**GenTech / Mar 28, 2023**

**Разбор Д/З**

**(3) Вывести названия стримов без пожертвований или без реакций**

db.streams.aggregate([

{

'$lookup': {

'from': 'donations',

'localField': 'id',

'foreignField': 'stream\_id',

'as': 'donation'

}

},

{

'$lookup': {

'from': 'reactions',

'localField': 'id',

'foreignField': 'stream\_id',

'as': 'reaction'

}

},

{

'$match': {

'$or': [

{'donation': { '$size': 0 }},

{'reaction': { '$size': 0 }}

]

}

},

{

'$project': {

'title': 1,

'\_id': 0

}

}

])

**(4) Вывести максимальный размер пожертвования для каждого стримера.**

db.donations.aggregate([

{

'$lookup': {

'from': 'streams',

'localField': 'stream\_id',

'foreignField': 'id',

'as': 'streams'

}

},

{ '$unwind': '$streams' },

{

'$group': {

'\_id': '$streams.user\_id',

'max\_value': { '$max': '$amount' }

}

},

{

'$lookup': {

'from': 'users',

'localField': '\_id',

'foreignField': 'id',

'as': 'users'

}

},

{ '$unwind': '$users' },

{

'$project': {

'\_id': 0,

'stream\_fullname': '$users.fullname',

'max\_value': 1

}

}

])

С оптимизацией

db.donations.aggregate([

{

'$group': {

'\_id': '$stream\_id',

'max': {'$max': '$amount'}

}

},

{

'$lookup': {

'from': 'streams',

'localField': '\_id',

'foreignField': 'id',

'as': 'streams'

}

},

{ '$unwind': '$streams' },

{

'$group': {

'\_id': '$streams.user\_id',

'max\_value': { '$max': '$max' }

}

},

{

'$lookup': {

'from': 'users',

'localField': '\_id',

'foreignField': 'id',

'as': 'users'

}

},

{ '$unwind': '$users' },

{

'$project': {

'\_id': 0,

'stream\_fullname': '$users.fullname',

'max\_value': 1

}

}

])

**(5) Вывести ТОП-3 пожертвований из Германии (имя донатора и размер пожертвования)**

db.donations.aggregate([

{

'$lookup': {

'from': 'users',

'localField': 'donator\_id',

'foreignField': 'id',

'as': 'donator'

}

},

{ '$unwind': '$donator' },

{

'$match': {

'donator.country': 'Germany'

}

},

{ '$sort': { 'amount': -1 } },

{ '$limit': 3 },

{

'$project': {

'donator\_fullname': '$donator.fullname',

'amount': 1,

'\_id': 0

}

}

]);

**Проектирование БД**

1. На одном сервере могут находиться несколько **СУБД** (**DBMS**)

* Сетевой порт - сетевой идентификатор процесса (программы)

1. В одной **СУБД** может находиться несколько **БД**
2. В одной **БД** может храниться несколько типов сущностей
3. Стандартный подход: **ОДНА БД == ОДИН ПРОЕКТ**
4. Стандартный подход: **ОДНА СУЩНОСТЬ == ОДНА ЗАПИСЬ**

**Общие этапы проектирования БД**

1. Анализ предметной области (бизнес-процессов)
2. Выделение сущностей и их характеристик
3. Связывание сущностей между собой (установление связей)
   1. с помощью ключей (ключевое поле)
   2. применяются для связывания первичный ↔ внешний ключи

**Первичный ключ**

Является идентификатором каждой сущности

**Свойства:**

* уникальный
* ненулевой
* неизменяемый

**Стандартные подходы к значению**

1. Авто-инкремент
2. UUID, GUID, ULID - случайные значения

**Примеры унификация хранимых значений**

Приведение к единому виду

* телефоны
* email

**Базовые виды связей между сущностями**

* 1:1 (один к одному)
* 1:M (один ко многим)
* M:M (многие ко многим)

**При проектировании БД могут использоваться доп. инструменты, например**

**ERD** - диаграмма “сущность-связь”

**Задача**

**(1) Описать связи между коллекциями в БД platform\_be**

(ответ дожен содержать название ключей)

**(2) Описать связи между коллекциями БД (”чат поддержки”)**

users.id (1:M) messages.author\_id

chats.id (1:M) messages.chat\_id

reactions.message\_id (M:1) messages.id

reactions.author\_id (M:1) users.id

chats.part1 (M:1) users.id

chats.part2 (M:1) users.id

**(3) Описать связи между коллекциями БД “банк”**

txs.sender\_id (M:1) users.id

txs.recipient\_id (M:1) users.id

txs.account1 (M:1) accounts.id

txs.account2 (M:1) accounts.id

users.id (1:M) accounts.user\_id

calls.user\_id (M:1) users.id

calls.worker\_id (M:1) workers.id

users.manager\_id (M:1) workers.id

**Совместная работа в Git**

pull = fetch + merge

git push --force принудительная выгрузка

**Подходы**

1. Наличие доступа **на запись** к репо
2. Через запросы слияния (**Pull-requests**)
   1. “предложение **улучшение** через заявку”

**Этапы работы через запросы слияния**

1. **Сделать форк репо**
   1. форк (fork) - копия
2. **Выгрузить репо**
   1. git clone <ссылка>
3. **Внесение правок**
   1. отдельная ветка
      1. git checkout -b новое\_название создать и переклються на новую ветку
4. **Выгрузка на GitHub**
   1. git push -u origin ветка
5. **Оформить запрос слияния**
   1. через веб-интерфейс

**Справочная документация**

[Нормализация БД](https://learn.microsoft.com/ru-ru/office/troubleshoot/access/database-normalization-description)