VoiceQuerySystem: Natural Language Interface for Database Querying



Calvin Hickey

Code: https://github.com/00Calvin00/VoiceQuerySystem.git Video: https://youtu.be/rjq7ffxBv5U

What's the Problem?

Accessing structured data through SQL can be intimidating or inaccessible to non-technical users.

Querying a database shouldn't feel like programming!

I just want to know the top-selling artist... how hard could it be?

% select * from artists where sales == MAX

ERROR 1111 (HY000): Invalid use of group function

I finally figured it out after hours of research!



[home_folder@mac ~ % SELECT ar.Name, SUM(il.Quantity) AS total_quantity FROM InvoiceLine il]
 JOIN Track t ON il.TrackId = t.TrackId JOIN Album al ON t.AlbumId = al.AlbumId JOIN Artis
 t ar ON al.ArtistId = ar.ArtistId GROUP BY ar.Name ORDER BY total_quantity DESC LIMIT 1;



That took me forever, and it was just one question!

How do I do filters? Aggregates? Groupings?

Why is my query returning duplicates?

What's a join?

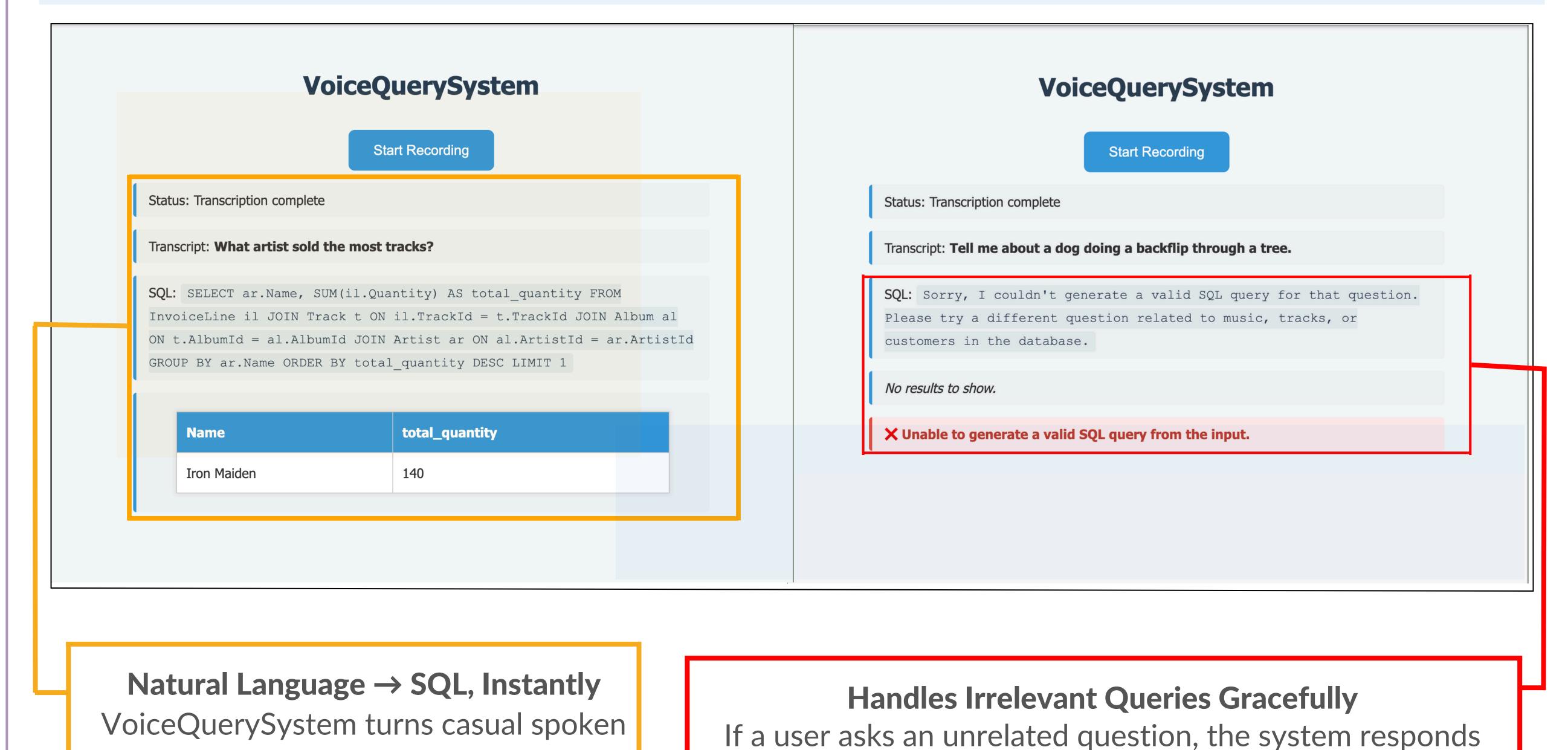
Is this even the right answer?

What does this SQL error even mean??

Challenges with Automating Database Queries

- How do we understand vague or ambiguous spoken questions?
- How do we translate natural language into precise SQL syntax?
- How do we know if the generated SQL query is correct?
- How should the system handle irrelevant or nonsensical voice input?





Bringing Humans into the Loop to Explore Data by Voice

politely without crashing — keeping the experience smooth.

VoiceQuerySystem enables users — even without SQL knowledge — to interact with databases naturally, by:

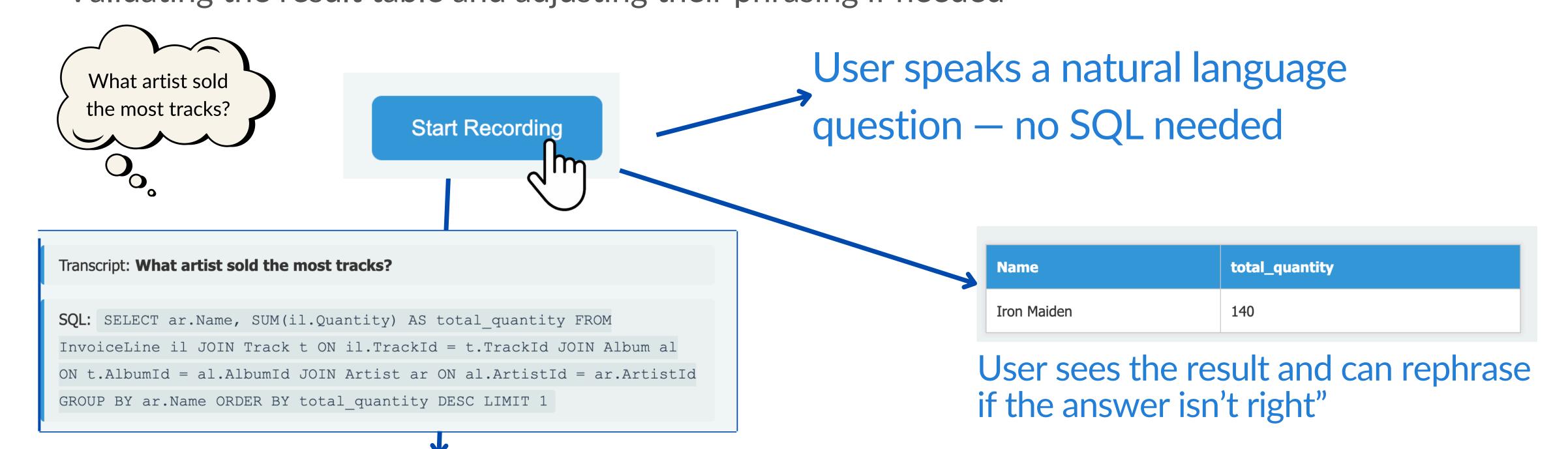
Asking simple questions out loud (no typing required)

VoiceQuerySystem transcribes and translates to SQL instantly

questions into real SQL queries — no

typing, no syntax memorization.

- Seeing exactly how their question was interpreted (transcript + SQL)
- Validating the result table and adjusting their phrasing if needed



VoiceQuerySystem in a nutshell

Smart Architecture from Input to Insight

- Whisper API for robust speech-to-text transcription
- smolagents + LLaMA 3 for powerful text-to-SQL translation
- Detects irrelevant questions and avoids hallucinated SQL
- Warns users with clear error messages when needed

Transparent by design:

- Shows users exact SQL being executed
- Enables learning and debugging for beginners
- Promotes trust and clarity in Al-driven interactions

Notable Development Stats:

- Over 20 unique natural language queries tested
- Handles joins, aggregations, filters, and ordering
- Automatically transcribes and parses queries in ~5 seconds
- Custom logic detects malformed or empty SQL blocks to avoid crashes

Educational Potential:

This tool acts as a learning bridge between humans and databases:

- Introduces beginners to SQL syntax without tutorials
- Encourages exploration without penalty
- Makes data accessible and less intimidating

References:

[1] Tao Guo, Xiaoxue Zang, Ziru Zhou, and Weikang Qian.
Voice2SQL: Towards End-to-End Spoken Language Interface to Databases
In SIGMOD Conference 2022. [1] S Gulwani. 2011. Automating string processing in spreadsheets using input-output examples. In POPL. 317–330. https://doi.org/10.1145/1926385.1926423
[2] M Raza and S Gulwani. 2017. Automated data extraction using predictive program synthesis. In AAAI, Vol. 31. https://doi.org/10.1609/aaai.v31i1.10668
[2] OpenAl.
Whisper: Robust Speech Recognition via Self-Supervised Learning.
GitHub repository: [1] S Gulwani. 2011. Automating string processing in spreadsheets using input-output examples. In POPL. 317–330. https://doi.org/10.1145/1926385.1926423
[2] M Raza and S Gulwani. 2017. Automated data extraction using predictive program synthesis. In AAAI, Vol. 31. https://doi.org/10.1609/aaai.v31i1.10668
[3] Hugging Face smol-ai/smol-developer.
smol developer: A tiny developer agent that builds software for you.
GitHub repository: [1] S Gulwani. 2011. Automating string processing in spreadsheets using input-output examples. In POPL. 317–330. https://doi.org/10.1145/1926385.1926423
[2] M Raza and S Gulwani. 2017. Automated data extraction using predictive program synthesis. In AAAI, Vol. 31. https://doi.org/10.1609/aaai.v31i1.10668
[4] Meta AI.
LLaMA 3: Open Foundation Language Models.

Meta AI Blog: [1] S Gulwani. 2011. Automating string processing in spreadsheets using input-output examples. In POPL. 317–330. https://doi.org/10.1145/1926385.1926423
[2] M Raza and S Gulwani. 2011. Automating string processing in spreadsheets using input-output examples. In POPL. 317–330. https://doi.org/10.1145/1926385.1926423
[2] M Raza and S Gulwani. 2017. Automated data extraction using predictive program synthesis. In AAAI, Vol. 31. https://doi.org/10.1609/aaai.v31i1.10668