

SQL Generation via LLMs

User Question

What cartoons were written by Joseph Kuhr?



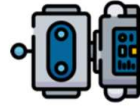
User

Schema

TABLE TV_Channel

TABLE TV_series

```
TABLE Cartoon
  "id" real,
  "Title" text,
  "Directed_by" text,
  "Written_by" text,
  "Production_code" real,
  "Channel" text,
  PRIMARY KEY ("id"),
  FOREIGN KEY ("Channel")
  REFERENCES "TV_Channel"("id")
```



LLM

Generated SQL Query

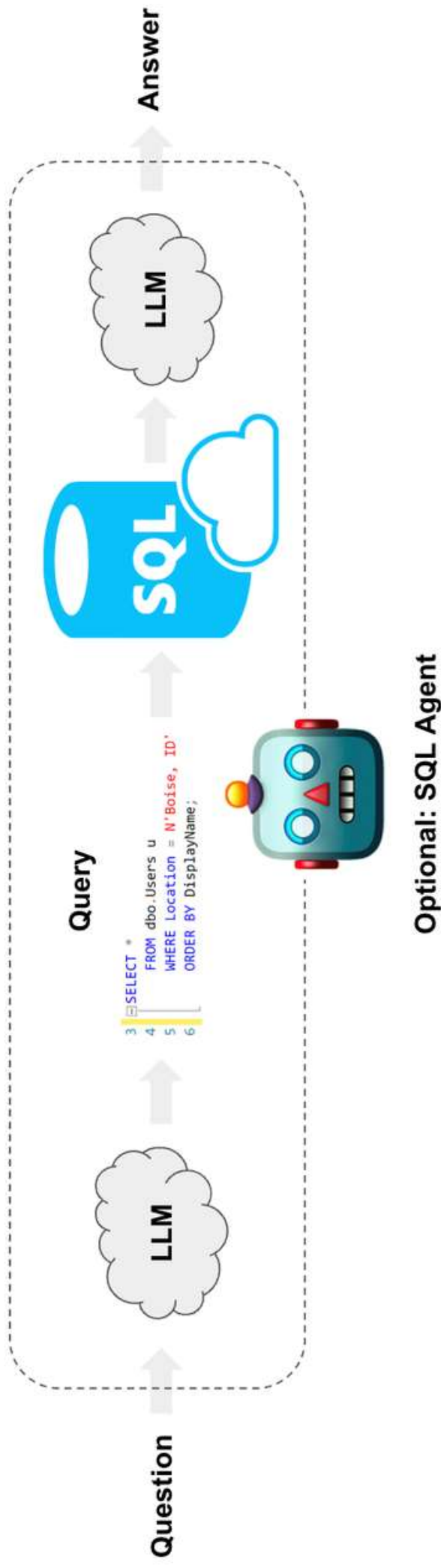
```
SELECT * FROM Cartoon
WHERE Written_by = "Joseph Kuhr"
```



Database



Batman
Series



Initial Prompt

```
-- language: PostgreSQL  
A SQL Query to ...
```

Schema Details

```
{ "name":  
  "bitdotio/palmerpenguins",  
  "schema": [ { "name":  
    "public",  
    ... } ]  
}
```

User Prompt

What is the average weight of penguins on Biscoe?

Final prompt composed from:

Initial Prompt +
Selection of Schema Details +
User Prompt

Send Complete Prompt to OpenAI
Codex Model API



Output SQL

```
SELECT *  
FROM ...
```

Review and Execute
Query



Idle ●

PING 33ms

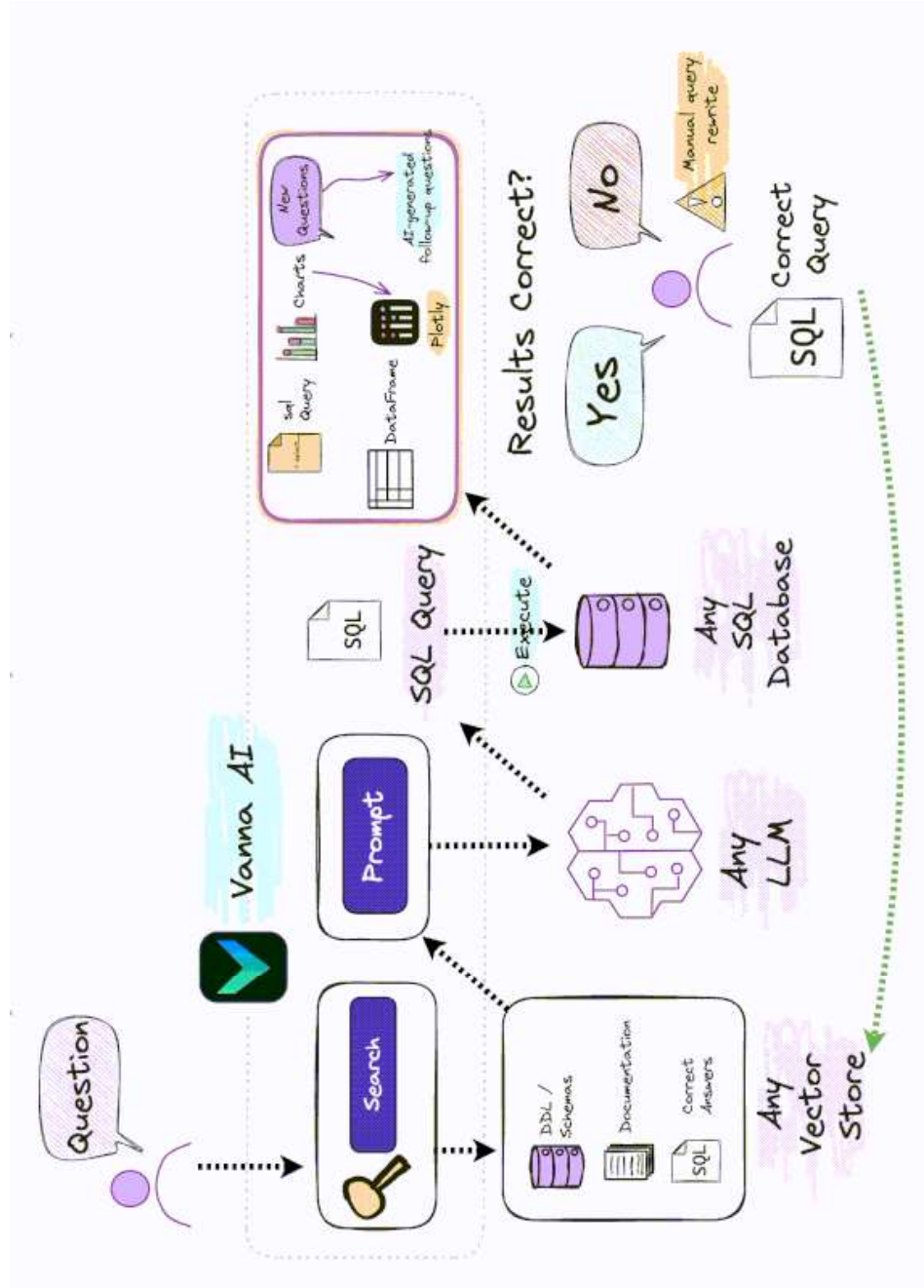
Run query

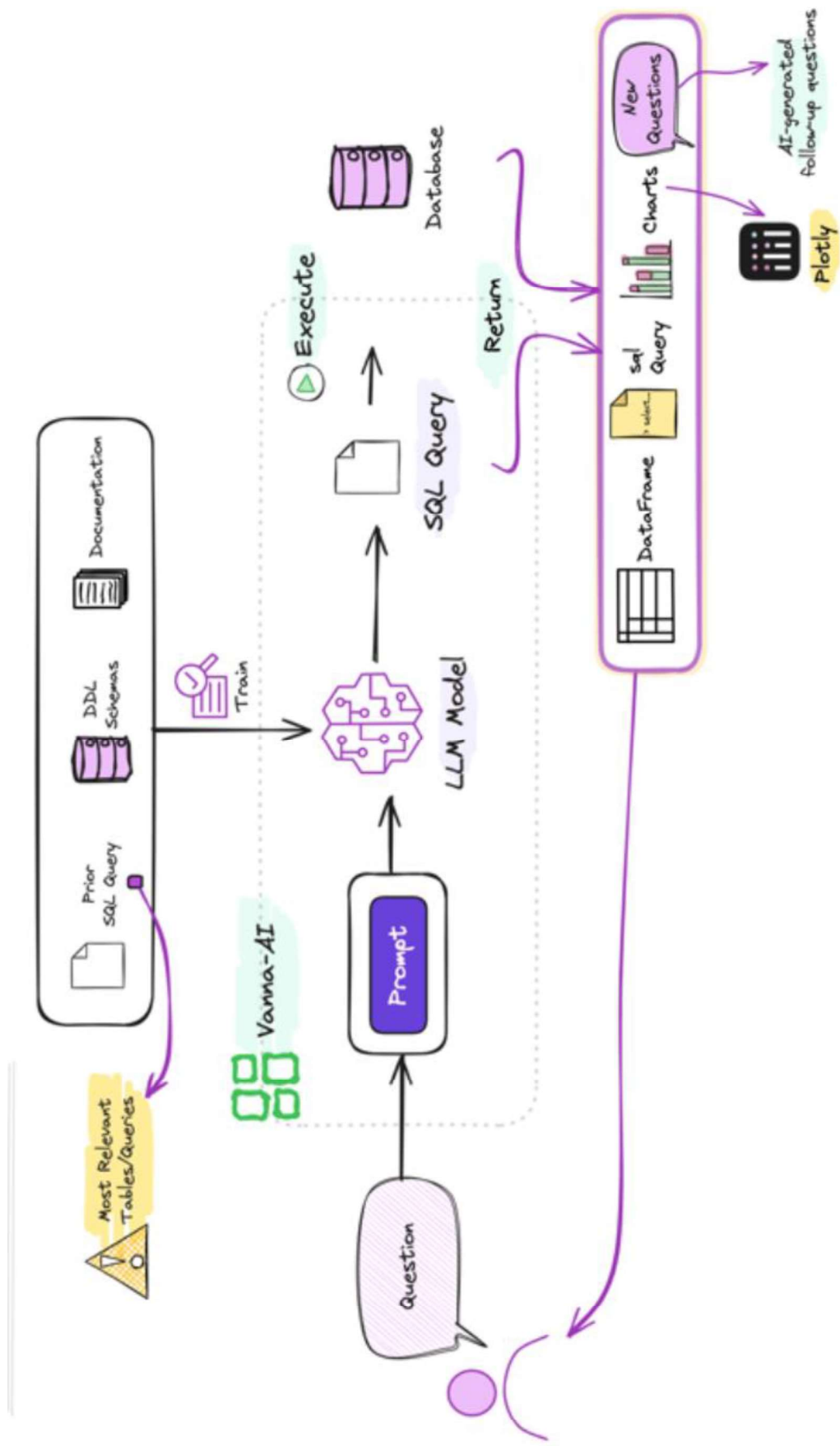
- 1 `#!translate:text`
- 2 what record with provider 'aws' has the lowest latency?

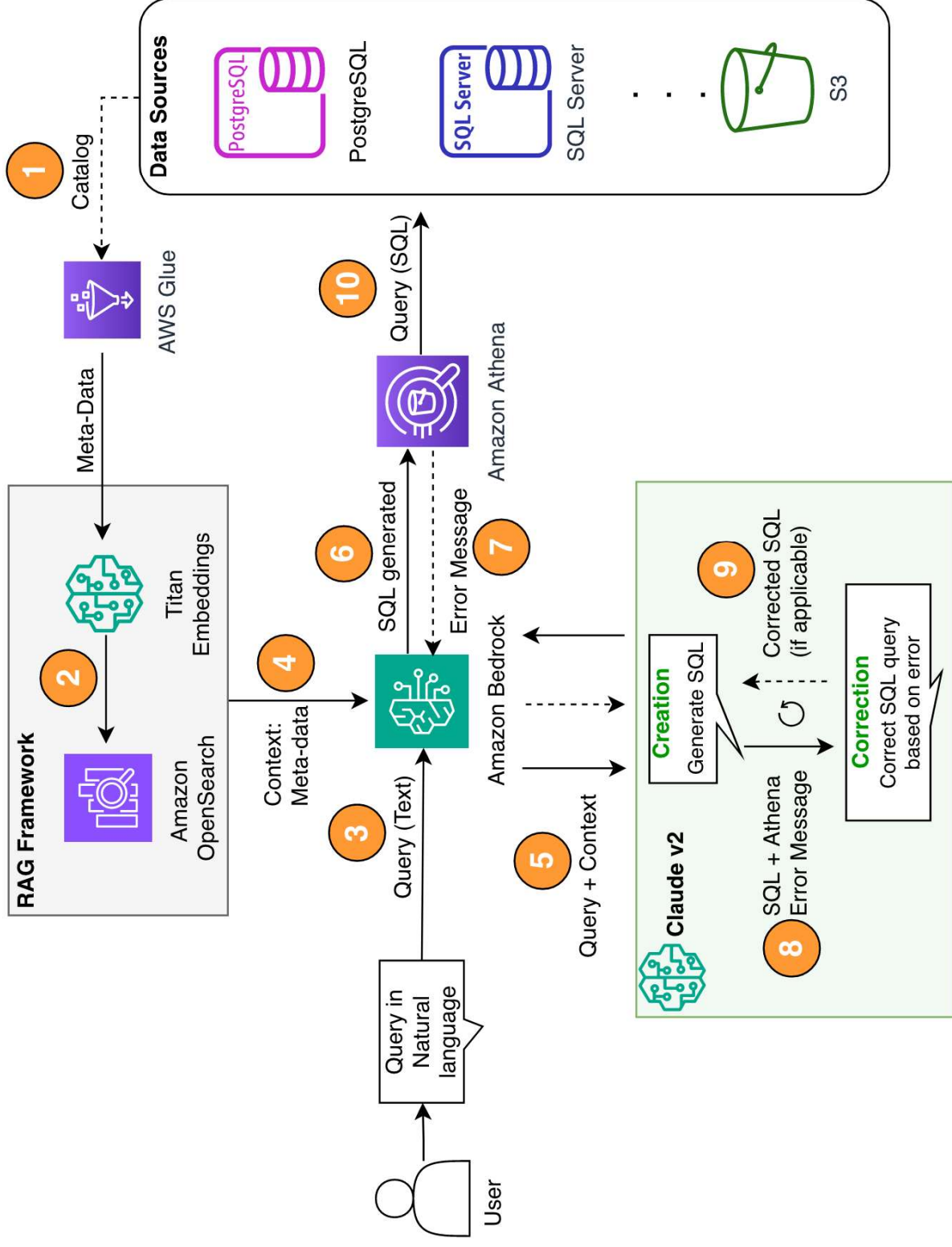
Q Hide Query Translation

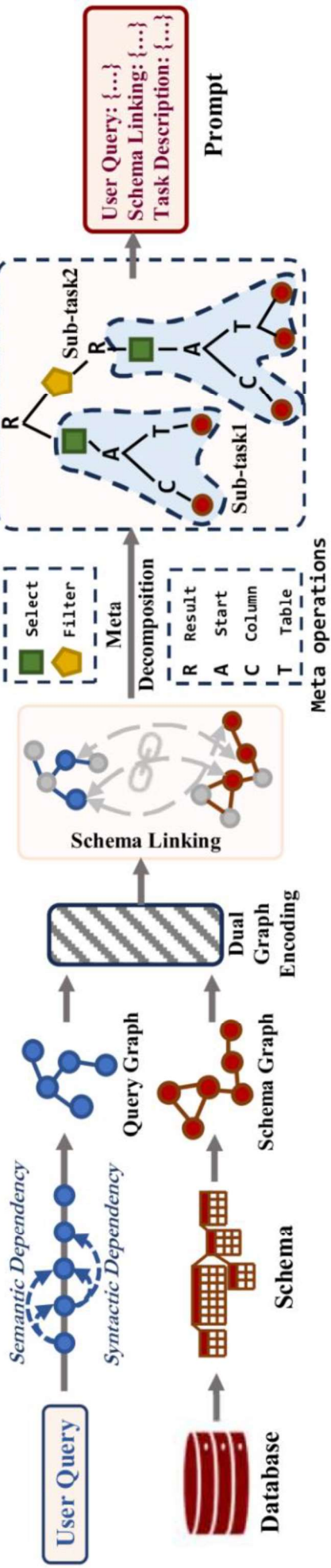
```
1 SELECT * FROM latency_from_datacenter WHERE provider =  
  'aws' ORDER BY latency_ms ASC LIMIT 1;
```

Replace in Editor







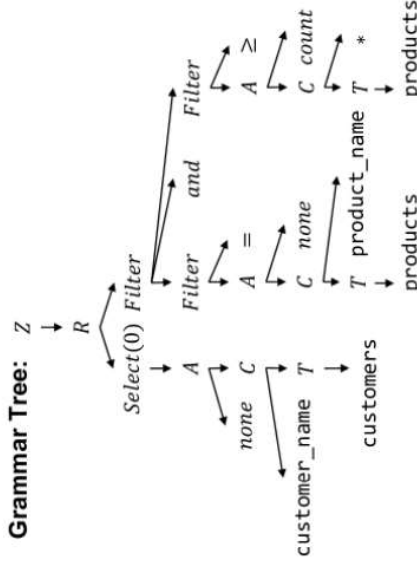


$Z ::= \text{intersect } R \ R \mid \text{union } R \ R \mid \text{except } R \ R \mid R$
 $R ::= \text{Select}(n)$
 $\quad \mid \text{Select}(n) \text{ Filter} \mid \text{Select}(n) \text{ Order}(d)$
 $\quad \mid \text{Select}(n) \text{ Sup}(p) \mid \text{Select}(n) \text{ Order}(d) \text{ Filter}$
 $\quad \mid \text{Select}(n) \text{ Sup}(p) \text{ Filter}$
 $\text{Select}(n) ::= A_0 \ A_1 \ \dots \ A_{n-1}$
 $\text{Order}(d) ::= A$
 $\text{Sup}(p) ::= A$
 $A ::= \text{max } C \mid \text{min } C \mid \text{count } C \mid \text{sum } C \mid \text{avg } C \mid \text{none } C$
 $C ::= c \ T$
 $T ::= t$
 $\text{Filter} ::= \text{and Filter Filter} \mid \text{or Filter Filter}$
 $\quad \mid > A \mid > A \ R \mid < A \mid < A \ R$
 $\quad \mid \geq A \mid \geq A \ R \mid = A \mid = A \ R$
 $\quad \mid \neq A \mid \neq A \ R \mid \text{between } A$
 $\quad \mid \text{like } A \mid \text{not like } A \mid \text{in } A \ R \mid \text{not in } A \ R$
 $n \in \{0, 1, 2, 3, \dots\}$
 $d \in \{\text{asc, desc}\}$
 $p \in \{\text{most, least}\}$
 c ranges over distinct column names
 t ranges over table names

NL: List the names of the customers who have once bought product "food".

SQL: `SELECT T1.customer_name`
`FROM customers AS T1 JOIN orders AS`
`T2 JOIN order_items AS T3 JOIN`
`products AS T4`
`WHERE T4.product_name = "food"`
`GROUP BY T1.customer_id`
`HAVING count(*) >= 1`

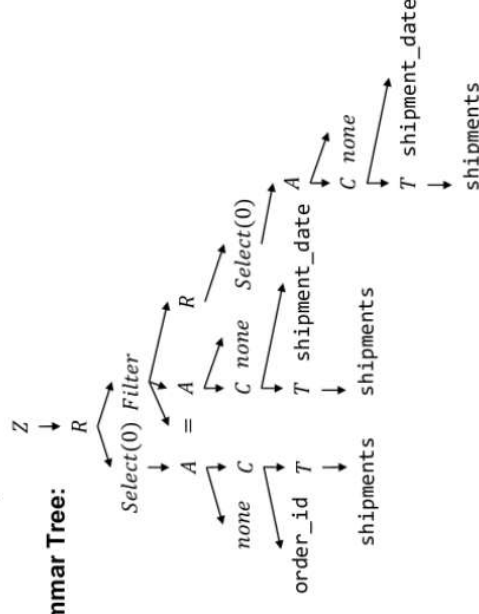
Grammar Tree:



NL: Which order has the most recent shipment? Give me the order id.

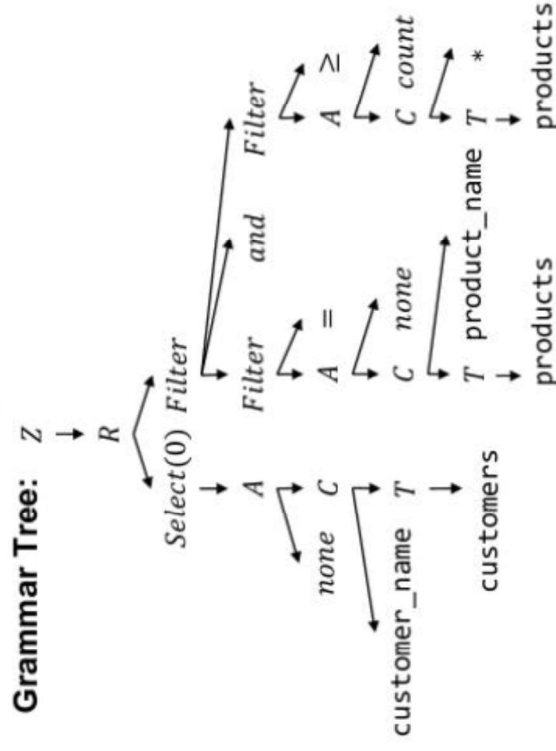
SQL: `SELECT order_id`
`FROM shipments`
`WHERE shipment_date = (`
`SELECT max(shipment_date)`
`FROM shipments`
`)`

Grammar Tree:



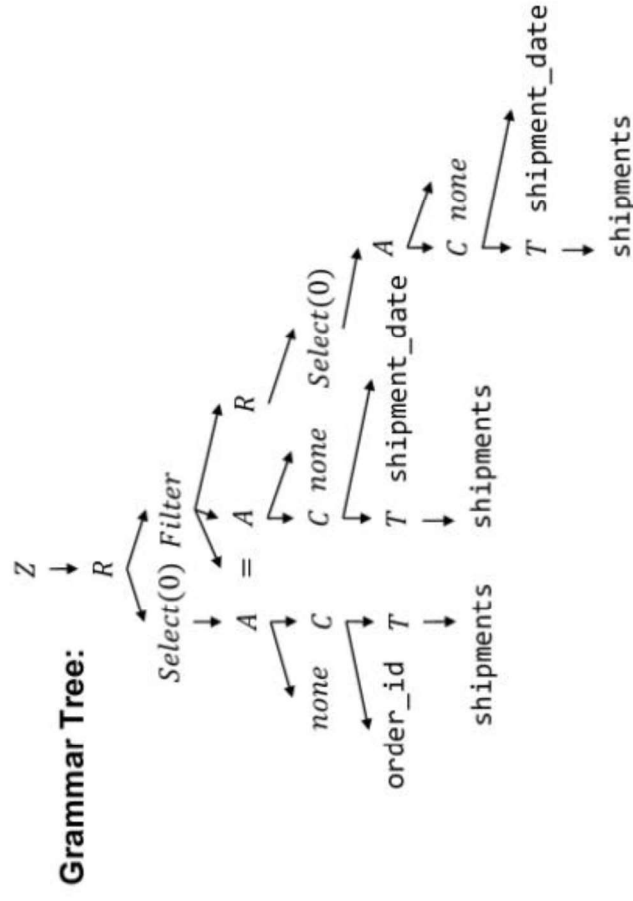
NL: List the names of the customers who have once bought product "food".

SQL: SELECT T1.customer_name
FROM customers AS T1 JOIN orders AS
T2 JOIN order_items AS T3 JOIN
products AS T4
WHERE T4.product_name = "food"
GROUP BY T1.customer_id
HAVING count(*) >= 1



NL: Which order has the most recent shipment?
Give me the order id.

SQL: SELECT order_id
FROM shipments
WHERE shipment_date = (
SELECT max(shipment_date)
FROM shipments
)



Questions:

- (1) How does this process differ from normal computer code generation?
- (1) Pros and Cons vs. symbolic Text-SQL generation?