Exploring SELECT AI Agents

with a Data Warehouse Operations Example.

Andrej Pashchenko

Copyright © 2026 Accenture. All rights reserved.

accenture

2021

IT consulting
Oracle DB, Data Warehousing, Performance, Knowledge Sharing

1990s

Naval engineering studies

trivadis
makes IT easier.

2009

IT consulting
Oracle DB, Data Warehousing, Performance, Knowledge Sharing

1995

Governmental bodies
Database applications, MS Foxpro, Oracle

meinestadt.de

2002

Internet portal
Database applications, Oracle DB, Oracle Forms, APEX



Oracle ACE Pro

blog.sqlora.com

linktr.ee/andrejpashchenko

SELECT AI

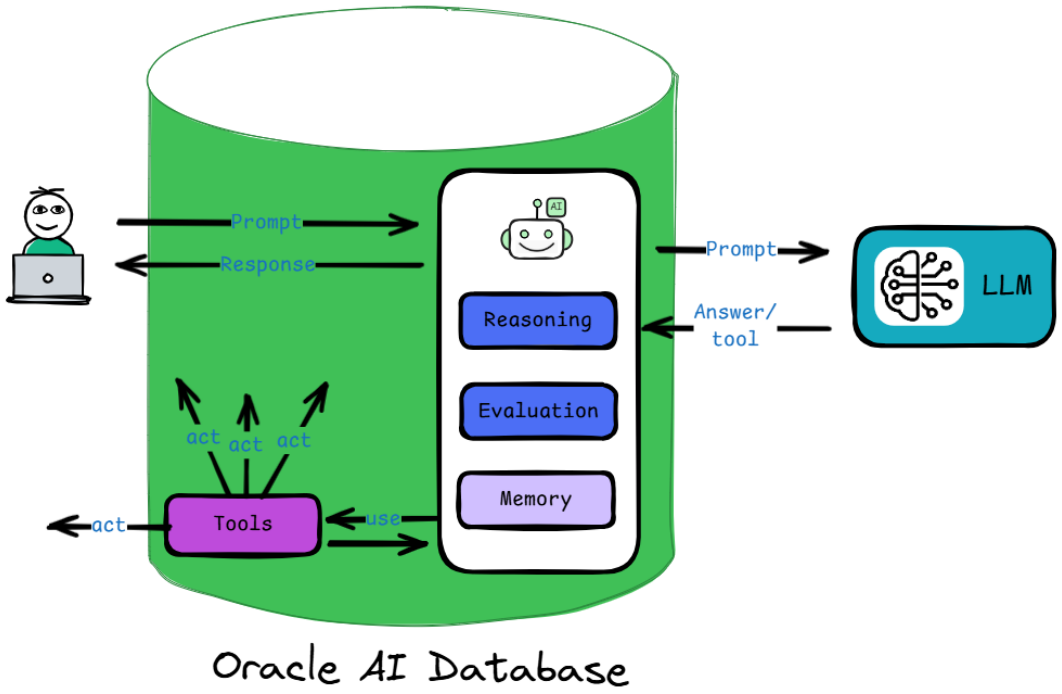
AI-assisted interaction layer for data in the database

Capability	Autonomous AI Database		Oracle AI Database	
	26ai	19c	23.26.1	19.30
Chat	✅	✅	✅	✅
NL2SQL	✅	✅	✅	✅
- Feedback	✅		✅	
- Auto Object Selection	✅		✅	
RAG	✅		✅	
SDG	✅	✅	✅	✅
AI agents	✅	✅	✅	✅
Summarization	✅	✅	✅	✅
Translation	✅	✅	✅	✅

Source: Select AI and Select AI Agent Capability Matrix

Copyright © 2026 Accenture. All rights reserved.

Oracle Select AI Agent Framework

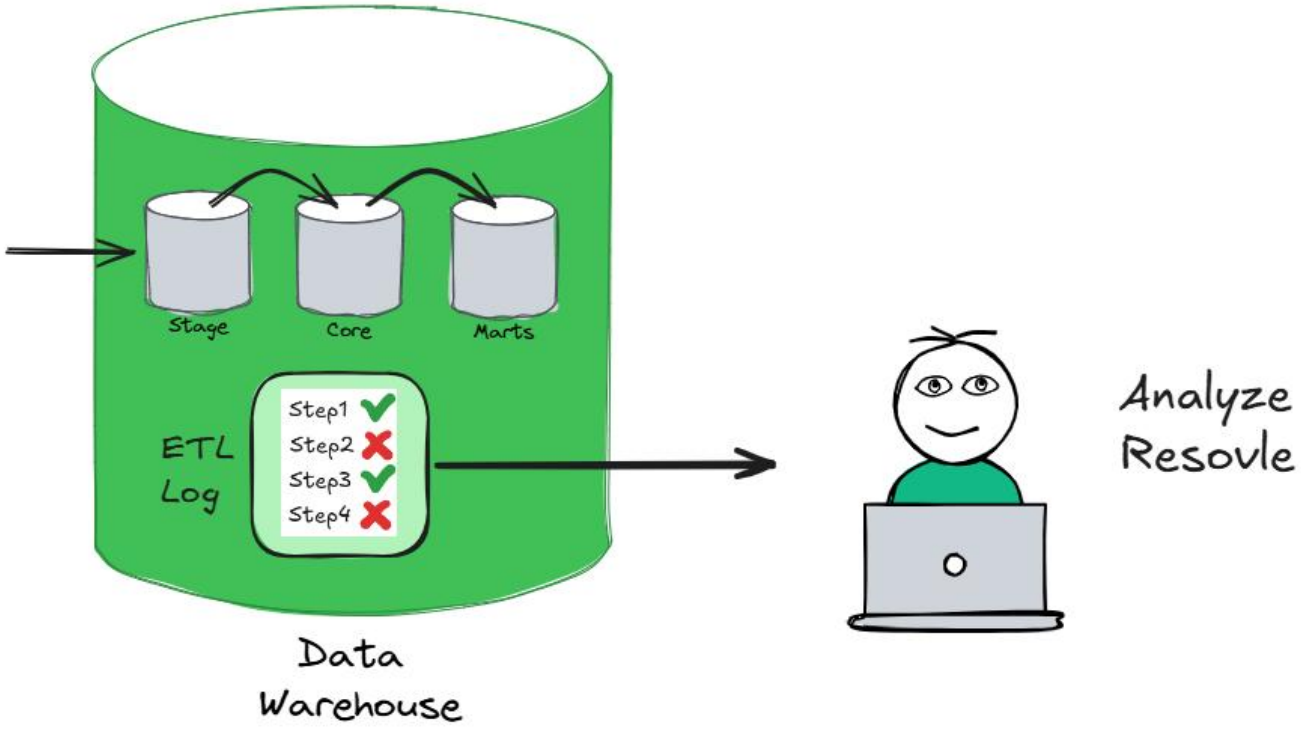


DBMS_CLOUD_AI_AGENT

- Build and run AI agents directly inside the database
- Based on a built-in ReAct-style execution pattern
- Flexible tool integration (PL/SQL, REST, cloud functions, pre-built tools)
- Stateful agents with memory
- Conversation-driven interaction (multi-turn)
- Runs natively in Oracle AI Database

Copyright © 2026 Accenture. All rights reserved.

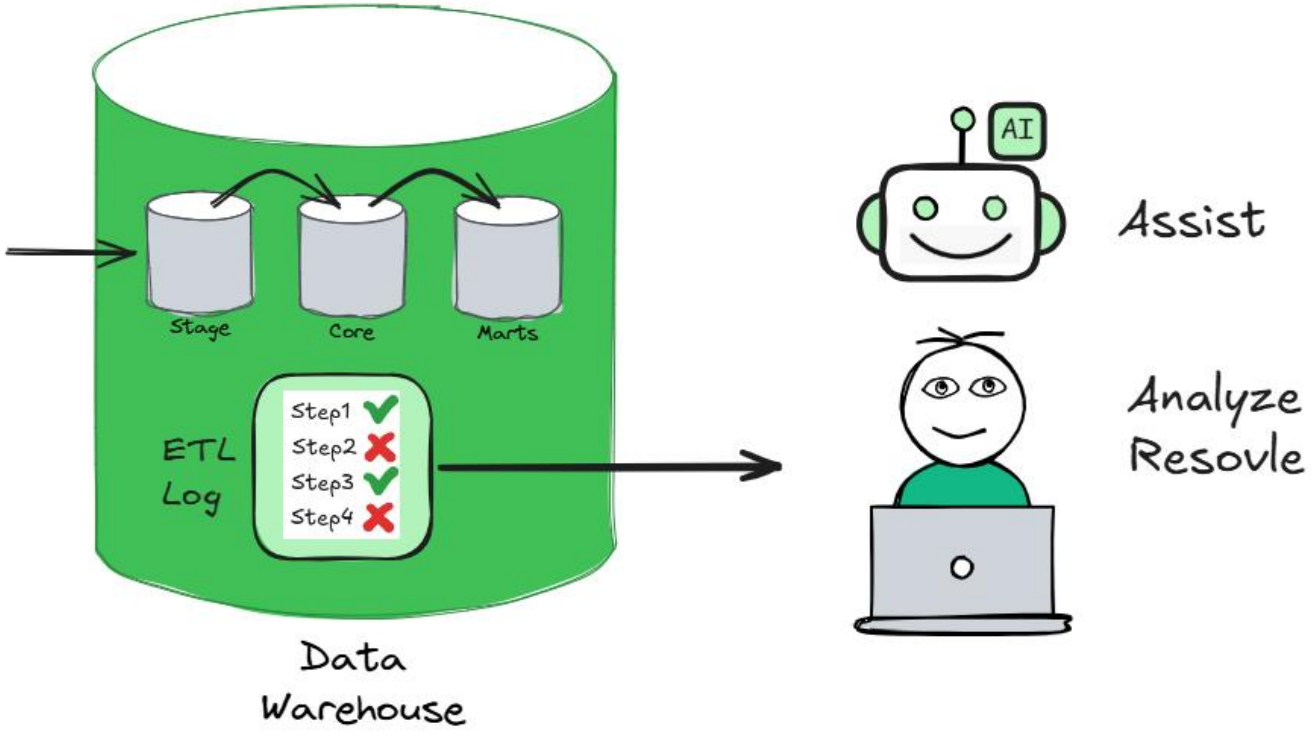
Use Case: ETL Operations



- ETL jobs run regularly and generate operational logs
- Errors and failures must be analyzed and resolved quickly
- Many issues are repetitive, but still require manual investigation
- Existing knowledge (previous fixes) is often not systematically reused
- Resolution typically involves DB operations (e.g. restart jobs, notify teams)

Copyright © 2026 Accenture. All rights reserved.

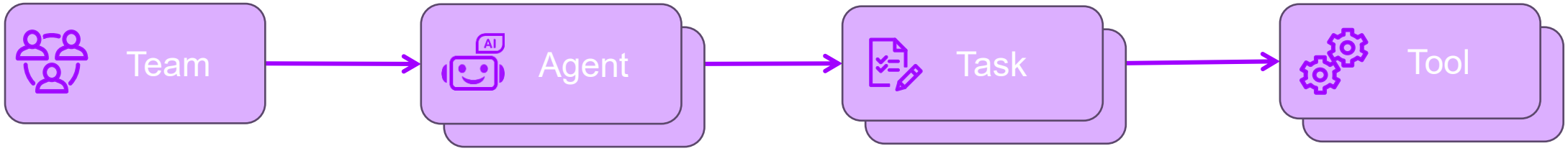
ETL Operations With Agentic Support



- Assist, not replace, operations teams
- Reduce repetitive manual analysis of known issues
- Suggest actions based on historical knowledge (RAG)
- Provide decision support for unknown problems
- Enable controlled automation with optional human approval
- Execute standard actions directly in the database when appropriate
- Continuously improve by learning from resolved incidents

Copyright © 2026 Accenture. All rights reserved.

DBMS_CLOUD_AI_AGENT



- Defines the execution workflow
- Orchestrates agent–task steps
- Controls execution order (e.g. sequential)

- Performs reasoning
- Associated with an AI profile
- Defines role and behavior
- Executes assigned tasks
- May invoke tools available to the active task

- Defines a unit of work
- Contains task-specific instructions
- Specifies which tools are available for this task
- Performed by an agent within a team context

- Registered callable capability
- Defined using structured attributes
- Built-in or custom tools
- Invoked by an agent during task execution

Copyright © 2026 Accenture. All rights reserved.

Prerequisites: Credentials And AI Profiles

- Each agent uses a reasoning model via **SELECT AI profile**
  AI profile assignment can also be changed dynamically at runtime.
- Within a team different agents can use different AI profiles
- Supported providers: OpenAI, Cohere, Azure OpenAI Service, OCI Generative AI, Google, Anthropic, Hugging Face, AWS, OpenAI-compatible providers / Private LLMs,
  e.g. self-hosted with Oracle Private AI Services Container
- Use DBMS_CLOUD_AI to setup the AI profiles
- Some tools also require AI-Profiles (see next slides)

	Name	Vendor	OCID	Version	Endpoint	Capability
◉	cohere.command-a-03-2025	cohere	...mhvggyb4ja	1.0	—	Chat
○	cohere.command-a-vision	cohere	...oqdbnsv6tq	1.0	—	Chat
○	cohere.command-latest	cohere	...xpov43ewga	1.7	—	Chat
○	cohere.command-plus-latest	cohere	...ohicrnx6eq	1.6	—	Chat
○	cohere.command-r-08-2024	cohere	...737hkjxryq	2.0	—	Chat
○	cohere.command-r-08-2024	cohere	...dcdhdu2whq	1.7	—	Chat
○	cohere.command-r-plus-08-2024	cohere	...cgzi4wlcra	2.0	—	Chat
○	cohere.command-r-plus-08-2024	cohere	...zzfx5cgkmq	1.6	—	Chat
○	google.gemini-2.5-flash	google	...ygode5o5xq	1.0.0	—	Chat
○	google.gemini-2.5-flash-lite	google	...pqnohxknaq	1.0.0	—	Chat
○	google.gemini-2.5-pro	google	...mebjgtvhra	1.0.0	—	Chat
○	meta.llama-3.3-70b-instruct	meta	...4hfu3mx6bq	1.0.0	—	Chat
○	openai.gpt-oss-120b	openai	...kwstriguvq	1.0.0	—	Chat
○	openai.gpt-oss-20b	openai	...skvftn4mna	1.0.0	—	Chat

Chat models available in eu-frankfurt-1 as of March 2026

Copyright © 2026 Accenture. All rights reserved.

Tools – What an Agent Can Invoke

Built-in tools



- Translates natural language (coming from agent's reasoning model) to SQL



- Enable agents to ground responses in enterprise knowledge base



- To search information on the web



- To send notification via email or Slack

Custom tools



- Custom capabilities using specified PL/SQL procedures or functions

- All tools can be specified using the parameter `ATTRIBUTES` – a JSON document with the attributes specific to each tool type

```
DBMS_CLOUD_AI_AGENT.CREATE_TOOL(
     tool_name           IN  VARCHAR2,
     attributes          IN  CLOB,
     status              IN  VARCHAR2  DEFAULT NULL,
     description         IN  CLOB      DEFAULT NULL );

-- Also other lifecycle steps
DBMS_CLOUD_AI_AGENT.DROP_TOOL
DBMS_CLOUD_AI_AGENT.ENABLE_TOOL
DBMS_CLOUD_AI_AGENT.DISABLE_TOOL
```

Copyright © 2026 Accenture. All rights reserved.

Tools – SQL Built-in Tool



- Enables SQL execution during agent task processing
- First, create an AI profile to be used
- Tool_Type: "SQL"
- Mandatory parameter: **profile_name**

```
BEGIN
  DBMS_CLOUD_AI.CREATE_PROFILE(
      profile_name => 'MY_NL2SQL',
      attributes   => '{"provider": "oci",
        "credential_name": "MY_CRED",
        "oci_compartment_id": "ocid1.comp... ",
        "object_list": [
             {"owner": "ETL_OPS", "name": "ETL_LOG"},
             {"owner": "ETL_OPS", "name": "ETL_JOB"}
               ]
       }');

  DBMS_CLOUD_AI_AGENT.CREATE_TOOL(
      tool_name  => 'SQL_TOOL',
      attributes => '{
           "tool_type": "SQL",
           "tool_params": {
                "profile_name": "MY_NL2SQL"
                          }
                     }');
END;
/
```

Copyright © 2026 Accenture. All rights reserved.

Tools – RAG Built-in Tool



- Performs **R**etrieval-**A**ugmented **G**eneration
- Automatically ingests documents from Object Storage, generates embeddings, and updates a vector index asynchronously
- First, create an AI profile to be used
- Tool_Type: "RAG"
- Mandatory parameter: **`profile_name`**

```
BEGIN
  DBMS_CLOUD_AI.CREATE_PROFILE(
      profile_name =>'MY_RAG_PROFILE',
      attributes   =>'{"provider": "oci",
         "credential_name": "GENAI_CRED",
         "vector_index_name": "MY_RAG_INDEX",
         "oci_compartment_id": "ocid1.compar...",
         "temperature": 0.2,
         "max_tokens": 3000
          }');
  DBMS_CLOUD_AI.CREATE_VECTOR_INDEX(
      index_name  => 'MY_RAG_INDEX',
      attributes  => '{
           "vector_db_provider": "oracle",
           "location": "https://swiftobj...",
           "object_storage_credential_name": "OCI_CRED",
           "profile_name": "MY_RAG_PROFILE",
           "chunk_overlap":128, "chunk_size":1024 }');
  DBMS_CLOUD_AI_AGENT.CREATE_TOOL(
      tool_name  => 'RAG_TOOL',
      attributes => '{ "tool_type": "RAG",
           "tool_params": {"profile_name": "MY_RAG_PROFILE"} }');
END;
```

Copyright © 2026 Accenture. All rights reserved.

Tools – Websearch Built-in Tool



- Retrieves external web information
- Can enrich reasoning with external data
- OpenAI is the only Websearch AI provider supported.
- Setup OpenAI credentials and ACL
- Tool_Type: "WEBSEARCH"
- Mandatory parameter: **`credential_name`**

```
BEGIN
  DBMS_CLOUD.CREATE_CREDENTIAL(
    credential_name => 'OPENAI_CRED',
    username        => 'OPENAI',
    password        => '<OPENAI_API_KEY>'
  );

  DBMS_CLOUD_AI_AGENT.create_tool(
    tool_name  => 'WEBSEARCH_TOOL',
    attributes => '{
       "instruction": "This tool can be used for searching the
                       details about Oracle error messages
                       and ways to fix them",
       "tool_type": "WEBSEARCH",
       "tool_params": {"credential_name": "OPENAI_CRED"}}'
   );
END;
/
```

Copyright © 2026 Accenture. All rights reserved.

Tools – Notification Built-in Tool



- Sends notifications during workflow execution
- Supports EMAIL and SLACK
- Tool_Type: "NOTIFICATION"
- Mandatory parameter: **`notification_type, credential_name`** and some type-specific

```
BEGIN
  DBMS_CLOUD.CREATE_CREDENTIAL(
     credential_name => 'EMAIL_CRED',
     username        => 'ocid1.user.oc1... ',
     password        => '<PASSWORD>');

  DBMS_CLOUD_AI_AGENT.CREATE_TOOL(
      tool_name  => 'EMAIL_TOOL',
      attributes => '{
          "tool_type": "NOTIFICATION",
          "tool_params": {
              "notification_type": "EMAIL",
              "credential_name": "EMAIL_CRED",
              "to": "etl_opsr@etl-ops.com",
              "smtp_host": "smtp.email.eu-frankfurt-1.oci..."
                           }
                  }');
END;
/
```

Copyright © 2026 Accenture. All rights reserved.

Tools – Custom Tools



- Extends agent capabilities with user-defined logic
- Can call PL/SQL functions or procedures
- If you **don't** specify one of the built-in tool types, then it will be a **custom** tool
- Mandatory parameter:
  - **`instruction -`** clear, concise statement, what the tool should accomplish and how
  - **`function -`** PL/SQL function or procedure name, can be a packaged one

```
BEGIN
  DBMS_CLOUD_AI_AGENT.CREATE_TOOL(
    tool_name  => 'ETL_LOG_READER_TOOL',
    attributes => JSON_OBJECT(
       'instruction' VALUE q'[Fetch exactly ONE next ETL log
            record after the provided start log id. ...
            Do not interpret or summarize;
            return factual data only.]',
       'function' VALUE 'RETRIEVE_NEXT_LOG_RECORD',
       'tool_inputs' VALUE JSON_ARRAY(
           JSON_OBJECT('name' VALUE 'p_log_id',
                       'description' VALUE 'Last processed log
                        id (exclusive).'))
                           )
                      );
END;
/
```

Copyright © 2026 Accenture. All rights reserved.

Tasks - Units of Work in Select AI Agent



- A task defines work that a Select AI agent can include in its reasoning process.
- Each task has a unique name and **attributes** that specify the agent's behavior when planning and performing the task:
  - **`instruction (mandatory) -`** concise statement, what the task should accomplish, included in the prompt
  - **`tools -`** tool names available for this task
  - **`input -`** outputs of other tasks are automatically provided as input to this task
  - **`enable_human_tool -`** allows the agent to ask the user for clarification or approvement (the default is true and it overrides the setting at the agent level)

```
DBMS_CLOUD_AI_AGENT.CREATE_TASK(
    task_name  => 'ETL_FETCH_LOGS_TASK',
    attributes => JSON_OBJECT(
             'instruction' VALUE q'[Fetch exactly ONE next ETL log record after start_log_id (from input).
                                    Call ETL_LOG_READER_TOOL with p_start_log_id=start_log_id. ... ]',
             'tools' VALUE JSON_ARRAY('ETL_LOG_READER_TOOL'),
             'input' VALUE 'ETL_GET_CHECKPOINT_TASK',
             'enable_human_tool' VALUE 'false'  )
             );
```

Copyright © 2026 Accenture. All rights reserved.

Agents - The Reasoning Entity



- An agent represents an AI entity that performs reasoning.
- Associated with an AI profile
- Accomplishes assigned tasks as part of a team workflow
- Attributes:
  - **`role -`** describes the agent's role/behavior, sent to LLM
  - **`profile_name -`** defines which model/profile is used
  - **`enable_human_tool -`** allows the agent to ask the user for clarification or approvement

```
BEGIN
  DBMS_CLOUD_AI_AGENT.CREATE_AGENT(
    agent_name => 'ETL_RESOLVER_AGENT',
    attributes => JSON_OBJECT(
           'profile_name' VALUE 'OCI_GPT_OSS120',
           'role' VALUE 'You are an ETL Incident Resolver. You use logs + optional KB (DIY-RAG) and policy checks
                         to produce a minimal Decision JSON. You never execute actions.'
               )
       );
END;
```

Copyright © 2026 Accenture. All rights reserved.

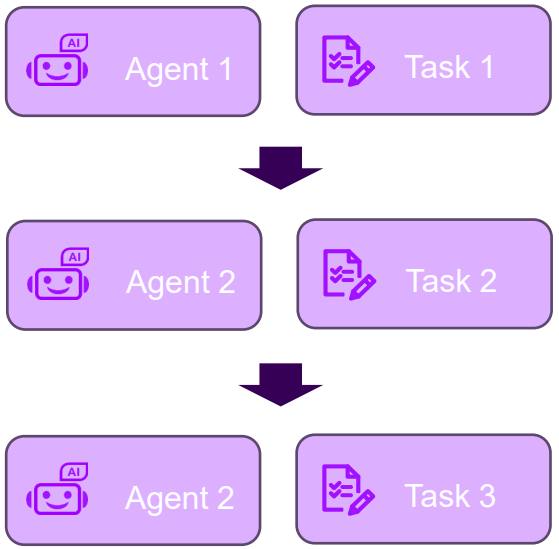
Team - Workflow Orchestration Layer



- Groups agents and tasks into an executable workflow
- Defines which agent performs which task – and in what order
- A single agent can appear multiple times with different tasks
- Each run belongs to a conversation (conversation_id)
- Maintains state across multi-step executions

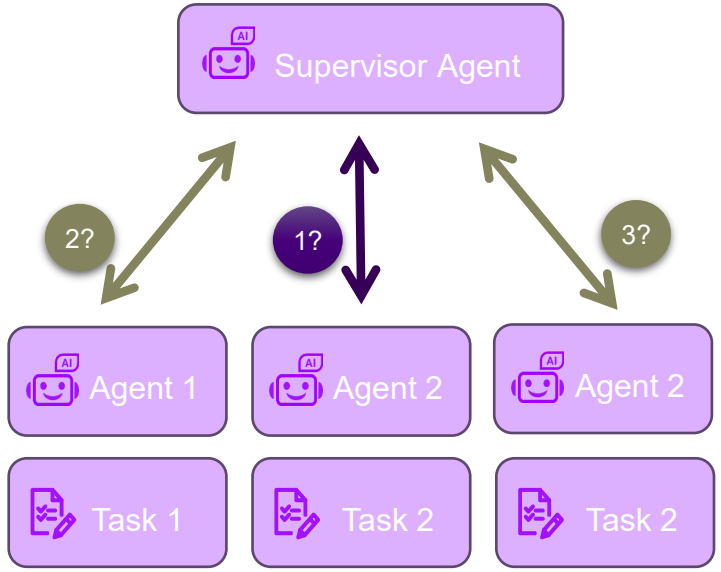
Two orchestration modes:

Fixed sequence



Workflow known in advance

Supervisor driven



Next step depends on previous result
Supervisor selects one at a time

Copyright © 2026 Accenture. All rights reserved.

Team - Fixed Sequence Team



- Tasks execute in predefined order, one by one
- Every step runs – no conditional branching
- Predictable, auditable, easy to reason about
- Best for well-defined operational workflows
- Attributes:
  - **`agents -`** a list of objects with mapping **`name / task`**
  - **`process -`** "sequential" (currently the only execution mode)

```
DBMS_CLOUD_AI_AGENT.CREATE_TEAM(
    team_name  => 'ETL_OPS_TEAM',
    attributes => '{ "agents": [
                            {"name":"ETL_TRIAGE_AGENT","task":"ETL_GET_CHECKPOINT_TASK"},
                            {"name":"ETL_TRIAGE_AGENT","task":"ETL_READ_LOG_TASK"},
                            {"name":"ETL_TRIAGE_AGENT","task":"ETL_RAG_TASK"},
                            {"name":"ETL_TRIAGE_AGENT","task":"ETL_DECIDE_TASK"},
                            {"name":"ETL_EXECUTOR_AGENT","task":"ETL_EXECUTE_TASK"}    ],
      "process":"sequential"   }'
  );
```

Copyright © 2026 Accenture. All rights reserved.

Team - Supervisor Agent



- A supervisor agent receives the user prompt
- Dynamically selects which worker agent-task pair to call next
- Waits for the result, then decides what comes next – or stops
- Can skip steps, repeat, or request human input
- **`process:"sequential"`** still applies (not parallel) – but order is not fixed
- New attribute **`supervisor_agent`** - names the coordinating agent

```
DBMS_CLOUD_AI_AGENT.CREATE_TEAM(
    team_name  => 'ETL_DYNAMIC_TEAM',
    attributes => '{
      "supervisor_agent": "ETL_SUPERVISOR",  -- new
      "agents": [
        {"name":"ETL_TRIAGE_AGENT", "task":"ETL_READ_LOG_TASK"},
        {"name":"ETL_RESOLVER_AGENT","task":"ETL_DECIDE_TASK"},
        {"name":"ETL_EXECUTOR_AGENT","task":"ETL_EXECUTE_TASK"}
      ],
      "process":"sequential"
    }');
```

Copyright © 2026 Accenture. All rights reserved.

Runtime Configuration

Precision Control Over Data, Cost, and Context

Namespace	Purpose	Typical Use
state_variables	Runtime state shared between tools	🔒 Sensitive data, tool-to-tool handoff
instruction_variables	Replace {placeholders} in agent roles & task instructions	🔀 Schema, tenant, target, environment per run
attribute_variables	Override agent/tool attributes at runtime	💰 AI profile/model selection, tool parameters

- **`SET_TEAM`** sets session-level defaults for subsequent runs; **`RUN_TEAM`** can supply or override values for a single run.
- **`SET_VARIABLE / GET_VARIABLE`** keep sensitive or large intermediate data in tool state instead of passing it through the LLM context.

```
-- Team
DBMS_CLOUD_AI_AGENT.SET_TEAM('ETL_OPS_TEAM', '{"state_variables":{"env":"PROD"}}');
-- Tool to Tool
DBMS_CLOUD_AI_AGENT.SET_VARIABLE('state_variables', 'action', 'RESTART_ETL');
...
l_action := DBMS_CLOUD_AI_AGENT.GET_VARIABLE('state_variables', 'action');
```

Copyright © 2026 Accenture. All rights reserved.

Observability – Monitoring Agent Execution

- Transparency via dictionary views (retention time)
- Inspect definitions, runs, and attributes
- Analyze prompts, tool usage, and outputs
- Standard SQL-based debugging
- No black box – everything is queryable
- Dictionary views to inspect object definitions

- DBA|USER_AI_AGENT_TEAM_HISTORY
- DBA|USER_AI_AGENT_TASK_HISTORY
- DBA|USER_AI_AGENT_TOOL_HISTORY
- DBA|USER_CLOUD_AI_CONVERSATIONS
- DBA|USER_CLOUD_AI_CONVERSATION_PROMPT
- DBA|USER_SCHEDULER* views can also be useful, since tasks are running using DBMS_SCHEDULER

```
SELECT team.team_name, team.state, team.start_date, task.task_name, p.prompt_response
FROM   user_ai_agent_team_history team
       JOIN  user_ai_agent_task_history task    ON team.team_exec_id = task.team_exec_id
       LEFT JOIN  user_cloud_ai_conversation_prompts p
             ON p.conversation_id = JSON_VALUE(task.conversation_params, '$.conversation_id')
ORDER BY team.start_date desc, task.start_date desc  ;

TEAM_NAME       STATE      START_DATE TASK_NAME            PROMPT_RESPONSE
--------------- ---------- ---------- -------------------- ----------------------------------
ETL_OPS_TEAM    SUCCEEDED  05.05.2026 ETL_EXECUTE_TASK     Thought: Need to notify DB Ops
                                                           about the error via email as
                                                           specified by ACTION=NOTIFY_DB_OPS.
...
```

Copyright © 2026 Accenture. All rights reserved.

Running a Team

SQL Interface (interactive)

- Interactive usage (SQL client, SQL Developer, etc.)
- Conversation context maintained within the session
- Next SELECT AI AGENT continues the conversation
- Suitable for exploration and debugging

```
BEGIN
   DBMS_CLOUD_AI_AGENT.SET_TEAM('OPS_TEAM');
END;
/

SELECT AI AGENT 'Analyze recent ETL errors';
...

BEGIN
   DBMS_CLOUD_AI_AGENT.CLEAR_TEAM;
END;
```

PL/SQL Interface (programmatic)

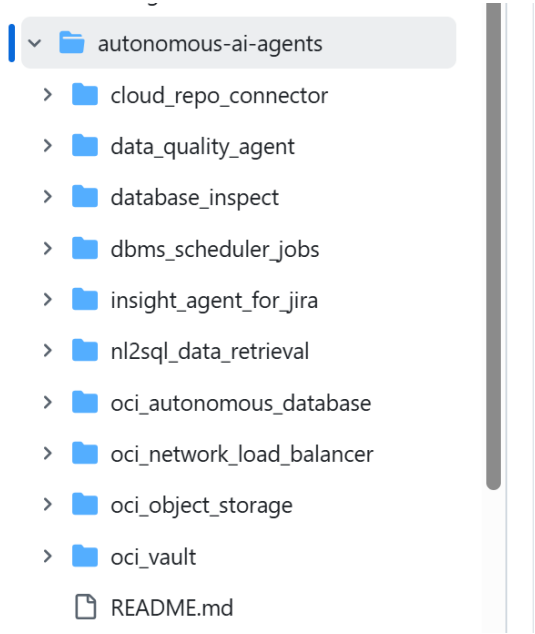
- Designed for applications and APIs
- Explicit control over conversation_id
- Enables resuming workflows across sessions
- Suitable for automation and integrations
- Each run belongs to a **conversation,** identified by **conversation_id**
- Maintains workflow state
- Allows continuation of paused executions (e.g., WAITING_FOR_HUMAN)

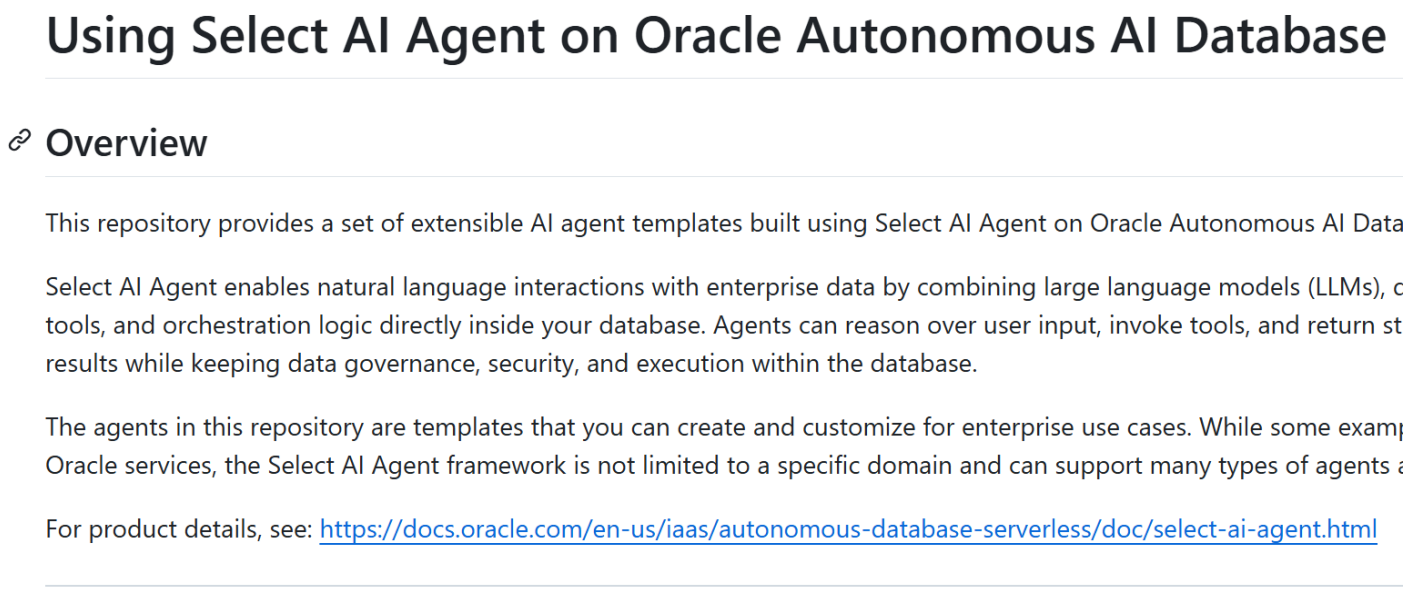
```
l_final_answer := DBMS_CLOUD_AI_AGENT.RUN_TEAM(
      team_name => ' 'OPS_TEAM',
      user_prompt => 'I approve',
      params => '{"conversation_id": "'
                    ||l_conversation_id || '"}'
     );
```

Copyright © 2026 Accenture. All rights reserved.

Pre-built Agents – Extending Capabilities

- Oracle provides pre-built agents for common tasks via GitHub-Repo
- Examples:
  - Database schema exploration
  - Object Storage interaction
  - Jira integration
- Ready-to-use building blocks, can be extended and combined with custom agents





Using Select AI Agent on Oracle Autonomous AI Database

Overview

This repository provides a set of extensible AI agent templates built using Select AI Agent on Oracle Autonomous AI Data

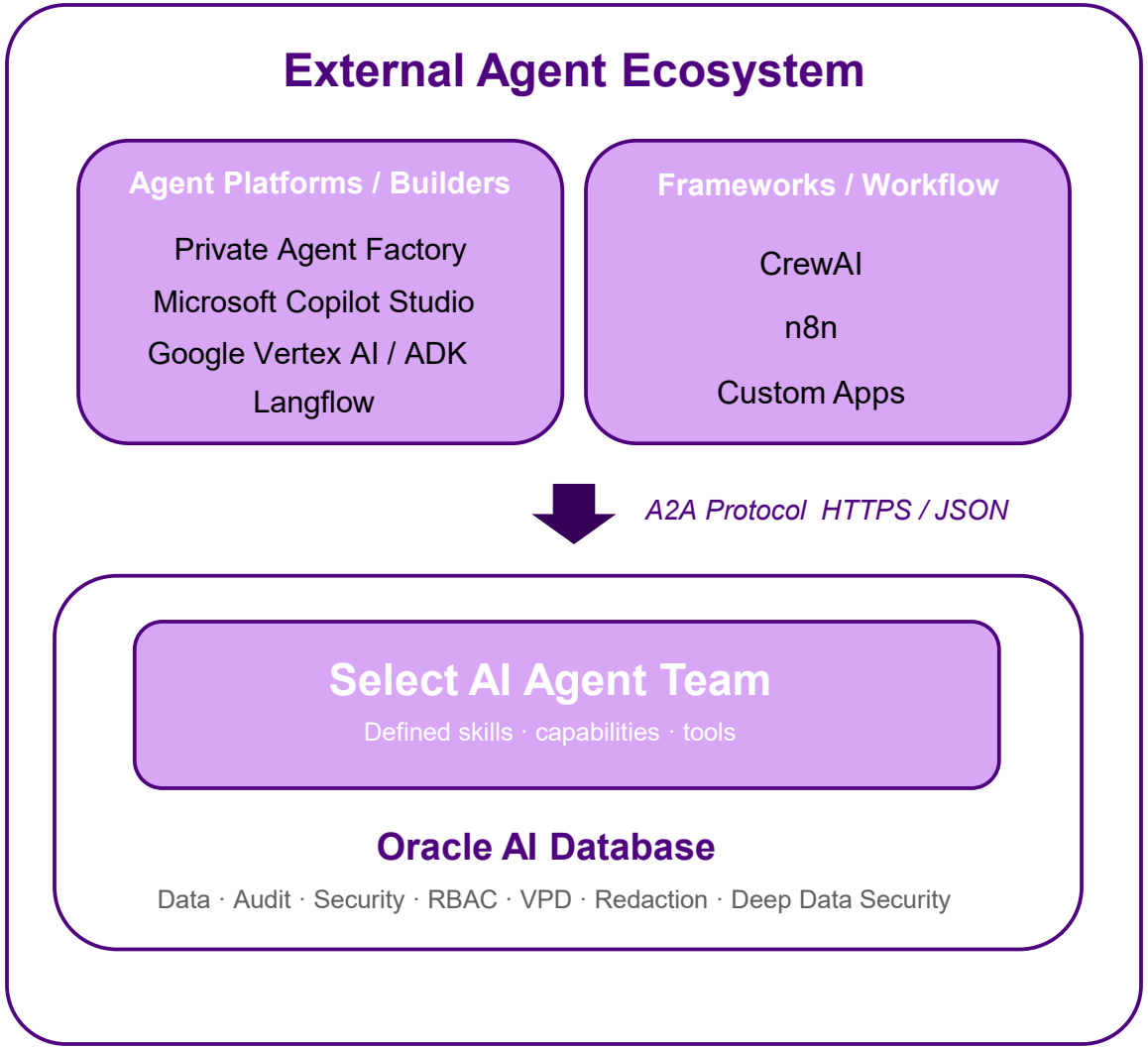
Select AI Agent enables natural language interactions with enterprise data by combining large language models (LLMs), c tools, and orchestration logic directly inside your database. Agents can reason over user input, invoke tools, and return st results while keeping data governance, security, and execution within the database.

The agents in this repository are templates that you can create and customize for enterprise use cases. While some examp Oracle services, the Select AI Agent framework is not limited to a specific domain and can support many types of agents a

For product details, see: https://docs.oracle.com/en-us/iaas/autonomous-database-serverless/doc/select-ai-agent.html

Copyright © 2026 Accenture. All rights reserved.

A2A: DB Agents in External Workflows



Why A2A?

Not a binary choice

DB-native agents and external agent frameworks can coexist. Expose selected DB agent teams through A2A.

Controlled DB interface

Expose defined agent capabilities instead of general database access.
Agent logic controls what is accessed, processed, and returned.

Standard interface & discovery

Agent Cards expose identity, skills and capabilities.
A2A-compatible clients discover and invoke agent teams via HTTPS.

Copyright © 2026 Accenture. All rights reserved.

SELECT AI Agent - The Pace of Innovation

Oct 2025 — **SELECT AI Agent Framework**

ReAct pattern | Built-in tools | Multi-turn conversations | PL/SQL interface

Dec 2025 — **MCP Server**

External AI clients access DB agents via Model Context Protocol

Mar 2026 — **Pre-built Agents Suite**

8 ready-to-use agents on GitHub

Jul 2026 — **Supervisor Agent + A2A Server**

Dynamic orchestration | Enterprise agent interoperability

Aug 2026 — **Import / Export Agent Teams**

Migration | Backup | Version control across environments

Sep 2026 — **Cross-Schema Team Sharing**

Reuse teams across database schemas

Oct 2026 — ***This talk.***

Already outdated by the time you read this.

Copyright © 2026 Accenture. All rights reserved.

Resources

- Getting Started with Select AI Agent
- DBMS_CLOUD_AI_AGENT Package
- Autonomous AI Database Learning Lounge - Get started with the Select AI Agent Framework and Pre-built AI Agents
- Select AI Agent Framework Live Lab
- Select AI Prebuilt agents
- Demos used in this talk

Copyright © 2026 Accenture. All rights reserved.

Key Takeaways

- Agentic AI directly inside the database
- Combines reasoning with data and actions
- Strong fit for data-centric workflows
- Transparent, controllable, SQL-driven
- Still evolving – but already practical



https://github.com/andrejsql/oracle-demo

Copyright © 2026 Accenture. All rights reserved.

Thank You

Oracle ACE Pro

blog.sqlora.com

linktr.ee/andrejpashchenko