Design Your Databases Straight to the Cloud

Heli Helskyaho, Elise Valin-Raki

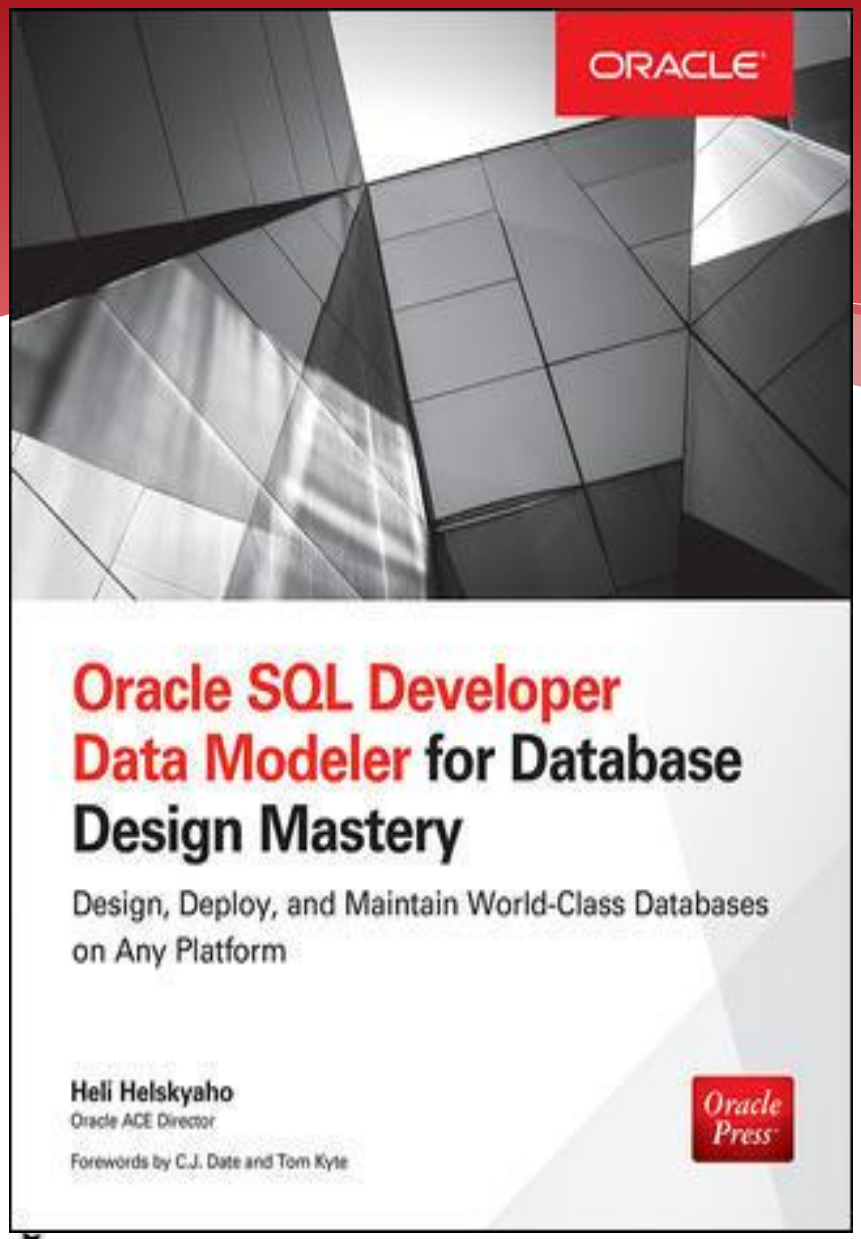
MIRACLE
Miracle Finland Oy

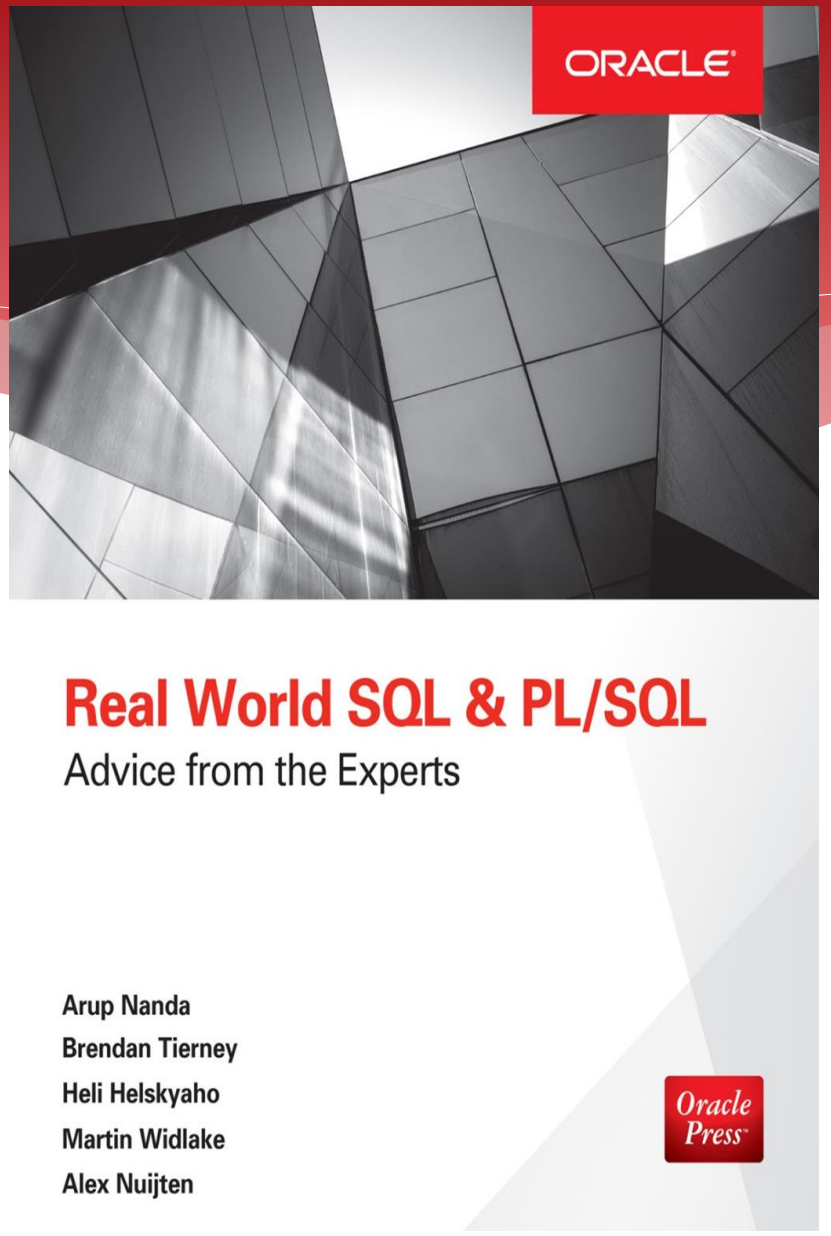
Introduction, Heli

* Graduated from University of Helsinki (Master of Science, computer science), currently a doctoral student, researcher and lecturer (databases, Big Data, Multi-model Databases, methods and tools for utilizing semi-structured data for decision making) at University of Helsinki
* Worked with Oracle products since 1993, worked for IT since 1990
* Data and Database!
* CEO for Miracle Finland Oy
* Oracle ACE Director
* Ambassador/EOUC (EMEA Oracle Users Group Community)
* Public speaker and an author
* Winner of Devvy for Database Design Category, 2015
* Author of the book Oracle SQL Developer Data Modeler for Database Design Mastery (Oracle Press, 2015), co-author for Real World SQL and PL/SQL: Advice from the Experts (Oracle Press, 2016)

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy





MIRACLE
Miracle Finland Oy

Helskyaho Heli, Valin-Raki Elise

Copyright © Miracle Finland Oy

Introduction, Heli

* Been an Oracle Designer user since 1996 and Data Modeler user since 2010
* Been solving performance problems since 1993
  * Absolutely convinced that good database *design* and *documentation* will *reduce* this work.
  * Absolutely convinced that it is *easier* to solve problems on database that has been *documented*

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

Introduction, Elise

* Oracle Database Lead for Fennia Insurance Company
* Master of Science degree from Helsinki High School of Economics (with major in Management Science)
* IT sector since 1987, different positions
* OUGF (Oracle User Group Finland):
  * Exa-SIG founder
* Oracle ACE Associate
* DBA

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

500+ Technical Experts Helping Peers Globally

ORACLE ACE PROGRAM

---

3 Membership Tiers

- Oracle ACE Director
- Oracle ACE
- Oracle ACE Associate

bit.ly/OracleACEProgram

Connect:

oracle-ace_ww@oracle.com

Facebook.com/oracleaces

@oracleace

ORACLE Developer Community

Nominate yourself or someone you know: **acenomination.oracle.com**

Why to design?

* ”Data is the most valuable property in our company”
* ”Why do we need to design the database? We already design the application!”

Miracle Finland Oy

Copyright © Miracle Finland Oy

Why is designing the application not enough?

* Point of view (saving and retrieving data vs. UI)
* First increment vs. 20 years from now
* ”the whole picture” vs. increments
* Different goals/targets:
  * Code tables vs. Code files (how about the data integrity?)
  * How about analysis, reports, … everything else but the UI that the data is used for
* Same terminology, different meaning -> misunderstandings
* …

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

Why to model the data?

* To facilitate communication about the requirements
* To find the questions that should be asked
* To understand the requirements

Copyright © Miracle Finland Oy

How is Cloud different?

* It has a bigger chance to have latency (usually a hybrid solution)
  * Good reason to add as much business logic to the database as possible!
  * But you would do that anyway ☺
* Everything you have learned about database designing still applies….

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

What is database designing?

* 4 (5) phases, over and over again
  * Requirement analysis (DM: logical)
  * Conceptual design (DM: logical)
  * Logical design (DM: relational)
  * Physical design (DM: physical)
  * (Transaction design) (DM: process)

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

It would be crazy…

* …to design databases without a tool!

MIRACLE
Miracle Finland Oy

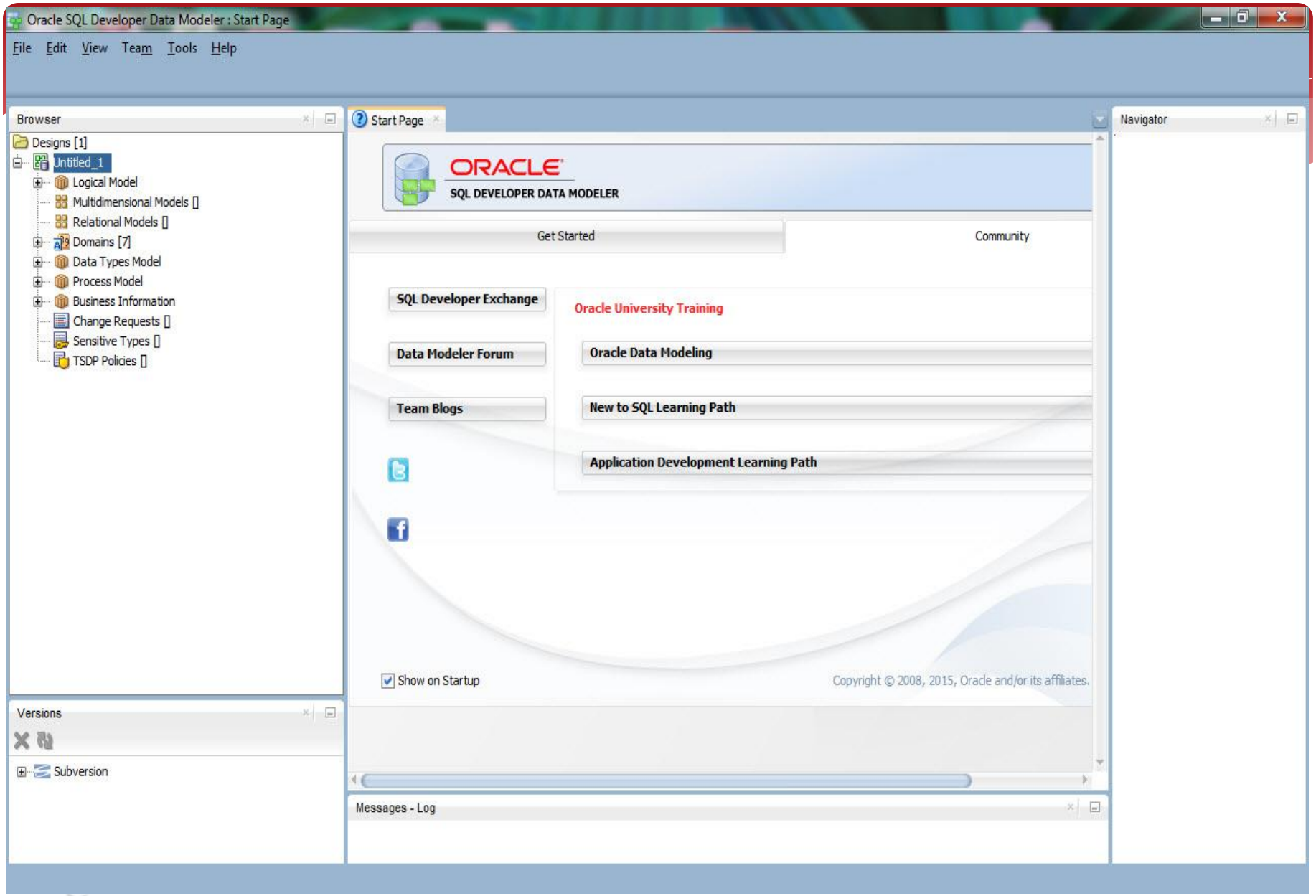
Copyright © Miracle Finland Oy

Oracle SQL Developer Data Modeler

* To be efficient in designing you need a tool: my recommendation is Data Modeler
  * Free of charge
  * Support for many different databases (Oracle, MS SQL Server, DB2,…)
  * Support for both documenting the existing databases and designing a new one (and mainintaining that)
  * Support for reporting, naming standards, glossaries, design rules, ...
  * Support for version control and multiuser environment
  * Support for everything you need for database design plus more

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy



MIRACLE
Miracle Finland Oy

Designing the database

1. **Requirement analysis:** finding and analysing the requirements the future end users have

Result: specification of user requirements

- **data** requirements
- **functional** requirements

Also requirements for security, performance, …

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

Designing the database

2. **Conceptual design**. ”Interpretation” of all the requirements to a formal presentation (conceptual model).

Result: conceptual schema, also textual documentation is possible/recommended (to make sure all the knowledge is documented)

This is a tool for communication with end users.

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

Requirement analysis and Conceptual design

* Collecting requirements and analyzing them
* Fact-finding: interviews, questionnaires, existing documentation,… (recordings)
* Requirements specifications
  * data requirements
  * functional requirements (performance, security, backup/recovery,..)
* Completely neutral to any technology

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

Requirement analysis and Conceptual design

* Why entity-relationship model (ER)
  * Defining the tables directly based on requirements can be too difficult and lead to a wrong db schema.
  * Based on a good ER it is easy to generate the relational model (which is at least on 3NF)
* Data Flow Diagrams (DFD)

Copyright © Miracle Finland Oy

Conceptual design

* Use right **terminology** and clear names, much easier to communicate with the end users (one of the reasons to model!)
* Try to find and understand the **main** concepts and their relationships (these are the most difficult to change during the iterations)

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

Conceptual design

* Modeling is **difficult** because
  * Spoken/written language is not exact
  * Usually all the ”important” things are those that ”everybody knows” so they are not told.
  * At this stage we do not know one important thing: how the data will be **retrieved**? That will be on iteration 9…
* Modeling is **mandatory** because when modeling the database designer realizes **what must be asked**!

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

Logical model

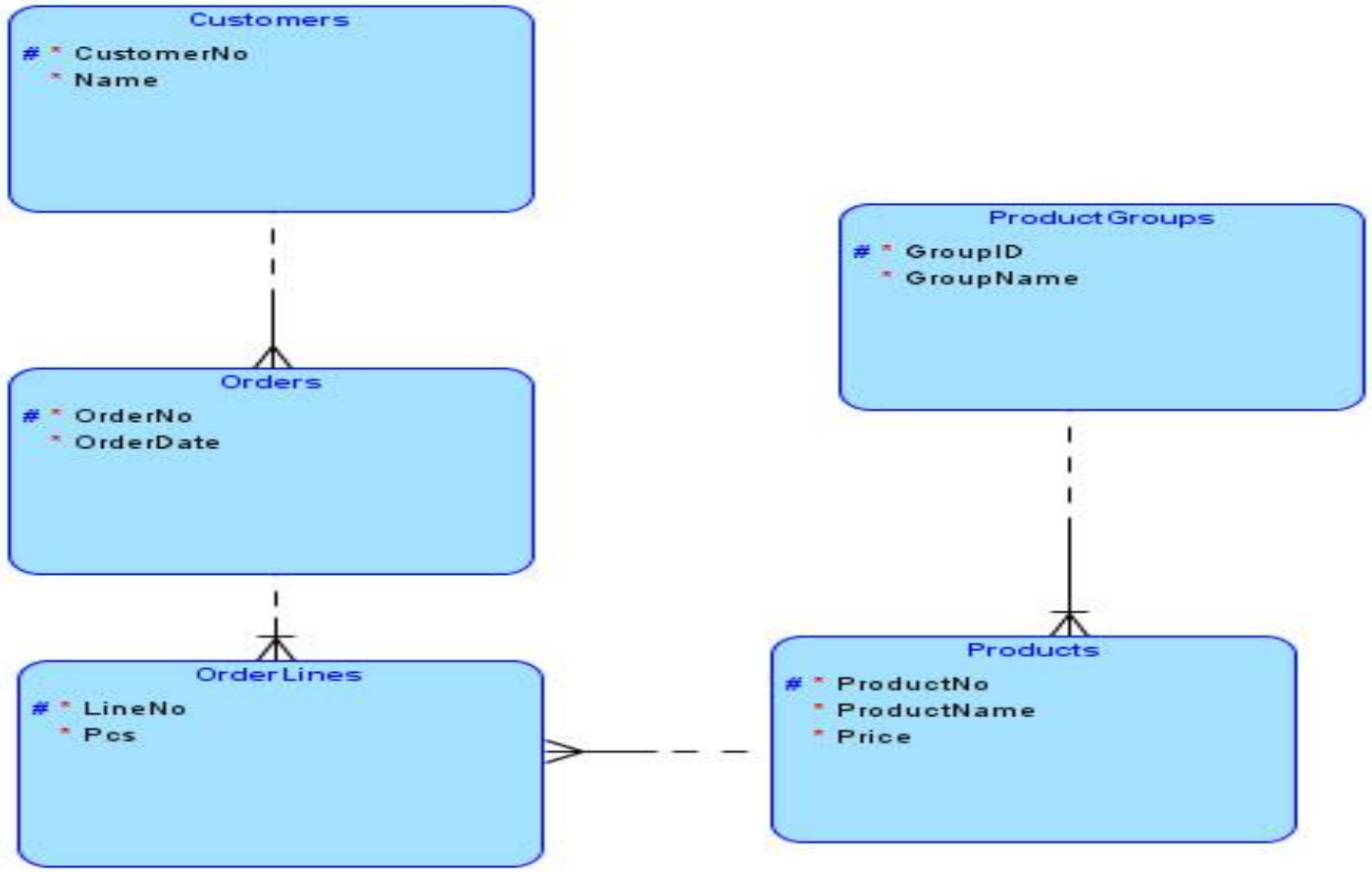
Miracle Finland Oy

Copyright © Miracle Finland Oy

Demo (Logical)

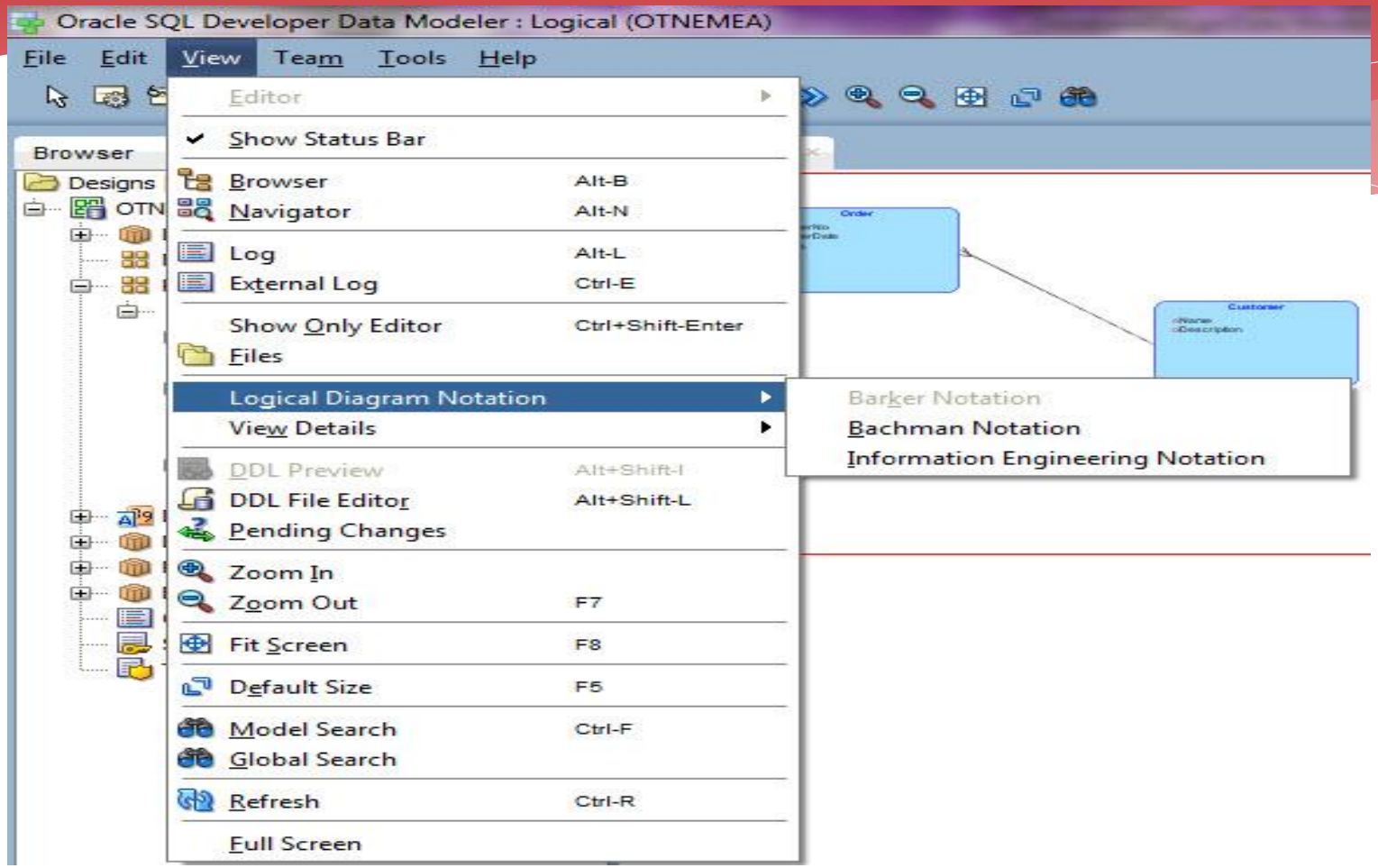
MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy



MIRACLE
Miracle Finland Oy

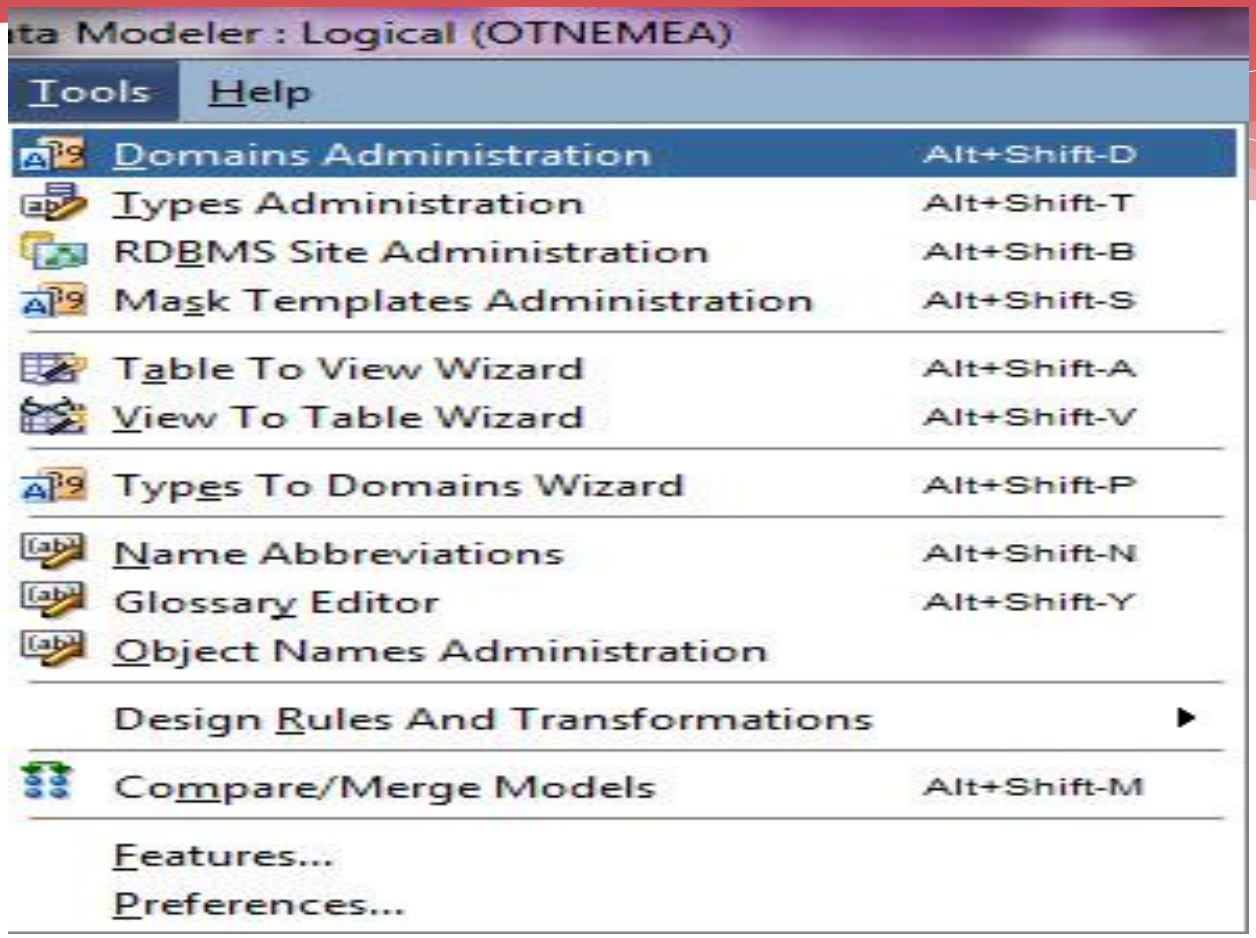
Notation



MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

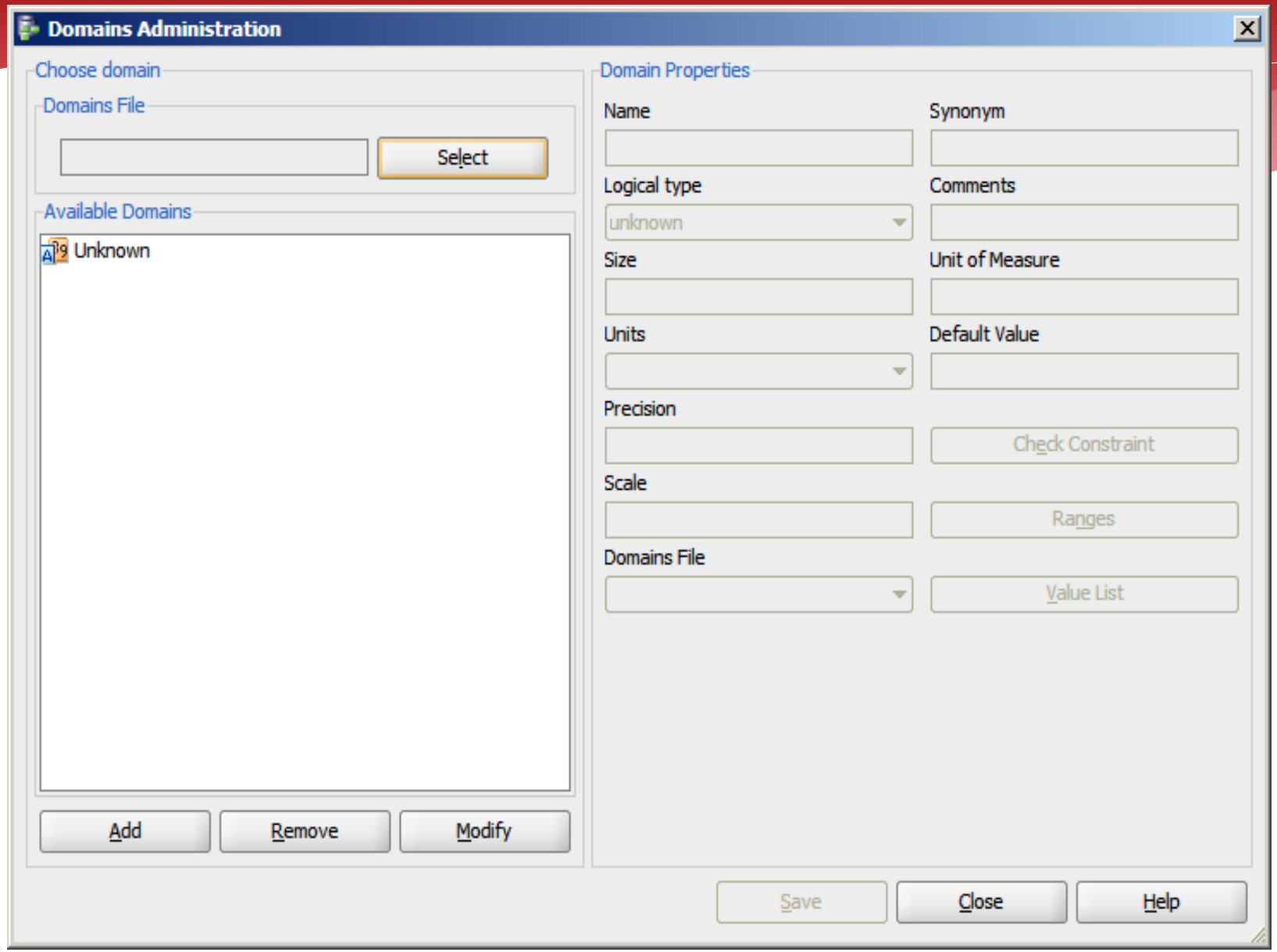
Domains administration



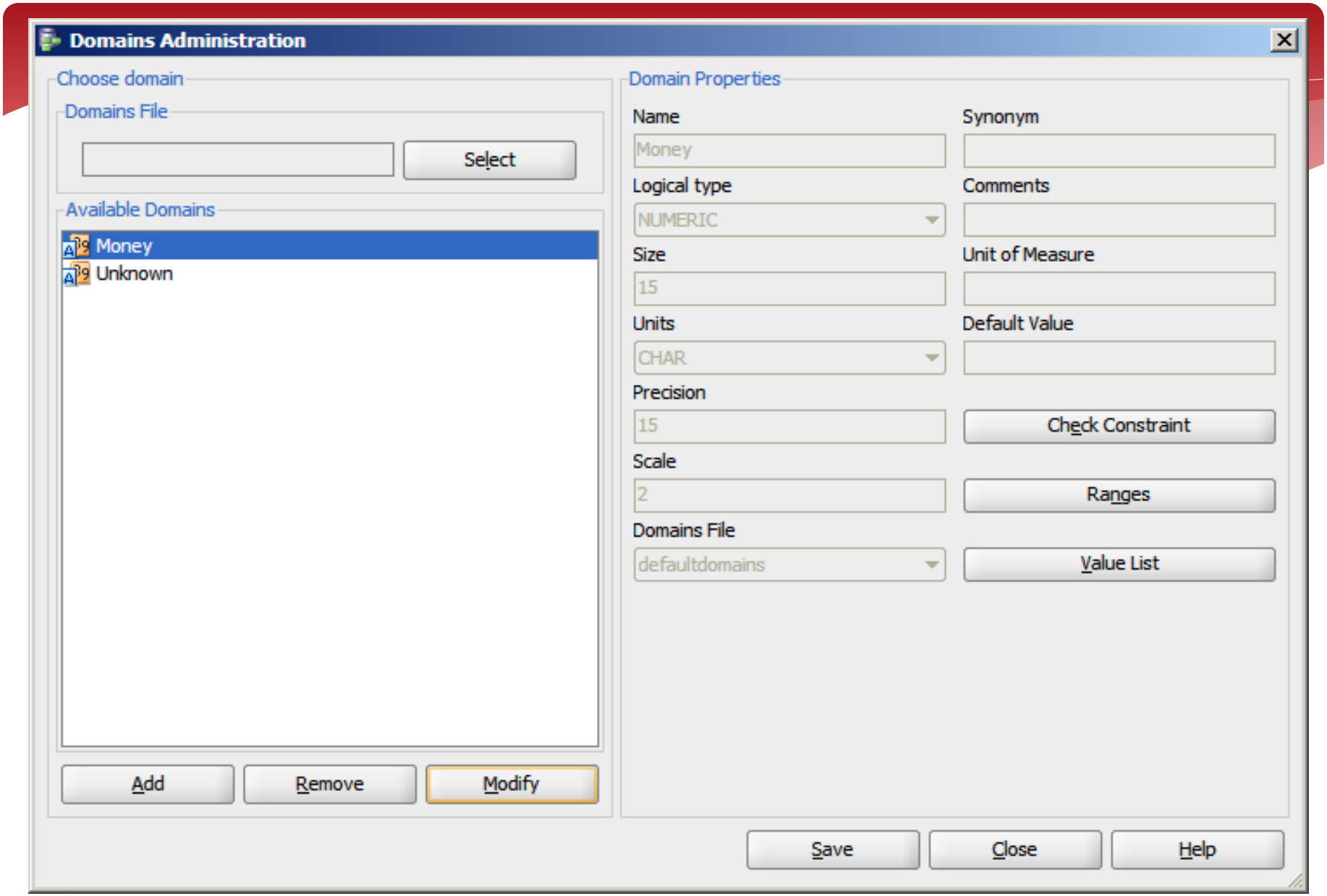
MIRACLE
Miracle Finland Oy

© Miracle Finland Oy

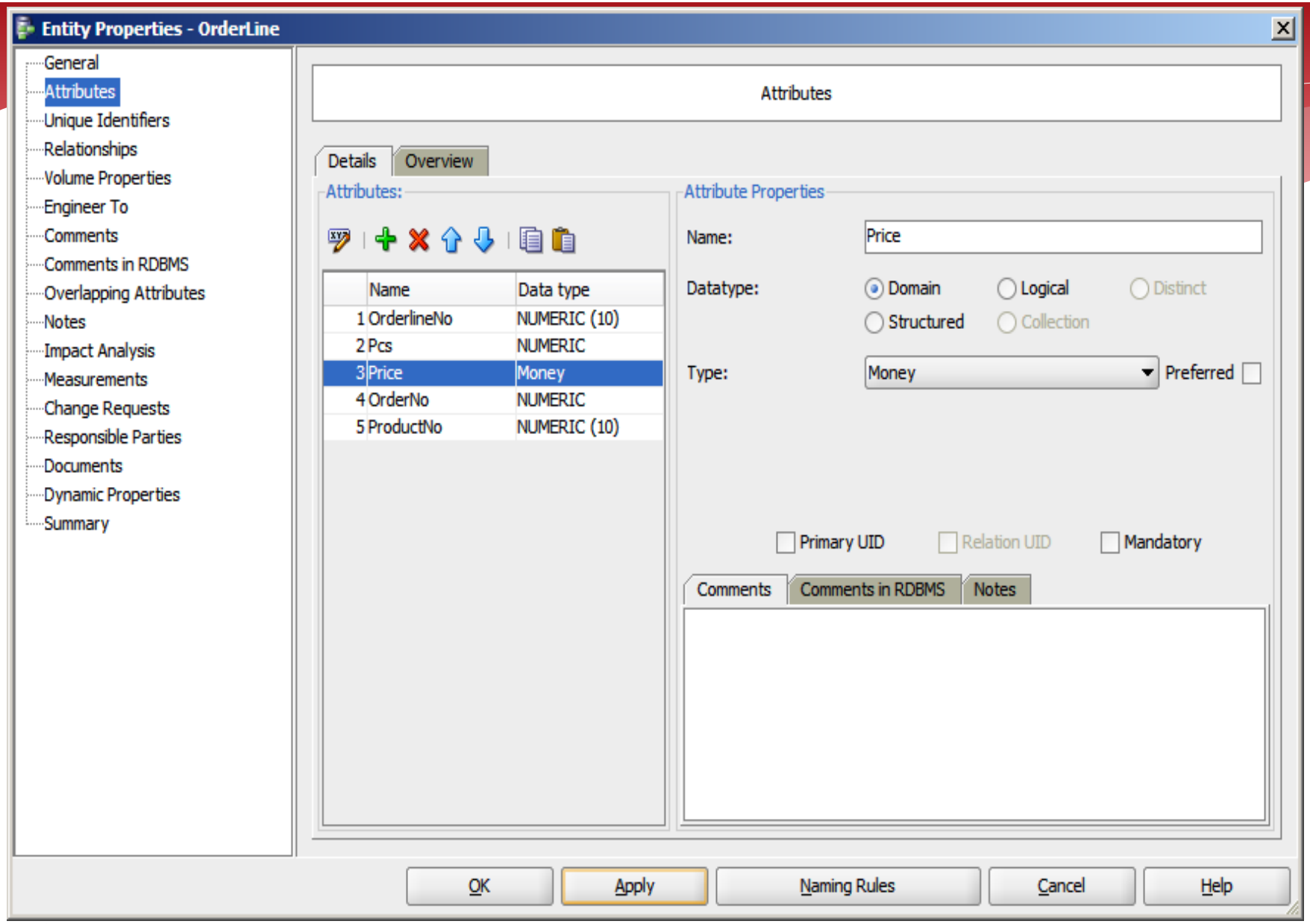
Copyright © Miracle Finland Oy



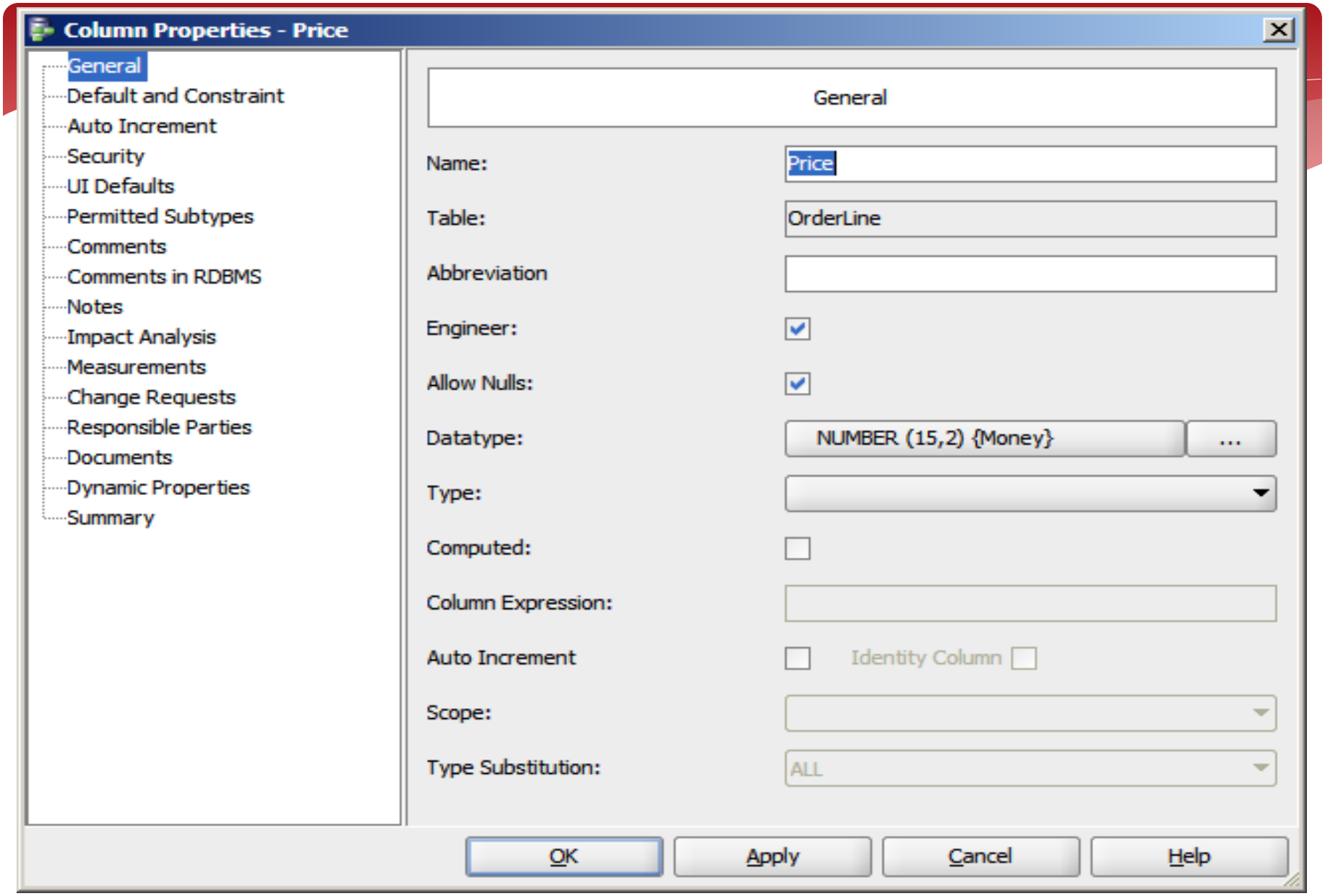
© Miracle Finland Oy



© Miracle Finland Oy



© Miracle Finland Oy

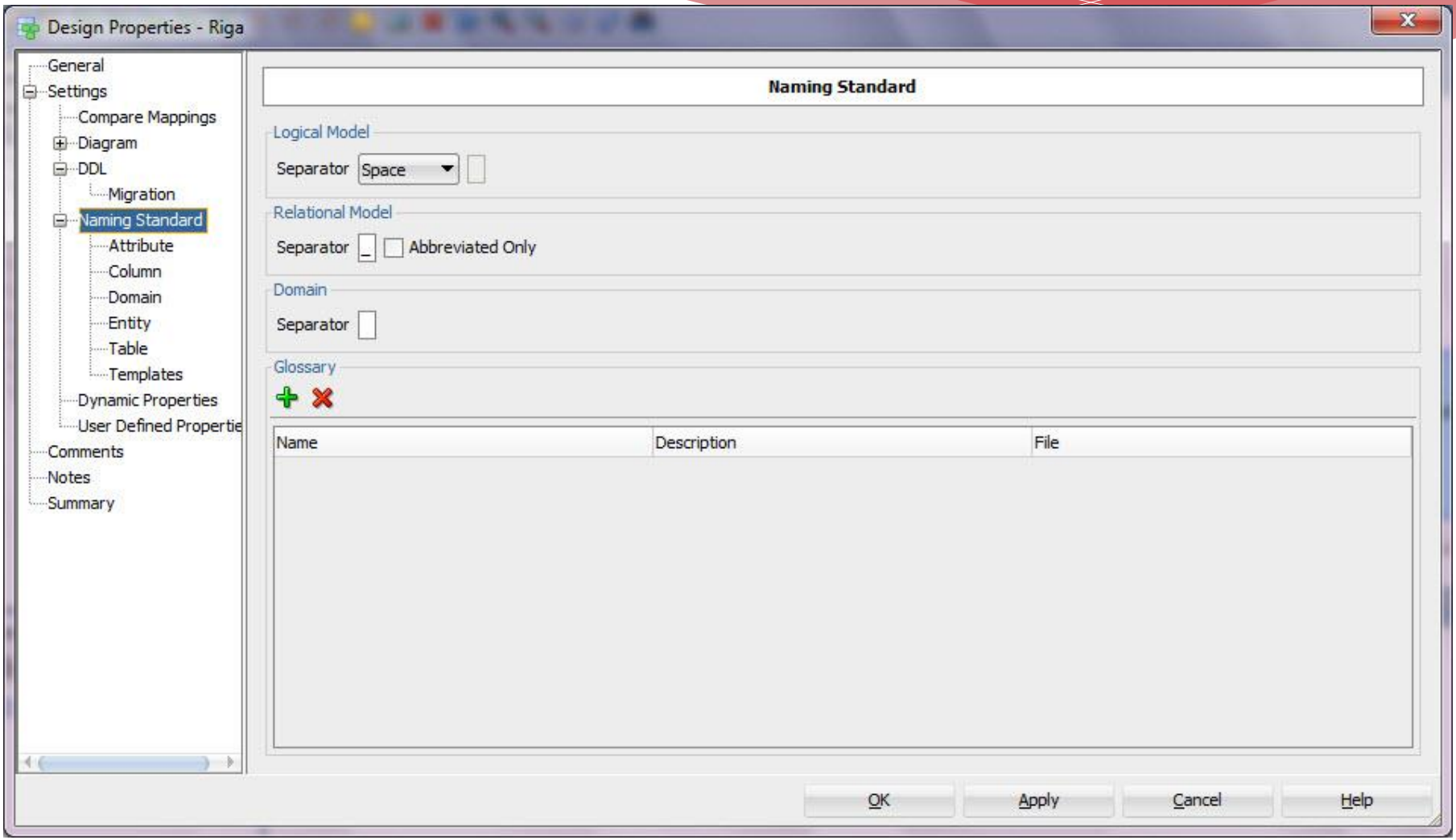


© Miracle Finland Oy

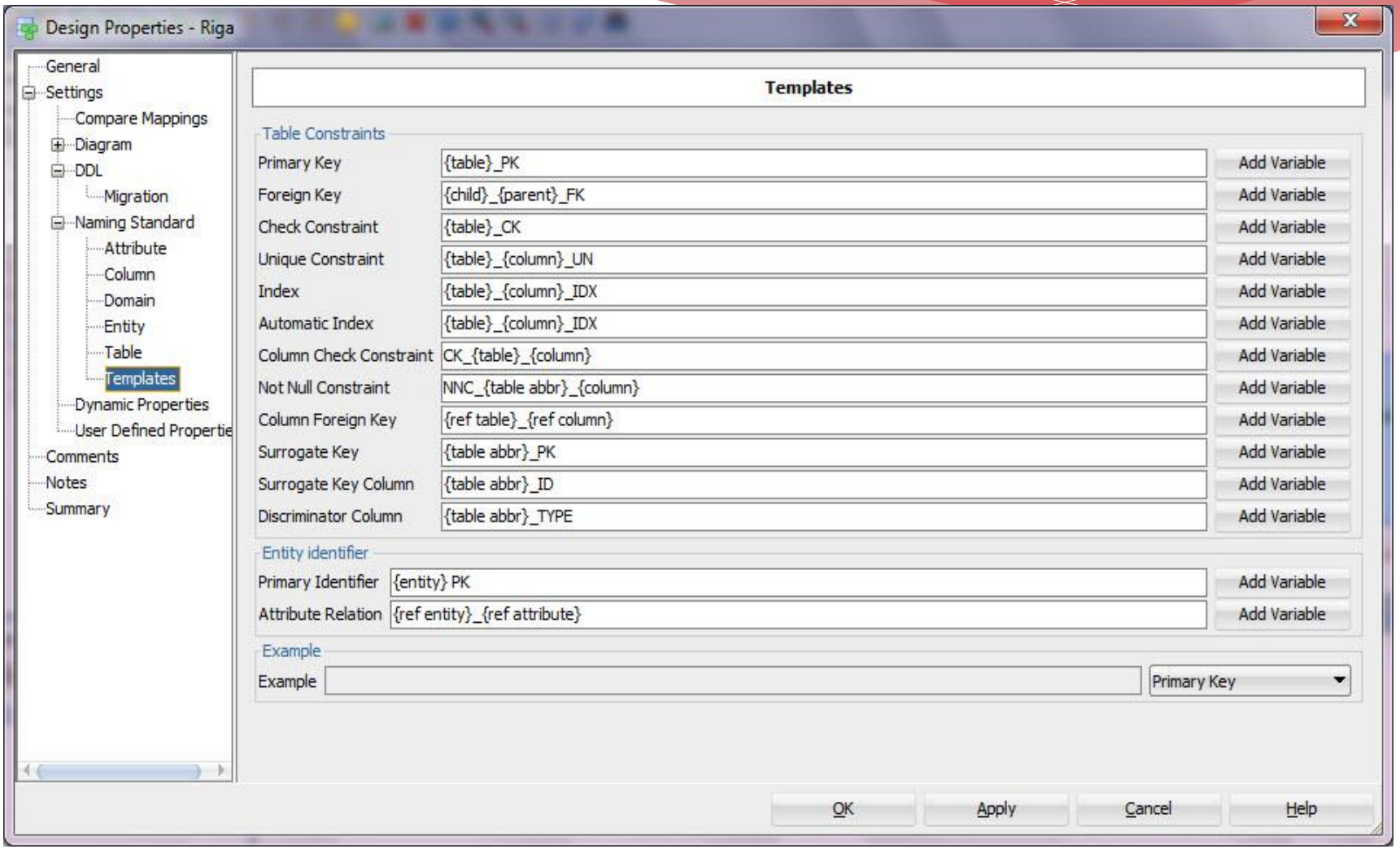
Design Properties

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

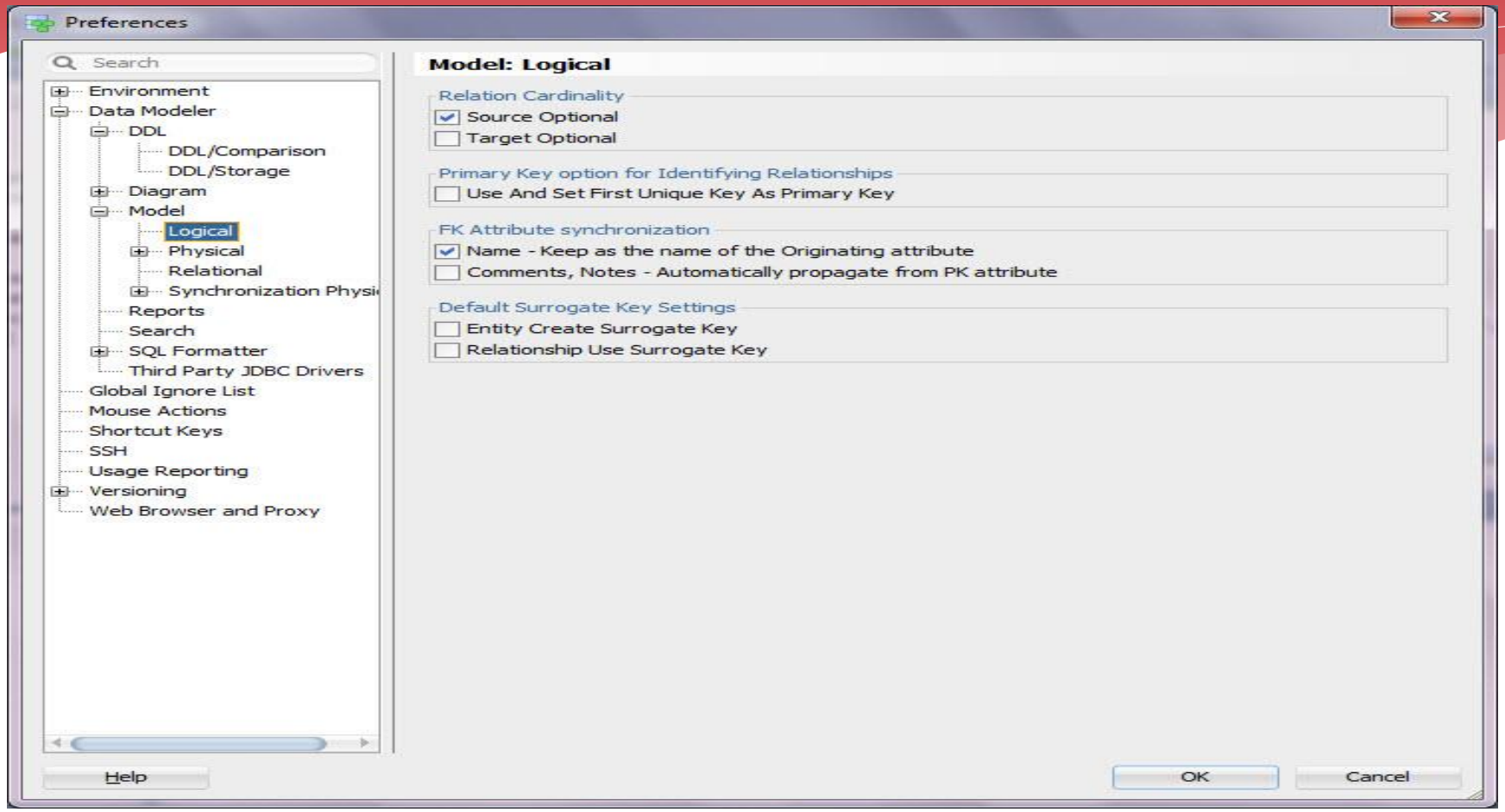


MIRACLE
Miracle Finland Oy



© Miracle Finland Oy

Preferences



© Miracle Finland Oy

Copyright © Miracle Finland Oy

Designing the database

3. **Logical design:** transforming the conceptual model into a logical data model and a logical schema that the RDMS understands

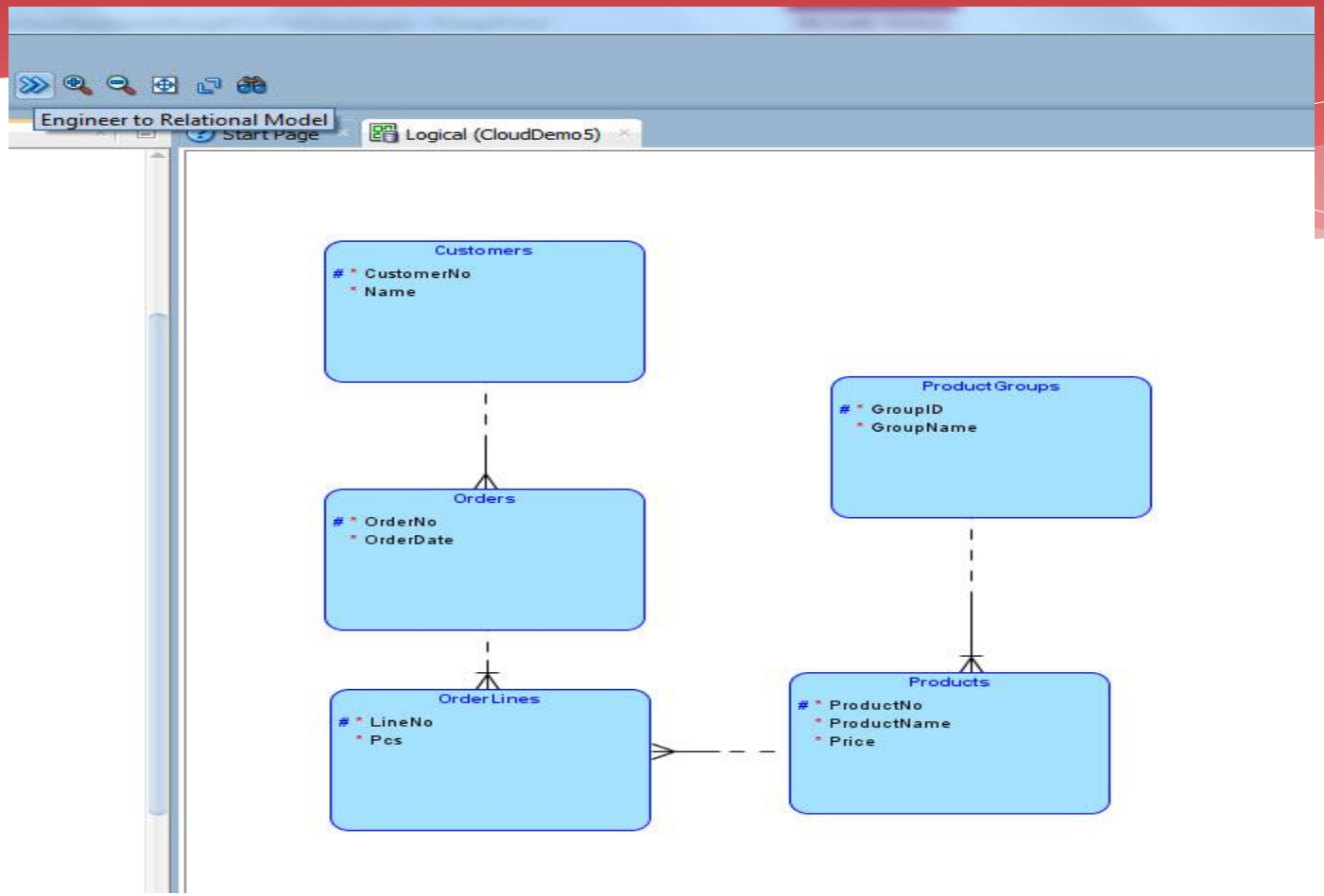
Result: relational-database schema

(relational schemas and constraints)

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

From logical to relational



MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

Demo (Relational)

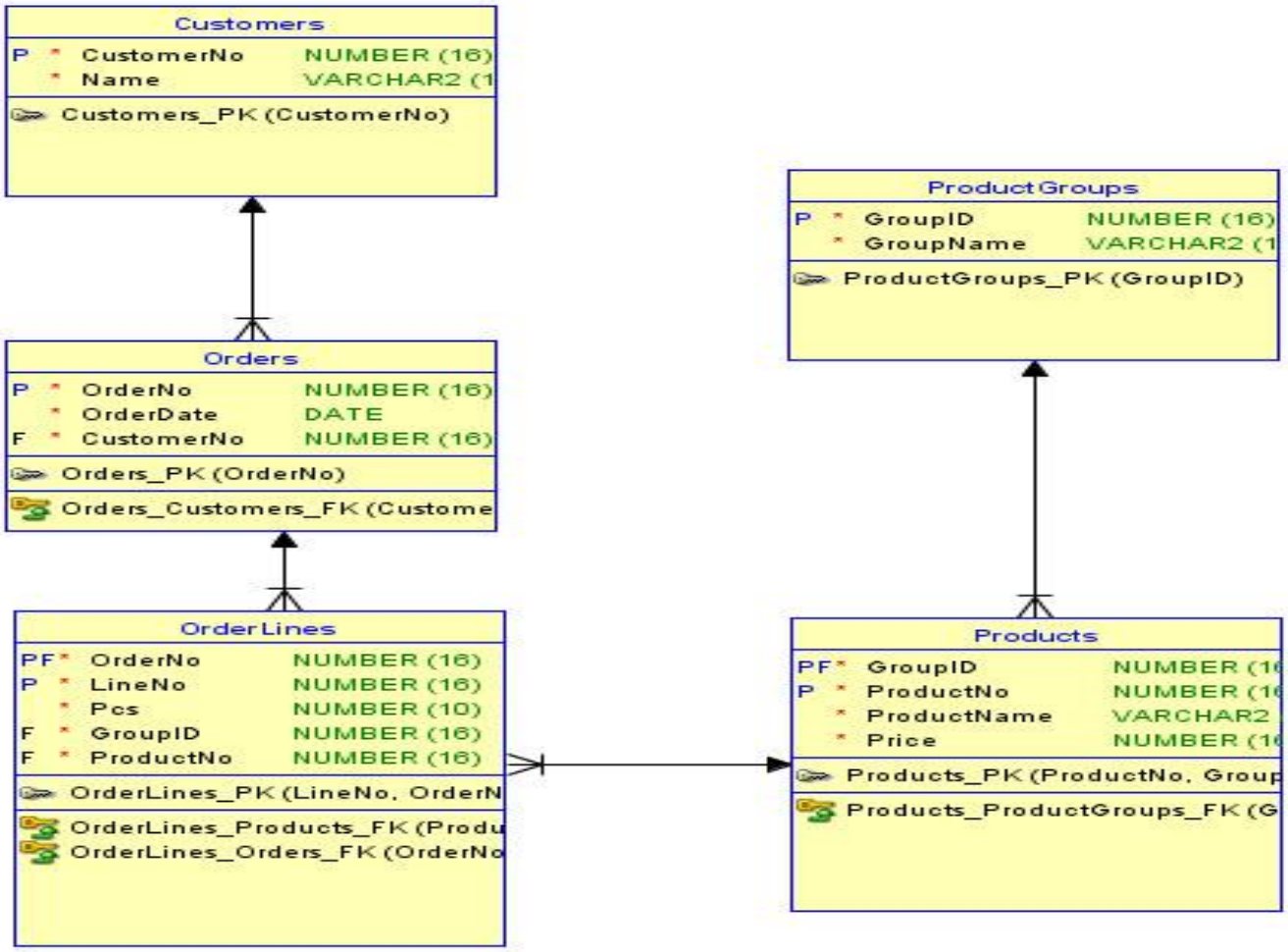
MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

Step 1

* Check that you got what you wanted…

Copyright © Miracle Finland Oy

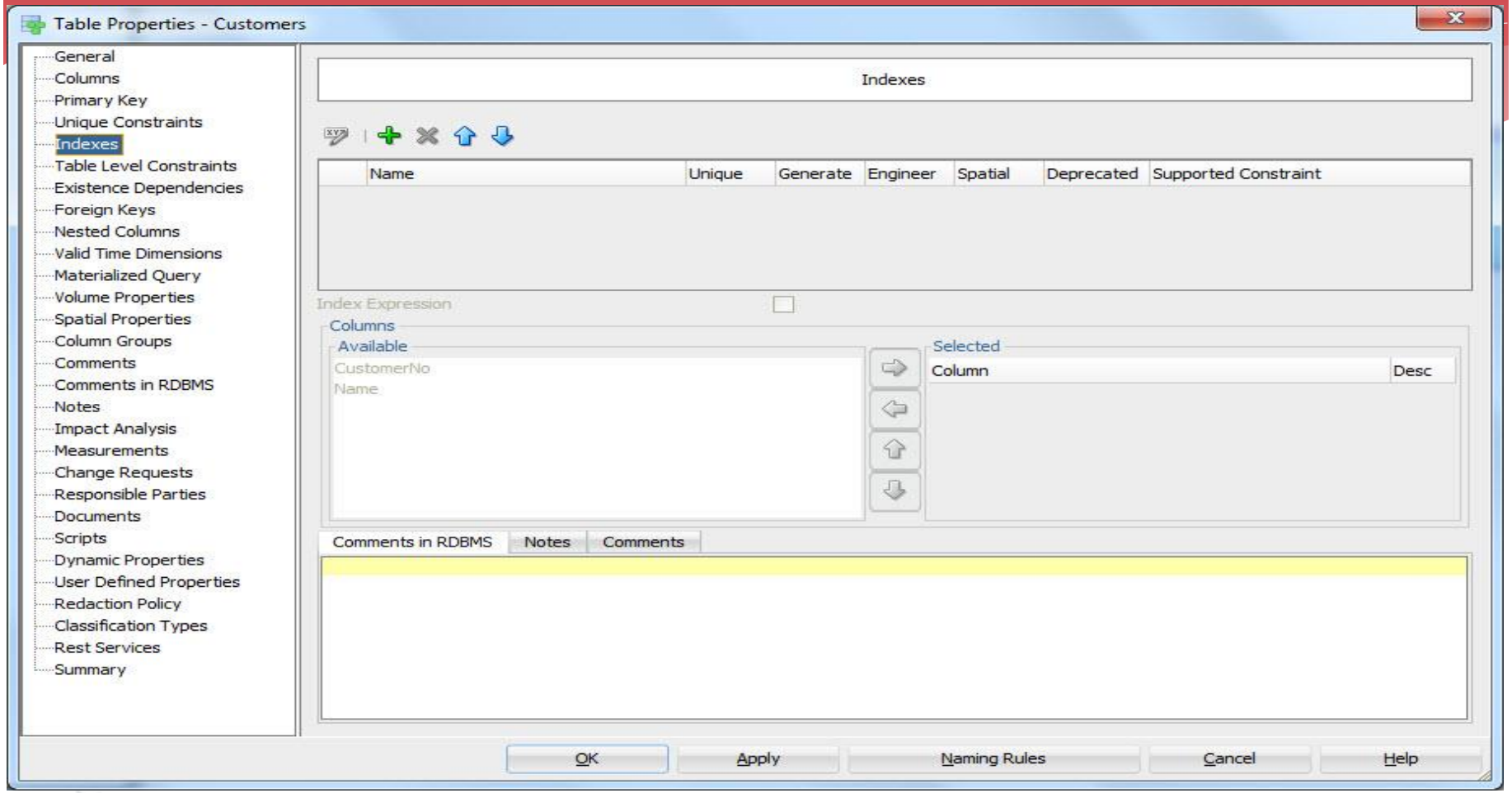


MIRACLE
Miracle Finland Oy

Indexes

Copyright © Miracle Finland Oy

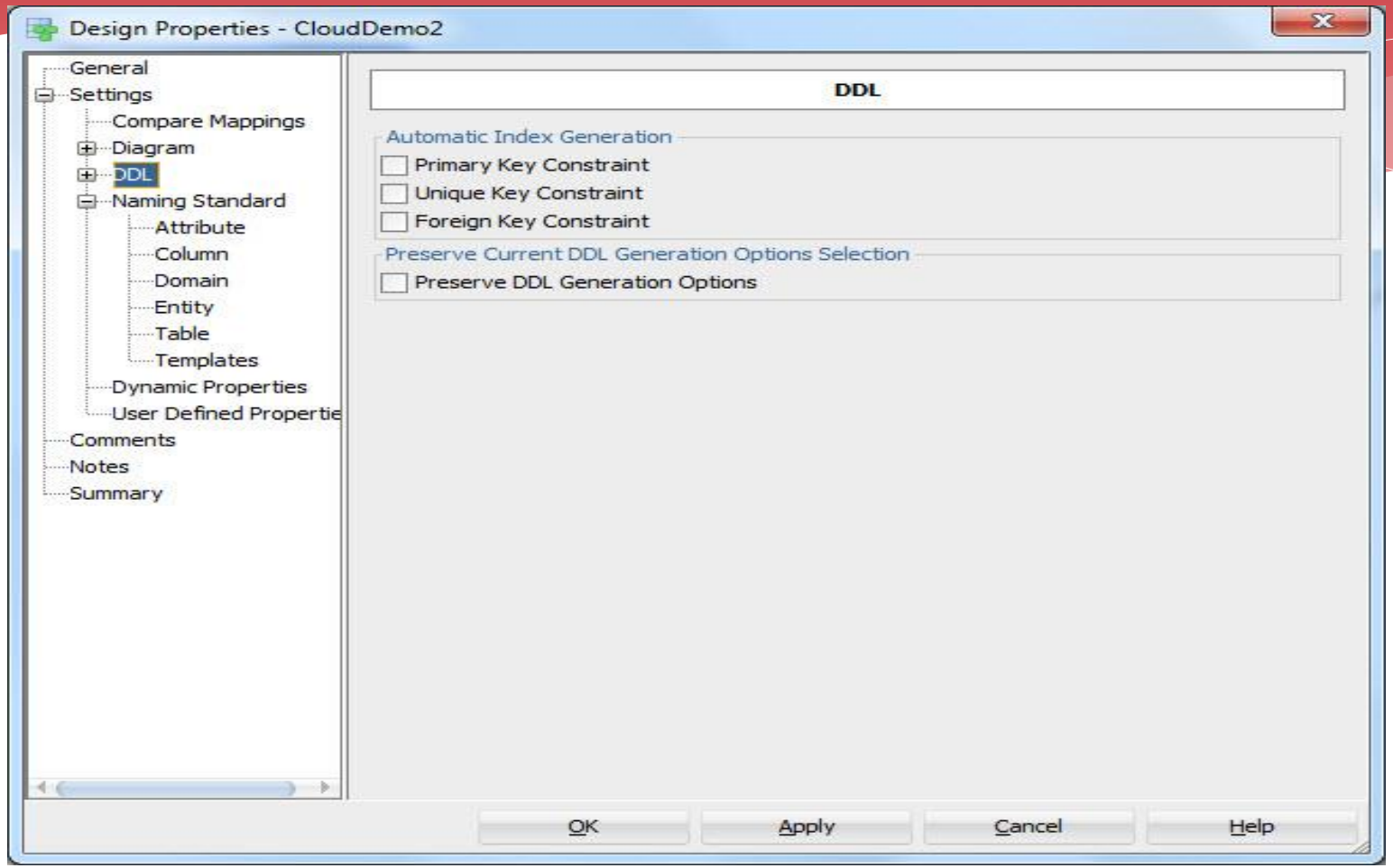
Manually



MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

Automatically



MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

Designing the database

4. **Physical design**: instances, tablespaces, indexes, disks …

And all of these phases over and over again…

(this is different, we have always done that but not so many times and in such a short cycles)

Copyright © Miracle Finland Oy

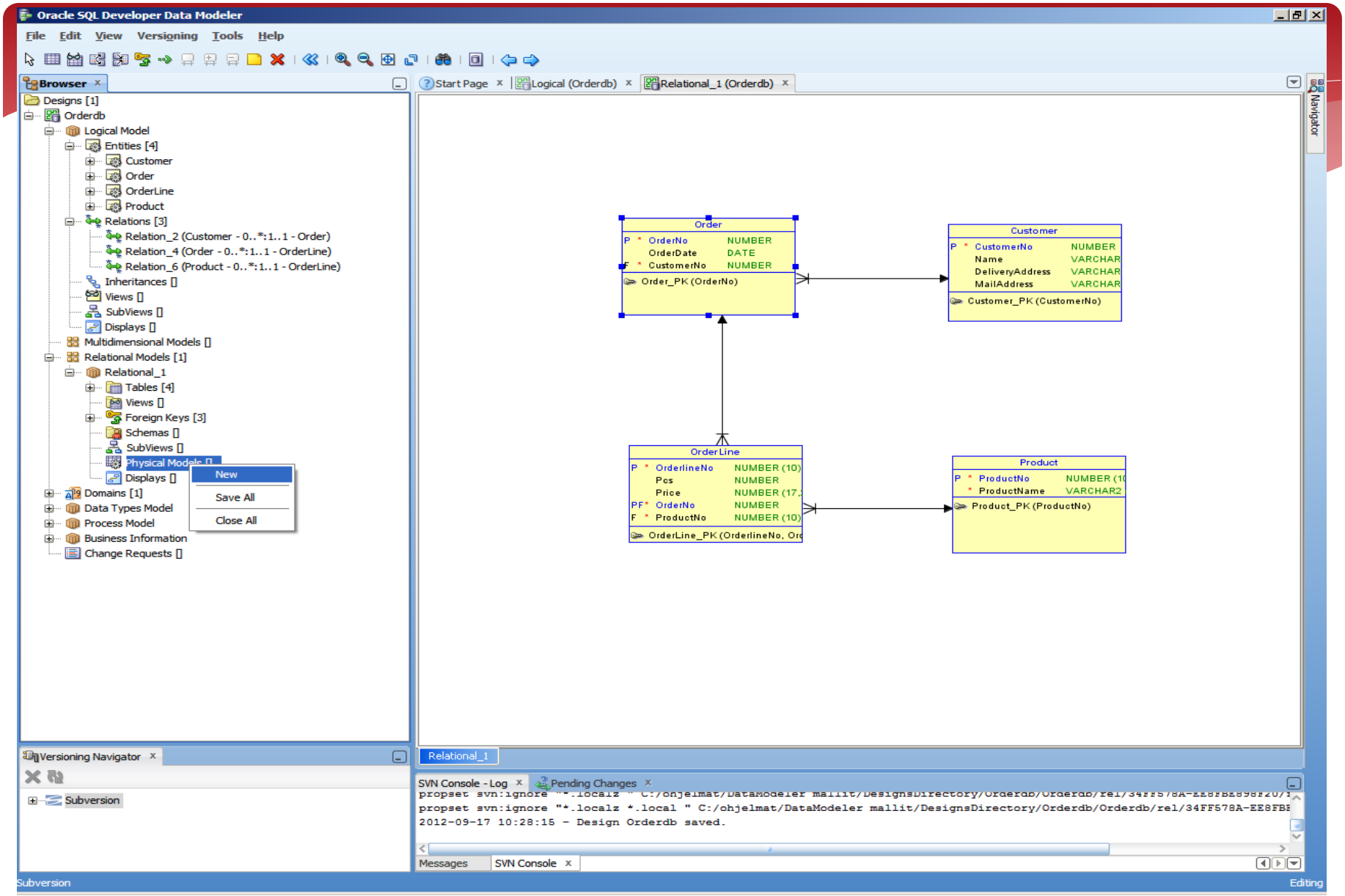
From relational to physical

Copyright © Miracle Finland Oy

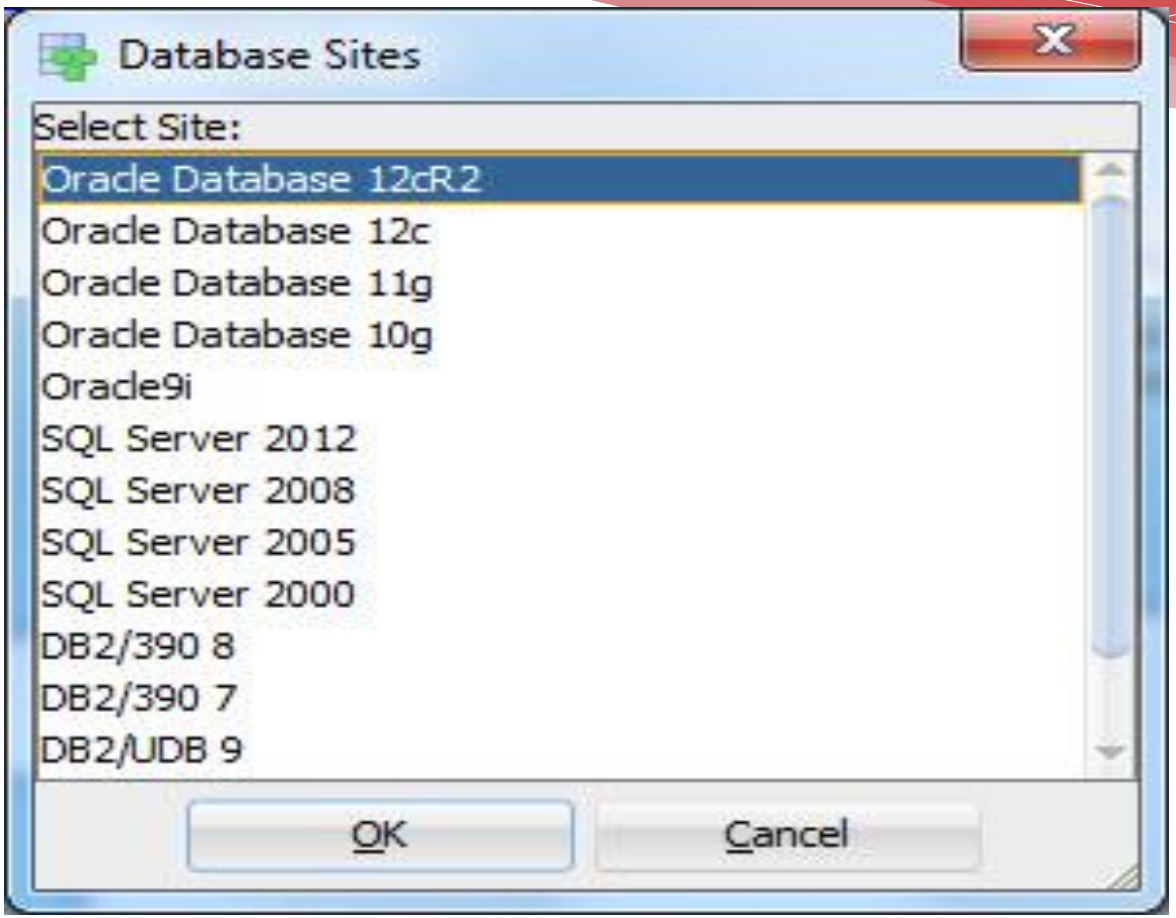
Demo (Physical)

MIRACLE
Miracle Finland Oy

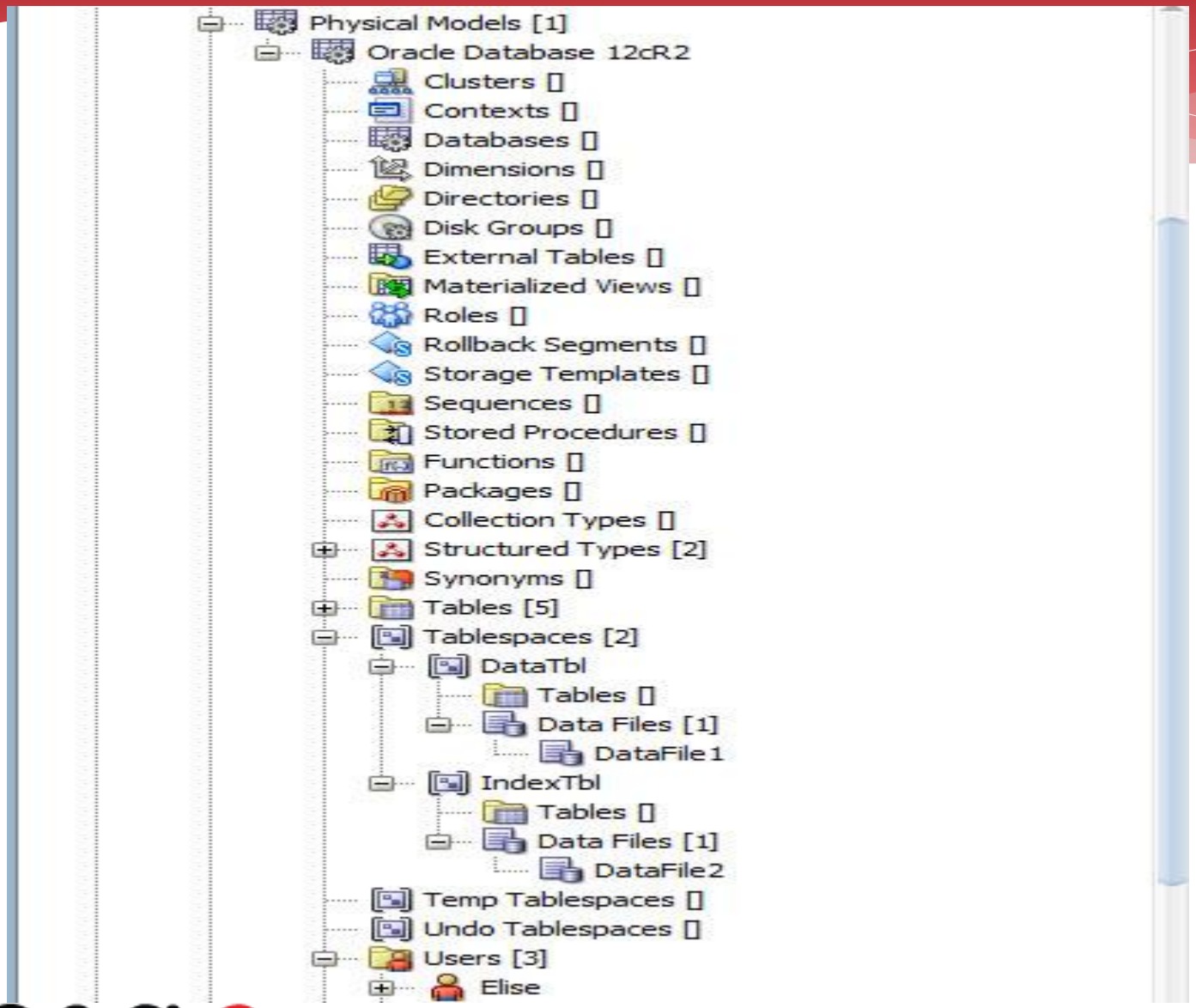
Copyright © Miracle Finland Oy



© Miracle Finland Oy



MIRACLE
Miracle Finland Oy



© Miracle Finland Oy

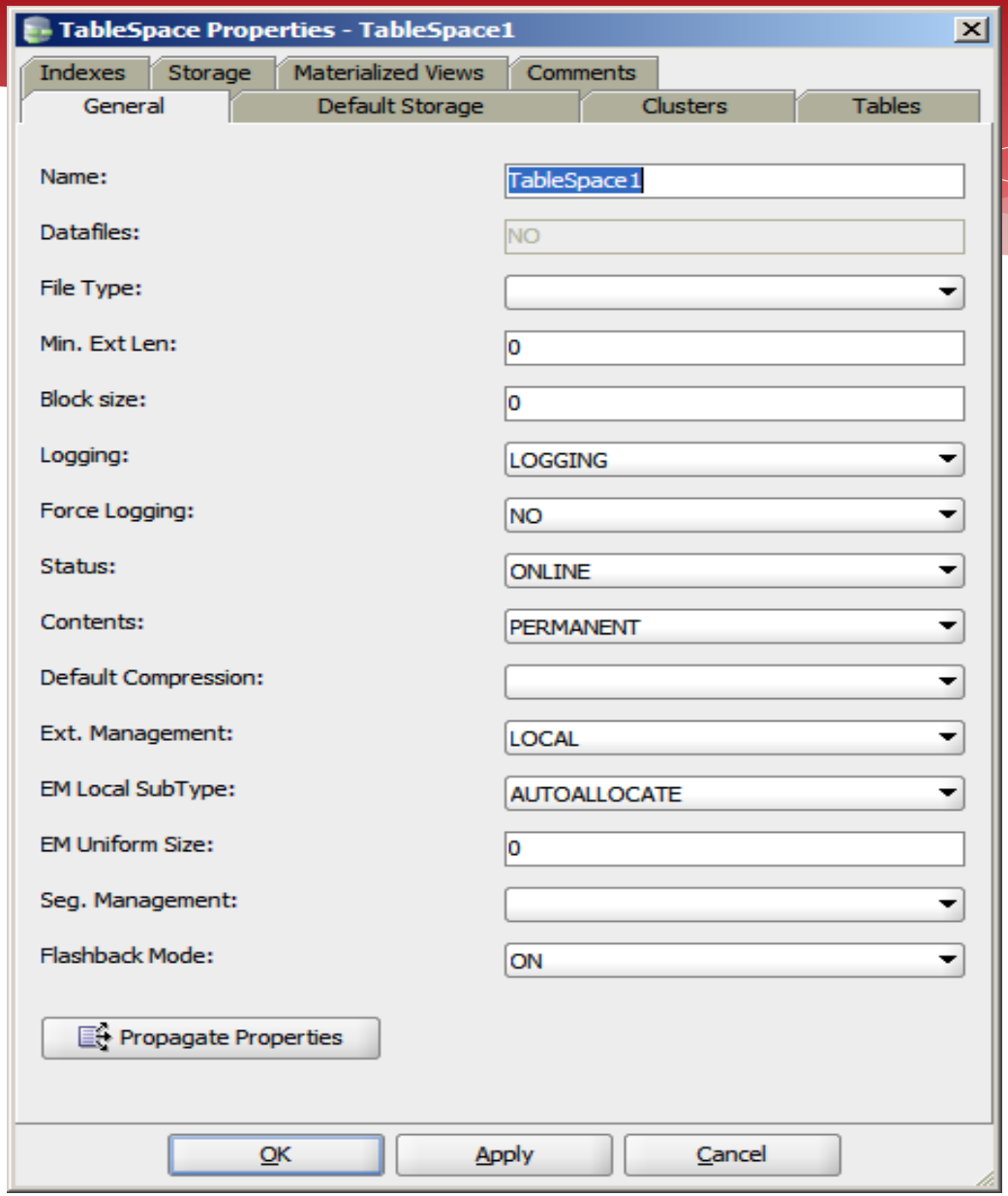
Physical Design

* Creating physical elements (tablespaces, users, …)
* Privileges…

Copyright © Miracle Finland Oy

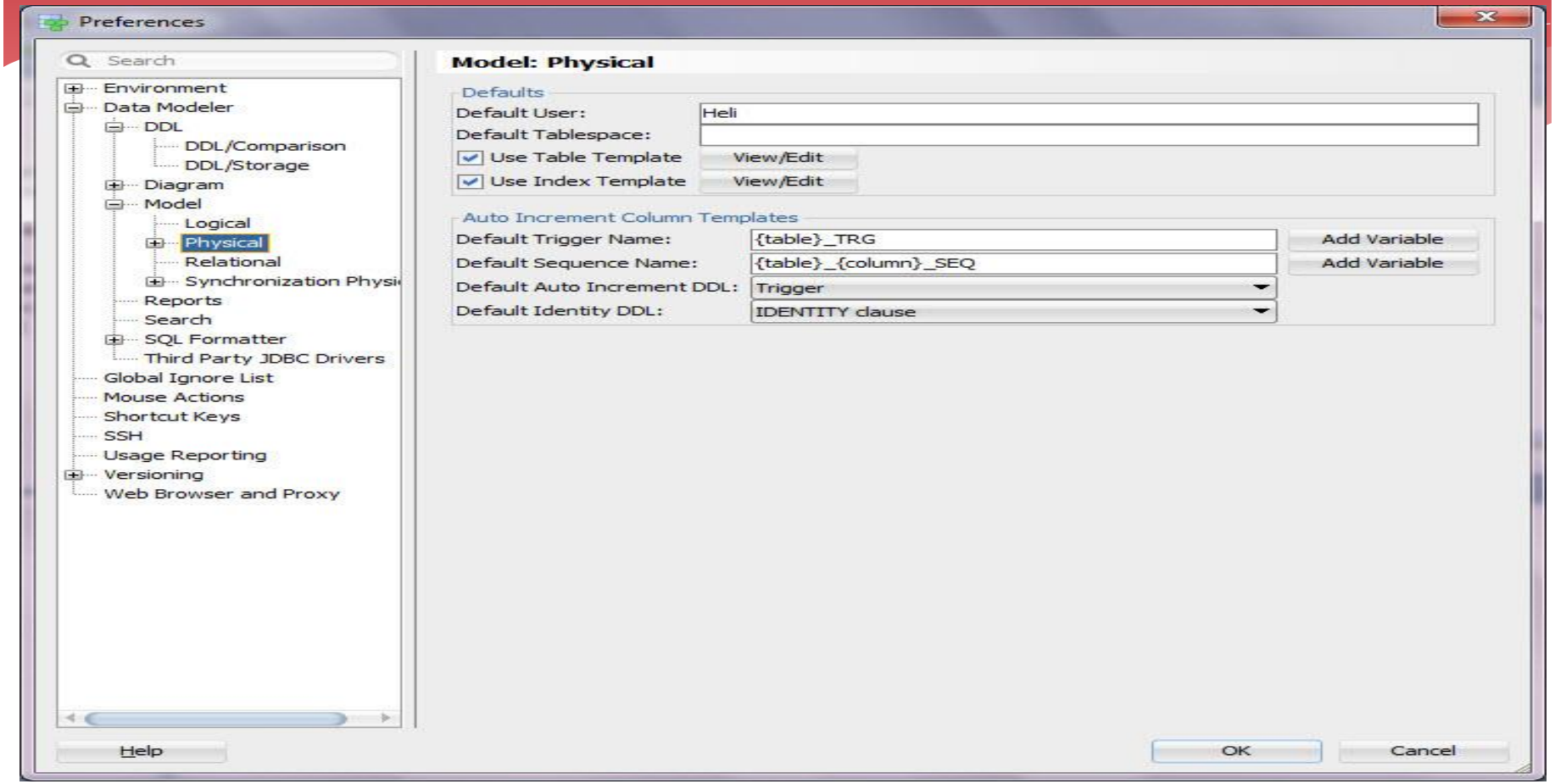
Setting up the Properties

Copyright © Miracle Finland Oy



© Miracle Finland Oy

Setting the Defaults



© Miracle Finland Oy

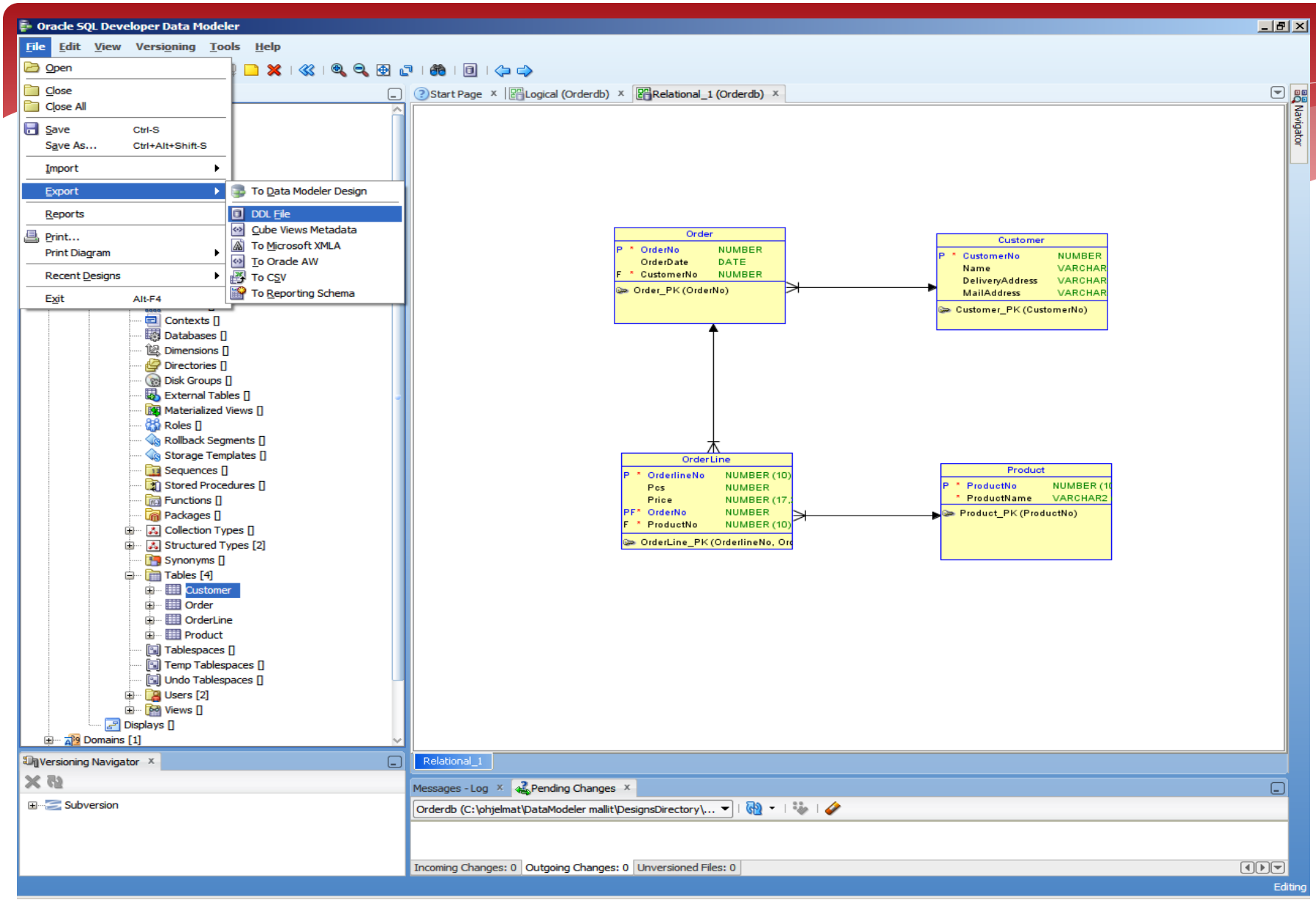
Copyright © Miracle Finland Oy

Physical to DDLs

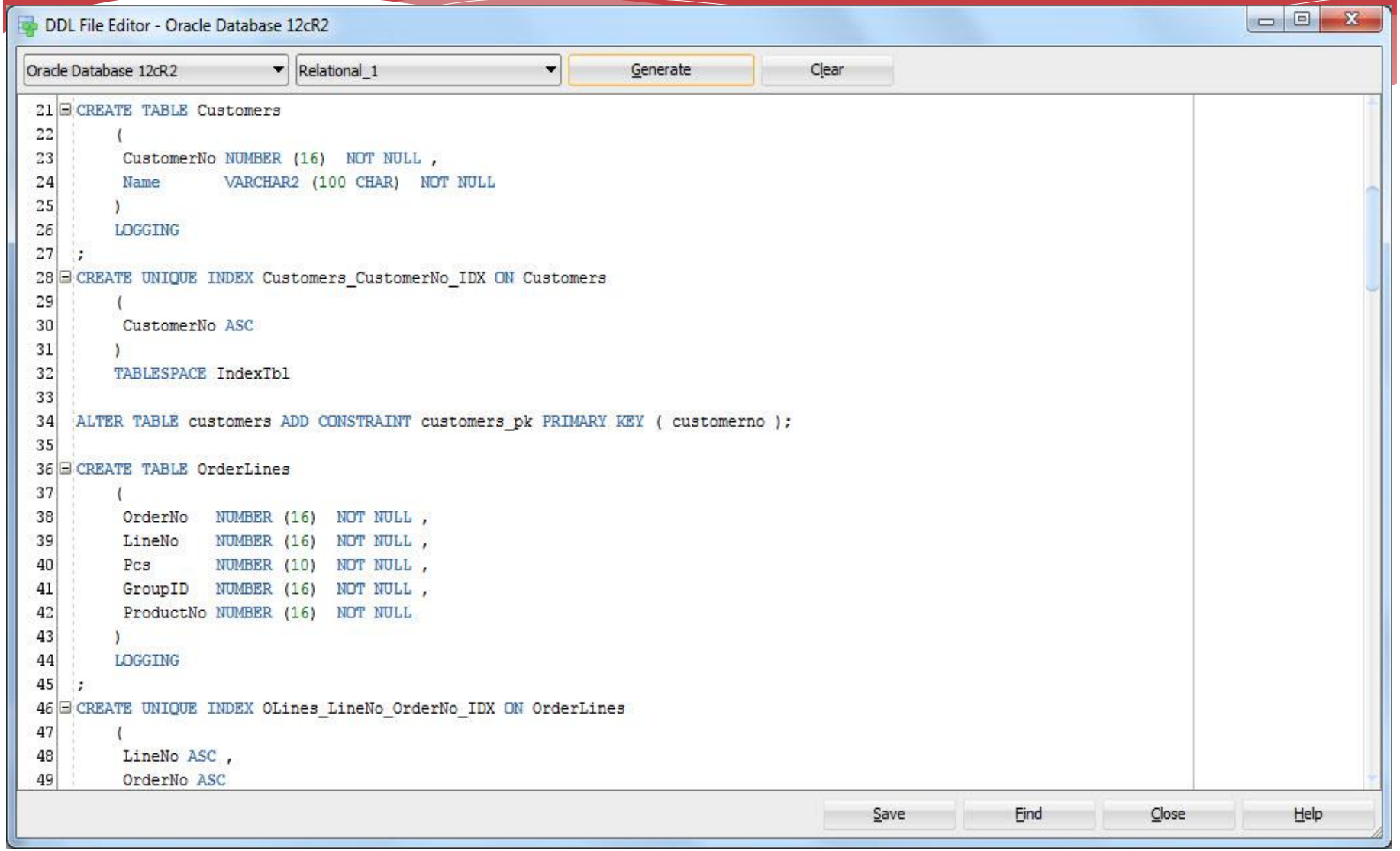
Copyright © Miracle Finland Oy

Demo (DDL)

Copyright © Miracle Finland Oy



© Miracle Finland Oy



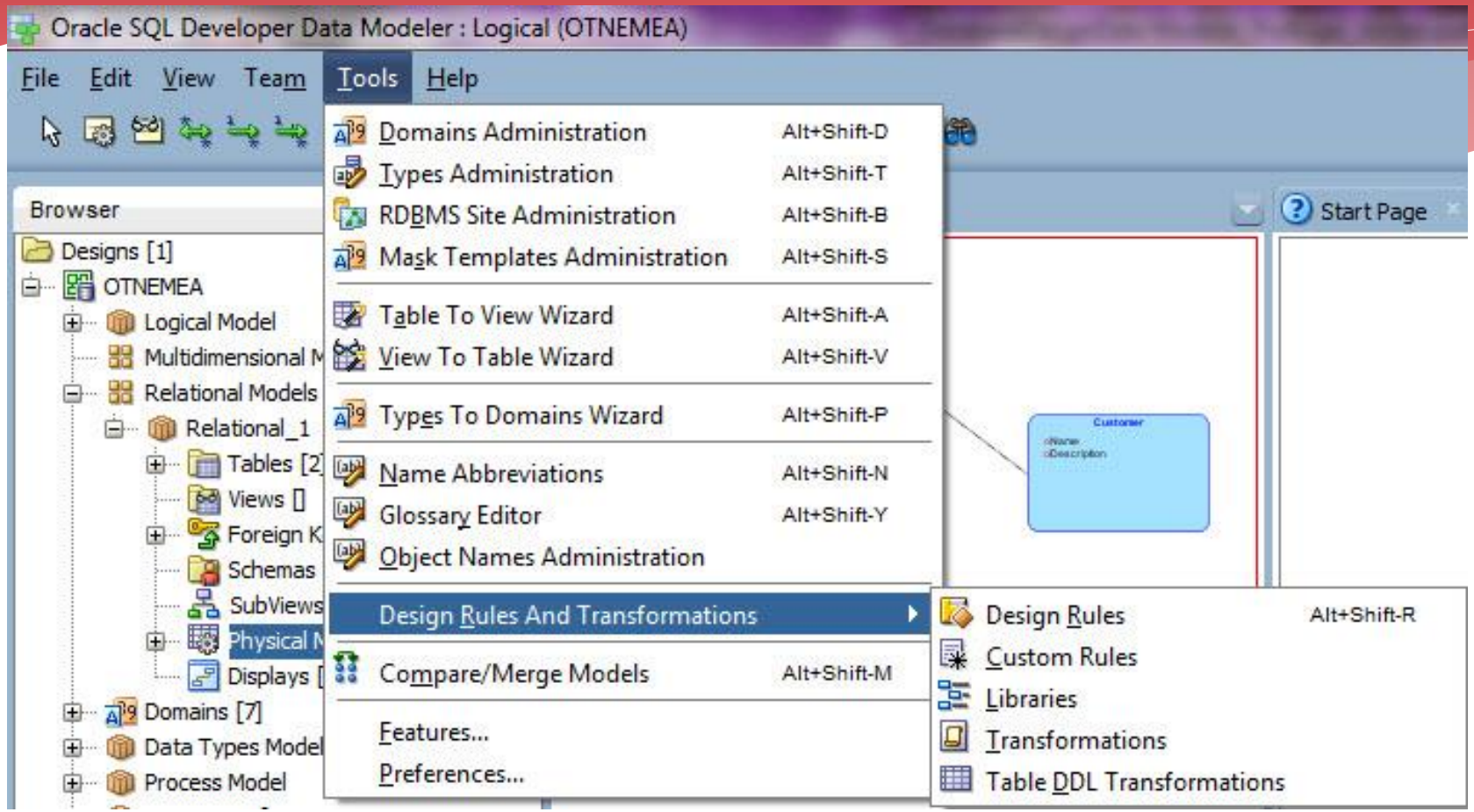
© Miracle Finland Oy

Demo (executing the DDLs to the Cloud)

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

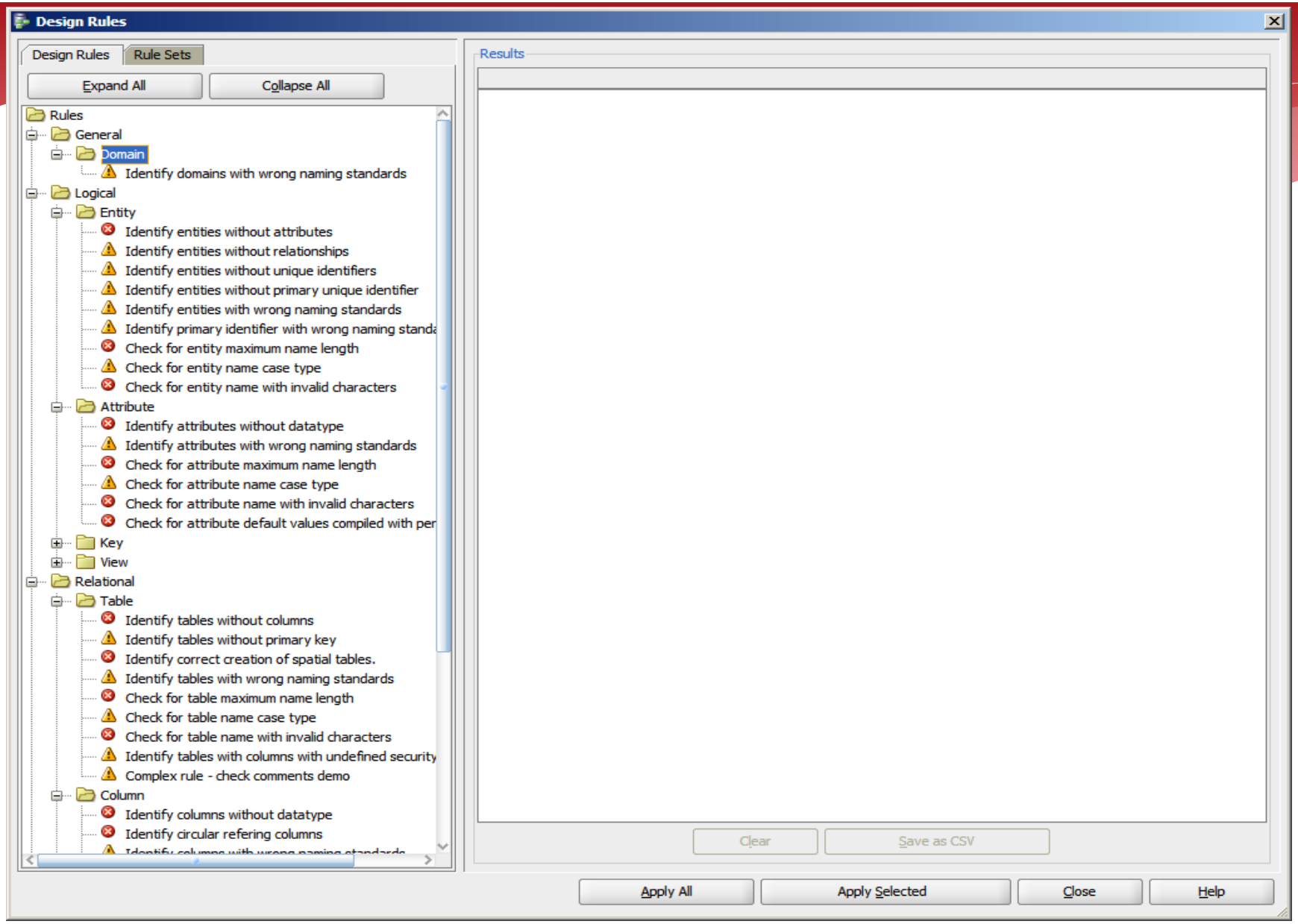
Design Rules



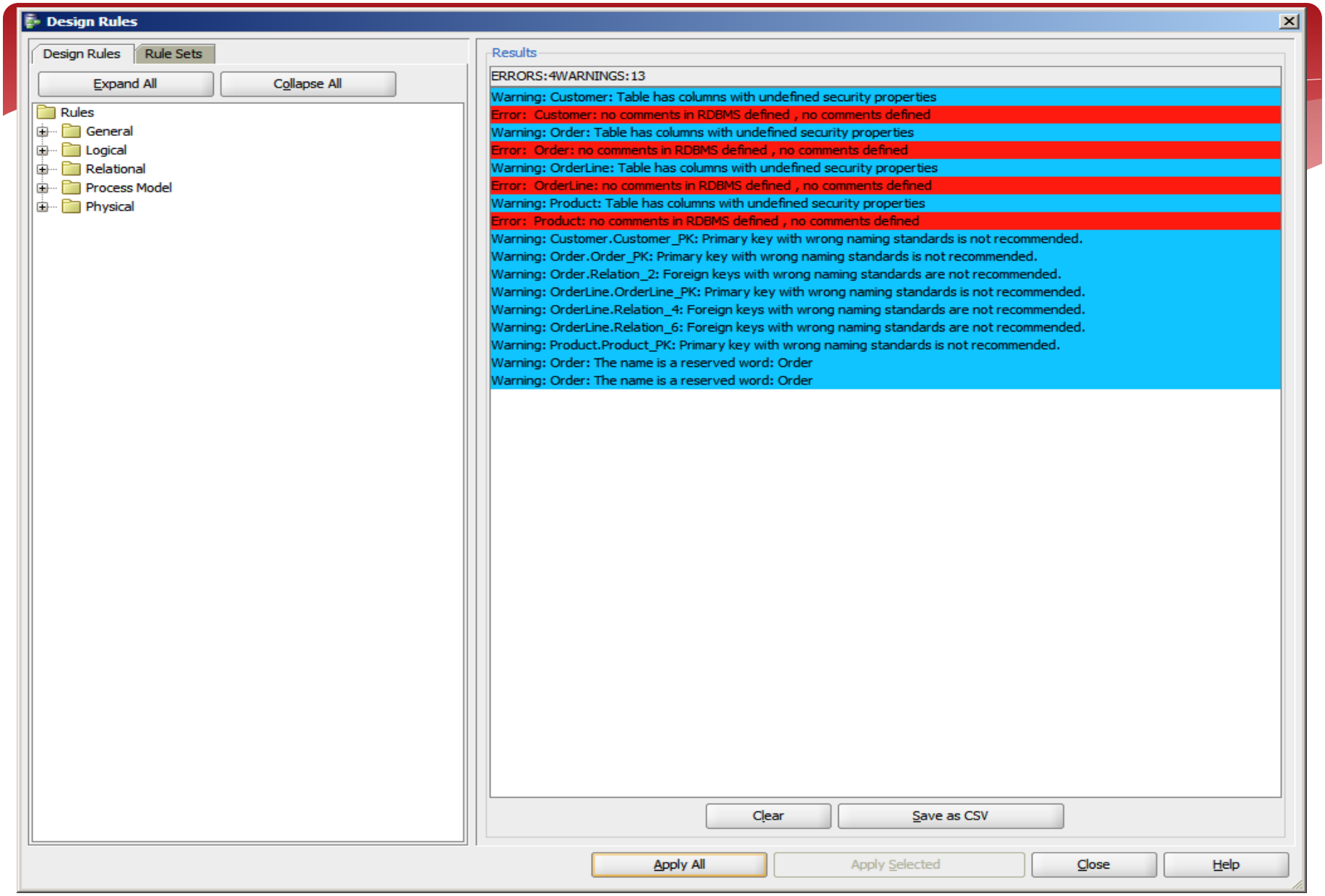
MIRACLE
Miracle Finland Oy

© Miracle Finland Oy

Copyright © Miracle Finland Oy



© Miracle Finland Oy



MIRACLE
Miracle Finland Oy

© Miracle Finland Oy

Different Compares

* Design vs Design
* Design vs Database
* Database vs Design

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

Different Compares, Designs

* File | Import | Data Modeler Design
  * Compares ”everything”
* Tools | Compare/Merge Models
  * Compares only relational and physical models
  * ALTER DDLs can be generated

* These can be used for instance to compare different versions of the design from version control

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

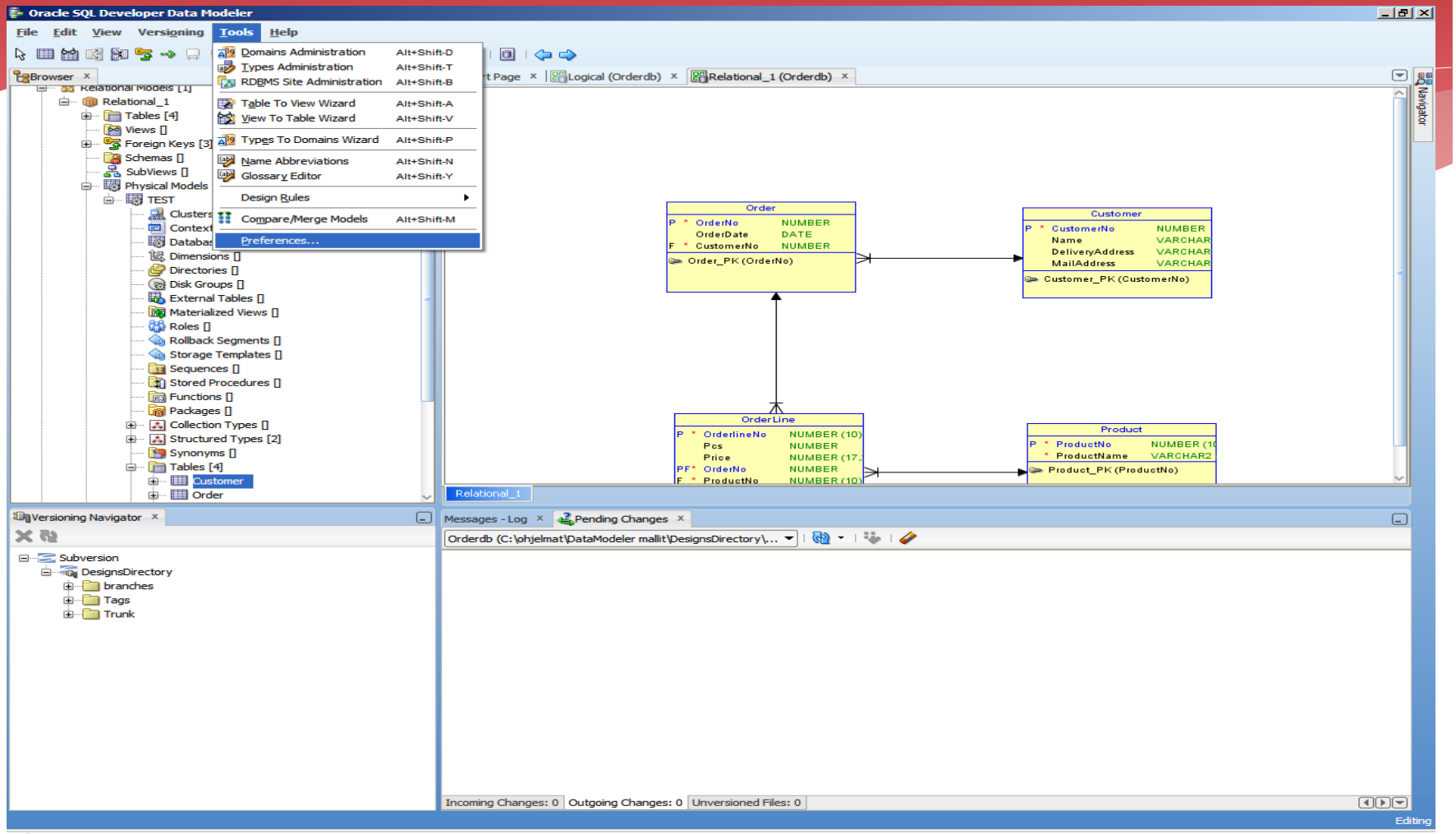
Different Compares, Database

* Synchronize Model With Data Dictionary
  * Target: model
* Synchronize Data Dictionary With Model
  * Target: database
* File | Import | DDL File
* File | Import | Data Dictionary

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

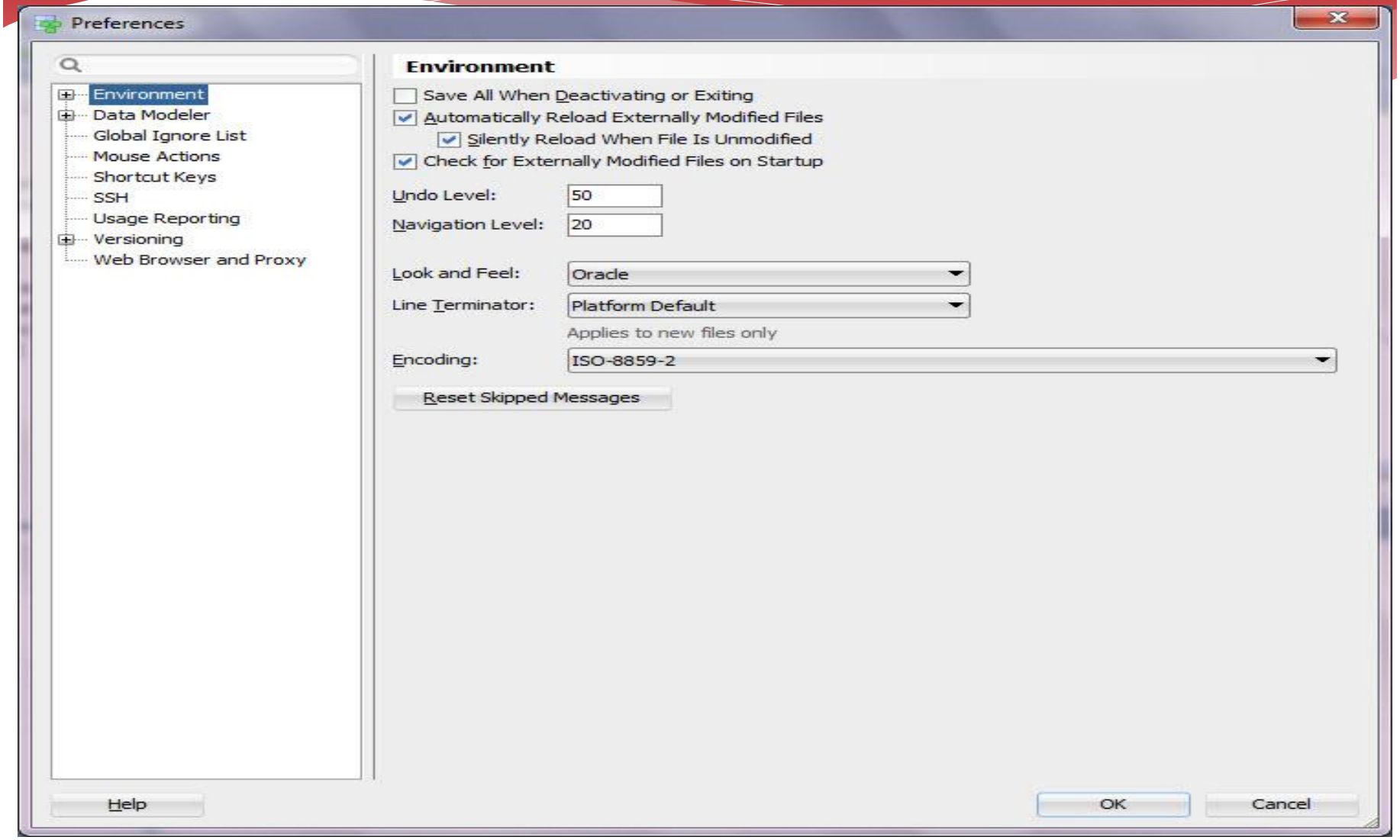
Preferences



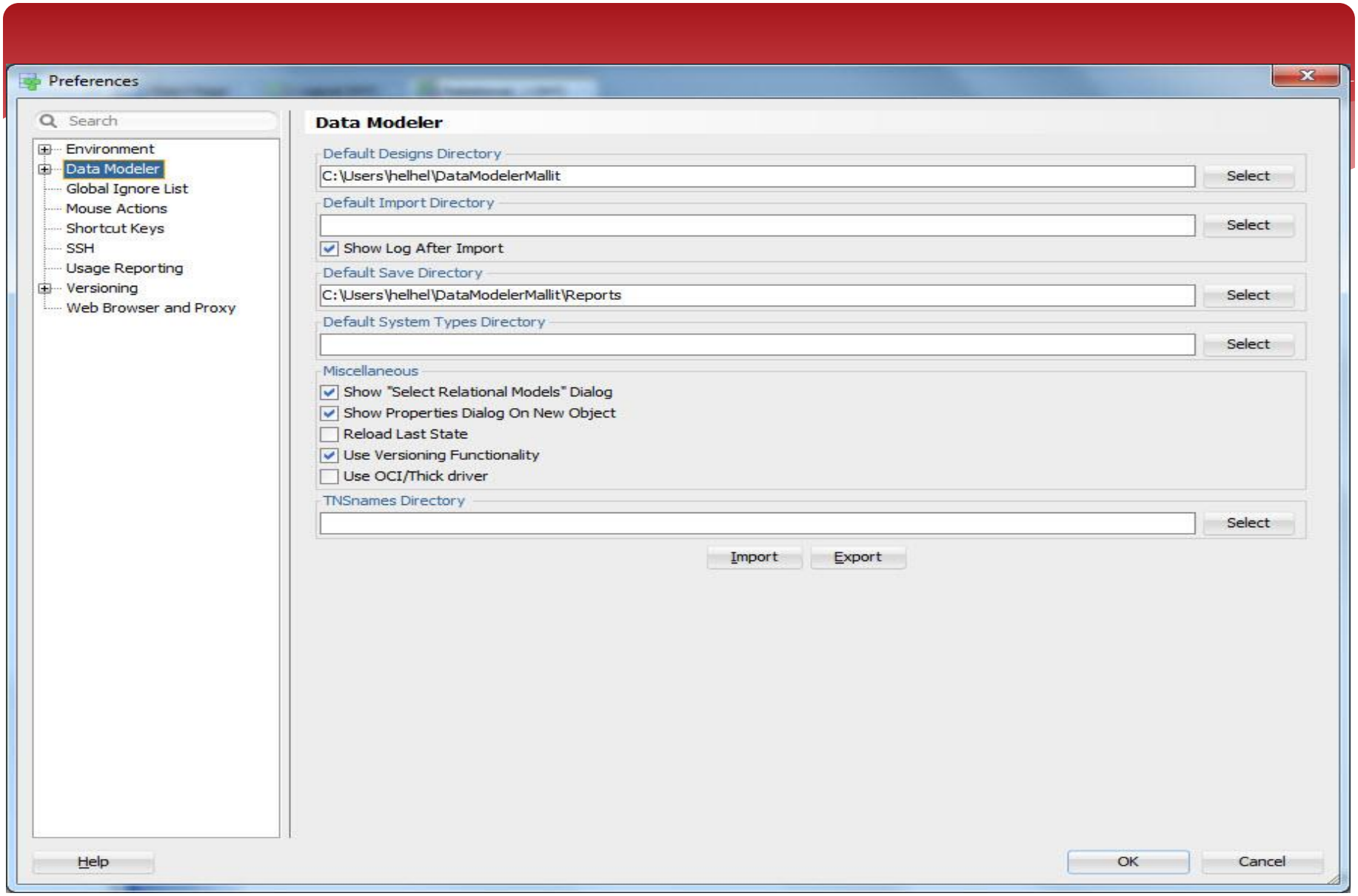
MIRACLE
Miracle Finland Oy

© Miracle Finland Oy

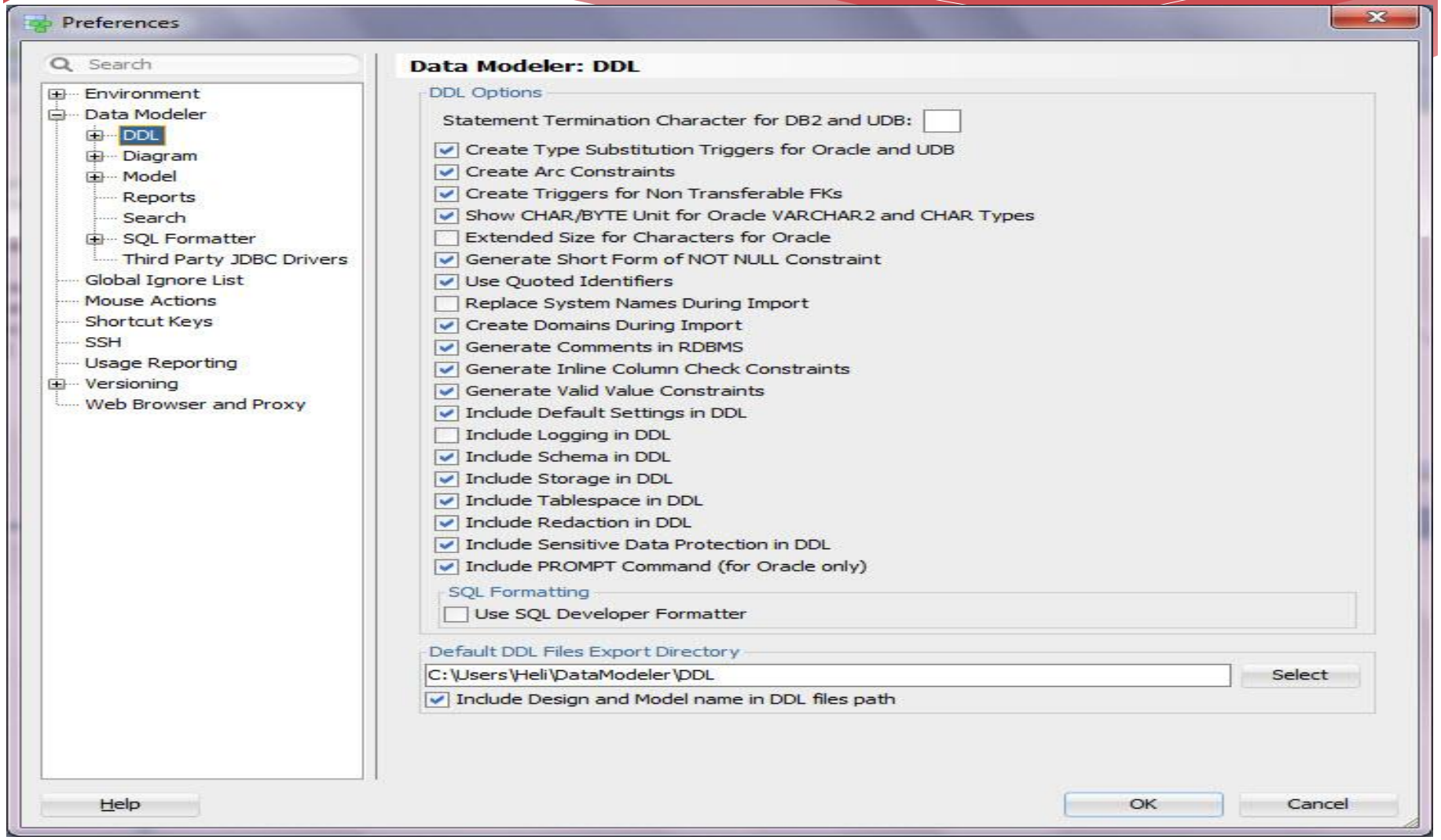
Copyright © Miracle Finland Oy



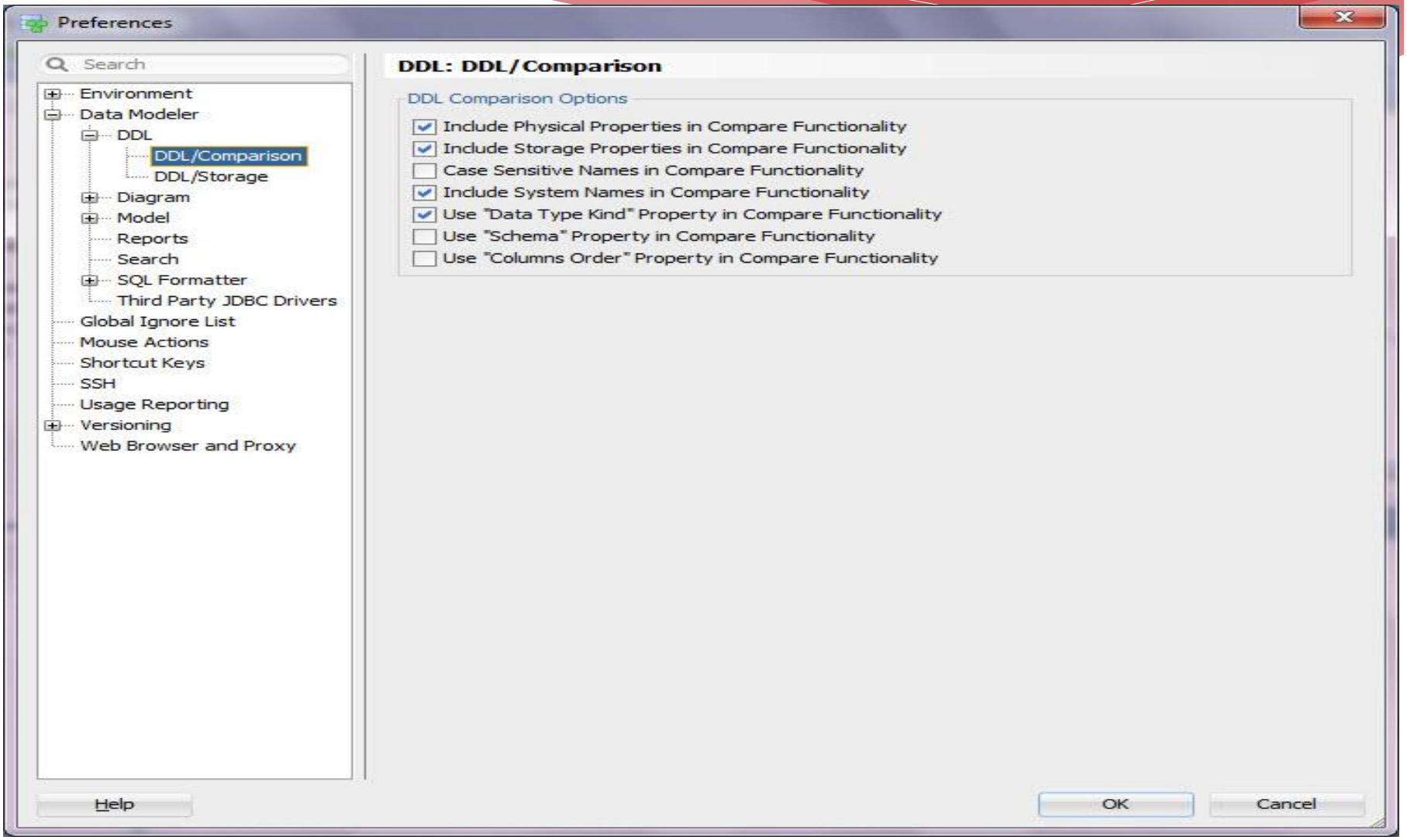
© Miracle Finland Oy



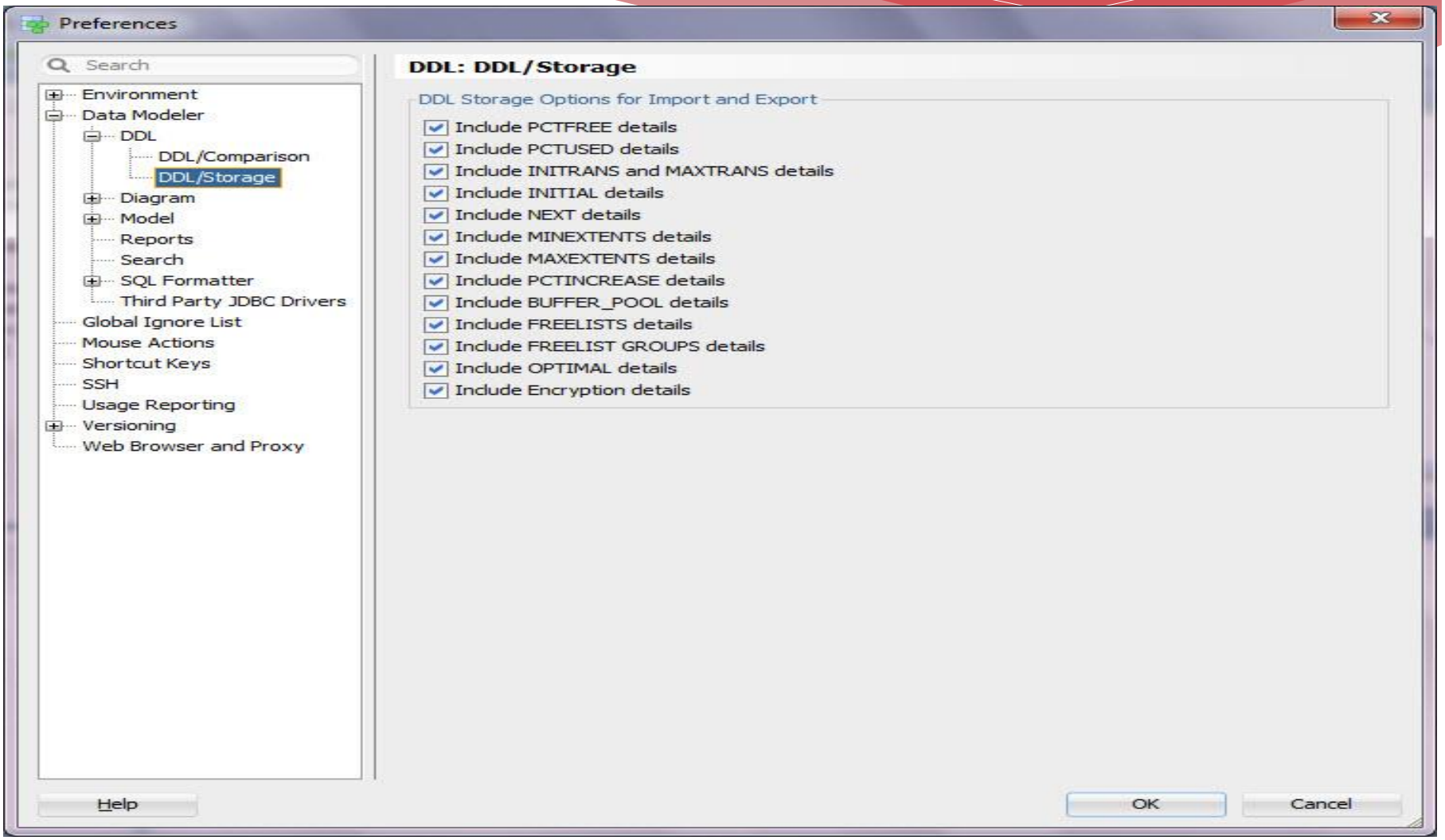
© Miracle Finland Oy



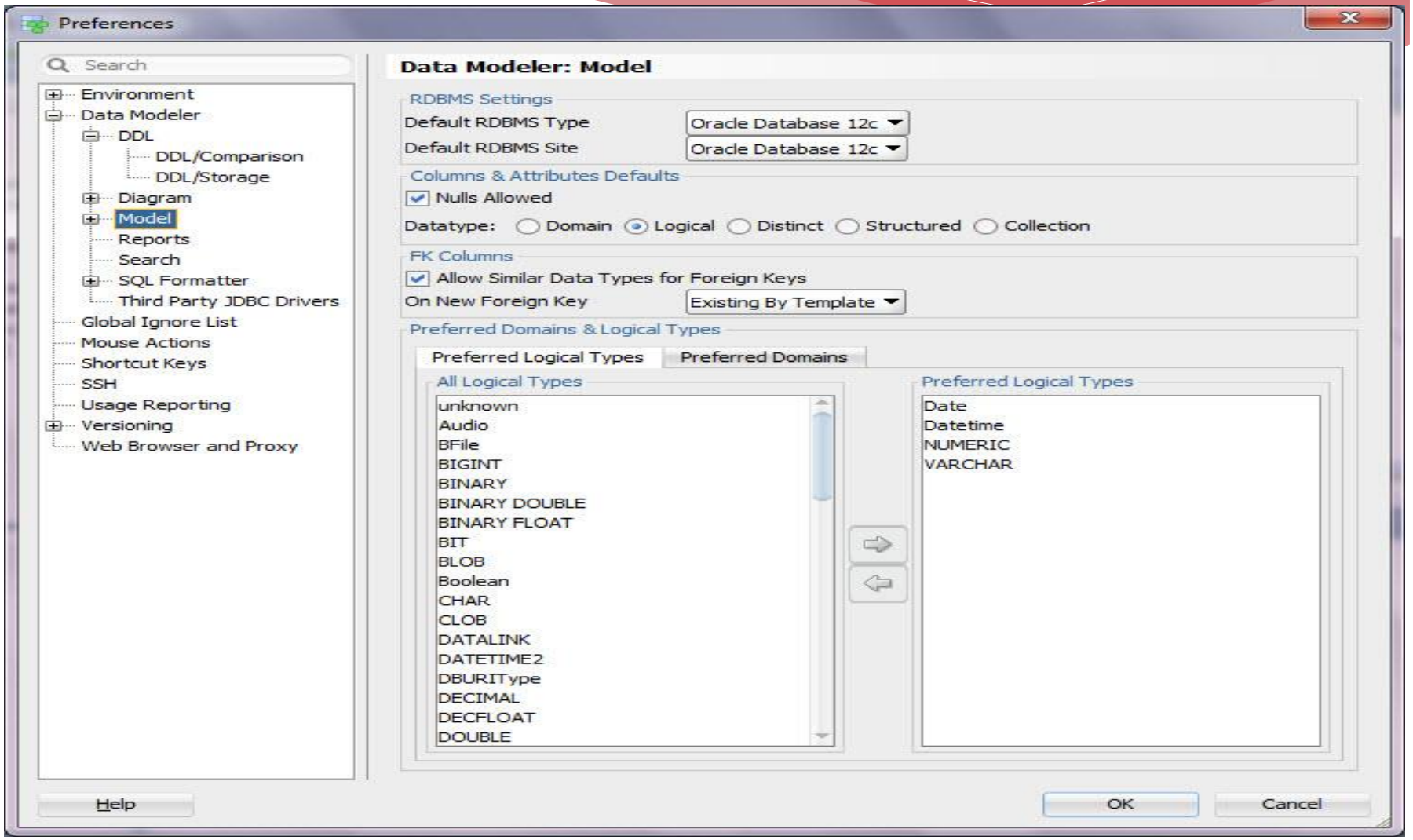
© Miracle Finland Oy



© Miracle Finland Oy



MIRACLE
Miracle Finland Oy

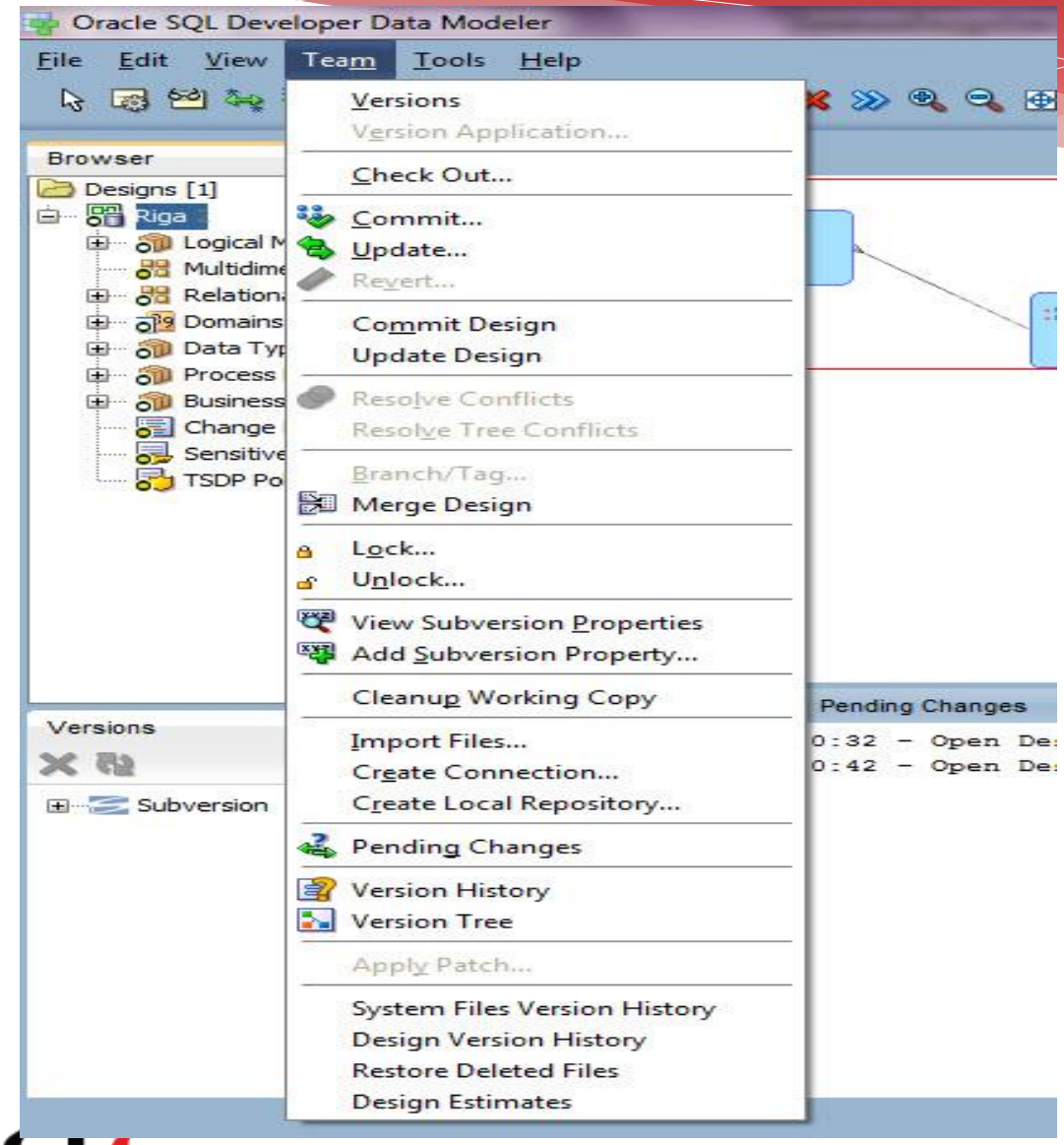


© Miracle Finland Oy

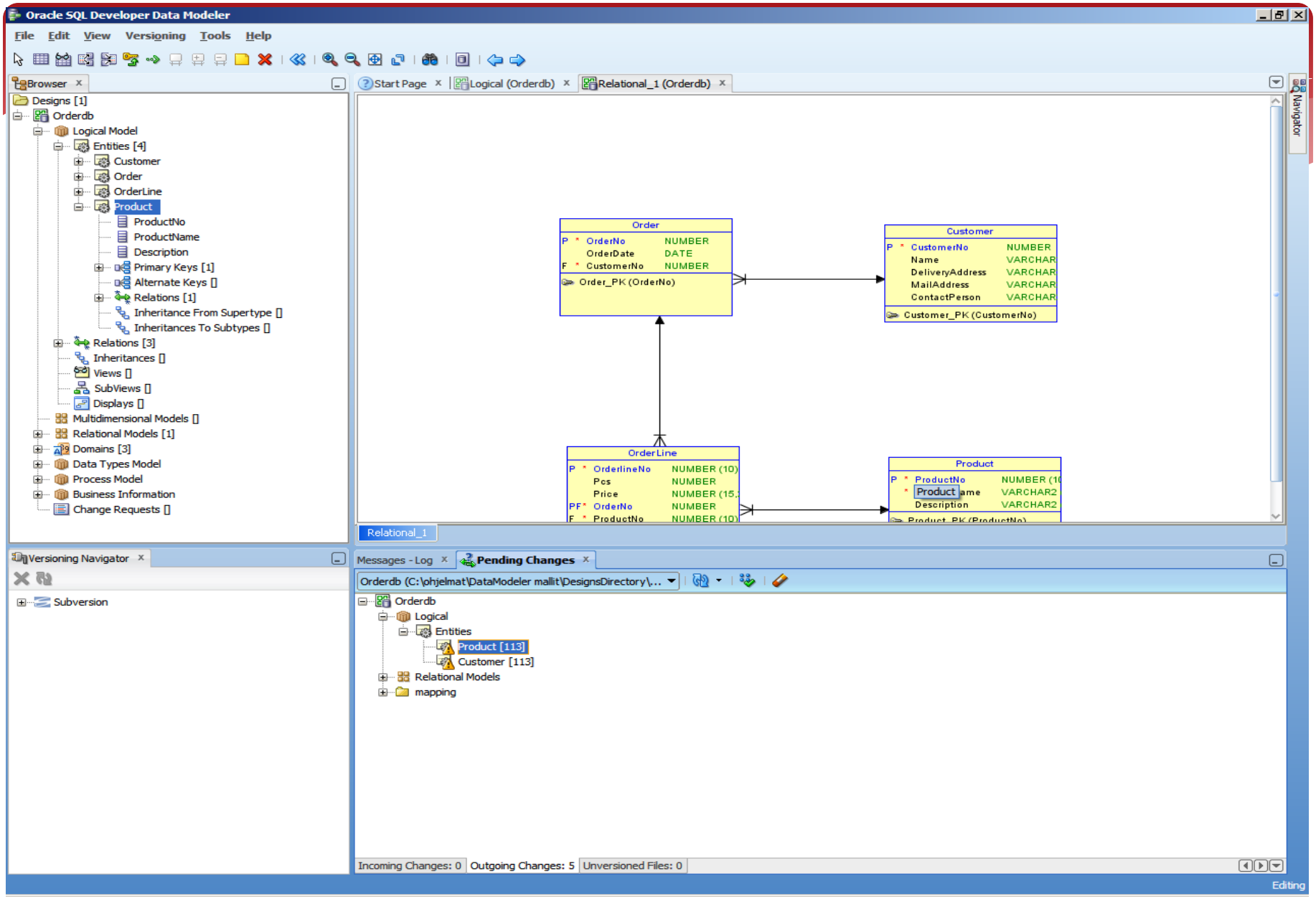
Version control (Subversion)

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

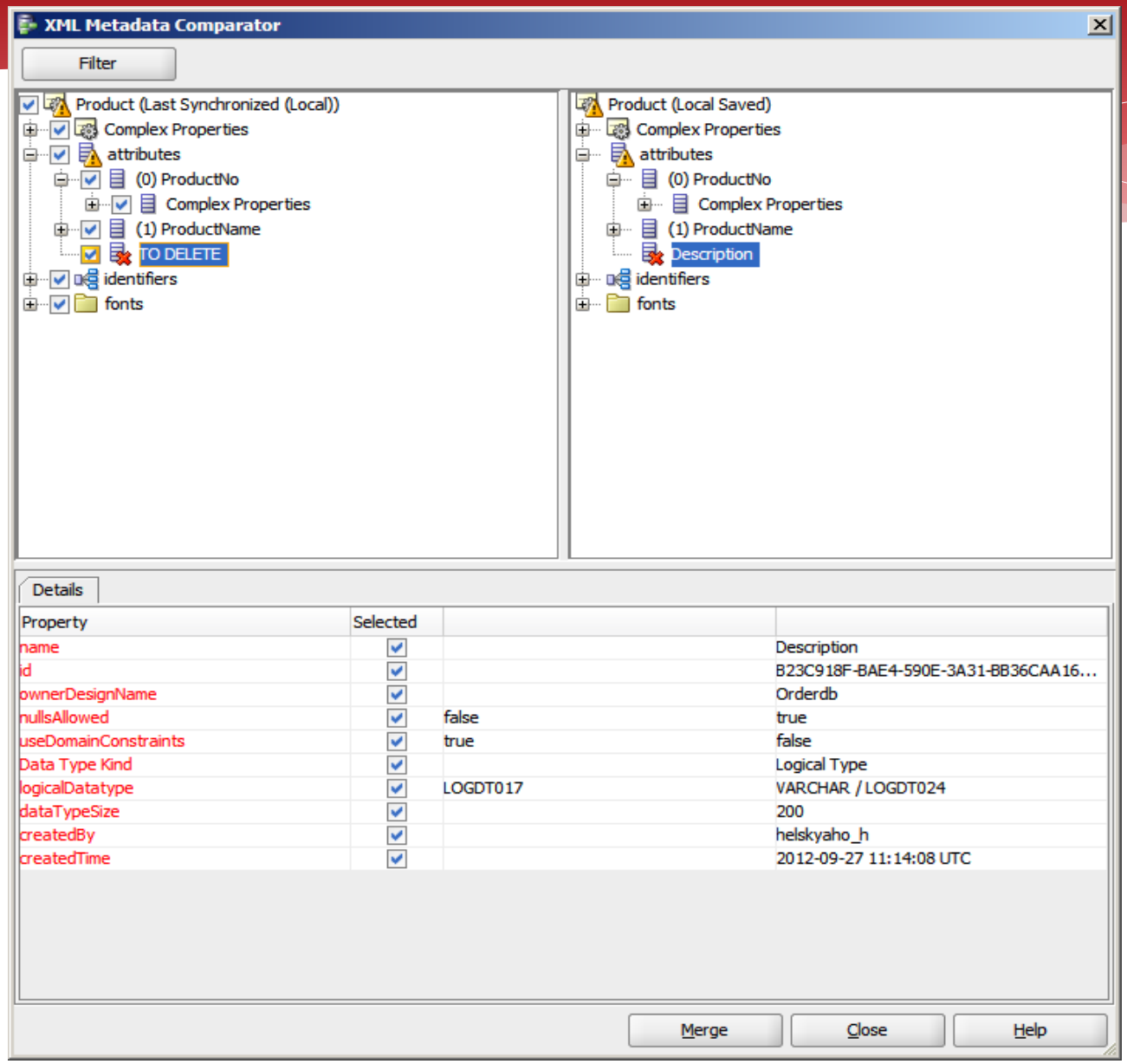


© Miracle Finland Oy

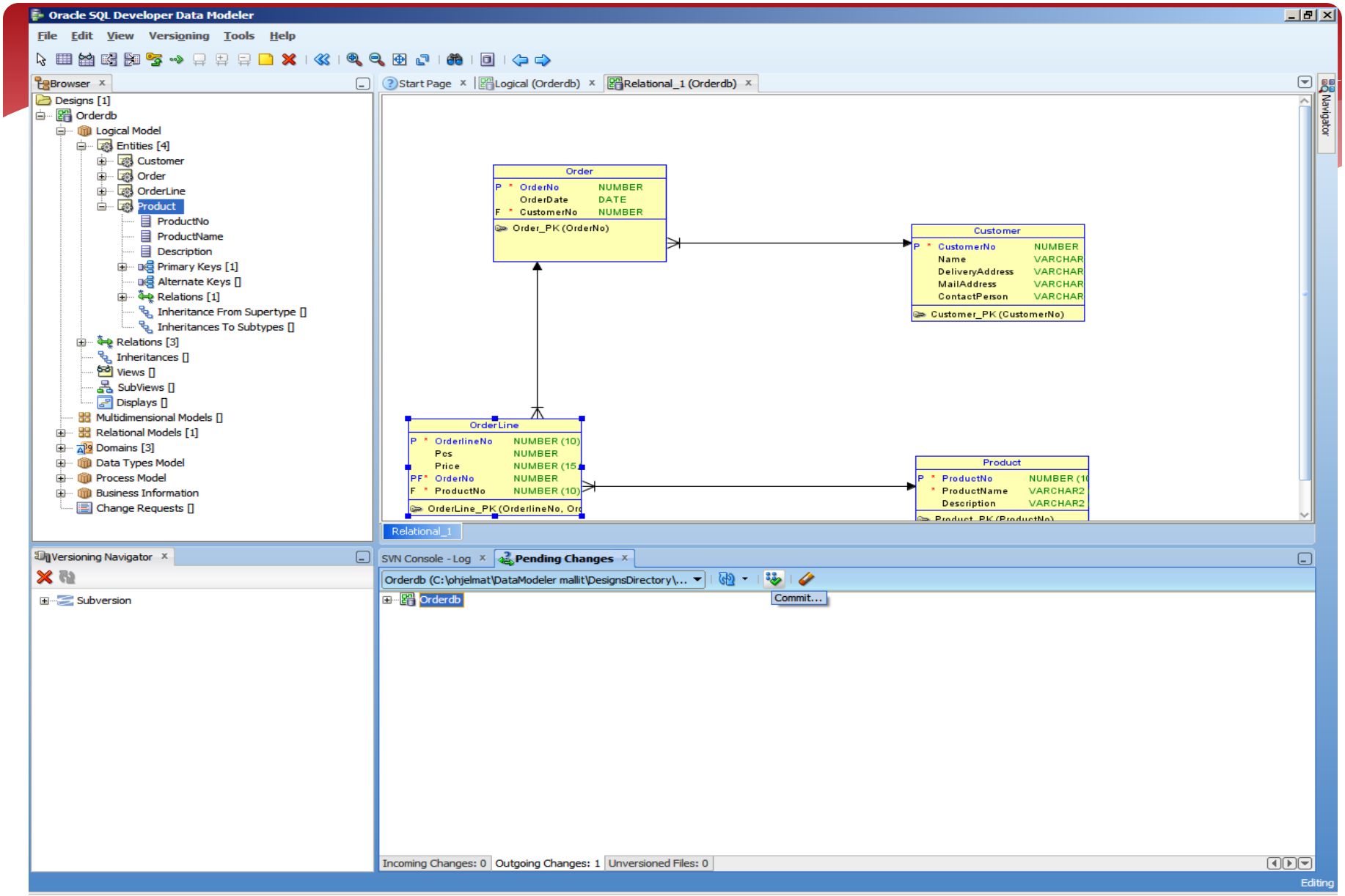


MIRACLE
Miracle Finland Oy

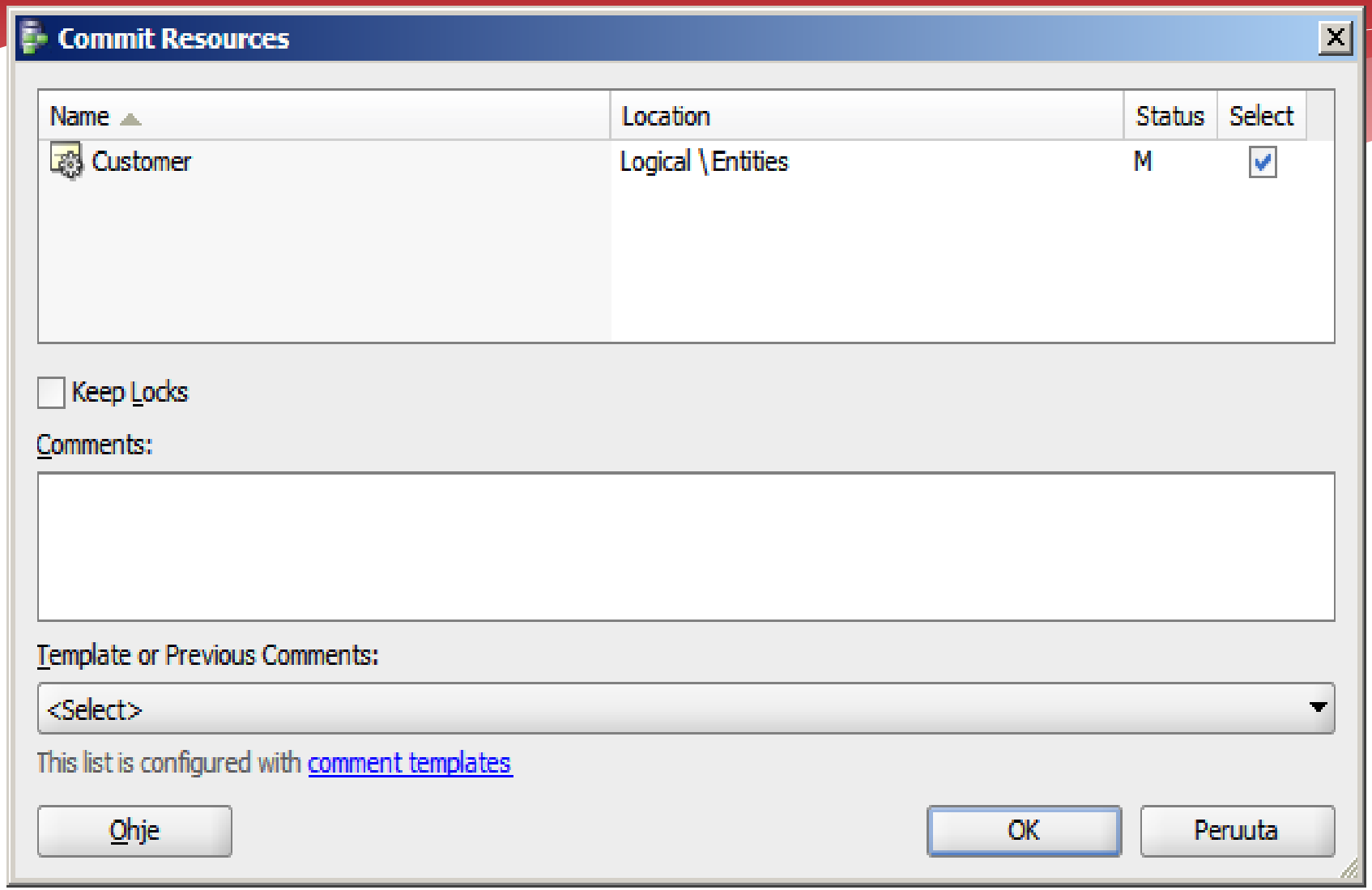
© Miracle Finland Oy



© Miracle Finland Oy



© Miracle Finland Oy



MIRACLE
Miracle Finland Oy

© Miracle Finland Oy

Conclusions

* To be able to connect to the database, to see its content online and to maintain the data: **SQL Developer**
* To *design* the database and to maintain the data structures: **Data Modeler**
* To *design* the data architecture: **Data Modeler**

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

Conclusions

* Data Modeler is a good tool; good support for iterative processes
* Enables documenting and versioning (and comparing the versions)
* Enables multiuser environment
* Is free to use
* Support for other databases as well
* I see no reason not to use it!

MIRACLE
Miracle Finland Oy

Copyright © Miracle Finland Oy

THANK YOU!

QUESTIONS?

Heli:

Email: heli@miracleoy.fi

Twitter: @HeliFromFinland

Blog: Helifromfinland.com

Elise:

Email: elise.valin-raki@fennia.fi

Twitter: @ValinRaki

Blog: evrellai.wordpress.com

MIRACLE
Miracle Finland Oy