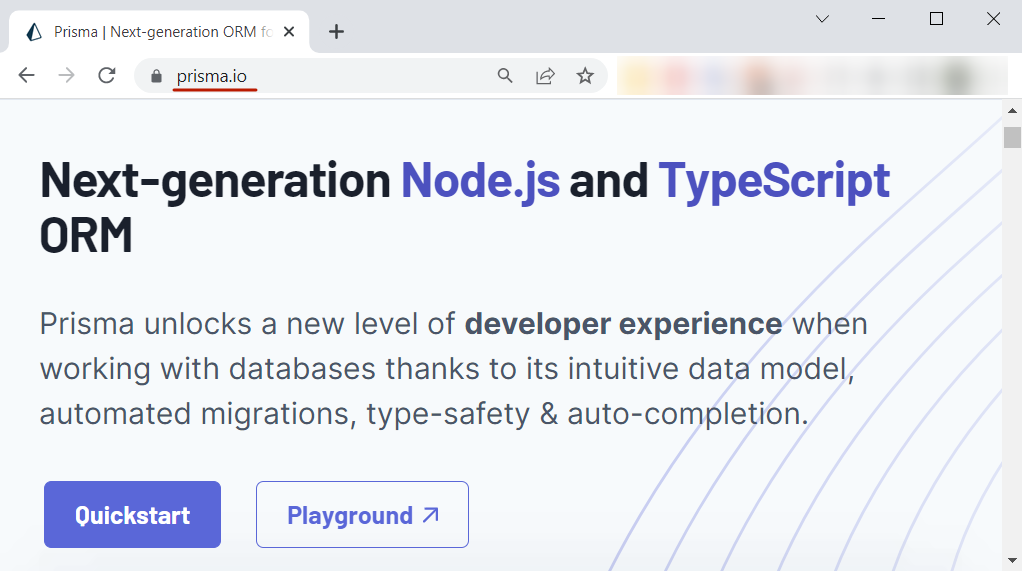
Лекция 17

ПСКП, ПОИТ-3

**Node.js: Prisma**

1. **Prisma:** это ORM нового поколения с открытым исходным кодом для Node.js и TypeScript. Она состоит из следующих инструментов:

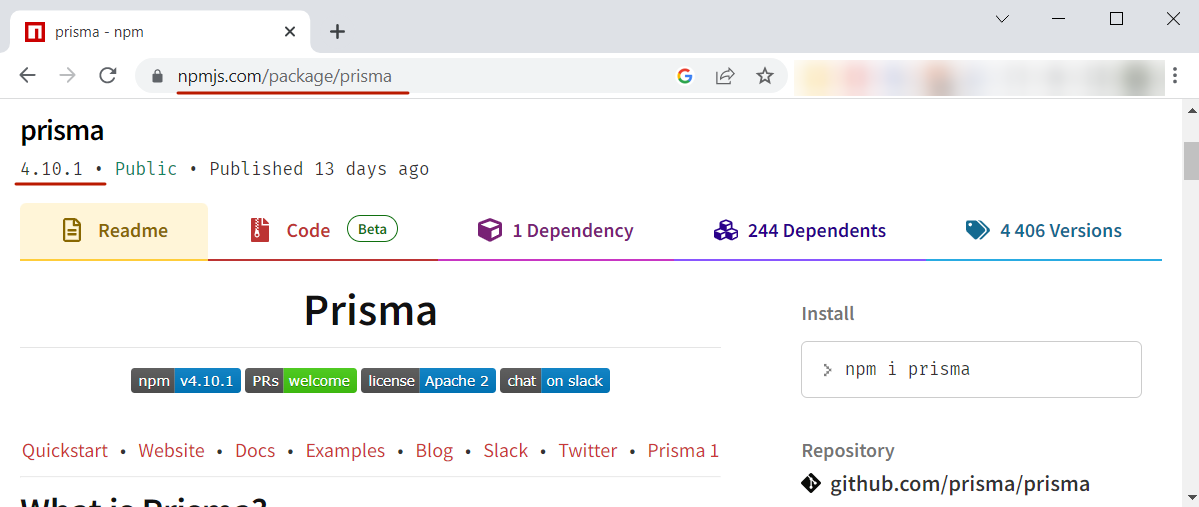
* **Prisma Client**: автогенерируемый и типобезопасный клиент базы данных;
* **Prisma Migrate**: система миграций;
* **Prisma Studio**: пользовательский интерфейс для просмотра и редактирования данных.

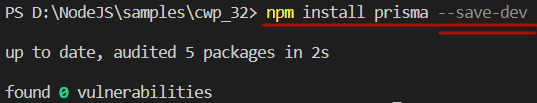


1. **Prisma:** основные характеристики:

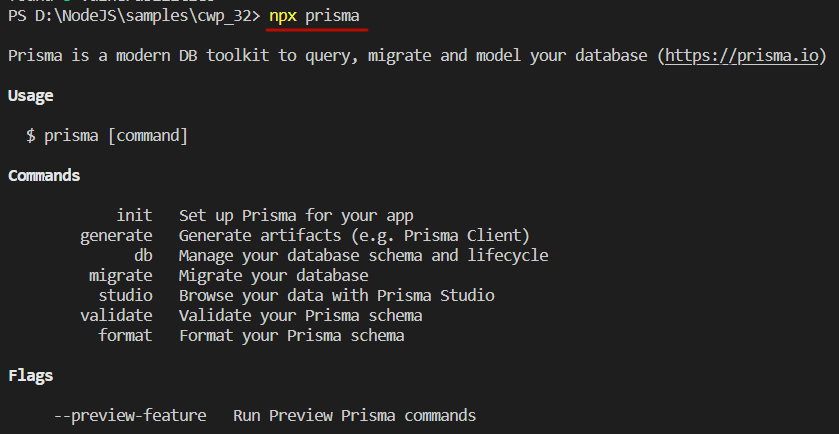
* написан на Rust;
* реализует паттерн Data Mapper;
* поддерживаемые языки: TypeScript, JavaScript;
* поддерживаются MySQL, PostgreSQL, MSSQL и SQLite (поддержка MongoDB, CockroachDB и PlanetScale в предварительной версии);
* предоставляет типобезопасный API (TypeScript);
* имеет набор инструментов для работы с базами данных (отправка запросов, моделирование, миграции, прототипирование, data seeding, студия для просмотра и изменения);
* способен генерировать определение схемы базы данных и клиентский код на основе структуры базы данных;
* поддерживает различные параметры запросов (фильтрация, сортировка, группировка, пагинация и др.);
* документация: <https://www.prisma.io/docs>
* Prisma Client API: [https://www.prisma.io/docs/ reference/api-reference/prisma-client-reference](https://www.prisma.io/docs/%20reference/api-reference/prisma-client-reference)

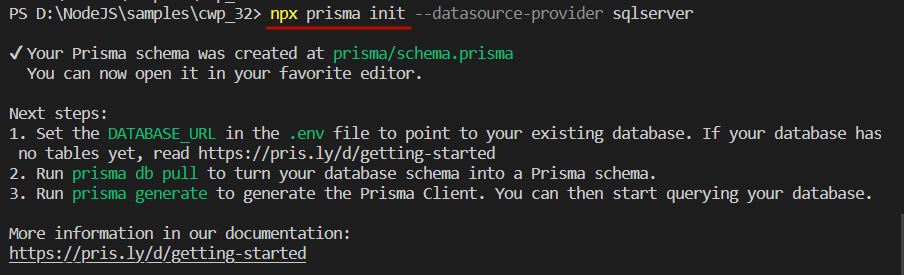
1. **Prisma:** npm install prisma –save-dev



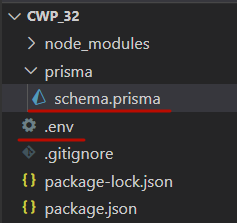
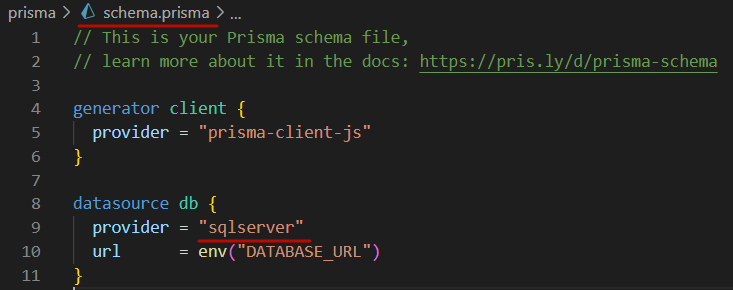


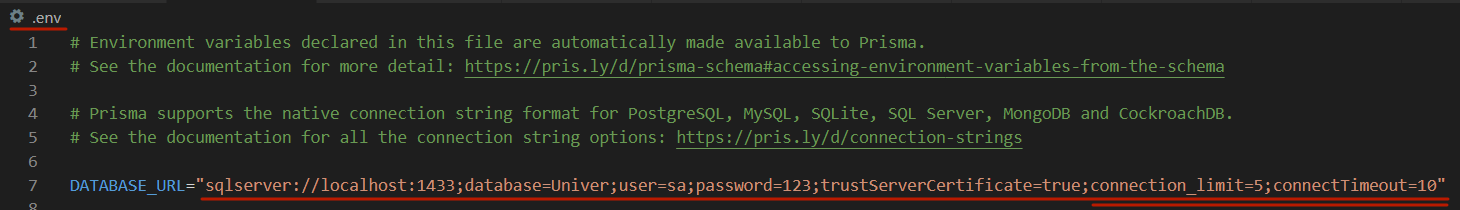
1. **Prisma:** npx – это инструмент, который нужен для упрощения использования утилит и исполняемых файлов.
2. **Prisma:** настройка проекта Prisma, database first
3. Инициализация проекта:



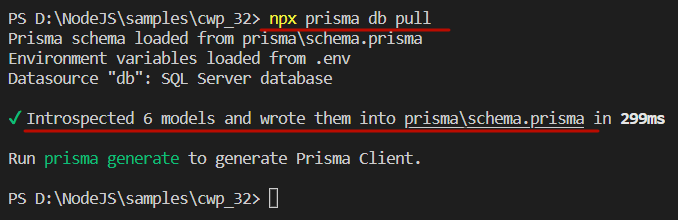


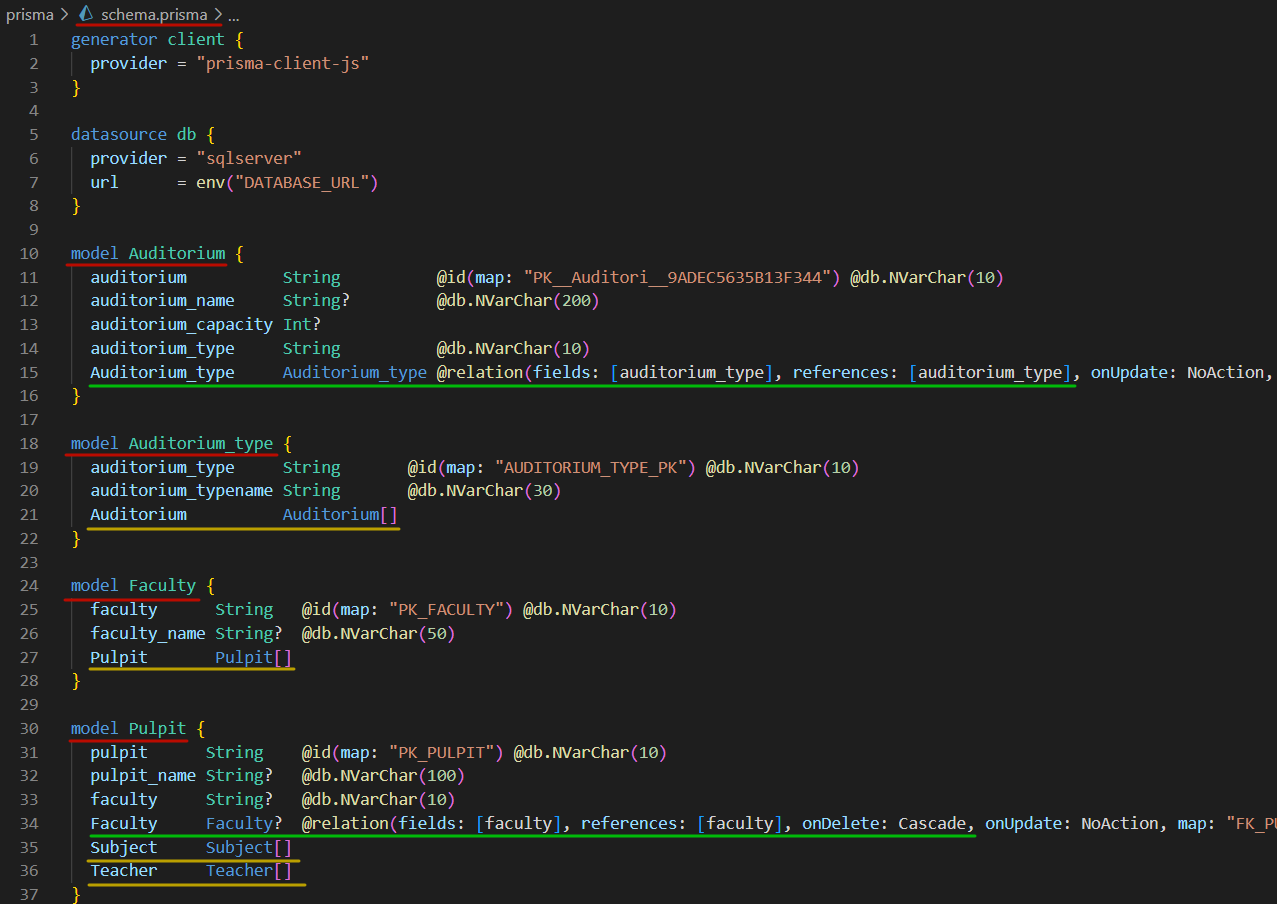
1. Настройка связи с БД:

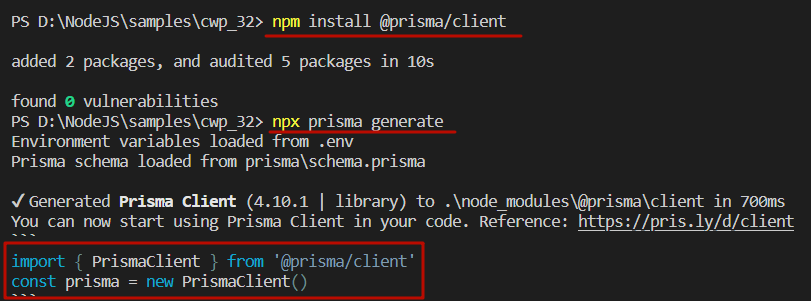


1. Генерация моделей:

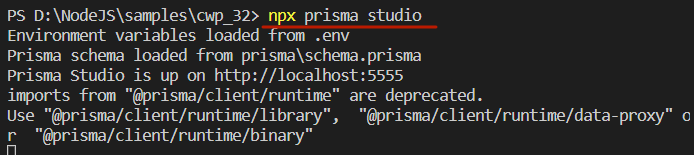


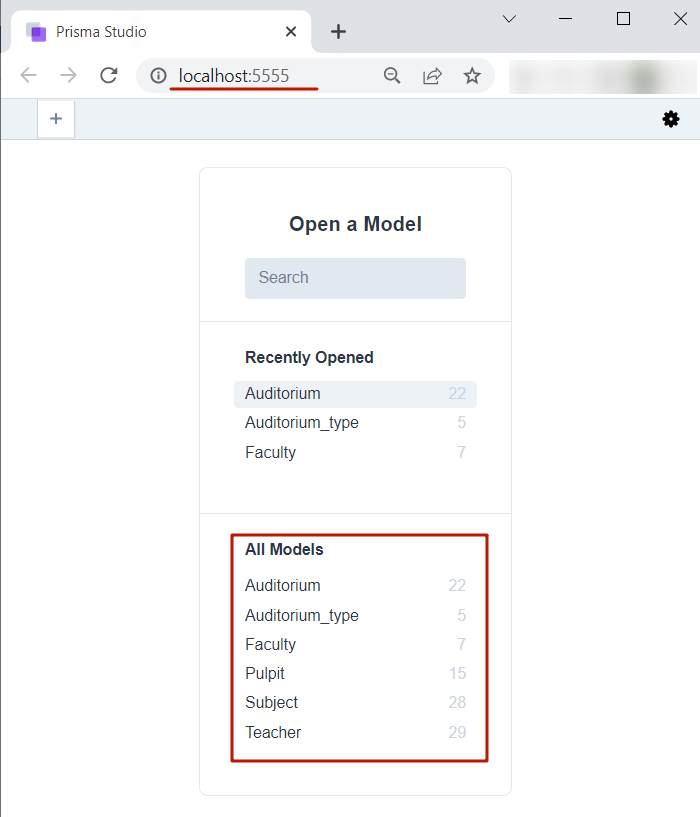


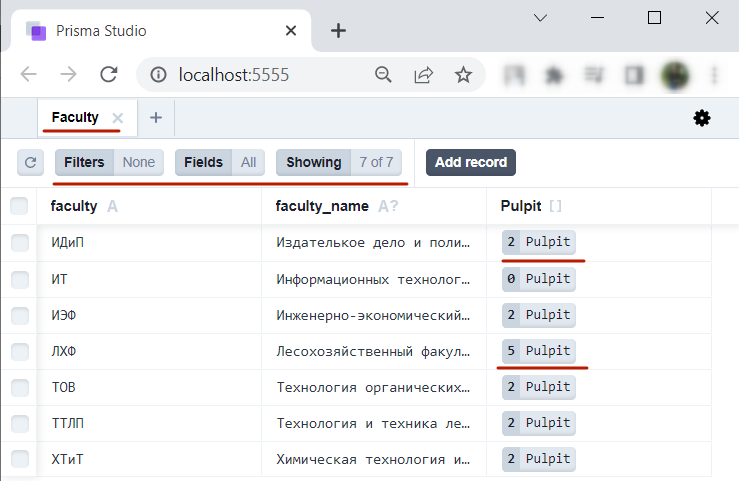
1. Генерация клиента:

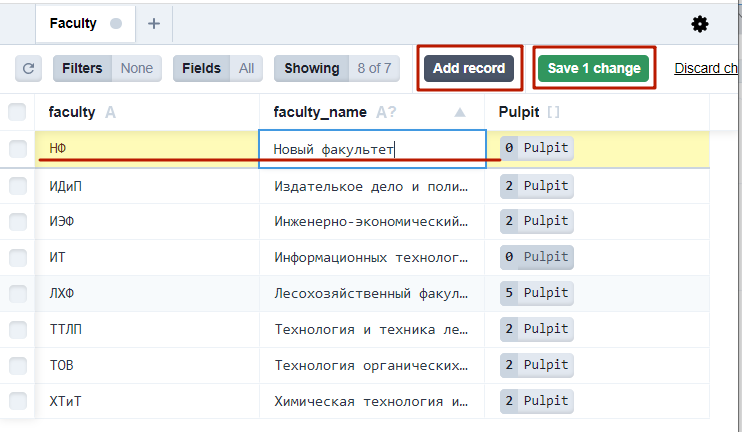


1. **Prisma:** Prisma Studio



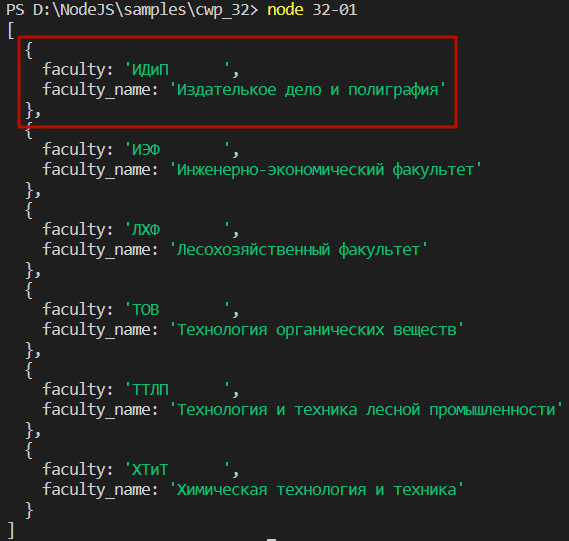




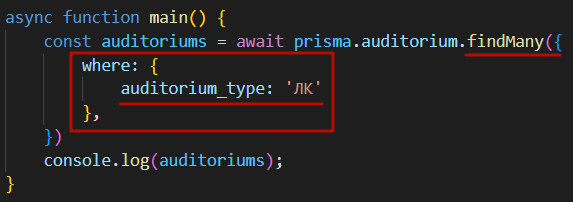


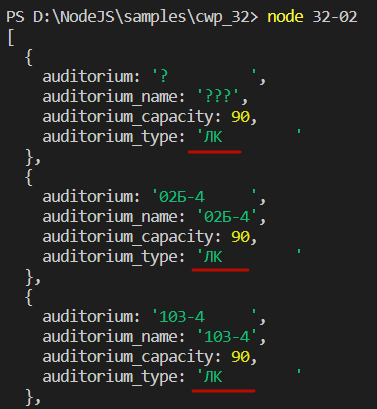
1. **Prisma:** client, findMany, $disconnect



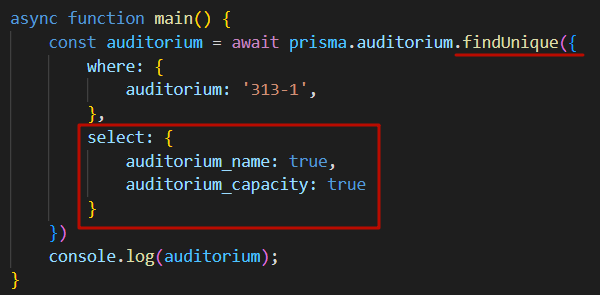


1. **Prisma:** findMany, where (критерий выборки)



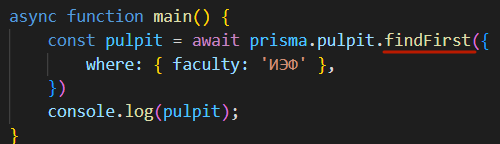


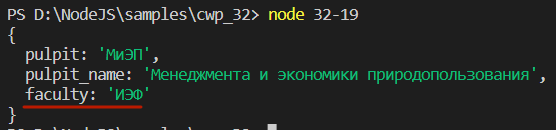
1. **Prisma:** findUnique (findUniqueOrThrow), select (набор полей модели)

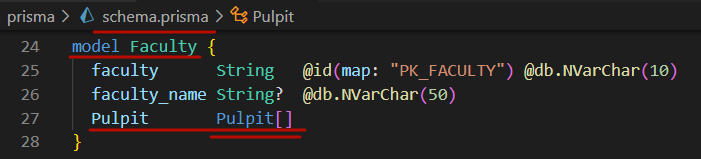


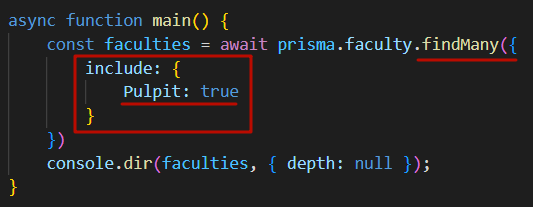
http://dl4.joxi.net/drive/2023/02/21/0030/0226/2023650/50/b6b522c8ff.png

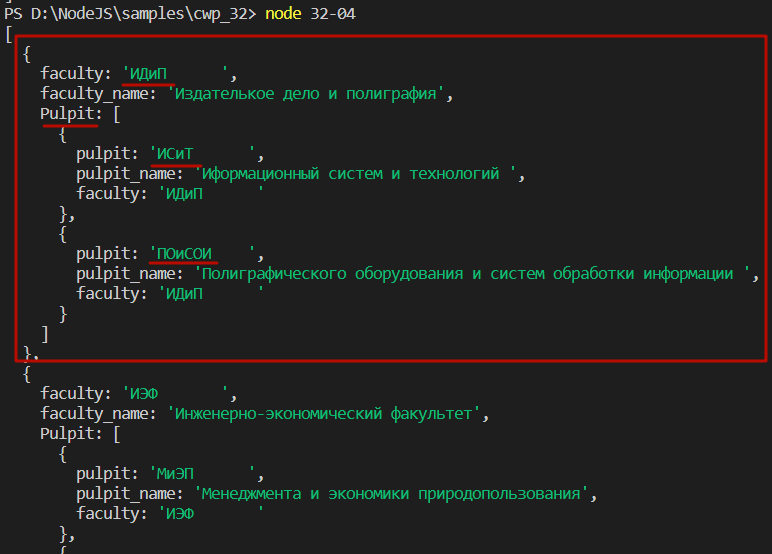
1. **Prisma:** findFirst (findFirstOrThrow)



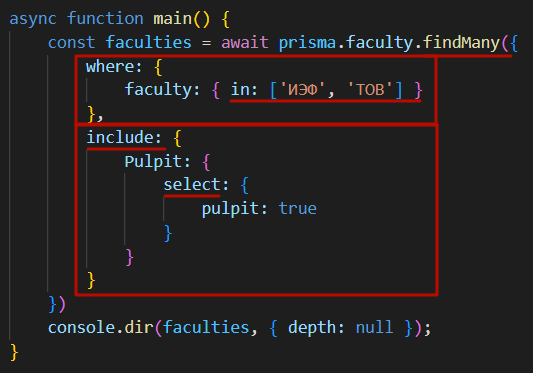


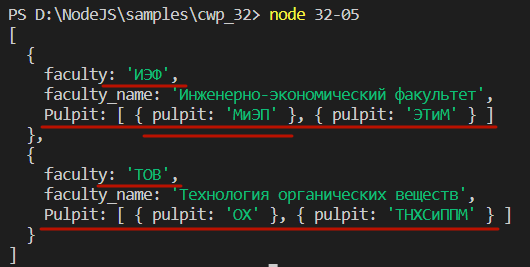
1. **Prisma:** include (включает связанные записи, join)  
   



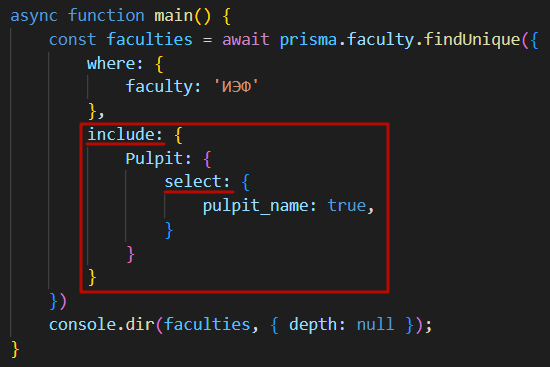


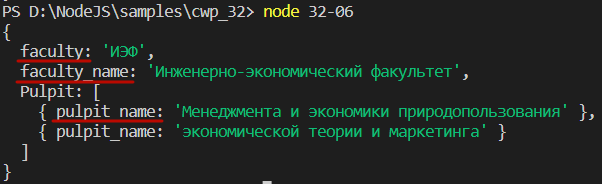
1. **Prisma:** where, in, include, select



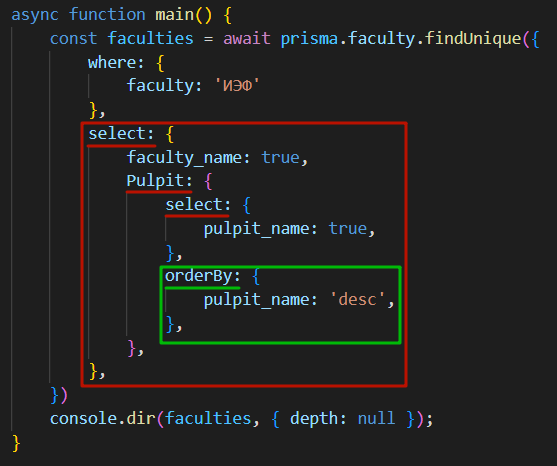


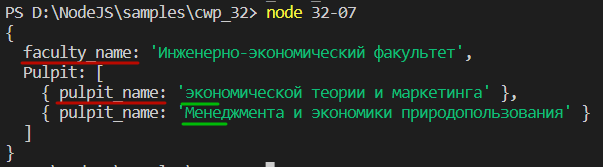
1. **Prisma:** select полей только связанных записей (include => select)



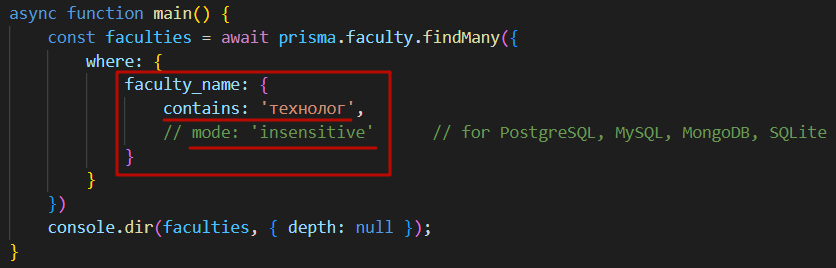


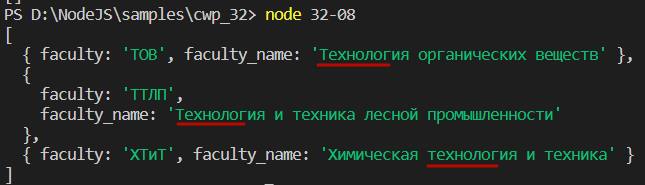
1. **Prisma:** select полей модели и полей связанных записей (вложенный, select => select, в таком случае include не нужен), orderBy (сортировка по возрастанию или убыванию)



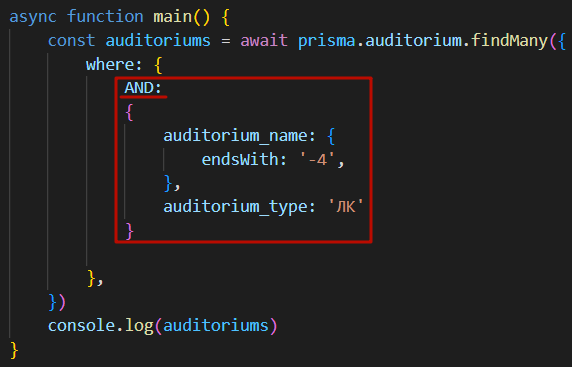


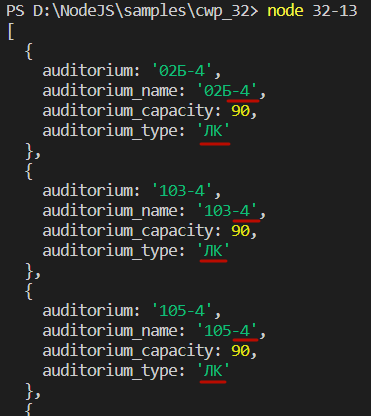
1. **Prisma: *filter conditions & operators*,** where => contains (содержимое поля)

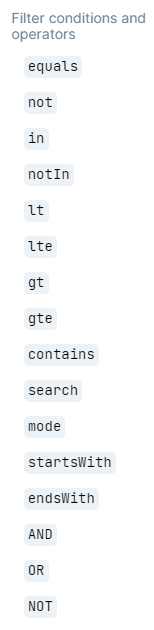
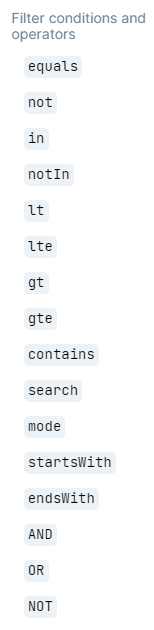




1. **Prisma: *filter conditions & operators*,** AND, OR, NOT

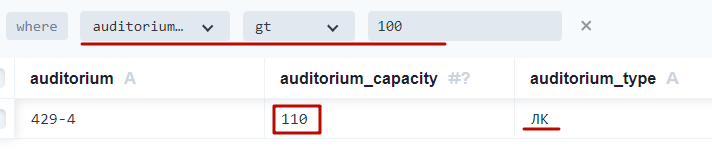


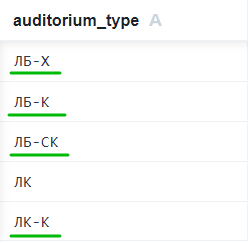


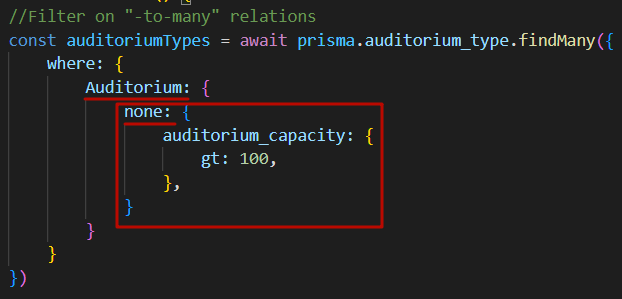
 

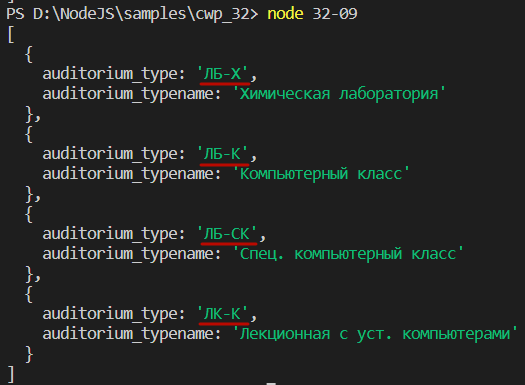
Подробнее: <https://www.prisma.io/docs/reference/api-reference/prisma-client-reference#filter-conditions-and-operators>

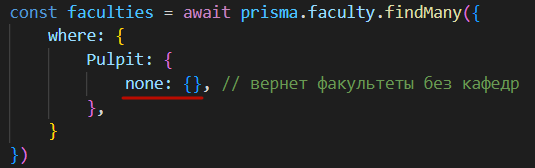
1. **Prisma: *relation fields*, *фильтр на связь "-ко-многим",*** none (возвращает все записи, в которых ни одна связанная запись не соответствует критериям фильтрации)





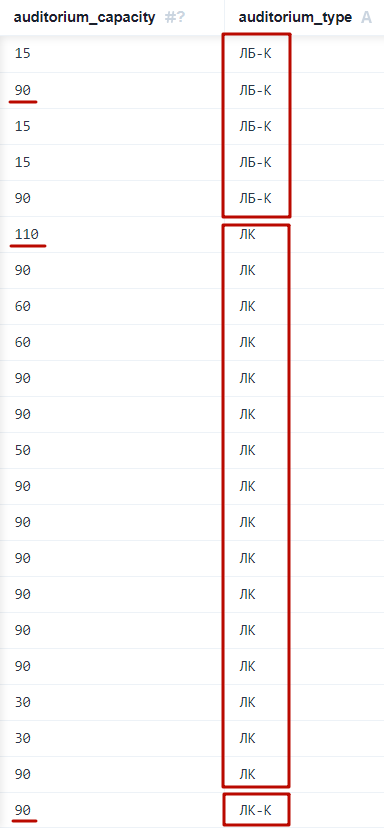
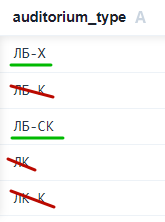


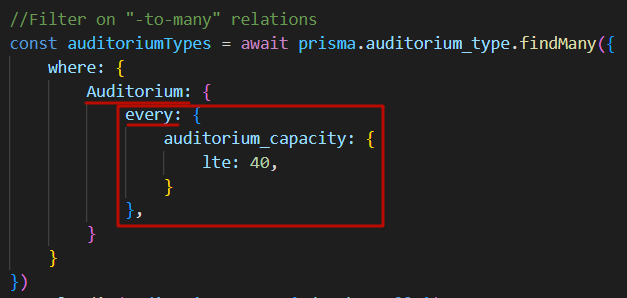


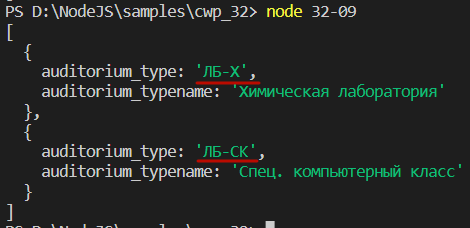




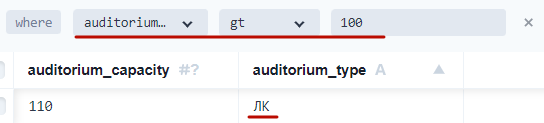
1. **Prisma: *relation fields*, *фильтр на связь "-ко-многим",*** every (возвращает все записи, в которых все связанные записи соответствует критериям фильтрации)

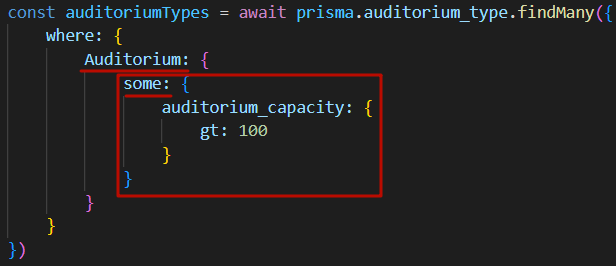
 





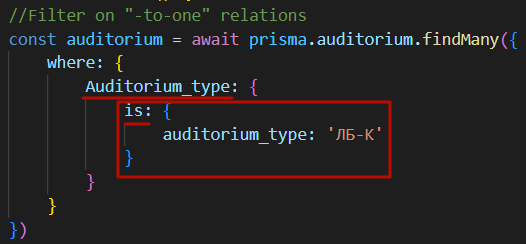
1. **Prisma: *relation fields*, *фильтр на связь "-ко-многим",*** some (возвращает все записи, в которых хотя бы одна связанная запись соответствует критериям фильтрации)

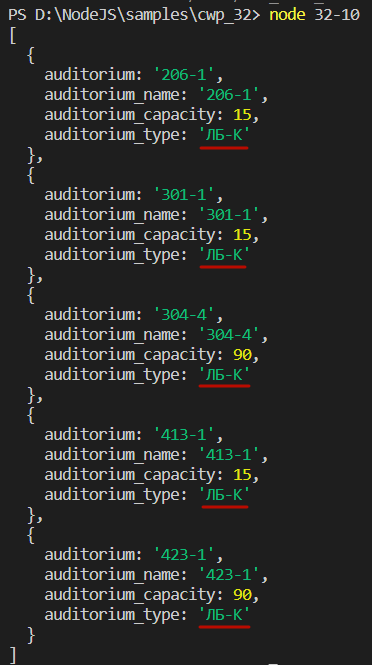
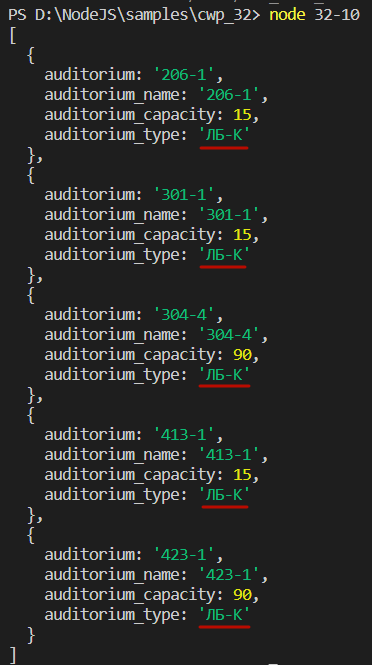




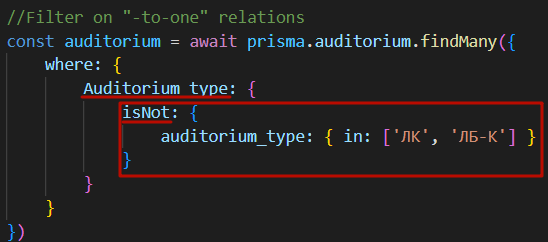
http://dl4.joxi.net/drive/2023/02/22/0030/0226/2023650/50/7d250c1a95.png

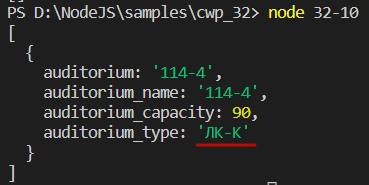
1. **Prisma: *relation fields*, *фильтр на связь "-к-одному",*** is (возвращает все записи, в которых связанная запись соответствует критериям фильтрации)

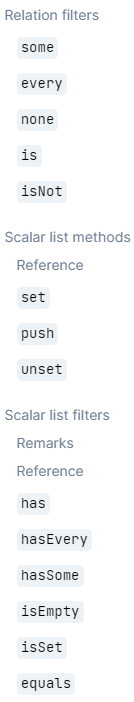




1. **Prisma: *relation fields*, *фильтр на связь "-к-одному",*** isNot (возвращает все записи, в которых связанная запись не соответствует критериям фильтрации)

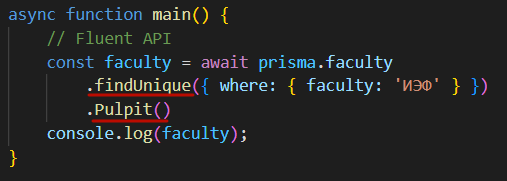


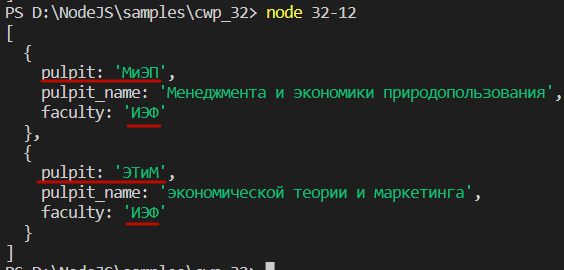




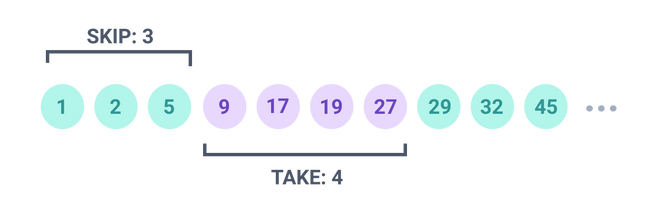
Подробнее: <https://www.prisma.io/docs/reference/api-reference/prisma-client-reference#relation-filters>

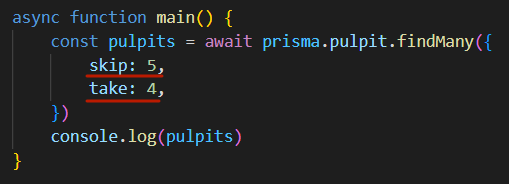
1. **Prisma: *Fluent API*** – потоковый интерфейс (цепочка методов). Последний вызов функции определяет возвращаемый тип всего запроса (в примере pulpit). Единственное требование для цепочки состоит в том, что предыдущий вызов функции должен возвращать только один объект.

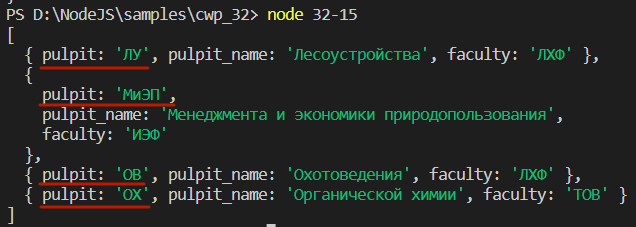




1. **Prisma:** skip(n) (пропустить n элементом)**,** take(m) («взять» m элементов), используются для пагинации

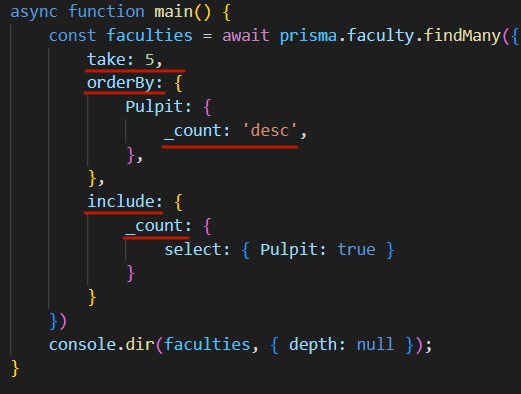


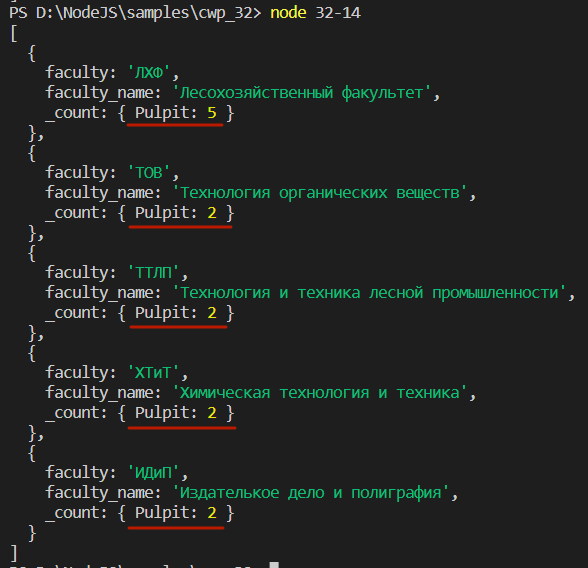




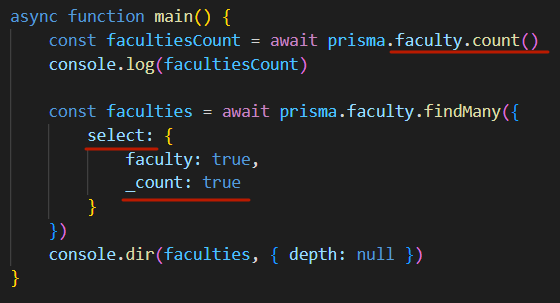
1. **Prisma:** \_count (подсчет количества записей). Может использоваться внутри include или select верхнего уровня. Кроме того, можно сортировать по количеству связанных записей.

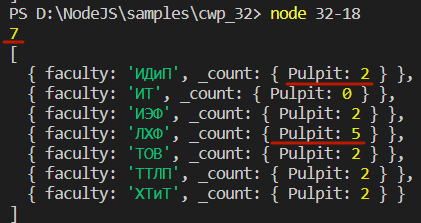
include => ***\_count*** => ***select*** => relation (подсчет кол-ва только указанных связанных записей)



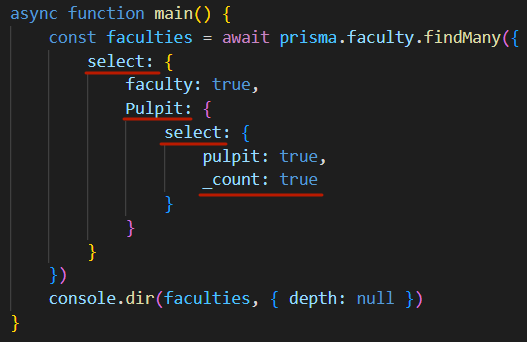


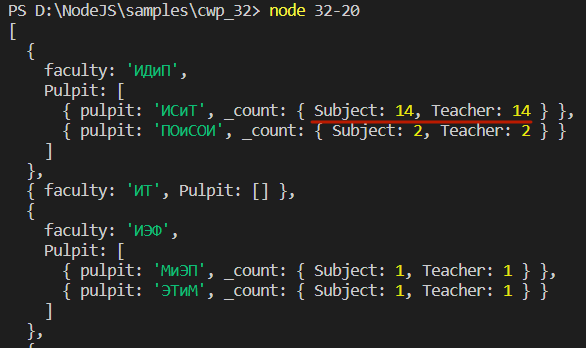
select => ***\_count: true*** (подсчет кол-ва всех связанных записей)





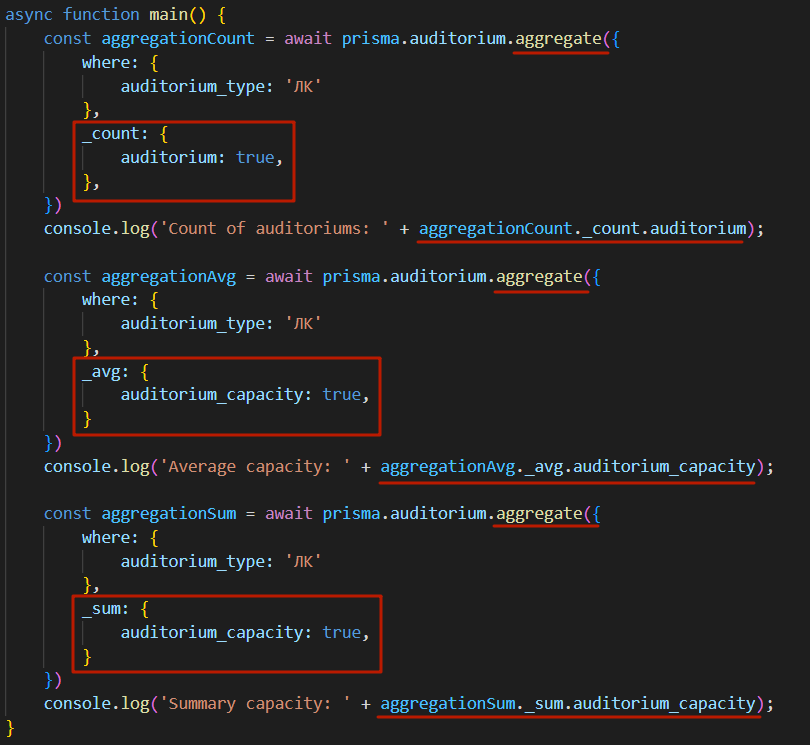
select => ***relation*** => select => ***\_count: true*** (подсчет кол-ва всех связанных с relation записей)

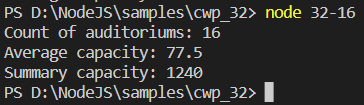




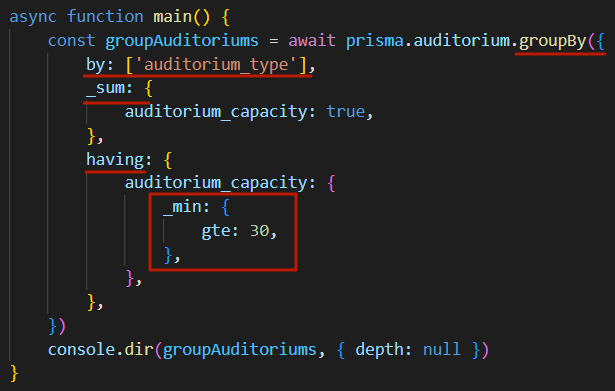
1. **Prisma:** aggregate (агрегирование):

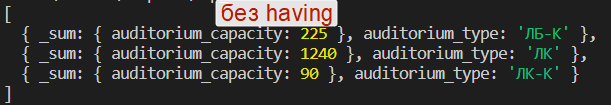
* \_count: возвращает количество записей или непустых полей;
* \_avg: возвращает среднее значение всех значений указанного поля;
* \_sum: возвращает сумму всех значений указанного поля;
* \_min: возвращает наименьшее значение указанного поля;
* \_max: возвращает наибольшее значение указанного поля.

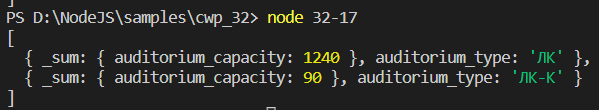




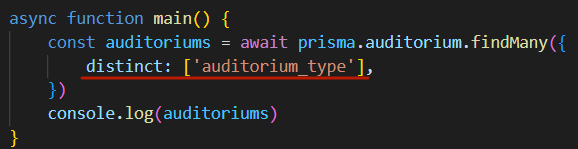
1. **Prisma:** groupBy (группировка записей по одному или нескольким полям)

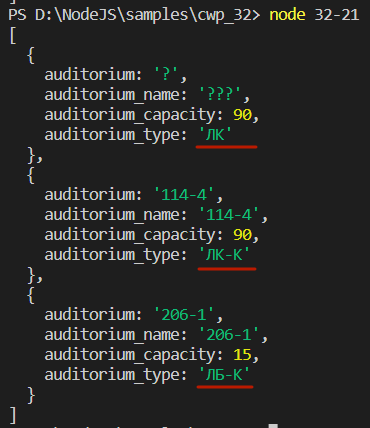






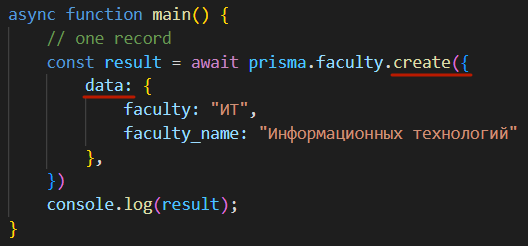
1. **Prisma:** distinct (возвращает только уникальные значения по одному или нескольким полям)





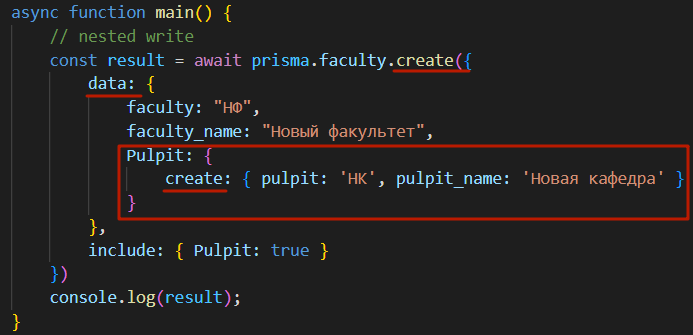
1. **Prisma:** create (создание одной записи)

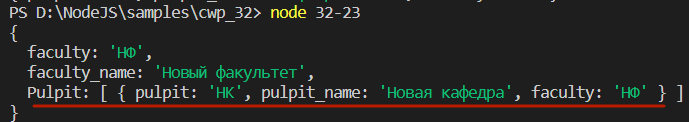




http://dl3.joxi.net/drive/2023/02/22/0030/0226/2023650/50/14ff6d6220.png

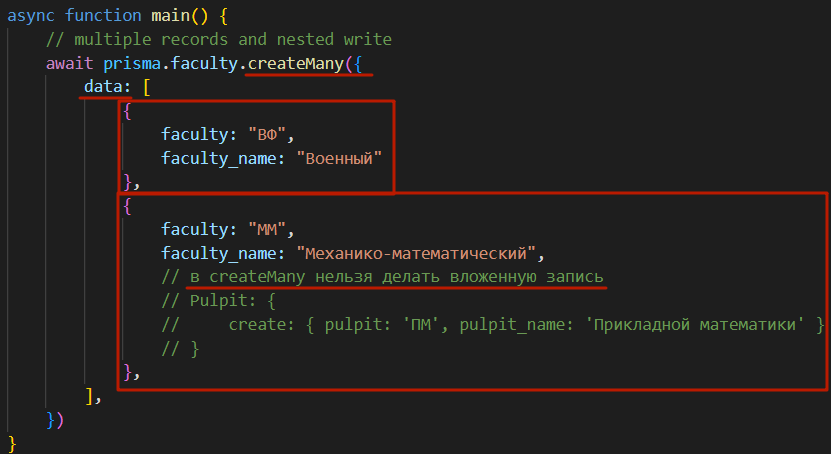
1. **Prisma:** create => create/createMany (создание одной или нескольких связанных записей), вложенная запись





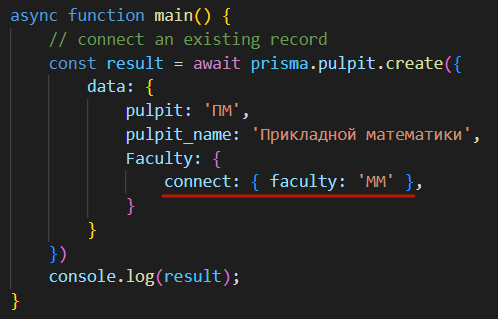
http://dl4.joxi.net/drive/2023/02/22/0030/0226/2023650/50/f8f9bf5add.png

1. **Prisma:** createMany (создание нескольких записей), вложенная запись



http://dl3.joxi.net/drive/2023/02/22/0030/0226/2023650/50/d34c56f3c7.png

1. **Prisma:** create => connect(присоединение существующей записи, в createMany нельзя так сделать)



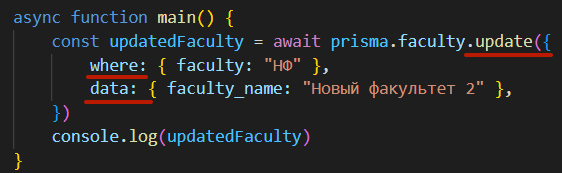


1. **Prisma:** create => connectOrCreate (присоединение существующей записи или создание новой, если ранее указанной не существует)



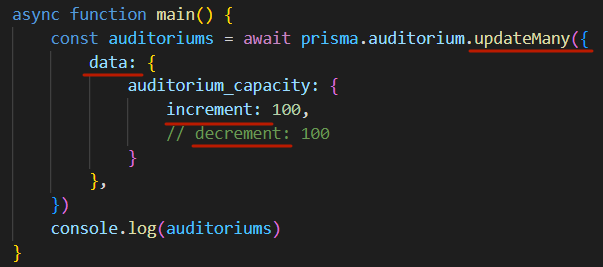


1. **Prisma:** update (изменяет одну указанную запись)



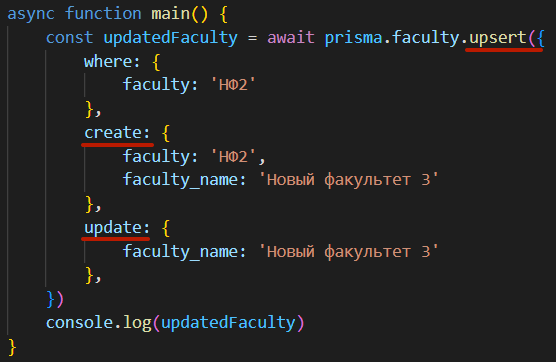
http://dl4.joxi.net/drive/2023/03/07/0030/0226/2023650/50/16dc1d7847.png

1. **Prisma:** updateMany (изменяет все записи, подходящие под критерий выборки)



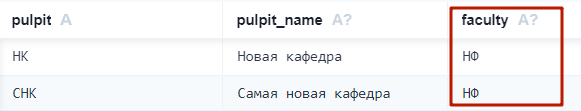
http://dl4.joxi.net/drive/2023/03/07/0030/0226/2023650/50/a34c3ef89f.png

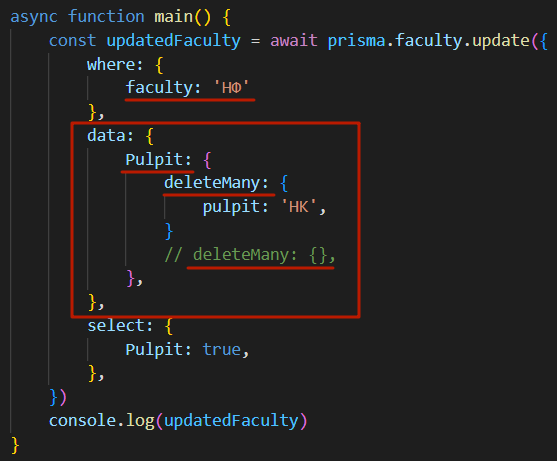
1. **Prisma:** upsert (изменяет указанную запись, если она уже существует, или создает новую, если указанной нет)

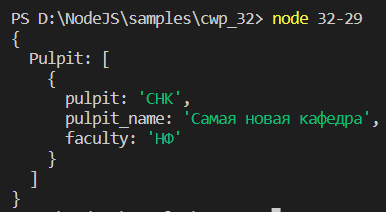


http://dl4.joxi.net/drive/2023/03/07/0030/0226/2023650/50/1e4d034757.png

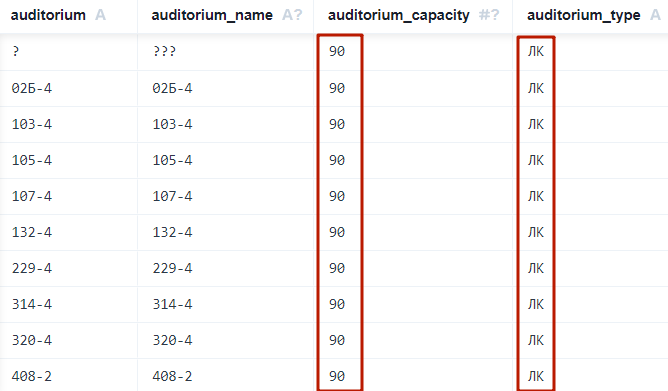
1. **Prisma:** update => deleteMany/delete (удаление одной или нескольких связанных записей)

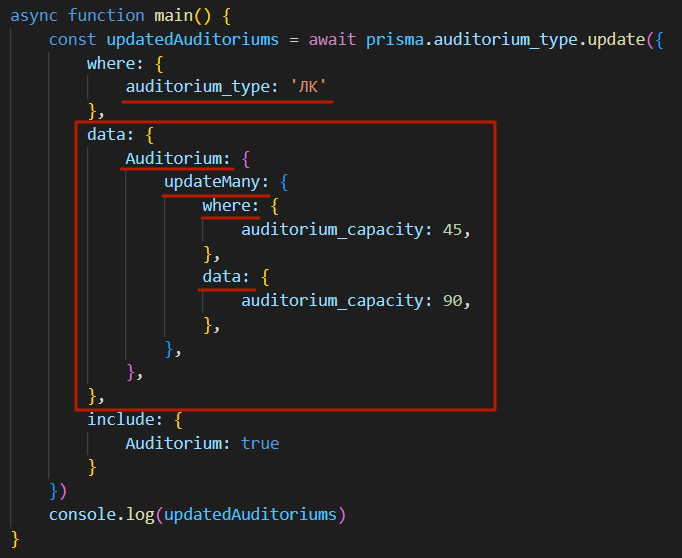


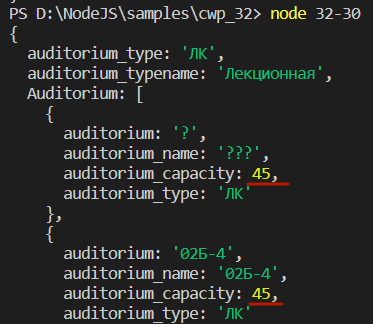


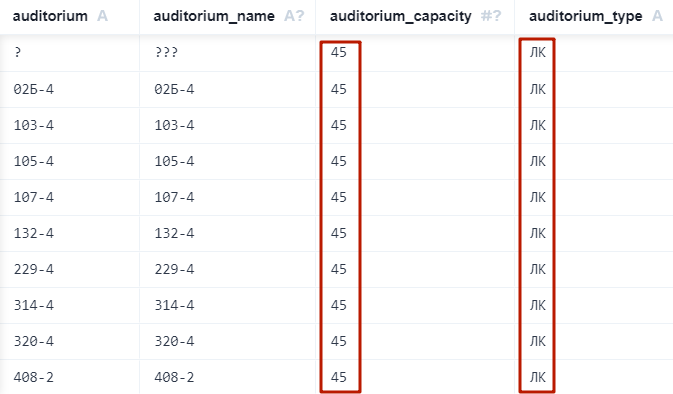


1. **Prisma:** update => updateMany (обновление нескольких связанных записей)



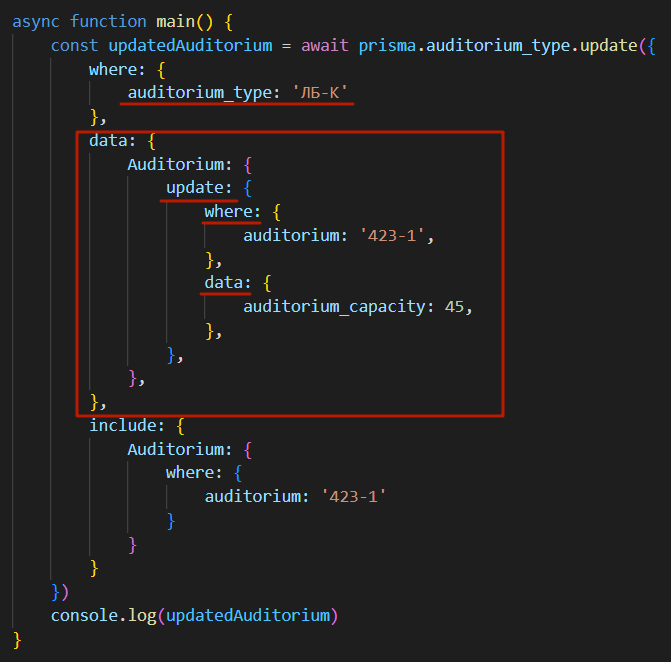


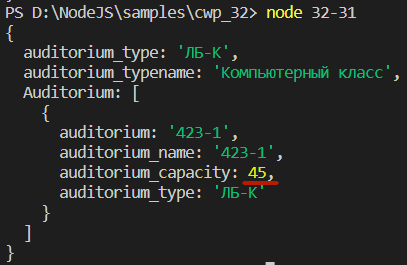




1. **Prisma:** update => update (обновление одной связанной записи)

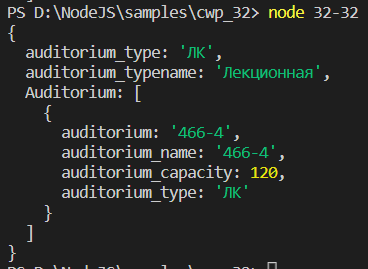
http://dl4.joxi.net/drive/2023/03/07/0030/0226/2023650/50/b0f8300148.png





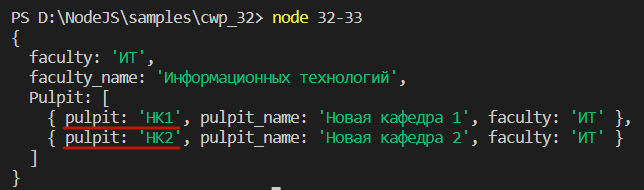
1. **Prisma:** update => upsert (изменяет указанную связанную запись, если она существует, иначе создает новую)



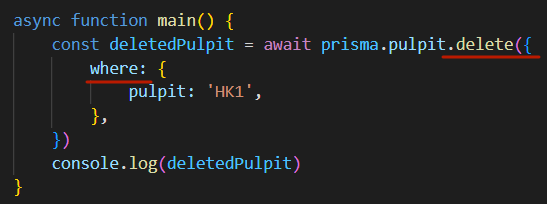


1. **Prisma:** update=> create/createMany (создает одну или несколько связанных записей)



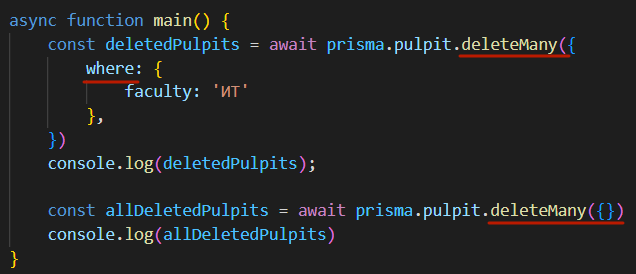


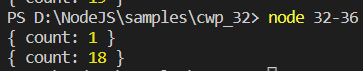
1. **Prisma:** delete (удаляет одну указанную запись)



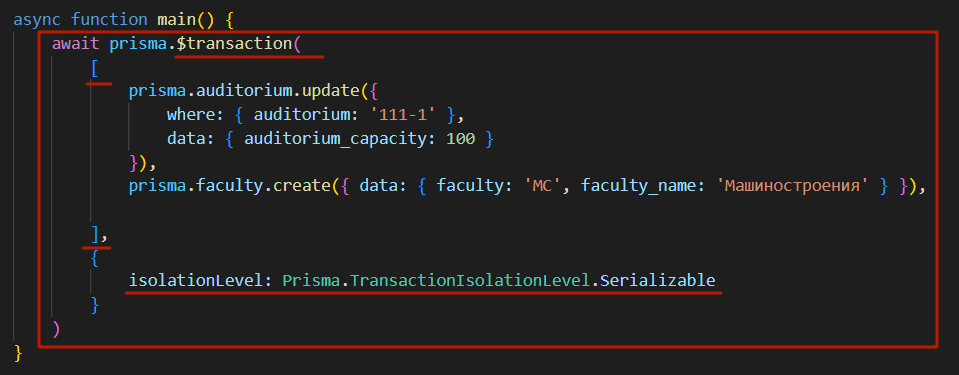
http://dl3.joxi.net/drive/2023/03/07/0030/0226/2023650/50/4bfbcf520a.png

1. **Prisma:** deleteMany (удаляет все записи, подходящие под критерий выборки)



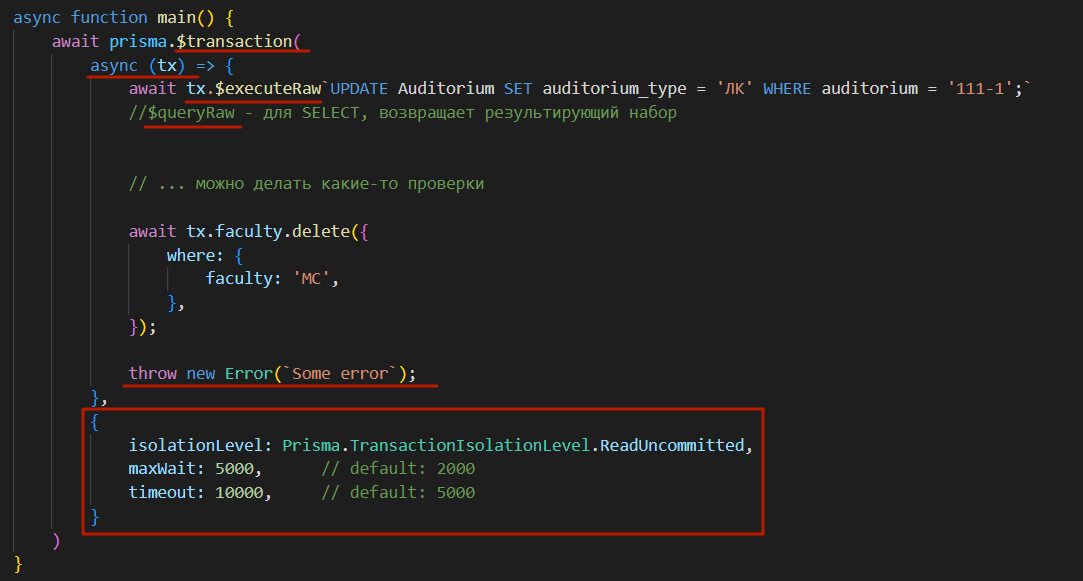


1. **Prisma:** транзакции: вложенные записи, пакет запросов (методы с суффиксом Many), $transaction.
2. **Prisma:** $transaction, последовательные операции





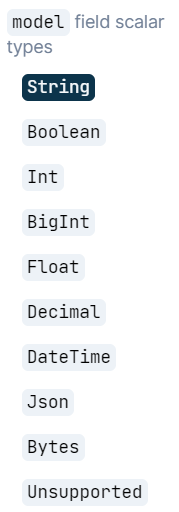
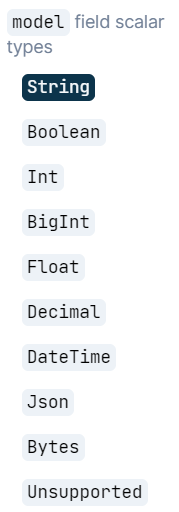
1. **Prisma:** $transaction, интерактивные транзакции, $queryRaw (возвращает выборки, например, SELECT), $executeRaw (возвращает количество затронутых строк, например, UPDATE, DELETE)



1. **Prisma:** схема(data source, generator, data model), модель, Prisma Schema Language (PSL)



1. **Prisma:** *поля* – свойства модели, состоят из имени, типа, модификаторов и атрибутов.
2. **Prisma:** *типы*: скалярные, поля отношений (relation field), атрибут собственных типов БД (@db).



1. **Prisma:** также можно определять enum’ы и составные типы, если они поддерживаются СУБД.
2. **Prisma:** *модификаторы типа*: ?(необязательное), [] (список)
3. **Prisma:** *атрибуты*: изменяют поведение полей (префикс @) или модели (префикс @@)

@id – поле является PK таблицы.

@@id – составной PK таблицы.

@unique – уникальное поле в пределах таблицы.

@@unique – составное ограничение уникальности для указанных полей.

@relation – отношение между таблицами (fields – поля текущей модели, references – поля другой модели, map – имя FK в БД, …)

@map – привязка поля модели к указанному столбцу таблицы.

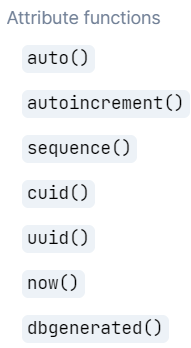
@@map – привязка модели к указанной таблице.

@updateAt – обновляет поле текущей датой и временем при модификации записи.

@ignore – невалидное поле.

@@ignore – невалидная модель.

1. **Prisma:** функции атрибутов



1. **Prisma:** модели можно определить двумя способами:

1) определение модели вручную + Prisma Migrate;

2) генерация модели с помощью интроспекции (в начале лк)

1. **Prisma:** code first, миграции

