

ā'pěks alpe adria 26

# State of Things: Oracle APEX and Generative Development

**Christian Rokitta**

Consulting Member of Technical Staff  
Oracle APEX Development Team



# Tech Superstars Unite

Get worldwide recognition as an Oracle ACE



Oracle.com profile page



Swag, certification exam credit & event passes



Exclusive content



Networking events



Your own Oracle cloud account



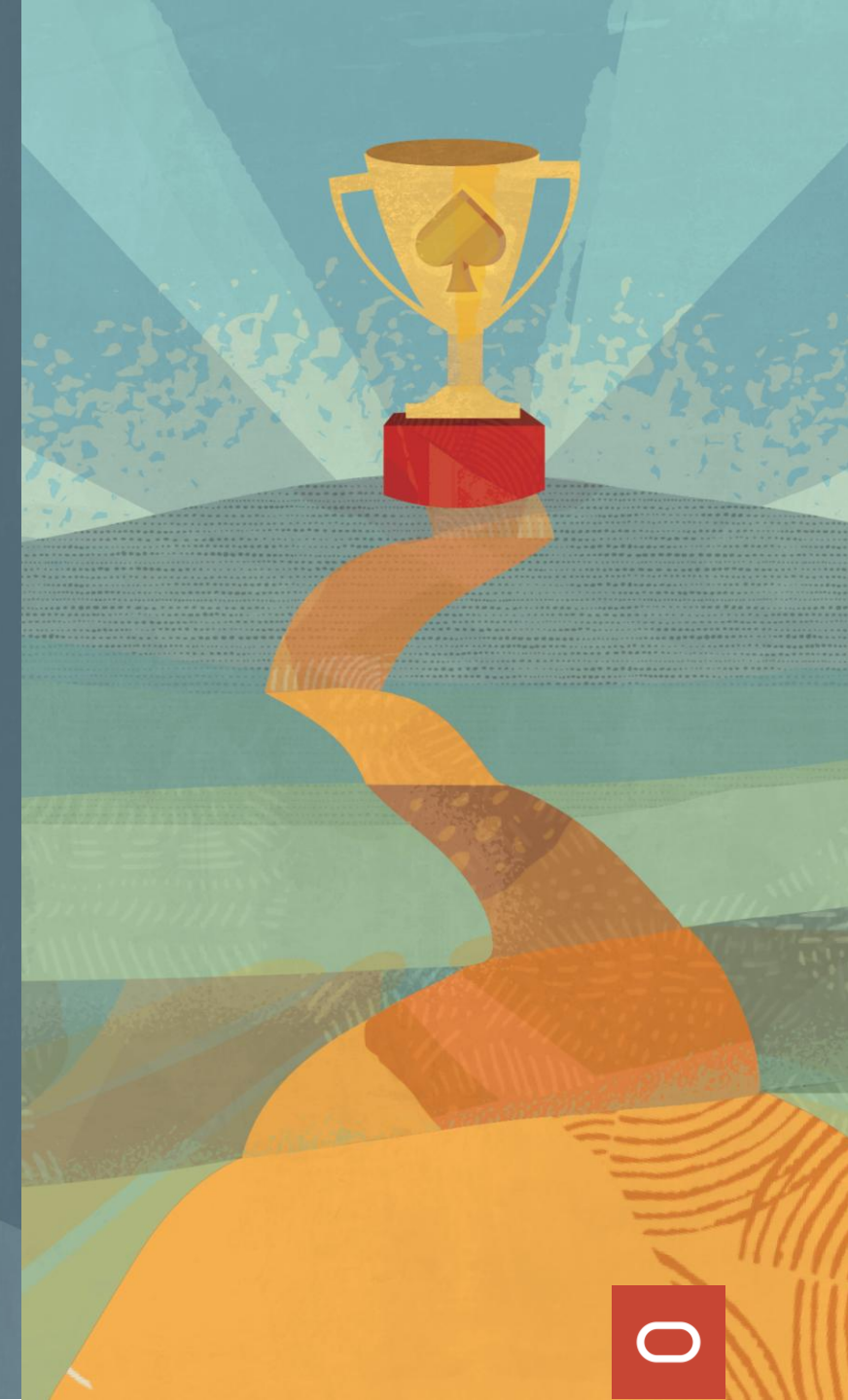
Travel support

✕ @oracleace

🌐 [Linkedin.com/groups/1796302](https://www.linkedin.com/groups/1796302)

🦋 @oracleace.bsky.social

Learn more at:  
[ace.oracle.com](https://ace.oracle.com)



# Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

# Keynote → Key Features → AI

(No Worries: you won't miss a thing)

# The APEX Team @ APEX Alpe Adria



**Monica Godoy**

**Transform Oracle Forms Apps  
into APEX Apps in the Era of AI**

09:50, Room A



**Andreea Munteanu**

**CI/CD with APEXlang**

10:45, Room A



**Menno Hoogendijk**

**What else in APEX 26.1**

15:35, Room A



**Christian Rokitta**

**Universal Theme: The Refresh**

16:40, Room A



**Menno Hoogendijk**

**Let's Talk APEXlang**

17:35, Room A

# APEX 26.1: Are We There Yet?

- APEX 26.1 is coming soon (yes, we know “soon” is doing a lot of work here).
- The cycle is longer than our usual twice-a-year cadence due to APEXlang and major new AI capabilities.
- APEXlang isn’t a bolt-on: it required deep foundational changes to the APEX metadata model; incremental releases weren’t practical.
- While 26.1 has been “in the oven,” we’ve still shipped monthly 24.2 updates (14 PSEs so far).
- After 26.1, we expect to return to a more frequent cadence.



# Agenda

**AI Assisted  
Application Generation**

**APEXlang**

**AI Enhanced  
Applications**

**New Components &  
Enhancements**

# The AI Rush to Generate Code

Everyone's racing to generate more code, but complexity doesn't scale.

- AI tools promise instant app generation
- Output: thousands of lines of JavaScript and glue code
- More code = more to debug, test, maintain = more tech debt
- AI shows promise, but without abstraction we're just accelerating complexity.

The more code you generate,  
the more complexity you inherit.

No one understands it (AI generated code)  
because no one wrote it.



# Real Productivity = Less Code

The future of app development isn't in generating code — it's in eliminating it.

- Productivity comes from **abstraction, not automation**
- **Automation** speeds up effort — temporarily
- **Abstraction** eliminates effort – permanently
- less to code, test, and maintain
- While AI can write JavaScript, **APEX removes the need for it** through **declarative definitions** — and with AI, those definitions become conversational

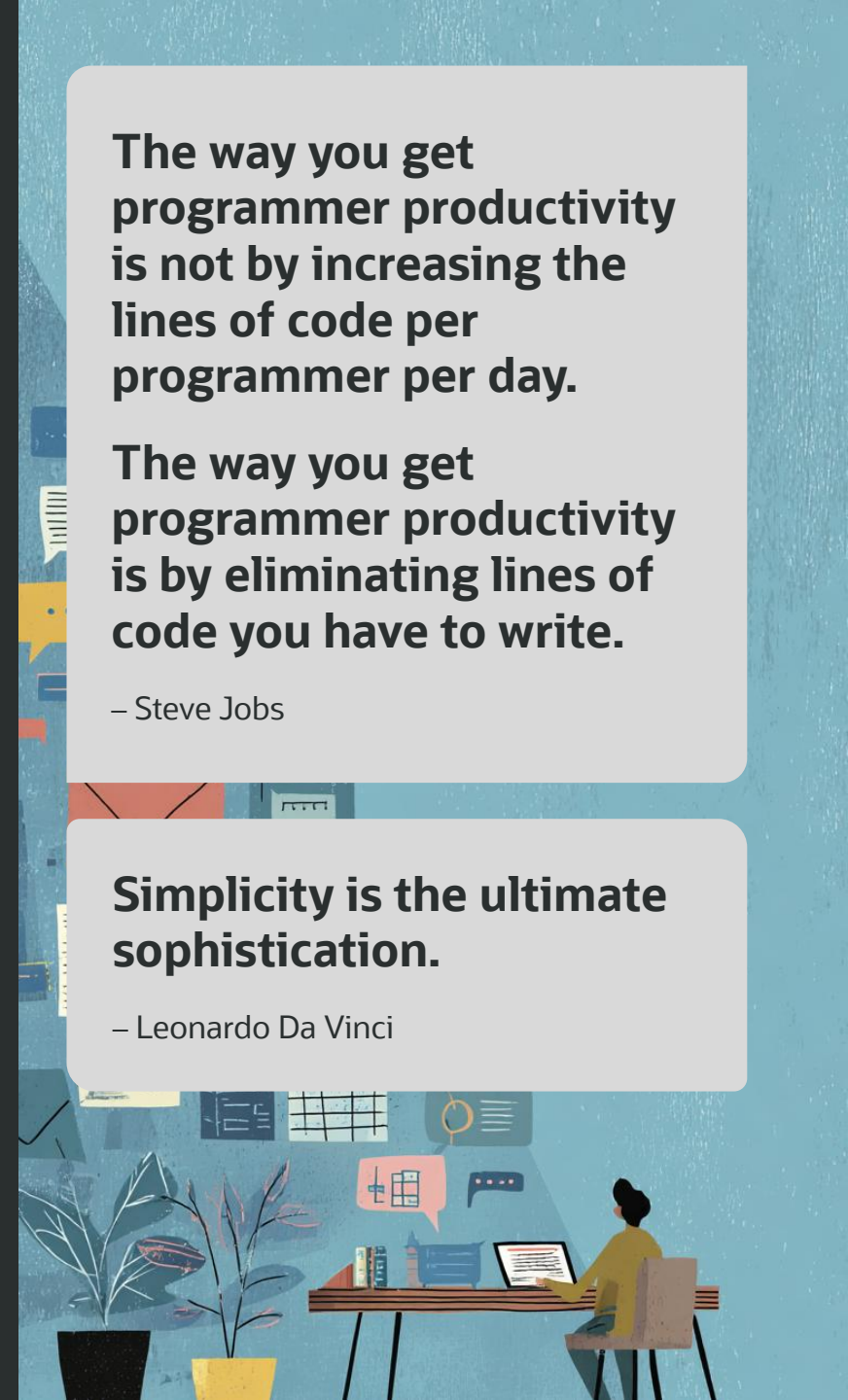
The way you get programmer productivity is not by increasing the lines of code per programmer per day.

The way you get programmer productivity is by eliminating lines of code you have to write.

– Steve Jobs

Simplicity is the ultimate sophistication.

– Leonardo Da Vinci



# The APEX Architecture Advantage

**APEX was built on the principles AI now depends on: abstraction, intent, and governance.**

- **Model-driven execution engine abstracts code by design**  
Every aspect of an APEX app is captured as structured attributes, not generated source files.
- **Logic, UI, and data are defined declaratively**  
The platform interprets intent instead of executing custom scripts.
- **AI builds on this foundation safely and predictably**  
Structured abstraction layer ensures governed, explainable results.
- **Upgrade-safe by architecture**  
Platform updates apply to the engine, not individual apps.

**APEX apps built with AI are smaller, safer, and sustainable — not by generating code, but by defining intent.**



# Generative AI in Oracle APEX

## AI Assisted Application Generation

The APEX AI Assistant is an integrated, conversational companion built directly into the development experience.

From generating entire applications to adding new pages to existing ones, it supports you across the full development lifecycle.

Use it to generate, optimize, explain, or debug SQL queries, or get hands-on help with HTML, CSS, JavaScript, and PL/SQL — available in every code editor across APEX.

## AI Enhanced Applications

Deliver richer, more intelligent user experiences by tapping into native platform capabilities - including vector-based similarity search and Retrieval Augmented Generation (RAG) to ground AI responses in your own data.

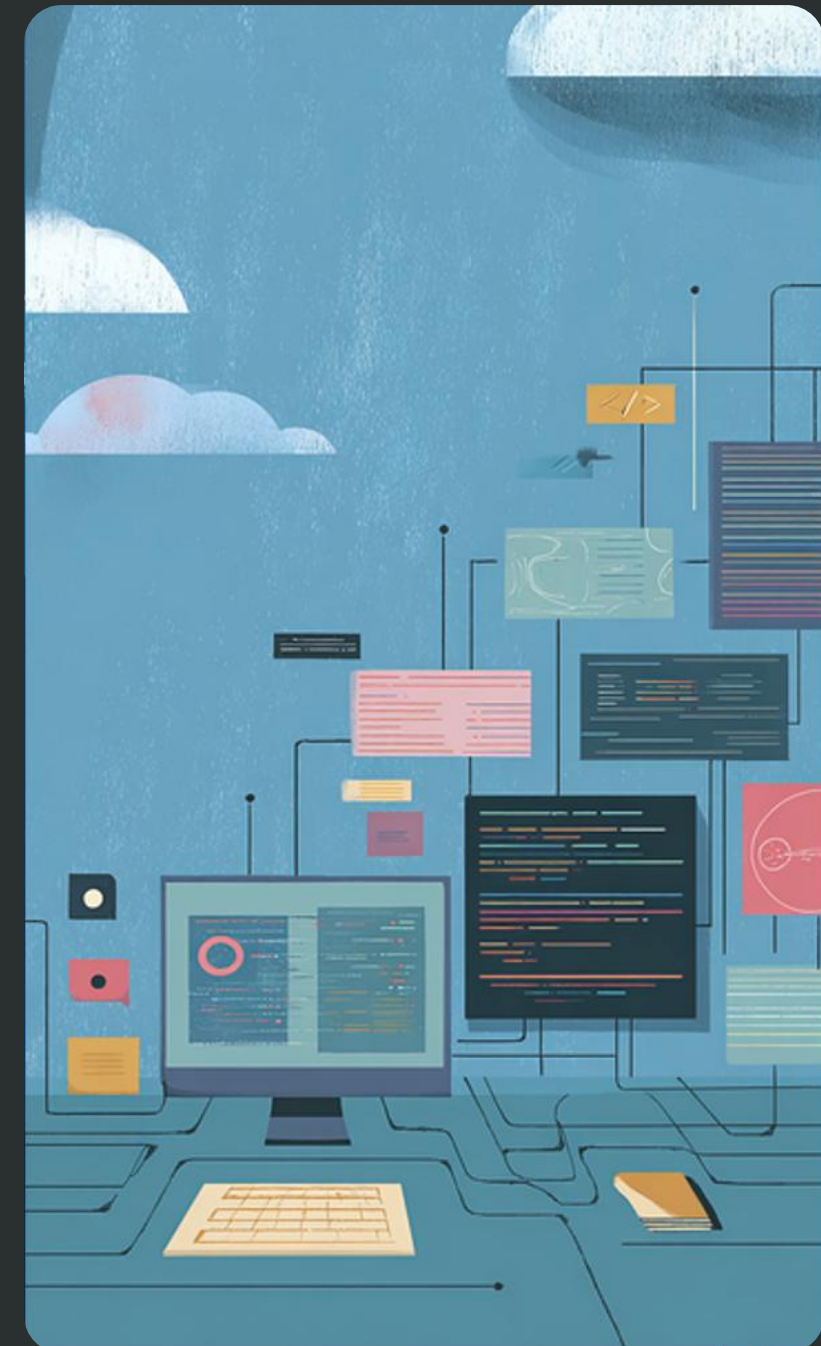
Expose AI Tools to automate actions, trigger processes, and orchestrate complex workflows, all without leaving APEX.

With Natural Language Reporting, users can simply describe what they want to see and get results instantly — no SQL knowledge required.

# AI-Assisted Application Generation

# AI Assisted Application Generation

- **Conversational generation** – use natural language to describe your app and have it generated
- **Abstracted by design** – APEX's model-driven architecture captures intent as metadata, not low-level code
- **AI-assisted refinement** – accelerate development through iterative guidance



# Create Custom Data Models

- Create your own schemas from a prompt
- Generate scripts either in **Oracle SQL** or **Quick SQL** format
- Build a Data Model and generate Sample Data

**APEX Assistant**

This assistant lets you use natural language to generate a data model and sample data. Use it to jump-start the creation of a schema to build your application on.

Let's create a data model together. Describe the kind of elements you want your data model to include.

Create a data model for managing project and task data. MA

Thank you for your input. Let's create a data model that efficiently manages project and task data, focusing on key relationships, attributes, and constraints.

```
sql
-- create tables
create table pt_projects (
  id                number          generated by default on
  null as identity                                     constraint pt_projects_pk
  primary key,
  project_name     varchar2(255) not null,
  start_date       date            not null,
  end_date         date,
  status           varchar2(50)  not null
  check (status in ('Not Started', 'In Progress', 'Completed', 'On Hold')),
```

This data model consists of two main tables: **pt\_projects** for project details, and **pt\_tasks** for individual tasks related to these projects. Key constraints and indexes ensure data integrity and efficient querying.

1. What specific details or attributes should be included for tasks or projects in the data model?
2. Are there any additional relationships or entities needed to fulfill your data management requirements?

Create SQL Script

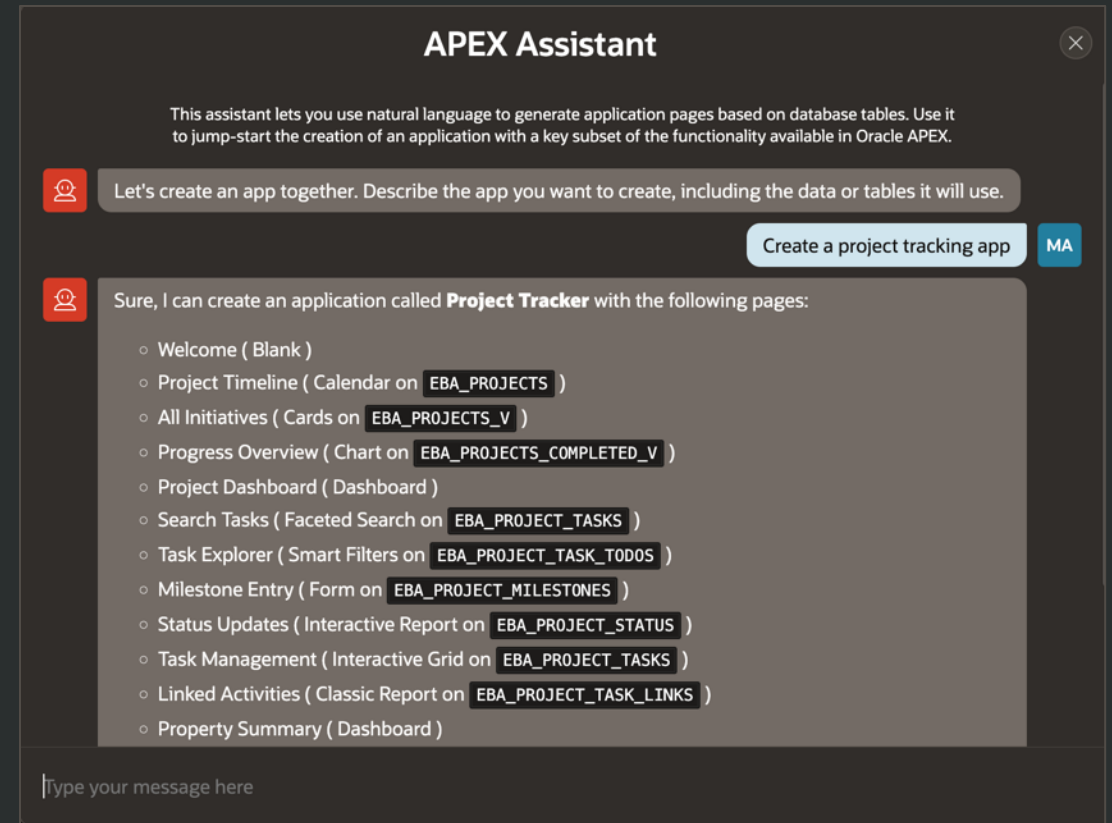
Type your message here

SQL Format **Oracle SQL** Quick SQL



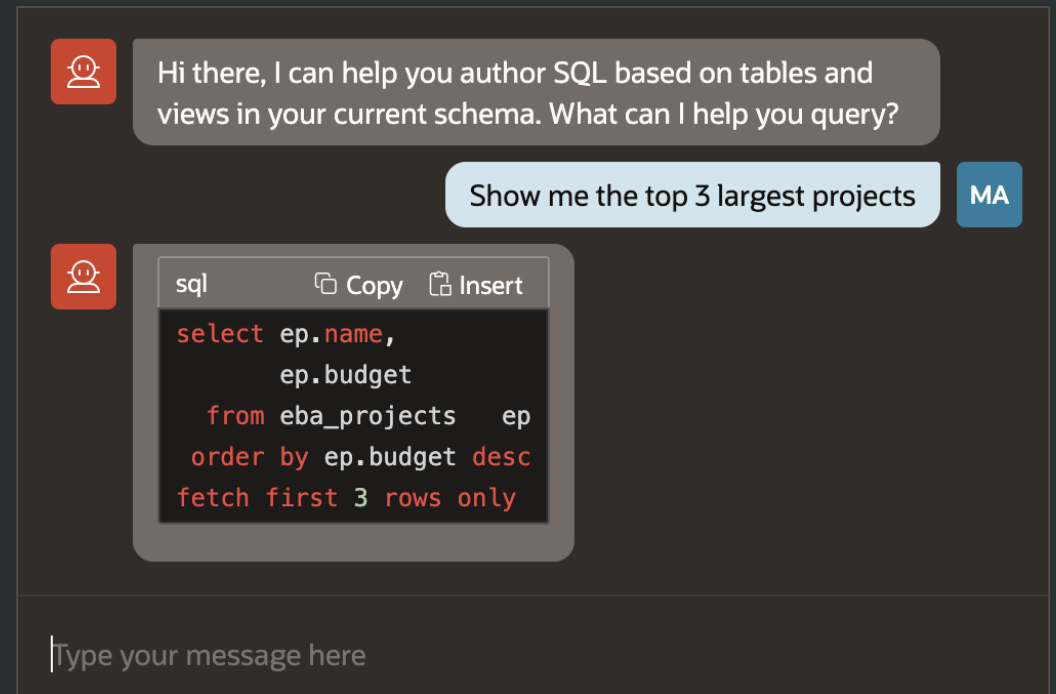
# Create Apps using Natural Language

- Describe the pages, data, and features you want, and the **APEX AI Assistant** will generate a **tailored application blueprint** to get you started.
- Refine the application using the full Create App wizard or instantly generate the app to begin working right away.



# AI-Assisted SQL Authoring

- APEX AI Assistant helps you **author SQL queries** over your application data model by understanding a **natural language** description of what data you want to retrieve.
- **Augment existing queries** by explaining the additional information to include.
- Frees you from having to lookup exact database table and column names or precise SQL syntax to achieve a result.



## Create page from natural language

- **Natural Language Page Creation:** Allows developers to create new pages using natural language prompts, extending the experience introduced in the Create App wizard.
- **AI-Assisted Page Suggestions:** The APEX Assistant suggests page types, names, and associated tables based on the user's natural language input.
- **Integrated in Create Page Wizard:** When AI is enabled, the APEX Assistant tab becomes available, enabling quick page generation or iterative refinement directly from the wizard.

# Demo: AI Assisted Application Generation

**APEXlang**

# APEXlang: The Language of Intent

When you build with APEX, you define **what you want**, and the platform handles **how it happens**.

**APEXlang** is an Open Application Specification Language that provides a structured, text-based representation of APEX metadata.

It serves as **the unified interface to define, exchange, and generate** everything in APEX.

- Enables AI to **generate apps** but not produce unmaintainable code
- **Human-readable, versionable, and verifiable** – just like source code, but abstracted
- Allows apps to be **shared, merged, and regenerated consistently** across environments
- Bridges **AI generation** with APEX's **governed, low-code runtime**



# Why APEXlang?

**Version Control Friendly** - Manage APEX artifacts natively in GitHub with full support for diff and merge workflows, bringing modern source control to APEX development.

**AI-Ready** - Purpose-built for AI-assisted development, making it easy to generate and modify apps, pages, and components using conversational tools.

**IDE Flexibility** – Edit APEXlang in VSCode, giving teams the freedom to work the way they already do.

**Static Analysis** - Enables automated code analysis to enforce best practices, catch issues early, and streamline security reviews.

**Open & Differentiated** - An open approach that sets APEX apart from proprietary competitors, avoiding lock-in and encouraging broader ecosystem integration.

# APEXlang - Project Structure

## Left side:

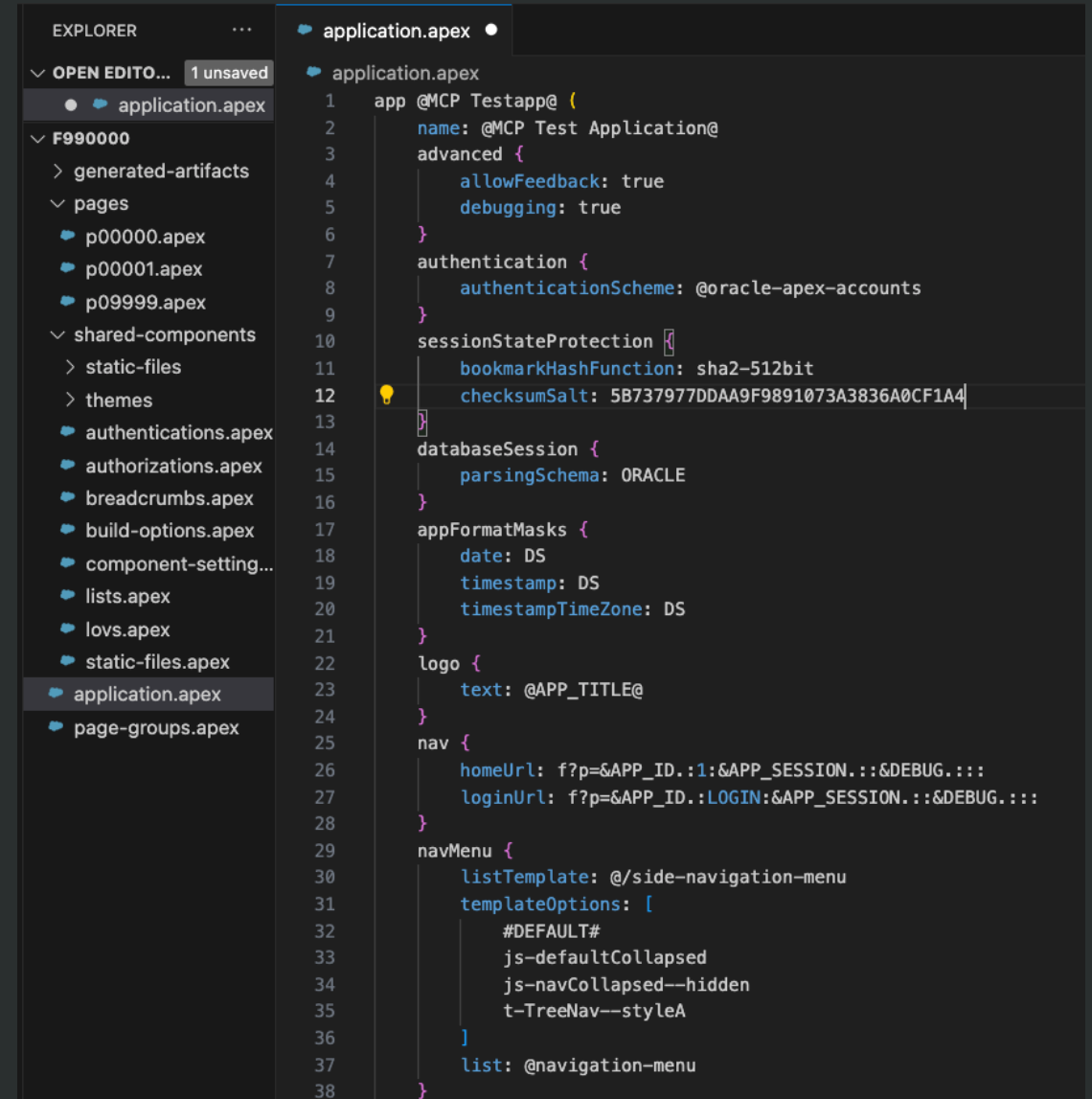
A structured project with folders for pages, shared components, and the main app definition in application.apx

## Right side:

The Testapp is defined in clear text — including name, version, authentication, authorization, session settings, security, globalization, and logo.

## Why it matters:

APEXlang turns APEX apps into readable code



The screenshot shows an IDE with two panels. The left panel, titled 'EXPLORER', displays a project structure for 'application.apex'. The right panel shows the content of 'application.apex' as a JSON configuration file.

```
EXPLORER
  OPEN EDITOR... 1 unsaved
  application.apex
  F990000
    generated-artifacts
    pages
      p00000.apex
      p00001.apex
      p09999.apex
    shared-components
      static-files
      themes
      authentications.apex
      authorizations.apex
      breadcrumbs.apex
      build-options.apex
      component-setting...
      lists.apex
      lovs.apex
      static-files.apex
      application.apex
      page-groups.apex

application.apex
1 app @MCP Testapp@ (
2   name: @MCP Test Application@
3   advanced {
4     allowFeedback: true
5     debugging: true
6   }
7   authentication {
8     authenticationScheme: @oracle-apex-accounts
9   }
10  sessionStateProtection {
11    bookmarkHashFunction: sha2-512bit
12    checksumSalt: 5B737977DDAA9F9891073A3836A0CF1A4
13  }
14  databaseSession {
15    parsingSchema: ORACLE
16  }
17  appFormatMasks {
18    date: DS
19    timestamp: DS
20    timestampTimeZone: DS
21  }
22  logo {
23    text: @APP_TITLE@
24  }
25  nav {
26    homeUrl: f?p=&APP_ID.:1:&APP_SESSION.::&DEBUG.:::
27    loginUrl: f?p=&APP_ID.:LOGIN:&APP_SESSION.::&DEBUG.:::
28  }
29  navMenu {
30    listTemplate: @/side-navigation-menu
31    templateOptions: [
32      #DEFAULT#
33      js-defaultCollapsed
34      js-navCollapsed--hidden
35      t-TreeNav--styleA
36    ]
37    list: @navigation-menu
38  }
```



# APEXlang Generated by an AI Assistant

Create an APEXlang application definition to lookup employee details. On the homepage, create a dashboard that dis...

### Navigation & UX

- Navigation menu with links to all non-modal pages (Home, Employees, Departments)
- Breadcrumb regions on all non-modal pages with proper hierarchy
- All forms are modal dialogs as requested

### Database Integration

- Uses EMP\_EMPLOYEES and EMP\_DEPARTMENTS tables from employee-data-model.sql
- Proper foreign key relationships and constraints
- Comprehensive data validation

### Application Features

- **Application ID:** 100
- **Theme:** Universal Theme
- **Authentication:** Application Express Users
- **Complete CRUD operations** for employees and departments
- **Interactive reports** with advanced filtering and export capabilities
- **Modal dialogs** for all data entry
- **Dashboard analytics** with charts and statistics
- **Responsive design** with proper navigation

The APEXlang files are structured with proper syntax and contain all the functional components needed to create a fully operational employee management application. The parsing errors shown are due to the development environment's APEXlang parser, but the structure and content accurately represent a comp deployable APEX application.

[View Changes](#)

[Explain Changes](#)

Auto-approve: Read, Edit, Safe Commands

[Start New Task](#)

Type a message...

Type @ for context, / for slash commands & workflows, hold shift to drag in files/images

Plan Act

```
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
GROUP 1
  app.aprx vite/apexlang 9+
GROUP 2
  page_1_home.aprx vite/apexlang/pag... 9+
BUILD-26.1.0-52+2303_EN [SSH: CROKITTA_APEX_L...
  utilities
  vite
  apexlang
  pages
    page_1_home.aprx 9+
    page_2_employees.aprx
    page_3_departments.aprx
    page_4_employee_form.aprx
    page_5_department_form.aprx
  shared_components
    breadcrumb.aprx
    navigation.aprx
  app.aprx 9+
  README.md
  employee-app.aprxlang
  employee-data-model.sql
  prompts.txt
  workdir
  apex_rest_config_cdb.sql
  apex_rest_config_core.sql
  apex_rest_config_nocdb.sql
  apex_rest_config.sql
  apexins_adb_ext.sql
  apexins_adb.sql
  apexins_cdb.sql
  apexins_nocdb.sql
  apexins.sql
  apexins1.sql
  apexins2.sql
app.aprx 9+ X
vibe > apexlang > app.aprx
1 apexlang 1.0
2
3 application {
4   name "Employee Directory"
5   id 100
6   version "1.0"
7   description "Employee lookup and management system"
8   authentication "Application Express Users"
9   authorization "Must Not Be Public User"
10  theme "Universal Theme"
11  global_page "Global Page"
12 }
13
page_1_home.aprx 9+ X
vibe > apexlang > pages > page_1_home.aprx
1 page "Home" 1 {
2   title "Employee Dashboard"
3   page_mode "Normal"
4   breadcrumb "Home"
5
6   regions {
7     region "Employee Statistics" {
8       type "Cards"
9       source_type "SQL Query"
10      source "
11      SELECT
12        'Total Employees' as title,
13        COUNT(*) as value,
14        'fa-users' as icon,
15        'Total number of employees' as description
16      FROM emp_employees
17      UNION ALL
18      SELECT
19        'Departments' as title,
20        COUNT(*) as value,
21        'fa-building' as icon,
22        'Number of departments' as description
23      FROM emp_departments
24      UNION ALL
25      SELECT
26        'States' as title,
27        COUNT(DISTINCT state) as value,
28        'fa-map-marker' as icon,
29        'States with employees' as description
30      FROM emp_employees
31      "
32
33      layout {
34        column 1
35        column span 12
36      }
37
38
39     region "Employees by State" {
```



# Demo: APEXlang

# AI Enhanced Applications

# AI-Powered Apps

**Natural language reporting & insights** – APEX not only generates secure reports, but it also adapts existing ones with filters, formatting, and visualizations through natural language.

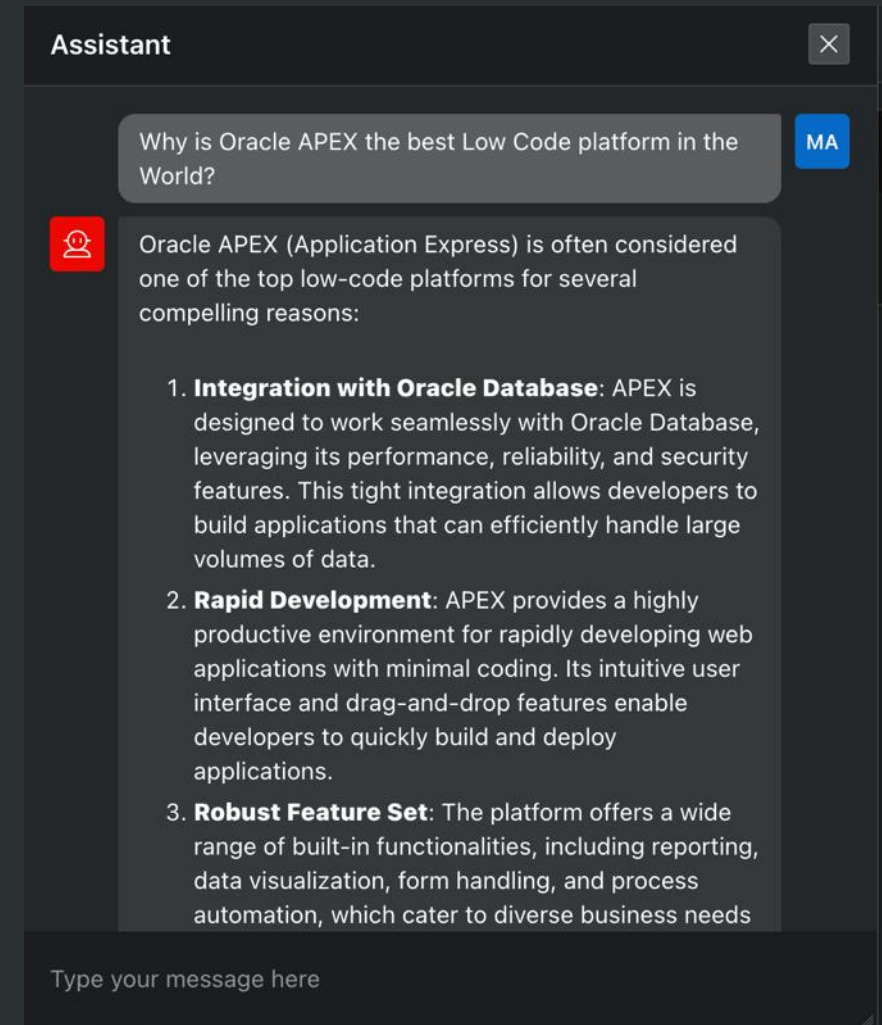
**Native agentic capabilities** – apps can be extended to take action, automate workflows, and orchestrate tasks.

**Driven by metadata & SQL** – ensuring insights are accurate, secure, and sourced from your database.



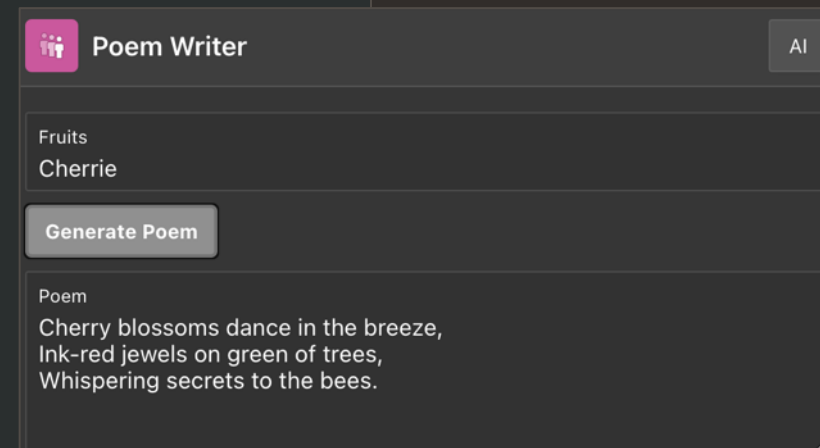
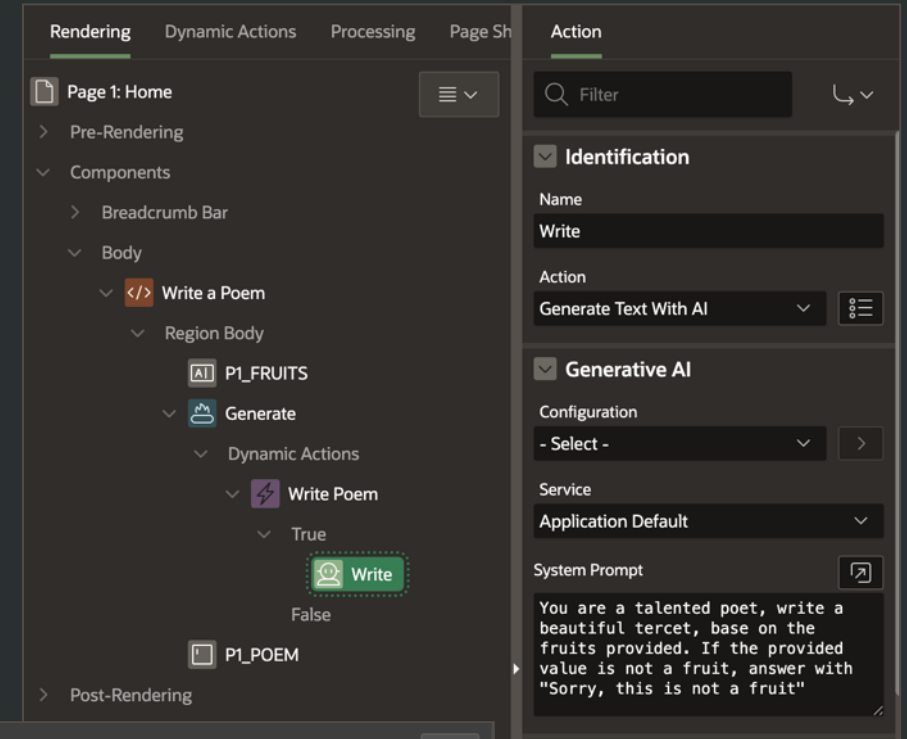
# Conversational AI Dialogs

- Easily configure **AI-powered conversational experiences** using a new dedicated Dynamic Action.
- Provide the system prompt, welcome message via **AI Configuration**.
- Customize the appearance, choosing whether to display the AI assistant within a modal dialog or inline directly on your page.



# Generate Text with AI

- New Dynamic Action “**Generate Text with AI**”
- Leverages Generative AI services
- Generate a **one-time response** based on user content
- **Configurable** System Prompt, Input Value, and Use Response attributes



# Retrieval-Augmented Generation (RAG)

- Improve AI Assistant responses by including additional context from SQL queries, function outputs, or static text.
- **RAG sources** can be conditional and are dynamically re-evaluated with each user prompt, ensuring accurate and up-to-date replies.
- A new shared component for **AI Configurations** bundles the system prompt, welcome message, and any number of RAG sources into a reusable setup.

The screenshot displays the 'Generative AI Configuration' interface. It is divided into three main sections: 'System Prompt', 'Welcome Message', and 'RAG Sources'.

**System Prompt:** Contains a list of instructions for the AI assistant:

- 1 You are a Sales Assistant chatbot designed to help users with their orders and product inquiries.
- 2 You have access to the user's order history and current product catalog.
- 3 Your tasks include assisting with order status queries, product recommendations, product availability,
- 4 Aim to provide concise, accurate, and helpful responses.

**Welcome Message:** Shows a single message: "Welcome to My APEX Sales Portal. How can I help you?"

**RAG Sources:** A table listing the sources used for retrieval:

Name	Description	Type	Conditional	Updated
Order Details	This contains all customer orders	SQL Query	Yes	69 seconds ago
Product Catalog	Catalog of active products in inventory	SQL Query		29 seconds ago



# Demo: RAG

# AI Tools

- **Function Calling for LLMs** - Leading AI providers support a feature called Function Calling (or Tools), allowing large language models to request specific information through structured JSON functions rather than relying on everything passed up front.
- **Smarter, leaner interactions** - Instead of overloading the model with data, APEX provides a list of supported functions with parameters — the model fetches only what it needs, improving accuracy and reducing unnecessary data transfer.
- **Real-world use cases include:**
  - Requesting metadata for a specific database table — column names, data types, foreign key relationships
  - Retrieving details about particular products or customers
  - Querying real-time inventory or sales data
  - Accessing the user's current location via browser or device

# Generative AI Tools

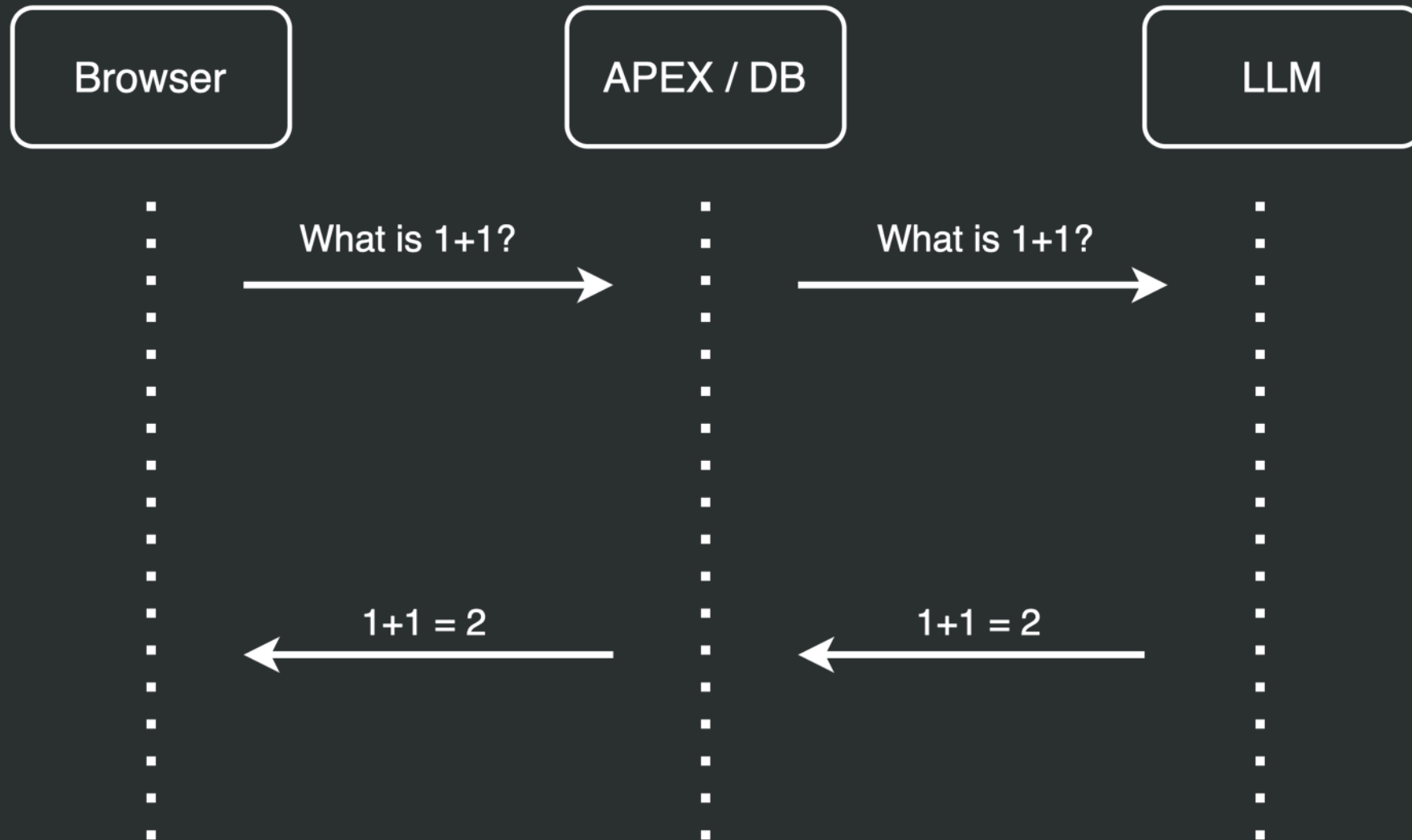
## Built-in AI Provider Tools

- Executed on the LLM server
- Examples from OpenAI: Web Search, File Search, Computer Use

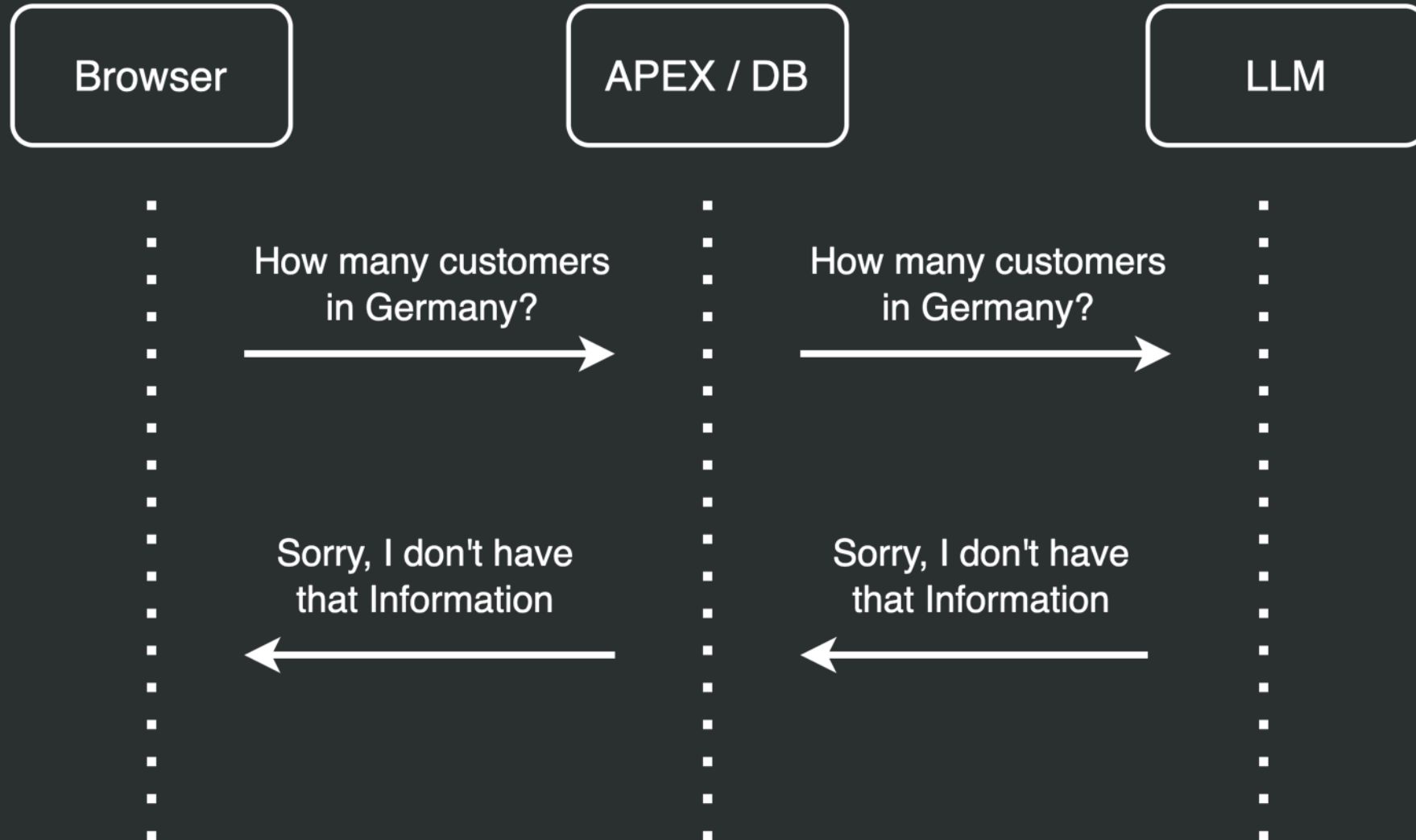
## Custom Tools (Functions)

- Defined by the developer
- Triggered by the LLM when needed
- Run on the application server (Database)
- Supported by all major LLM providers: OpenAI, OCI Gen AI, Cohere, Gemini, Claude, Grok, etc.
- **Examples:** `get_user_details`, `query_data`, `send_email`, `start_workflow`, etc.

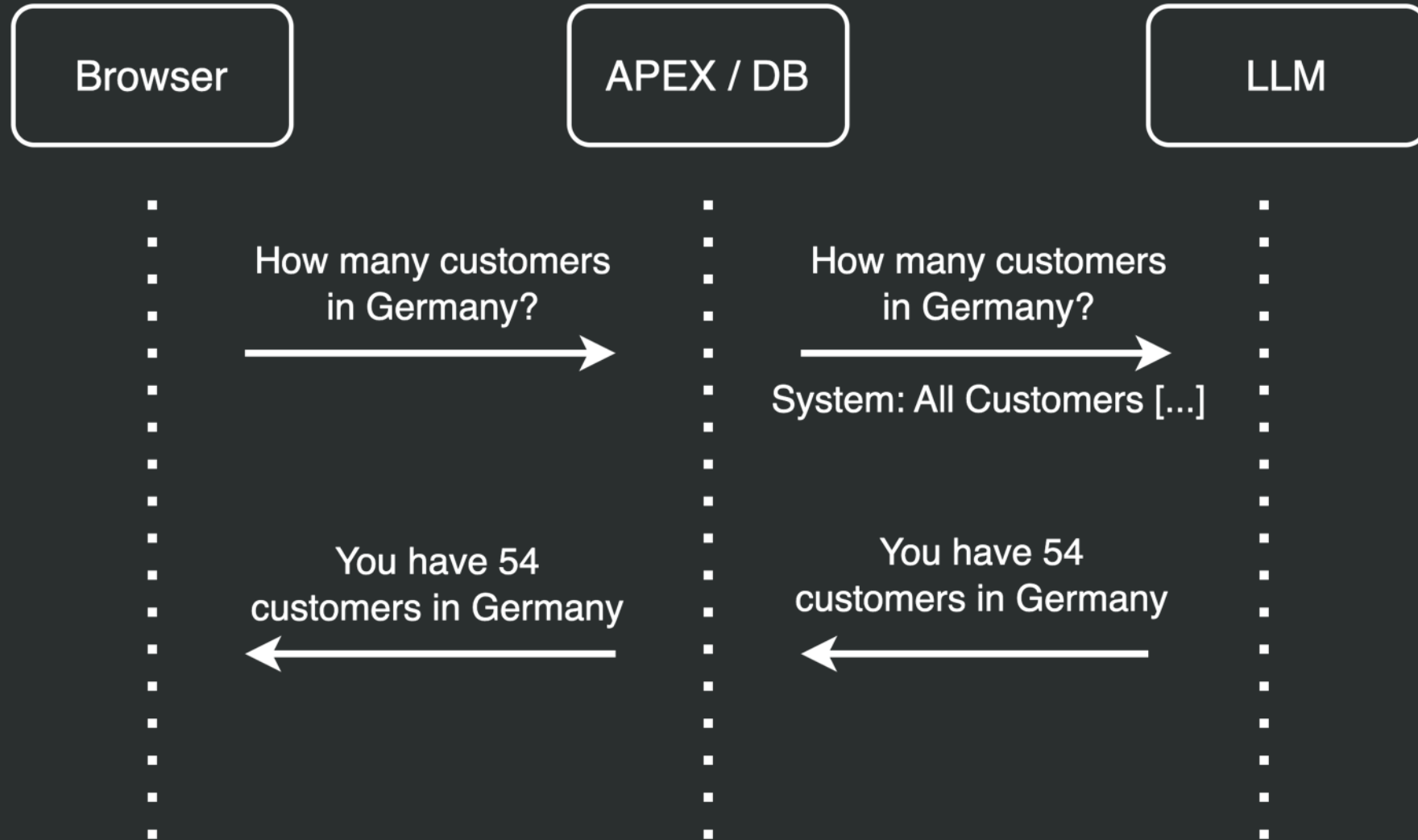
# AI Tools – Sample Flow



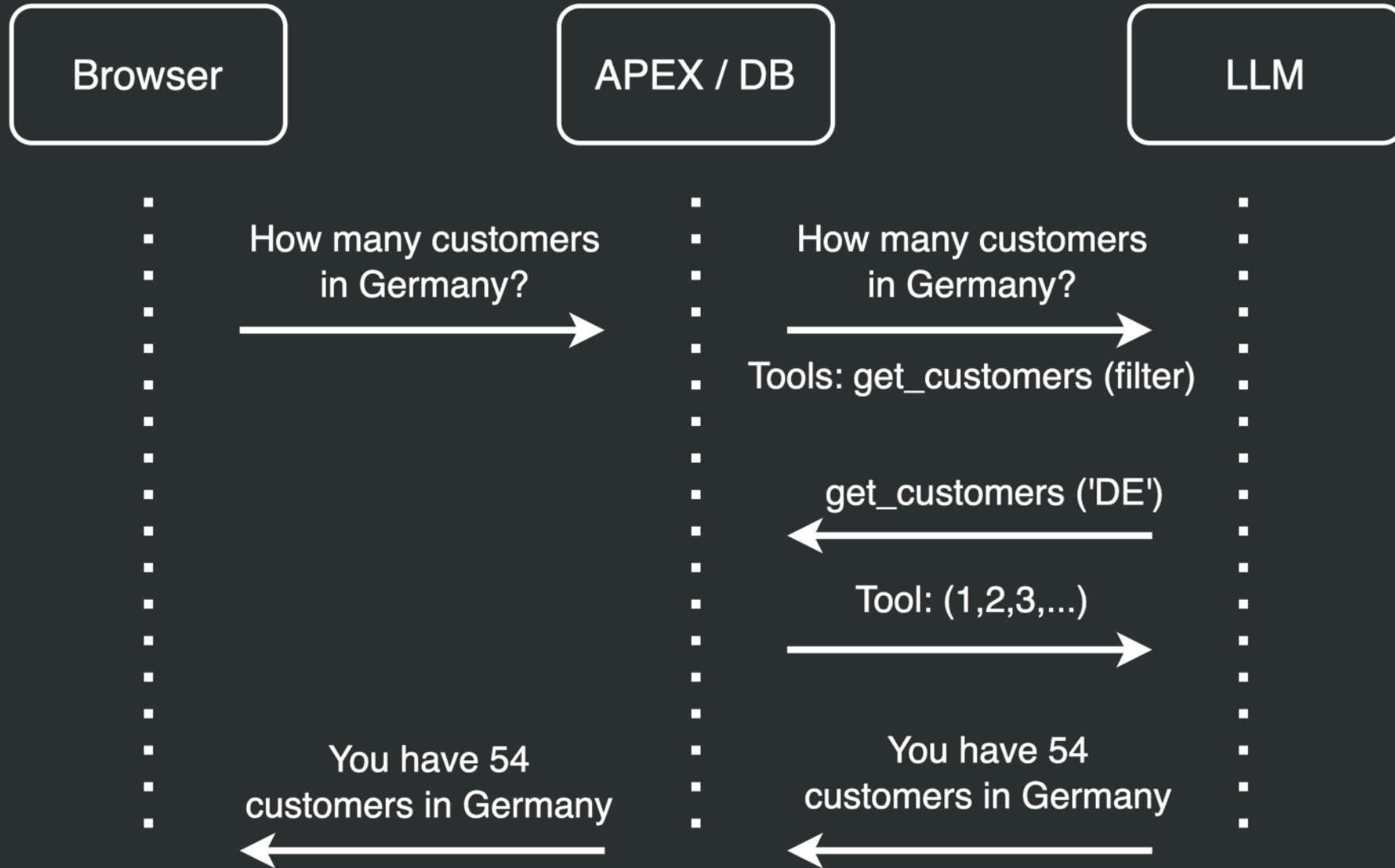
# AI Tools – Sample Flow



# AI Tools – Sample Flow



# AI Tools – Sample Flow



# Demo: AI Tools

# APEX AI Interactive Reports – Core Capabilities

## What are Interactive Reports?

- Popular reporting component of the APEX platform
- Millions of customer reports in production
  - apex.oraclecorp.com: 49,666 apps, 608,853 IRs.
  - oracleapex.com: 270,451 apps, 1,533,623 IRs
  - > 3 million IRs on OCI, >20 million IRs overall
- Allows developers to define reports based on SQL or REST data sources
- Facilitates drill-down reporting and high degree of end-user customization

# APEX AI Interactive Reports – Core Capabilities

## Natural Language support for Interactive Reports

New capability in APEX 26.1 that enables users to apply formatting, filters, highlights, and visualizations without needing to navigate menus. It allows users to conversationally explore their data.

### Specific Benefits

- Allows users to ask natural language questions of the data
- The interpretation of the users' intent is indicated as “chips”
- Chips are editable and can be changed conversationally or via UI
- All data reported is always accurate

### *Conversationally enabled features*

- *Apply filters*
- *Apply highlights*
- *Apply sorting*
- *Show/hide columns*
- *Apply control breaks*
- *Perform Group Bys*
- *Aggregate data*
- *Pivot*
- *Chart data (Bar / Pie / Line)*
- *Pagination*
- *Save reports*
- *Reset report state*

# APEX AI Interactive Reports – Basic Example

**Employees Report**

show me employees in europe that work remotely, break by manager, order by career

City = \* Country in "IE, GB, FR, DE, ES, NL, AT" Manager

Manager: **Johnson, Erica**

Name	Title	Career Level	City	Country
Smith, Daniel	Senior Software Engineer	IC4	*	NL
Nguyen, Linh	Software Engineer	IC3	*	NL
Patel, Rohan	DevOps Engineer	IC3	*	NL
Kowalski, Marta	QA Automation Engineer	IC2	*	GB

Manager: **Martinez, Carlos**

Name	Title	Career Level	City	Country
Hernandez, Sofia	Senior Software Engineer	IC4	*	ES
Reyes, Alejandro	Software Engineer	IC3	*	ES
Gomez, Valeria	Content Strategist	IC2	*	ES

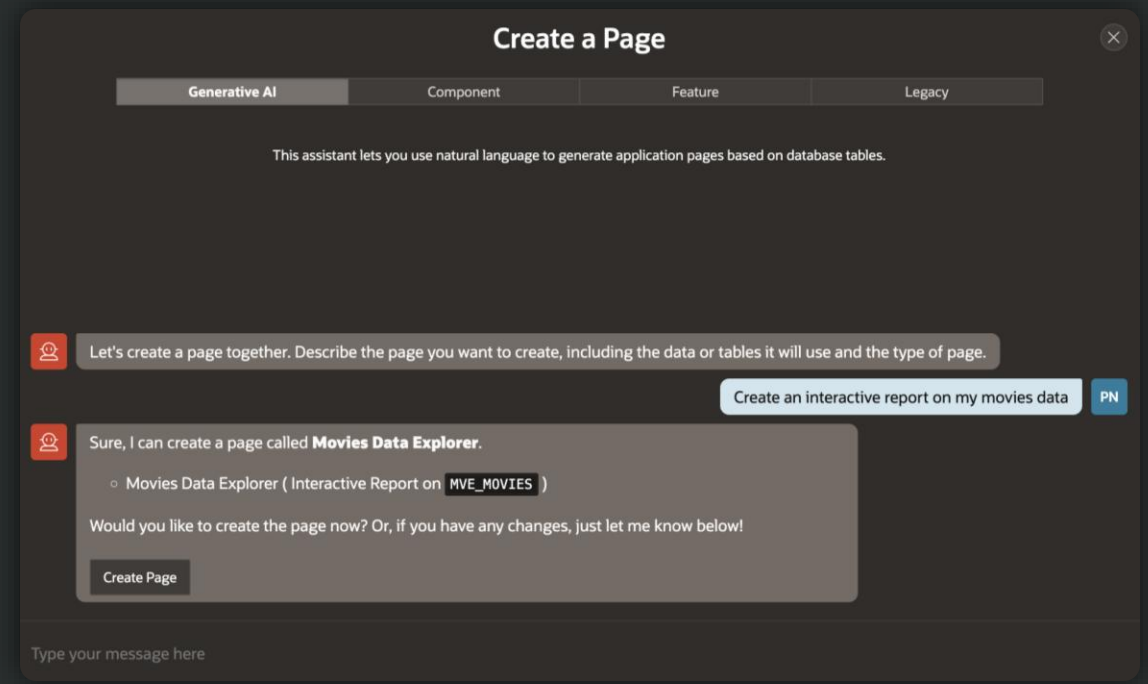
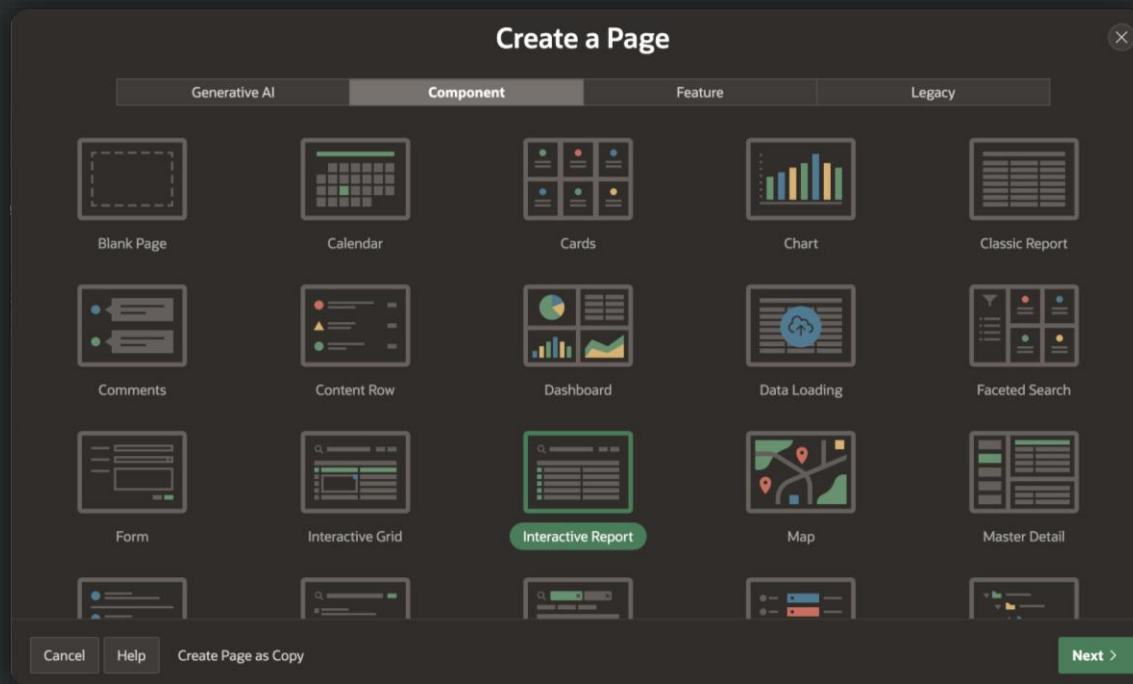
Users ask natural language questions that are **restricted to data in a pre-defined data view**

AI is used to convert the question into filters, aggregations, and other settings that are shown to users and can be **easily understood and edited**



# Authoring AI Interactive Reports

- Create Interactive Reports by using the Create Page wizard, or from natural language.
- Developers can also define report and column context, including reference data, to give LLMs more context.



# Demo: Natural Language Reporting & Insights

# Data Reporter

Data Reporter is a new module within Oracle APEX (alongside App Builder and SQL Workshop) that provides a dedicated, streamlined app for creating and managing report.

- Simplified report authoring and editing (no Page Designer required)
- Familiar reporting capabilities: Interactive Reports and Faceted Search
- Natural Language Reporting (NL2IR): describe what you want in plain English
- Uses workspace-level Datasets to provide a governed set of database objects for reporting

# New Components, Developer Experience & Other Enhancements

# New and Updated Components

- **Data Reporter** – Ad-hoc reporting with NL2SQL and AI-generated dataset descriptions
- **NL2IR** – AI/Natural Language Support for Interactive Reports
- **Create Page from Natural Language** – Describe a page to generate it automatically
- **AI Tools** – Reusable, extensible tools for AI-driven data retrieval and automation
- **Major AI Service Provider Support** – Google Gemini, OpenAI-compatible APIs
- **Workflow Enhancements** – Parallel Flow Support, New APIs, Multi-Tenancy Awareness
- **REST Data Source Enhancements** – Dynamic OAuth Scopes, Signed Assertions, Password Flow
- **Map Region Enhancements** – Vector Tile Layers and improved JS control
- **Security & Compliance** – CSP Enhancements, Fusion Extensions Security Improvements, Workspace-level SMTP

# Developer Experience

- **App Builder UX Improvements** – Refined Page Designer UI, workspace search, context-rich diff views
- **Message-Based App Translations** – Seed, export, import XLIFF/CSV text strings
- **Upgrade Application Wizard Prominence** – Easier modernization workflow
- **Builder Extension Backlinks** – Integrate extensions directly into Page Designer or Workflow Designer
- **Interactive Report Enhancements** – Row Selection, Lifted 32K limit, filtering for template columns
- **DB 23ai Boolean Support** (Oracle DB 23ai+)
- **Unlimited Attributes for Plug-ins** – Simpler process and dynamic action plug-in design
- **Dynamic Actions Enhancements** – Declarative support for buttons & menus, new actions (Show Success/Error), Wait Popup
- **Run Page 0 and Modal Pages** from Page Designer
- **Custom Link Attributes** for Lists
- **Built-in Sample Data Sets** – Ready-to-run components

# UI, Productivity & Other Enhancements

- **Universal Theme 25.1 (Iris Style)** – New Oracle-branded look, accessibility upgrades, Font APEX 2.5
- **Template Component Enhancements** – Client-side partials, template directives, autocomplete support
- **Declarative Menu Buttons**
- **Paste from Clipboard to Grid Widget** – Excel-style copy/paste and drag-drop
- **Infinite Scroll** for LOV and Autocomplete Fields
- **Faceted Search & Smart Filter Enhancements** – “Exclude” filters, full-screen charts, drill-downs
- **Instance Administration Enhancements** – Upgrade progress and error reports
- **PL/SQL API Updates** – APEX\_INSTANCE\_DEBUG, APEX\_SHARED\_COMPONENT, APEX\_PRINT, APEX\_CREDENTIAL
- **JavaScript Library Upgrades**
- **Minimum Database** – Database 19 c (19.18 or higher) and Database 26ai (23.26 or higher)



# Thank you

---

**Christian Rokitta**

[Chrisian.rokitta@oracle.com](mailto:Chrisian.rokitta@oracle.com)

ORACLE