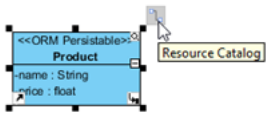
Define custom implementations for ORM Persistable class

Written Date : March 01, 2016

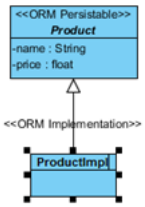
We assume you already have Visual Paradigm Standard installed and integrated with Eclipse. SQL Server should also be set up and ready to use. Suppose we have a simple ORM class, **Product**, and we want to define extra features to calculate the product price after a volume discount:

1. Click on the *Product* class and drag out the resource icon.

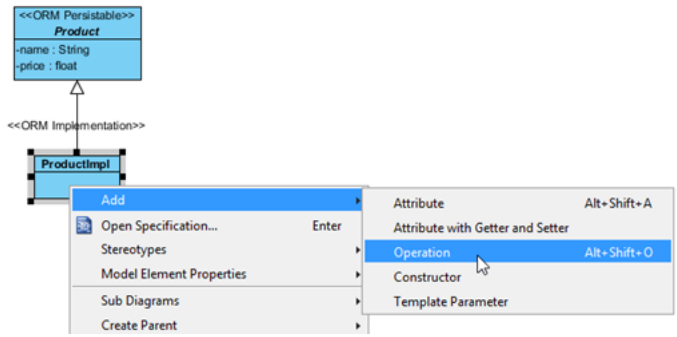


2. Release the mouse button at the desired location in the diagram.
3. Select **Create ORM Implementation Class** in the **Resource Catalog**.

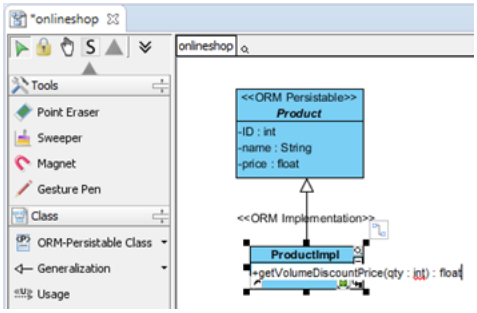
4. Name the created class as *ProductImpl*.



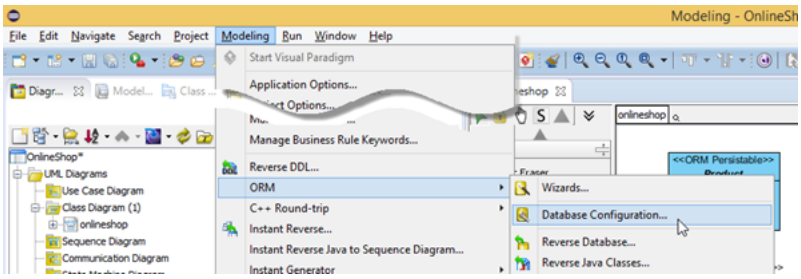
5. Right-click on *ProductImpl* and select **Add > Operation**.



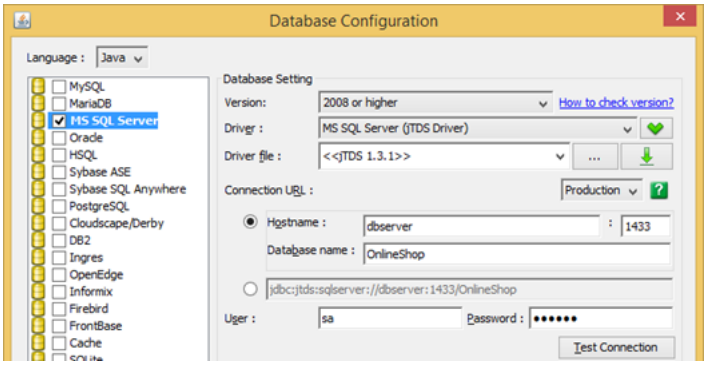
6. Name the operation as *getVolumeDiscountPrice(qty : int) : float*.



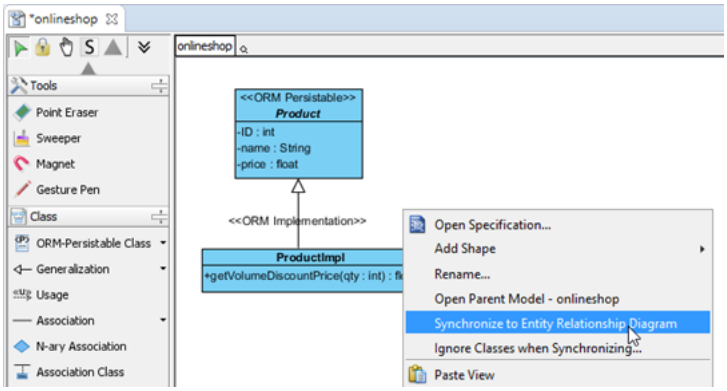
7. Now the model is ready, and we can proceed to generate the ER model and Hibernate code. First, we define the default database for our project. Select **Modeling > ORM > Database Configuration...**



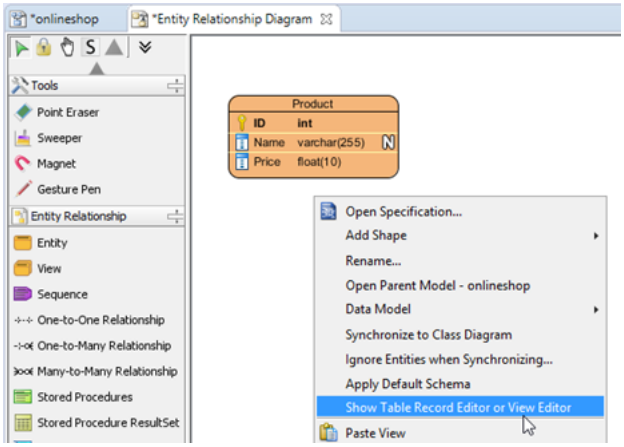
8. Select **SQL Server** as our default database. Specify the connection details to the SQL Server. Use the **Test Connection** button to make sure your configuration details work. Then, press **OK** to close the **Database Configuration** dialog.



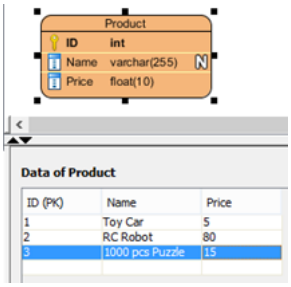
9. Right-click on the blank area of the class diagram and select **Synchronize to Entity Relationship Diagram**. Follow the wizard to perform the synchronization with default settings.



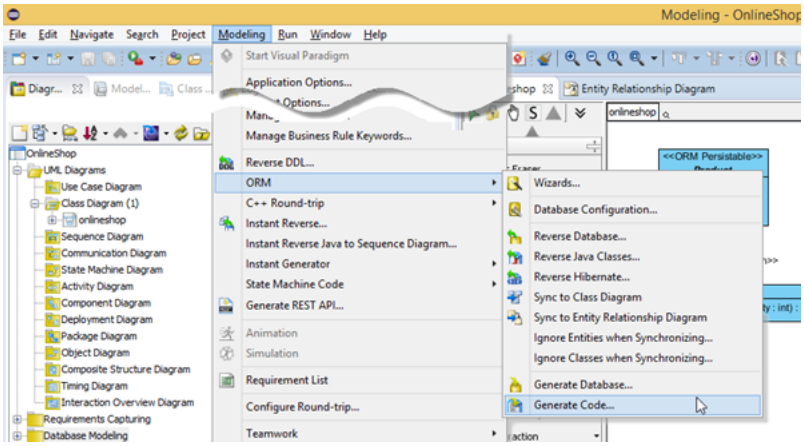
10. The ERD for our model is generated. To simplify testing, we can predefine some sample data for our database. Right-click on the blank area of the ERD and select **Show Table Record Editor or View Editor**.



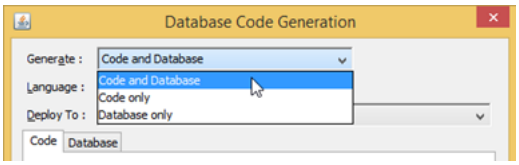
11. Enter the sample record below into the **Table Record Editor**.



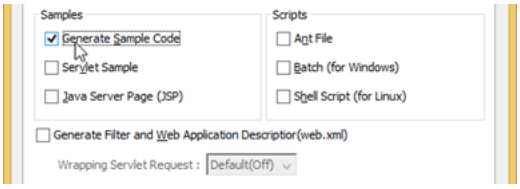
12. We are almost there. Select **Modeling > ORM > Generate Code...**



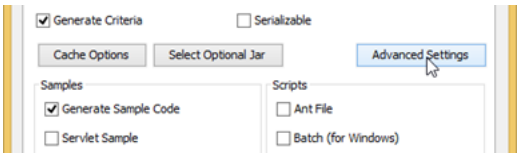
13. In the **Database Code Generation** dialog, select to generate **Code and Database**.



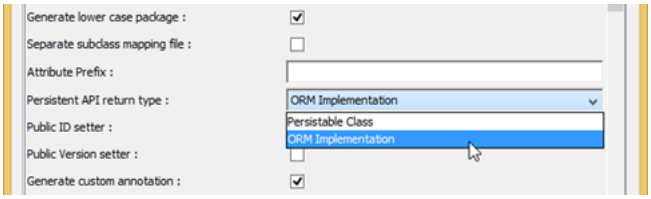
14. Select **Generate Sample Code** so that we can test the implementation with the generated testing program.



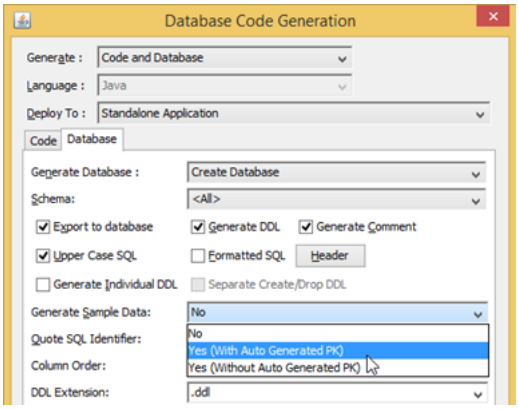
15. Press the **Advanced Settings** button.



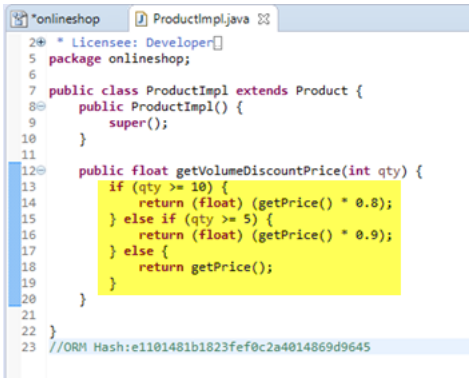
16. In the **Advanced Settings** dialog, select **ORM Implementation** as the **Persistent API return type**.



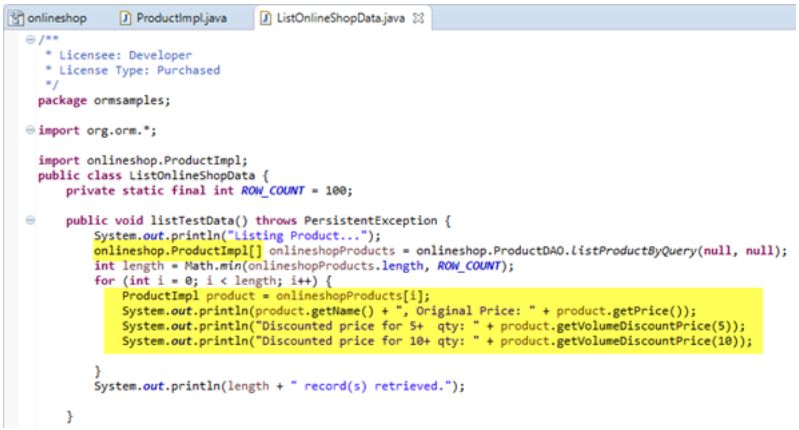
17. Switch to the **Database** tab and select **Export to database**. Select **Yes (With Auto Generated PK)** in **Generate Sample Data**. Press **OK** to proceed with code generation.



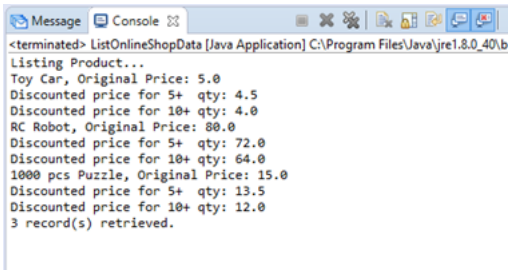
18. Now the Hibernate code is being generated, and we can start to define the custom implementation in the getVolumeDiscountPrice method. For example, we offer 10% off for purchases of 5+ copies, and 20% off for purchases of 10+ copies.



19. Done. Let's modify the printout section in the list data sample to test our custom implementation.



20. Run the modified list data sample, and the custom implementation method will be called.



Related Links

- Tutorial - Generate Hibernate Mapping for Oracle database
- Tutorial - Begin UML Modeling in Eclipse

- Tutorial - Working with Hibernate in Eclipse
- User's Guide - Eclipse Integration
- What is Entity Relationship Diagram (ERD)?

Visual Paradigm home page
(https://www.visual-paradigm.com/)

Visual Paradigm tutorials
(https://www.visual-paradigm.com/tutorials/)