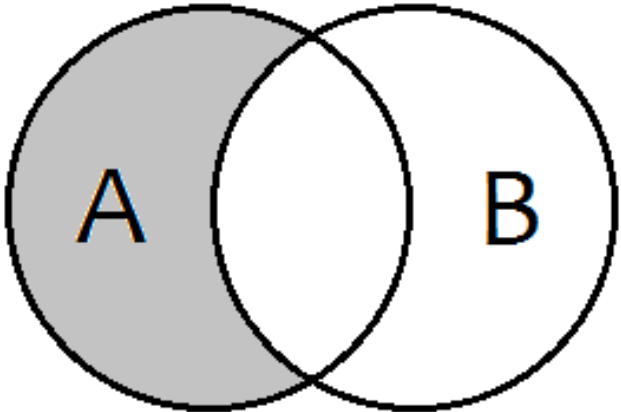
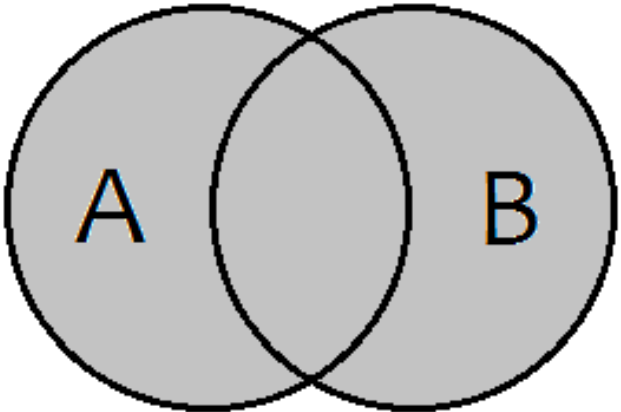
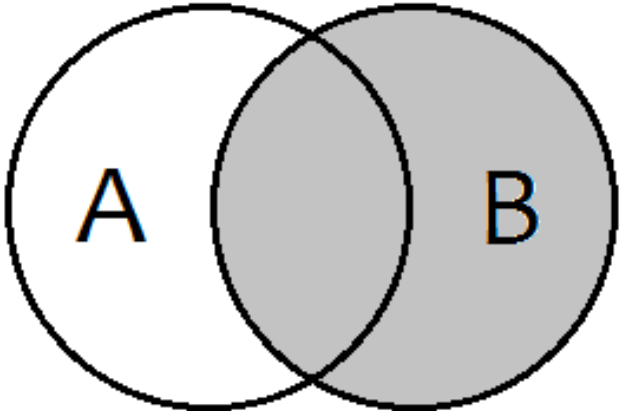
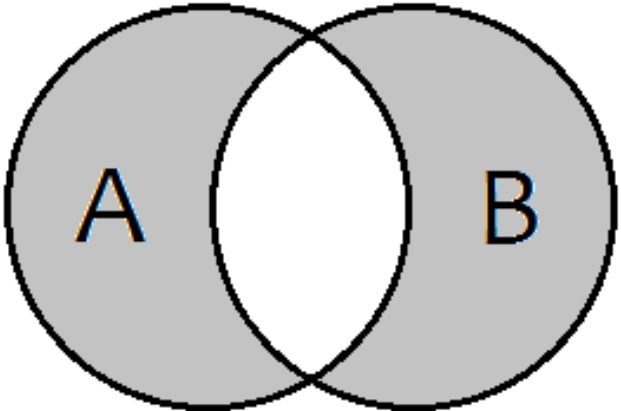
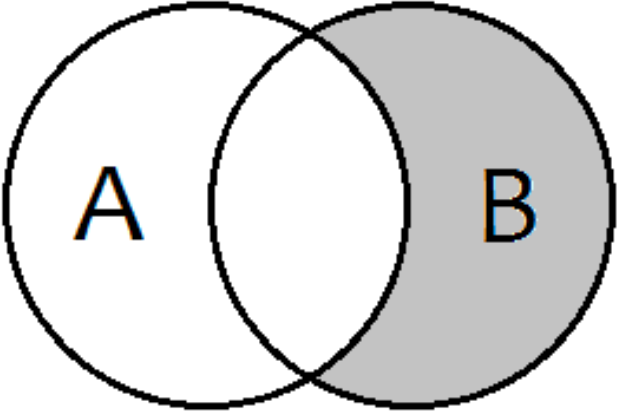
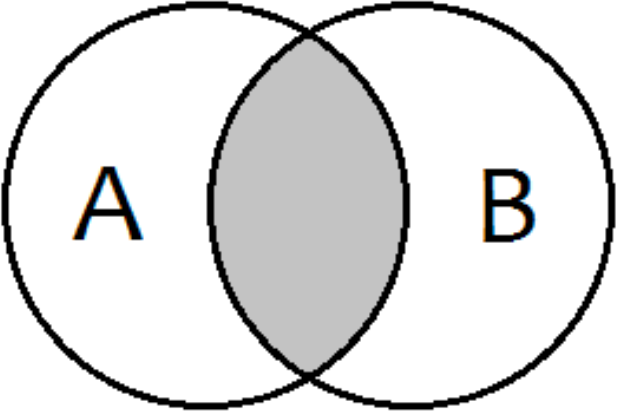
Номер сотрудника
325
567
225
976



COUNT(Номер сотрудника)
4

Объединения (joins)

- Left join
- Right Join
- Inner Join
- Full Join

SQL JOINS	
 <pre>SELECT * FROM TableA a LEFT JOIN TableB b ON a.Key = b.Key</pre>	
 <pre>SELECT * FROM TableA a LEFT JOIN TableB b ON a.Key = b.Key WHERE b.Key IS NULL</pre>	 <pre>SELECT * FROM TableA a FULL OUTER JOIN TableB b ON a.Key = b.Key</pre>
 <pre>SELECT * FROM TableA a RIGHT JOIN TableB b ON a.Key = b.Key</pre>	 <pre>SELECT * FROM TableA a FULL OUTER JOIN TableB b ON a.Key = b.Key WHERE a.Key IS NULL OR b.Key IS NULL</pre>
 <pre>SELECT * FROM TableA a RIGHT JOIN TableB b ON a.Key = b.Key WHERE a.Key IS NULL</pre>	 <pre>SELECT * FROM TableA a INNER JOIN TableB b ON a.Key = b.Key</pre>

Порядок выполнения запросов:

FROM Сначала обращаемся к таблицам

WHERE Фильтруем записи по условию

GROUP BY Выполняем агрегацию

HAVING Из агрегированных отбираем по условию

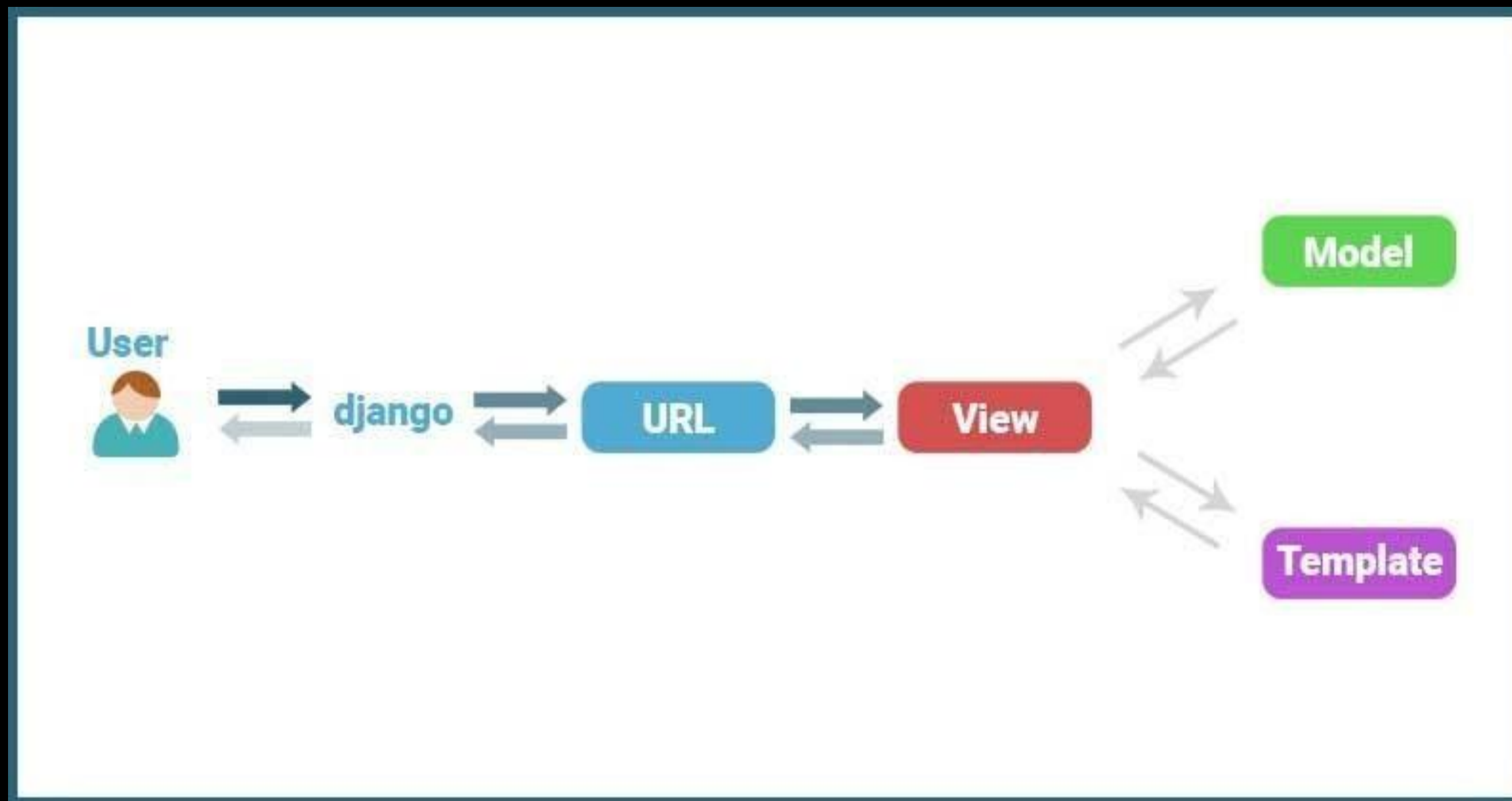
SELECT Формируем данные выборки

ORDER BY Упорядочиваем результат

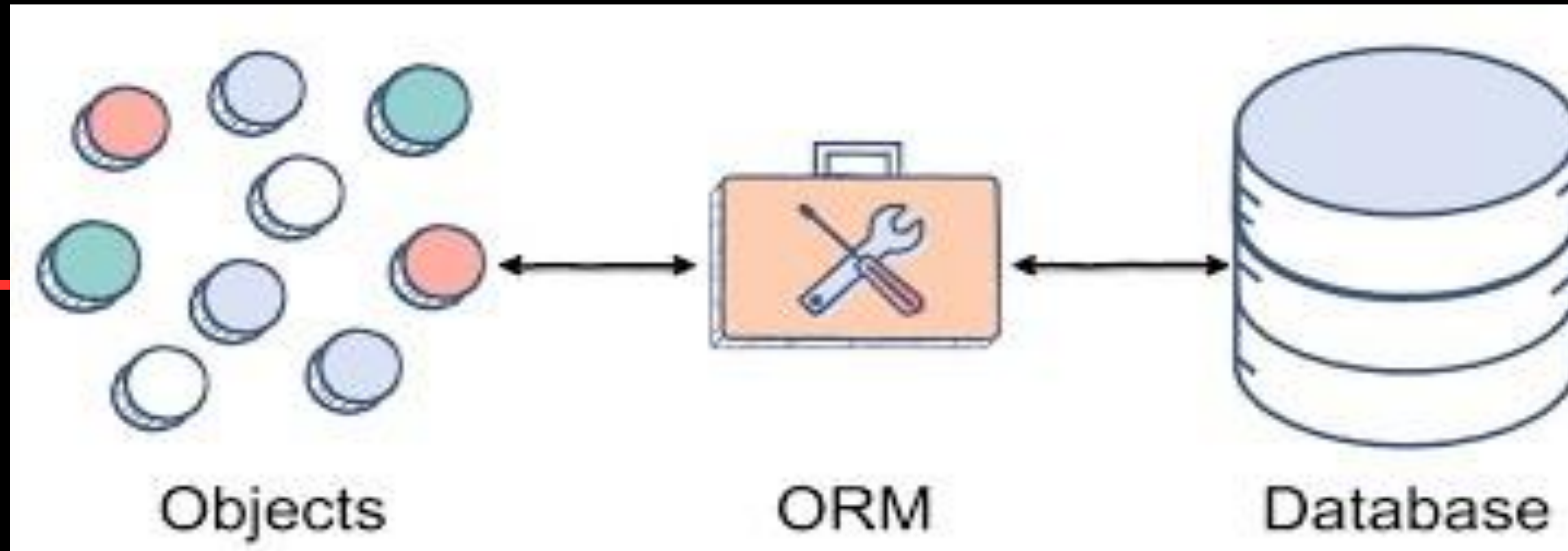
The image features a solid black background with several thin, white, straight lines. These lines are arranged in a way that suggests a larger, partially visible geometric shape, possibly a parallelogram or a series of connected triangles. The lines are positioned on the left and right sides of the frame, with one line extending from the top left towards the center and another from the bottom left towards the center. On the right side, a line extends from the top right towards the center, and another from the bottom right towards the center. The overall effect is minimalist and modern.

DjangoORM

Модель-шаблон-представление (MTV)



Object-Relational Mapping



Возможности Django

- ORM, API доступа к БД с поддержкой транзакций
- Встроенный интерфейс администратора
- Диспетчер URL на основе регулярных выражений
- Расширяемая система шаблонов с тегами и наследованием
- Система кеширования
- Интернационализация
- Подключаемая архитектура приложений
- “generic views” – шаблоны функций контроллеров
- Авторизация и аутентификация, подключение внешних пмодулей аутентификации LDAP, OpenID, ...
- Система фильтров (middleware) для построения дополнительных обработчиков запросов
- Библиотека работы с формами

Models

8

```
class Department(models.Model):
    name = models.CharField(max_length=100, verbose_name="Название подразделения")
    description = models.TextField(verbose_name="Описание")
    created_at = models.DateTimeField(auto_now_add=True, verbose_name="Дата создания")
    updated_at = models.DateTimeField(auto_now=True, verbose_name="Дата обновления")

    def __str__(self):
        return self.name

class Employee(models.Model):
    name = models.CharField(max_length=100, verbose_name="Имя")
    surname = models.CharField(max_length=100, verbose_name="Фамилия")
    patronymic = models.CharField(max_length=100, verbose_name="Отчество", null=True, blank=True)
    date_of_birth = models.DateField(verbose_name="Дата рождения")
    email = models.EmailField(verbose_name="Электронная почта", null=True, blank=True)
    phone = models.CharField(max_length=20, verbose_name="Телефон", null=True, blank=True)
    is_active = models.BooleanField(default=True, verbose_name="Сотрудник активен?")
    created_at = models.DateTimeField(auto_now_add=True, verbose_name="Дата создания")
    updated_at = models.DateTimeField(auto_now=True, verbose_name="Дата обновления")
    department = models.ForeignKey(Department, on_delete=models.CASCADE, related_name="employees",
                                   verbose_name="Подразделение")

    def __str__(self):
        return f'{self.name} {self.surname} {self.patronymic}'
```

structure_department

name	varchar(100)
description	text
created_at	datetime
updated_at	datetime
id	integer

department_id:id

structure_employee

name	varchar(100)
surname	varchar(100)
patronymic	varchar(100)
date_of_birth	date
email	varchar(254)
phone	varchar(20)
is_active	bool
created_at	datetime
updated_at	datetime
department_id	bigint
id	integer

Models. Типы полей

- CharField
- TextField
- EmailField
- GenericIPAddressField
- SlugField
- URLField
- UUIDField
- AutoField
- IntegerField
- DecimalField
- FloatField
- TimeField
- DateField
- DateTimeField
- DurationField
- FileField
- ImageField
- BinaryField
- JSONField

ORM methods

Метод	Описание
Model.objects.filter(id__in=[1,2,3])	Поиск информации
Model.objects.get(id=1)	Поиск информации
Model.objects.create(name="Название")	Создание записи
Model.objects.filter(id__in=[1,2,3]).update(name="Новое название")	Обновление записей
Model.objects.filter(id__in=[1,2]).delete()	Удаление записей