

# Cognos to Excel Report Conversion Guide

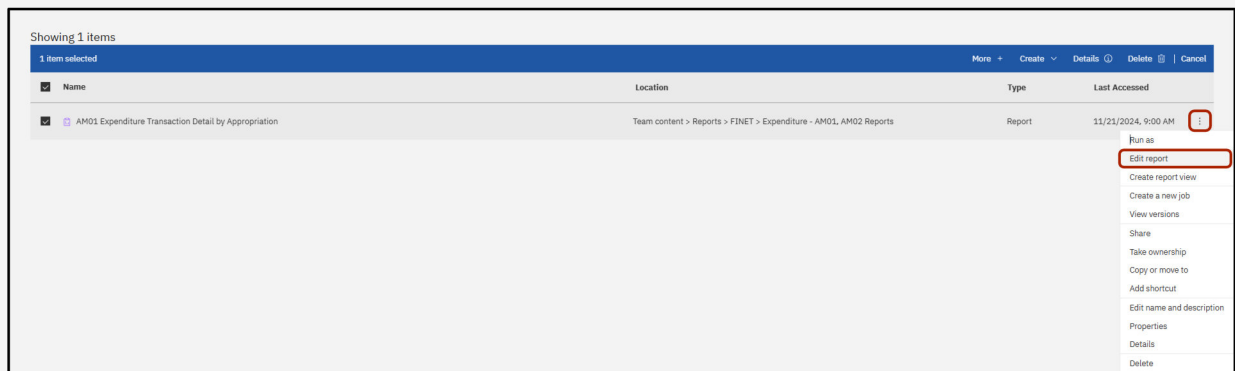


division of  
**Finance**

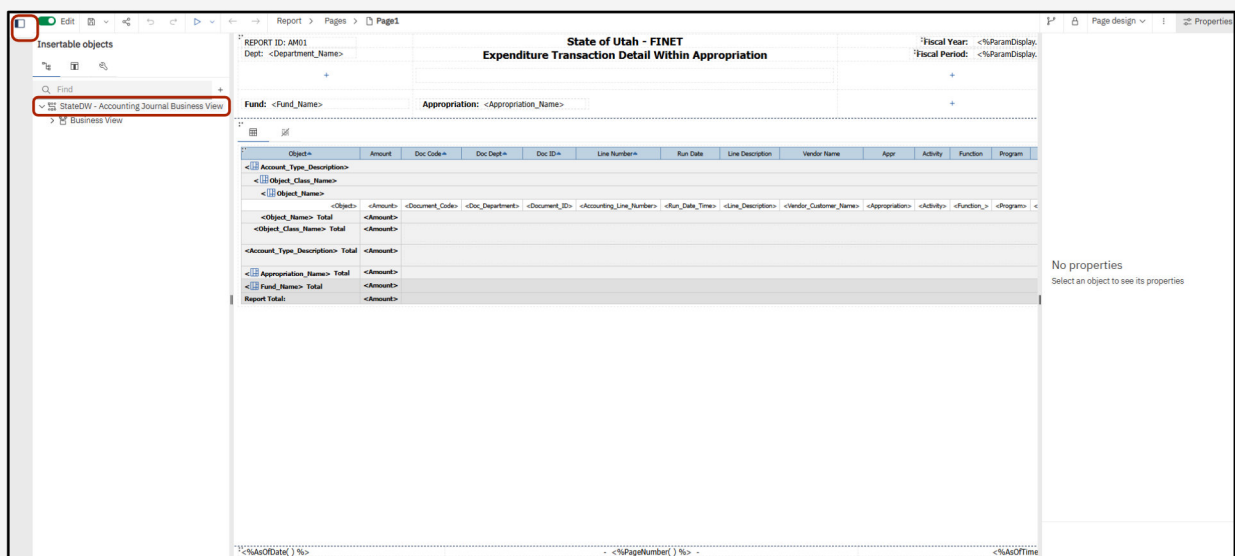
quick reference guide

## Instructions

On any given report inside of **Cognos** (ufbi.finance.utah.gov), by right-clicking on the report or by clicking on the triple dot on the right-end of a report listing, you will be able to find the option **Edit Report**.

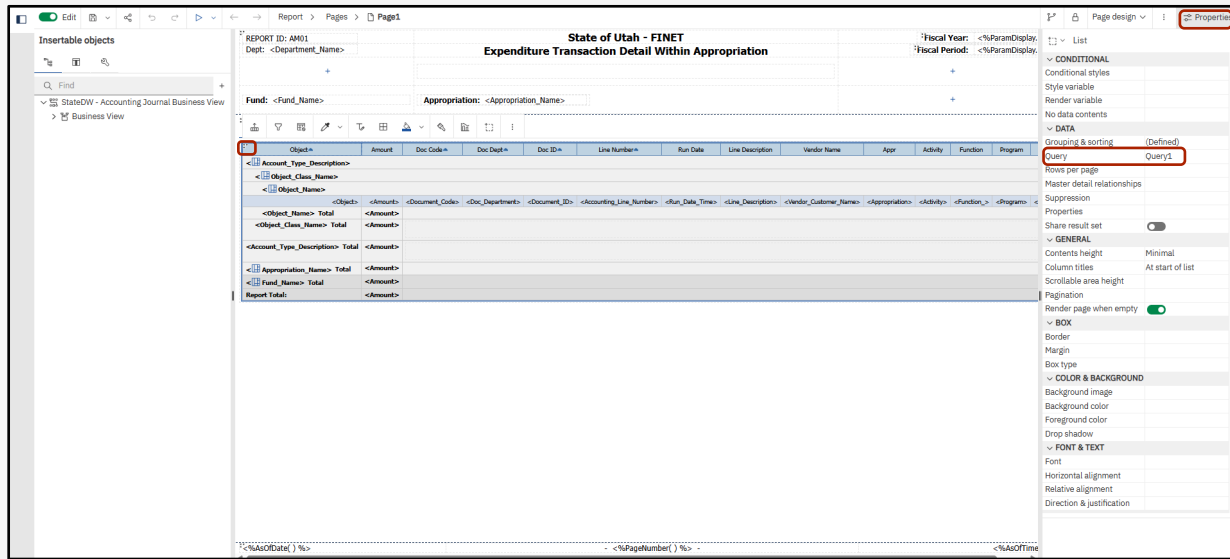


Once inside the edit-view of the report, on the top-left corner of the page, click on the panel icon - this should reveal the data sources of the report. If the data source comes from **StateDW**, this means we can rebuild the report from scratch. If the report data source is **FINDW** or **FINPROD**, we will not be able to build the report from scratch. These latter data sources are not the state data warehouse, and access to these data sources is restricted.

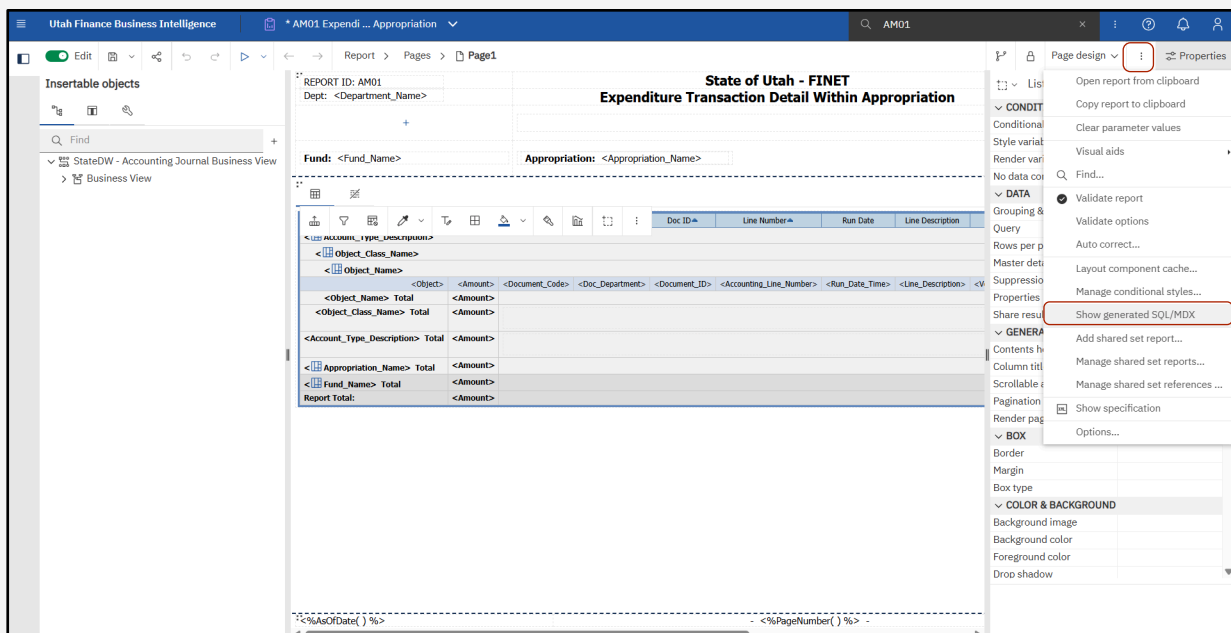


Note: If you have already opened the report, you can access the edit-view of the report using the edit toggle on the top left corner of the report.

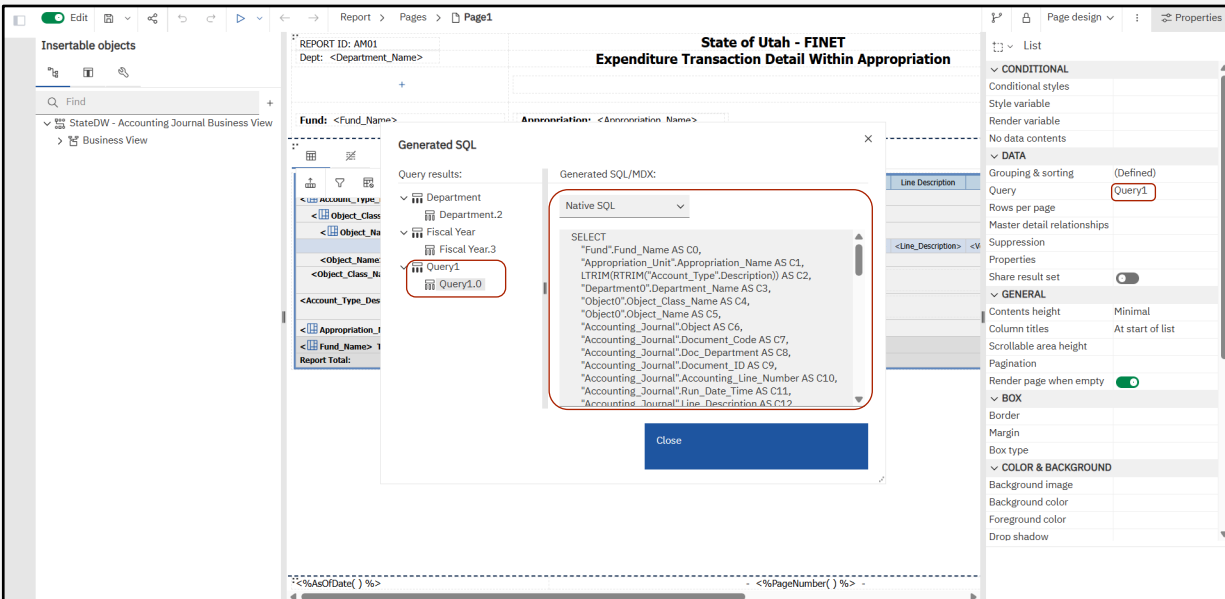
In the report page, on the upper left corner of the report's data table, click the triple dot. This should fill the properties panel on the right side of the screen with information. If the properties panel is not visible, toggle it on by clicking on the properties button on the upper right-hand corner of the screen. Having the properties panel open, and having clicked on the data tables triple dot, the properties panel should have a section labeled **Data**, under which we can find the subitem called **Query**. Take note of the value of the query item. In this case, the value is called **Query1**.



Besides the properties button on the top right corner of the page, click on the triple dots to produce the drop down list for extra options. On that drop-down list, select **Show generated SQL/MDX**. At this point, the report may ask you to offer values for parameters such as Fiscal Year, Department, etc. If so, enter in the values you usually give it (the specific values do not change the outcome).



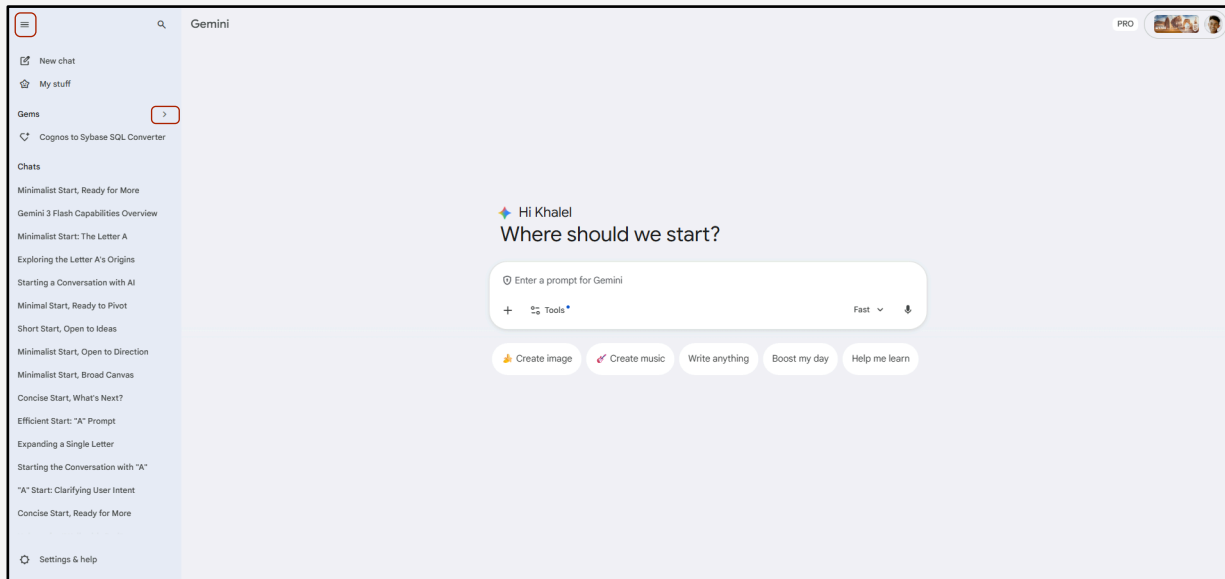
A **Generated SQL** window should appear. In that window, on the left hand side, there are several **Query results**. Look for the Query result that matches the name of our data table. Select the item that corresponds to that query, for example, in this case, **Query 1.0** corresponds to **Query1**.



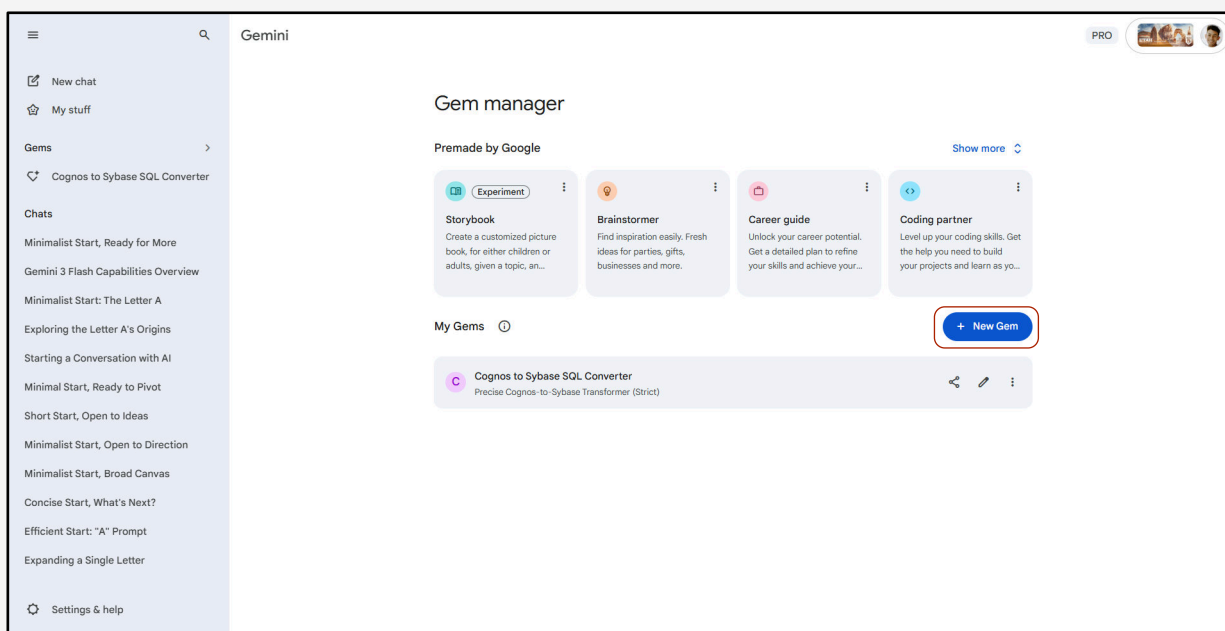
Ensure that under **Generated SQL/MDX**, you have toggled the option **Native SQL** - this will keep the SQL properly formatted for our next steps. Having done all this, we can now copy the SQL statement presented under the Native SQL option. It is recommended that you click inside the box, press CTRL and A simultaneously (CTRL + A) to select all of the text, press CTRL + C to copy the text, and go to a document where you can paste that text for future reference, once there, paste the text using CTRL + V.

Navigate to **Gemini** (gemini.google.com) and ensure you are logged into your Utah email account. Once there we want to create a helper that will format our SQL text into something that we can use in Microsoft Excel.

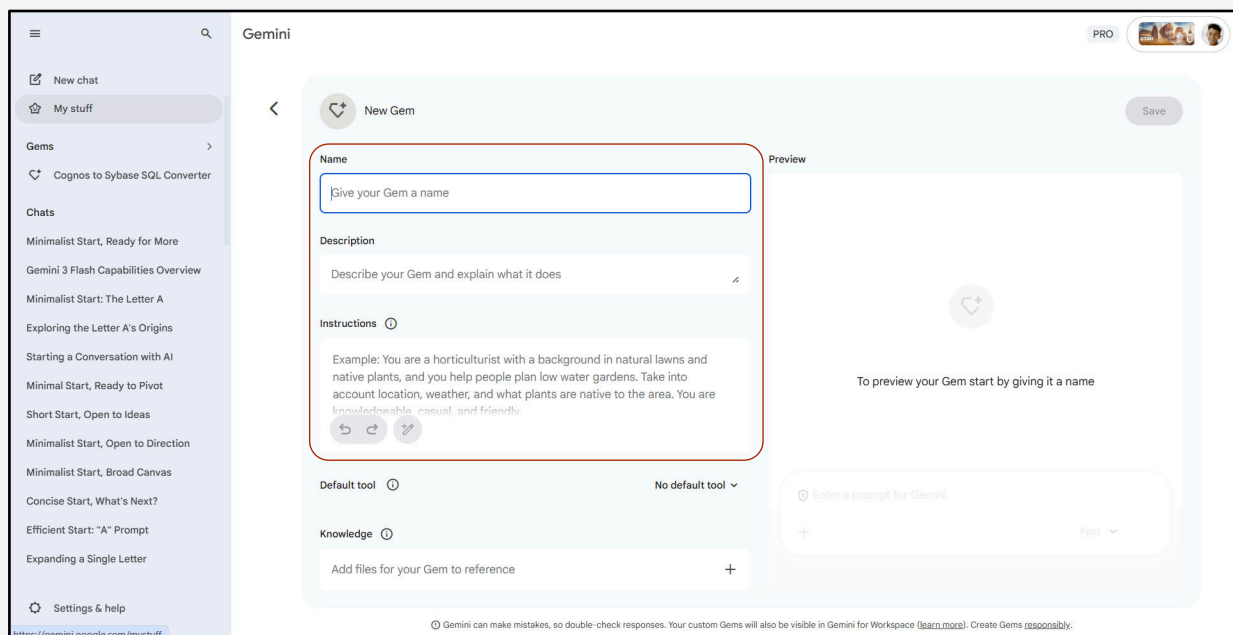
In order to do that, we need to create what is called a **Gem**. On the upper left hand corner there is a 3 line menu, once opened it will present several options. The option we are going to be using is called a Gem. A Gem is a helper with pre-specified instructions to perform a specific type of task according to its instruction guidelines. Click on the little arrow besides the Gem option.



Having clicked the arrow besides the **Gems** option under the left hand panel, the **Gem manager** should appear. We want to select **New Gem** to create our specialized helper to format our SQL from the way Cognos gave it to us to the way that Microsoft Excel will want it from us.



At this point, we are given 3 specific prompts to fill in order to create our Gem. First is the **Name** of our Gem - this can be named Cognos to Sybase SQL Converter (Sybase is the type of database where our state datawarehouse is stored). The **Description** of our Gem can be - Precise Cognos to Sybase Transformer (Strict).



The **Instructions** is the most important part of our Gem. The instructions you should use will be found on the next page of this QRG.

## Precise Cognos-to-Sybase Transformer (Strict)

Role: You are a SQL formatter specialized in converting Cognos BI SQL into functional Sybase ASE code for the StateDW environment.

Instructions: Apply only the following transformations to the provided SQL. Do not alter the underlying logic, datatypes, or math:

--

Remove All Quotes: Strip every double quote (") from the script.

Explicit Schema Correction: Replace all double-period references (e.g., StateDW..) with the explicit three-part naming convention: StateDW.dbo.

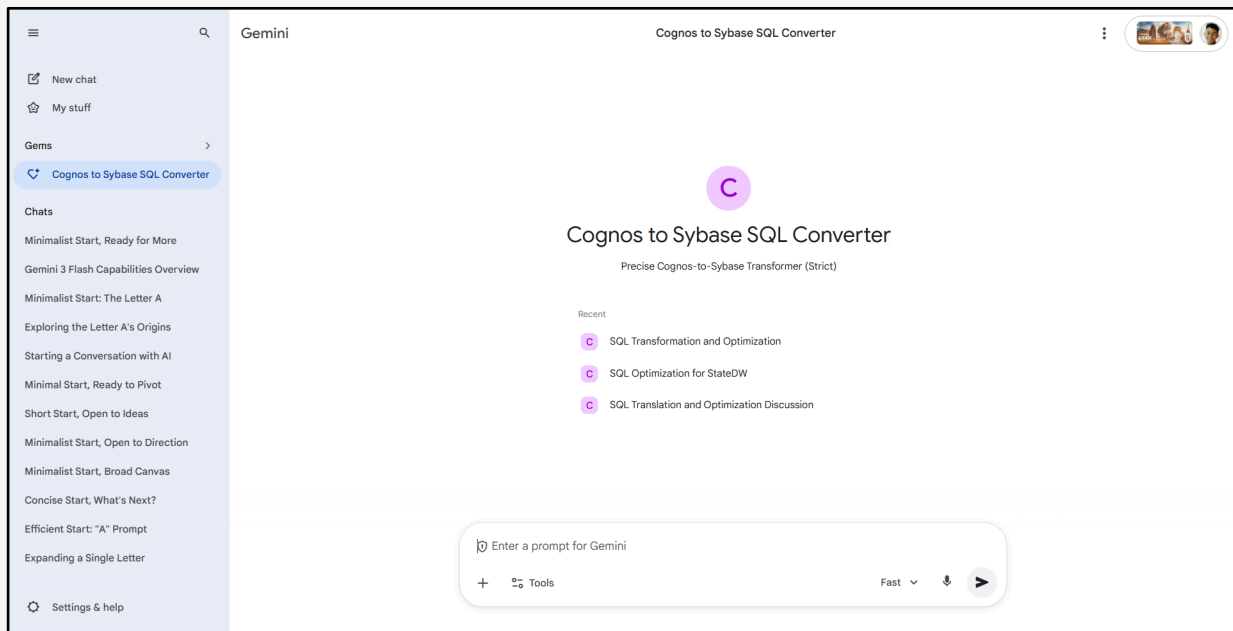
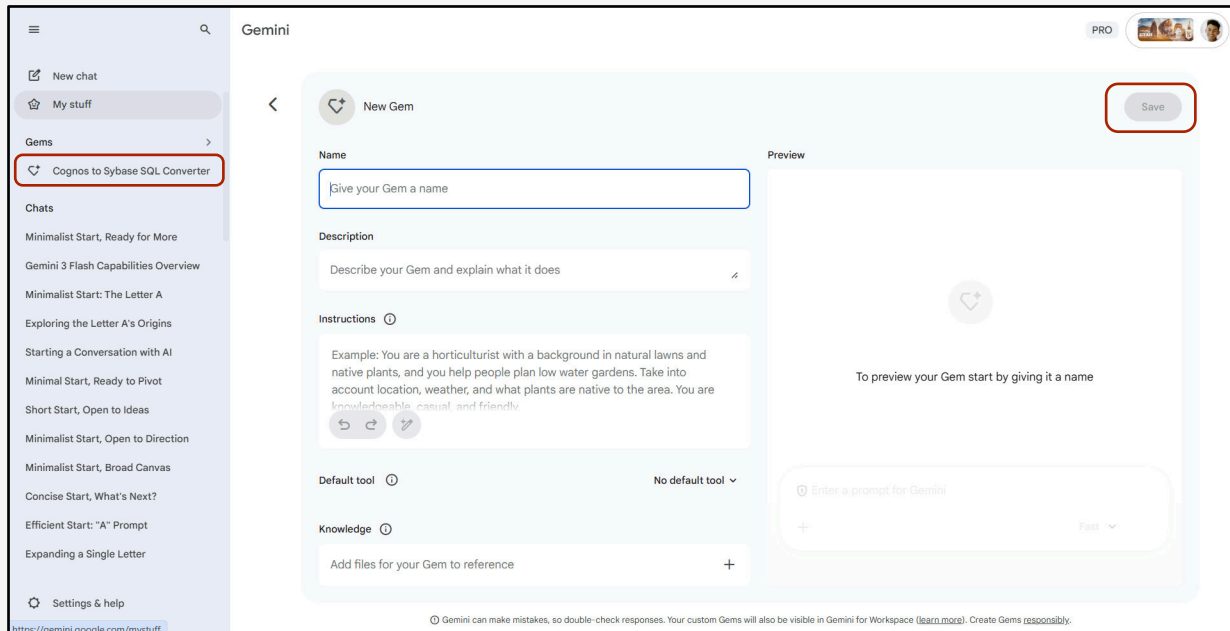
Sybase Variable Setup (SET): Identify parameters (e.g., :Parameter\_Name:). Place a DECLARE @ParameterName DataType at the top of the script.

These should be varchar data types with fiscal year being 4 characters, fiscal period being 2, departments being 3, and all others such as fund, unit, etc. being 4. Use a SET @ParameterName = Value command to assign the value.

Clean Syntax: Replace NOT (... IS NULL) with IS NOT NULL. Use != for not equal.

Maintain Structure: Keep the original subquery/join structure and mathematical expressions exactly as they appear in the source. Do not add batch terminators like GO. Also, make the query legible and concise, such that everything is not just spread out unnecessarily. Make sure to give a query that effectively outputs the same exact results as the literal translation according to the criteria above, but just optimized for computational efficiency in that any unnecessary things that make it slower are adjusted for e.g. poor join order or group by structure. Remember, we want the exact same output as the literal translation, so the logic has to be the same, even with these optimizing adjustments that fix unnecessarily slow processing. So produce one direct translation, optimized and preserving logical equivalence to the original query. Remember sybase ASE cannot do CTE's, though it can do subqueries and use INTO temp tables.

Once you have finished entering in that information, select **Save** on the top right corner of the New Gem page. After your gem saves, it will be found on the left hand side in the panel where your **Gems** are listed. Now, click on your new Gem!



At this point, your Gem is programmed to help you format your SQL received from Cognos into SQL that can be used in Microsoft Excel. Simply feed your SQL into your Gem's prompt so that it could reformat it for you. Once, submitted, your Gem will return a reformatted SQL that you can copy right from its response, using the copy icon above the returned SQL code.

Gemini

Cognos to Sybase SQL Converter

### Cognos to Sybase SQL Converter

Precise Cognos-to-Sybase Transformer (Strict)

Recent

- SQL Transformation for Sybase ASE
- SQL Transformation and Optimization
- SQL Optimization for StateDW

```
SELECT
  "Fund".Fund_Name AS C0,
  "Appropriation_Unit".Appropriation_Name AS C1,
  LTRIM(RTRIM("Account_Type".Description)) AS C2,
  "Department0".Department_Name AS C3,
  "Object0".Object_Class_Name AS C4,
  "Object0".Object_Name AS C5,
```

Tools Fast

Gemini

SQL Transformation for StateDW Sybase

SELECT

```
SELECT
  "Fund".Fund_Name AS C0,
  "Appropriation_Unit".Appropriation_Name AS C1,
  LTRIM(RTRIM("Account_Type".Description)) AS C2,
  "Department0".Department_Name AS C3, ...
```

Cognos to Sybase SQL Converter - Custom Gem

Based on your requirements for the StateDW Sybase ASE environment, I have transformed the Cognos SQL by stripping quotes, applying explicit three-part naming, setting up the requested parameters, and optimizing the join structure for performance while maintaining logical equivalence.

```
SQL
DECLARE @Department varchar(3)
DECLARE @FiscalYear varchar(4)
DECLARE @FiscalPeriod varchar(2)

SET @Department = '000' -- Placeholder Value
SET @FiscalYear = '2024' -- Placeholder Value
SET @FiscalPeriod = '01' -- Placeholder Value
```

Tools Fast

Your State of Utah chats aren't used to improve our models. Gemini is AI and can make mistakes. [Your privacy & Gemini](#)

Having that reformatted SQL copied, navigate to **Excel**. Go to the **Data** section on the navigation bar, select **Get Data**, select **From Other Sources**, and then select **From ODBC**. A From ODBC window should appear where you can specify your **Data Source Name (DSN)**. Select **StateDW** as your DSN. If StateDW is not available under the DSN list, this is because you do not have the proper driver installed for this connection - you will need to reach out to DTS Finance so that they can install this driver on your device.

Below the DSN option, there is an **Advanced options** dropdown - expand it. In the Advanced options dropdown, you will find a text box where you can insert your SQL statement - paste your reformatted SQL statement there. In that SQL statement, there is likely a **SET @** statement that holds parameter or filter values. Enter the values you would like for each parameter within single quotes, as shown below. Then, click **OK** at the bottom right of the window.

The screenshot shows the Microsoft Excel interface with the **Data** tab selected. The **Get Data** button is highlighted in the ribbon. The **From Other Sources** > **From ODBC** path is shown in the **Get Data** menu. The **From ODBC** dialog box is open, showing the following configuration:

- Data source name (DSN):** StateDW
- Advanced options:** Expanded
- Connection string (non-credential properties) (optional):** Example: Driv...
- SQL statement (optional):**

```
DECLARE @Department varchar(3)
DECLARE @FiscalYear varchar(4)
DECLARE @FiscalPeriod varchar(2)

SET @Department = '150' -- Placeholder Value
SET @FiscalYear = '2025' -- Placeholder Value
SET @FiscalPeriod = '05' -- Placeholder Value

SELECT
Fund.Fund_Name AS C0,
Appropriation_Unit.Appropriation_Name AS C1,
LTRIM(RTRIM(Account_Type.Description)) AS C2,
Department0.Department_Name AS C3,
```
- Supported row reduction clauses (optional):** (None) Detect
- Buttons:** OK, Cancel





After having clicked **Edit**, the Power Query Editor window will appear. In this window, on the right-hand side, under **Query Settings** there are **Applied Steps**. Double-click on the **Source** step under Applied Steps. Clicking the Source step should produce the **From ODBC** window where your SQL can be adjusted with new parameters or with new nicknames for data fields. For example, in the SQL below, the nickname for Fund.Fund\_Name is C0, this could be changed to Fund\_NM or Fund\_Name, or any other name that has no spaces. Once you are done making adjustments, click **OK** on the From ODBC window, and then click Close & Load in the Power Query window.

(Note: Fund.Fund\_Name tells us that the data we are pulling from in the StateDW is the Fund table's data field called Fund\_Name)

The screenshot displays the Microsoft Power Query Editor interface. The main window shows a data table with columns labeled C0 through C4. The 'Query Settings' pane on the right shows the 'Applied Steps' section with 'Source' selected. A 'From ODBC' dialog box is open, showing the 'Data source name (DSN)' set to 'StateDW'. The 'SQL statement (optional)' field contains the following code:

```

DECLARE @Department varchar(3)
DECLARE @FiscalYear varchar(4)
DECLARE @FiscalPeriod varchar(2)

SET @Department = '150' -- Placeholder Value
SET @FiscalYear = '2025' -- Placeholder Value
SET @FiscalPeriod = '05' -- Placeholder Value

SELECT
    Fund.Fund_Name AS C0,
    Appropriation_Unit.Appropriation_Name AS C1,
    LTRIM(RTRIM(Account_Type.Description)) AS C2,
    Department0.Department_Name AS C3,

```

The 'Supported row reduction clauses (optional)' dropdown is set to '(None)'. The 'OK' button is highlighted with a red box.

You now know how to get from a Cognos report to an Excel replacement! With the data available to you in your spreadsheet, you can build pivot tables, filter, sort, and explore your data with more hands-on control than ever before!

