Oracle® Cloud

Using the PostgreSQL Adapter with Oracle Integration 3

F71142-11
November 2025

Oracle Cloud Using the PostgreSQL Adapter with Oracle Integration 3,

F71142-11

Primary Author: Oracle Corporation

Contents

About This Content

This guide describes how to configure this adapter as a connection in an integration in Oracle Integration.

Audience

This guide is intended for developers who want to use this adapter in integrations in Oracle Integration.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Resources

See these Oracle resources:

- Oracle Cloud at `http://cloud.oracle.com`
- *Using Integrations in Oracle Integration 3*
- *Using the Oracle Mapper with Oracle Integration 3*
- Oracle Integration documentation on the Oracle Help Center.

Conventions

The following text conventions are used in this document.

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Understand the PostgreSQL Adapter

Review the following conceptual topics to learn about the PostgreSQL Adapter and how to use it as a connection in integrations in Oracle Integration. A typical workflow of adapter and integration tasks is also provided.

Topics:

> **Note**
>
> The PostgreSQL Adapter is only available in Oracle Integration 3.

PostgreSQL Adapter Capabilities

The PostgreSQL Adapter enables you to integrate a PostgreSQL cloud database with Oracle Integration. In addition, the PostgreSQL Adapter enables you to integrate a PostgreSQL database residing behind the firewall of your on-premises environment with Oracle Integration through use of the on-premises connectivity agent. You can configure the PostgreSQL Adapter as a trigger or an invoke connection in an integration in Oracle Integration.

The PostgreSQL Adapter provides the following benefits:

For trigger endpoints:

- Support for polling new and updated records for processing in the PostgreSQL Adapter without use of the connectivity agent. See Perform Inbound Polling Without the Connectivity Agent.
- Supports polling new and updated records for processing in the PostgreSQL on-premises database with the connectivity agent. The PostgreSQL Adapter supports distributed polling. Distributed polling provides high availability and improves performance.
- Supports a logical delete polling strategy. This strategy involves updating a special field on each row once it is processed.
- Supports processing message payloads up to 10 MB in size. In the case of polling, you must set the **Rejected Value** property to **REJECTED** on the Polling Strategy and Options page. If the incoming message is greater than 10 MB, that particular record is updated to **REJECTED** instead of **READ**. If the message payload is greater than 10 MB, a fault response is sent to the calling clients.
- Supports connectivity to a PostgreSQL cloud/on-premise database over SSL through the connectivity agent.

For invoke endpoints:

- Supports bulk data import by selecting the **Perform Bulk Data Import Operation** on the Basic Info page of the Adapter Endpoint Configuration Wizard. See Import Data Files Using the Bulk Data Import Operation and Import Transactional Data Using the Bulk Data Import Operation.
- Supports invocation of stored procedures in the PostgreSQL database.
- Supports execution of DML statements such as `Select`, `Insert`, `Update`, and `Delete` using the **Run a SQL Statement** option.

  Select **Run a SQL Statement** on the Basic Info page of the Adapter Endpoint Configuration Wizard to execute simple SQL queries.
- Supports performing the `Select`, `Insert`, `Update`, and `Merge` operations against database tables.

  For complex SQL queries, use stored procedures by selecting the **Invoke a Stored Procedure** option on the Basic Info page of the Adapter Endpoint Configuration Wizard. Stored procedures can reduce the complexity of a SQL query.
- Supports pagination. You can implement pagination when fetching a large number of records for a Select query and receive sorted data in chunks. See Use Pagination in an Integration.
- Supports connectivity to a PostgreSQL cloud/on-premises database over SSL through the connectivity agent.

The PostgreSQL Adapter is one of many predefined adapters included with Oracle Integration.

Supported Data Types for the Stored Procedure and Operation On Table Operations

The PostgreSQL Adapter supports the following data types for stored procedure and Operation On Table operations.

- int
- int2
- int4
- int8
- integer
- bigInt
- smallint
- numeric
- double precision
- real
- float
- float4
- float8
- money
- decimal
- oid

- varchar
- char
- text
- bpchar
- name
- date
- timestamp
- timestamptz
- time
- timetz
- boolean

What Application Version Is Supported?

For information about which application version is supported by this adapter, see the Connectivity Certification Matrix.

Workflow to Create and Add a PostgreSQL Adapter Connection to an Integration

You follow a very simple workflow to create a connection with an adapter and include the connection in an integration in Oracle Integration.

This table lists the workflow steps for both adapter tasks and overall integration tasks, and provides links to instructions for each step.

Step	Description	More Information
1	Decide where to work	• Work in a project (see why working with projects is preferred in *Using Integrations in Oracle Integration 3*). • Work outside a project.
2	Create the adapter connections for the applications you want to integrate. The connections can be reused in multiple integrations and are typically created by the administrator.	Create a PostgreSQL Adapter Connection
3	Create the integration. When you do this, you add trigger (source) and invoke (target) connections to the integration.	Understand Integration Creation and Best Practices in *Using Integrations in Oracle Integration 3* and Add the PostgreSQL Adapter Connection to an Integration
4	Map data between the trigger connection data structure and the invoke connection data structure.	Map Data in *Using Integrations in Oracle Integration 3*
5	(Optional) Create lookups that map the different values used by those applications to identify the same type of object (such as gender codes or country codes).	Manage Lookups in *Using Integrations in Oracle Integration 3*

Step	Description	More Information
6	Activate the integration.	Activate an Integration in *Using Integrations in Oracle Integration 3*
7	Monitor the integration on the dashboard.	Monitor Integrations During Runtime in *Using Integrations in Oracle Integration 3*
8	Track payload fields in messages during runtime.	Assign Business Identifiers for Tracking Fields in Messages and Track Integration Instances in *Using Integrations in Oracle Integration 3*
9	Manage errors at the integration level, connection level, or specific integration instance level.	Manage Errors in *Using Integrations in Oracle Integration 3*

2

Create a PostgreSQL Adapter Connection

A connection is based on an adapter. You define connections to the specific cloud applications that you want to integrate. The following topics describe how to define connections.

Topics:

Prerequisites for Creating a Connection

You must satisfy the following prerequisites to create a connection with the PostgreSQL Adapter:

PostgreSQL Database Prerequisites

- Ensure that you have write permissions on the database.
- Ensure that you have the required permissions to run stored procedures and packages against the PostgreSQL database.
- Know the database hostname or IP address and the port number.
- Know the database name.
- Know the user name and password for connecting to the database.
- Ensure that the same database name, which you configured on the Connections page, is selected on the Invoke a Stored Procedure page in Oracle Integration. See No Package/ Procedure Found.
- The connectivity agent is required to connect Oracle Integration with a PostgreSQL on-premises database. See Configure an Agent Group.
- Allowlist Oracle Integration instance public IP addresses in the security group of the PostgreSQL cloud instance.

Bulk Data Import Operation Prerequisites

Before using the bulk data import operation, ensure that file header names match with target table fields and the order of fields is the same.

SSL Prerequisites

Before connecting Oracle Integration with a PostgreSQL cloud/on-premises database over Secure Sockets Layer (SSL), ensure that a connectivity agent is configured. See Configure the Endpoint Access Type.

OCI Managed Database Prerequisites

1. Install the connectivity agent within the same Virtual Cloud Network (VCN) as the PostgreSQL OCI instance.
2. Establish a connection to the PostgreSQL OCI instance using its private IP address from Oracle Integration through the configured connectivity agent.

Create a Connection

Before you can build an integration, you must create the connections to the applications with which you want to share data.

> **Note**
>
> You can also create a connection in the integration canvas. See Define Inbound Triggers, Outbound Invokes, and Actions.

To create a connection in Oracle Integration:

1. Decide where to start:
   - Work in a project (see why working with projects is preferred).
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations**.
     d. In the **Connections** section, click **Add** if no connections currently exist or **+** if connections already exist. The Create connection panel opens.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Connections**.
     b. Click **Create**. The Create connection panel opens.
2. Select the adapter to use for this connection. To find the adapter, scroll through the list, or enter a partial or full name in the **Search** field.
3. Enter the information that describes this connection.

Element	Description
Name	Enter a meaningful name to help others find your connection when they begin to create their own integrations.
Identifier	Automatically displays the name in capital letters that you entered in the **Name** field. If you modify the identifier name, don't include blank spaces (for example, `SALES_OPPORTUNITY`).

Element	Description
Role	Select the role (direction) in which to use this connection. **Note**: *Only* the roles supported by the adapter you selected are displayed for selection. Some adapters support all role combinations (trigger, invoke, or trigger and invoke). Other adapters support fewer role combinations. When you select a role, only the connection properties and security policies appropriate to that role are displayed on the Connections page. If you select an adapter that supports both invoke and trigger, but select only one of those roles, you'll get an error when you try to drag the adapter into the section you didn't select. For example, assume you configure a connection for the Oracle Service Cloud (RightNow) Adapter as only an **invoke**. Dragging the adapter to a **trigger** section in the integration produces an error.
Keywords	Enter optional keywords (tags). You can search on the connection keywords on the Connections page.
Description	Enter an optional description of the connection.
Share with other projects	**Note**: This field only appears if you are creating a connection in a project. Select to make this connection publicly available in other projects. Connection sharing eliminates the need to create and maintain separate connections in different projects. When you configure an adapter connection in a different project, the **Use a shared connection** field is displayed at the top of the Connections page. If the connection you are configuring matches the same type and role as the publicly available connection, you can select that connection to reference (inherit) its resources. See Add and Share a Connection Across a Project.

4. Click **Create**.

   Your connection is created. You're now ready to configure the connection properties, security policies, and (for some connections) access type.

5. Follow the steps to configure a connection.
   The connection property and connection security values are specific to each adapter. Your connection may also require configuration with an access type such as a private endpoint or an agent group.

6. Test the connection.

Configure Connection Properties

Enter connection information so your application can process requests.

1. Go to the **Properties** section.

2. Enter the host name or IP address of the database server.
3. Enter the database server port number.
4. Enter the database name.

Configure Connection Security

Configure security for your PostgreSQL Adapter connection.

1. Go to the **Security** section.
2. In the **Security Policy** list, select the security policy.
   - **Username Password Token**: Select this security policy to connect to a PostgreSQL cloud/on-premise database.
   - **Username Password Token With SSL Support**: Select this security policy to connect to a PostgreSQL cloud/on-premise database over Secure Sockets Layer (SSL).
3. If you select **Username Password Token**:
   a. In the **Username** field, enter the user name.
   b. In the **Password** field, enter the password.
4. If you select **Username Password Token With SSL Support**:
   a. In the **Choose Instance Type** field, select the instance type:
      - **AWS RDS / Azure Managed DB / OCI Managed DB**: Select to connect to a PostgreSQL cloud database over SSL. If you select this option, a root certificate is required.
        – In the **Root Certificate** field, upload a trusted root certificate. To download the certificate, see Using SSL/TLS to encrypt a connection to a DB instance.
      - **GCP CloudSQL / Others** (On-premise or Certified Public Clouds): Select to connect to a PostgreSQL on-premise database over SSL. If you select this option, the following are required.
        – In the **Root Certificate** field, upload a trusted root certificate.
        – In the **SSL Certificate** field, upload an SSL certificate file. The PostgreSQL administrator can convert an SSL key file to `.der` format using the following command:

          ```
          openssl pkcs8 -topk8 -inform PEM -in ssl-key.pem -outform DER
          -nocrypt -out ssl-key.der
          ```

        – In the **SSL Key** field, upload an SSL key file.

        > **Note**
        >
        > You can contact your PostgreSQL administrator to obtain the root certificate, SSL certificate, and SSL key.

   b. In the **Username** field, enter the user name.
   c. In the **Password** field, enter the password.

> **Note**
>
> - A connectivity agent is required to connect Oracle Integration with a PostgreSQL cloud/on-premises database over SSL. See SSL Prerequisites.
> - The **Root Certificate**, **SSL Certificate**, and **SSL Key** fields are displayed under **Optional security**. This incorrectly implies that these fields are not mandatory. These certificates are used to validate SSL connections.

Configure the Endpoint Access Type

Configure access to your endpoint. Depending on the capabilities of the adapter you are configuring, options may appear to configure access to the public internet, to a private endpoint, or to an on-premises service hosted behind a fire wall.

Select the Endpoint Access Type

1. Go to the **Access type** section.
2. Select the option for accessing your endpoint.

Option	This Option Appears If Your Adapter Supports ...
Public gateway	Connections to endpoints using the public internet.
Connectivity agent	Connections to on-premises endpoints through the connectivity agent. a. Click **Associate agent group**. The Associate agent group panel appears. b. Select the agent group, and click **Use**. To configure an agent group, you must download and install the on-premises connectivity agent. See Download and Run the Connectivity Agent Installer and About Creating Hybrid Integrations Using Oracle Integration in *Using Integrations in Oracle Integration 3*.

Test the Connection

Test your connection to ensure that it's configured successfully.

1. In the page title bar, click **Test**. What happens next depends on whether your adapter connection uses a Web Services Description Language (WSDL) file. Only some adapter connections use WSDLs.

If Your Connection...	Then...
Doesn't use a WSDL	The test starts automatically and validates the inputs you provided for the connection.

If Your Connection...	Then...
Uses a WSDL	A dialog prompts you to select the type of connection testing to perform: • **Validate and Test**: Performs a full validation of the WSDL, including processing of the imported schemas and WSDLs. Complete validation can take several minutes depending on the number of imported schemas and WSDLs. No requests are sent to the operations exposed in the WSDL. • **Test**: Connects to the WSDL URL and performs a syntax check on the WSDL. No requests are sent to the operations exposed in the WSDL.

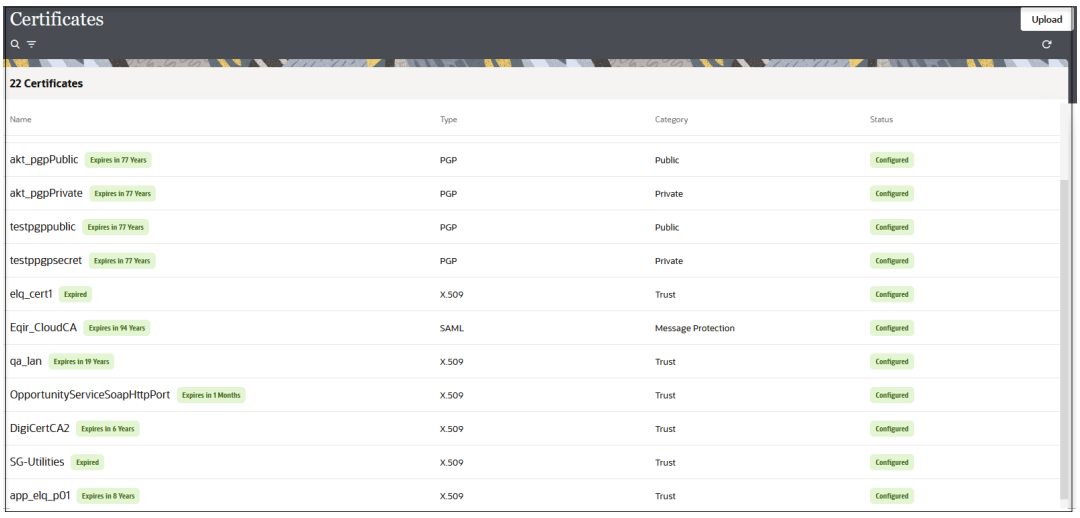
2. Wait for a message about the results of the connection test.
   - If the test was successful, then the connection is configured properly.
   - If the test failed, then edit the configuration details you entered. Check for typos and verify URLs and credentials. Continue to test until the connection is successful.
3. When complete, click **Save**.

Upload a Certificate to Connect with External Services

Certificates allow Oracle Integration to connect with external services. If the external service/endpoint needs a specific certificate, request the certificate and then import it into Oracle Integration.

If you make an SSL connection in which the root certificate does not exist in Oracle Integration, an exception error is thrown. In that case, you must upload the appropriate certificate. A certificate enables Oracle Integration to connect with external services. If the external endpoint requires a specific certificate, request the certificate and then upload it into Oracle Integration.

1. Sign in to Oracle Integration.
2. In the navigation pane, click **Settings**, then **Certificates**.
   All certificates currently uploaded to the trust store are displayed on the Certificates page.
3. Click **Filter** to filter by certificate expiration date, status, and type. Certificates installed by the system cannot be deleted.



4. Click **Upload** at the top of the page.
   The Upload certificate panel is displayed.

5. Enter an alias name and optional description.
6. In the **Type** field, select the certificate type. Each certificate type enables Oracle Integration to connect with external services.
   - Digital Signature
   - X.509 (SSL transport)
   - SAML (Authentication & Authorization)
   - PGP (Encryption & Decryption)
   - Signing key

Digital Signature

The digital signature security type is typically used with adapters created with the Rapid Adapter Builder. See Learn About the Rapid Adapter Builder in Oracle Integration in *Using the Rapid Adapter Builder with Oracle Integration 3*.

1. Click **Browse** to select the digital certificate. The certificate must be an X509Certificate. This certificate provides inbound RSA signature validation. See RSA Signature Validation in *Using the Rapid Adapter Builder with Oracle Integration 3*.
2. Click **Upload**.

X.509 (SSL transport)

1. Select a certificate category.
   a. **Trust**: Use this option to upload a trust certificate.
      i. Click **Browse**, then select the trust file (for example, `.cer` or `.crt`) to upload.
   b. **Identity**: Use this option to upload a certificate for two-way SSL communication.
      i. Click **Browse**, then select the keystore file (`.jks`) to upload.
      ii. Enter the comma-separated list of passwords corresponding to key aliases.

      > **Note**
      >
      > When an identity certificate file (`.jks`) contains more than one private key, all the private keys must have the same password. If the private keys are protected with different passwords, the private keys cannot be extracted from the keystore.

      iii. Enter the password of the keystore being imported.
   c. Click **Upload**.

SAML (Authentication & Authorization)

1. Note that **Message Protection** is automatically selected as the only available certificate category and cannot be deselected. Use this option to upload a keystore certificate with SAML token support. Create, read, update, and delete (CRUD) operations are supported with this type of certificate.
2. Click **Browse**, then select the certificate file (`.cer` or `.crt`) to upload.
3. Click **Upload**.

PGP (Encryption & Decryption)

1. Select a certificate category. Pretty Good Privacy (PGP) provides cryptographic privacy and authentication for communication. PGP is used for signing, encrypting, and decrypting files. You can select the private key to use for encryption or decryption when configuring the stage file action.
   a. **Private**: Uses a private key of the target location to decrypt the file.
      i. Click **Browse**, then select the PGP file to upload.
      ii. Enter the PGP private key password.
   b. **Public**: Uses a public key of the target location to encrypt the file.
      i. Click **Browse**, then select the PGP file to upload.
      ii. In the **ASCII-Armor Encryption Format** field, select **Yes** or **No**.
         - **Yes** shows the format of the encrypted message in ASCII armor. ASCII armor is a binary-to-textual encoding converter. ASCII armor formats encrypted messaging in ASCII. This enables messages to be sent in a standard messaging format. This selection impacts the visibility of message content.
         - **No** causes the message to be sent in binary format.
      iii. From the **Cipher Algorithm** list, select the algorithm to use. Symmetric-key algorithms for cryptography use the same cryptographic keys for both encryption of plain text and decryption of cipher text. The following supported cipher algorithms are FIPS-compliant:
         - AES128
         - AES192
         - AES256
         - TDES
   c. Click **Upload**.

Signing key

A signing key is a secret key used to establish trust between applications. Signing keys are used to sign ID tokens, access tokens, SAML assertions, and more. Using a private signing key, the token is digitally signed and the server verifies the authenticity of the token by using a public signing key. You must upload a signing key to use the OAuth Client Credentials using JWT Client Assertion and OAuth using JWT User Assertion security policies in REST Adapter invoke connections. Only PKCS1- and PKCS8-formatted files are supported.

1. Select **Public** or **Private**.
2. Click **Browse** to upload a key file.
   If you selected **Private**, and the private key is encrypted, a field for entering the private signing key password is displayed after key upload is complete.
3. Enter the private signing key password. If the private signing key is not encrypted, you are not required to enter a password.
4. Click **Upload**.

3

Add the PostgreSQL Adapter Connection to an Integration

When you drag the PostgreSQL Adapter into the trigger or invoke area of an integration, the Adapter Endpoint Configuration Wizard is invoked. This wizard guides you through configuration of the PostgreSQL Adapter as a trigger or invoke in an integration.

The following sections describe the wizard pages that guide you through configuration of the PostgreSQL Adapter as an invoke in an integration.

Topics:

Basic Info Page

Specify a name, description, and operation type on the Basic Info page of each invoke connection in your integration.

Element	Description
What do you want to call your endpoint?	Provide a meaningful name so that others can understand the connection. For example, if you are creating a database connection for adding new employee data, you may want to name it `CreateEmployeeInDB`. You can include English alphabetic characters, numbers, underscores, and dashes in the name. You cannot include the following: • Blank spaces (for example, `My DB Connection`) • Special characters (for example, `#;83&` or `righ(t)now4`) • Multibyte characters
What does this endpoint do?	Enter an optional description of the connection's responsibilities.

Element	Description
What operation do you want to perform? (**Note**: This option is only displayed when you configure the PostgreSQL Adapter as an invoke connection in an integration.)	Select the type of operation for this connection to perform: • **Run a SQL Statement**: Select to run a SQL query against the database. • **Invoke a Stored Procedure**: Select to invoke a stored procedure in the database. • **Perform an Operation On a Table**: Select to perform one of the following operations on a table. – **Insert** – **Update** – **Insert or Update (Merge)** – **Select** • **Perform Bulk Data Import Operation**: Select to import bulk data into the PostgreSQL database.

Trigger Polling Page

The following table describes the key information on the trigger Polling page.

> **Note**
>
> Sequential order is not always maintained while polling records.

Element	Description
Add Tables	Import tables and the root database table for the service query.
Remove Tables	Removes tables. Select **Remove Tables**, clear the check box to the right of the table you want to remove, and click **OK**. You cannot remove the root database table.
Review and manage parent database table relationships	Appears after importing tables. Select **Edit** to open the Relationships page where you can view, create, and remove relationships between tables.
Review and verify table and relationship attributes	Appears after importing tables. Select **Edit** to open the Attributes Filtering page where you can review, verify, select, or deselect the attributes in the object model created from the imported tables and the defined relationships.
Review the polling strategy and specify polling options	Appears after importing tables. Select **Edit** to open the Polling Strategy and Options page where you can define the polling strategy and specify polling options.

Trigger Manage Tables Page

The following table describes the key information on the trigger Manage Tables page. The trigger Manage Tables page appears when you select **Add Tables** on the adapter trigger Poll for a New or Changed Records page.

Element	Description
Schema	Select the schema for the tables and views you are importing.

Element	Description
Table Type	The type of the table to which the schema or view is applied. The list allows these selections: • **All** — Select all available tables and views. • **Table** — Select tables. • **View** — Select views.
Table Name	Specify the table name. Table names are case sensitive.
Search	Click to search for the specified table.
Available	Lists the tables that meet the selection criteria.
Selected	Lists the selected table.
Filter By	Type the initial letters to filter the display of table names.
Primary Key	Appears when you select tables without a primary key defined. Select the virtual primary key for the table. **Note**: Having the primary key at the database level is the best practice.

Trigger Relationships Table

The following table describes the key information on the trigger Relationships page. The trigger Relationships page appears when you select **Edit** to review and manage the parent database table relationships option on the adapter trigger Poll for a New or Changed Records page.

Element	Description
Create New	Opens the Create Relation page with these options: • **Parent Table** — Select the parent table for the relationship between tables. • **Child Table** — Select the child table for the relationship between tables. • **Relationship** — Defines the relationship between the parent and child tables. • **Attribute Name** — Apply attributes to the table relationship. • **Mapping** — Provide the mapping for the table relationship.

Trigger Polling Strategy and Options Page

The following table describes the key information on the trigger Polling Strategy and Options page. The trigger Polling Strategy and Options page appears when you select **Edit** to review the polling strategy and specify polling options on the adapter trigger Poll for a New or Changed Records page.

Element	Description
Logical Delete Field	Selects a field in the root database table. To allow the selection, polling must be enabled in the **Status** column.
Read Value	Identifies the value that is used to indicate a row has been read (for example, **PROCESSED**). Surrounding quotes are not required.

Element	Description
Unread Value	Indicates the rows to process. Only rows with logical delete field and column values that match the **Unread Value** are read.
Rejected Value	Set to **REJECTED**. If the incoming message is greater than the threshold size, that particular record is updated to **REJECTED** instead of **READ**. If the outbound operation returns a response greater than the size allowed, the response message is ignored and a fault response is sent to the calling client. For details about supported sizes, see Service Limits in *Provisioning and Administering Oracle Integration 3*.
Polling Frequency (Sec)	Specifies the polling frequency (in seconds) for new records or events.
Advanced Options	Click **Edit** to access the **Batch Size** field to specify the number of table rows to process during a single transaction. The default value is **1** and the maximum value is **50**.

Invoke a Stored Procedure Page

Enter the PostgreSQL Adapter invoke stored procedure parameters. The Invoke a Stored Procedure page is the wizard page that is displayed if you selected **Invoke a Stored Procedure** as the operation type on the Basic Info page.

> **Note**
>
> Adapter input/output parameters are defined based on the stored procedure IN/OUT parameters. The IN parameter corresponds to the request and the OUT parameter is translated as the response.

Element	Description
Select Schema	Select a database schema from the list that includes the data you want to query (for example, you want to query details about an employee based on their employee ID). This action refreshes the page to display fields for selecting a package or procedure to invoke. **Note**: Starting with the 24.06 release, actual schemas are displayed in place of databases.
Select Package	Select the database package. **Note**: Starting with the 24.06 release, select ***default package***.
Select Procedure	Select the stored procedure. The page is refreshed to display the in (inbound), out (outbound), and in/out (inbound/outbound) parameters available with this procedure.
Arguments	Displays the in, out, and in/out parameters that are passed with this procedure.

Invoke Run a SQL Statement Page

Enter the SQL statement parameters. The Run a SQL Statement page appears when you select the **Run a SQL Statement** operation on the Basic Info page. You can specify the following values on the Run a SQL Statement page.

Note

- Do not use schema/database names in SQL queries. Configure the details in the connection. For example:

  ```
  Update HR.employee set HR.employee.first_name = 'Name' where
  HR.employee.employee_id='1'
  ```

  can be changed to a simple query, such as:

  ```
  Update employee set first_name = 'Name' where employee_id='1'
  ```

  where `HR` is used in the connection details. This restricts a user with specific privileges to a particular schema/database.
- When configuring the adapter as an invoke connection, ensure that proper spaces are provided between key words for a pure SQL statement. For example, the following statement fails during integration activation because there is no blank space between `VALUES` and `(#`.

  ```
  INSERT INTO table_name VALUES(#EMPNO, #EMPNAME)
  ```

  Add a blank space between `VALUES` and `(#`, and the statement is successfully processed.

  ```
  INSERT INTO table_name VALUES (#EMPNO, #EMPNAME)
  ```

Element	Description
SQL Query	Enter the SQL query.
Status	Display the results of the SQL query validation. The **Status** field displays `Success!` when a query is successfully validated.

Invoke Operations On Table Page

Select the database tables. To use the bulk extract feature, you must choose the **SELECT** operation from the **Perform an Operation On a Table** list on the Basic Info page.

Operations on Table Page

Element	Description
Schema	Select the database schema that includes the tables to process.
Table Name	Enter a filter with which to search the schema (for example, %TAB to search for tables with TAB in the name).
Table Type	Specify the table type filter to get a subset of the appropriate database objects, then click **Search**. • **ALL** • **TABLE** • **VIEW**
Filter By	Enter the initial letters to filter the display of table names.
Available	Lists the tables that meet the selection criteria.
Selected	Lists your table selection.
Import Tables	Click to import the tables. The page is refreshed for you to select the parent database table.
Primary Keys	Appears when you select tables without a primary key defined. Select the virtual primary key for the table. **Note**: Having the primary key at the database level is the best practice.
Select the parent database table	Select the parent (root) table from the list. If using multiple related tables, this is the top-level parent table in the relationship. After making your selection, the page is refreshed for you to view and edit the table relationships.
Add || Remove Tables	Click to add more tables or remove tables no longer in use.
Review and manage parent database table relationships	Click **Edit** to view and edit the table relationships. The relationships automatically identified by the adapter are displayed. See Review and manage parent database table relationships Option.
Review and filter columns from selected database tables	Click **Edit** to view and edit the table attributes. You can deselect any attributes to exclude from the database queries. Primary key attributes cannot be excluded. See Review and filter columns from selected database tables Option.
Review and edit SQL query (Displayed only if the **Select** operation is selected on the Basic Info page.)	Click **Edit** to view and edit the default SQL query. See Review and edit SQL query Option.

Review and manage parent database table relationships Option

Specify values for the **Review and manage parent database table relationships** option.

Element	Description
Create New	Click to create a new relationship.
Parent Table	Select the parent table.
Child Table	Select the child table.
Relationship	Select the relation type (one-to-many, one-to-one, or one-to-one with the foreign key on the child table). For example, if you selected **employee** as the parent table and **datatypes** as the child table, the following options are displayed: • **employee has a 1:1 Relationship with datatypes** • **employee has a 1:1 Relationship with datatypes (Foreign Key on Child table)** • **employee has a 1:M Relationship with datatypes**
Attribute Name	Applies attributes to the table relationship.
Mapping	Displays the mapping for the table relationship.

Review and filter columns from selected database tables Option

Specify values for the **Review and filter columns from selected database tables** option.

Element	Description
Select the Columns	View and deselect attributes automatically created by the adapter. Deselect any attributes to exclude from the database query.

Review and edit SQL query Option

Specify values for the **Review and edit SQL query** option.

> **Note**
>
> This is only applicable for a `Select` operation on a table.

Element	Description
SQL Edit	Click to manually edit the query in the **SQL Query** field.
Parameter	Click to specify a bind parameter.
Add New	Click to add new criteria to the SQL query.
Remove	Click to remove the SQL criteria you specified.

Element	Description
Maximum Number of Records to be fetched	Select the number of records to fetch with this SQL query.
Pagination	Select the check box. When you must fetch a large number of results/records, you can implement the PostgreSQL Adapter's pagination feature and receive sorted results/records in chunks. **Offset** and **Limit** fields are displayed in the mapper. You can specify the number of results per page by providing the offset and limit values according to your requirement. Enter a starting value in the **Offset** field and an end value in the **Limit** field to receive sorted results. See Use Pagination in an Integration.

Invoke Bulk Load from File to Table Page

The following table describes the key information on the Bulk Load from File to Table page.

Element	Description
Delimiter	Select one of the following supported file delimiter options as per the source file: • **Single space** • **Comma** • **Semicolon** • **Pipe** (for example, Name|City|Country)
Select Schema	Select the database schema that includes the tables to process.
Select Table	Select the table name.
Search Field	Enter the initial letters to filter the display of table names.
Table columns	Displays the table columns that meet the selection criteria.

See Import Data Files Using the Bulk Data Import Operation and Import Transactional Data Using the Bulk Data Import Operation.

Summary Page

You can review the specified adapter configuration values on the Summary page.

Element	Description
Summary	Displays a summary of the configuration values you defined on previous pages of the wizard. The information that is displayed can vary by adapter. For some adapters, the selected business objects and operation name are displayed. For adapters for which a generated XSD file is provided, click the XSD link to view a read-only version of the file. To return to a previous page to update any values, click the appropriate tab in the left panel or click **Go back**. To cancel your configuration details, click **Cancel**.

4

Implement Common Patterns Using the PostgreSQL Adapter

You can use the PostgreSQL Adapter to implement the following common patterns.

Topics:

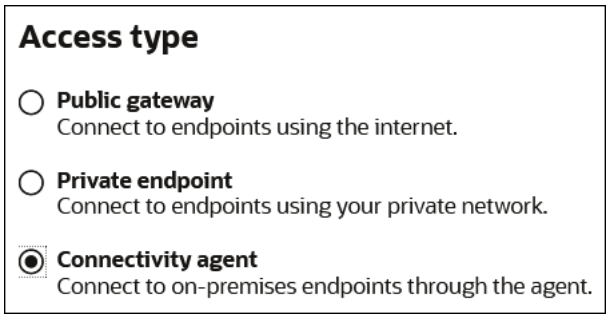
> **Note**
>
> Oracle Integration offers a number of pre-assembled solutions, known as recipes, that provide you with a head start in building your integrations. You can start with a recipe, and then customize it to fit your needs and requirements. Depending upon the solution provided, a variety of adapters are configured in the pre-assembled solutions.
> See the Recipes page on the Oracle Help Center.

Perform Inbound Polling Without the Connectivity Agent

You can perform inbound polling without the connectivity agent. You can create new connections or update existing connections to bypass the connectivity agent. Depending on your database or messaging service location, you can use the public internet or a private endpoint for polling. If you want, you can continue to use the connectivity agent for inbound polling.

Perform the following steps to create a new connection or update an existing connection to bypass the connectivity agent.

1. Go to the **Access type** section of the Connections page.



2. Select an option based on the location of your database or messaging service.

- If the database or messaging service to access is in a private network, select **Private endpoint** to poll without the connectivity agent. This selection requires that you first configure private endpoint support in the Oracle Cloud Infrastructure. See Connect to Private Resources in *Provisioning and Administering Oracle Integration 3*.
- If the database or messaging service to access is public, select **Public gateway** to poll without the connectivity agent. Private endpoint configuration is not required in these scenarios and the adapter polls directly without the connectivity agent.

3. Deactivate and then reactivate the integration.

> **Note**
>
> After completing these steps, the connectivity agent is no longer used for inbound polling. However, it still continues to run. If you no longer need to use the connectivity agent, you can manually stop it.

Several adapters support inbound polling without the connectivity agent. Watch a video to learn more about how to use this feature with one such adapter.

Video

Import Data Files Using the Bulk Data Import Operation

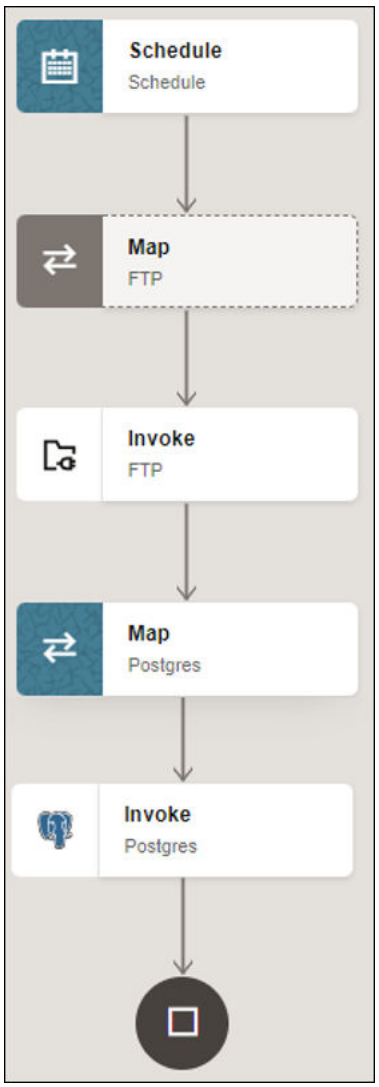
This use case describes how to import data files from an FTP server into the PostgreSQL database. Similarly, you can import data files from an application into the PostgreSQL database using the PostgreSQL Adapter.

To perform this operation, you create the FTP Adapter and PostgreSQL Adapter connections in Oracle Integration. The PostgreSQL Adapter first validates an input file header with the target table header (columns), and then inserts the data into the target table if the data is in the expected format.

1. Create a scheduled orchestration integration.
2. Drag an FTP Adapter into the integration canvas.
3. Configure the FTP Adapter as follows.
   a. On the Basic Info page, provide a name.
   b. On the Operations page, select **Download File** from the **Select Operation** list.
   c. Select **Binary** from the **Select a Transfer Mode** list.
   d. Provide the input directory, file name, and download directory.
   e. On the Summary page, review your selections.
4. Drag a PostgreSQL Adapter into the integration canvas.
5. Configure the PostgreSQL Adapter endpoint:
   a. On the Basic info page, provide an endpoint name, and select **Perform Bulk Data Import Operation**.
   b. On the Bulk load from File to Table page, select the delimiter (for example, comma), schema, table, and table columns.
   c. On the Summary page, review your selections.
6. In the mapper, map the file reference from the FTP response to pass the data to the target table. The FTP response (file reference) provides an input to the PostgreSQL database.

7. Click **Validate**.

   The completed integration looks as follows.



8. When complete, save and activate the integration. As a result, the PostgreSQL Adapter inserts the data into the target table if the data is in the expected format.

Import Transactional Data Using the Bulk Data Import Operation

This use case describes how to import transactional records in chunks from an application (for example, Salesforce) into the PostgreSQL database. In this use case, the Salesforce application is used. Similarly, you can import data files from other applications into the PostgreSQL database using the PostgreSQL Adapter.

To perform this operation, you create the Salesforce Adapter and PostgreSQL Adapter connections in Oracle Integration. The PostgreSQL Adapter first validates an input file header with the target table header (columns), and then inserts data into the target table if the data is in the expected format.

1. Create an app-driven orchestrated integration.
2. Drag a REST Adapter into the integration as a trigger connection.

a. On the Basic Info page, provide a name.

b. On the Resource Configuration page, select the **GET** action and the **Configure this endpoint to receive the response** check box.

c. On the Response page, select **JSON Sample** in the **Select the response payload format** field.

d. Select **JSON** in the **What is the media-type of Response Body? (Accept Header)** field.

e. Review your selections on the Summary page.

3. Drag a Salesforce Adapter into the integration canvas.
4. Configure the Salesforce Adapter endpoint:

a. On the Basic Info page, provide a name.

b. On the Action page, select **Query Information**.

c. On the Operations page, select **Query** as an operation and the **Exclude** option for the deleted and achieved records.

d. Enter a valid SOQL query statement and select the **Use Default Header** check box.

e. Review your selections on the Summary page.

5. Drag a stage file action into the integration canvas after the Salesforce Adapter and configure it to write (insert) transactional records in a file.

a. On the Basic Info page, provide a name.

b. On the Configure Operation page, select **Write File** from the **Choose Stage File Operation** field.

c. Specify the X Path expression for the file name in the **Specify the File Name** field.

d. Enter the file name with an extension (for example, `Accountant`).

e. Specify the directory name in the **Specify the Output Directory** field.

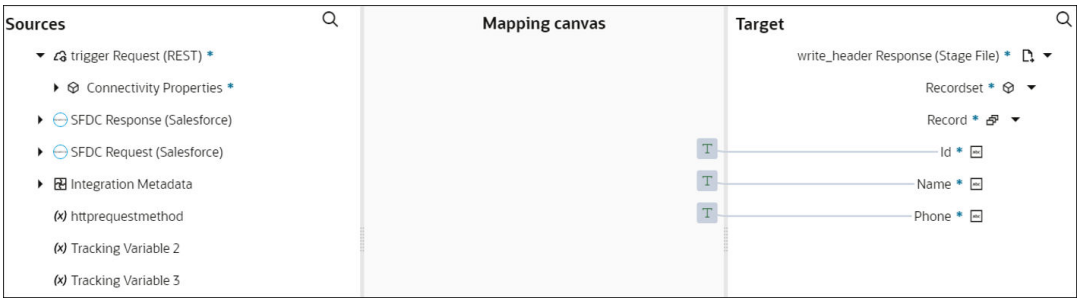
f. On the Scheme Options page, select **Yes** in the **Do you want to specify the structure for the contents of the file** field.

g. Select **Sample delimited document (e.g. CSV)** in the **Which one of the following choices would be used to describe the structure of the file contents** field.

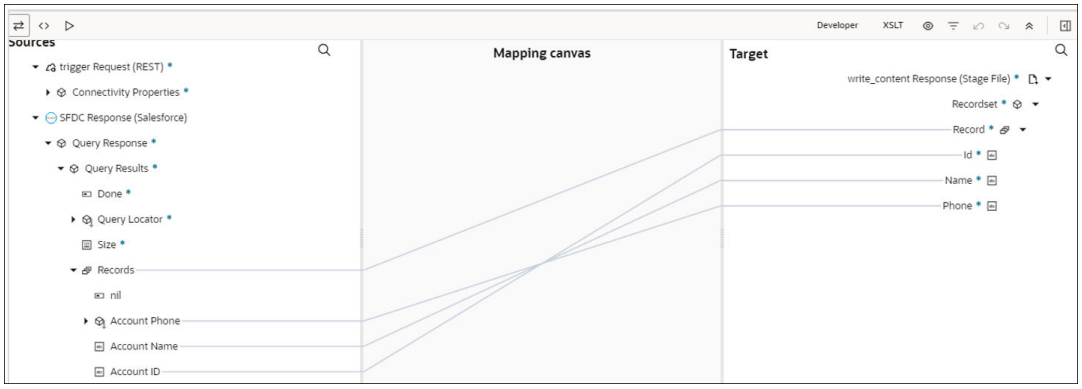
h. On the Format Definition page, click **Drag and Drop** and upload the sample CSV file in the **Select a New Delimited Data File** field.

i. Review your selections on the Summary page.

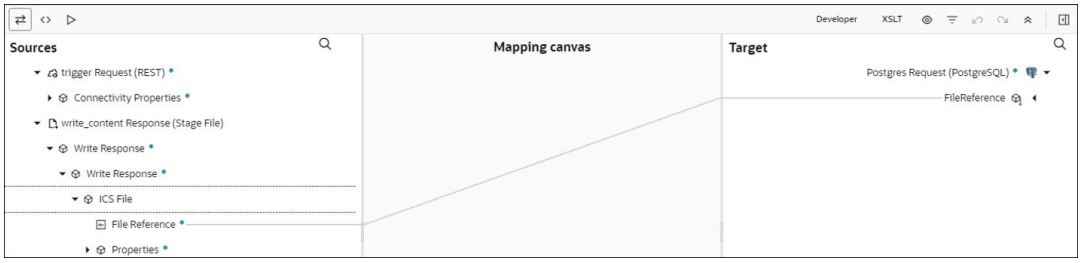
6. In the mapper, map headers as an input for the **write_header** (stage) file.



7. Drag a second stage file action into the integration canvas. The stage file action helps to add the header to a file and then append the data to the same file.
   a. On the Basic Info page, provide a name.
   b. On the Configure Operation page, select **Write File from the Choose Stage File Operation** field.
   c. Specify the X Path expression for the file name in the **Specify the File Name** field.
   d. Enter the same file name and extension that you provided for the headwaiter (stage).
   e. Specify the directory name in the **Specify the Output Directory** field.
   f. Enter the same output directory that you provided for the headwaiter (stage).
   g. Under **Append to Existing File**, select the **Append** check box.
   h. On the Scheme Options page, select **Yes** in the **Do you want to specify the structure for the contents of the file** field.
   i. Select **Sample delimited document (e.g. CSV)** in the **Which one of the following choices would be used to describe the structure of the file contents** field.
   j. On the Format Definition page, click **Drag and Drop** and upload a sample CSV file in the **Select a New Delimited Data File** field.
   k. Review your selections on the Summary page.
8. In the mapper, map the Salesforce response to the **write_content** (stage) request.

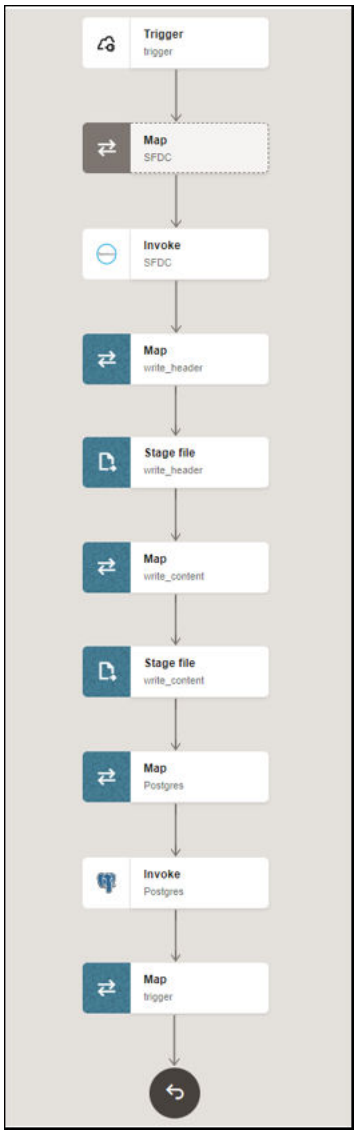


9. Drag a PostgreSQL Adapter into the integration canvas.
10. Configure the PostgreSQL Adapter endpoint:
    a. On the Basic info page, provide an endpoint name, and select **Perform Bulk Data Import Operation**.
    b. On the Bulk load from File to Table page, select the delimiter (for example, comma), schema, table, and table columns.
    c. On the Summary page, review your selections.
11. In the mapper, map the file reference from the **ICS file** response to pass the data to the target table.



12. Click **Validate**.

    The completed integration looks as follows.



13. When complete, save and activate the integration. As a result, the PostgreSQL Adapter inserts data into the target table if the data is in the expected format.

Use Pagination in an Integration

When you must fetch a large number of results, you can use the PostgreSQL Adapter's pagination feature and receive sorted results/records in chunks. This use case describes how to use pagination to receive sorted results/records from the PostgreSQL database insert into an application (for example, SAP ASE) or files.

In this use case, the PostgreSQL database is the source, and the SAP ASE database is the target application. Similarly, you can receive and insert sorted results/records from the PostgreSQL database into other applications or files using the PostgreSQL Adapter.

To perform this operation, you create the PostgreSQL Adapter andSAP ASE (Sybase) Adapter connections in Oracle Integration.

This use case uses the following features or operations:

- **Offset and Limit Fields**: These fields are displayed in the mapper. You can specify the number of results per page by providing the offset and limit values according to your requirement. Enter a starting value in the **Offset** field and an end value in the **Limit** field to receive sorted results.
- **Run a SQL Statement** (PostgreSQL Adapter): Retrieves the total number of records.
- **Select** (PostgreSQL Adapter): Enables you to enter the required query and select the pagination option.
- **Insert or Update** (SAP ASE (Sybase) Adapter): Inserts or updates the records into the SAP ASE database.

1. Create a scheduled orchestration integration.
2. Assign the following two variables to your integration. Specify the values to the variables as follows:

   ```
   offset = "0"
   limit= "schedule_limit"
   ```

3. Drag a PostgreSQL Adapter into the integration canvas.
4. Configure the PostgreSQL Adapter as follows:
   a. On the Basic info page, provide an endpoint name, and select **Run a SQL Statement**.
   b. On the Run a SQL Statement page, enter a SQL query, and click **Validate SQL Query**.
   c. On the Summary page, review your selections.
5. Assign the following variable to your integration. Specify the value to the variable as follows:

   ```
   count="COUNT"
   ```

6. Drag a while action below the assign action.
7. Set the condition for the specified variable as follows:

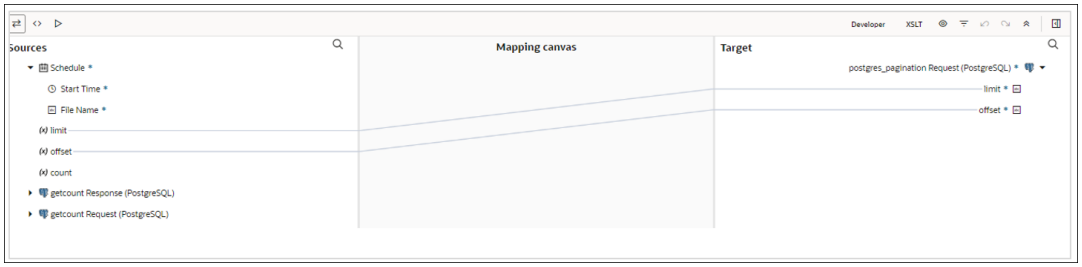
   ```
   offset < count
   ```

8. Drag a PostgreSQL Adapter inside the while action.

9. Configure the PostgreSQL Adapter as follows:
   a. On the Basic info page, provide an endpoint name, and choose **Select** from the **Perform an Operation On a Table** list.
   b. On the Operation on Table page, select the parent database table, and click **Edit** under **Review and edit SQL Query**.
   c. In the SQL Query field, enter the required query. For this example:

      ```
      SELECT emp_id, employee_name, "location", designation
      FROM public.employee rows limit #limit offset #offset.
      ```

   d. Select the **Pagination** check box.
   e. On the Summary page, review your selections.
10. In the mapper, map **offset** to **offset** and **limit** to **limit**.



11. Drag an SAP ASE (Sybase) Adapter inside the while action.
12. Configure the SAP ASE (Sybase) Adapter as follows:
    a. On the Basic info page, provide an endpoint name, and choose **Insert** from the **Perform an Operation On a Table** list.
    b. On the Operation on Table page, select the parent database table, and click **Edit** under **Review and filter columns from selected database tables**.
    c. Select the required columns.
    d. On the Summary page, review your selections.
13. Drag an assign action inside the while action to update the specified variables as follows:

    ```
    offset = offset+limit
    ```

14. When complete, save and activate the integration.

    The completed integration looks as follows.

5

Troubleshoot the PostgreSQL Adapter

Review the following topics to learn about troubleshooting issues with the PostgreSQL Adapter.

Topics:

Primary Key Error While Importing Table for the Merge Operation

The following error occurs when a table without a primary key is selected for the **Insert or Update (Merge)** operation on the Operations On Table page.

```
One or more tables imported, doesn't have a primary key. Merge operation
cannot proceed without a primary key.
```

Solution: Ensure that a table with a primary key is selected.

No Package/Procedure Found

You can receive the following message when you a select a database name other than the ones you configured on the Connections page in Oracle Integration.

```
No package/procedure definition found
```

Solution: Select the database name you configured on the Connections page in Oracle Integration.