



**VERSION 1**

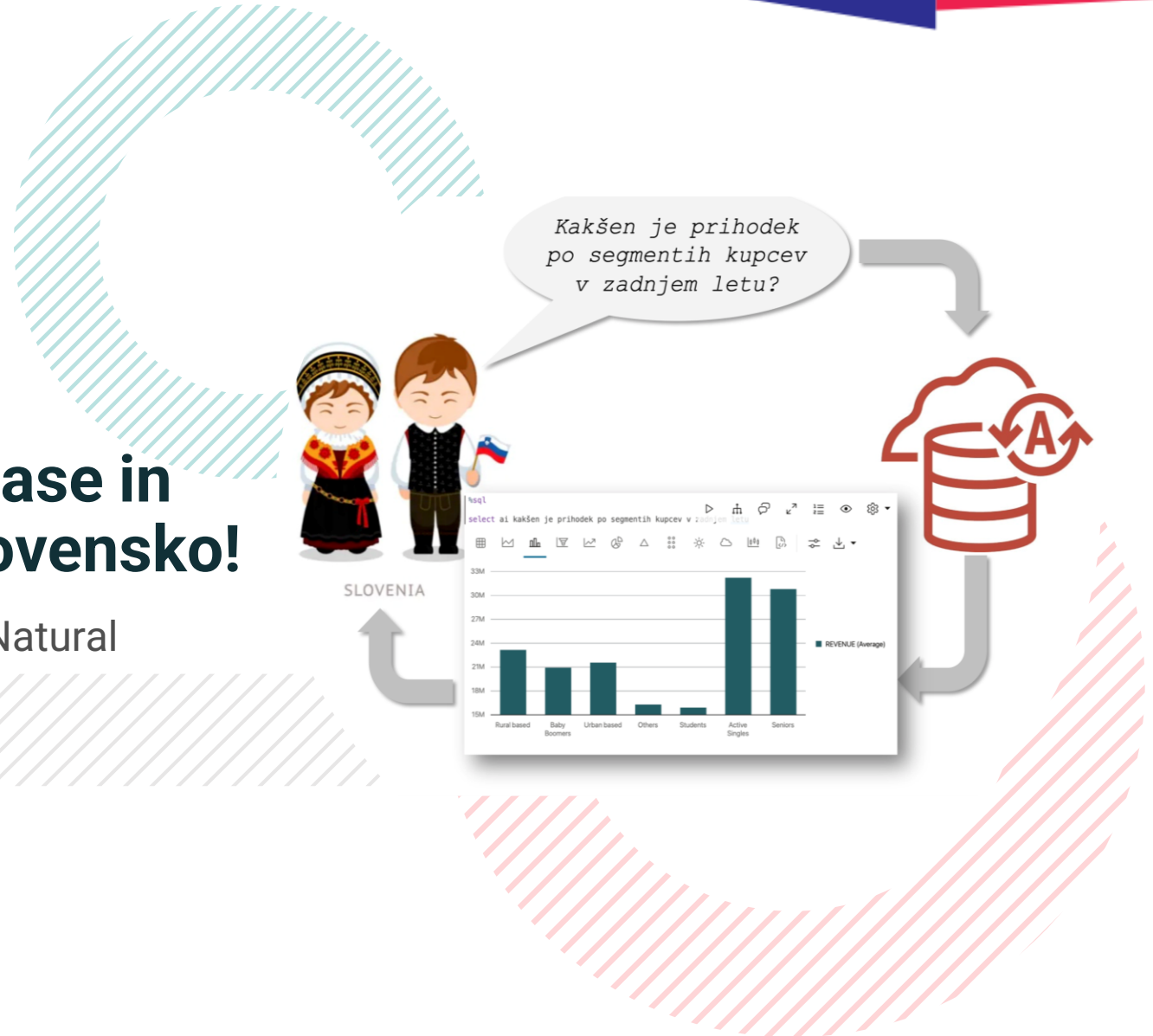
# Talking to Oracle Database in plain English ... in po slovensko!

Use Select AI to Generate SQL from Natural Language Prompt

**Presented By:** Žiga Vaupot

**Date:** Portorož, 30. maj 2024

**Company Classification:** Controlled





# Žiga Vaupot



Country Manager, Slovenia  
Principal Consultant



Board Member




 [ziga.vaupot@version1.com](mailto:ziga.vaupot@version1.com)


 [zigavaupot.blogspot.com](http://zigavaupot.blogspot.com)

 [linkedin.com/in/zigavaupot](https://linkedin.com/in/zigavaupot)


# Version 1



**26 Years**  
Proving Value of I.T.



**3400+**  
People



**12**  
Global Locations



**€350M+**  
Revenue

---

**88%**  
Employee Engagement  
Score (Q2 2023)

---

**€3.5M**  
Annual investment in  
Employee Skills and  
Development

---

**86%**  
Customer Satisfaction  
Score (Q3 2023)

**6**  
Strategic Technology  
Partners










---

**12 Years**  
As a Great Place to Work







---

**Industry Best Practice**






---











---









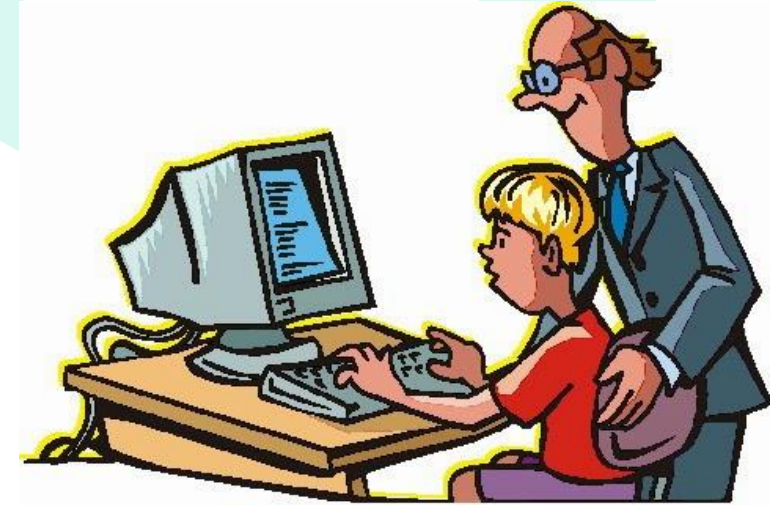


# SQL

SQL stands for **Structured Query Language**. As its name suggests, SQL is a programming language used to interact with databases. You can use SQL to ask questions (queries) of databases, retrieve data, modify records, and manage database structures. It's primarily used to communicate with **relational databases**.

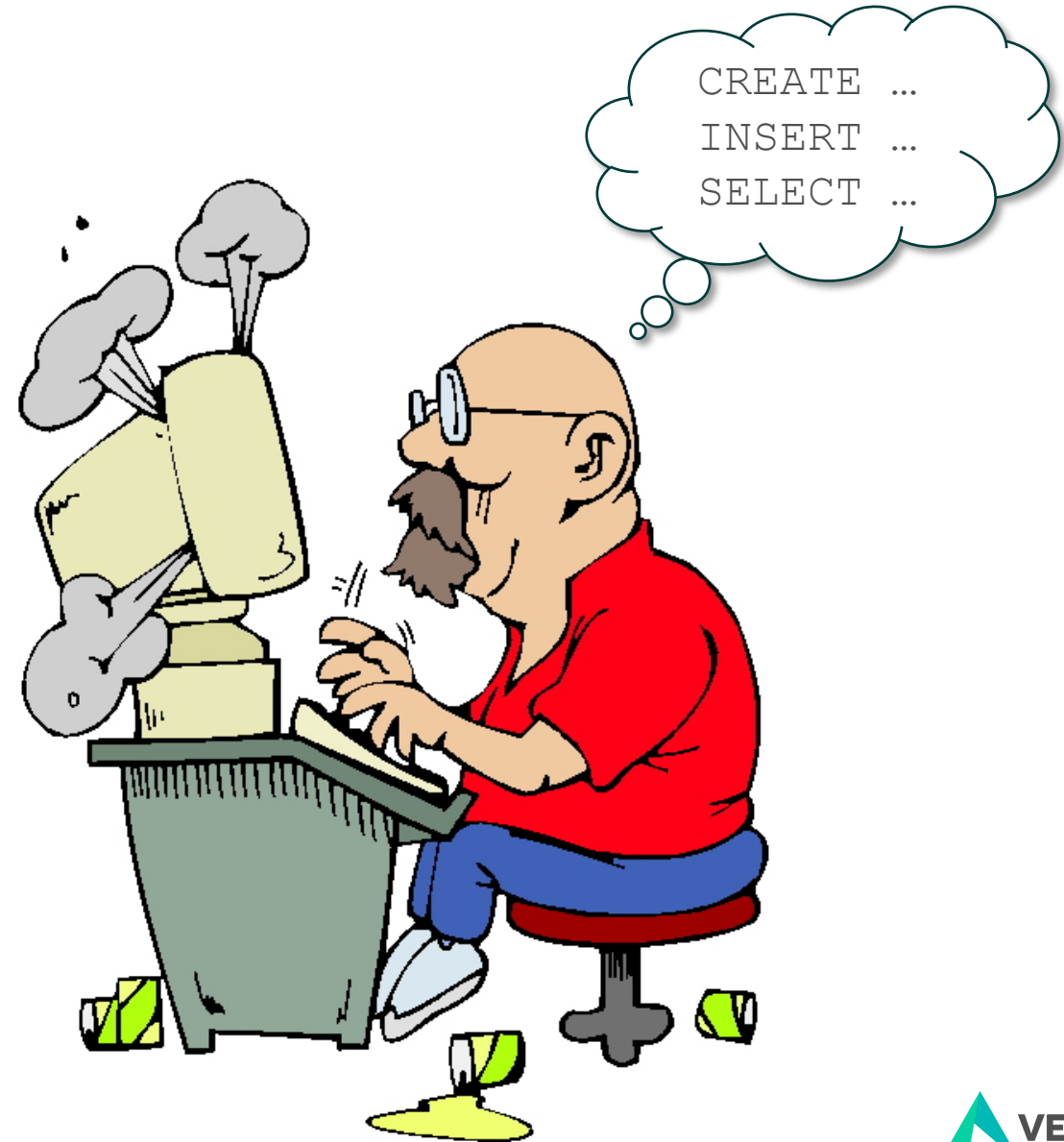
- 1970: Codd invents relational model / algebra
- 1974: Chamberlin defines SEQUEL (Structured English Query Language)
- 1976: SEQUEL/2 defined and renamed SQL
- 1978: 1<sup>st</sup> Commercial implementation by Oracle
- 1982: SQL standardisation begins
- 1986/1987: ANSI/ISO standard
- Several updates followed in the following years and the rest is history ...

# Back in 1988 (or was it in 1989?)...



`SELECT * FROM table`

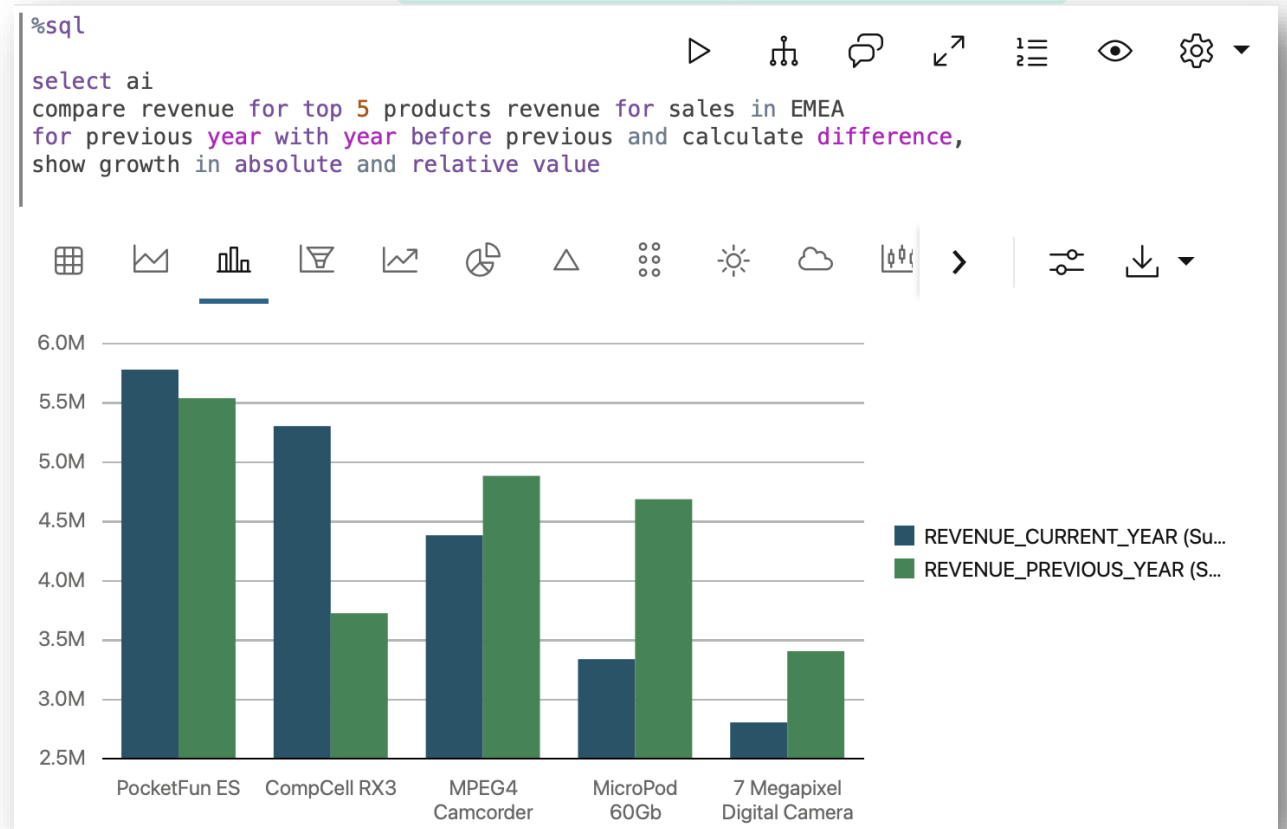
“SQL is the closest  
to natural language  
as it gets!”



# ORACLE Database 23<sup>ai</sup>

# SELECT AI

- Oracle Autonomous Database enables users to query data in natural language.
- Select AI feature allows ADB to use Generative AI with LLMs to convert input text into SQL.
- Select AI processes the natural language prompt, supplements it with metadata and generates and runs a SQL query.



# SQL Query Generation Process Flow



## 1. AI Configuration

Set up metadata and choose the desired Large Language Model (LLM), such as OpenAI or Cohere.

## 2. User Input

Users provide plain text instructions or questions through the product prompt.

3. SQL Generation: The provided text is transformed into an SQL query using the selected LLM.

4. Result Delivery: The generated SQL query yields results that are then displayed in the application



# Prerequisites

- Access to an Oracle Cloud Infrastructure cloud account and to an Autonomous Database instance.
- A paid API account for a supported AI provider, one of:
  - OpenAI
  - Cohere
  - Azure OpenAI Service
  - OCI Generative AI
- Network ACL privileges to access your external AI provider.
- A credential that provides access to the AI provider.

# Configure DBMS\_CLOUD\_AI

- Grant EXECUTE privilege on the DBMS\_CLOUD\_AI package to the user which will use Select AI.
- Grant network ACL access to the user who wants to use Select AI and for the AI provider endpoint.
- Create a credential to enable access to your AI provider.

```
grant execute on DBMS_CLOUD_AI to ADB_USER;

BEGIN
  DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE (
    host => 'api.openai.com',
    ace => xs$ace_type(privilege_list =>
      xs$name_list('http'),
      principal_name => 'ADB_USER',
      principal_type => xs_acl.ptype_db)
  );
END;
/

EXEC DBMS_CLOUD.CREATE_CREDENTIAL('OPENAI_CRED',
  'OPENAI', 'your_api_token');
```

# Configure AI Profiles

- ADB uses AI profiles to:
  - facilitate and configure access to an LLM and
  - to setup for the generation of SQL statements from natural language prompts.
- AI profiles include database objects that are the target for natural language queries.
- Metadata used from these targets can include database table names, column names, column data types, and comments.
- AI profiles can be created and configured using DBMS\_CLOUD\_AI:
  - DBMS\_CLOUD\_AI.CREATE\_PROFILE
  - DBMS\_CLOUD\_AI.SET\_PROFILE
- Targets can be also external tables.

# Create and set profile

```
dbms_cloud_ai.create_profile(  
  profile_name => 'openai_gpt',  
  attributes =>  
    '{"provider": "openai",  
     "credential_name": "OPENAI_CRED",  
     "object_list": [{"owner": "OABOOTCAMP", "name": "D_CUSTOMERS"},  
                    {"owner": "OABOOTCAMP", "name": "D_GEOGRAPHY"},  
                    {"owner": "OABOOTCAMP", "name": "D_PRODUCTS"},  
                    {"owner": "OABOOTCAMP", "name": "D_TIME"},  
                    {"owner": "OABOOTCAMP", "name": "F_BILL_REV"}  
                    ]  
    }'  
);  
  
dbms_cloud_ai.set_profile(  
  profile_name => 'openai_gpt'  
);
```

# AI Keyword

- The AI keyword in a SELECT statement instructs the SQL execution engine to use the LLM identified in the active AI profile to process natural language and to generate SQL.
- You can use the AI keyword in a query with Oracle clients such as SQL Developer, OML Notebooks, and third-party tools, to interact with database in natural language.
- Note that PL/SQL, DDL or DML statements cannot be run using AI keyword

# Use AI to enter prompts

```
SELECT AI action natural_language_prompt
```

Action	Description
<code>runsql</code>	Run the provided SQL command using a natural language prompt. This is the default action and it is optional to specify this parameter.
<code>showsql</code>	Displays the SQL statement for a natural language prompt.
<code>narrate</code>	The output of the prompt is explained in natural language. This option sends the SQL result to the AI provider to produce a natural language summary.
<code>chat</code>	Generates a response directly from the LLM based on the prompt. If conversation in the <code>DBMS_CLOUD_AI.CREATE_PROFILE</code> function is set to true, this option includes content from prior interactions or prompts, potentially including schema metadata.
<code>explainsql</code>	The SQL generated from the prompt is explained in natural language. This option sends the generated SQL to the AI provider to produce a natural language explanation.

# Where to start?

Oracle Machine Learning Notebooks might be a good starting point

The screenshot shows the Oracle Machine Learning interface. At the top, it says "ORACLE Machine Learning" and "OABOOTCAMP Project OABOOTCAMP Workspace". The notebook title is "SELECT AI Demo on OABOOTCAMP\_1". The code cell contains the following SQL query:

```
%sql  
select ai what is the revenue by customer segments in the last year
```

The query result is a bar chart titled "TOTAL\_REVENUE (Average)". The y-axis represents revenue in millions (M), ranging from 8M to 18M. The x-axis lists customer segments: Rural based, Baby Boomers, Urban based, Others, Students, Active Singles, and Seniors. The bars show the following approximate revenue values:

Customer Segment	Total Revenue (Average)
Rural based	11.5M
Baby Boomers	10.5M
Urban based	11M
Others	8.2M
Students	8.1M
Active Singles	16.5M
Seniors	15.5M

The chart also includes a legend for "TOTAL\_REVENUE (Average)" and a timestamp "2s 524ms @ a day ago".

# Can SELECT AI speak other languages than English?

%sql

```
select ai
prikaži prihodke, skupne stroške, profit in profitne stopnje,
delež fiksnih stroškov v skupnih stroških, delež variabilnih stroškov v skupnih stroških,
po blagovni znamki, segmentih kupcev in regijah za lani
```

Type to search

BRAND	CUSTOMER_SEGMENT	REGION	REVENUE	TOTAL_COSTS	PROFIT	PROFIT_MARGIN
FunPod	Students	AMERICAS	1052404 59	952922 11725	94481 47275	8.9691522192228272425414782082922194216E00
FunPod	Active Singles	AMERICAS	22			595E00
BizTech	Rural based	Ur				102E00
FunPod	Urban based	Al			85	8.60093072305221634670175105953121954728E00

%sql

```
select ai
εμφάνιση εσόδων, συνολικού κόστους, κέρδους και ποσοτών κέρδους,
μερίδιο σταθερού κόστους στο συνολικό κόστος, μερίδιο μεταβλητού κόστους στο συνολικό κόστος,
ανά επωνυμία, τμήματα πελατών και περιοχές για το περασμένο έτος
```

%sql

```
select ai
収益、総コスト、利益、利益率を表示します。
総費用に占める固定費の割合、総費用に占める変動費の割合、
昨年のブランド、顧客セグメント、地域別
```

%sql

```
select ai
приказати приходе, укупне трошкове, профит и стопе профита,
учешће фиксних трошкова у укупним трошковима, учешће варијабилних трошкова у укупним трошковима,
по бренду, сегментима купаца и регионима за прошлу годину
```

# Note: when using SQL Worksheet

```
EXEC DBMS_CLOUD_AI.set_profile('openai_gpt');
```

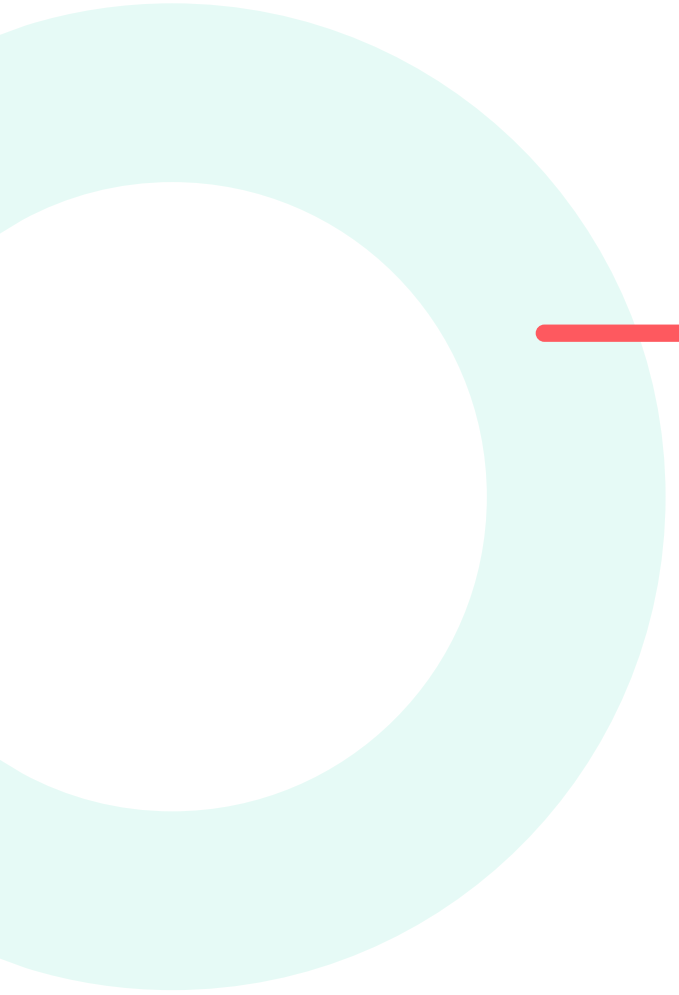
```
SELECT DBMS_CLOUD_AI.GENERATE(prompt      => 'what is the revenue by customer  
                                segments in the last year',  
                                profile_name => 'openai_gpt',  
                                action       => 'narrate')  
FROM dual;
```

The revenue by customer segments in the last year is as follows:

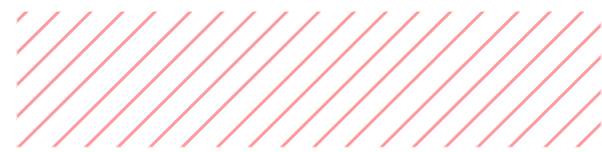
- Rural based: \$11,509,313.81
- Baby Boomers: \$10,499,165.66
- Urban based: \$10,987,702.76
- Others: \$8,254,998.03
- Students: \$8,156,688.95
- Active Singles: \$16,410,287.84
- Seniors: \$15,385,910.97

```
DBMS_CLOUD_AI.GENERATE (  
    prompt      IN    CLOB,  
    profile_name IN    VARCHAR2 DEFAULT NULL,  
    action       IN    VARCHAR2 DEFAULT NULL,  
    attributes   IN    CLOB      DEFAULT NULL  
) RETURN CLOB;
```

Make IT



**DEMO**





# VERSION 1