

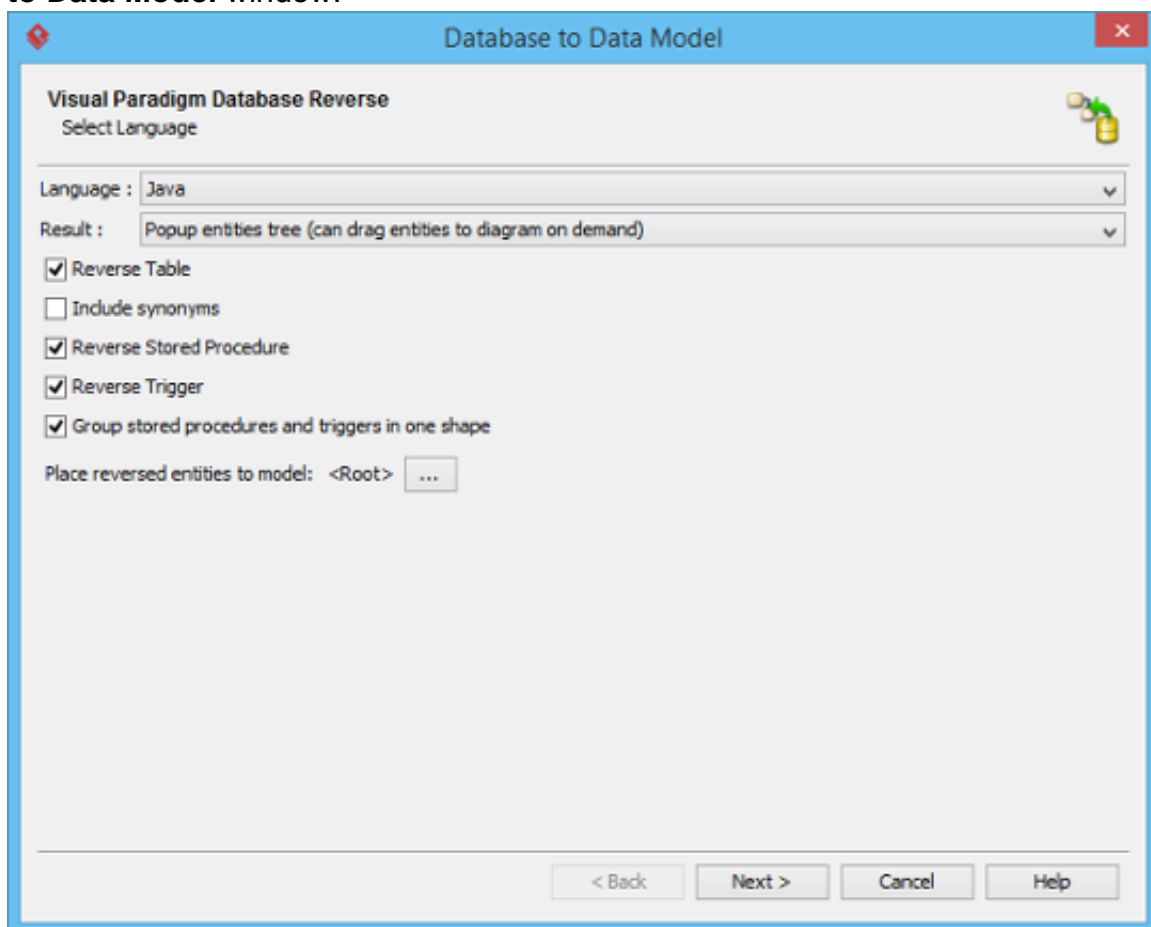
---

## How to reverse engineer ERD from database

Visual Paradigm supports [reverse engineering ERD from existing database](#). By visualizing a database schema in [ERD](#), you can re-edit it and patch changes back to the database, or you can produce a data specification that acts as a reference for programmers who need to access the database. Besides, Visual Paradigm automates [object-relational mapping](#), object model can thus be generated from the data model produced by reverse engineered.

To reverse engineering ERD from database:

1. Select **Tools > DB > Reverse Database...** from the toolbar. This opens the **Database to Data Model** window.



2. In the **Database to Data Model** window, set the language, which is the method to use in connecting to your database. By selecting Java, JDBC will be used. By selecting C#, adapter file will be used. If you are uncertain about this option, just leave it unchanged.
3. Select the Result, which is the post-action of database reversal. Here is a list of available options:

Result	Description
Popup entities tree	

You will be presented a window with reversed entities listed. You can form ERD(s) manually by dragging and dropping entities from that window onto diagram(s).

Form a new diagram with the reversed entities

Automatically create an ERD and place all the reversed entities in it. Make sure you

---

Result

Description

are clear about the amount of entities that will be reversed and the complexity of diagram that will be formed. Having too many entities on one diagram may result in a poor database design that is hard to read and maintain.

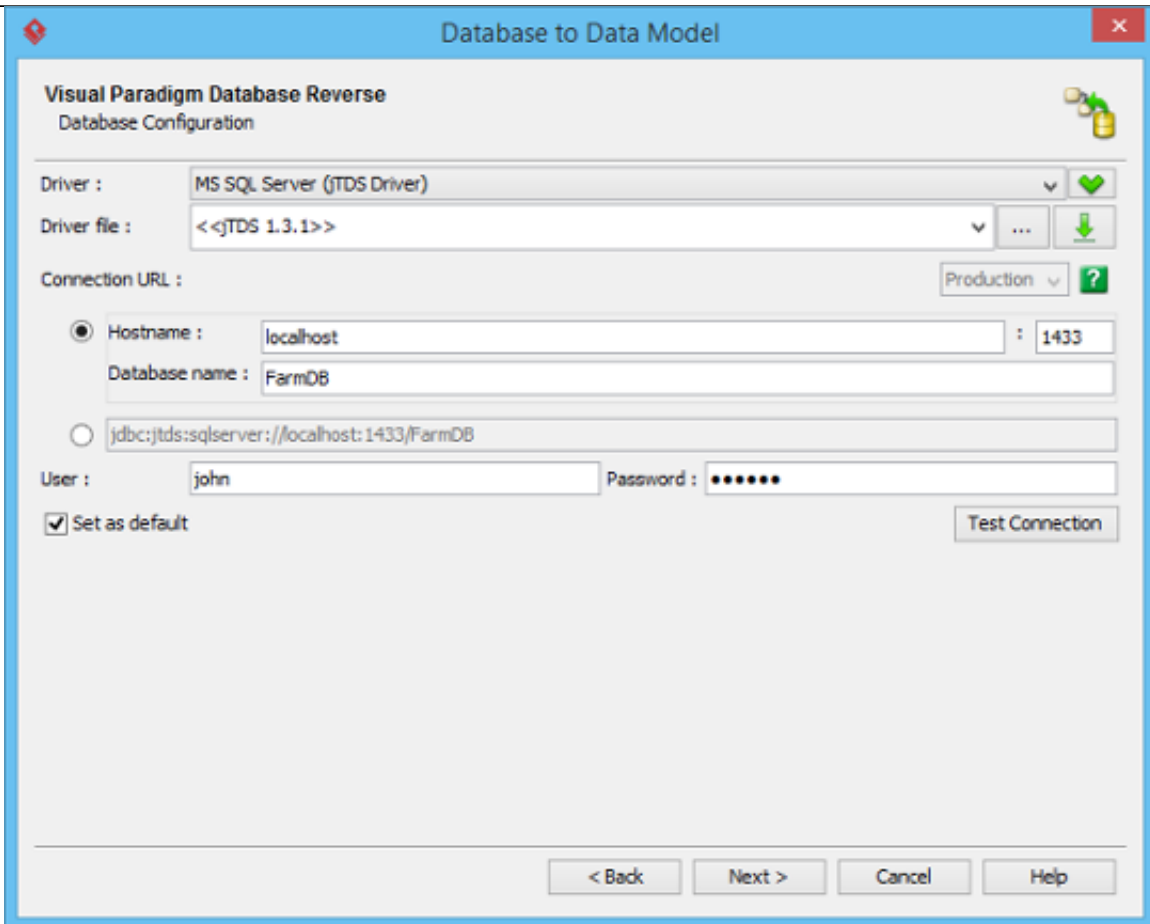
Add reversed entities to ERD

Automatically place all the reversed entities onto the first ERD in your project. Again, make sure the amount of entities to be reversed and the complexity of diagram is under your control.

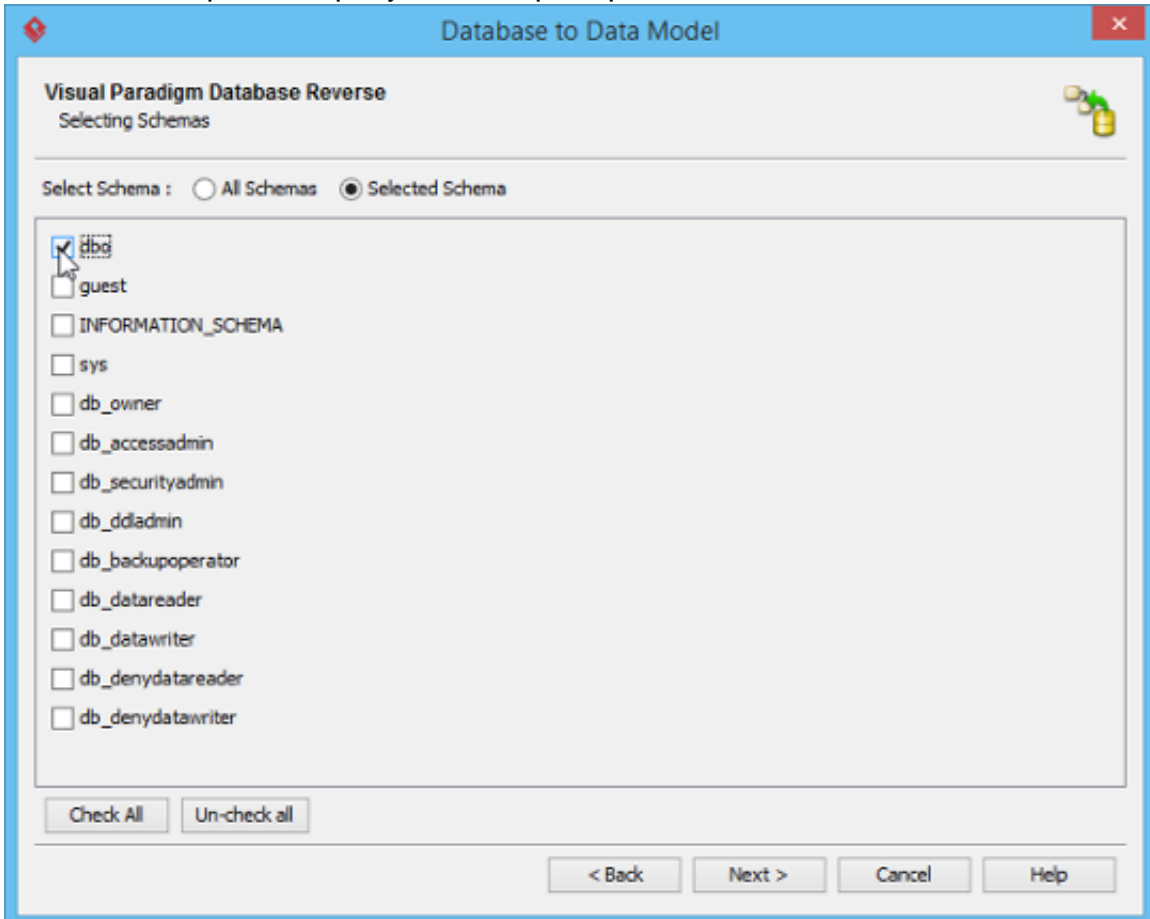
Do not form diagram with the reversed entities

Just produce entities without visualizing them with any ERD. You can visualize them later on if you want to.

4. Sometimes, you may just want to reverse engineer stored procedures or triggers. If that's the case, uncheck **Reverse Table**.
5. Synonym is an alternative name for objects such as tables, view, etc. If you are going to reverse an Oracle database, you can optionally check **Include synonyms** to reverse synonyms as entities.
6. Keep **Reverse Stored Procedure** checked if you want to restore stored procedures.
7. Keep **Reverse Trigger** checked if you want to restore database triggers.
8. By checking **Group stored procedures and triggers in one shape**, we will create at most one stored procedures shape and one triggers shape to hold all the reversed procedures and triggers. When unchecked, one procedures shape will be created for each procedure being reversed.
9. You can also place the reversed entities into a dedicated model by clicking ... next to **Place reversed entities to model** to select the model in which you want the entities be placed.
10. Click **Next**.
11. Configure the database connection and click **Next**.



12. For DBMS like MSSQL, you can select the schema to reverse. Make your choice and click **Next**. Skip this step if you aren't prompted to select schema.



13. By default, all database tables in the specified database (or schema) will be reversed. If you want to reverse just some tables, select the tables you do not want and click