ER to SQL (generic / Oracle) transformations

On this page

Both generic and Oracle transformation flavors are very similar, so they will be described together. These transformations are based on and are very similar to the UML to SQL(Generic / Oracle) transformations with several extensions relevant to ER modeling.

This chapter only describes the extended behavior of ER to SQL(Generic / Oracle) transformations.

The full transformation feature set includes:

- conversion of many-to-many relationships into an intermediate table
- three different methods of transforming generalizations into table layouts
- autogenerating primary keys, unique constraints and indexes
- generating additional tables for multivalue attributes
- type remapping between UML and database worlds
- sequence generation
- package hierarchy flattening

For more information, see UML to SQL transformation.

> **Note**
> Please note that the SQL model produced by the transformation is usually not optimal (e.g. all generalizations are transformed using the same chosen strategy, while different strategies are chosen for each particular case at the discretion of the DBA). It is frequently advisable to refine / edit the produced model after the transformation.

Identifying relationships

Identifying relationships are transformed the same way as the UML to SQL transformation. The foreign key of the transformation will be included into the primary key of the dependent entity (the one at the multiple end of the relationship). The difference in an ER to SQL transformation case is that the ER model eliminates guessing which relationships are identifying and which ones are not. In UML to SQL transformation guesses, which UML associations should identify using a heuristic method, composition associations are treated as identifying. This heuristic is controlled by the **Treat compositions as identifying** transformation property. In ER models, identifying relationships are explicitly marked as such; there is no need to guess. “Identifying Relationships and Dependent Entities” on page 9 specifies how identifying relationships are modeled.

Key transformation

Keys in ER models are transformed into constraints in a DDL model. The rules for key transformations into DDL constraints are:

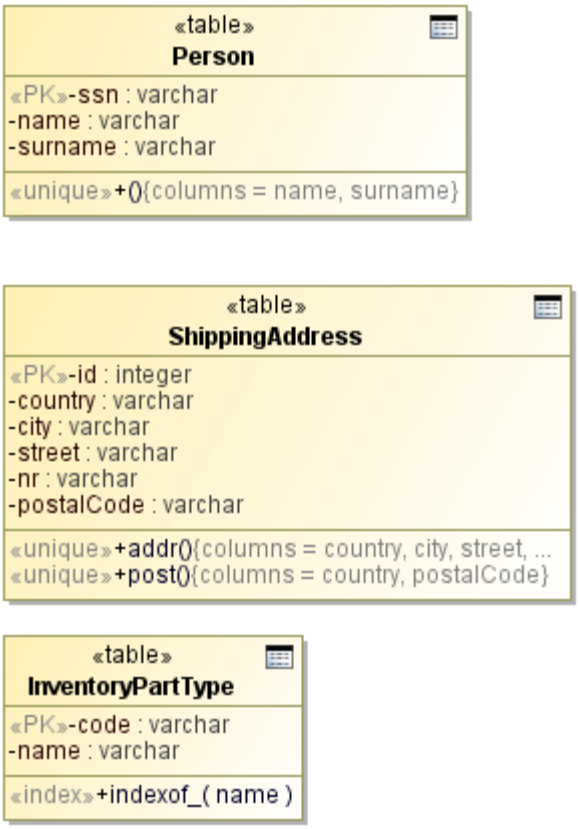
1. The Primary key of the entity in the ER model is transformed into a primary key constraint in the SQL model.

2. The Alternative keys of the entities in the ER model are transformed into unique constraints in the SQL model.

3. The Inversion entries of the entities in the ER model are transformed into indexes in the SQL model.

4. If a key or entry in ER model has a name (**identifier** tag), this information is preserved in the SQL model. The corresponding key / index will also have a name in the SQL model.

Let's review an example of key modeling, described in Key modeling. After the transformation, the three entities of the ER model are transformed into the three tables of the SQL model, respectively.



TBD screenshot example of key transformation results.

Virtual entity transformation

Virtual entities of ER models can be transformed into different elements of SQL models as:

- Tables (ordinary, non-virtual entities).
- SQL views (the ER to SQL(Oracle) transformation has an additional choice of simple views or materialized views).

The choice is controlled by the **Virtual Entities Transformed To** transformation property.

Tracing between data model layers

After the transformation, a relationship is established between the logical data model layer, represented by the ER model, and the physical data model layer, represented by the SQL model. You can navigate between the connected elements both forward (ER -> SQL) and backward (SQL -> ER) using the dedicated submenu **Go To** on the element's shortcut menu.

To go forward to the corresponding element

---

1. Right-click the element.
2. On its shortcut menu, click **Go To** > **Traceability** > **Model Transformations** > **Transformed To**.

To go backward to the corresponding element

---

1. Right-click the element.
2. On its shortcut menu, click **Go To** > **Traceability** > **Model Transformations** > **Transformed From**.

The same tracing information is visible in the element's Specification window and **Properties** panel under the **Traceability** tab. This information is also reflected in the Entity-Relationship and SQL Report using navigable references between the report sections. You can also depict traceability information in a relation map or in a tabular format using the capabilities of the custom dependency matrix feature.