### БД. SQL

* Что такое реляционная БД. Отличие relation и relationship
  + Реляционная база данных – это набор данных с предопределенными связями между ними. Эти данные организованы в виде набора таблиц, состоящих из столбцов (атрибутов) и строк (кортежей). В таблицах хранится информация об объектах, представленных в базе данных. В каждом столбце таблицы хранится определенный тип данных, в каждой ячейке – значение атрибута. Каждая строка таблицы представляет собой набор связанных значений, относящихся к одному объекту или сущности.
  + relation - это таблица
  + relation ship - связь между таблицами
  + <https://elearning.sumdu.edu.ua/free_content/lectured:8f6f49a23bd675196ec11fa19fedb2394e1a09be/latest/43131/index.html>
* Foreign key. Зачем нужен, как работает, что дает. Можно ли связывать таблицу без Foreign key? (да) Нужно ещё знать, что есть БД, в которых нет механизма Foreign Key и таблицы связывать можно вручную.
  + Внешние ключи позволяют установить связи между таблицами. Внешний ключ устанавливается для столбцов из зависимой, подчиненной таблицы, и указывает на один из столбцов из главной таблицы. Как правило, внешний ключ указывает на первичный ключ из связанной главной таблицы.
* ON DELETE и ON UPDATE
  + С помощью выражений ON DELETE и ON UPDATE можно установить действия, которые выполняются соответственно при удалении и изменении связанной строки из главной таблицы. В качестве действия могут использоваться следующие опции:
  + <https://djbook.ru/rel1.9/ref/models/fields.html#django.db.models.ForeignKey.on_delete>
  + CASCADE (В Django CASCADE): автоматически удаляет или изменяет строки из зависимой таблицы при удалении или изменении связанных строк в главной таблице.
  + SET NULL (SET\_NULL): при удалении или обновлении связанной строки из главной таблицы устанавливает для столбца внешнего ключа значение NULL. (В этом случае столбец внешнего ключа должен поддерживать установку NULL)
  + RESTRICT (PROTECT): отклоняет удаление или изменение строк в главной таблице при наличии связанных строк в зависимой таблице.
  + NO ACTION (DO\_NOTHING): то же самое, что и RESTRICT.
  + SET DEFAULT (SET\_DEFAULT): при удалении связанной строки из главной таблицы устанавливает для столбца внешнего ключа значение по умолчанию, которое задается с помощью атрибуты DEFAULT. Несмотря на то, что данная опция в принципе доступна, однако движок InnoDB не поддерживает данное выражение.
  + В Django ещё есть SET() - Устанавливает [ForeignKey](https://djbook.ru/rel1.9/ref/models/fields.html#django.db.models.ForeignKey) в значение указанное в [SET()](https://djbook.ru/rel1.9/ref/models/fields.html#django.db.models.SET).
* Виды связей (1к1, 1кМ, МкМ). Как сделать эти связи на уровне БД
  + МкМ через дополнительную 3ю таблицу, где будут лежать два FOREIGN KEY на соответственно ID из первой таблицы и ID и второй
  + Мк1(1 ко многим) FOREIGN KEY
  + как делать 1 к 1. Синтаксис такой же, что и в случае с 1 ко многим, только в 2 таблице ID должно быть UNIQUE
* Виды Join’ов. Inner, Right, Left, Natural, Full.
  + [INNER, LEFT, RIGHT (ссылка)](https://anton-pribora.ru/articles/mysql/mysql-join/)
  + [Inner, Right, Left, Natural, Full (ссылка)](http://s.chervach.com/naglyadnoe-obyasnenie-printsipa-obedineniya-tablits-v-mysql/)
* Две таблицы, связанные 1кМ связью, как сделать запрос, который из одной таблицы выведет имя человека, из другой адрес его.
  + Первая таблица persons: id name. Вторая addresses: id address person\_id
  + SELECT name, address FROM persons INNER JOIN addresses ON persons.id=addresses.person\_id;
* Есть таблица people, есть address. Какой запрос нужно сделать, чтобы вывести людей, у которых нет адреса
  + Первая таблица persons: id name. Вторая addresses: id address person\_id
  + SELECT name, address FROM persons LEFT JOIN addresses ON persons.id=addresses.person\_id WHERE addresses.person\_id is NULL;
* А если у некоторых людей два адреса, как построить SELECT запросы между двумя таблиц, чтоб вывелись только люди с двумя адресами.
  + Та же таблица
  + SELECT name, address FROM persons INNER JOIN addresses ON persons.id=addresses.person\_id GROUP BY name HAVING count(name) >= 2;
* Две таблицы, связанные МкМ связью, как сделать запрос, который из одной таблицы выведет имя человека, из другой адрес его.
  + Первая таблица persons: id name. Вторая addresses: id address. Третья связующая person\_address: person\_id, address\_id
  + SELECT name, address FROM persons p, addresses a, person\_address pa WHERE p.id=pa.person\_id AND a.id=pa.address\_id;
  + <https://toster.ru/q/11916>
* Две таблицы, связанные МкМ связью, как сделать запрос, который из одной таблицы выведет имя человека, из другой адрес его. Вывести только те записи, у которых один человек связан с одним адресом.
  + Та же таблица, что и в вопросе выше
  + SELECT name, address FROM persons p, addresses a, person\_address pa WHERE p.id=pa.person\_id AND a.id=pa.address\_id GROUP BY name HAVING COUNT(name) = 1;

### Django

* Content Type зачем нужен. Content Type на уровне БД как организован
  + <https://djbook.ru/rel1.5/ref/contrib/contenttypes.html>
  + <https://aliev.me/prosto-o-django-content-types-framework/>
* Что такое GenericForeignKey на уровне БД. Как устроен, как работает. Зачем нужен?
  + <https://djbook.ru/rel1.5/ref/contrib/contenttypes.html#django.contrib.contenttypes.generic.GenericForeignKey>
  + <https://djbook.ru/examples/88/>
* Manager’ы. Кастомные (свои) менеджеры. Зачем их делать. Как их делать. Можно ли переопределить дефолтный менеджер objects и зачем,
* а когда этого делать не стоит
  + <https://djbook.ru/rel1.7/topics/db/managers.html>
  + Можно переопределить и дефолтный. Но делать так не желательно, т.к. это неявно и пользователь (или другая библиотека) ожидает от дефолтного менеджера определенное поведение. А если оно у вас будет переопределено))
* Select\_related, prefetch\_related. Что по дефолту включено? (ничего) Что такое параметр depth (это глубина). Что по умолчанию?
  + <https://medium.com/@lucasmagnum/djangotip-select-prefetch-related-e76b683aa457>
  + По умолчанию этот функционал выключен
* Как можно писать view в Django. Есть два варианта: функциями или классами (class-based view). Какие преимущества у одного и второго варианта, какие минусы. Приведите пример, когда что использовать
  + Функции: если у вас нужно обработать только один url и одно действие – функция будет проще. Зачем делать большой класс)
  + А вообще лучше и особенно для большого функционала (для того чтобы в будущем возможность расширяться была) нужно использовать класс. Это ООП, это проще читать, это проще расширять и отлаживать.

### Сеть

* Отличии http от https. Когда можно не использовать http? (Всегда нужно шифровать и юзать https)
  + В https есть защита – шифрование всех передаваемых данных с помощью ssl или tls. В http её нет.
  + Даже если на сайте просто какой-то контент и ничего важного юзер не передаёт. Нужно юзать шифрацию. Т.к. атака man-in-the-middle может быть. Надо про эту атаку знать. Вот, например, ты подключился к простому сайту с котиками, где больше ничего важного нет. Просто контент – картинки, которые никак не важны злоумышленнику. А он вдруг перехватил твой запрос и в страницу ответа встроил небольшой js-скрипт, который начал майнить на твоём компе. Упс)

### Python

* В чём отличие Python 2 от Python 3.
  + <https://pythonworld.ru/osnovy/python2-vs-python3-razlichiya-sintaksisa.html>
* Зачем наследоваться от object классам
  + В версии питона 2.2 -2.7 чтобы создать класс нового стиля.
  + Разница между классами нового и старого стиля:
  + <http://qaru.site/questions/1498090/why-does-inheriting-from-object-make-a-difference-in-python>
  + <https://riptutorial.com/ru/python/example/1402/new-style-vs--old-style-classes>
* В чём разница между версиями питона 3.6 и 3.7 (или, например, что нового в версии 3.8?)
  + Вопрос на то, следишь ли ты за изменениями в версиях питона))
  + <https://python-scripts.com/python-3-7#timing-precision>
* Магические методы. Назови парочку
  + \_\_init\_\_, \_\_getitem\_\_, \_\_str\_\_ и пр.
* Зачем нужна конструкция if \_\_name\_\_ == ‘’\_\_main\_\_’’:
  + [https://ru.stackoverflow.com/questions/515852/Что-делают-if-name-main](https://ru.stackoverflow.com/questions/515852/%D0%A7%D1%82%D0%BE-%D0%B4%D0%B5%D0%BB%D0%B0%D1%8E%D1%82-if-name-main)
  + Чтобы код, который после этого if идёт, выполнялся только в том случае, когда данный .py файл запущен для выполнения. Этот код не выполниться, когда мы данный файл заимпортим в другой модуль. Такая вот защита.