Zoho Corporation

PostgreSQL [Cloud & On-Prem]

What is PostgreSQL?

PostgreSQL is an open-source relational database management system (RDBMS) used to store, manage, retrieve, replace, and materialize structured data efficiently.

How to connect your PostgreSQL account to Zoho Flow

1. Select the required trigger or action. If you select a trigger, click **Next**.
2. If there are no existing PostgreSQL connections in your account, click Connect. Otherwise, click **New**. Alternatively, you can create a new connection by navigating to **Settings**, then **Connections**. Click **Create connection** and choose PostgreSQL.
3. Enter a connection name.
4. Check the **App is installed on-premises box** if you're using the on-prem version of PostgreSQL. Leave it unchecked for the cloud version or if your account is hosted publicly.
   Note: For a cloud database, you need to allow-list Zoho Flow's IP addresses in your inbound rule.
5. For an on-prem setup, choose an on-prem agent from the dropdown to create a Zoho Flow connection with your local machine. If you don't already have an agent installed, click **New**.
6. Configure the following fields to create the connection:
   Host: The URL of your hosted server
   Port: The port number that your server is running on, typically 5432.
   Username: Your PostgreSQL username
   Password: Your PostgreSQL password
   Database: The name of the PostgreSQL database that you want to access
   Schema: The name of the schema you want to connect within your database
   Enable SSL?: The option to use SSL (Secure Sockets Layer) for encrypted communication between the PostgreSQL client and server
7. Once configured, click **Authorize**.

Triggers and actions available in Zoho Flow

Triggers

Row added

Use this trigger to automate actions whenever a new record is added to a PostgreSQL table. For example, you can send a notification or update another database when a new order is added in your ecommerce platform.

Row added or updated

This trigger can be used to keep data synchronized between different systems. For example, if a customer's contact information is added or updated in your PostgreSQL database, you can automatically update their profile in your email marketing tool.

Note:
To use the Row added and the Row added or updated triggers, you need to configure the following fields:

- **Trigger column**: Select a unique column with incremental values — ideally timestamp columns such as created_at, updated_at, published_at, etc. By default, the result is sorted in ascending order, and the limit is 100.
- **Where condition**: Use a Where condition to filter the rows. Zoho Flow will retrieve only rows added after the last execution of the flow, so you receive only the latest updates that satisfy the Where condition.

You can use Filter criteria when using these triggers to selectively pass information based on your custom criteria instead of passing the entire payload.

Actions

Fetch row

This action is useful for scenarios where you need to access specific data based on a primary or unique key. For example, if you have an order number and want to retrieve all details of that order, you can use this action to fetch the row and perform further actions like sending order confirmation emails.

Note: The maximum number of records that can be fetched in a single execution is 100.

To use the Fetch row action, you need to configure the following fields:

- **Unique key**: A primary key or a column with a unique constraint to de-duplicate rows. An indexed column will give better results.
- **Unique key - Value**: The value of the unique key.
- **Additional key**: Apart from the unique key, you can also add another column as an additional key for deduplication.
- **Additional key - Value**: The value of the additional key.

Insert row

Use this action to add new data to your PostgreSQL database. For example, when a user submits a form on your website, you can use this action to insert their information into your customer database.

Update row

This action allows you to update existing records in your PostgreSQL table. For example, if a customer requests to change their email address, you can use this action to update the email field in your database.