



EVEN CHILDREN CAN WORK WITH POSTGRES

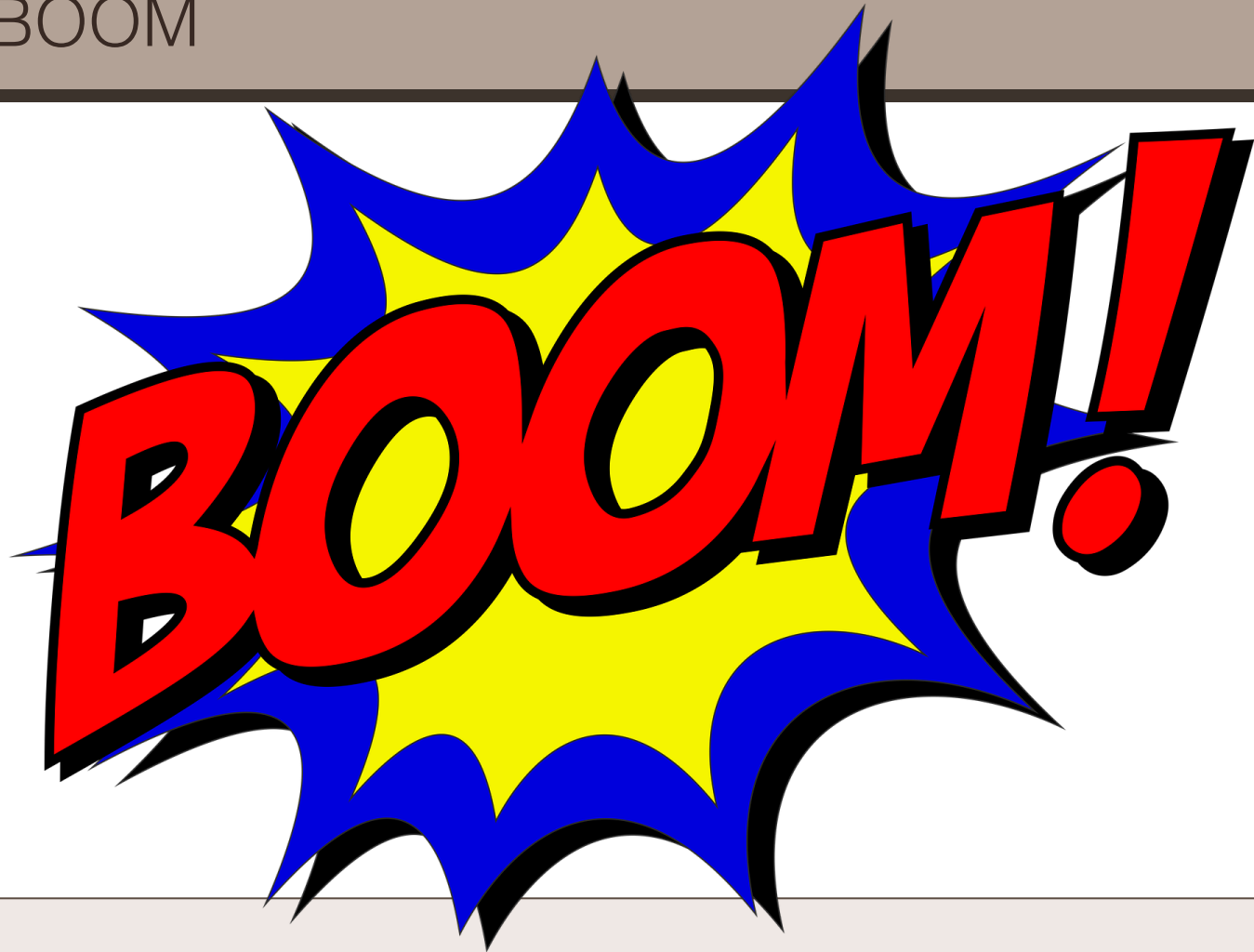
TATIANA KRUPENYA

March 15, 2024

NICE TO MEET YOU

- I am the CEO of DBeaver Corporation
- DBeaver is a UI tool for database management and administration
- DBeaver was born as an open-source project in 2011
- The number of DBeaver users has grown to more than 8 million
- DBeaver is used in every country in the world





ChatGPT FOR QUERIES

chat.openai.com

“write a SQL statement to show all invoices from Postgres”



You

write a SQL statement to show all invoices from Postgres



ChatGPT

To retrieve all invoices from a PostgreSQL database, you need to have a table named `invoices` (or whatever name your invoice table has) containing the necessary data. Assuming you have a table named `invoices`, and it has columns like `invoice_id`, `customer_id`, `invoice_date`, and any other relevant columns, you can use the following SQL statement:

```
sql Copy code  
  
SELECT *  
FROM invoices;
```

This SQL statement selects all columns (`*`) from the `invoices` table. If you want to select specific columns, you can replace `*` with the column names you want to retrieve, separated by commas. For example:

```
sql Copy code  
  
SELECT invoice_id, customer_id, invoice_date  
FROM invoices;
```


Replace `invoices` with the actual name of your invoice table if it's different.




ChatGPT FOR QUERIES

chat.openai.com

“show all invoices”

 **You**
write a SQL statement to show all invoices from Postgres


 **ChatGPT**
To retrieve all invoices from a PostgreSQL database, you need to have a table named `'invoices'` (or whatever name your invoice table has) containing the necessary data. Assuming you have a table named `'invoices'`, and it has columns like `'invoice_id'`, `'customer_id'`, `'invoice_date'`, and any other relevant columns, you can use the following SQL statement:


```
sql Copy code  
  
SELECT *  
FROM invoices;
```

This SQL statement selects all columns (`*`) from the `'invoices'` table. If you want to select specific columns, you can replace `*` with the column names you want to retrieve, separated by commas. For example:

```
sql Copy code  
  
SELECT invoice_id, customer_id, invoice_date  
FROM invoices;
```

Replace `'invoices'` with the actual name of your invoice table if it's different.

 **You**
show all invoices

 **ChatGPT**
If you want to display all the invoices with their details, you can use the following SQL query:

```
sql Copy code  
  
SELECT *  
FROM invoices;
```

This query will retrieve all columns and all rows from the `'invoices'` table, displaying all invoice records with their respective details. If you only want specific columns, you can list them instead of using `*`. For example:

```
sql Copy code  
  
SELECT invoice_id, customer_id, invoice_date, total_amount  
FROM invoices;
```

Replace `'invoices'` with the actual name of your invoice table if it's different.

ChatGPT FOR QUERIES

chat.openai.com

“show all invoices”



You

show all invoices



ChatGPT

I'm sorry for any confusion, but I can't access or display specific invoices as I don't have access to databases or real-time information. However, I can certainly help guide you on how to generate or manage invoices depending on your needs.

If you're looking to generate invoices, there are several software tools available, such as QuickBooks, FreshBooks, Zoho Invoice, or even templates in Microsoft Excel or Google Sheets. These tools allow you to input the necessary information such as itemized charges, customer details, and payment terms, and generate professional-looking invoices.

If you're looking to manage existing invoices within a system or software, you would typically access that through your organization's accounting or invoicing software. You would log in to the system and navigate to the section that manages invoices. From there, you can typically view, search, and manage existing invoices.

If you need assistance with a specific aspect of invoicing or have any other questions, feel free to ask!



CONTEXT IS THE KEY

chat.openai.com

“Show all invoices”

SQL

Copy code

```
SELECT * FROM invoices;
```

1

“write a **SQL statement** to show all invoices from **Postgres**”

2

ChatGPT



Examples

"Explain quantum computing in simple terms" →

"Got any creative ideas for a 10 year old's birthday?" →

"How do I make an HTTP request in Javascript?" →



Capabilities

Remembers what user said earlier in the conversation

Allows user to provide follow-up corrections

Trained to decline inappropriate requests



Limitations

May occasionally generate incorrect information

May occasionally produce harmful instructions or biased content

Limited knowledge of world and events after 2021

WHAT IS YOUR QUERY CONTEXT?

The screenshot shows the DBeaver Database Navigator interface. The left sidebar displays a tree view of the database structure for 'pagila', including schemas, tables, and views. The 'film' table is selected. The main pane shows the 'Properties' tab for the 'film' table, displaying metadata such as Table Name, Tablespace, Object ID, Owner, and Extra Options. Below this, the 'Columns' tab is active, showing a table with 13 columns and their respective data types, identities, collations, nullability, and default values.

Column Name	#	Data type	Identity	Collation	Not Null	Default
123 film_id	1	serial4			[v]	nextval('film_film_id_seq':regc
ABC title	2	varchar(255)	default		[v]	
ABC descript	3	text	default		[]	
123 release_year	4	year			[]	
123 language_id	5	int2			[v]	
123 original_language_id	6	int2			[]	
123 rental_duration	7	int2			[v]	3
123 rental_rate	8	numeric(4, 2)			[v]	4.99
123 length	9	int2			[]	
123 replacement_cost	10	numeric(5, 2)			[v]	19.99
ABC rating	11	mpaa_rating			[]	'G':mpaa_rating
last_update	12	timestamp			[v]	now()
special_features	13	_text	default		[]	

- Database type
- Database name
- Schema name
- Table name
- Column name
- Primary keys
- Foreign keys

IT'S TOO MUCH

The screenshot shows the DBeaver interface with the following components:

- Database Navigator:** Shows a tree view of the 'pagila' database, including schemas like 'information_schema', 'pg_catalog', and 'public'. Under 'public', various tables and views are listed.
- Properties Panel:** Shows details for the 'film' table, including 'Table Name: film', 'Tablespace: pg_default', and 'Object ID: 5409'.
- Columns Panel:** A table listing the columns of the 'film' table with their data types and constraints.

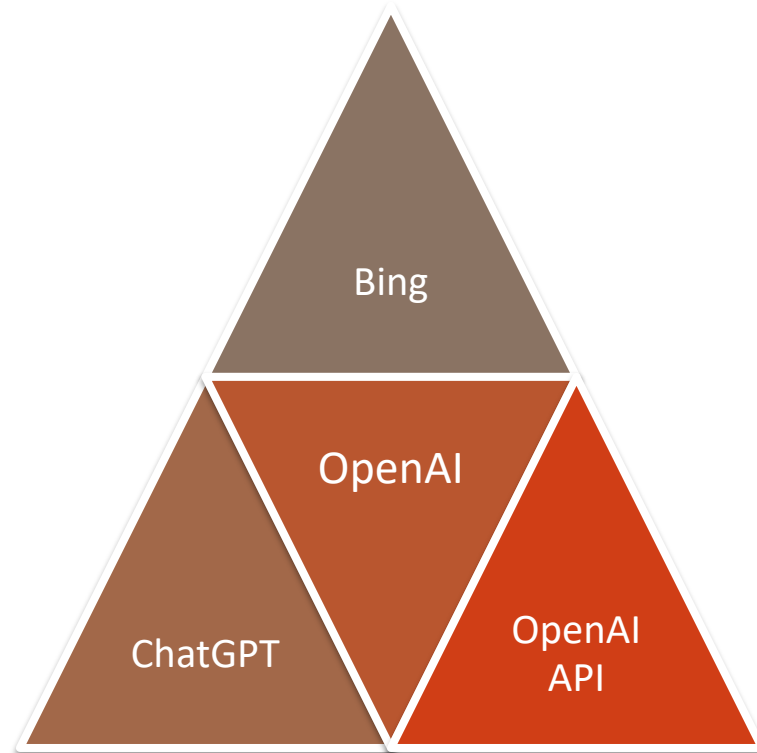
Column Name	Data Type	Ident	Collation	Not Null	Default
film_id	serial			[v]	nextval('film_film_id_seq'::regc
title	varchar(255)			[v]	
description	text		default	[]	
release_year	year			[]	
language_id	int2			[v]	
original_language_id	int2			[]	
rental_rate	int2			[v]	3
replacement_cost	numeric(4, 2)			[v]	4.99
length	int2			[]	
replacement_cost	numeric(5, 2)			[v]	19.99
rating	mpaa_rating			[]	'G'::mpaa_rating
last_update	timestamp			[v]	now()
special_features	text		default	[]	

- Database type
- Database name
- Schema name
- Column name
- Primary keys
- Foreign keys

OPEN AI INTERFACES

platform.openai.com

OpenAI API is a good option if you want to automate your work.



WHAT APIS DOES DBEAVER SUPPORT?



OpenAI



Azure
OpenAI



Google
Gemini

HOW IT WORKS

Determine context



Write request



Form prompt



Get response



Show results

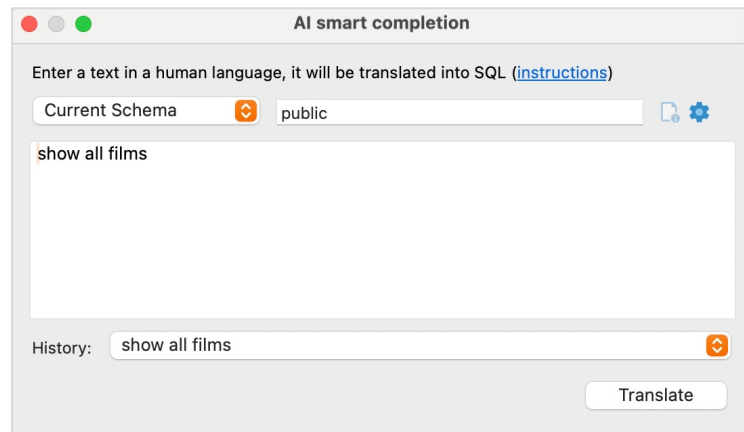
DETERMINE CONTEXT



- More information is better than less
- The number of tokens is the main limitation of your context:
 - 1000 words are around 750 tokens
 - GPT 3.5 – 16,000 tokens
 - GPT 4.0 – 32,000 tokens
 - Azure gpt-35-turbo – 4,000 tokens
 - Azure gpt-35-turbo-16k – 16,000 tokens
 - Gemini 1.0 – 32,000 tokens
 - Gemini 1.5 – 1,000,000 tokens

WRITE REQUEST – IT'S SIMPLE

- ✓ Any human language
- ✓ You can formulate your request in any words
- ✓ Not necessary to be grammatically correct
- ✓ You can request data from different tables
- ✓ You can provide hints for OpenAI, like “use joins”



FORM PROMPT

SIMPLE PROMPT

Context

Database
metadata

+

Query

Human-readable
request

+

Task

Create SQL
statement

+

Beginning

Start from the
word SELECT

FORM PROMPT

PROMPT FOR AI CHAT

Context

Database
metadata

+

Query

Human-readable
request

+

Task

Create SQL
statement

+

Beginning

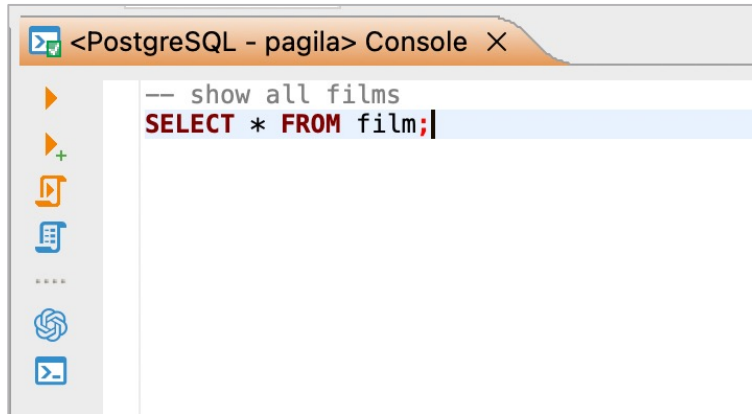
Add comment

Previous response

Previous request

GET RESPONSE

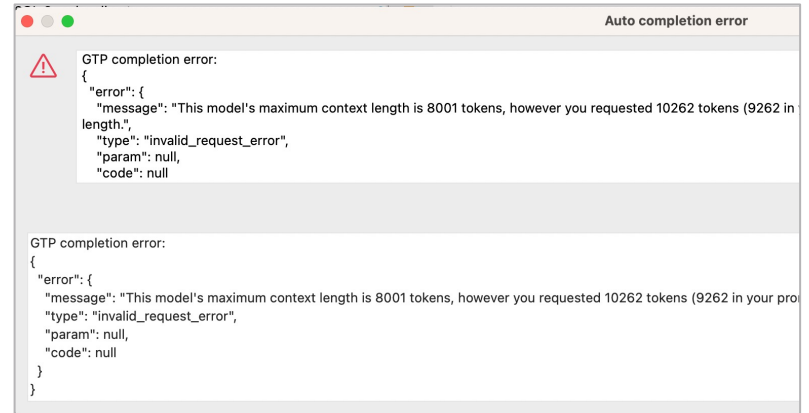
POSITIVE



<PostgreSQL - pagila> Console X

```
-- show all films  
SELECT * FROM film;
```

NEGATIVE



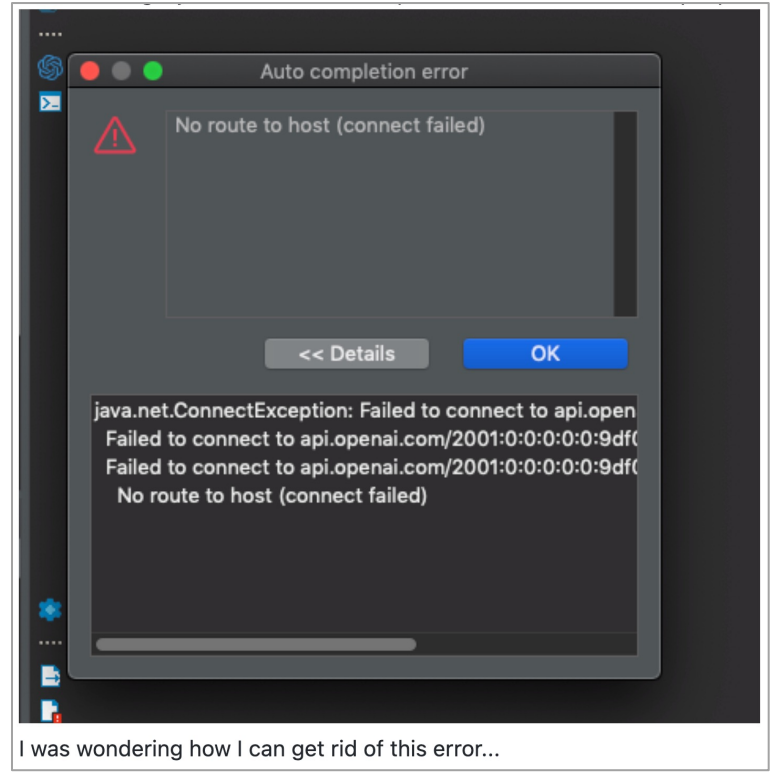
Auto completion error

```
GTP completion error:  
{  
  "error": {  
    "message": "This model's maximum context length is 8001 tokens, however you requested 10262 tokens (9262 in length.",  
    "type": "invalid_request_error",  
    "param": null,  
    "code": null  
  }  
}
```

GTP completion error:
{
 "error": {
 "message": "This model's maximum context length is 8001 tokens, however you requested 10262 tokens (9262 in your pro",
 "type": "invalid_request_error",
 "param": null,
 "code": null
 }
}

WHY DID SOMETHING GO WRONG?

- ✗ You do not have internet access
- ✗ AI API is under a high load
- ✗ Your API token is invalid
- ✗ You forgot to pay for access
- ✗ You chose the wrong model
- ✗ You sent too many tokens



NO GUARANTEES

The screenshot shows the DBeaver Ultimate 24.0.0 interface. On the left is the Database Navigator showing the PostgreSQL - pagila database structure. The main window displays a SQL query in the SQL Editor:

```
/*
 * We need to retrieve all films that were rented on May 25, 2005.
 * To do this, we will join the rental and inventory tables to get the film_id,
 * and then join the film table to get the film details.
 * Finally, we will filter the results to only include rentals on May 25, 2005.
 */
select
  f.film_id,
  f.title,
  f.description,
  f.release_year
from
  rental r
join inventory i on
  r.inventory_id = i.inventory_id
join film f on
  i.film_id = f.film_id
where
  r.rental_date::date = '2005-05-25';
```

Below the query, the Results pane shows an error message:

```
SQL Error [42703]: ERROR: column
f.description does not exist
Hint: Perhaps you meant to reference
the column "i.description".
Position: 40
```

The error message is repeated at the bottom of the interface. The status bar at the bottom shows: PST en_US Writable Smart Insert 12 : 5 : 358 Sel: 0 | 0

Asking the Right Question Is Half the Answer

YOUR SUCCESS

The screenshot displays the DBeaver Ultimate 24.0.0 interface. The top window is titled "DBEaver Ultimate 24.0.0 - <PostgreSQL - pagila> Console". The left sidebar shows the "Database Navigator" with a tree view of the "pagila" database, including schemas, tables, and views. The main console area shows a SQL query that retrieves all films rented on May 25, 2005, without showing the description. The query is as follows:

```
select
  f.film_id,
  f.title,
  f.release_year,
  f.language_id,
  f.original_language_id,
  f.rental_duration,
  f.rental_rate,
  f.length,
  f.replacement_cost,
  f.rating,
  f.last_update,
  f.special_features,
  f.fulltext
from
  film f
join inventory i on
  f.film_id = i.film_id
join rental r on
  i.inventory_id = r.inventory_id
where
  r.rental_date::date = '2005-05-25'::date;
```

The results are displayed in a table with 15 rows and 12 columns. The columns are: film_id, title, release_year, language_id, original_language_id, rental_duration, rental_rate, and several other columns that are partially visible. The data shows 15 different film titles, all released in 2006, with rental rates ranging from 0.99 to 4.99.

film_id	title	release_year	language_id	original_language_id	rental_duration	rental_rate					
1	AIRPORT POLLOCK	2,006	1	[NULL]	6	4.99					
2	ALLEY EVOLUTION	2,006	1	[NULL]	6	2.99					
3	AMERICAN CIRCUS	2,006	1	[NULL]	3	4.99					
4	ANTHEM LUKE	2,006	1	[NULL]	5	4.99					
5	APACHE DIVINE	2,006	1	[NULL]	5	4.99					
6	ARMAGEDDON LOST	2,006	1	[NULL]	5	0.99					
7	BACKLASH UNDEFEATED	2,006	1	[NULL]	3	4.99					
8	BANGER PINOCCHIO	2,006	1	[NULL]	5	0.99					
9	BEACH HEARTBREAKERS	2,006	1	[NULL]	6	2.99					
10	BINGO TALENTED	2,006	1	[NULL]	5	2.99					
11	BINGO TALENTED	2,006	1	[NULL]	5	2.99					
12	BLOOD ARGONAUTS	2,006	1	[NULL]	3	0.99					
13	BOILED DARES	2,006	1	[NULL]	7	4.99					
14	BOOGIE AMELIE	2,006	1	[NULL]	6	4.99					
15	BORROWERS BEDAZZLED	2,006	1	[NULL]	7	0.99					

The interface also shows a status bar at the bottom indicating "137 row(s) fetched - 0.478s (0.004s fetch), on 2024-03-14 at 19:43:15".

UNOBVIOUS LIMITATIONS

- Huge schemas with hundreds and thousands of tables
- Long names for tables and columns
- Analytical databases
- Brand new databases with a lack of documentation

SINGLE REQUEST VS CHAT

SINGLE REQUEST

- Every request is brand new
- The user plays with a formulation
- The query structure is fixed

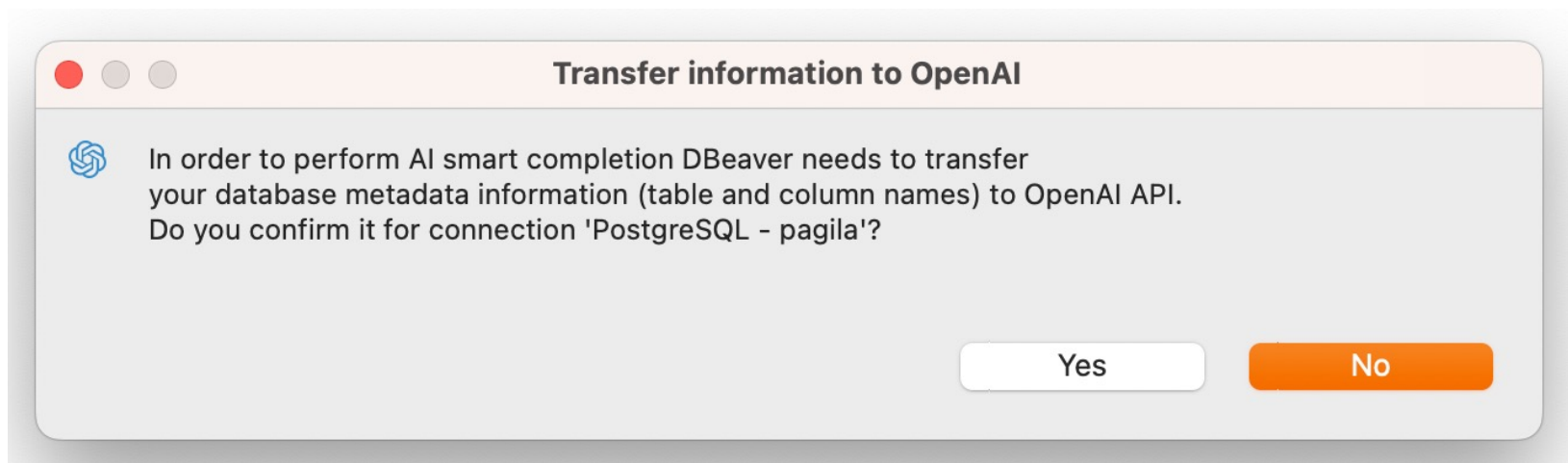
CHAT MODE

- A dialog instead of an exact match
- The user can add details one by one
- Any query can be created in the dialog

WHERE ELSE CAN AI HELP?

- ✓ Get a general overview of the existing database schema
- ✓ Get an explanation about the particular database object
- ✓ Optimize selects if they return errors or work too slow
- ✓ Generate mock data through insert statements

IS IT SAFE?



Enterprise privacy at OpenAI

Trust and privacy are at the core of our mission at OpenAI. We're committed to privacy and security for ChatGPT Enterprise and our API Platform.

[Read commitments ↓](#)

Our commitments

Ownership: You own and control your data

- ✓ We do *not* train on your business data, and our models don't learn from your usage
- ✓ You own your inputs and outputs (where allowed by law)
- ✓ You control how long your data is retained (ChatGPT Enterprise)

Control: You decide who has access

- ✓ Enterprise-level authentication through SAML SSO
- ✓ Fine-grained control over access and available features
- ✓ Custom models are yours alone to use, they are not shared with anyone else

Security: Comprehensive compliance

- ✓ We've been audited for SOC 2 compliance
- ✓ Data encryption at rest (AES-256) and in transit (TLS 1.2+)
- ✓ Visit our [Trust Portal](#) to understand more about our security measures

<https://openai.com/policies/api-data-usage-policies>

IS IT SAFE?

Content License and Data Use

The license you grant to Google under the "[Submission of Content](#)" section in the API Terms also extends to any content (e.g., prompts, including associated files such as images or documents) you submit through the Services or any other API interface (e.g., Firebase Extensions). Google uses this data, consistent with our [Privacy Policy](#), to provide, improve, and develop Google products and services and machine learning technologies, including Google's enterprise features, products, and services.

To help with quality and improve our products, human reviewers may read, annotate, and process your API input and output. Google takes steps to protect your privacy as part of this process. This includes disconnecting this data from your Google Account and API key before reviewers see or annotate it. **Do not submit sensitive, confidential, or personal information to the Services.**

Google only uses data that you import or upload to the Services to tune models for that express purpose. Tuning data may be retained in connection with your tuned models for purposes of re-tuning when supported models change. When you delete a tuned model, the related tuning data is also deleted.

<https://ai.google.dev/terms>

WHAT ELSE DO WE HAVE?



GitHub Co-pilot



Llama 2



AWS CodeWhisperer



Claude 3

HOW GOOD IS AI?

Table 2: The Execution Accuracy (EX) of SOTA text-to-SQL models in BIRD. The human performance is also provided.

Models	Development Data		Testing Data	
	w/o knowledge	w/ knowledge	w/o knowledge	w/ knowledge
<i>FT-based</i>				
T5-Base	6.32	11.54 (+5.22)	7.06	12.89 (+5.83)
T5-Large	9.71	19.75 (+10.04)	10.38	20.94 (+10.56)
T5-3B	10.37	23.34 (+12.97)	11.17	24.05 (+12.88)
<i>ICL-based</i>				
Codex	25.42	34.35 (+8.93)	24.86	36.47 (+11.61)
ChatGPT	24.05	37.22 (+13.17)	26.77	39.30 (+12.53)
ChatGPT + COT	25.88	36.64 (+10.76)	28.95	40.08 (+11.24)
Human Performance	-	-	72.37	92.96 (+20.59)

Can LLM Already Serve as A Database Interface? A Big Bench for Large-Scale Database Grounded Text-to-SQLs,
Jinyang Li, Binyuan Hui, Ge Qu
arXiv:2305.03111v1 [cs.CL] 4 May 2023

WHAT ARE THE BEST USE CASES

- Good for users who are not familiar with SQL but need to work with databases
- Good for simple queries, even if they include a few joins
- Good for well-known databases with detailed documentation, like PostgreSQL
- Good point to start if you are learning the SQL language

USEFUL LINKS

EMAIL:	tati@dbeaver.com
DBEAVER GITHUB:	github.com/dbeaver
DBEAVER COMMUNITY:	dbeaver.io
DBEAVER PRO:	dbeaver.com

THANK YOU