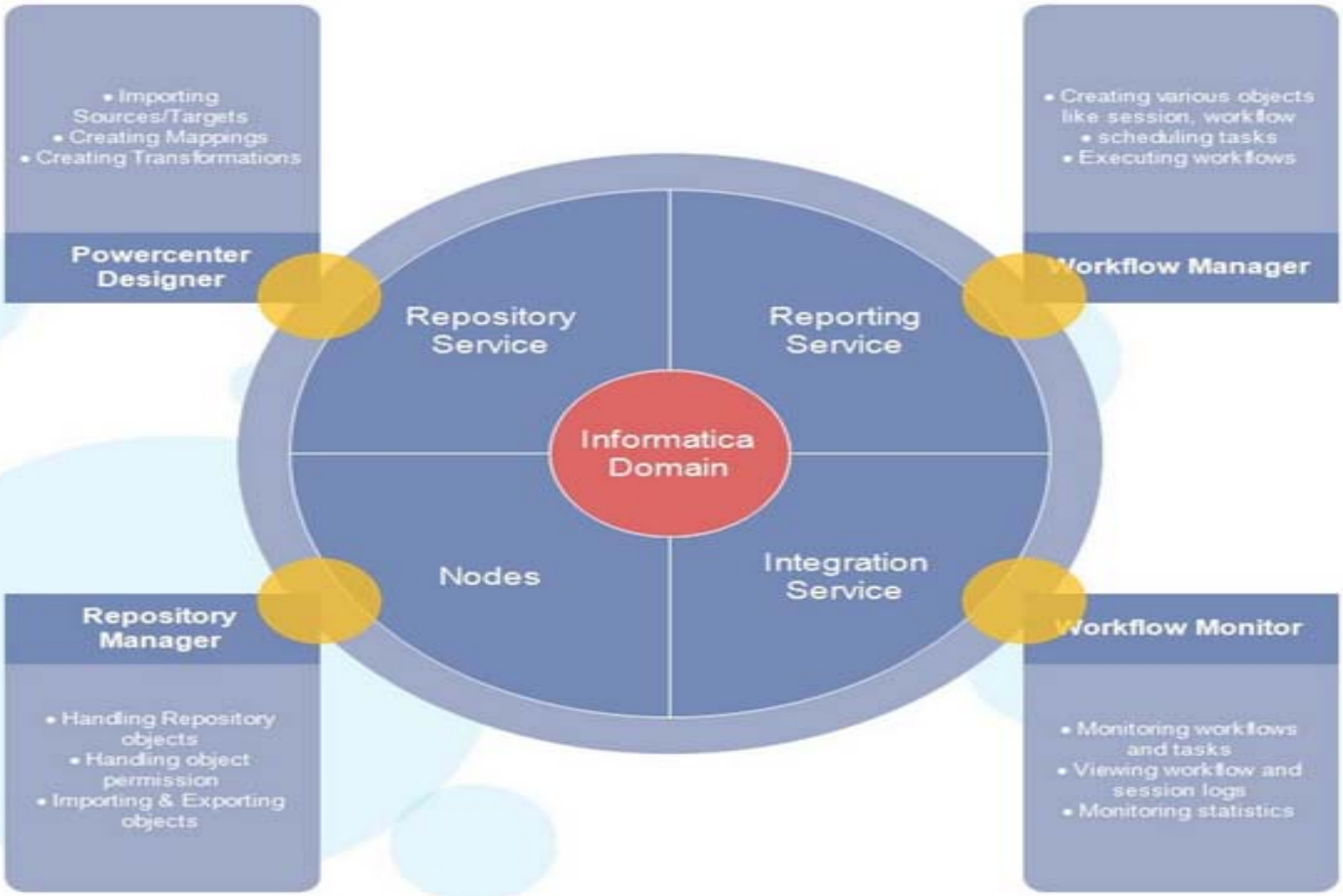
Informatica Architecture

- Informatica ETL tool consists of following services & components
    - Repository Service – Responsible for maintaining Informatica metadata & providing access of same to other services.
    - Integration Service – Responsible for the movement of data from sources to targets.
    - Reporting Service - Enables the generation of reports.
    - Nodes – Computing platform where the above services are executed.
    - Informatica Designer - Used for creation of mappings between source and target.
    - Workflow Manager – Used to create workflows and other task & their execution.
    - Workflow Monitor – Used to monitor the execution of workflows.
    - Repository Manager – Used to manage objects in repository.


• Importing Sources/Targets
• Creating Mappings
• Creating Transformations
Powercenter Designer
• Creating various objects like session, workflow
• scheduling tasks
• Executing workflows
Workflow Manager
Repository Service
Reporting Service
Informatica Domain
Nodes
Integration Service
Repository Manager
• Handling Repository objects
• Handling object permission
• Importing & Exporting objects
Workflow Monitor
• Monitoring workflows and tasks
• Viewing workflow and session logs
• Monitoring statistics

Informatica Domain

- The overall architecture of Informatica is Service Oriented Architecture (SOA).
- Informatica Domain is the fundamental administrative unit in Informatica tool
- It is a collection of nodes and services. Further, this nodes and services can be categorized into folders and sub-folders based on the administration requirement.
- Node is a logical representation of a machine inside the domain. Node is required to **run services and processes for Informatica.**
- You can have multiple nodes in a domain. In a domain, you will also find a gateway node.
- The gateway node is responsible for receiving requests from different client tools and routing those requests to different nodes and services.
- There are two types of services in Domain
  - **Service Manager**: Service manager manages domain operations like authentication, authorization, and logging. It also runs application services on the nodes as well as manages users and groups.
  - **Application Services**: Application service represents the server specific services like integration service, repository service, and reporting service. These services run on different nodes based upon the configuration.

PowerCenter Repository

- PowerCenter repository is a relational database like Oracle, Sybase, SQL server and it is managed by repository service. It consists of database tables that store metadata.
- There are three Informatica Client tools available in Informatica Powercenter. They are Informatica
  - Designer
  - Workflow Monitor
  - Workflow Manager
- These clients can access to the repository using repository service only.
- To manage a repository there exists an Informatica service called Repository Service. A single repository service handles exclusively only one repository. Also, a repository service can execute on multiple nodes to increase the performance.
- The repository services use locks on the objects, so multiple users cannot modify the same object same time.
- You can enable version control in the repository. With the version control feature, you can maintain different versions of the same object

PowerCenter Repository

- Objects created in the repository can have following three state

- Valid: Valid objects are those objects whose syntax is correct according to Informatica. These objects can be used in the execution of workflows.

- Invalid: Invalid objects are those who does not adhere to the standard or rules specified. When any object is saved in Informatica, it is checked whether its syntax and properties are valid or not, and the object is marked with the status accordingly.

- Impacted: Impacted objects are those whose child objects are invalid. For example in a mapping if you are using a reusable transformation, and this transformation object becomes invalid then the mapping will be marked as impacted.

Powercenter client & Server Connectivity

- PowerCenter client tools are development tools which are installed on the client machines. Powercenter designer, workflow manager, a repository manager, and workflow monitor are the main client tools.

- The mappings and objects that we create in these client tools are saved in the Informatica repository which resides on the Informatica server. So the client tools must have network connectivity to the server.

- On the other hand, PowerCenter client connects to the sources and targets to import the metadata and source/target structure definitions. So it also must have connectivity to the source/target systems.

- To connect to the integration service and repository service, PowerCenter client uses TCP/IP protocols and to connect to the sources/targets PowerCenter client uses ODBC drivers.

Repository Service

- The repository service maintains the connections from Powercenter clients to the PowerCenter repository. It is a separate multi-threaded process, and it fetches, inserts and updates the metadata inside the repository. It is also responsible for maintaining consistency inside the repository metadata.

Integration Service

- Integration service is the executing engine for the Informatica, in other words, this is the entity which executes the tasks that we create in Informatica. This is how it works
- A user executes a workflow
- Informatica instructs the integration service to execute the workflow
- The integration service reads workflow details from the repository
- Integration service starts execution of the tasks inside the workflow
- Once execution is complete, the status of the task is updated i.e. failed, succeeded or aborted.
- After completion of execution, session log and workflow log is generated.
- This service is responsible for loading data into the target systems
- The integration service also combines data from different sources
- For example, it can combine data from an oracle table and a flat file source.
- So, in summary, Informatica integration service is a process residing on the Informatica server waiting for tasks to be assigned for the execution. When we execute a workflow, the integration service receives a notification to execute the workflow. Then the integration service reads the workflow to know the details like which tasks it has to execute like mappings & at what timings. Then the service reads the task details from the repository and proceeds with the execution

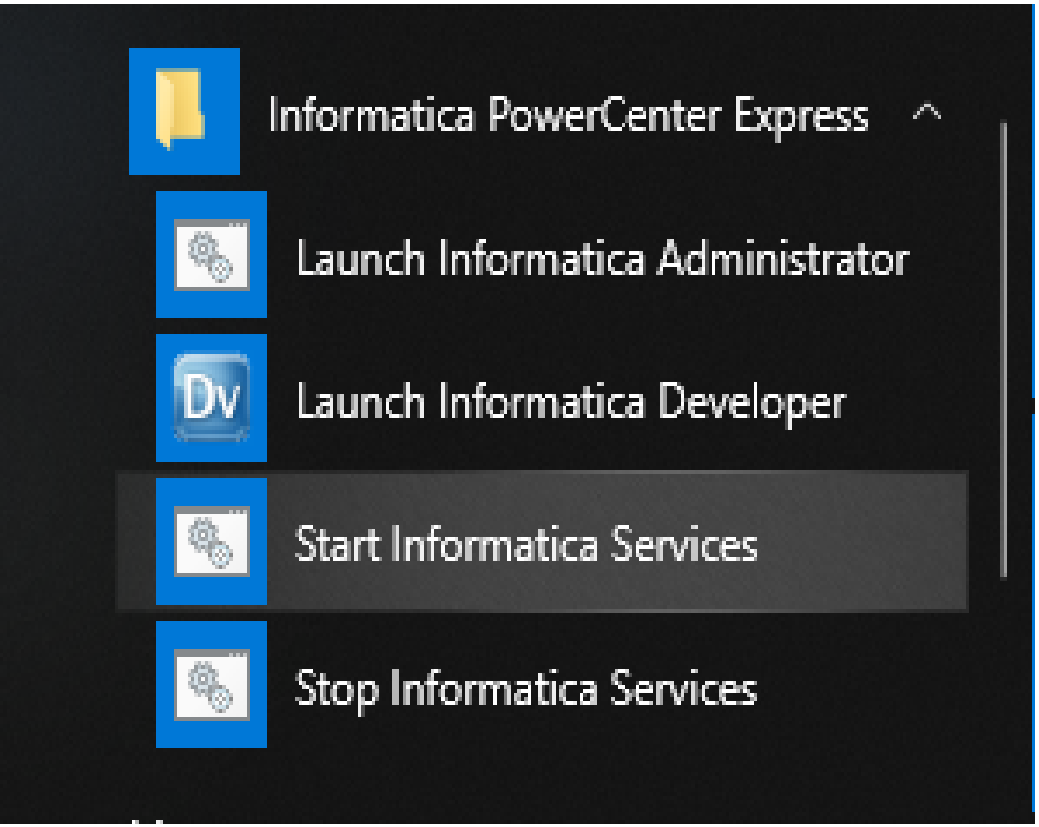
Sources & Targets

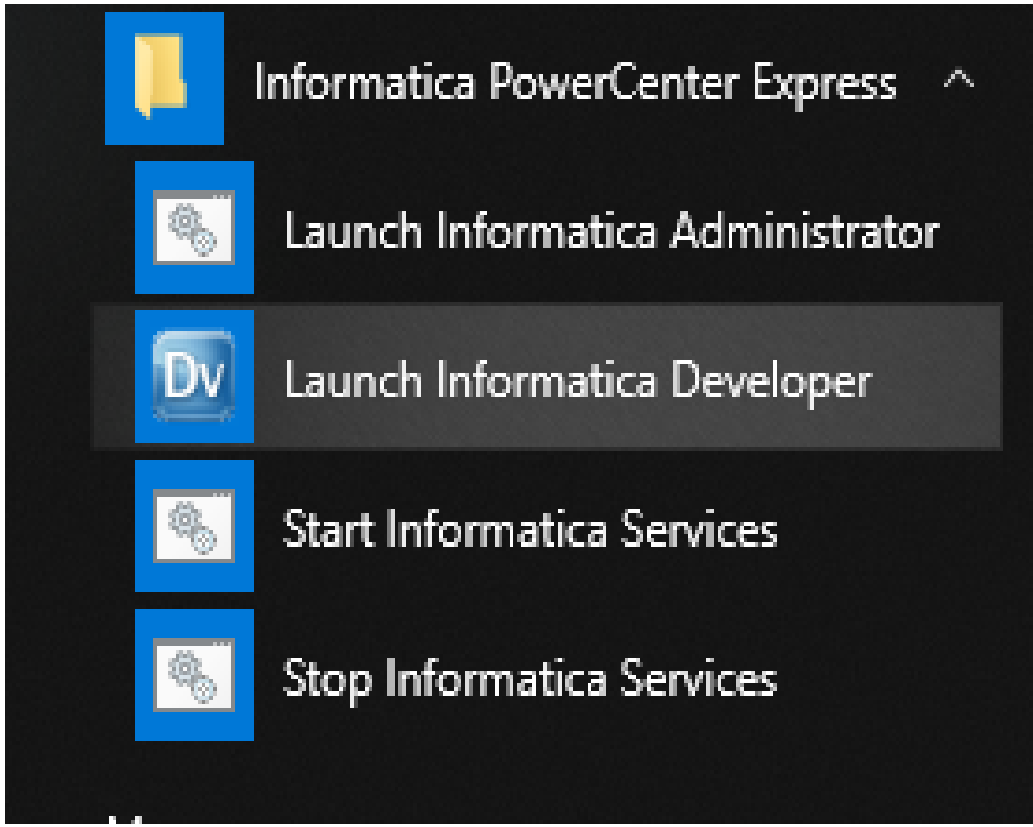
- Informatica being an ETL and Data integration tool, you would be always handling and transforming some form of data. The input to our mappings in Informatica is called source system. We import source definitions from the source and then connect to it to fetch the source data in our mappings. There can be different types of sources and can be located at multiple locations. Based upon your requirement the target system can be a relational or flat file system.

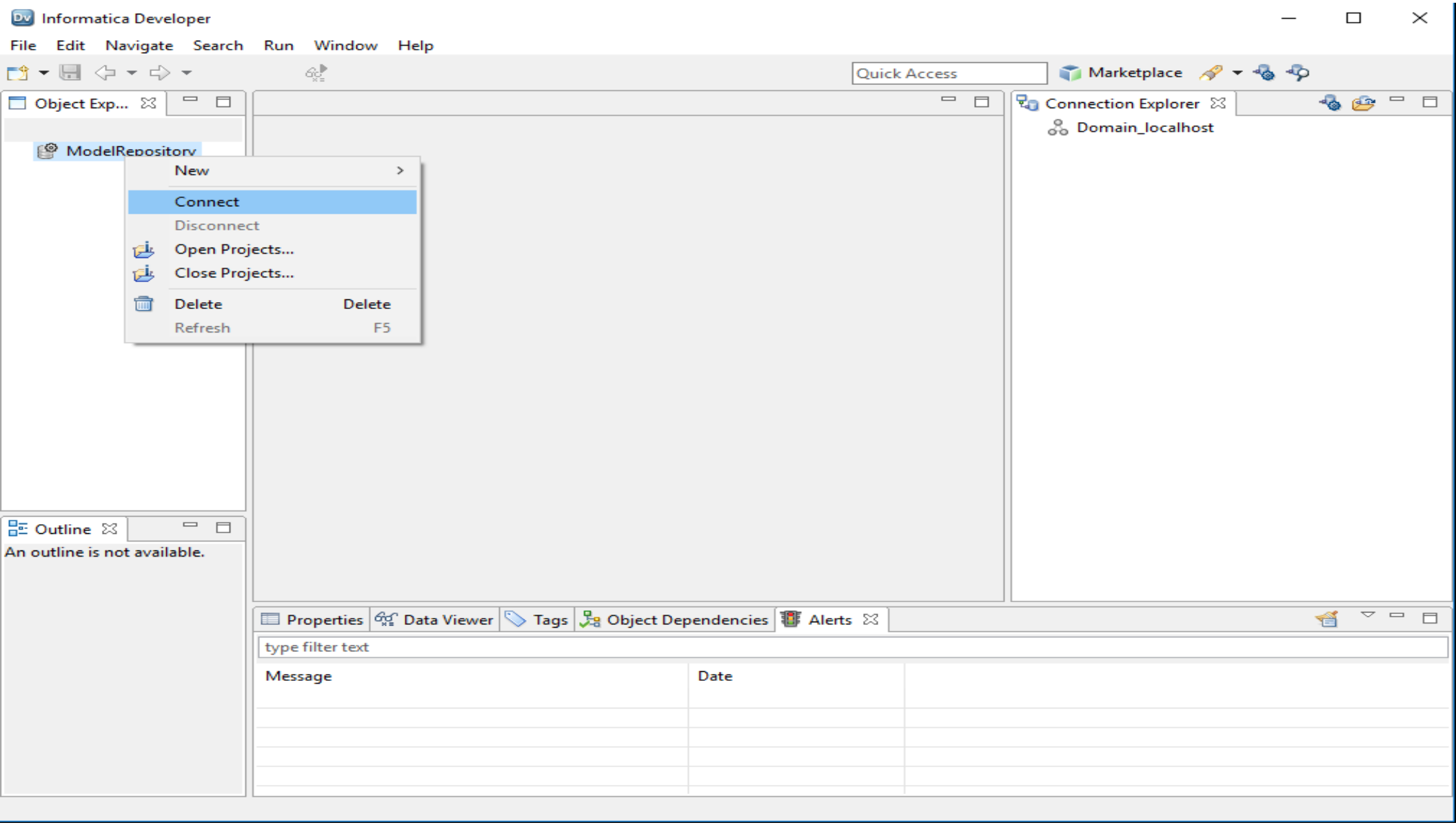
- Relational– these types of sources are database system tables. These database systems are generally owned by other applications which create and maintain this data. It can be a Customer Relationship Management Database, Human Resource Database, etc. for using such sources in Informatica we either get a replica of these datasets, or we get select privileges on these systems.

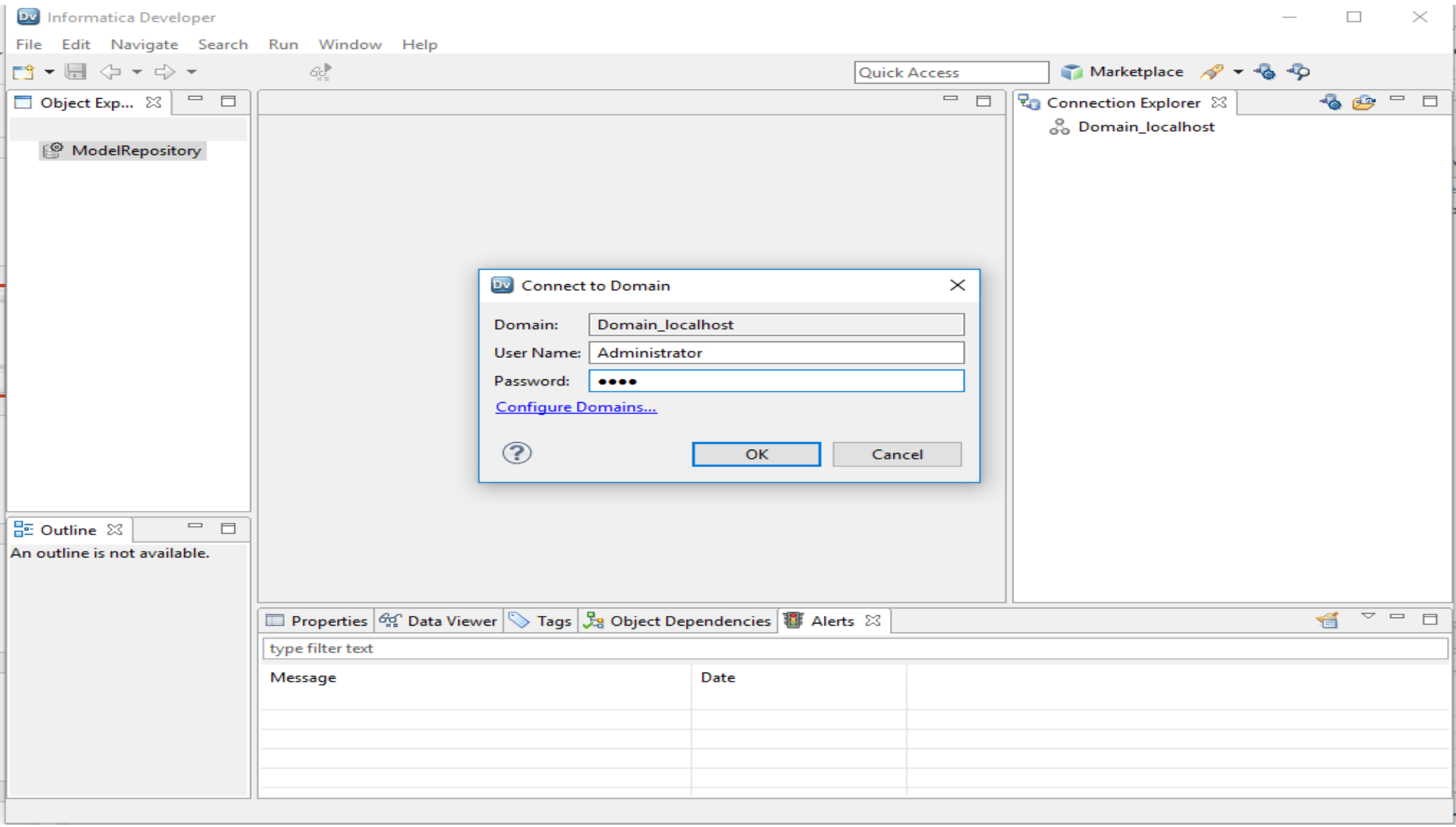
- Flat Files - Flat files are most common data sources after relational databases in Informatica. A flat file can be a comma separated file, a tab delimited file or fixed width file. Informatica supports any of the code pages like ascii or Unicode. To use the flat file in Informatica, its definitions must be imported similar to as we do for relational tables.

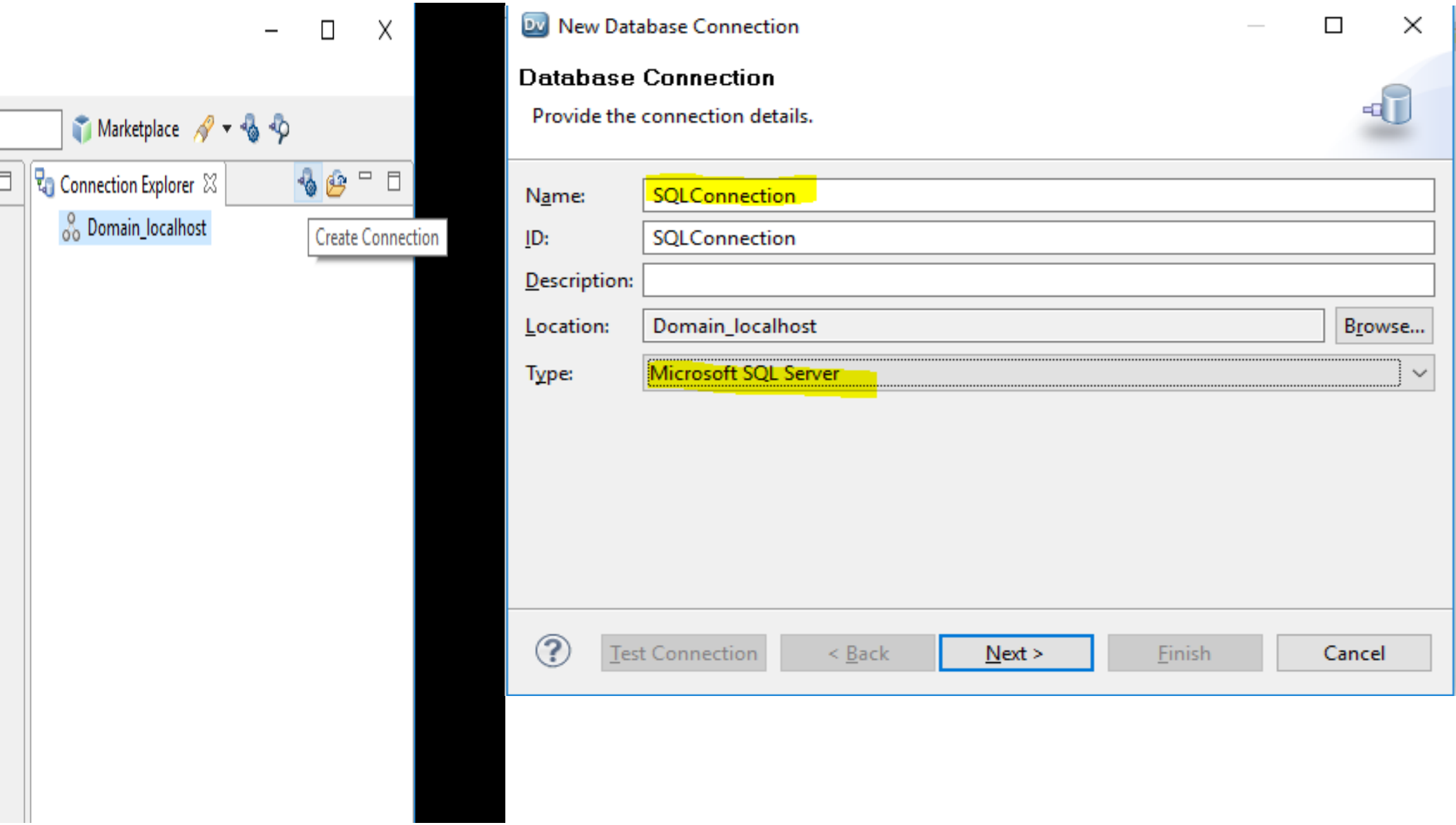
How to Use Informatica Developer

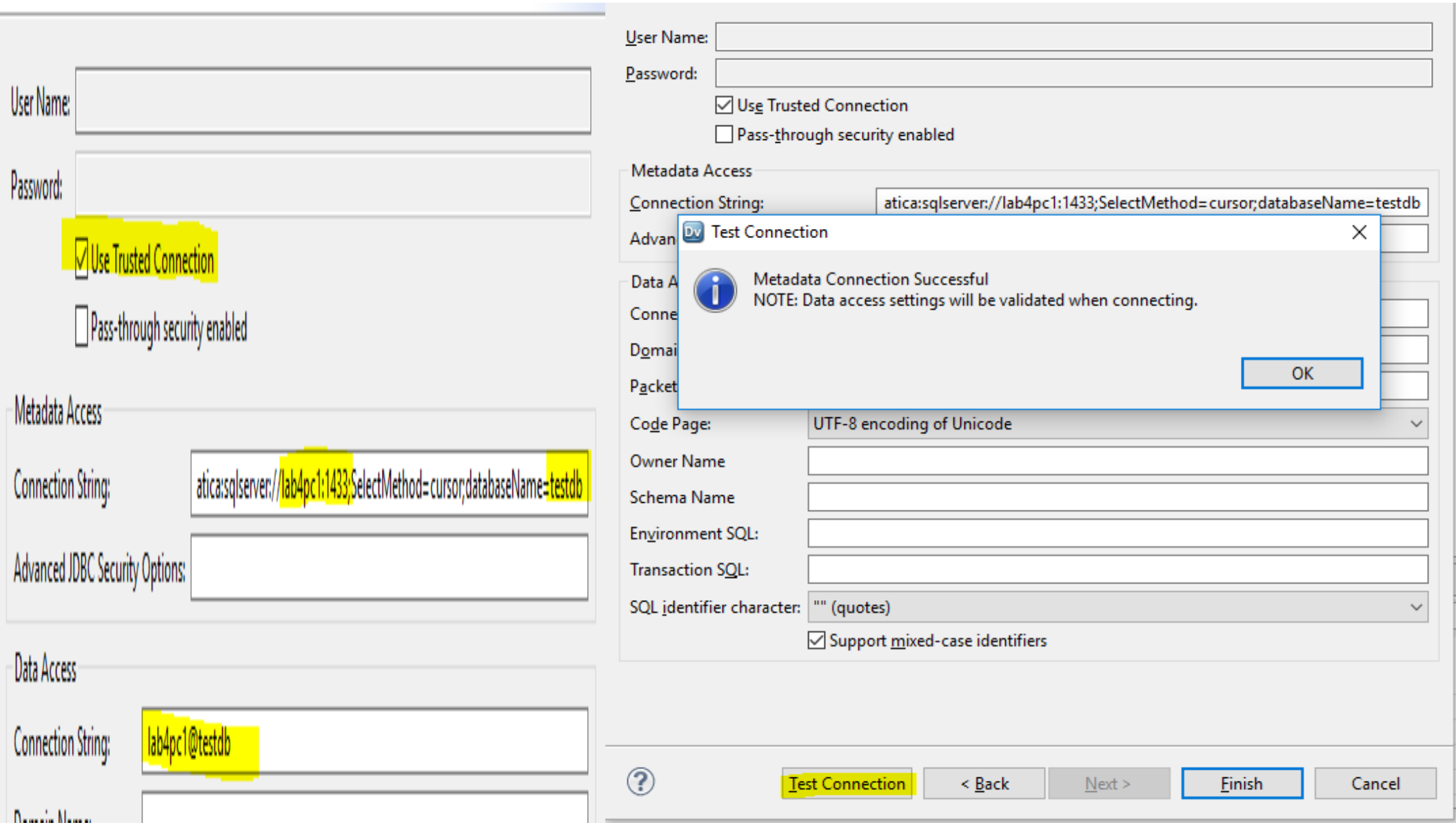


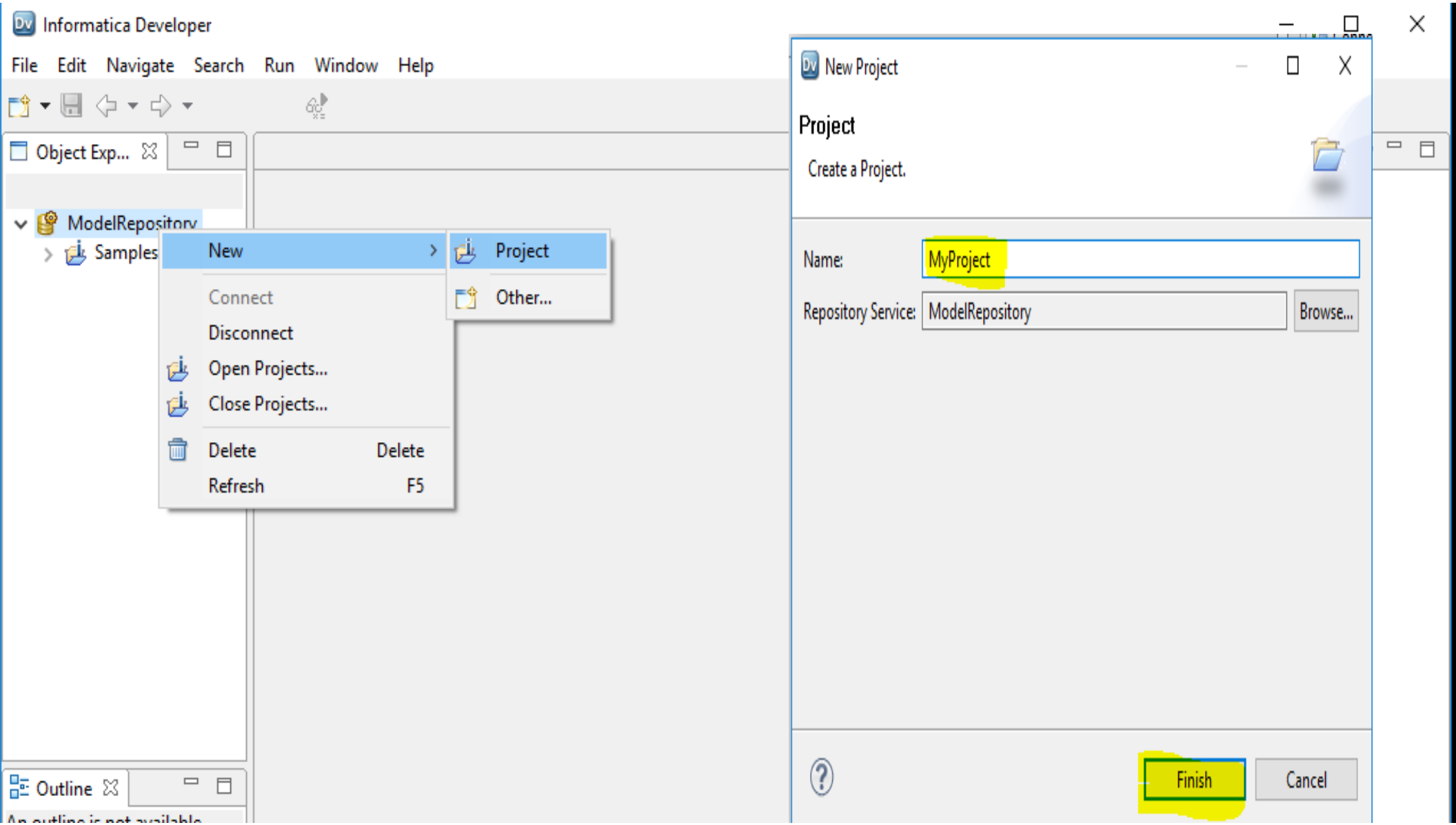


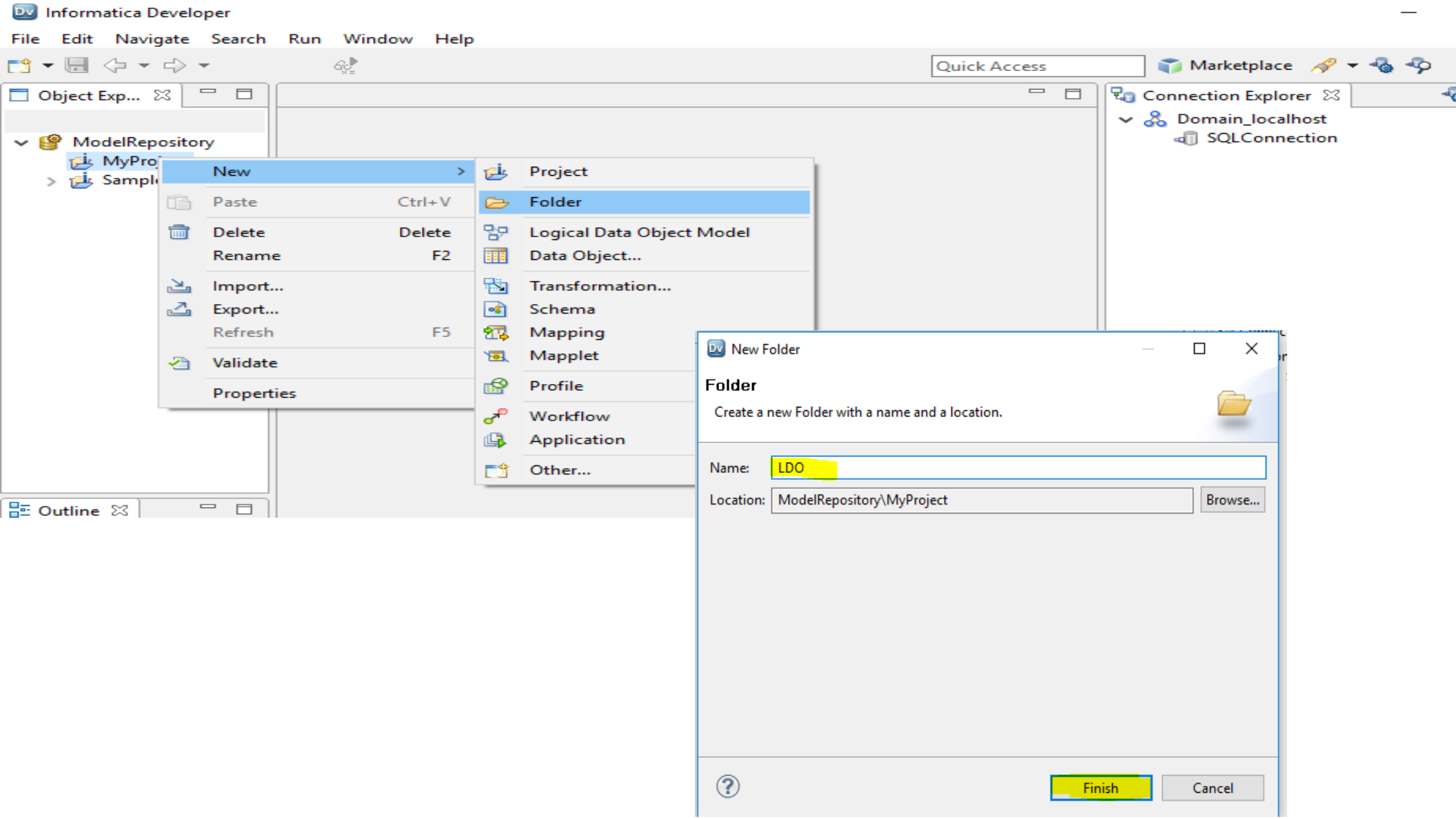

Informatica Developer
File
Edit
Navigate
Search
Run
Window
Help
Quick Access
Marketplace
Object Exp...
ModelRepository
New
Connect
Disconnect
Open Projects...
Close Projects...
Delete
Delete
Refresh
F5
Connection Explorer
Domain_localhost
Outline
An outline is not available.
Properties
Data Viewer
Tags
Object Dependencies
Alerts
type filter text
Message
Date

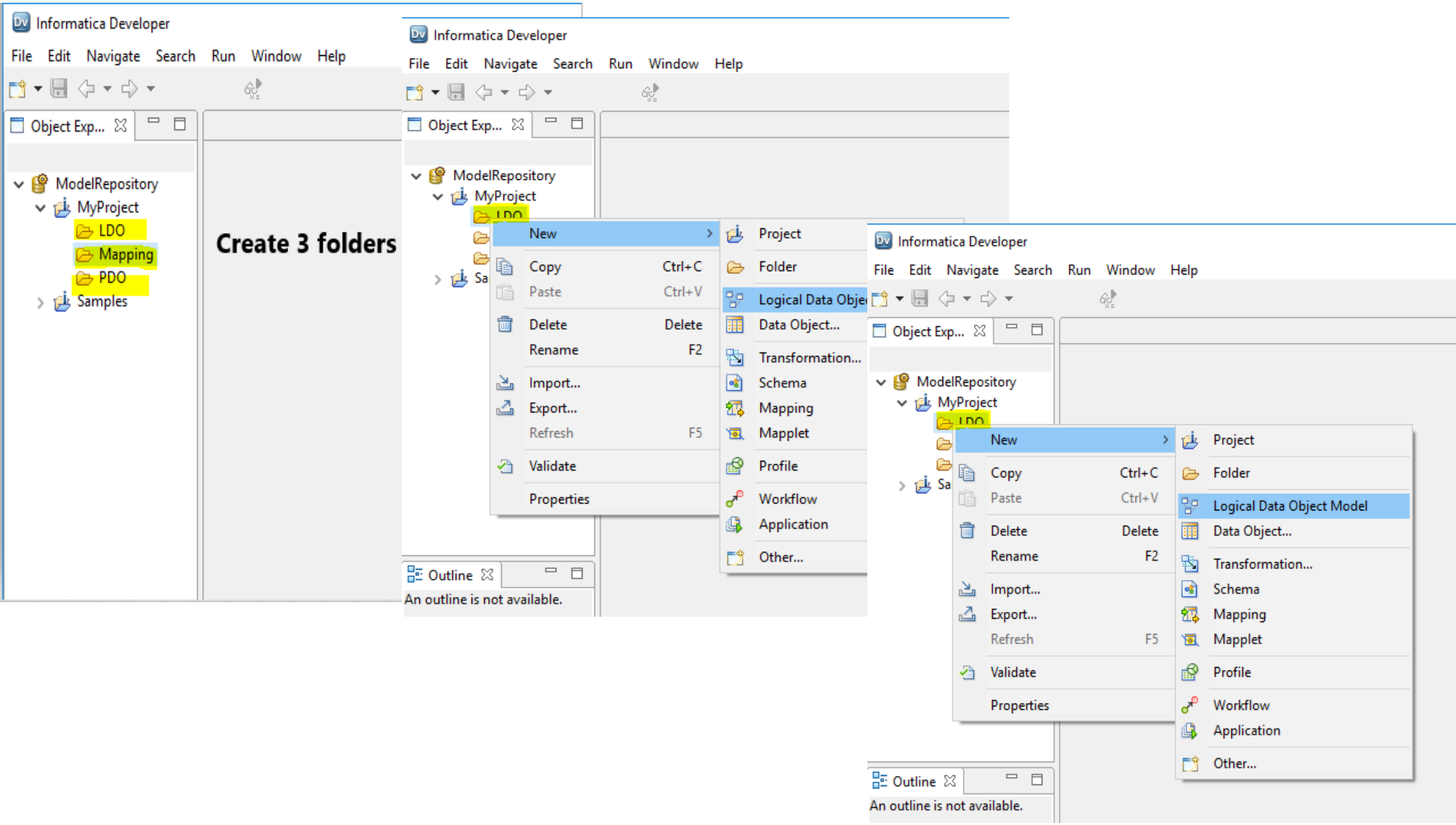

Informatica Developer
File Edit Navigate Search Run Window Help
Quick Access
Marketplace
Object Exp...
ModelRepository
Connection Explorer
Domain_localhost
Connect to Domain
Domain: Domain_localhost
User Name: Administrator
Password: ••••
Configure Domains...
OK
Cancel
Outline
An outline is not available.
Properties
Data Viewer
Tags
Object Dependencies
Alerts
type filter text
Message
Date

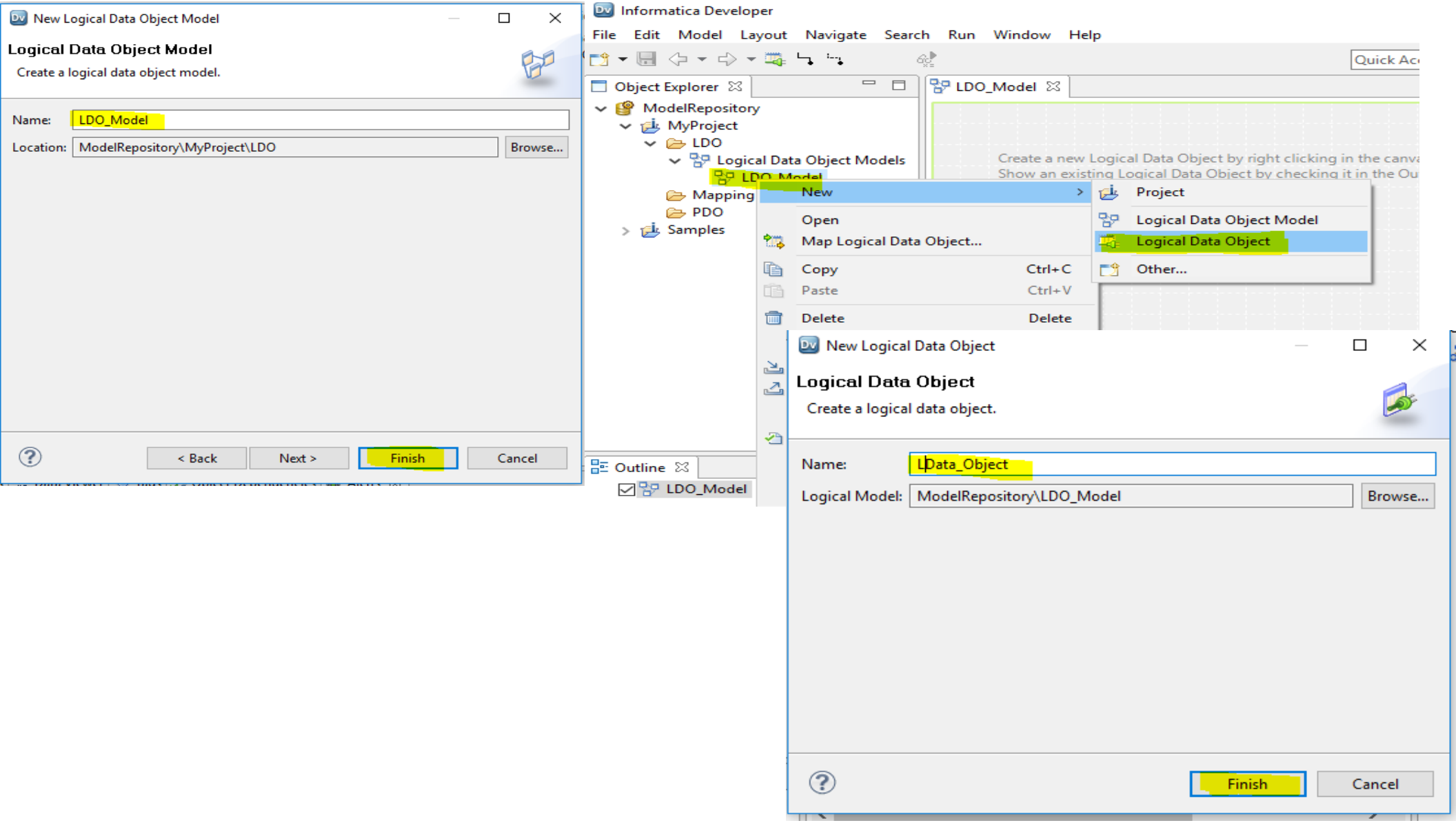

Marketplace
Connection Explorer
Domain_localhost
Create Connection
New Database Connection
Database Connection
Provide the connection details.
Name:
SQLConnection
ID:
SQLConnection
Description:
Location:
Domain_localhost
Browse...
Type:
Microsoft SQL Server
Test Connection
< Back
Next >
Finish
Cancel

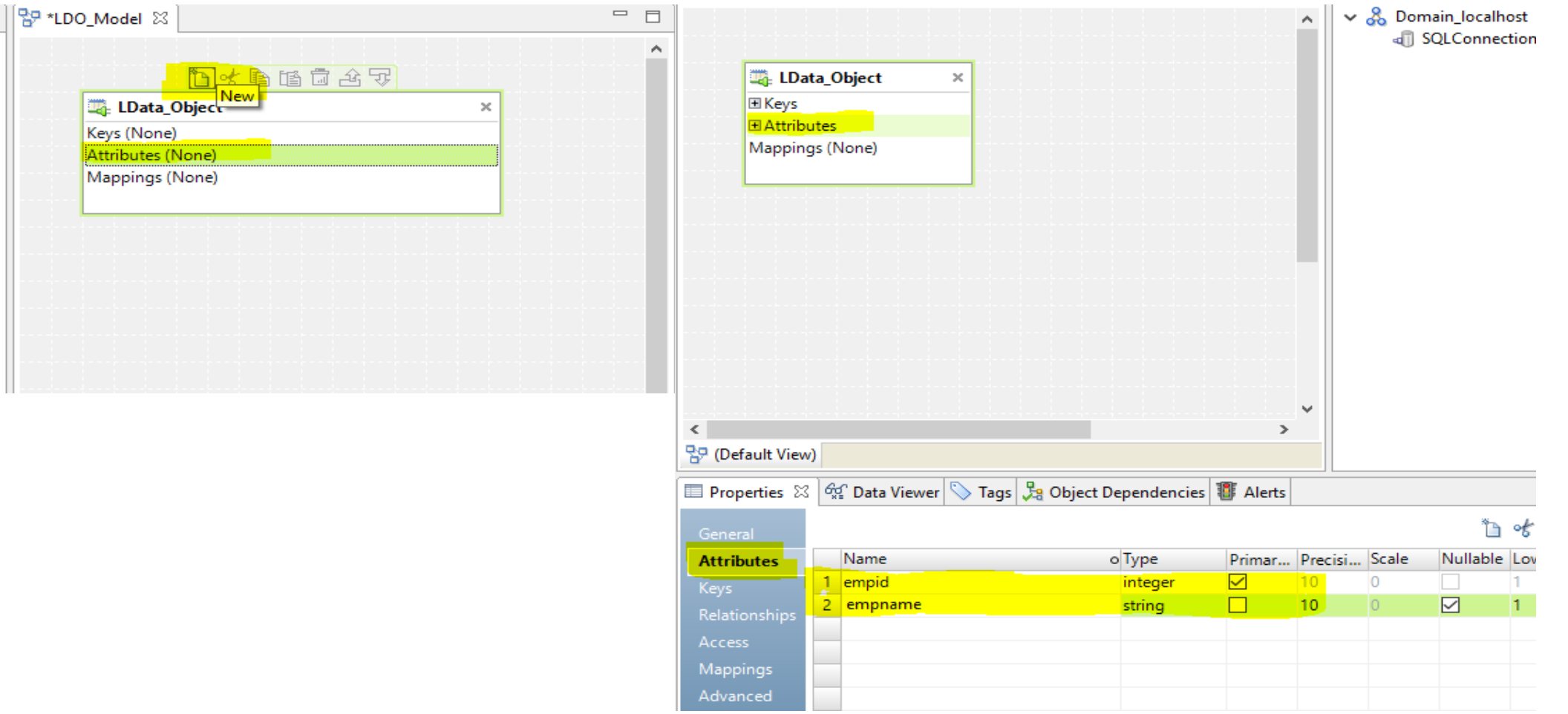

User Name:
Password:
Use Trusted Connection
Pass-through security enabled
Metadata Access
Connection String:
atica:sqlserver://lab4pc1:1433;SelectMethod=cursor;databaseName=testdb
Advanced JDBC Security Options:
Data Access
Connection String:
lab4pc1@testdb
Use Trusted Connection
Pass-through security enabled
Metadata Access
Connection String:
atica:sqlserver://lab4pc1:1433;SelectMethod=cursor;databaseName=testdb
Test Connection
Metadata Connection Successful
NOTE: Data access settings will be validated when connecting.
OK
Code Page:
UTF-8 encoding of Unicode
Owner Name
Schema Name
Environment SQL:
Transaction SQL:
SQL identifier character:
"" (quotes)
Support mixed-case identifiers
Test Connection
< Back
Next >
Finish
Cancel

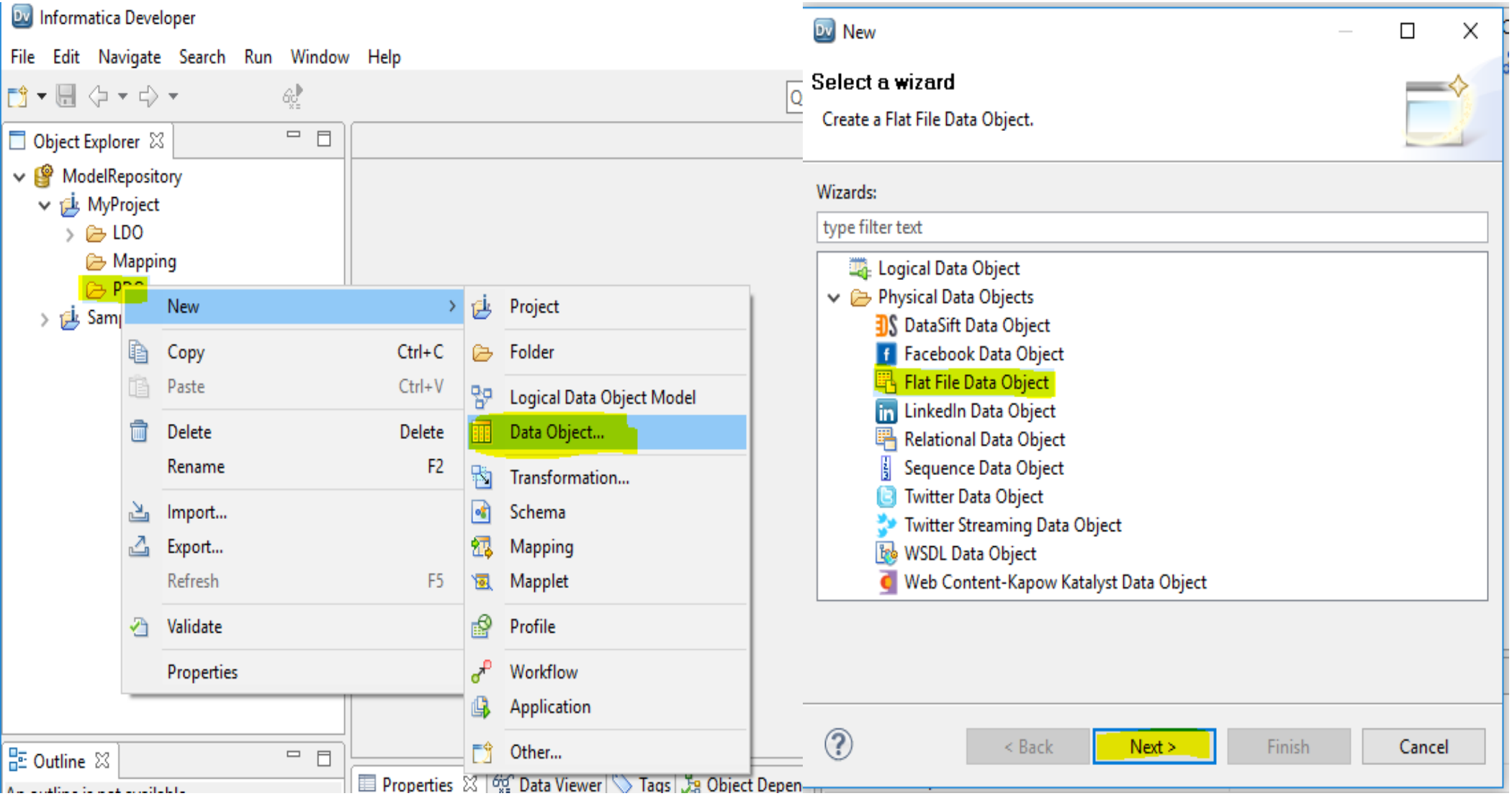

Informatica Developer
File Edit Navigate Search Run Window Help
Object Exp...
ModelRepository
Samples
New
Connect
Disconnect
Open Projects...
Close Projects...
Delete Delete
Refresh F5
Project
Other...
Outline
New Project
Project
Create a Project.
Name: MyProject
Repository Service: ModelRepository
Browse...
Finish
Cancel

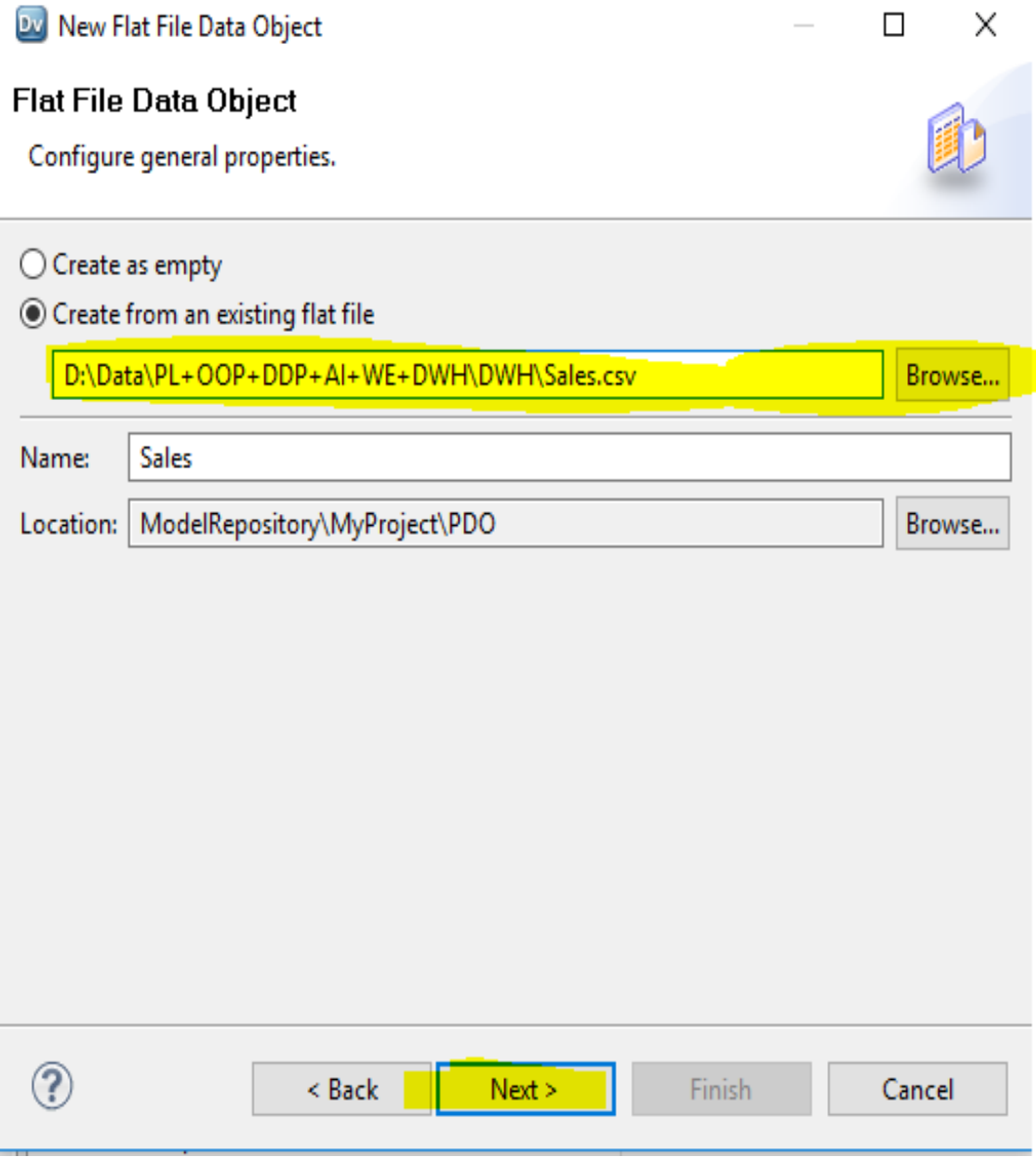

Informatica Developer
File Edit Navigate Search Run Window Help
Quick Access
Marketplace
Object Exp...
ModelRepository
MyProj
Sampl
Connection Explorer
Domain_localhost
SQLConnection
New
Paste Ctrl+V
Delete Delete
Rename F2
Import...
Export...
Refresh F5
Validate
Properties
Project
Folder
Logical Data Object Model
Data Object...
Transformation...
Schema
Mapping
Mapplet
Profile
Workflow
Application
Other...
Outline
New Folder
Folder
Create a new Folder with a name and a location.
Name: LDO
Location: ModelRepository\MyProject
Browse...
Finish
Cancel

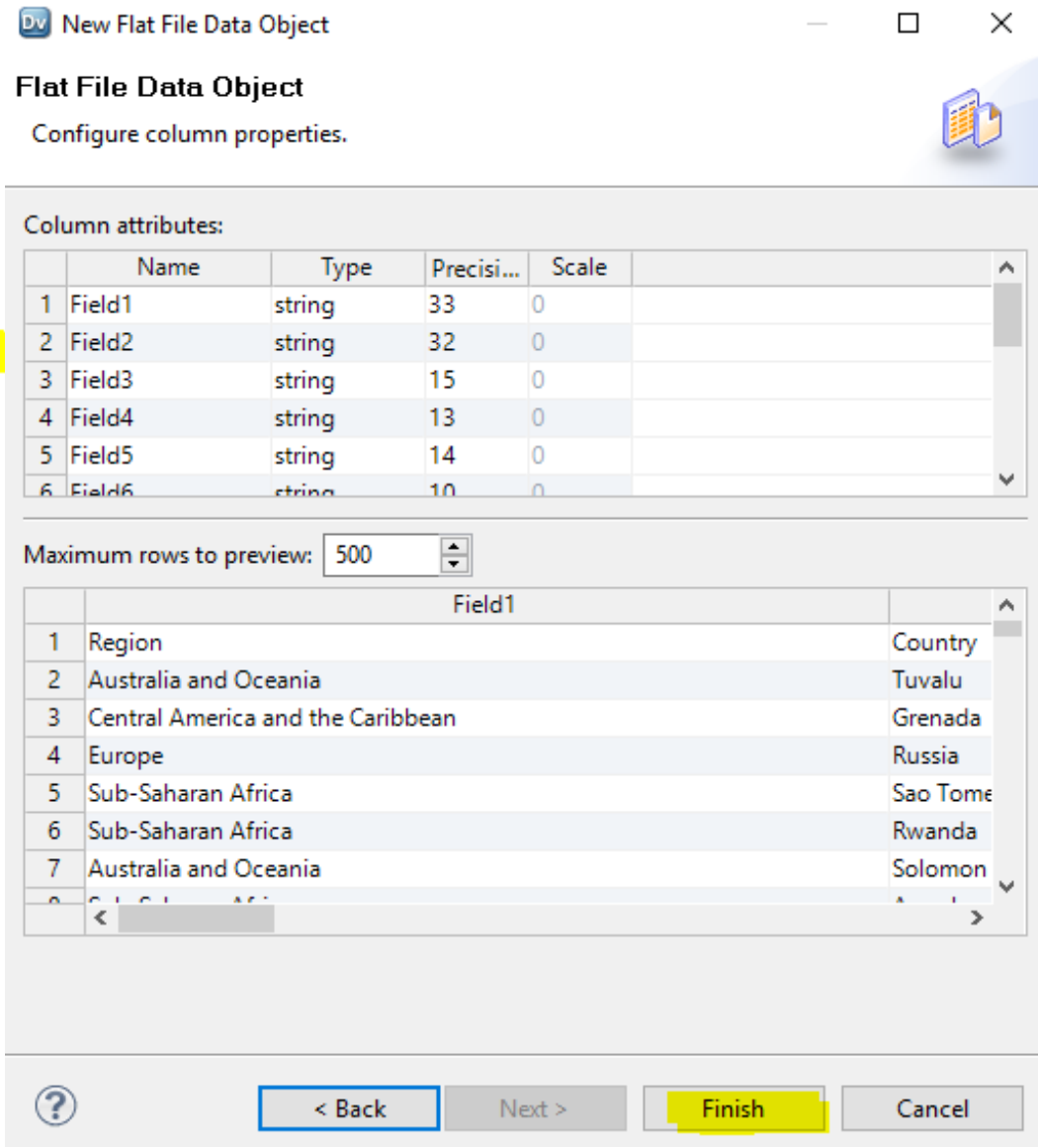

Informatica Developer
File Edit Navigate Search Run Window Help
Object Exp...
ModelRepository
MyProject
LDO
Mapping
PDO
Samples
Create 3 folders
Outline
An outline is not available.
New
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Rename F2
Import...
Export...
Refresh F5
Validate
Properties
Project
Folder
Logical Data Object Model
Data Object...
Transformation...
Schema
Mapping
Mapplet
Profile
Workflow
Application
Other...

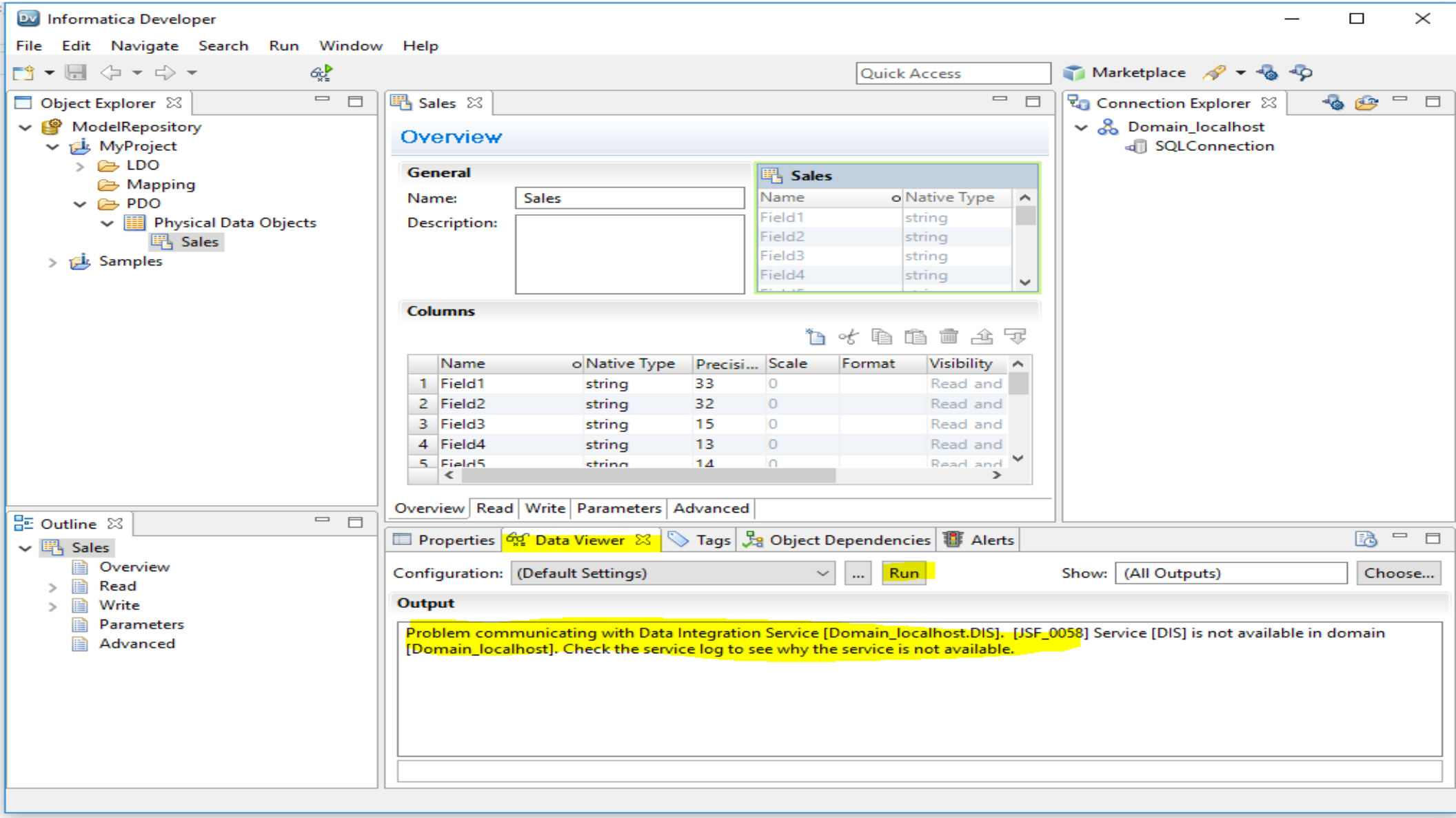

New Logical Data Object Model
Logical Data Object Model
Create a logical data object model.
Name: LDO_Model
Location: ModelRepository\MyProject\LDO
Browse...
< Back
Next >
Finish
Cancel
Informatica Developer
File Edit Model Layout Navigate Search Run Window Help
Object Explorer
ModelRepository
MyProject
LDO
Logical Data Object Models
LDO_Model
Mapping
PDO
Samples
LDO_Model
Create a new Logical Data Object by right clicking in the canva
Show an existing Logical Data Object by checking it in the Ou
New
Open
Map Logical Data Object...
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Project
Logical Data Object Model
Logical Data Object
Other...
Outline
LDO_Model
New Logical Data Object
Logical Data Object
Create a logical data object.
Name: LData_Object
Logical Model: ModelRepository\LDO_Model
Browse...
Finish
Cancel

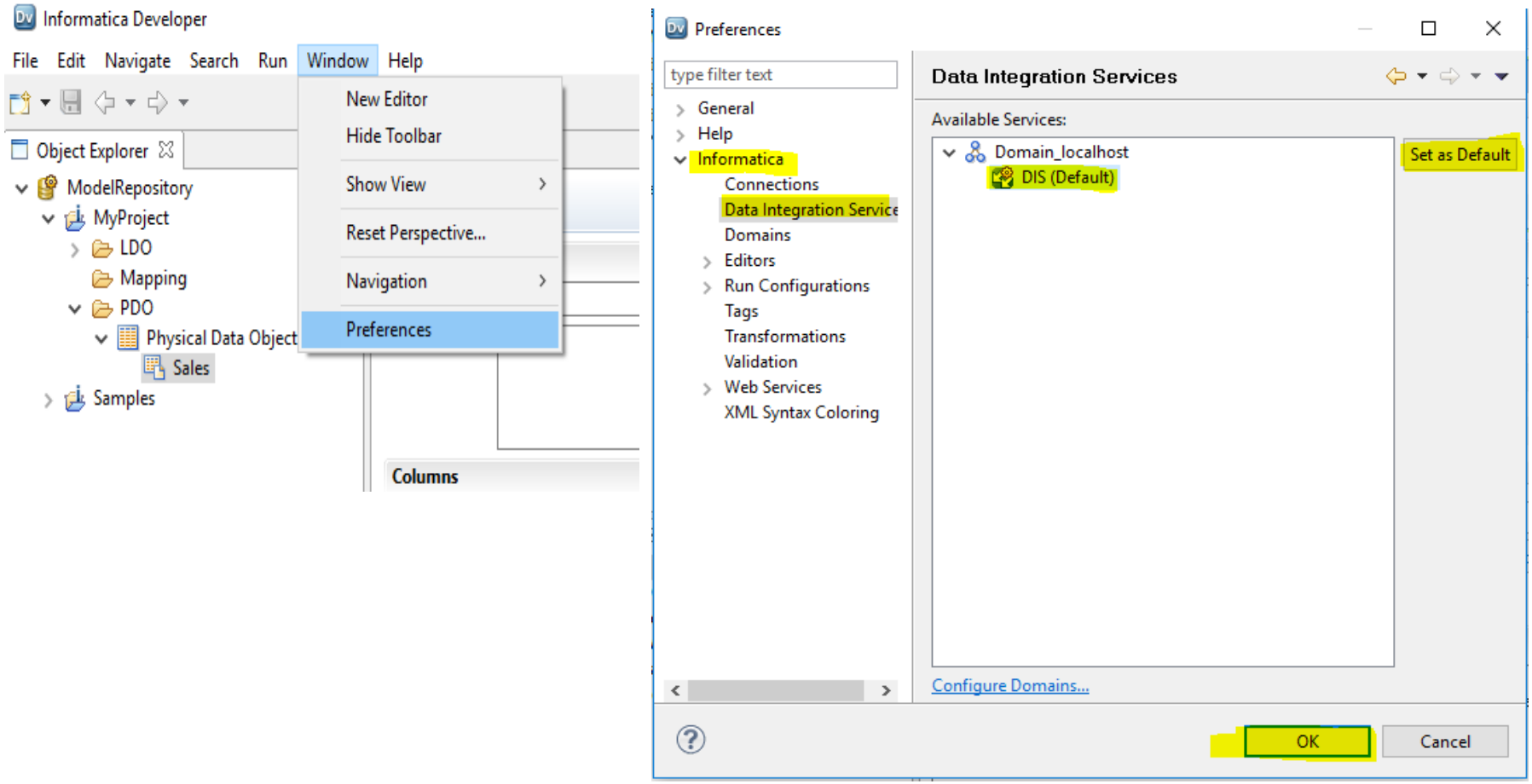

*LDO_Model
New
LData_Object
Keys (None)
Attributes (None)
Mappings (None)
LData_Object
Keys
Attributes
Mappings (None)
(Default View)
Domain_localhost
SQLConnection
Properties
Data Viewer
Tags
Object Dependencies
Alerts
General
Attributes
Keys
Relationships
Access
Mappings
Advanced
Name
Type
Primar...
Precisi...
Scale
Nullable
Low
1 empid integer 10 0 1
2 empname string 10 0 1

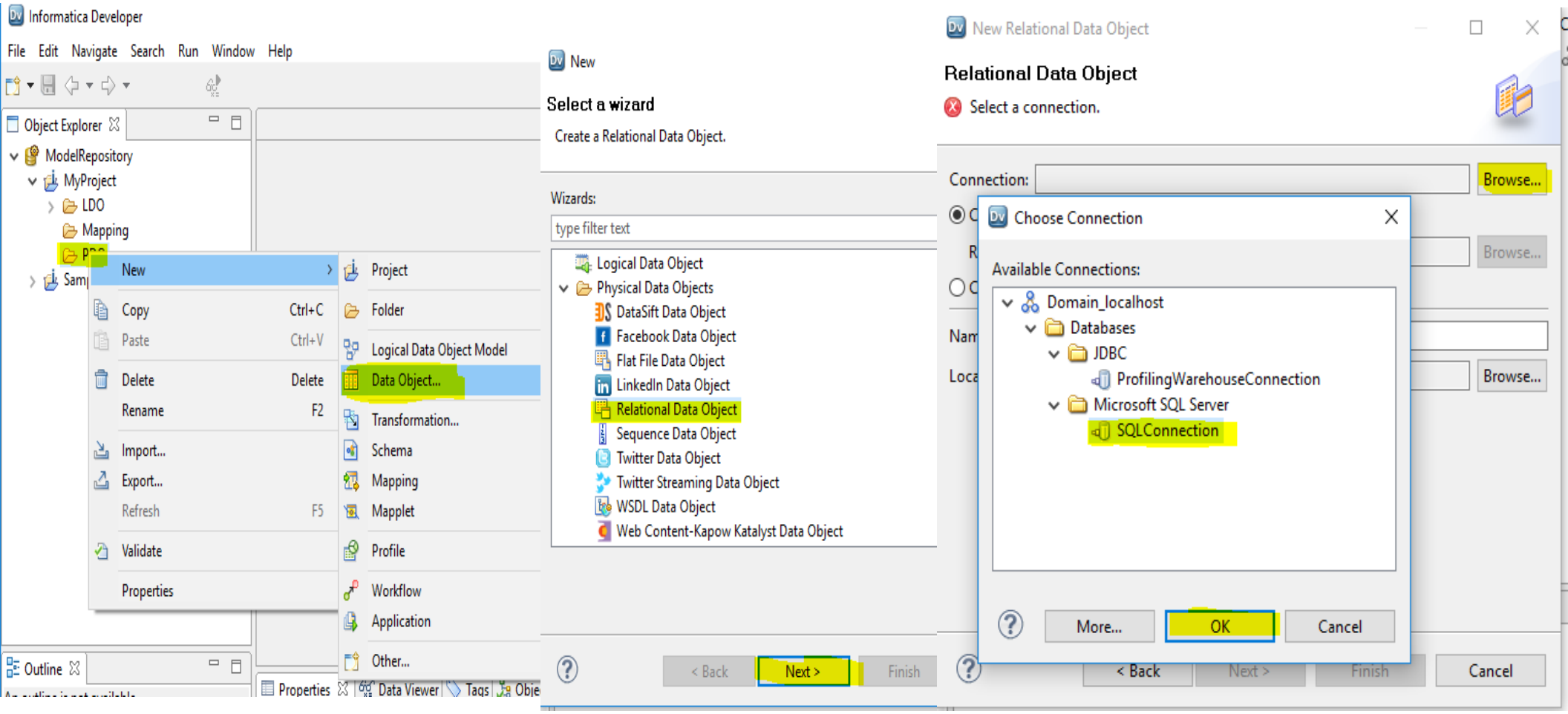

Informatica Developer
File Edit Navigate Search Run Window Help
Object Explorer
ModelRepository
MyProject
LDO
Mapping
New
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Rename F2
Import...
Export...
Refresh F5
Validate
Properties
Project
Folder
Logical Data Object Model
Data Object...
Transformation...
Schema
Mapping
Mapplet
Profile
Workflow
Application
Other...
Outline
Properties
Data Viewer
Tags
New
Select a wizard
Create a Flat File Data Object.
Wizards:
type filter text
Logical Data Object
Physical Data Objects
DataSift Data Object
Facebook Data Object
Flat File Data Object
LinkedIn Data Object
Relational Data Object
Sequence Data Object
Twitter Data Object
Twitter Streaming Data Object
WSDL Data Object
Web Content-Kapow Katalyst Data Object
< Back
Next >
Finish
Cancel

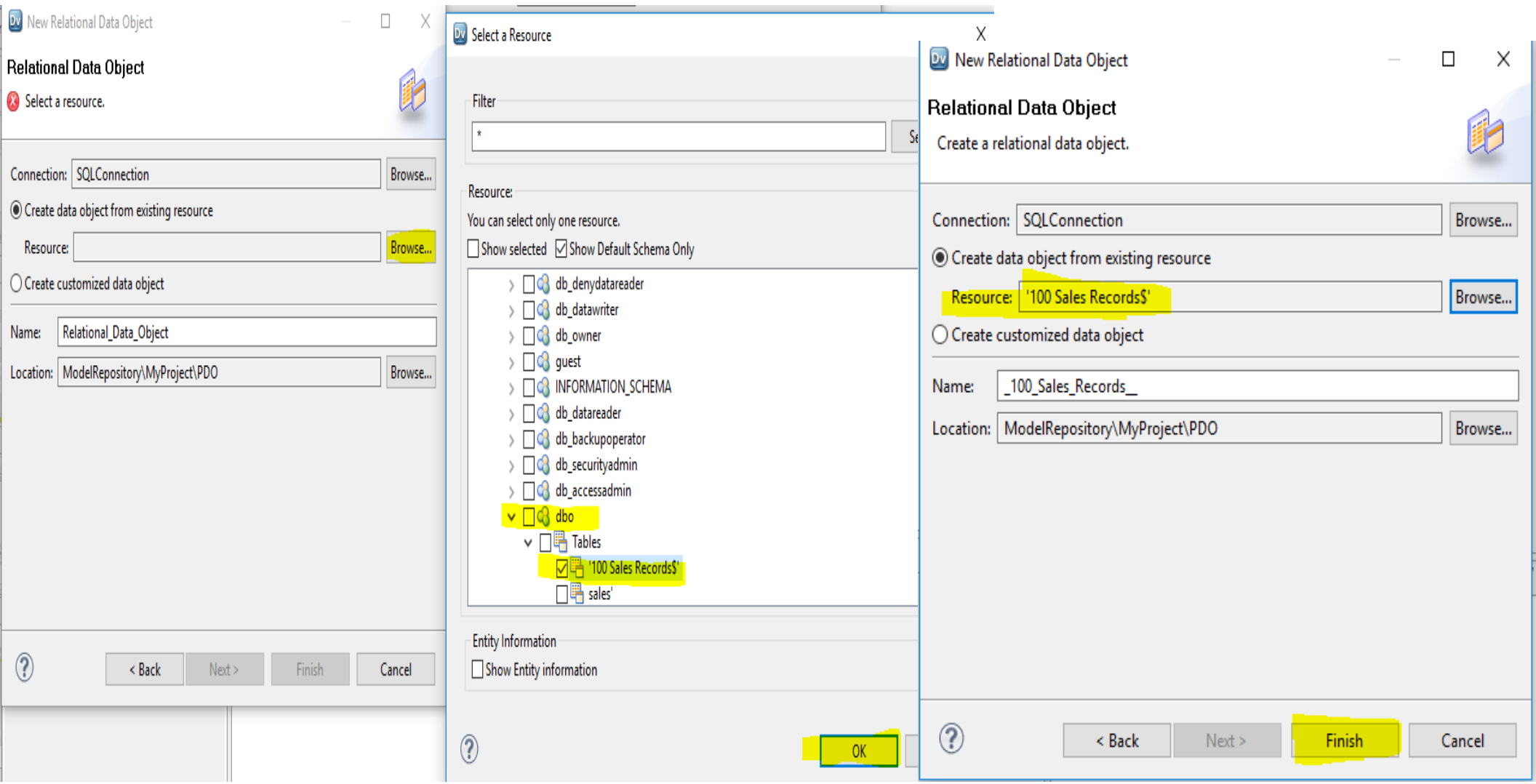

New Flat File Data Object
Flat File Data Object
Configure general properties.
Create as empty
Create from an existing flat file
D:\Data\PL+OOP+DDP+AI+WE+DWH\DWH\Sales.csv
Browse...
Name:
Sales
Location:
ModelRepository\MyProject\PDO
Browse...
< Back
Next >
Finish
Cancel

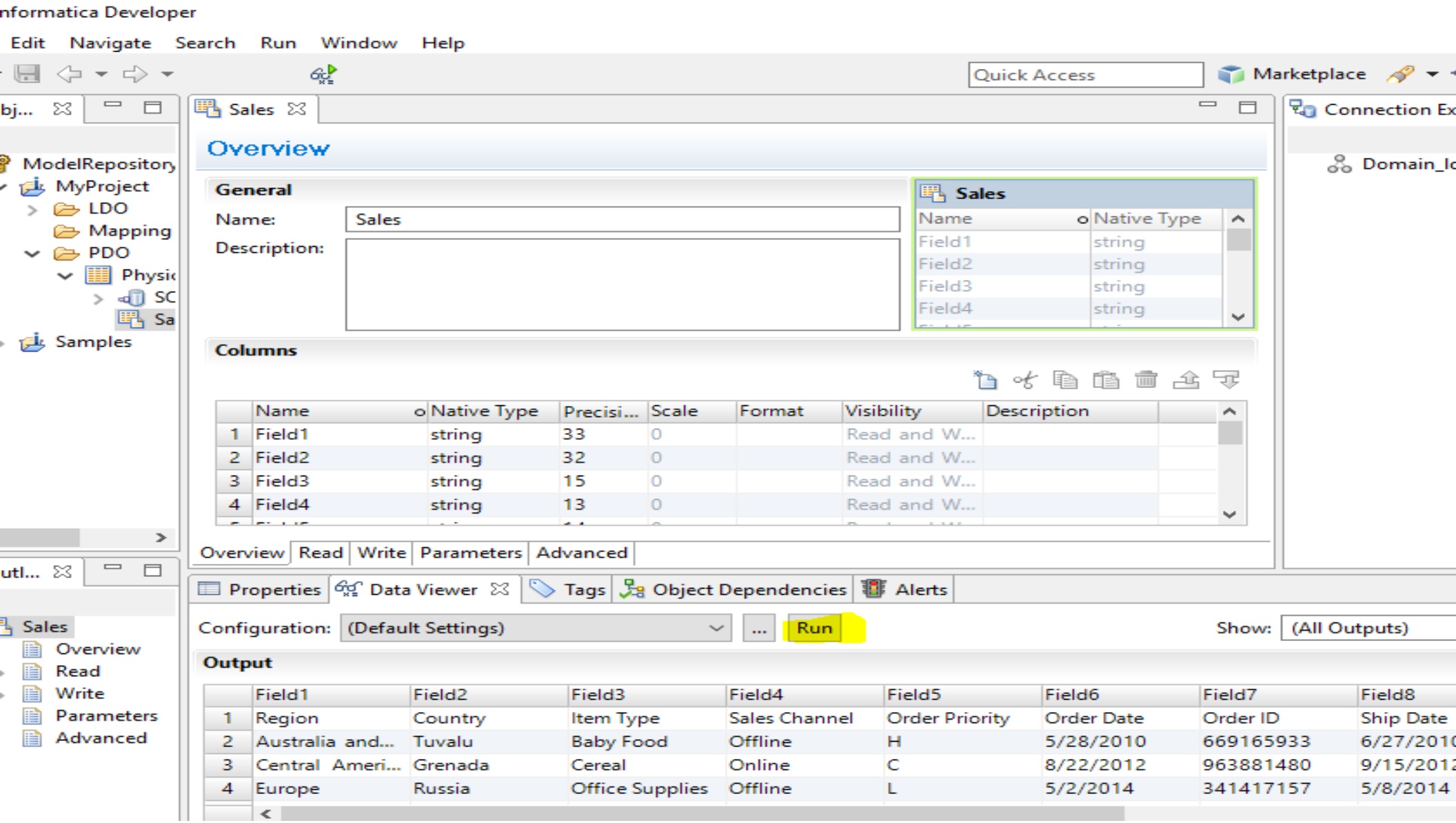

New Flat File Data Object
Flat File Data Object
Configure column properties.
Column attributes:
Name
Type
Precisi...
Scale
1 Field1 string 33 0
2 Field2 string 32 0
3 Field3 string 15 0
4 Field4 string 13 0
5 Field5 string 14 0
Maximum rows to preview:
500
Field1
1 Region Country
2 Australia and Oceania Tuvalu
3 Central America and the Caribbean Grenada
4 Europe Russia
5 Sub-Saharan Africa Sao Tome
6 Sub-Saharan Africa Rwanda
7 Australia and Oceania Solomon
< Back
Next >
Finish
Cancel

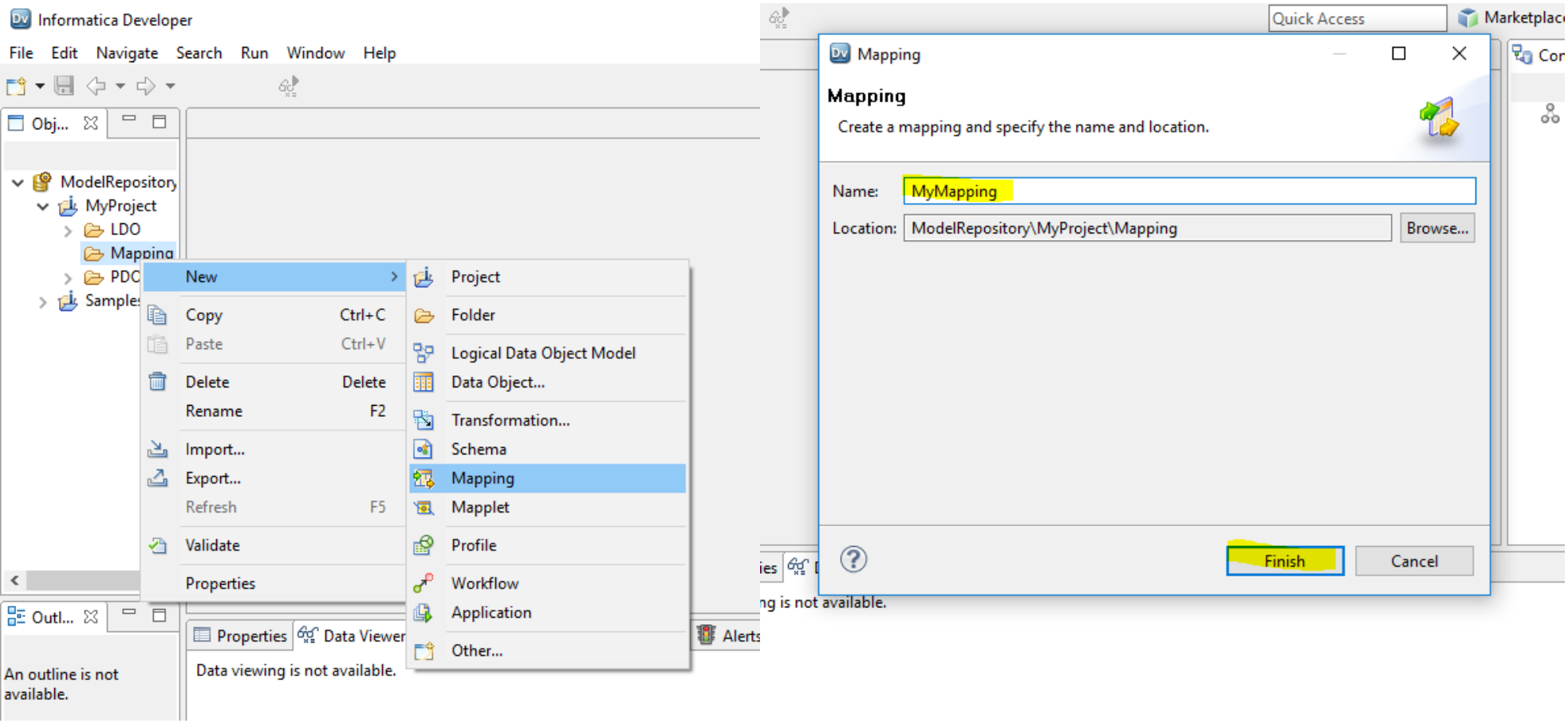

Informatica Developer
File Edit Navigate Search Run Window Help
Quick Access
Marketplace
Object Explorer
ModelRepository
MyProject
LDO
Mapping
PDO
Physical Data Objects
Sales
Samples
Sales
Overview
General
Name: Sales
Description:
Sales
Name Native Type
Field1 string
Field2 string
Field3 string
Field4 string
Columns
Name Native Type Precisi... Scale Format Visibility
1 Field1 string 33 0 Read and
2 Field2 string 32 0 Read and
3 Field3 string 15 0 Read and
4 Field4 string 13 0 Read and
5 Field5 string 14 0 Read and
Overview Read Write Parameters Advanced
Connection Explorer
Domain_localhost
SQLConnection
Outline
Sales
Overview
Read
Write
Parameters
Advanced
Properties Data Viewer Tags Object Dependencies Alerts
Configuration: (Default Settings) ... Run
Show: (All Outputs) Choose...
Output
Problem communicating with Data Integration Service [Domain_localhost.DIS]. [JSF_0058] Service [DIS] is not available in domain [Domain_localhost]. Check the service log to see why the service is not available.

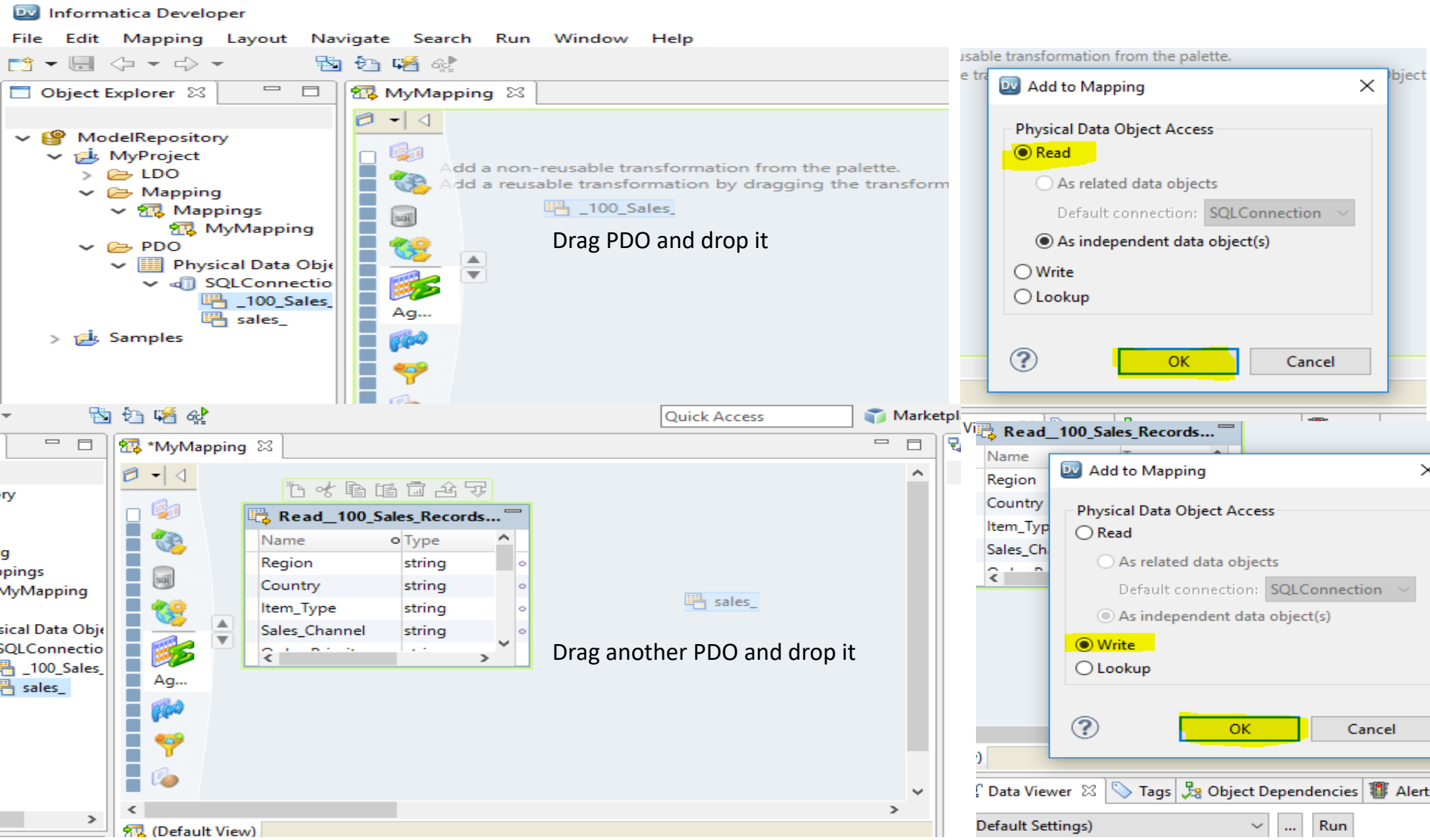

Informatica Developer
File Edit Navigate Search Run Window Help
New Editor
Hide Toolbar
Show View
Reset Perspective...
Navigation
Preferences
Object Explorer
ModelRepository
MyProject
LDO
Mapping
PDO
Physical Data Object
Sales
Samples
Columns
Preferences
type filter text
General
Help
Informatica
Connections
Data Integration Service
Domains
Editors
Run Configurations
Tags
Transformations
Validation
Web Services
XML Syntax Coloring
Data Integration Services
Available Services:
Domain_localhost
DIS (Default)
Set as Default
Configure Domains...
OK
Cancel

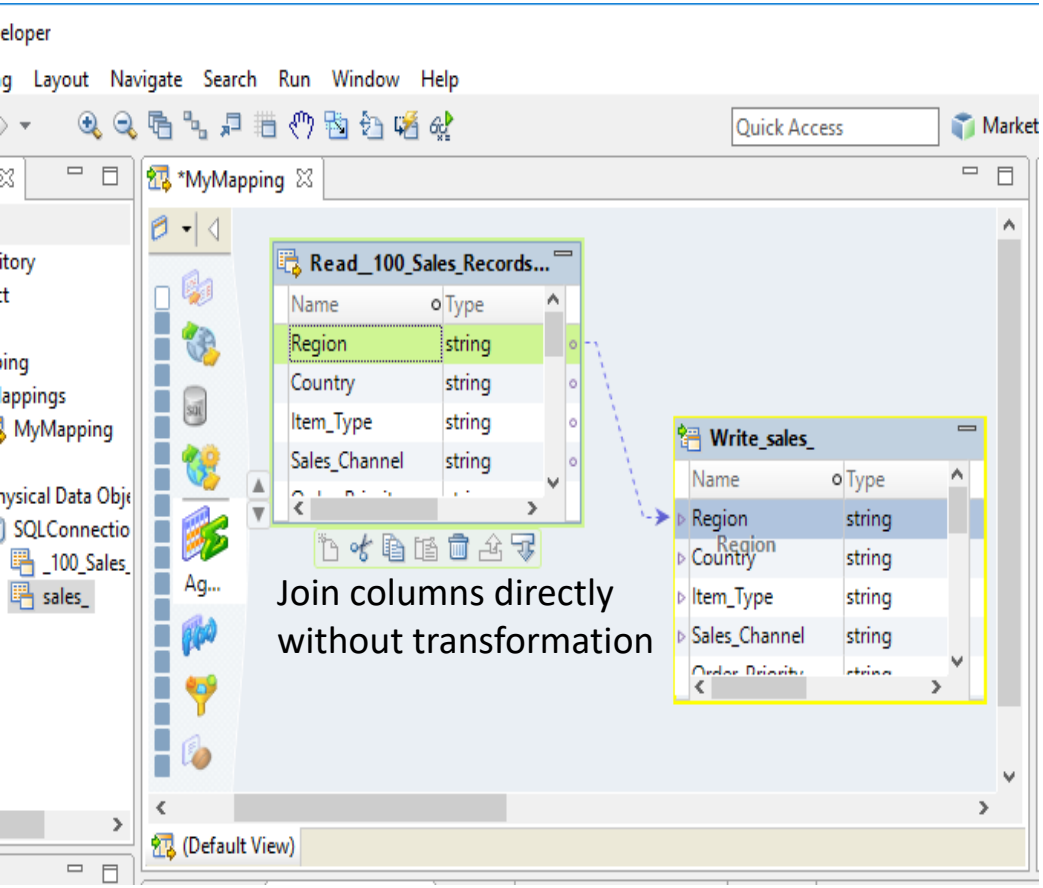

Informatica Developer
File Edit Navigate Search Run Window Help
Object Explorer
ModelRepository
MyProject
LDO
Mapping
New
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Rename F2
Import...
Export...
Refresh F5
Validate
Properties
Project
Folder
Logical Data Object Model
Data Object...
Transformation...
Schema
Mapping
Mapplet
Profile
Workflow
Application
Other...
Outline
Properties
Data Viewer
Tags
New
Select a wizard
Create a Relational Data Object.
Wizards:
type filter text
Logical Data Object
Physical Data Objects
DataSift Data Object
Facebook Data Object
Flat File Data Object
LinkedIn Data Object
Relational Data Object
Sequence Data Object
Twitter Data Object
Twitter Streaming Data Object
WSDL Data Object
Web Content-Kapow Katalyst Data Object
< Back
Next >
Finish
New Relational Data Object
Relational Data Object
Select a connection.
Connection:
Browse...
Choose Connection
Available Connections:
Domain_localhost
Databases
JDBC
ProfilingWarehouseConnection
Microsoft SQL Server
SQLConnection
More...
OK
Cancel
< Back
Next >
Finish
Cancel


New Relational Data Object
Relational Data Object
Select a resource.
Connection: SQLConnection
Browse...
Create data object from existing resource
Resource:
Browse...
Create customized data object
Name: Relational_Data_Object
Location: ModelRepository\MyProject\PDO
Browse...
< Back
Next >
Finish
Cancel
Select a Resource
Filter
*
Resource:
You can select only one resource.
Show selected
Show Default Schema Only
db_denydatareader
db_datawriter
db_owner
guest
INFORMATION_SCHEMA
db_datareader
db_backupoperator
db_securityadmin
db_accessadmin
dbo
Tables
'100 Sales Records$'
sales'
Entity Information
Show Entity information
OK
New Relational Data Object
Relational Data Object
Create a relational data object.
Connection: SQLConnection
Browse...
Create data object from existing resource
Resource: '100 Sales Records$'
Browse...
Create customized data object
Name: _100_Sales_Records__
Location: ModelRepository\MyProject\PDO
Browse...
< Back
Next >
Finish
Cancel

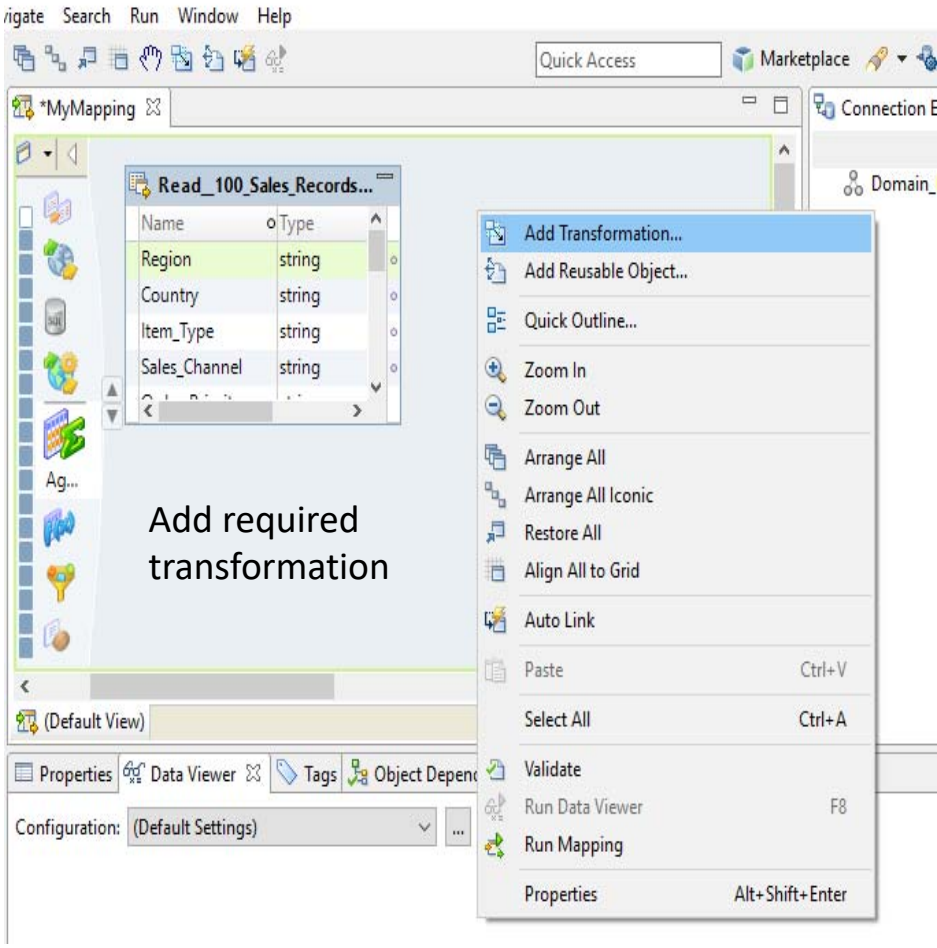

nformatica Developer
Edit Navigate Search Run Window Help
Quick Access
Marketplace
Sales
Overview
General
Name:
Sales
Description:
Sales
Name
Native Type
Field1 string
Field2 string
Field3 string
Field4 string
Columns
Name
Native Type
Precisi...
Scale
Format
Visibility
Description
1 Field1 string 33 0 Read and W...
2 Field2 string 32 0 Read and W...
3 Field3 string 15 0 Read and W...
4 Field4 string 13 0 Read and W...
Overview
Read
Write
Parameters
Advanced
Connection Ex
Domain_Io
ModelRepository
MyProject
LDO
Mapping
PDO
Samples
Sales
Overview
Read
Write
Parameters
Advanced
Properties
Data Viewer
Tags
Object Dependencies
Alerts
Configuration:
(Default Settings)
...
Run
Show:
(All Outputs)
Output
Field1 Field2 Field3 Field4 Field5 Field6 Field7 Field8
1 Region Country Item Type Sales Channel Order Priority Order Date Order ID Ship Date
2 Australia and... Tuvalu Baby Food Offline H 5/28/2010 669165933 6/27/2010
3 Central Ameri... Grenada Cereal Online C 8/22/2012 963881480 9/15/2012
4 Europe Russia Office Supplies Offline L 5/2/2014 341417157 5/8/2014

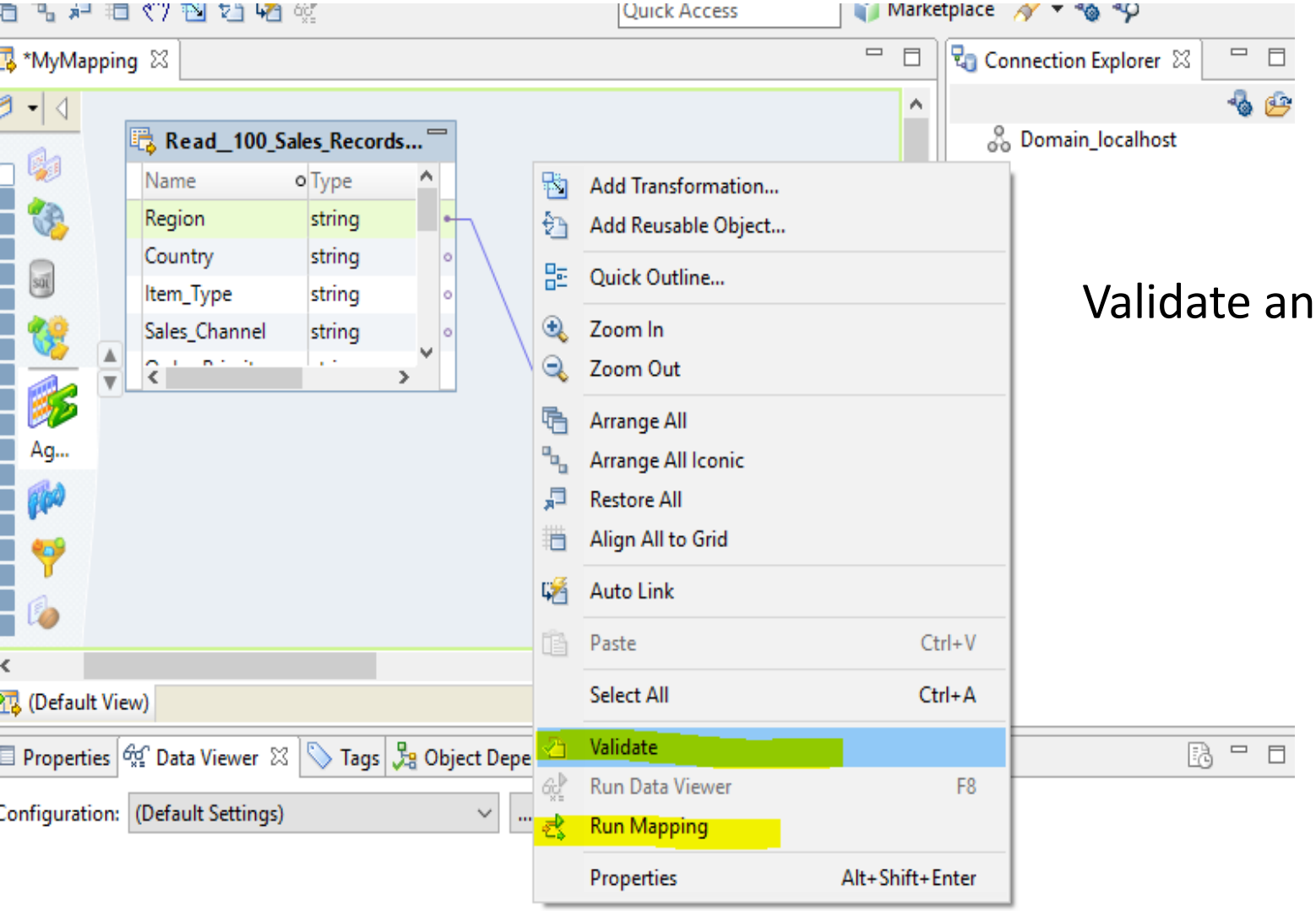

Informatica Developer
File Edit Navigate Search Run Window Help
Obj...
ModelRepository
MyProject
LDO
Mapping
PDC
Sample
New
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Rename F2
Import...
Export...
Refresh F5
Validate
Properties
Project
Folder
Logical Data Object Model
Data Object...
Transformation...
Schema
Mapping
Mapplet
Profile
Workflow
Application
Other...
Outl...
An outline is not available.
Properties
Data Viewer
Data viewing is not available.
Alerts
Quick Access
Marketplac
Mapping
Mapping
Create a mapping and specify the name and location.
Name: MyMapping
Location: ModelRepository\MyProject\Mapping
Browse...
Finish
Cancel
ng is not available.


Informatica Developer
File Edit Mapping Layout Navigate Search Run Window Help
Object Explorer
ModelRepository
MyProject
LDO
Mapping
Mappings
MyMapping
PDO
Physical Data Obje
SQLConnectio
_100_Sales_
sales_
Samples
MyMapping
Add a non-reusable transformation from the palette.
Add a reusable transformation by dragging the transform
_100_Sales_
Drag PDO and drop it
Add to Mapping
Physical Data Object Access
Read
As related data objects
Default connection: SQLConnection
As independent data object(s)
Write
Lookup
OK
Cancel
Quick Access
*MyMapping
Read__100_Sales_Records...
Name
Type
Region
string
Country
string
Item_Type
string
Sales_Channel
string
sales_
Drag another PDO and drop it
Add to Mapping
Physical Data Object Access
Read
As related data objects
Default connection: SQLConnection
As independent data object(s)
Write
Lookup
OK
Cancel
Data Viewer
Tags
Object Dependencies
Alerts
Default Settings)
Run
(Default View)


eloper
g Layout Navigate Search Run Window Help
Quick Access
Market
*MyMapping
Read_100_Sales_Records...
Name
Type
Region
string
Country
string
Item_Type
string
Sales_Channel
string
Write_sales_
Name
Type
Region
string
Country
string
Item_Type
string
Sales_Channel
string
Join columns directly
without transformation
(Default View)

OR


igate Search Run Window Help
Quick Access
Marketplace
*MyMapping
Read_100_Sales_Records...
Name
Type
Region
string
Country
string
Item_Type
string
Sales_Channel
string
Add required
transformation
(Default View)
Properties
Data Viewer
Tags
Configuration:
(Default Settings)
Add Transformation...
Add Reusable Object...
Quick Outline...
Zoom In
Zoom Out
Arrange All
Arrange All Iconic
Restore All
Align All to Grid
Auto Link
Paste
Ctrl+V
Select All
Ctrl+A
Validate
Run Data Viewer
F8
Run Mapping
Properties
Alt+Shift+Enter



Validate and then run mapping

Mapping in Informatica

- Mapping is a collection of source and target objects linked together by a set of transformations. These transformations consist of a set of rules, which define the data flow and how the data is loaded into the targets.
- A mapping consists of following set of objects:
- **Source Definition** - Source definition defines the structure and characteristic of the source, its underlying data types, type of the data source, etc.
- **Transformation** - Transformation objects define how the source data is transformed, and various functions can be applied during the process.
- **Target Definition** - Target definition defines the final target where the data will be loaded.
- **Links** - Links connect the source definition to different transformations and target tables. It defines how the data flows from source to target and the transformations.

Why do you need Mapping?

- Mapping is an object in Informatica with the help of which you can define how the source data is modified before it reaches the destination or target object. Like if you have employee name as "Bill Clinton" in your source system and in the target system the requirement is to have employee name in the format as "Clinton Bill", such operations can be designed at the mapping level. In basic terms, what you do with the source data is defined at the mapping level.

- Mapping is the basic Informatica object with the help of which we can define the data transformation details and source/target object characteristics. Mappings help us to define the data transformation at the individual column levels for each row. Even in a single mapping you can handle multiple sources and targets.

Components of Mapping

- Source tables
- Mapping parameters and variables
- Target objects
- Mapping transformations
- Mapping Source: Mapping sources are the objects from where you fetch the source data. It can be a database table, flat file, XML source or COBOL file source
- Mapping target: Mapping target is our destination objects where final processed data gets loaded. Mapping target can be a relational table of a database, a flat file or XML file. Sources and targets are mandatory in any mapping, their type can differ
- Mapping Parameters and Variables: Mapping parameters and variables helps you to create temporary variable objects which will help you to define and store temporary values while mapping data processing. Mapping parameters and variables are optional users defined data types, which can be created for a mapping and can be referenced and updated for a specific requirement. We will learn more about mapping parameters and variables in this section
- Mapplets: They are objects which consist of a set of transformation, source or targets. Mapplets are generally created to reuse the existing functionality of a set of transformations. It can be used in any no of mappings.

Stage Mapping

- A stage mapping is a mapping in where we create the replica of the source table. For Example, in a production system if you have an "employee" table then you can create an identical table "employee_stage" in ETL schema.
- Having a local stage table offers various advantages, like production downtime, won't affect your ETL system because you have your own "employee_stage" table, instead of referring to production "employee" table. In a Production system, there can be other operations and processes which affect the performance. However, when you have replica staging table, only ETL processes will access it. This offers performance benefits.
- In Stage Mappings,
- Source and Target tables have identical structures
- The data in the target table is a replica of source table data or
- Data in stage (target) table is a subset of source data.
- For example, if your source table contains employee details of deptno 10, 20, 30, and 40. The staging table can be a table having employee records of deptno 10 & 30 only.
- The purpose of creating stage tables in Data warehouse is to make the process of data transformation efficient by fetching only those data which is relevant to us and also to minimize the dependency of ETL/Data Warehouse from the real-time operational system.

What is Transformation?

- Transformations is in Informatica are the objects which creates, modifies or passes data to the defined target structures (tables, files or any other target).
- The purpose of the transformation in Informatica is to modify the source data as per the requirement of target system. It also ensures the quality of the data being loaded into the target.
- Informatica provides various transformations to perform specific functionalities.
- For example, performing tax calculation based upon source data, data cleansing operation, etc. In transformations, we connect the ports to pass data to it, and transformation returns the output through output ports.

Classification of Transformation

- Transformation is classified into two categories, one based on connectivity, and other based on the change in no of rows.

- Types of transformation based on connectivity
- Connected Transformations
- Unconnected Transformations

- Types of transformations based on the change in no of rows
- Active Transformations
- Passive Transformations

Connected and Unconnected Transformation

- In Informatica, during mappings the transformations which are connected to other transformations are called connected transformations.
- For example, Source qualifier transformation of Source table EMP is connected to filter transformation to filter employees of a dept.
- Those transformations that are not connected to any other transformations are called unconnected transformations.
- Their functionality is used by calling them inside other transformations like Expression transformation. These transformations are not part of the pipeline.
- The connected transformations are preferred when for every input row, transformation is called or is expected to return a value. For example, for the zip codes in every row, the transformation returning city name.
- The unconnected transformations are useful when their functionality is only required periodically or based upon certain conditions. For example, calculation the tax details if tax value is not available.

Active and Passive Transformation

- Active Transformations are those who modifies the data rows and the number of input rows passed to them. For example, if a transformation receives ten number of rows as input, and it returns fifteen number of rows as an output then it is an active transformation. The data in the row is also modified in the active transformation.
- Passive transformations are those who does not change the number of input rows. In passive transformations the number of input and output rows remain the same, only data is modified at row level.
- In the passive transformation, no new rows are created, or existing rows are dropped.

Transformation	Type	Description
Aggregator	Active/Connected	Performs aggregate calculations.
ApplicationSourceQualifier	Active/Connected	Represents therows that the Integration Service reads from an application, such as an ERP source, when it runs a session.
Custom	ActiveorPassive/Connected	Calls a procedure in a shared library or DLL.
DataMasking	Passive/Connected	Replaces sensitive production data with realistic test data for non-production environments.
Expression	Passive/Connected	Calculates a value.
ExternalProcedure	Passive/ConnectedorUnconnected	Calls a procedure in a shared library or in the COM layer of Windows.
Filter	Active/Connected	Filters data.
HTTP	Passive/Connected	Connects to an HTTP server to read or update data.
Input	Passive/Connected	Defines mapplet input rows. Available in the Mapplet Designer
Java	ActiveorPassive/Connected	Executes user logic coded in Java.The byte code for the user logic is stored in the repository
Joiner	Active/Connected	Joins data from different databases or flat file systems.
Lookup	ActiveorPassive/ConnectedorUnconnected	Lookup and return data from a flat file, relational table, view, or synonym.

Transformation	Type	Description
Normalizer	Active/Connected	Source qualifier for COBOL sources. Can also use in the pipeline to normalize data from relational or flat file sources.
Output	Passive/Connected	Defines mapplet output rows. Available in the Mapplet Designer.
Rank	Active/Connected	Limits records to a top or bottom range.
Router	Active/Connected	Routes data into multiple transformations based on group conditions.
SequenceGenerator	Passive/Connected	Generates primary keys.
Sorter	Active/Connected	Sorts data based on a sort key.
SourceQualifier	Active/Connected	Represents the rows that the Integration Service reads from a relational or flat file source when it runs a session.
SQL	ActiveorPassive/Connected	Executes SQL queries against a database.
StoredProcedure	Passive/ConnectedorUnconnected	Calls a stored procedure.
TransactionControl	Active/Connected	Defines commit and rollback transactions.
Union	Active/Connected	Merges data from different databases or flat file systems.
UnstructuredData	ActiveorPassive/Connected	Transforms data in unstructured and semi-structured formats.
UpdateStrategy	Active/Connected	Determines whether to insert, delete, update, or reject rows.
XMLGenerator	Active/Connected	Reads data from one or more input ports and outputs XML through a single output port.
XMLParser	Active/Connected	Reads XML from one input port and outputs data to one or more output ports.
XMLSourceQualifier	Active/Connected	Represents the rows that the Integration Service reads from an XML source when it runs a session.

- **Filter Transformation**
- Is an active transformation as it changes the no of records.
- Using the filter transformation, we can filter the records based on the filter condition. Filter transformation is an active transformation as it changes the no of records.
- For example, for loading the employee records having deptno equal to 10 only, we can put filter transformation in the mapping with the filter condition deptno=10. So only those records which have deptno =10 will be passed by filter transformation, rest other records will be dropped.
- **Source Qualifier Transformation**
- Is an active, connected transformation which is used to represent the rows that the integrations service read. Whenever we add a relational source or a flat file to a mapping, a source qualifier transformation is required. When we add a source to a mapping, source qualifier transformation is added automatically. With source qualifier, we can define and override how the data is fetched from the source.
- **Properties of Source Qualifier**
- You can use various properties of Source Qualifier, to determine what type of source data needs to transform to target table.
- **Source Filter** – Using the source filter property you can filter the number of source records. For example, you want to fetch only the employees of deptno 10, then you can enter the filter condition deptno=10 in source filter property and execute the data.
- **Number for sorted ports** – In source qualifier transformation, you can also sort the input records based on the ports number. So when the data is passed on to the transformations inside the mapping, it will read the port number and sort the data accordingly.
- As data can be sorted based on a single or multiple ports, you have to give the number of ports which will be used in sorting. If you give value as 1, then only empno data will be sorted. If you give value as 2 then on empno and ename on both columns data will be sorted.
- **Select Distinct** – you can fetch only distinct records from the source using this property. When you select the select distinct option, only distinct combination of source data will be fetched by source qualifier.

- **Aggregator transformation**
- Is an active transformation is used to performs aggregate calculations like sum, average, etc.
- For example, if you want to calculate the sum of salaries of all employees department wise, we can use the Aggregator Transformation.
- The aggregate operations are performed over a group of rows, so a temporary placeholder is required to store all these records and perform the calculations.
- For this, aggregator cache memory is used. This is a temporary main memory which is allocated to the aggregator transformation to perform such operations.
- **Router transformation**
- Is an active and connected transformation which is similar to filter transformation, used to filter the source data.
- The additional functionality provided beside filtering is that the discarded data (filtered out data) can also be collected in the mapping, as well as the multiple filter conditions can be applied to get multiple sets of data.
- For example, when filtering the data form deptno =10, we can also get those records where deptno is not equal to 10. So, router transformation gives multiple output groups, and each output group can have its own filter condition.

- In addition there is also a default group, this default group has those record sets which doesn't satisfy any of the group conditions. For example, if you have created two groups for the filter conditions deptno=10 & dept=20 respectively, then those records which are not having deptno 10 and 20 will be passed into this default group. In short the data which is rejected by the filter groups will be collected by this default group and sometimes there can be a requirement to store these rejected data. In such scenarios, default output group can be useful.
- To allow multiple filter condition, the router transformation provides group option.
- There is a default input group which takes input data
- There is also a default output group which provides all those data which is not passed by any filter condition
- For every filter condition, an output group is created in router transformation. You can connect different targets to these different groups.
- **Joiner transformation**
- Is an active and connected transformation that provides you the option to create joins in Informatica. The joins created using joiner transformation are similar to the joins in databases. The advantage of joiner transformation is that joins can be created for heterogeneous systems (different databases)
- In joiner transformation, there are two sources which we are going to use it for joins. These two sources are called
- Master Source
- Detail Source

- In the properties of joiner transformation, you can select which data source can be Master and which source can be detail source.
- During execution, the master source is cached into the memory for joining purpose. So it is recommended to select the source with less number of records as the master source.
- The following joins can be created using joiner transformation
- **Master outer join**
- In Master outer join, all records from the Detail source are returned by the join and only matching rows from the master source are returned.
- **Detail outer join**
- In detail outer join only matching rows are returned from the detail source, and all rows from the master source are returned.
- **Full outer join**
- In full outer join, all records from both the sources are returned. Master outer and Detail outer joins are equivalent to left outer joins in SQL.
- **Normal join**
- In normal join only matching rows are returned from both the sources.

- **Rank transformation**
- Is an active and connected transformation that performs the filtering of data based on group and ranks. For example, you want to get ten records of employees having highest salary, such kind of filtering can be done by rank transformation
- Rank transformation also provides the feature to do ranking based on groups. Like if you want to get top ten salaried employee department wise, then this grouping can be done with this transformation.
- Rank transformation is an active transformation, as it affects the number of output rows.
- The rank transformation has an output port by which it assigns a rank to the rows.
- **Sequence generator transformation**
- is passive so it does not affect the number of input rows. The sequence generator is used to generate primary key values & it's used to generate numeric sequence values like 1, 2, 3, 4, 5 etc.
- For example, you want to assign sequence values to the source records, then you can use sequence generator. The generated sequence values can be like 5, 10, 15, 20, 25 etc. or 10, 20, 30, 40, 50 etc. depending upon the configured properties of the transformation.
- Sequence generator has two output ports
- CURRVAL
- NEXTVAL

- CURRVAL port value is always NEXTVAL+1.
- To generate the sequence numbers, we always use the NEXTVAL column.
- Properties of Sequence Generator Transformation
- Start Value – It is the first value that will be generated by the transformation, the default value is 0.
- Increment by – This is the number by which you want to increment the values. The default value is 1.
- End value – It is the maximum value that the transformation should generate.
- Cycle – if this option is set then after reaching the end of the value, the transformation restarts from the start value.
- **Transaction Control Transformation**
- is an active and connected transformation which allows us to commit or rollback transactions during the execution of the mapping. Commit and rollback operations are of significant importance as it guarantees the availability of data.
- When processing a high volume of data, there can be a situation when to commit the data to the target. If a commit is performed too frequently, then it will be an overhead to the system. If a commit is performed too late then in the case of failure there are chances of data loss.
- So to provide flexibility Transaction control transformation is provided. There are five in-built variables available in this transformation to handle the operation.

- There are five in-built variables available in this transformation to handle the operation.
- **TC_CONTINUE_TRANSACTION** there are no operations performed, the process of data load continues as it is
- **TC_COMMIT_BEFORE** when this flag is found set, a commit is performed before the processing of current row.
- **TC_COMMIT_AFTER** the current row is processed then a commit is performed.
- **TC_ROLLBACK_BEFORE** rollback is performed first then data is processed to write
- **TC_ROLLBACK_AFTER** data is processed then the rollback is performed.
- **Lookup transformation**
- Is a passive transformation used to look up a source, source qualifier, or target to get the relevant data. Basically, it's a kind of join operation in which one of the joining tables is the source data, and the other joining table is the lookup table.
- **Reusable transformation**
- A normal transformation is an object that belongs to a mapping and can be used inside that mapping only. However, by making a transformation reusable it can be re-used inside several mappings.
- For example, a lookup transformation which fetches employee details based on employee number can be used at multiple mappings wherever employee details are required. By using reusable transformation, it reduces the overwork of creating same functionality again.

- **Normalizer transformation**
- Is an active transformation, used to convert a single row into multiple rows and vice versa. It is a smart way of representing your data in more organized manner. If in a single row there is repeating data in multiple columns, then it can be split into multiple rows. Sometimes we have data in multiple occurring columns. For example

Student Name	Class 9 Score	Class 10 Score	Class 11 Score	Class 12 Score
Student 1	50	60	65	80
Student 2	70	64	83	77

- In this case, the class score column is repeating in four columns. Using normalizer, we can split these in the following data set.

Student Name	Class	Score
Student 1	9	50
Student 1	10	60
Student 1	11	65
Student 1	12	80
Student 2	9	70
Student 2	10	64
Student 2	11	83
Student 2	12	77

What is Workflow?

- Workflow is a group of instructions/commands to the integrations service in Informatica. The integration service is an entity which reads workflow information from the repository, fetches data from sources and after performing transformation loads it into the target.
- Workflow - It defines how to run tasks like **session task, command task, email task**, etc.
- To create a workflow you first need to create tasks and then add those tasks to the workflow.
- A Workflow is like an empty container, which has the capacity to store an object you want to execute. You add tasks to the workflow that you want to execute. In this tutorial, we are going to do following things in workflow.

- Workflow execution can be done in two ways
- **Sequence** : Tasks execute in the order in which they are defined
- **Event based :** Tasks gets executed based on the event conditions.

Dimension Modelling

- A denormalized relational model
  - Made up of tables with attributes
  - Relationships defined by keys and foreign keys
- Organized for understandability and ease of reporting rather than update
- Queried and maintained by SQL or special purpose management tools.

Entity-Relationship vs. Dimensional Models

- One table per entity
- Minimize data redundancy
- Optimize update
- The Transaction Processing Model

- One fact table for data organization
- Maximize understandability
- Optimized for retrieval
- The data warehousing model

Fact Tables

- Contains two or more foreign keys
- Tend to have huge numbers of records
- Useful facts tend to be numeric and additive

Dimension Tables

- Contain text and descriptive information
- 1 in a 1-M relationship
- Generally the source of interesting constraints
- Typically contain the attributes for the SQL answer set.

Strengths of the Dimensional Model

- Predictable, standard framework
- Respond well to changes in user reporting needs
- Relatively easy to add data without reloading tables
- Standard design approaches have been developed
- There exist a number of products supporting the dimensional model

The Business Model

Identify the data structure, attributes and constraints for the client's data warehousing environment.

- Stable
- Optimized for update
- Flexible

As always in life, there are some disadvantages to 3NF:

- Performance can be truly awful. Most of the work that is performed on denormalizing a data model is an attempt to reach performance objectives.
- The structure can be overwhelmingly complex. We may wind up creating many small relations which the user might think of as a single relation or group of data.

The 4 Step Design Process

- Choose the Data Mart
- Declare the Grain
- Choose the Dimensions
- Choose the Facts

Building a Data Warehouse from a Normalized Database

The steps

- Develop a normalized entity-relationship business model of the data warehouse.
- Translate this into a dimensional model. This step reflects the information and analytical characteristics of the data warehouse.
- Translate this into the physical model. This reflects the changes necessary to reach the stated performance objectives.

Structural Dimensions

- The first step is the development of the structural dimensions. This step corresponds very closely to what we normally do in a relational database.
- The star architecture that we will develop here depends upon taking the central intersection entities as the fact tables and building the foreign key => primary key relations as dimensions.

Steps in dimensional modeling

- Select an associative entity for a fact table
- Determine granularity
- Replace operational keys with surrogate keys
- Promote the keys from all hierarchies to the fact table
- Add date dimension
- Split all compound attributes
- Add necessary categorical dimensions
- Fact (varies with time) / Attribute (constant)

Converting an E-R Diagram

- Determine the purpose of the mart
- Identify an association table as the central fact table
- Determine facts to be included
- Replace all keys with surrogate keys
- Promote foreign keys in related tables to the fact table
- Add time dimension
- Refine the dimension tables

Choosing the Mart

- A set of related fact and dimension tables
- Single source or multiple source
- Conformed dimensions
- Typically have a fact table for each process

Fact Tables

- Represent a process or reporting environment that is of value to the organization
- It is important to determine the identity of the fact table and specify exactly what it represents.
- Typically correspond to an associative entity in the E-R model

Grain (unit of analysis)

The grain determines what each fact record represents: the level of detail.

- For example
  - Individual transactions
  - Snapshots (points in time)
  - Line items on a document
- Generally better to focus on the smallest grain

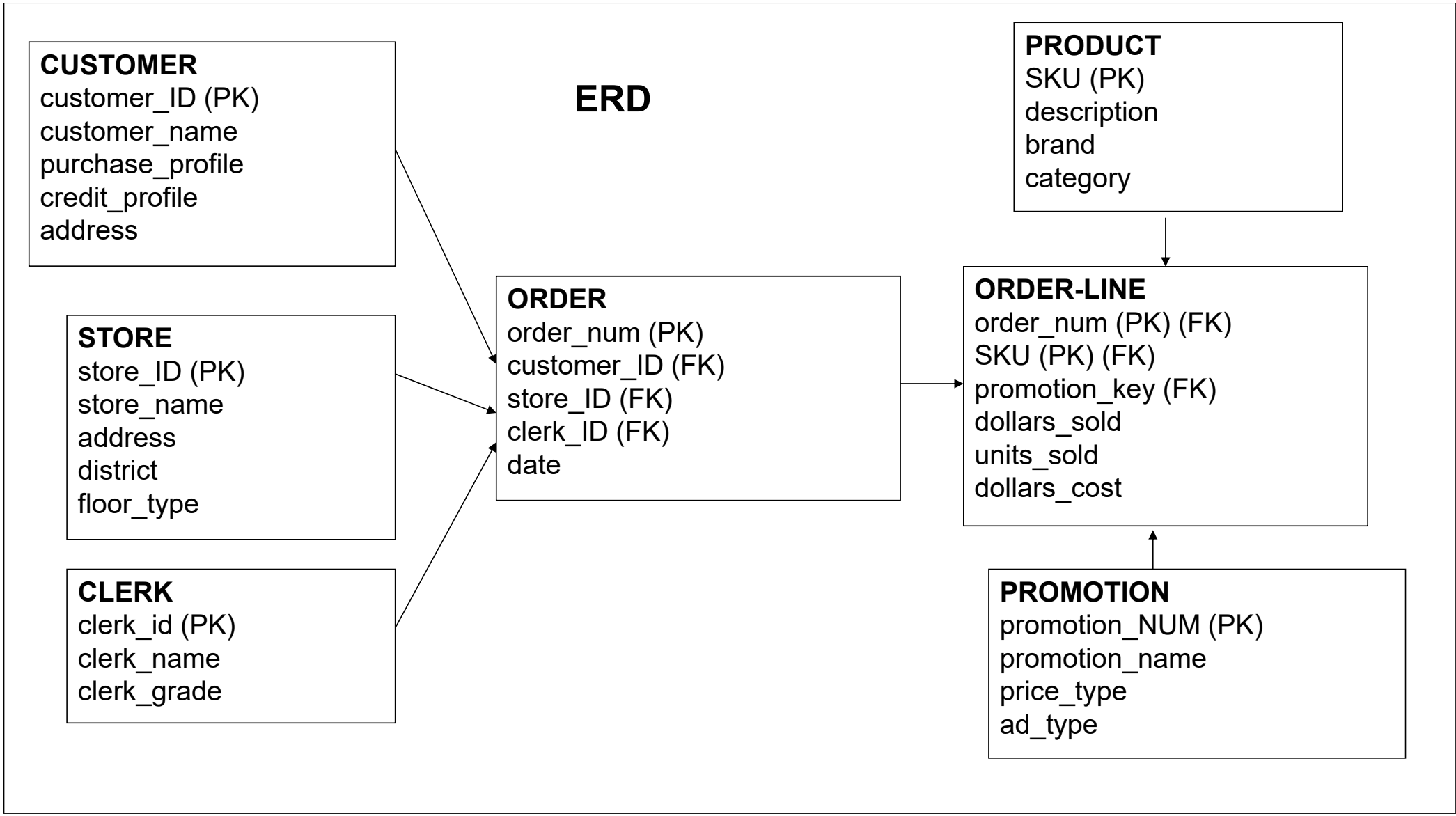
Facts

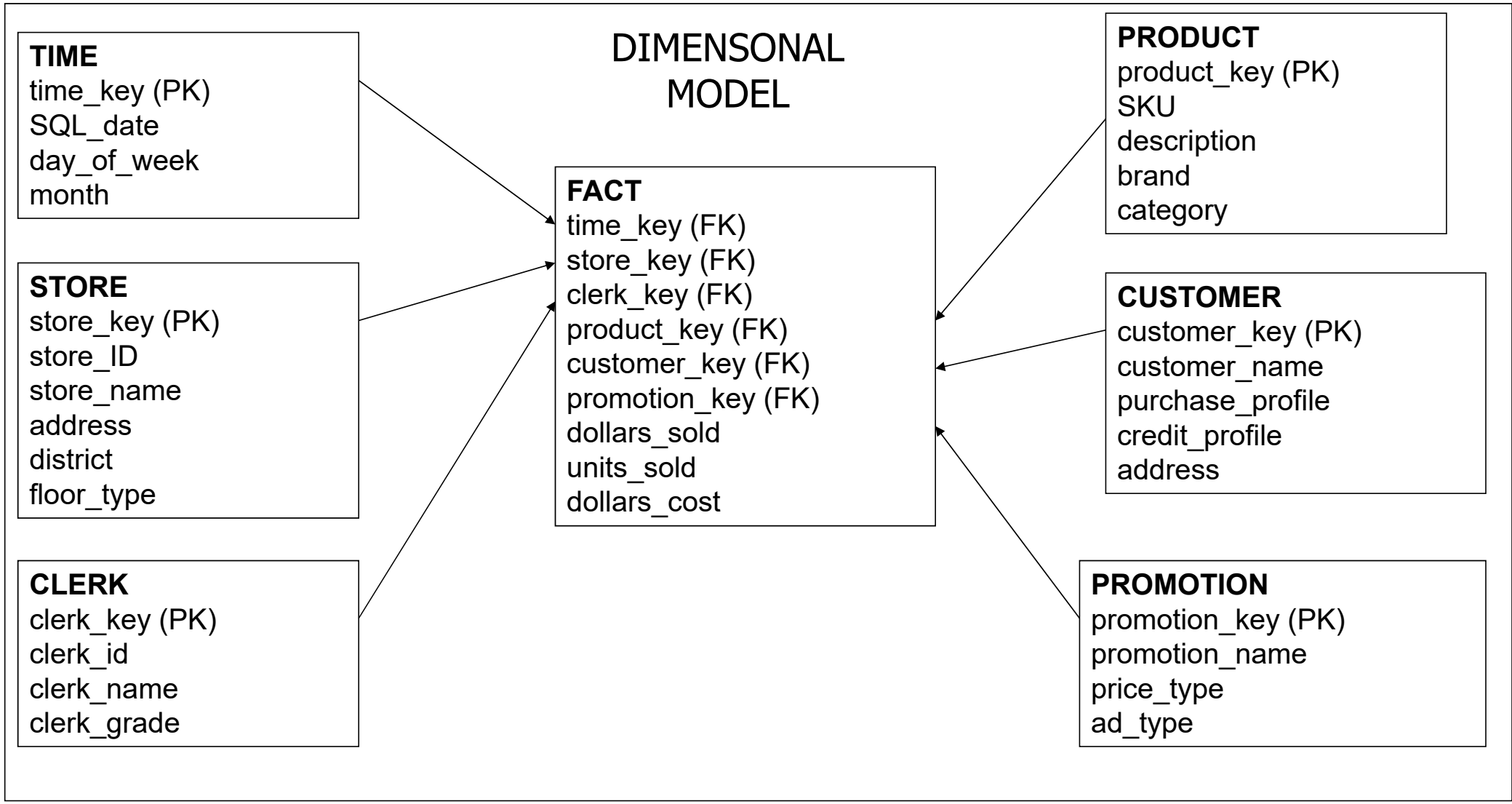
- Measurements associated with fact table records at fact table granularity
- Normally numeric and additive
- Non-key attributes in the fact table

  Attributes in dimension tables are constants. Facts vary with the granularity of the fact table

Dimensions

- A table (or hierarchy of tables) connected with the fact table with keys and foreign keys
- Preferably single valued for each fact record (1:m)
- Connected with surrogate (generated) keys, not operational keys
- Dimension tables contain text or numeric attributes


ERD
CUSTOMER
customer_ID (PK)
customer_name
purchase_profile
credit_profile
address
STORE
store_ID (PK)
store_name
address
district
floor_type
CLERK
clerk_id (PK)
clerk_name
clerk_grade
ORDER
order_num (PK)
customer_ID (FK)
store_ID (FK)
clerk_ID (FK)
date
PRODUCT
SKU (PK)
description
brand
category
ORDER-LINE
order_num (PK) (FK)
SKU (PK) (FK)
promotion_key (FK)
dollars_sold
units_sold
dollars_cost
PROMOTION
promotion_NUM (PK)
promotion_name
price_type
ad_type


DIMENSONAL MODEL
TIME
time_key (PK)
SQL_date
day_of_week
month
STORE
store_key (PK)
store_ID
store_name
address
district
floor_type
CLERK
clerk_key (PK)
clerk_id
clerk_name
clerk_grade
FACT
time_key (FK)
store_key (FK)
clerk_key (FK)
product_key (FK)
customer_key (FK)
promotion_key (FK)
dollars_sold
units_sold
dollars_cost
PRODUCT
product_key (PK)
SKU
description
brand
category
CUSTOMER
customer_key (PK)
customer_name
purchase_profile
credit_profile
address
PROMOTION
promotion_key (PK)
promotion_name
price_type
ad_type

Comparison chart

Edit	Snowflake Schema	Star Schema
Ease of maintenance / change	No redundancy, so snowflake schemas are easier to maintain and change.	Has redundant data and hence less easy to maintain/change
Ease of Use	More complex queries and hence less easy to understand	Lower query complexity and easy to understand
Query Performance	More foreign keys and hence longer query execution time (slower)	Less number of foreign keys and hence shorter query execution time (faster)
Type of Datawarehouse	Good to use for datawarehouse core to simplify complex relationships (many:many)	Good for datamarts with simple relationships (1:1 or 1:many)
Joins	Higher number of Joins	Fewer Joins
Dimension table	A snowflake schema may have more than one dimension table for each dimension.	A star schema contains only single dimension table for each dimension.
When to use	When dimension table is relatively big in size, snowflaking is better as it reduces space.	When dimension table contains less number of rows, we can choose Star schema.
Normalization/ De-Normalization	Dimension Tables are in Normalized form but Fact Table is in De-Normalized form	Both Dimension and Fact Tables are in De-Normalized form
Data model	Bottom up approach	Top down approach