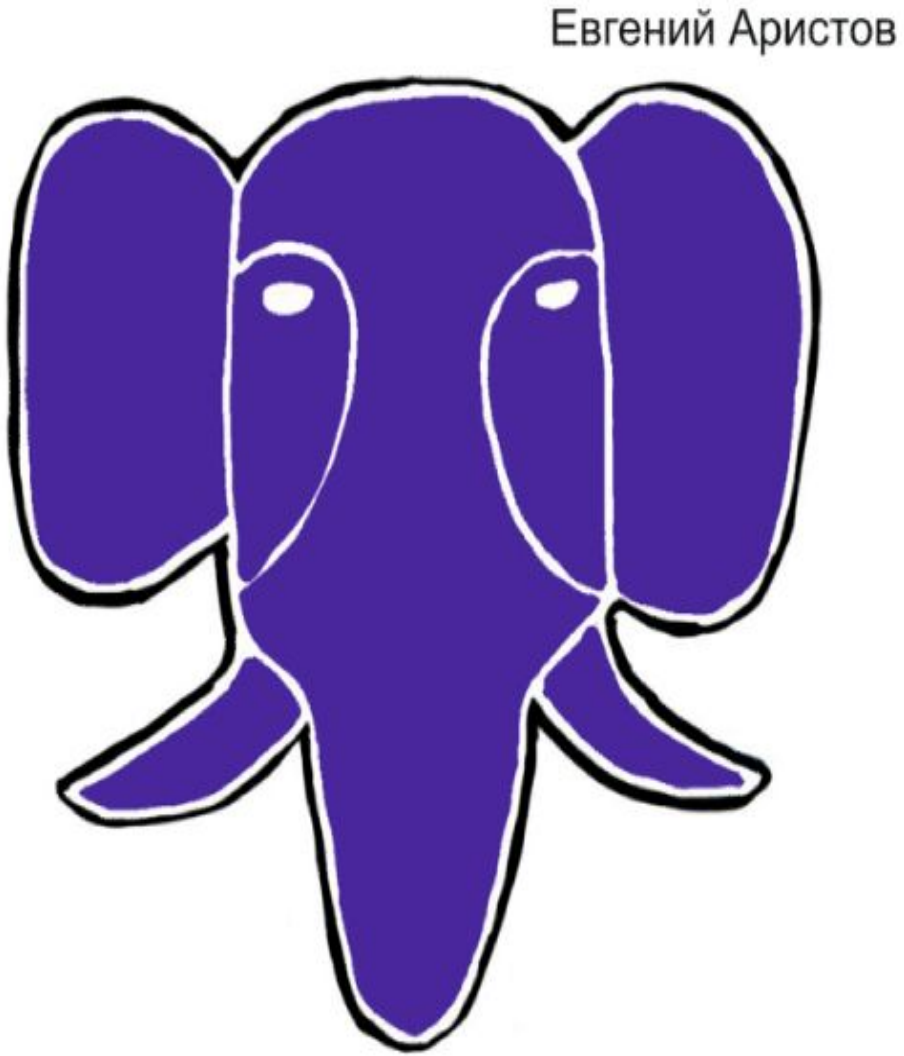
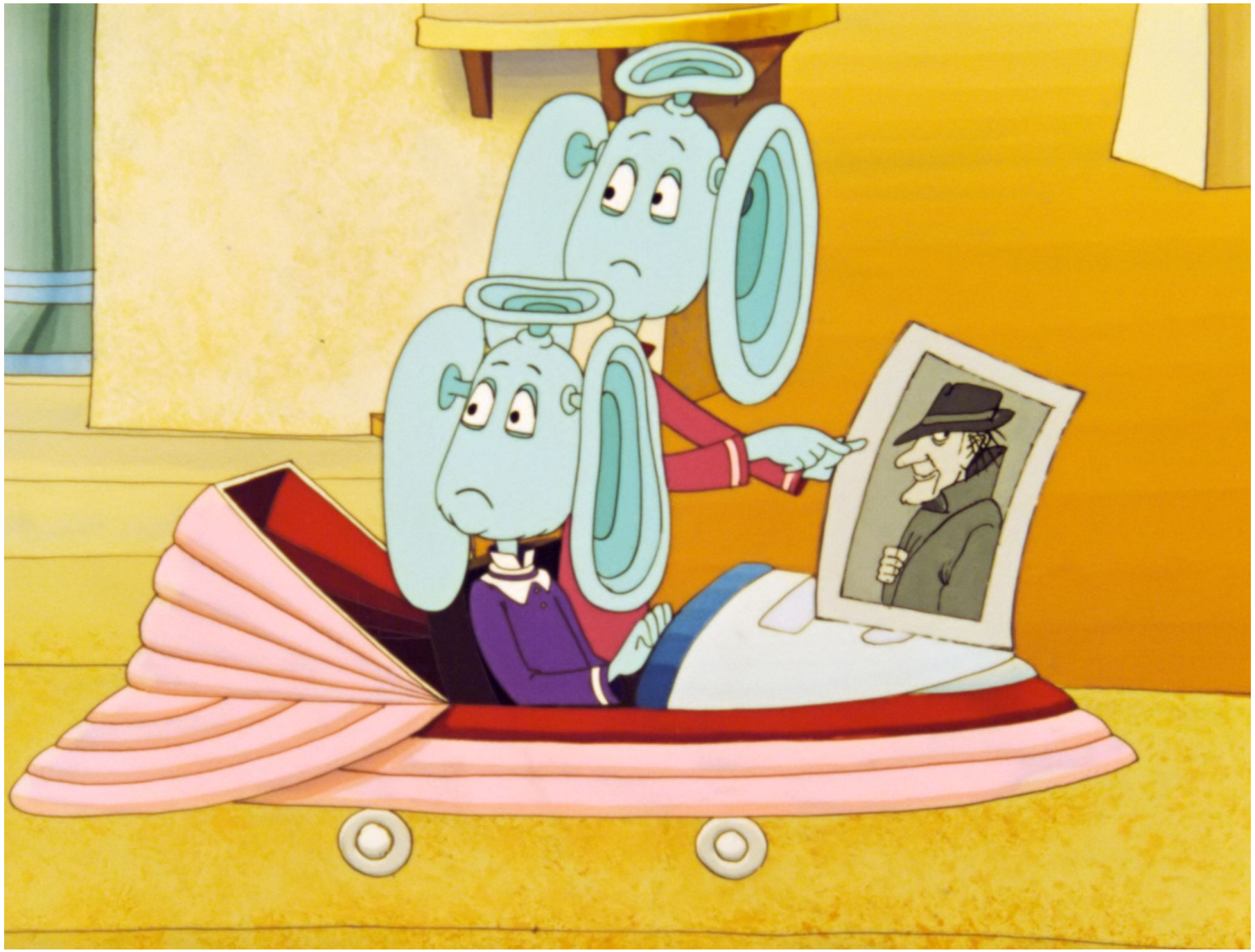
INSERT





Правила вебинара

Задаем вопрос в чат

Вопросы вижу, отвечу в момент логической паузы

Если есть вопрос голосом - поставьте знак ? в чат

Если остались вопросы, можно их задать на следующем занятии

Маршрут вебинара

1. Варианты INSERT
2. ON CONFLICT
3. RETURNING
4. Batch load

https://aristov.tech

INSERT

https://aristov.tech

INSERT

Смысла в схеме данных без данных нет. Для того, чтобы их добавить стандартная команда - INSERT

INSERT INTO *table_name* [AS *alias*] [(*column_name* [, ...])]
{ DEFAULT VALUES | VALUES ({ *expression* | DEFAULT } [, ...]) [, ...]

Можем выборку вставить в УЖЕ существующую таблицу:

INSERT INTO *table_name* [AS *alias*] [(*column_name* [, ...])]
SELECT …

ON CONFLICT

https://aristov.tech

INSERT ON CONFLICT

Иногда мы не можем добавить значение из-за ограничения целостности. Классика, такое уникальное значение уже существует.

[ON CONFLICT [*conflict_target*] *conflict_action*]

where *conflict_target* can be one of:

({ *index_column_name* | (*index_expression*) } [COLLATE *collation*] [*opclass*] [, ...]) [WHERE *index_predicate*]

ON CONSTRAINT *constraint_name*

and *conflict_action* is one of:

DO NOTHING

DO UPDATE SET { *column_name* = { *expression* | DEFAULT } |

(*column_name* [, ...]) = [ROW] ({ *expression* | DEFAULT } [, ...]) |

(*column_name* [, ...]) = (*sub-SELECT*)

} [, ...]

[WHERE *condition*]

Реализация UPSERT.

Returning

https://aristov.tech

Returning

После добавления значения, нам обычно хочется узнать его Id. Для этого есть функционал:

[RETURNING * | *output_expression* [[AS] *output_name*] [, ...]]

Batch load

https://aristov.tech

Batch insert. Best practice

- Оптимальные размеры батчей 100к-1кк - учитываем размер транзакций, аффект на WAL файлы и занимаемую память + сброс грязных буферов на диск
- самый быстрый метод, как ни странно, COPY
- отключаем индексы, если позволяет логика
- отключаем триггеры, внешние ключи
- не забываем ANALYZE после лоада
- ну и вернуть индексы и иже с ними

https://www.enterprisedb.com/blog/7-best-practice-tips-postgresql-bulk-data-loading

https://aristov.tech

Batch insert. Best practice

COPY in 16 Postgres can be 300% faster
https://pganalyze.com/blog/5mins-postgres-16-faster-copy-bulk-load

Parallel COPY
https://dev.to/josethz00/speed-up-your-postgresql-bulk-inserts-with-copy-40pk

Asynchronous I/O is not going to make it in Postgres 16, there were a couple of patches that relate to it like this one, but **Postgres 17 is going to be the first release where there is actually going to be interesting**

https://aristov.tech

Итоги

https://aristov.tech

Итоги

Остались ли вопросы?

Увидимся на следующем занятии

https://aristov.tech

Спасибо за внимание!

Когда дальше и куда?
В чате напишу
материалы для бесплатного доступа будут появляться на ютубе

Аристов Евгений