Guidelines for Enabling Oracle Analytics Assistant for Consumer Users on Workbooks

Describes best practices for Oracle Analytics Cloud administrators when exposing the Oracle Analytics Assistant to their consumer users.

September 2025, version 1.0

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. 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, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Revision History

The following revisions have been made to this document since its initial publication.

DATE	REVISION
September 2025	Initial publication

Author: Philippe Lions

Contributing Authors: Lalitha Venkataraman, Jacques Vigeant, Lax Mamillapalli

Table of Contents

1. Overview

This document exposes guidelines about best configuring data indexes for Oracle Analytics Assistant to provide the most appropriate responses to consumer user prompts.

Consumer users prompt Oracle Analytics Assistant directly from the workbook context, without any details on underlying datasets sourcing the workbook. Consumers aren't expected to have any knowledge about the scope of objects included in the datasets. They ask functional questions about the exact domain of the workbook and expect pertinent answers from Oracle Analytics Assistant in that context.

When workbooks are sourced from narrow data-sources that remain focused on a single functional domain, Oracle Analytics Assistant can directly leverage the same dataset as the workbook and produce sharp, contextual answers. However, when workbooks are sourced from wide data sources that span across multiple specific functional contexts, there is a risk of Oracle Analytics Assistant returning ambiguous responses. For example, in the case of large subject areas with many columns, indexing these numerous columns may lead Oracle Analytics Assistant to produce approximate answers expanding beyond the context of the workbook, as user prompts could be interpreted in different ways given the variety of columns indexed. More specifically, consumer users have an implicit expectation that Oracle Analytics Assistant answers will remain exactly in the context of their workbook domain and have no awareness that the underlying data source is much broader than just this context. Yet that large scope can dramatically influence and degrade the precision that Oracle Analytics Assistant answers will get them.

Therefore, when workbooks are sourced from large/wide datasets, a best practice is to create an alternate reduced source, referred to as **Dataset-Agent**, only to feed Oracle Analytics Assistant, and whose columns focus only on the specific workbook functional domain. The purpose of these Dataset-Agents is solely to feed Oracle Analytics Assistant for the functional context of a given workbook or set of workbooks. They can be attached to workbooks as specific sources only for feeding Oracle Analytics Assistant usage there.

This document lists various guidelines to help define these focused data-sources (Dataset-Agents).

- **Prerequisite**: identify the workbook or set of focused workbooks around a specific functional domain on which Oracle Analytics Assistant is enabled for consumer users.
- **Version**: This is V25-09-1 of the document, highlighting guidance and limitations based on functionalities available in **September 2025 Oracle Analytics Cloud update**. As the product constantly evolves and enrich, expect regular updates and resolutions to this content.

2. Defining Dataset-Agents

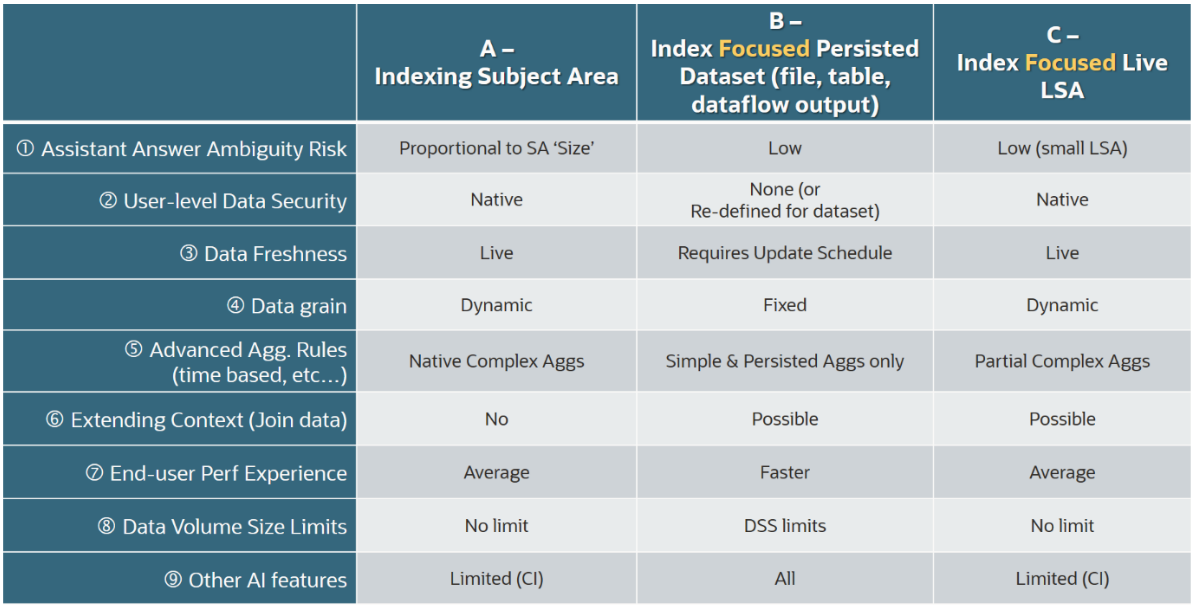
2.1. What Type of Datasets Can Be Used as a Dataset-Agent?

Dataset-Agents can be made of various types of static persisted datasets:

1. File-based or table-based datasets are the most robust recommendation when original subject areas are too large. In this case, extract data from your subject areas into a persisted dataset to use for the Oracle Analytics Assistant.

2. In some cases, authors may also leverage local subject areas as alternative focused sources, but these bring extra functional constraints as of September 2025. If a persisted dataset is an option (file or table), it is currently preferred over a local subject area. Note that several product developments are going on with Oracle Analytics that will rapidly mitigate these limitations in upcoming releases.

The following table lists various types of data-sources that can be used for a Dataset-Agents, as of September 2025.



	A – Indexing Subject Area	B – Index Focused Persisted Dataset (file, table, dataflow output)	C – Index Focused Live LSA
① Assistant Answer Ambiguity Risk	Proportional to SA 'Size'	Low	Low (small LSA)
② User-level Data Security	Native	None (or Re-defined for dataset)	Native
③ Data Freshness	Live	Requires Update Schedule	Live
④ Data grain	Dynamic	Fixed	Dynamic
⑤ Advanced Agg. Rules (time based, etc…)	Native Complex Aggs	Simple & Persisted Aggs only	Partial Complex Aggs
⑥ Extending Context (Join data)	No	Possible	Possible
⑦ End-user Perf Experience	Average	Faster	Average
⑧ Data Volume Size Limits	No limit	DSS limits	No limit
⑨ Other AI features	Limited (CI)	All	Limited (CI)

2.2. Which Columns to Add Into a Dataset-Agent

Create your Dataset-Agent by selecting columns from the sources used in the selected workbooks.

1. **Include a limited number of columns in Dataset-Agents.** Only include columns used or needed for the functional context of the workbook(s) in the Dataset-Agent. Even if it's not explicitly used in the workbook, a data column close to the context of the workbook can be a candidate for the Dataset-Agent, if the following conditions are met:

   a) Consumer users are expected to potentially ask a question or prompt involving these.

   b) Adding it to the Dataset-Agent won't introduce potential ambiguity with other existing columns already in the Dataset-Agent.

2. **Avoid high-cardinality columns in Dataset-Agent**. Don't bring *Order_id*, or *Customer name*, which potentially carry many distinct members (>1k). High cardinality columns typically increase the indexing time and may eventually not be of a critical need in answering many of the 'high level' consumer user prompts.

3. **Considerations about dates.**

a) Avoid multiple functional dates in a single Dataset-Agent.

1. When possible, only include one functional date (set of date hierarchy) in the Dataset-Agent, as opposed to multiple functional date hierarchies. For example, **order-date**, **shipped-date**, and **billed-date** are all distinct date hierarchies part of a subject area about sales. Ideally, only include one of these hierarchies in your Dataset-Agent. Having multiple date hierarchies in one Dataset-Agent will result in risk of ambiguity with Oracle Analytics Assistant responses. For example, "*Show me opportunity amount over the past two quarters*." Does "*past two quarters*" refer to the past two **billed date** quarters, or past two **ordered date** quarters? This ambiguity can be reduced by clearly marking names of date objects, but it's likely that questions like the above may not result in the date that a consumer user expects when prompting the initial question, as no **date type** is provided.

2. If you must give consumer user access to multiple dates in a single Dataset-Agent, require that their prompts always specify which **functional date type** they refer to when they ask time-related questions. This may not always intuitive for consumer users. For example, when **order date** and **ship date** objects are both included in the Dataset-Agent.

b) Custom (non-Gregorian) calendars (fiscal calendar): If your workbook source is leveraging custom calendars where a fiscal year doesn't match Gregorian format, January 01 to December 31, (for example, Oracle FY runs from 6/01 to 5/31), then you need to cautiously include time level objects in your Dataset-Agent for indexing. In this use-case, simply adding the **date** datatype column to the Dataset-Agent won't allow Oracle Analytics Assistant to always return the correct response. Oracle Analytics Assistant will automatically aggregate time according to Gregorian calendar and will consider a full year as January 01 to December 31. To get correct time aggregation results, you need to remove the **date** datatype column from your Dataset-Agent, and add only the time hierarchy level columns (**month, quarter, year**) that are calculated for your custom calendar. For example, in the case of the Oracle Fiscal Calendar, quarter Q1 includes the months of June, July, and August.

4. **Properly set Dataset-Agent metrics aggregation rules**. As you compose your Dataset-Agent (file/table-based dataset or local subject areas), metric type columns default with a Sum aggregation rule configured. That means that Sum will be applied to the metric for every aggregation operation in a query.

   a) In general, you need to consider if this is the correct aggregation rule for each metric in your Dataset-Agent. For example, if a metric isn't additive, like **Age** for example, you need to override the aggregation rule to the Avg. Like in any other dataset, it's required to manually override the aggregation-rule that defaults in the Dataset-Agent, where needed for a metric to properly operate. At a high level, typical objects that need aggregation-rules override are **Count, Count Distinct,** and **Averages**. Metrics that are ratio calculations with a numerator and a denominator need to be overridden with an **Average** aggregation rule in the Dataset-Agent.

   b) The most efficient approach to confirm your Dataset-Agent metric needs to have its aggregation rule overridden is to manually test it. Manually build queries using Oracle Analytics Self-Service (no need to use Assistant for that test) combining this Dataset-Agent metric with other attributes objects from the same Dataset-Agent. Make sure you set totals to **tables of Pivot tables** (both rows and columns), then visually confirm that the query results are correct.

5. **Column name and description considerations.** This is a critical step that will impact the accuracy of results. Rename the column to something meaningful. Renaming a column is more effective than just

adding synonyms. While it is encouraged to also add synonyms, column names have a higher impact on the LLM resolution than synonyms. Column descriptions however only serve as a tooltip in the data panel. They aren't currently used by the LLM and don't surface in the Oracle Analytics Assistant panel.

6. **Enrich Dataset-Agent columns with calculations and groupings.**

    a) Add functional calculations in the Dataset-Agent. Add commonly used calculations to your Dataset-Agents if they don't already exist. For example, ratios. Note that Oracle Analytics Assistant can create simple calculations at runtime using public domain knowledge if they don't already exist in the dataset.

    b) Extend your data with groupings.

        1. Convert numerical values into easy-to-understand categories. For example: silver, gold, platinum customers (**total sales**), small, medium, large orders (**order amount**), high/low-performing employees (**performance rating**).

        2. Create groupings. Use data preparation tools to create quick-access groups for Oracle Analytics Assistant queries.

    c) Enrich data with Knowledge Repository. Use smart suggestions, such as linking countries to continents or drugs to categories.

    d) Null Values. Ensure that voice queries handle null values by converting them into descriptive terms like "Unknown" or "Not Available".

7. **Extend Dataset-Agent with multiple external sources.** A single Dataset-Agent can join multiple sources in a multi-table data model. The Dataset-Agent data model can include additional context friendly sources, properly joined in the Dataset-Agent data model (Oracle Analytics Cloud multi-table data model definition). Oracle Analytics Assistant Indexing can then run on the extended dataset and Oracle Analytics Assistant include the external data as part of its answers.

8. **Assign default map layers to geo-related columns in your Dataset-Agent**. Some data columns may require to default the rendering visualization as a map, for example, a column named **Warehouse Location**, or **Seat Location** may be meant to be visualized by default as a map. The Oracle Analytics Data-preparation experience allows Authors to assign a default map layer to a dataset column (not a subject area column), by right-clicking on the column and selecting **Location Details**. This allows Authors to assign the proper data layer to the object, avoiding confusion with other possible layers, and most importantly defaults the visualization to be a map when the object is used in Oracle Analytics Assistant. This results in a better experience and better insights for consumer users using Oracle Analytics Assistant.

2.3. Additional Guidance Specific to Using Local Subject Areas in Your Dataset-Agent

Using local subject areas includes additional restrictions and guidelines when used as Dataset-Agents. Note that several developments are going-on with Oracle Analytics that will resolve these limitations over upcoming releases.

1. **Local subject areas data access type.** Set local subject areas used in Dataset-Agent to live data access mode. That allows user data security criteria to pass into the query.

2. **Avoid subject area columns with complex aggregation rules in your local subject areas.** Try to use simple additive metrics in your local subject areas. Subject areas may include various metric types, such as:

    a) **Local subject area - friendly metrics.**

1. Plain and simple additive metrics (with an aggregation rule = **Sum** or **Count**), these objects will be a robust contribution in local subject areas.

2. Logical semantic calculations are calculated objects computed directly in the semantic layer using other semantic objects as their calculation arguments. For example, profit ratio is defined as logical calculation dividing subject area column **Profit** by subject area column **Sales**. These objects are friendly to local subject areas.

3. Note that even local subject area-friendly metrics may require the local subject area designer to override their aggregation rules (with **Avg** aggregation rule), see step 3 “What do I do?” in the **Local subject area - non friendly metrics** section below.

b) **Local subject area - non friendly metrics.**

1. Advanced complex aggregated metrics like Dim-based or Time-based aggregated metrics. **Headcount** value for example, adds up over the **cost centers** dimension, but doesn’t cumulate up over the **time** dimension. Subject areas natively support the combination of multiple distinct aggregation-rules in a single metric. With a subject area, proper aggregation will dynamically and respectively apply according to which specific dimensions are invoked in the query. But when consumed over local subject areas, there is a critical risk that the query context doesn’t translate properly from the local subject areas down to the subject area. As a result, there is a real possibility that some of the results are not correct. Particularly in the case of prompts requiring trending reports. If you’re not sure if a metric has a complex aggregation, ask the semantic layer Administrator to check it in the modeling tool to confirm.

2. Metrics that have different aggregation rules depending on which logical table source is used. Let's illustrate this with an example. For some measures, when there are multiple physical sources that can honor a query, the Oracle Analytics Semantic layer server can identify the most efficient source to answer a given query. Let's say user asks for the number of **Service Requests** by quarter in the year 2025. The Oracle Analytics semantic layer server will transparently identify that this query may be best serviced by a physical source that holds aggregation of the number of service requests over time, by month. In this case it will add up monthly service request numbers to reach quarterly numbers with very high query performance. However, if the user requests the number of Service Requests by quarters in year 2025, for employee ABC, then the underlying server may not have a prebuilt aggregated source to leverage and will have to revert to a very detailed physical table to honor the request. In this case, it will need to run a count distinct of Service Request IDs for employee ABC, by quarters. The two cases above represent two distinct aggregation rules for a single metric, and depending on which physical source they leverage, two distinct logical table sources. There is a slight risk that a local subject area may confuse the context for these metrics, but to a lesser extent than time-based aggregation. It is highly recommended to test multiple combinations of queries that your local subject areas allow and validate results first.

3. What to do? When possible, avoid objects of Advanced complex aggregated metrics (see **Advanced complex aggregated metrics** above). If you must include a time-based metric in your local subject area, proceed with a manual testing of what may and may not work in your local subject area with the chosen metric. Manually build queries Oracle Analytics Self-Service. Combine this local subject area metric with attributes and filters using objects you have included in your local subject area then compare if the

query results in the local subject area exactly match the results of the same query built on the subject area directly. There is a probability that combination of dimension you have added to your local subject area may remain in a **friendly agg** context for the local subject area, and there may not be any issues. But there is a risk that a set of dimension combinations aren't friendly and could distort query results. Once you've identified dimension combinations that could create a problem, you can warn consumer users about properly combining them in their Oracle Analytics Assistant prompts.

3. **Override local subject area metrics aggregation rules.** Like with the general Dataset-Agent guidance, it's required to manually override the default aggregation rule that defaults in certain of your local subject area metrics. See **Which columns to add into a Dataset-Agent** section above. This recommendation applies to some of the **Local subject area - friendly metrics** columns defined in the previous section but doesn't help with the issue of **Local subject area – non friendly metrics** seen above. Just like in **Section 2.2, Which columns to add into a Dataset-Agent** above, some default aggregation rules need to be overridden with the right rule when the underlying objects aren't additive measures in the semantic layer. In the case of local subject areas particularly, the most efficient approach to check if your local subject area metric needs aggregation-rule overridden is to manually compare your local subject area result with subject area result. Manually build queries using Oracle Analytics Self-Service, combining this local subject area metric with other attributes columns from your local subject area. Make sure you set totals to **tables of Pivot tables** (both rows and columns), then compare the query results from the local subject area query with results of the exact same query using subject area objects directly. Both must exactly match, totals included. If you are using a local subject area-friendly metric and totals don't match, you need to override the default aggregation rule in the local subject area definition. At a high level, typical override guidelines are:

   a) Count & Count Distinct. Metrics with direct physical mapping (not semantic calculations) and aggregation rule of **Count** or **Count Distinct** in the semantic layer need to have a local subject area aggregation rule of **Count** or **Count Distinct**.

   b) Averages**.** Metrics with direct physical mapping and aggregation rule of **Average** in the semantic layer need to have a local subject area aggregation rule of **Average.**

   c) Ratios Logical Calcs. Metrics that are semantic calculations as ratios, with a numerator and a denominator, need to be overridden with a local subject area aggregation rule in the local subject area. Note that in this case, total lines of local subject area tables will not exactly match subject area **Avg** calculations as the local subject area may not accommodate all the weighting in the **Avg,** but the granular lines will show the correct value and total lines will approach correctness.

4. **What date level object to include from your subject area into your local subject area?**

   a) For any given date hierarchy, subject areas typically provide many level objects: a **date data type** object (usually labelled as day or date), along with a variety of **date levels** (month, week, quarter, year). Date levels are typically of data type **text**, not **date data type**. Only the day level is **date data type**. By default, only adding the **date level** object in your local subject area is sufficient for Oracle Analytics Assistant to natively compute various levels on the fly. This will simplify the interpretation of prompts by the LLM and the definition of unambiguous response. There is no need to bring multiple levels in the local subject area along with the date column itself. An exception to this guidance relates to custom calendars (fiscal calendars), see **Local subject area - non friendly metrics** on that topic above.

3. Configuring Indexing and Enabling Workbook Consumer Assistant

3.1. Configuring Oracle Analytics Assistant Indexing on Your Dataset-Agent Data

1. **Index type.** Index all columns in the Dataset-Agent. There are no use cases for including a column in a Dataset-Agent and not indexing it.

   a) Index attributes columns for **name** and **values.**

   b) Index metrics and dates for **name.**

2. **Add synonyms**.

   a) Add up to 20 synonyms for each column. As you add synonyms, make sure they don't cause ambiguity or confusion with existing columns or other synonyms with similar names within a Dataset-Agent. Be judicious about what synonyms you add, as adding multiple well-known terms may not always lead to the best results. Use synonyms to provide additional knowledge to help Oracle Analytics Assistant answer a question. Good examples to call out are domain-specific acronyms or abbreviations that may not be common knowledge to Oracle Analytics Assistant. Clarify technical metadata for the LLM, or help LLMs identify defaults (e.g., Default Date).

   b) Some words, like **Year**, **Month**, **Date,** are reserved words, and aren't allowed as synonyms.

3. **Refreshing indexes.** Indexing schedule can be set to refresh when dataset is refreshed or can be scheduled.

   a) When you are still building or editing your local subject area, leave the default refresh setting of **When dataset is refreshed**. By doing this, as you're editing your local subject area by adding or modifying columns, indexes are refreshed, and you can test Oracle Analytics Assistant with the ongoing changes.

   b) Once you are done editing the local subject area, change the setting to a predefined schedule that ideally matches with the schedule of data refresh of your underlying data warehouse.

3.2. Configuring Oracle Analytics Assistant on Your Workbooks

1. **Add Dataset-Agent source to your workbook** (when needed). If you have designed a Dataset-Agent for your consumer user's Oracle Analytics Assistant experience, then you need to add this Dataset-Agent into the workbook you want it to operate for.

   a) Extend a new data source in the workbook but don't mash-it-up with any other existing data sources already in the workbook. Leave the Dataset-Agent un-joined in the workbook with any other dataset or subject area. The purpose of the Dataset-Agent in the workbook is solely to support Oracle Analytics Assistant requests by consumer users.

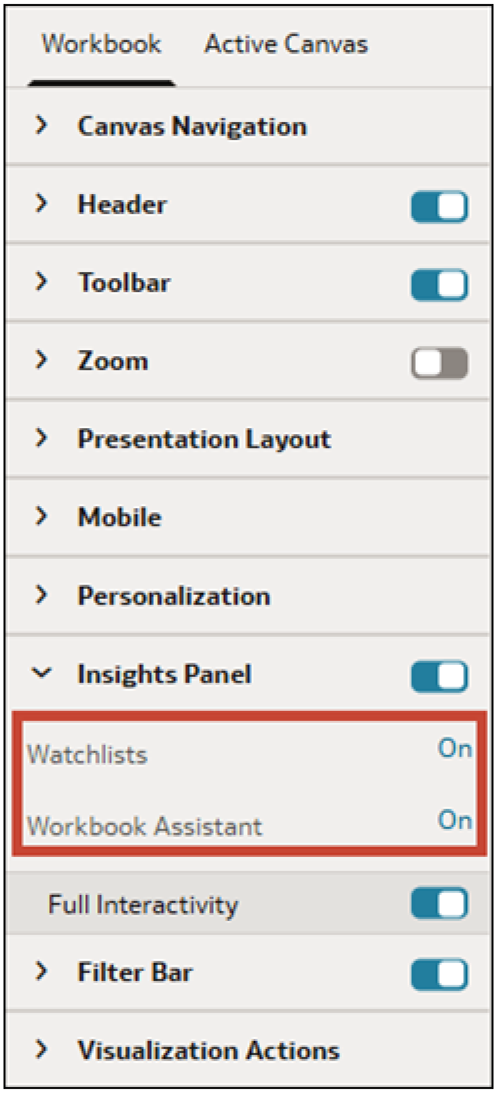
   b) Only have a single Dataset-Agent per workbook. Using more than one Dataset-Agent on a workbook will require a disambiguation by the consumer user as to which one each prompt refers to.

2. **Activate Consumer Assistant for your workbook present mode**. This action will allow you to show the Assistant Panel for this workbook when viewed by consumer users. Navigate to the **Present properties** tab of your workbook and turn on the **Insights Panel** property. This activates the Oracle Analytics Assistant and allows your users to access **Workbook Assistant** and **Watchlists** directly on their workbook. Then select which tabs to display to consumer users. The options are:

a) **Watchlists** - allows consumer users to see and track saved metrics.

b) **Workbook Assistant** - enables Oracle Analytics Assistant queries on the workbook's dataset.

Oracle recommends displaying both for a full experience.



Workbook | Active Canvas

- Canvas Navigation
- Header
- Toolbar
- Zoom
- Presentation Layout
- Mobile
- Personalization
- Insights Panel
  - Watchlists: On
  - Workbook Assistant: On
  - Full Interactivity
- Filter Bar
- Visualization Actions

3. **Enable or disable datasets that will be used by Oracle Analytics Assistant.** Just below on **Present Properties** tab, you can also see details of all the data-sources in your workbook and flag which ones will be used by Oracle Analytics Assistant and which ones won't be. This allows you to specify that only the Dataset-Agent will be sourcing Oracle Analytics Assistant for user prompt interactions from this workbook.

3.3. Set Up Permissions for Consumer Users to Access Oracle Analytics Assistant

A specific granular permission item allows administrators to grant access to Oracle Analytics Assistant only to specific consumer users. Hence, the appropriate Oracle AI Assistant permission needs to be granted to the group of users before they can leverage this capability. To grant proper permissions:

1. **Create a Custom Role.** In Oracle Analytics, user access is controlled through application roles. These can be **Vanilla/Predefined** — with fixed permissions like *Consumer* or *DV Content Author*, or they can be **Custom** — created by Administrators to tailor exactly which features users can access. To make the

Oracle Analytics Assistant available to consumer users, administrators need to create a custom role that includes the **Use Assistant in Workbooks** permission.

Go to **Console** → **Roles and Permissions** → **Application Roles**, then click **Create Application Role.**

2. **Add Privileges to the role**. In the newly created role, open the **Memberships** tab and add the *DV Consumer* 'vanilla role' so the new role inherits all standard consumer user capabilities. Then go to the **Permissions** tab and add the permission **Use Assistant in Workbooks.**
3. **Assign Users or Groups to the new role**. In the **Members** tab of the new role, assign individual users or entire groups. This grants them the ability to use the Oracle Analytics Assistant.

4. Assistant Consumer Prompting Guidelines

To maximize the effectiveness of your interactions with the Oracle Analytics Assistant, please follow the guidelines outlined below. These are intended to help consumer users who may not have any direct knowledge of the underlying data sets formulate questions based on business context. Even after all the above update users need to still follow general prompting guidelines.

1. **Structure and Paraphrasing**

   a) If the Oracle Analytics Assistant doesn't return a satisfactory or relevant answer:

      - **Try rephrasing** the question using simpler or more explicit language.
      - **Avoid overly complex** or nested queries. Break them into multiple smaller questions if needed.

   b) For example, instead of:
   "Which top 3 regions had declining profit last quarter but improved this quarter in electronics?"
   Try:

   "What was the profit trend last quarter for electronics by region?"

   Followed by:

   "How did it change this quarter?"

2. When referring to **column names or field values**, pause briefly (1–2 seconds) while typing.

   a) The Oracle Analytics Assistant displays auto-suggestions based on the active dataset fields.

   b) Selecting from the suggestions ensures alignment with the actual metadata, improving result accuracy.

   c) For example:

   Start typing *"customer"* and wait for *customer_segment, customer_region* to appear, then select the appropriate one.

3. **Understand contextual carry-forward behavior.**

   a) The Oracle Analytics Assistant **remembers the context** of previous questions and answers within the same conversation.

   b) Applied filters (e.g., for **region**, **product line**, **time period**) persist across follow-up questions, unless explicitly changed.

   c) A consumer user may use the following prompts, and the context is carried forward for prior quarter data. For example:

      - Q1: *"Show sales trend for last 5 years."*
      - Q2: *"Show only trend for Technology Products."*
      - Q3: *"Break it down by product subcategory."*
      - A user may then want to start a new fresh set of prompts. To do so, click **"Clear Assistant History"** option at the top to reset the context and begin a new analysis path.

4. **Functional focus of queries.**

   a) Consumer users are expected to ask functional, goal-oriented questions relevant to business operations, such as:

      - *"Who are the top performing Sales Reps this month?"*

- *“Which department has the highest attrition in last 6 months?”*

b) Don’t ask about:

- The structure, schema, or internal logic of the dataset.
- Technical metadata or data lineage.

c) The Oracle Analytics Assistant operates within a black-box abstraction over datasets and is focused purely on insight delivery from the workbook’s perspective.

5. Limitations & Workarounds

1. **Filters not being passed automatically into prompts**. Usually, workbooks are exposed to consumer users with a set of pre-built data filters that apply to the visualizations on each canvas. These data filters are designed by Authors and may consist of Canvas filters, Dashboard filters, or even user-based data-security filters. As consumer users interact with live prompts on Oracle Analytics Assistant, they likely expect that these filters are automatically passed into their prompts. For example, a consumer user is watching a dashboard on sales, and let's suppose that top filters for his canvas default the visualizations to a given organization “XYZ” and a given region “ABC” he is responsible for.

   a) **Dashboard/Canvas filters.** As this user asks questions to Oracle Analytics Assistant, the dashboard or canvas filters won’t automatically be sent to their prompts. They may have to explicitly specify the value of the filters in the prompt, such as: “*For organization XYZ and for region ABC*”. Automatically transposing the canvas and dashboard filters into the prompt is under development for the Oracle Analytics Assistant, but not yet part of the product.

   b) **User level data security filters.** These are honored by Oracle Analytics Assistant as they are part of the Oracle Analytics Query server generation attached to the user for the Dataset-Agent (either dataset or local subject area).

2. **Fiscal calendars or custom calendars.** If your Dataset-Agent source is leveraging custom calendars where a fiscal year doesn’t match Gregorian format from January 01 to December 31, (for example the Oracle fiscal year ends 5/31 and starts 6/01) then you need to warn consumer users to cautiously phrase their prompts so that Oracle Analytics Assistant picks the correct custom time columns that are already be part of Dataset-Agent for indexing. In the case of a custom calendar, asking plain English questions to Oracle Analytics Assistant such as “*show me sales trended over three quarters*”, or “*compare the past two years*”, could either return no response or may return a wrong one. There is a risk that Oracle Analytics Assistant may automatically aggregate time according to the Gregorian calendar and will consider a full year as January 01 to December 31. To get correct time aggregation results, you need to remove the **Date** datatype column from your Dataset-Agent and add only the time hierarchy levels columns (**month**, **quarter**, **year)** that are calculated for your custom calendar. For example, in the case of Oracle Fiscal Calendar, the quarter Q1 will roll up months of June, July, August. Refer to the Guidelines below to configure your Dataset-Agent according to this use in case you are using a custom calendar in your data.

3. **Oracle Analytics Assistant is picking up the wrong columns for filtering**. When Oracle Analytics Assistant prompts include filter values, for example, “*Show me sales for Region XYZ*”, Oracle Analytics Assistant will interpret both the column the prompt is about and what value is selected. If a user leaves the prompt like this, there is a risk that the exact value in the prompt may not match the actual data, or that some ambiguity may arise on which column name to pick. This may lead to wrong results delivered by Oracle Analytics Assistant. To mitigate this issue, leverage **Auto_Select** while typing prompts. Pause briefly (1–2 seconds) while typing the filter part of the prompt and let Oracle Analytics Assistant display auto-suggestions based on the active dataset field. The user can pick both the column and the exact filtered member values for the prompt. This removes all risk of ambiguity in results being returned by Oracle Analytics Assistant.

4. **Oracle Analytics Assistant results aren't properly aggregating 'count' / 'Avg' metrics**. Results returned by Oracle Analytics Assistant will aggregate measures according to the aggregation rule specified in the Dataset-Agent. The default rule in datasets or local subject areas is the sum aggregation rule. As stated in the **Properly set Dataset-Agent Metrics Aggregation rules** step above**,** it's important that Dataset-Agent is configured with the proper aggregation rule for each metric it includes. This is particularly true when Dataset-Agents are built as local subject areas.

5. **Oracle Analytics Assistant not properly leveraging complex aggregations metrics.** In the case of local subject areas, as mentioned above in the **Avoid subject area columns with complex aggregation rules in your local subject areas** step, some metrics from the underlying subject area may be configured as complex aggregation rules. For example, **Time based aggregated measures** or **Multi aggregation-rule measures**. The metrics may not properly work in all combinations of attributes of the local subject area. The main way to mitigate this limitation is to test where these metrics fail in your local subject area and edit the local subject area configuration to only contain attributes/dimensions that won't create these issues.

6. **Oracle Analytics Assistant duplicating calculations that may exist in Dataset-Agent**. There are cases where a calculation may already be defined in your Dataset-Agent, yet some prompts may lead Oracle Analytics Assistant to re-create its own version of the calculation. For example, **profit rate** exists in the Dataset-Agent, but a prompt asking for **profit ratio %** may lead Oracle Analytics Assistant to re-compute another calculation instead of leveraging the existing one. To reduce the risk of this happening, add synonyms to the existing column in the Indexing configuration so the Oracle Analytics Assistant can properly recognize this column in prompts.

7. **Calculation challenges for Oracle Analytics Assistant**. Oracle Analytics Assistant may currently not 'crack' all types of calculations that a user prompt may imply. Some questions may seem simple, but if a proper column isn't designed in the Dataset-Agent, Oracle Analytics Assistant will try to generate a calculation, and may encounter some limits. Oracle is working on enhancing this capability, but current limits are:

   a) *% of total*: users may ask to show sales by region, expressed as % of total sales. If that calculation does not already exist in the Dataset-Agent, it is likely that Oracle Analytics Assistant may not yet be able to resolve it

   b) *Members above avg*: users may ask to highlight only records that are above average for the results-set of their request. For example, "*Show products sold in France, highlight the ones performing above average sales for that selection*". Oracle Analytics Assistant may not yet be able to resolve this calculation.

   c) Metric range filtering: Oracle Analytics Assistant may also not properly render metrics range filtering such as: "*show me customer where sales are between 1000 and 5000".* This prompt may return correct results when the context of the query is simple but may also run into trouble when the query grain/filtering context is more complex.

8. Oracle Analytics Assistant may sometimes say "*There is no data to answer your question*" in a valid context and despite picking the right columns in the question.

Connect with us

Call **+1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle