

ORACLE

SQL Dialog

Program agenda

- 1 **What is SQL Dialog?**
- 2 Steps to build an SQL Dialog
- 3 SQL Dialog training data
- 4 SQL Dialog presentation
- 5 SQL Dialog queries

”

SQL Dialog provides
a **natural language
conversational interface to
databases**

How does it work?

Can you show me the employees from our office in New York?



Natural language to SQL conversion



EMPNO	ENAME		EMPNO
1	7839 BLAKE		10
2	7839 BLAKE		20
3	7839 BLAKE		30
4	7839 BLAKE		40
5	7839 BLAKE		50
6	7839 BLAKE		60
7	7839 BLAKE		70
8	7839 BLAKE		80
9	7839 BLAKE		90
10	7839 BLAKE		100
11	7839 BLAKE		110
12	7839 BLAKE		120
13	7839 BLAKE		130
14	7839 BLAKE		140

job	mgr	ename	
PRESIDENT	null	KING	>
MANAGER	7839	BLAKE	>
MANAGER	7839	CLARK	>
MANAGER	7839	JONES	>
ANALYST	7566	SCOTT	>
ANALYST	7566	FORD	>
CLERK	7902	SMITH	>
SALESMAN	7698	ALLEN	>
SALESMAN	7698	WARD	>
SALESMAN	7698	MARTIN	>

Showing 1-10 of 14 items

Next 4 Items

Is this answer helpful?

👍

👎



How does it work?

Oracle meaning representation query language (**OMRQL**) is a query based on the canonical names of the database table attributes.

Utterance	Can you show me the employees from our office in New York?
Interpretation	Show all the emp where the dept loc is "NEW YORK".
Translatable	Yes
Confidence Score	1
OMRQL	<pre>SELECT * FROM Emp WHERE dept.loc = 'NEW YORK'</pre>
SQL	<pre>SELECT T1.EMPNO, T1.ENAME, T1.JOB, T1.MGR, T1.HIREDATE, T1.SAL, T1.COMM FROM EMP T1 JOIN DEPT T2 ON T1.DEPTNO = T2.DEPTNO WHERE T2.LOC = ? FETCH FIRST 100 ROWS ONLY</pre>

OMRQL is then converted into the **SQL Query** which is used to query the database

Where do we define the intents?

” SQL Dialog does not require Intents!

It uses a special entity (**query entity**) that is mapped to the database schema

The **query entity** has attributes mapped to the table columns

No need to identify all potential queries ahead of time - that means no intents, no utterances and custom components!

SQL Dialog Benefits



Supports a large number of queries without having to create individual intents for each of them



Reduce skill creation time from days to hours



Add a quick conversational layer on top of DB-based applications

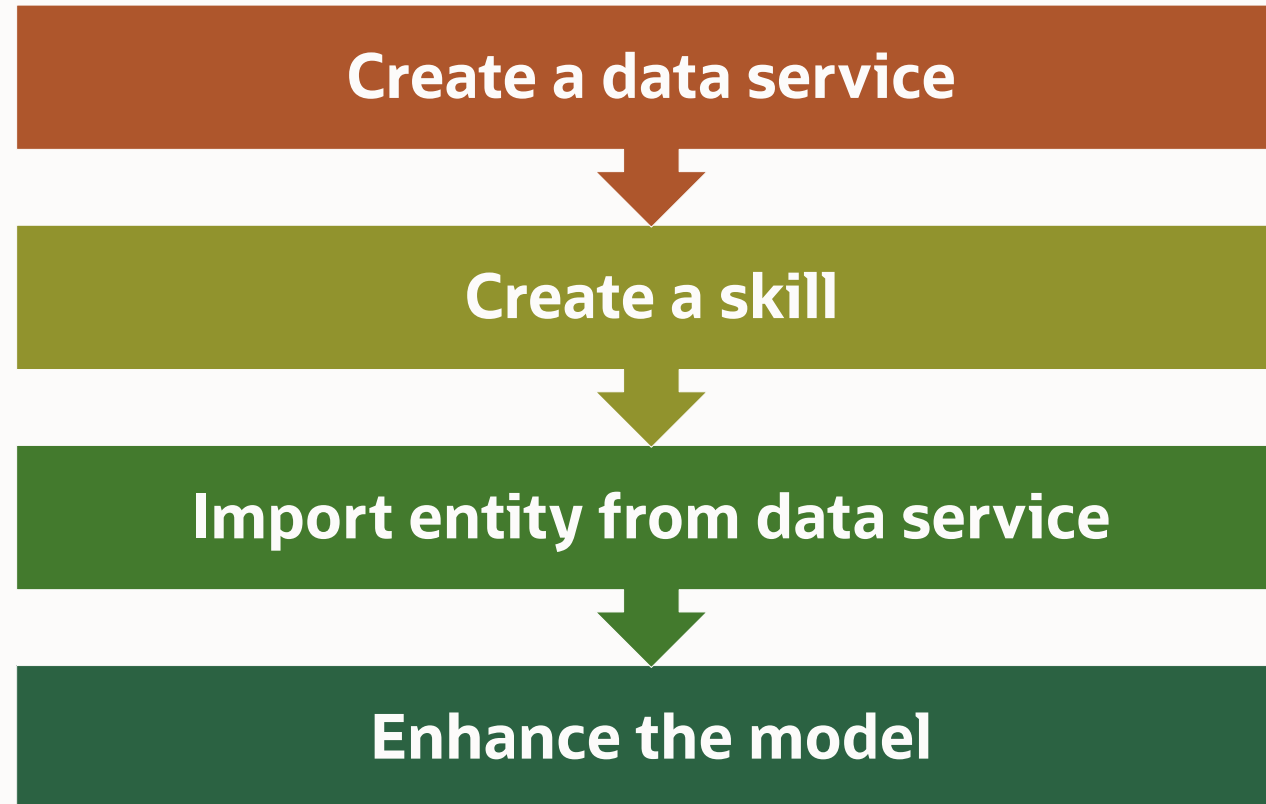


Use SQL Dialogs with small-talk, FAQs, and other answer intents to provide a robust conversational experience.

Program agenda

- 1 What is SQL Dialog?
- 2 **Steps to build an SQL Dialog**
- 3 SQL Dialog training data
- 4 SQL Dialog presentation
- 5 SQL Dialog queries

Steps to build a SQL Dialog



Create a data service

The first step is to configure a data service

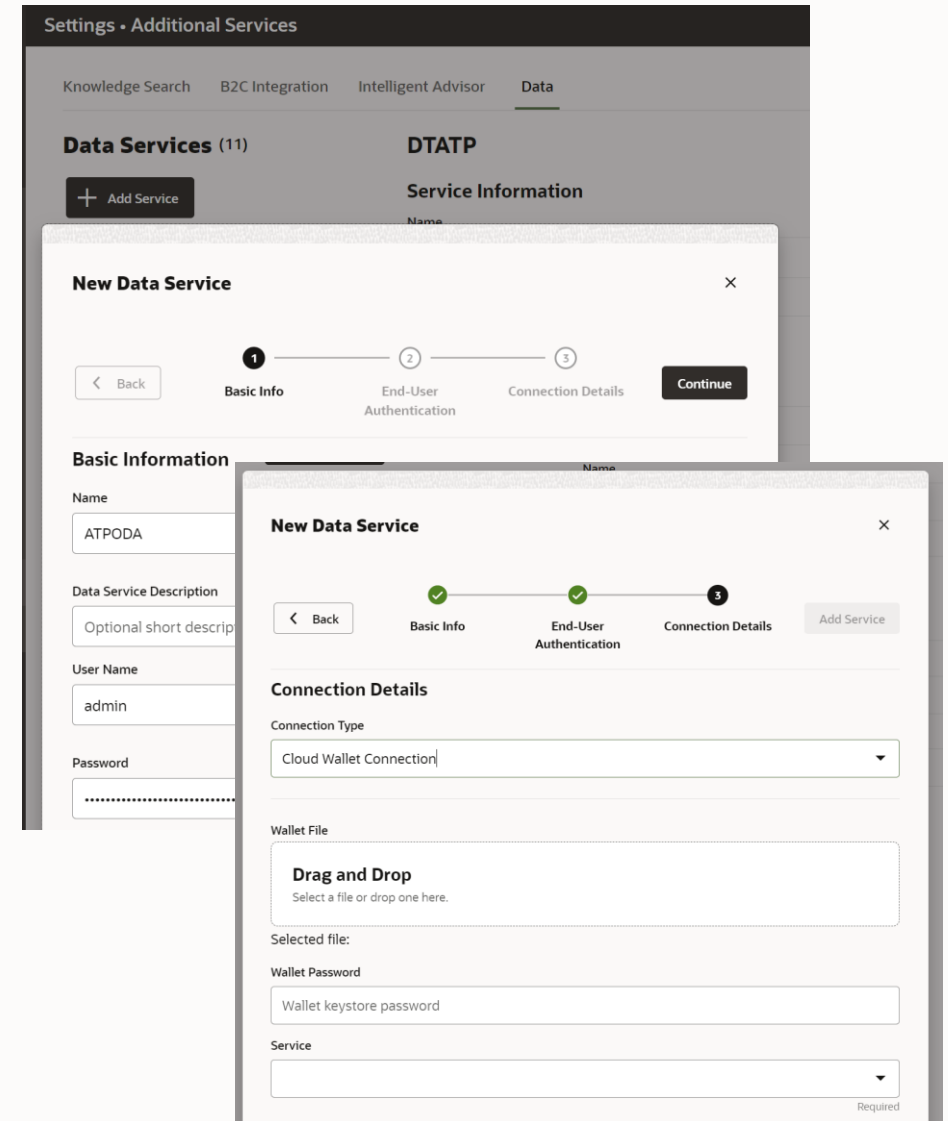
- You should have a database ready to use

Go to **Settings>Additional Services** and choose the data tab

The same data service can be used by multiple skills

Currently only the Oracle DBCS and ADB are supported

You can use a **Cloud Wallet** Connection Type or a **Basic** one



Create a skill

The skill must be using the **Visual Dialog Mode**

- YAML is not supported

The **Primary** Language needs to be English

- This version doesn't support multi-language SQL Dialog skills or skills where the primary language is not English

Create Skill [Close]

Display Name
sqldialog

Name
sqldialog

Skill Version: 1.0 Platform Version: 22.12 (Latest)

Dialog Mode [?]
 YAML Visual

Primary Language (Natively-Supported) [?]
English

One-Sentence Description

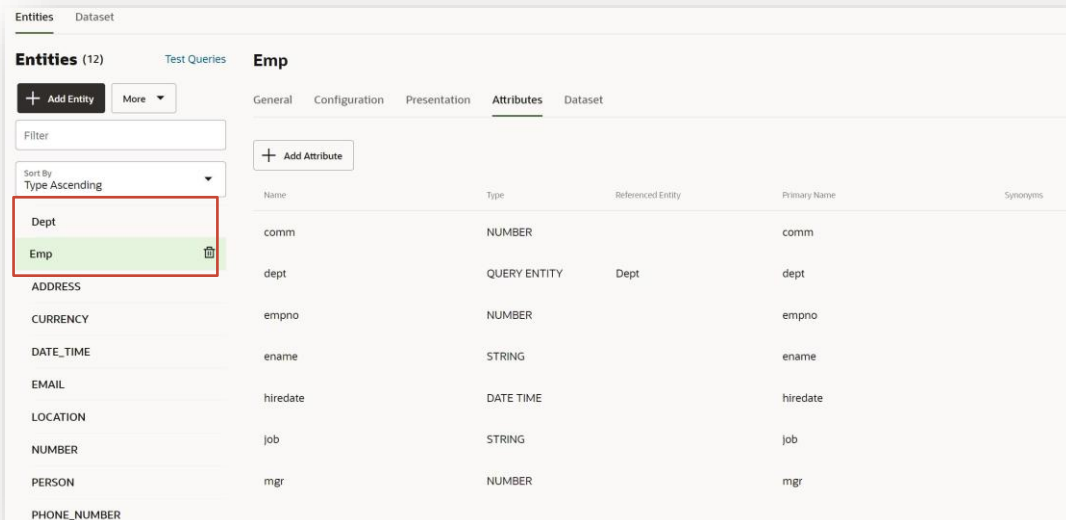
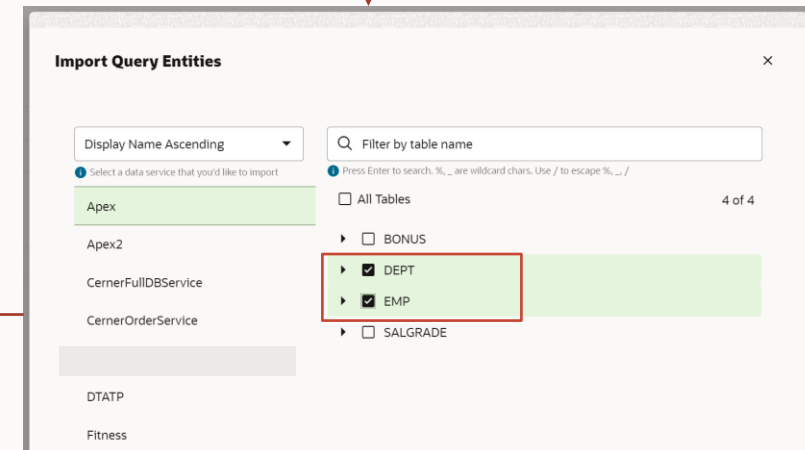
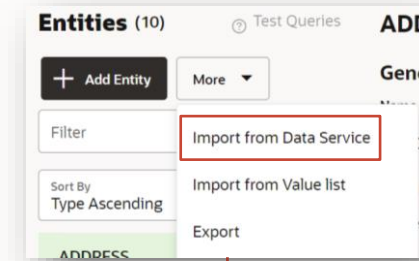
[Create]

Import entity from data service

In the entities tab, use the **Import From Data Service** to automatically create **Query Entities**

You can choose the required tables and attributes

Each table will create a **query entity** with the selected attributes



Enhance the model

To improve the model we should add **synonyms**, **value lists** and **utterances**, to help the skill associate natural language phrases, with the physical model's tables and columns

▼ **Natural Language**

Primary Language: English

Primary Name

Emp

Synonyms

employee,staff,employees,person,people,personnel,workforce [Edit](#)

Emp

General Configuration Presentation Attributes Dataset

+ Add Attribute

Name	Type	Referenced Entity	Primary Name	Synonyms
comm	NUMBER		comm	commission,bonus
dept	QUERY ENTITY	Dept	dept	department
empno	NUMBER		empno	number,employee number
ename	STRING		ename	name,employee name



Enhance the model

With an enhanced model, the better the natural language queries are associated with an SQL

“can you show me all the employees in the big apple?”

The screenshot shows a chat interface with a user query and a system response. The user asks, "can you show me all the employees in the big apple?". The system responds with a table of employee data and a feedback prompt.

I think you want to see: Show all the emp where the dept loc is "NEW YORK".

empno	ename	job
7782	CLARK	MANAGER
7934	MILLER	CLERK
7839	KING	PRESIDENT

Is this answer helpful?

Feedback buttons: thumbs up and thumbs down.

Program agenda

- 1 What is SQL Dialog?
- 2 Steps to build an SQL Dialog
- 3 **SQL Dialog training data**
- 4 SQL Dialog presentation
- 5 SQL Dialog queries

SQL Dialog training data

” We should add training data to improve the skill ability to interpret and route queries

If your skill has intents, or it is in a DA, then, just like with intents, your skill needs utterances to help it route SQL queries to the SQL Dialogs conversation

The routing mechanism uses the autocomplete suggestions, training data, generated routing utterances, and handcrafted routing utterances to learn how to recognize SQL queries.

You can see each type of utterance in the separate tabs on the **Query Entities Dataset** page.

Autocomplete

You can help users learn about the database queries that they can make by providing autocomplete suggestions

These suggestions provide hints about what kinds of questions the logical model can answer

The utterances also help the skill with routing

Create Utterance

1 Basic Info 2 Review Query Done

Review Query
Utterance
show all employees

Query
1 SELECT * FROM Emp

Interpretation
Show all the emp.

Reinterpret

Query Entities Dataset

Use this dataset to add suggestions for autocomplete of SQL queries, add utterance-to-OMRQL training data, generate

Autocomplete Suggestions Training Data Generated Routing Data Combined Routing Data

Entities

Select one or more entities Filter Utterance

Autocomplete Suggestions	Interpretation	Query
show all departments	Show all the dept.	SELECT * FROM Dept
show all employees	Show all the emp.	SELECT * FROM Emp

Training data

Where the skill is not interpreting natural language queries correctly, you can use custom training data to teach the correct interpretation

Creating utterances follows the same guidelines as for the *standard* utterances

- Balance the number of utterances
- Grammatical diversity
- Diversify Values
- Balance the training of values that match primary names or synonyms

Create Utterance



< Back

1 Basic Info 2 Review Query

Done

Review Query

Utterance

show all the managers

Query

1 SELECT * FROM Emp

Interpretation

Show all the emp.

Reinterpret

"show me all the managers" will create a query that is not correct, as it does not filter by the job attribute

We can manually correct the query

Query

1 SELECT * FROM Emp WHERE job='MANAGER'|

Generated routing data

Accurate routing of utterances to the SQL conversation requires a lot of sample utterances with the translated query. On the **Generated Routing Data** tab, you can quickly generate 100 utterances that reflect questions the logical model can answer

Those generated utterances are based on templates (buckets) and can be used to filter the data

Must be reviewed before use. Those that you approve are added to the **Combined Routing Data** tab and are marked as either *synthetic* or, if you edited them, *refined*

Bucket	Utterance	Interpretation	Query	Generated	
Order by limit	Show all the dept with the top 4 deptno.	Show all the dept with the top 4 deptno.	<code>SELECT * FROM dept ORDER BY deptno DESC LIMIT 4</code>	a few seconds ago	Approve
Group by attribute + order by limit	Find ename with the top average sal.	Find ename with the top average sal.	<code>SELECT ename FROM emp GROUP BY ename ORDER BY AVG(sal) DESC LIMIT 1</code>	a few seconds ago	Approve
Show two attributes	Give me loc and deptno of dept.	Give me loc and deptno of dept.	<code>SELECT loc, deptno FROM dept</code>	a few seconds ago	Approve
Show an aggregation	Display the maximum empno of emp.	Display the maximum empno of emp.	<code>SELECT MAX(empno) FROM emp</code>	a few seconds ago	Approve
Order by limit	Get all emp with the top 100 comm.	Get all emp with the top 100 comm.	<code>SELECT * FROM emp ORDER BY comm DESC LIMIT 100</code>	a few seconds ago	Approve

Bucket

- All
- Show entity
- Filter on numbers or dates
- Two conditions with OR
- Two conditions with AND
- Filter on a link attribute
- Show two attributes
- Show an aggregation



Combined routing data

The combined routing data is an aggregation of all autocomplete suggestions, custom training data, and generated and hand crafted routing data

Hand crafted data is manually defined routing data

All of these help the skill route SQL queries to the SQL conversation

The screenshot shows the 'Query Entities Dataset' interface. At the top, there are four tabs: 'Autocomplete Suggestions', 'Training Data', 'Generated Routing Data', and 'Combined Routing Data'. Below the tabs, there are three dropdown menus: 'Entities' (with a search box 'Select one or more entities'), 'Type' (set to 'Routing'), and 'Routing Subtype' (set to 'All'). A 'Filter Utterance' search box is also present. Below these filters is a table with columns: 'Type', 'Routing', 'Autocomplete/Utterance', 'Interpretation', and 'Query'. The table contains four rows of data.

Type	Routing	Autocomplete/Utterance	Interpretation	Query
Routing	Synt	Show all the dept with the top 4 ...	Show all the dept with the top 4 ...	SELECT *
Routing	Synt	Find ename with the top averag...	Find ename with the top averag...	SELECT e
Routing	Synthetic	Give me loc and deptno of dept.	Give me loc and deptno of dept.	SELECT 1
Routing	Synthetic	Display the maximum empno of...	Display the maximum empno of...	SELECT M

Query Entities Dataset

Use this dataset to add suggestions for autocompletion of SQL queries, add utterance-to-OMRQL training data, generate utterances for routing to S

Autocomplete Suggestions Training Data Generated Routing Data **Combined Routing Data**

The screenshot shows the 'Query Entities Dataset' interface. At the top, there are four tabs: 'Autocomplete Suggestions', 'Training Data', 'Generated Routing Data', and 'Combined Routing Data'. Below the tabs, there are three dropdown menus: 'Entities' (with a search box 'Select one or more entities'), 'Type' (set to 'All'), and 'Routing Subtype' (set to 'All'). A 'Filter Utterance' search box is also present. Below these filters is a table with columns: 'Type', 'Routing', 'Autocomplete/Utterance', 'Interpretation', and 'Query'. The table contains four rows of data.

Type	Routing	Autocomplete/Utterance	Interpretation	Query
Routing	Synt	Show all the dept with the top 4 ...	Show all the dept with the top 4 ...	SELECT *
Routing	Synt	Find ename with the top averag...	Find ename with the top averag...	SELECT e
Routing	Synthetic	Give me loc and deptno of dept.	Give me loc and deptno of dept.	SELECT 1
Routing	Synthetic	Display the maximum empno of...	Display the maximum empno of...	SELECT M



Program agenda

- 1 What is SQL Dialog?
- 2 Steps to build an SQL Dialog
- 3 SQL Dialog training data
- 4 **SQL Dialog presentation**
- 5 SQL Dialog queries

SQL Dialog presentation

You can control how to present the data to the user

Some of the things it can do:

- Configure Whether to Display Form or Table
- Show One or Two Horizontal Sections in Form
- Set the Title for the Results
- Define an Entity's Default Sort Order
- Define Which Attributes to Include When Not Specified by the Utterance
- Define Which Attributes to Always Include in the Results
- Configure the Results Page Size
- Add Buttons and Links to Results
- Add a Custom Attribute
- Use Event Handlers for more customization

The screenshot shows the configuration interface for an entity named 'Emp'. The 'Presentation' tab is active, showing options for 'Response Configuration' and 'Layout Conversion Configuration'. Under 'Response Configuration', there are sections for 'Default Attributes' (with an 'Edit' link), 'Minimum Attributes' (with an 'Edit' link), and 'Event Handler' (set to 'Not Defined'). There is a 'Display Name' text input field. Under 'Layout Conversion Configuration', there are two dropdown menus: 'Maximum Number of Rows per Page' (set to 10) and 'Number of Horizontal Sections in Form Layout' (set to 1). Below these, there are two more dropdown menus: 'Use form layout for this number of rows or less' (set to 3) and 'Switch table layout to table form layout when number of columns exceeds this number' (set to 3). A 'Required' label is visible next to the 'Number of Horizontal Sections in Form Layout' dropdown.

SQL Dialog presentation

When it comes to the response layout you should start by defining Default and Minimum attributes.

Emp

General Configuration **Presentation** Attributes

Response Configuration

Default Attributes
empno,ename,hiredate,job [Edit](#)

Minimum Attributes
empno,ename [Edit](#)

If the utterance doesn't name any attributes, then you probably want the results to include **Default Attributes**.

show all employees

Emp

empno	ename	hiredate	job
7839	KING	1981-11-17	PRESIDENT
7698	BLAKE	1981-05-01	MANAGER
7782	CLARK	1981-06-09	MANAGER
7566	JONES	1981-04-02	MANAGER

When an utterance identifies specific attributes, you might want the result to include not only the requested attributes, but also **Minimum Attributes**.

“show me the employees salary”

Here we only ask for the salary, but the minimum list will define the extra attributes to show

show me the employees sal

Emp

empno	ename	sal
7839	KING	5000
7698	BLAKE	2850
7782	CLARK	2450
7566	JONES	2975

SQL Dialog presentation

Formatting an attribute

Attributes can be formatted on an individual basis with a series of predefined parameters

- Date/Time/Number have specific **format masks**

For further formatting capabilities, entity event handlers have access to the attributes and allow further customization

EMP/DEPT SQL Dialogs Demo ... x
CONNECTED

Wed Oct 12, 11:20 PM

show all employees

employee

employee number	name	job
7839	KING	PRESIDENT
hire date		
1981-11-17		
7698	BLAKE	MANAGER
7782	CLARK	MANAGER
7566	JONES	MANAGER
7788	SCOTT	ANALYST

Type a message

Customized EMP/DEPT Demo ... x
CONNECTED

employee number	name	job
7839	KING	PRESIDENT
hire date		
Tue Nov 17 1981		
7698	BLAKE	MANAGER
7782	CLARK	MANAGER
7566	JONES	MANAGER
7788	SCOTT	ANALYST

Showing 1-5 of 14 items

Next 5 Items

Show Jobs Count

A few seconds ago

Type a message



SQL Dialog presentation

Global action – Show all departments

You can add buttons and links to a query entity's results at both the global level and the row level

We can optionally restrict when the action appears. with **Visibility Expression** where we can provide a FreeMarker expression, such as `${row.job = 'MANAGER'}`.

Follow-Up Actions Configuration

+ Add Global Action

Sequence	Type	Button Label	Query/URL
1	QUERY	Show all departments	SELECT * FROM DEPT

Add Global Action

Button Label

Action Type


Query Result Title

Query

```
1 SELECT * FROM DEPT
```

Visibility Expression

Add



empno	ename	hiredate	
7839	KING	1981-11-17	▼
7698	BLAKE	1981-05-01	▼
7782	CLARK	1981-06-09	▼
7566	JONES	1981-04-02	▼
7788	SCOTT	1987-04-19	▼
7902	FORD	1981-12-03	▼
7369	SMITH	1980-12-17	▼
7499	ALLEN	1981-02-20	▼
7521	WARD	1981-02-22	▼
7654	MARTIN	1981-09-28	▼

Showing 1-10 of 14 items

Show all departments





SQL Dialog presentation

Row action – show all employees with same job

For row actions you need to create a query that uses data from the row to invoke a new query

- `${row.<attribute_name>}` allows to dynamically do this

+ Add Row Action			
Sequence	Button Label	Query/URL	
1	Show all employees with same job	SELECT * FROM EMP WHERE job = '\${row.job}'	 

I think you want to see: Show all the emp.

Emp

empno	ename	hiredate	
7839	KING	1981-11-17	▼
7698	BLAKE	1981-05-01	▼
7782	CLARK	1981-06-09	▲

job
MANAGER

Show all employees with same job



7566	JONES	1981-04-02	▼
7788	SCOTT	1987-04-19	▼
7902	FORD	1981-12-03	▼
7369	SMITH	1980-12-17	▼
7499	ALLEN	1981-02-20	▼
7521	WARD	1981-02-22	▼
7654	MARTIN	1981-09-28	▼

Showing 1-10 of 14 items

Show all departments

Next 4 items

Is this answer helpful?



 

Show all employees with same job

Emp

empno	ename	hiredate	
7698	BLAKE	1981-05-01	▼
7782	CLARK	1981-06-09	▼
7566	JONES	1981-04-02	▼

Is this answer helpful?

Now



SQL Dialog presentation

Event Handlers – SQL Query Event Handler

An SQL Query Event Handler (SQEH) enables you to customize the SQL Dialogs query results.

The SQEH is deployed as part of a component service. You use below command to add an event handler to an existing package.

```
bots-node-sdk init component myEventHandler sqlQueryEventHandler
```

This example creates a component of type **sqlQueryEventHandler** that is named **myEventHandler**.

https://github.com/oracle/bots-node-sdk/blob/master/DATA_QUERY_EVENT_HANDLER.md

SQL Dialog presentation

Event Handlers – SQL Query Event Handler

You can have **Entity** level and **Attribute** level events

Entity

- changeUISettings
- changeResponseData
- changeBotMessages

Attribute

- changeUISettings
- format

```
changeBotMessages: async (event, context) => {  
  let message = event.messages[event.messages.length - 1];  
  let millis = context.getQueryExecutionTime();  
  let minutes = Math.floor(millis / 60000);  
  let seconds = ((millis % 60000) / 1000).toFixed(3);  
  message.footerText = `The query was executed in ${minutes > 0 ? minutes +  
    ' minutes and ' : ''}${seconds} seconds.\n\n${message.footerText}`;  
  return event.messages;  
}
```

Example that adds a footer text that displays the execution time of the query

Emp			
empno	ename	hiredate	
7782	CLARK	1981-06-09	>
7934	MILLER	1982-01-23	>
7839	KING	1981-11-17	>

The query was executed in 0.003 seconds.

Program agenda

- 1 What is SQL Dialog?
- 2 Steps to build an SQL Dialog
- 3 SQL Dialog training data
- 4 SQL Dialog presentation
- 5 **SQL Dialog queries**

SQL Dialog supported queries

” The SQL Dialogs natural language processing model supports queries that translate to the basic SQL clauses: **SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY** and **LIMIT**.

SQL Dialog supported queries

Examples of supported queries

Display

show me all the employees

name, salary and department name of employees

what are the unique jobs that an employee can have?

how many employees

return the highest salary of all employees

Filters

show the names of employees whose job title is clerk

return the salary of all clerks

employees whose names starts with Jo

view employees who joined in 2020 and earn above 7000

Filters with dates

Absolute example: who were the employees hired on 10 Dec 2020

Relative interval example: who were the employees hired last year

Relative Date example: who were the employees hired today

Date + time example: packages delivered yesterday at 6 pm

Date + time interval example: Packages delivered between 5th Jan 7 pm and 5th Feb 10 am

Ordering and limiting the number of rows

show employees sorted by their department names

return the name and salary of all employees in descending order of salary

what are the 10 highest salaries of all employees

which employee has the lowest salary

show the top 5 employees

Group by

the average salary of each job

what is the highest salary per department?

show the name, location and number of employees per department

show all jobs with an average salary above 3000

SQL Dialog un-supported queries

SQL Dialogs doesn't support the more complex queries that involve sub-queries and SET operators (INTERSECT, UNION, EXCEPT, and NONE)

Other un-supported queries (no conversion to OMRQL)

- Non-English query
- Use of pronouns -> *“What is **my** salary?”*
- Yes and no questions -> *“is John a clerk?”*
- Negation -> *“Which employee is **not** on location Amsterdam?”*
- Selecting more than one entity -> *“Show me all employees and departments”*
- SQL queries that involve sub-queries and SET operators (INTERSECT, UNION, EXCEPT, and NONE)
 - *“what is the total remuneration earned by each employee?”*

Check the [documentation](#) for a full list

More resources - Tutorial

The screenshot shows a web page titled "Get Started with SQL Dialogs" under the path "Cloud / Cloud Platform / Digital Assistant". The page is divided into a left sidebar and a main content area. The sidebar contains a "Table of Contents" with sections like "Before you Begin", "Background", "What Do You Need?", "Create the Employee Database", "Connect Digital Assistant to the Data Service", "Create the SQL Dialog Skill", "Import the Database Schema", "Add Natural Language Names and Synonyms for the Query Entities", "Add Natural Language Names and Synonyms for the Attributes", "Associate Attributes with Value Lists", "Add Synonyms to the Value Lists", "Test Queries Using the Primary Names, Synonyms, and Value List Values", "Optional Step: Train with Custom Data", "OMRQL Keyword Reference", and "Learn More". The main content area has a heading "Before you Begin" with a sub-heading "Before you Begin" and a paragraph: "This 30-minute tutorial shows you how to create a SQL dialog with Oracle Digital Assistant." Below this is a section "Background" explaining that SQL Dialogs are skills for natural language interaction. Another section "What Do You Need?" lists requirements: "Access to Oracle Digital Assistant" and "Access to Oracle Database Cloud Service Enterprise Edition". A tip icon indicates that a free trial of Oracle Cloud Service Enterprise Edition is needed. The next section is "Create the Employee Database", which includes a numbered step: "1. Go to your Autonomous Database instance and then click **Database Actions**." Below this step is a screenshot of the Oracle Cloud console showing the "Database actions" button highlighted in a red box. The console also displays "Autonomous Database Information" for instance T1PSZYJUE16HIZUF, including details like "Database name: T1PSZYJUE16HIZUF", "Workload type: Transaction Processing", and "Dedicated infrastructure: No".

<https://docs.oracle.com/en/cloud/paas/digital-assistant/tutorial-sql-dialogs/index.html>



ORACLE