

Лекция 4

Django. ORM

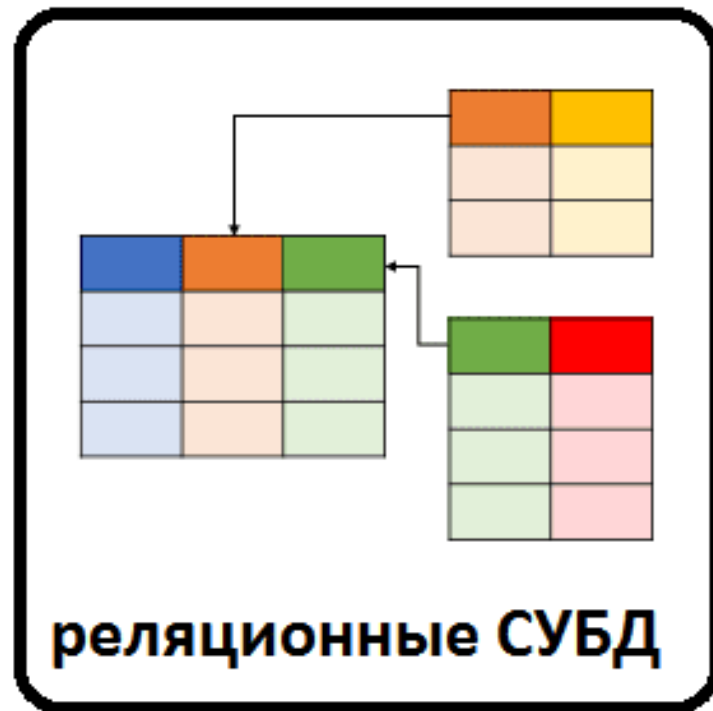
Разработка интернет приложений

Канев Антон Игоревич

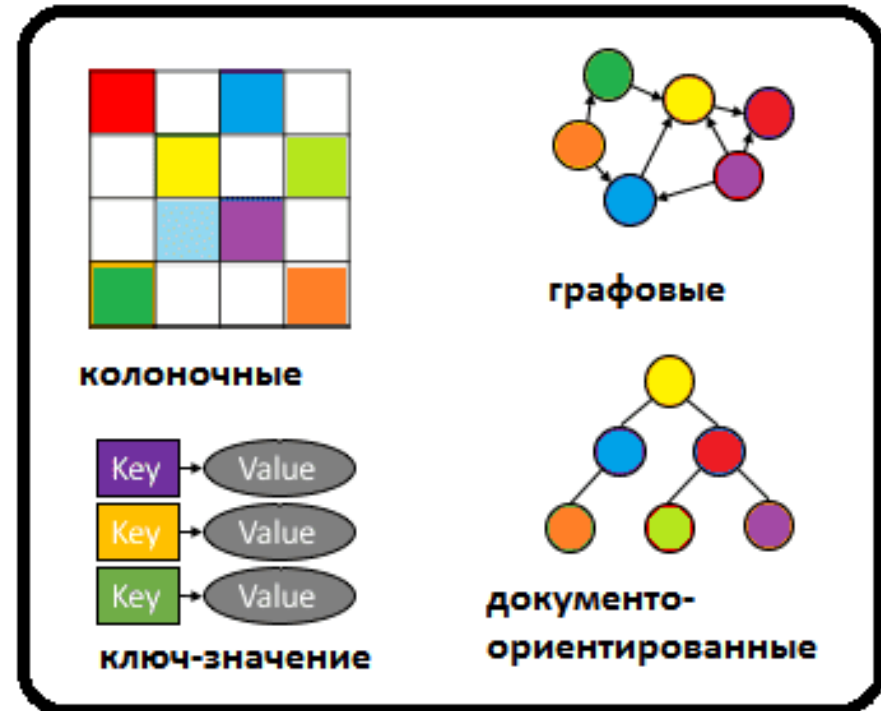
Виды баз данных

- Документо-ориентированные
- Реляционные
- in memory
- Графовые
- Ключ-значение

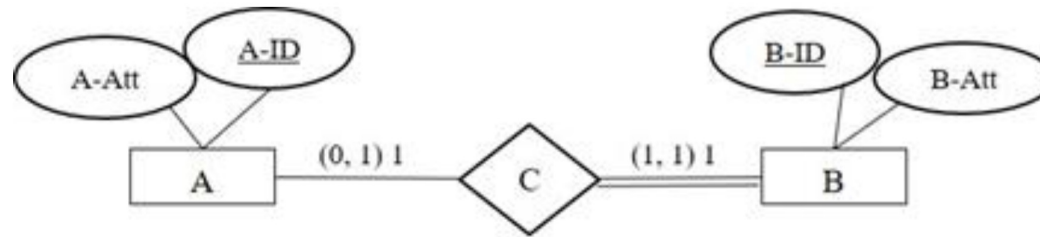
SQL



NoSQL

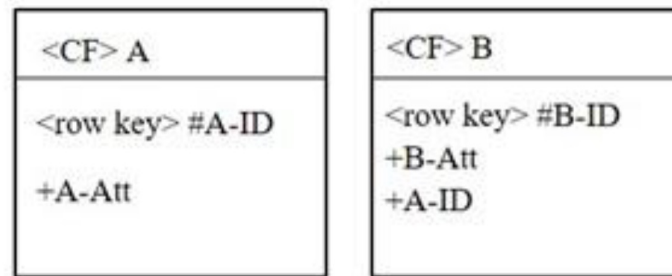


Виды баз данных

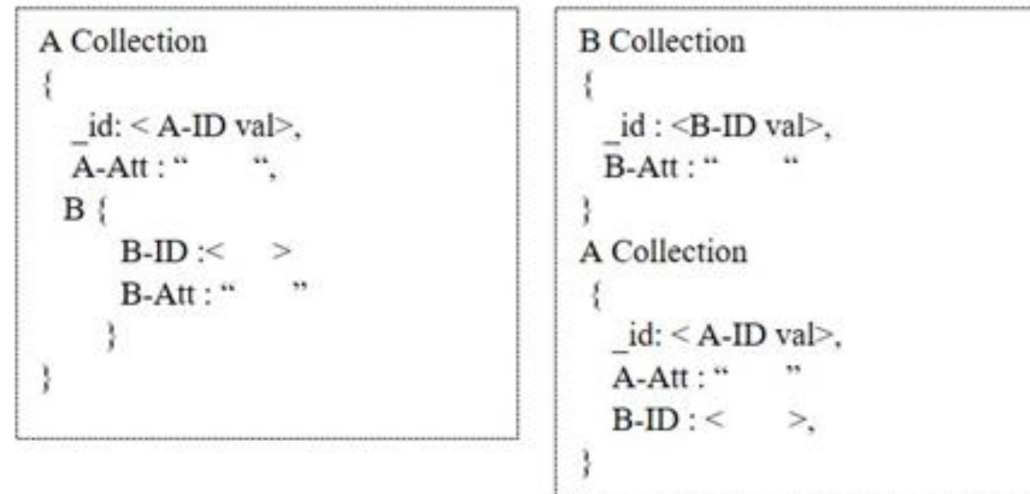


Tables:
 A (A-ID, A-Att)
 B (B-ID, B-Att, A-ID)

a. Relational Database ER and Tables



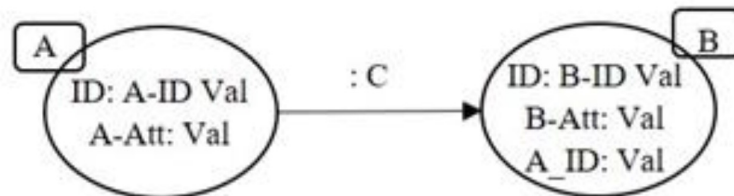
b. Column-based NoSQL Columns



1-Embedded document

2-Referencing document

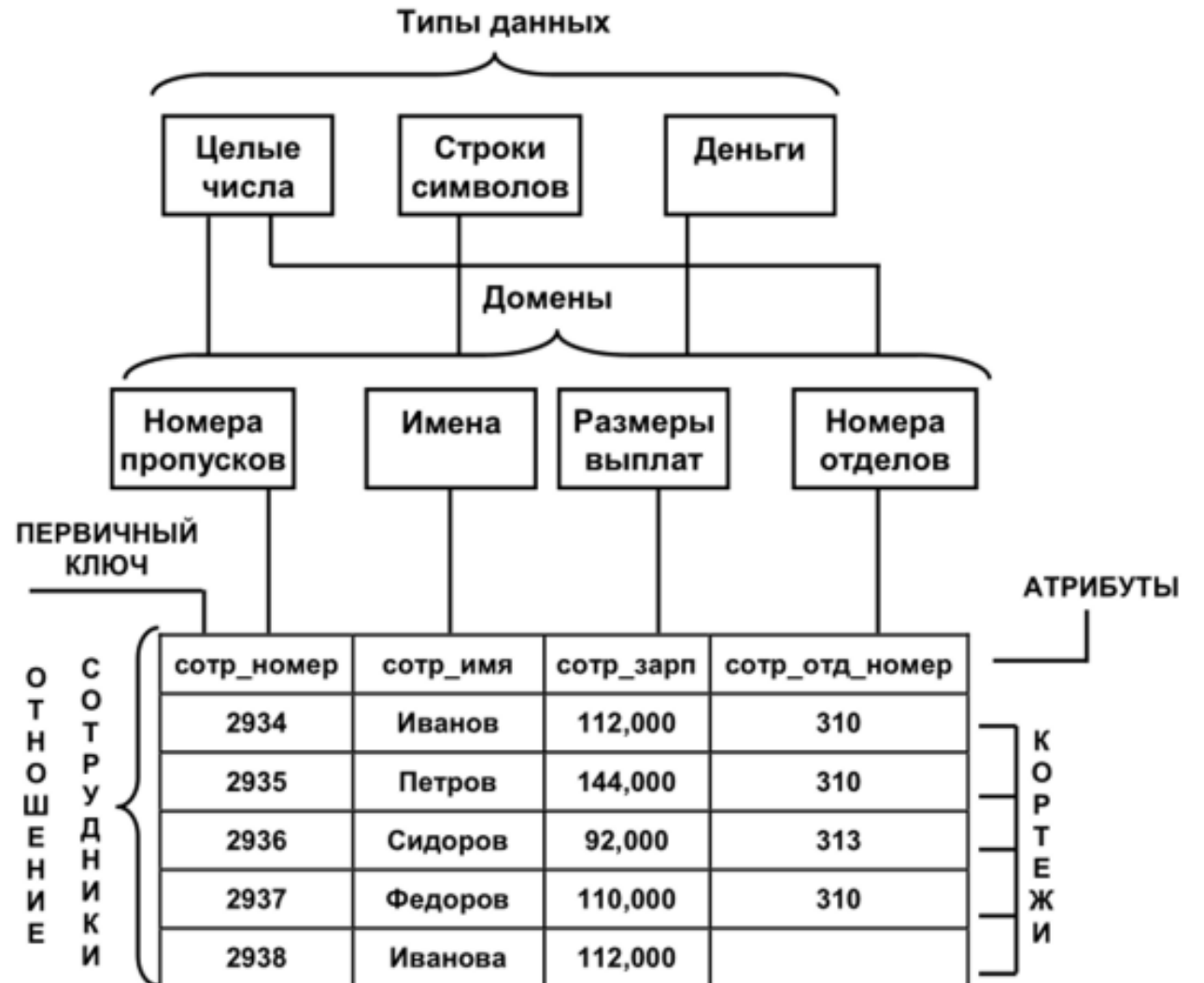
c. Document-based NoSQL Collections



d. Graph-based NoSQL Nodes

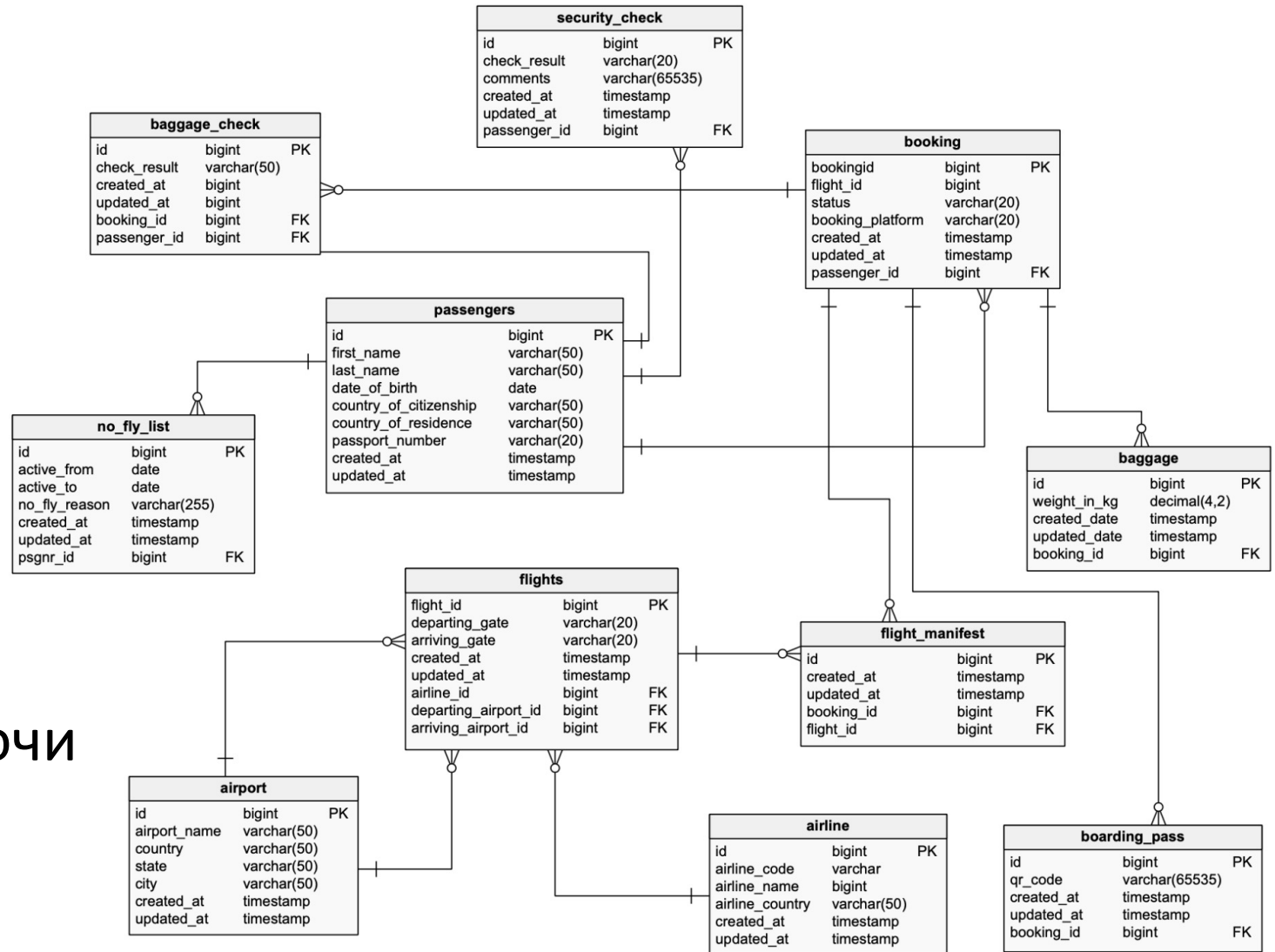
Реляционные базы данных

- Relation – отношение, связь
- Нормализация данных для устранения аномалий и избыточности



ER диаграмма

- Отношения - таблицы
- Атрибуты и типы данных
- Первичные и внешние ключи



Типы данных

- Дата
- Целочисленный
- Вещественный
- Строковый

	db2	mysql	openbase	oracle
:binary	blob(32768)	blob	object	blob
:boolean	decimal(1)	tinyint(1)	boolean	number(1)
:date	date	date	date	date
:datetime	timestamp	datetime	datetime	date
:decimal	decimal	decimal	decimal	decimal
:float	float	float	float	number
:integer	int	int(11)	integer	number(38)
:string	varchar(255)	varchar(255)	char(4096)	varchar2(255)
:text	clob(32768)	text	text	clob
:time	time	time	time	date
:timestamp	timestamp	datetime	timestamp	date

Типы данных

```
<event>
  <data name="SessionID">
    <value>CDE3D5</value>
  </data>
  <data name="NTUserName">
    <value>xyz</value>
  </data>
  <data name="NTDomainName">
    <value>ABC</value>
  </data>
  <data name="DatabaseName">
    <value>TestCube1</value>
  </data>
  <data name="ApplicationName">
    <value>Microsoft SQL Server Management Studio</value>
  </data>
  <data name="ServerName">
    <value>SERXYZ</value>
  </data>
  <data name="RequestID">
    <value>A737</value>
  </data>
  <action name="attach_activity_id_xfer" package="package0">
    <value>AE23</value>
  </action>
  <action name="attach_activity_id" package="package0">
    <value>CCCCB490</value>
  </action>
</event>
```

CustomerId	FirstName	LastName	Picture varbinary(max)
235	'John '	'Doe '	0x3B0E95AE3B292F0B...
236	'Sally '	'Smith '	0xF3000EEF2932002C...

- CLOB
- BLOB
- XML

SQL

- DDL – Data Definition Language
CREATE, ALTER, DROP

- DML – Data Manipulation Language
SELECT, INSERT, UPDATE, DELETE

- DCL – Data Control Language
GRANT, REVOKE

- TCL – Transaction Control Language
COMMIT, ROLLBACK, SAVEPOINT

```
SELECT DeptID, SUM(SaleAmount)
FROM Sales
WHERE SaleDate = '01-Jan-2000'
GROUP BY DeptID
HAVING SUM(SaleAmount) > 1000
```


Форматы дат

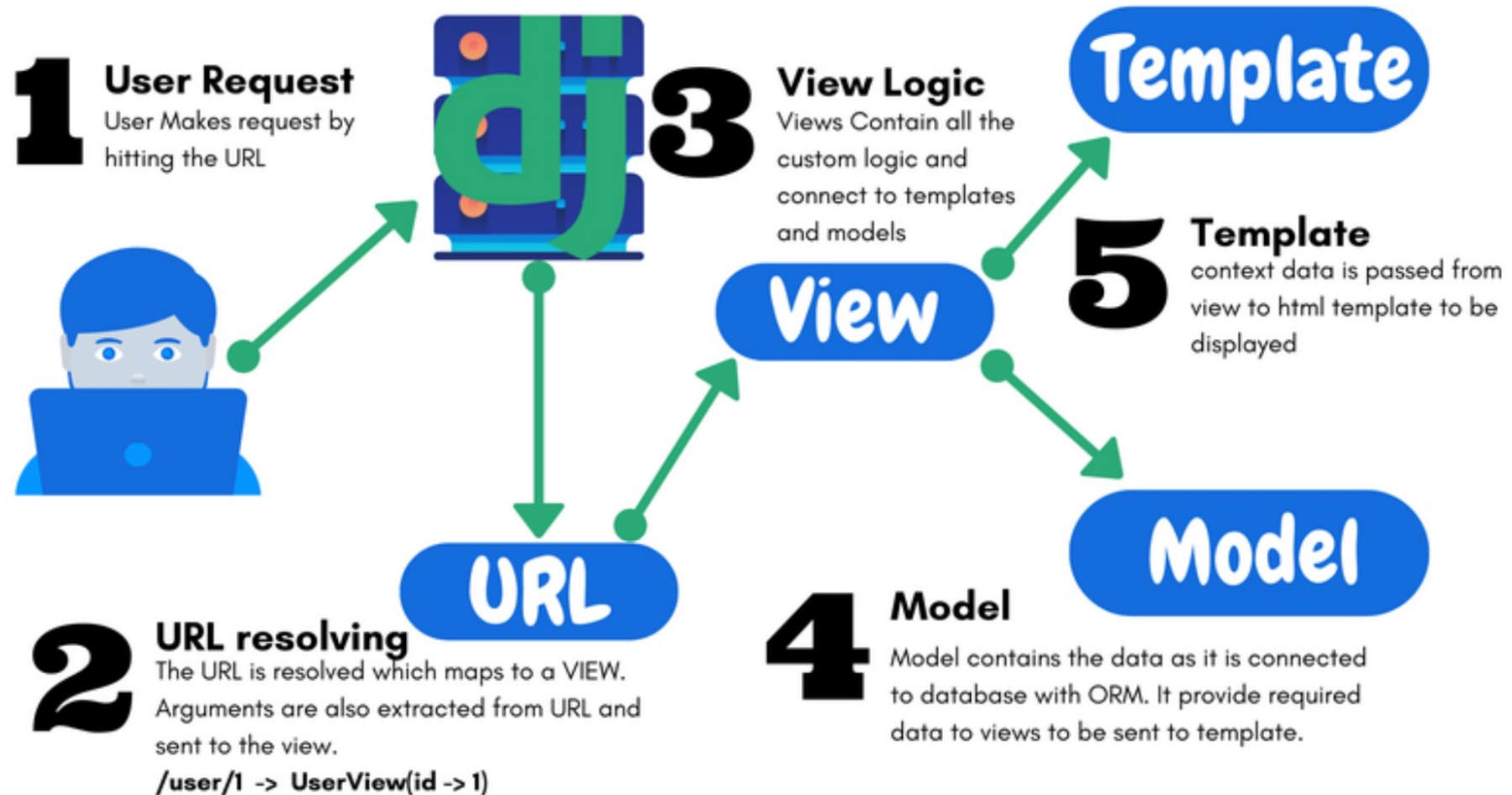
- Множество форматов дат и времени
- Ошибки в обработке даты в разных системах

MON	Yes	Abbreviated name of month.
MONTH	Yes	Name of month.
PM P.M.	Yes	Meridian indicator with or without periods.
Q		Quarter of year (1, 2, 3, 4; January - March = 1).
RM	Yes	Roman numeral month (I-XII; January = I).
RR	Yes	Lets you store 20th century dates in the 21st century using only two digits. See Also: "The RR Datetime Format Element"
RRRR	Yes	Round year. Accepts either 4-digit or 2-digit input. If 2-digit, provides the same return as RR. If you do not want this functionality, then enter the 4-digit year.
SS	Yes	Second (0-59).
SSSSS	Yes	Seconds past midnight (0-86399).
TS	Yes	Returns a value in the short time format. Makes the appearance of the time components (hour, minutes, and s

Достоинства и недостатки

- Борьба с избыточностью и аномалиями
- Высокоуровневый язык запросов
- Необходимость операций соединения (JOIN), которые замедляют скорость
- В реальности не нормализованные данные
- Теорема CAP – согласованность, доступность, устойчивость к разделению

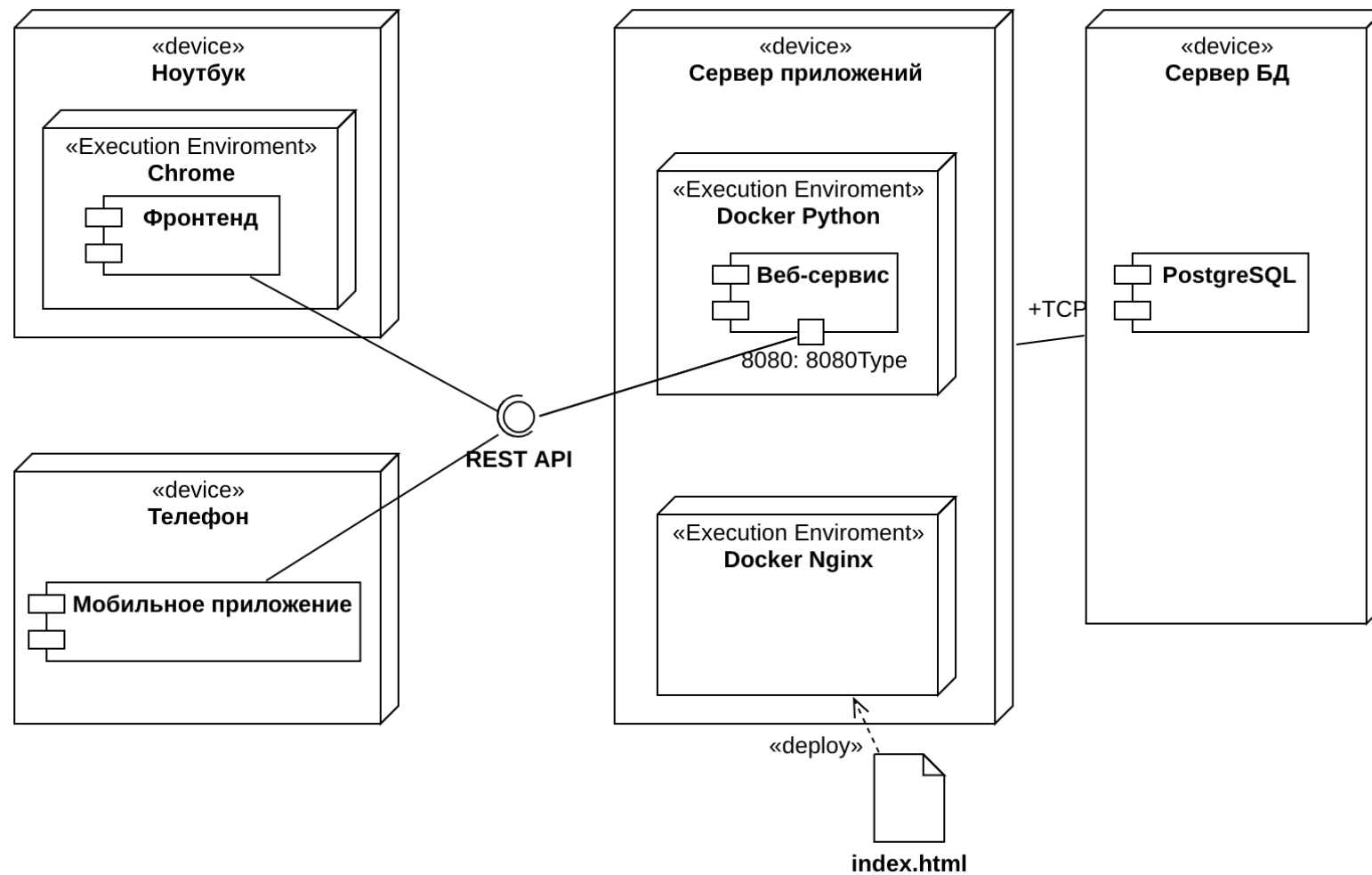
Фреймворк Django. MVC



Традиционный серверный фреймворк

- Статические файлы (статические HTML-документы, CSS, изображения, сценарии JavaScript и т.д.).
- Контроллеры (обработчики событий пользовательских действий).
- Модели (взаимодействие с БД).
- Представления (view). Шаблоны, генерирующие HTML-страницы и другое динамическое содержимое.
- Конфигурирование фреймворка: действия при запуске приложения, конфигурирование пользовательских сеансов (сессий), переписывание URL (привязка URL к контроллерам), безопасность (аутентификация и авторизация), кэширование, балансировка нагрузки, IOC / DI.
- Утилиты командной строки для управления фреймворком.
 - Скаффолдинг (создание структуры проекта, генерация кода контроллеров и представлений на основе моделей, генерация кода приложения на основе специализированных описаний, генерация форм ввода и редактирования данных во время работы приложения).
 - Миграции (изменение структуры базы данных на основе моделей).

Трехзвенная архитектура



Курсоры

- `import MySQLdb`
- `db = MySQLdb.connect(host="localhost", user="dbuser",
passwd="123", db="first_db")`
- `c=db.cursor()`
- `c.execute("INSERT INTO books (name, description) VALUES (%s, %s);",
('Book', 'Description'))`
- `db.commit()`
- `c.close()`
- `db.close()`

Подключение к базе данных

- DATABASES = {
- 'default': {
- 'ENGINE': 'django.db.backends.mysql',
- 'NAME': 'first_db',
- 'USER': 'dbuser',
- 'PASSWORD': '123',
- 'HOST': 'localhost',
- 'PORT': 3306, # Стандартный порт MySQL
- 'OPTIONS': {'charset': 'utf8'},
- 'TEST_CHARSET': 'utf8', } }

Модель

- `from django.db import models`
- `# Create your models here.`
- `class Book(models.Model):`
 - `name = models.CharField(max_length=30)`
 - `description = models.CharField(max_length=255)`
 - `class Meta:`
 - `managed = False`
 - `db_table = 'books'`

ORM

```
class Recipe(models.Model):  
    name = models.CharField(max_length=150)  
    image = models.ImageField(upload_to="images/")
```



ORM



Database



```
CREATE TABLE recipe (  
    id int NOT NULL PRIMARY KEY,  
    name VARCHAR(150),  
    image BLOB,  
);
```

Миграция

- `$ python manage.py migrate`
- Operations to perform:
 - Apply all migrations: admin, auth, contenttypes, polls, sessions
- Running migrations:
 - Rendering model states... DONE
 - Applying polls.0001_initial... OK

Миграции

Внесение изменений в модель:

- Изменение модели (models.py).
- Запуск команды `python manage.py makemigrations` для создания миграций этих изменений
- Выполнение команды `python manage.py migrate` для применения этих изменений в базе данных.

QuerySet

- `>>> from blog.models import Blog, Entry`
- `>>> entry = Entry.objects.get(pk=1)`
- `>>> cheese_blog = Blog.objects.get(name="Cheddar Talk")`
- `>>> entry.blog = cheese_blog`
- `>>> entry.save()`