|  |
| --- |
| EPAM Systems, RD Dep., RD Dep. |
| MTN.BI.03 SQL for Analysis  **Introduction to OLAP** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1.0 | Initial Version | Aliaksei Belablotski | 12-Feb-2012 |  |  |
|  |  |  |  |  |  |

Contents

[1. Data Warehouse 3](#_Toc316860226)

[1.1. Data Warehouse Definition 3](#_Toc316860227)

[1.2. Data Warehouse Goals 3](#_Toc316860228)

[1.3. Data Warehouse Benefits 4](#_Toc316860229)

[1.4. Dimensional Approach 4](#_Toc316860230)

[1.5. Normalized Approach 5](#_Toc316860231)

[2. Oracle SQL Developer 5](#_Toc316860232)

[2.1. Download and Install 5](#_Toc316860233)

[2.2. Establish Connection 5](#_Toc316860234)

[2.3. Oracle Data Modeler 6](#_Toc316860235)

[2.4. Oracle SQL Developer Reports 8](#_Toc316860236)

# Data Warehouse

## Data Warehouse Definition

A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process.

**Subject-Oriented**: A data warehouse can be used to analyze a particular subject area. For example, "sales" can be a particular subject.

**Integrated**: A data warehouse integrates data from multiple data sources. For example, source A and source B may have different ways of identifying a product, but in a data warehouse, there will be only a single way of identifying a product.

**Time-Variant**: Historical data is kept in a data warehouse. For example, one can retrieve data from 3 months, 6 months, 12 months, or even older data from a data warehouse. This contrasts with a transactions system, where often only the most recent data is kept. For example, a transaction system may hold the most recent address of a customer, where a data warehouse can hold all addresses associated with a customer.

**Non-volatile**: Once data is in the data warehouse, it will not change. So, historical data in a data warehouse should never be altered.

## Data Warehouse Goals

**The data warehouse must make an organization’s information easily accessible.**

The contents of the data warehouse must be understandable. The data must be intuitive and obvious to the business user, not merely the developer. Understandability implies legibility; the contents of the data warehouse need to be labeled meaningfully. Business users want to separate and combine the data in the warehouse in endless combinations, a process commonly referred to as slicing and dicing. The tools that access the data warehouse must be simple and easy to use. They also must return query results to the user with minimal wait times.

**The data warehouse must present the organization’s information consistently.**

The data in the warehouse must be credible. Data must be carefully assembled from a variety of sources around the organization, cleansed, quality assured, and released only when it is fit for user consumption. Information from one business process should match with information from another. If two performance measures have the same name, then they must mean the same thing. Conversely, if two measures don’t mean the same thing, then they should be labeled differently. Consistent information means high-quality information. It means that all the data is accounted for and complete. Consistency also implies that common definitions for the contents of the data warehouse are available for users.

**The data warehouse must be adaptive and resilient to change.**

We simply can’t avoid change. User needs, business conditions, data, and technology are all subject to the shifting sands of time. The data warehouse must be designed to handle this inevitable change. Changes to the data warehouse should be graceful, meaning that they don’t invalidate existing data or applications. The existing data and applications should not be changed or disrupted when the business community asks new questions or new data is added to the warehouse. If descriptive data in the warehouse is modified, we must account for the changes appropriately.

**The data warehouse must be a secure bastion that protects our information assets.**

An organization’s informational crown jewels are stored in the data warehouse. At a minimum, the warehouse likely contains information about what we’re selling to whom at what price—potentially harmful details in the hands of the wrong people. The data warehouse must effectively control access to the organization’s confidential information.

**The data warehouse must serve as the foundation for improved decision making.**

The data warehouse must have the right data in it to support decision making. There is only one true output from a data warehouse: the decisions that are made after the data warehouse has presented its evidence. These decisions deliver the business impact and value attributable to the warehouse. The original label that predates the data warehouse is still the best description of what we are designing: a decision support system.

**The business community must accept the data warehouse if it is to be deemed successful.**

It doesn’t matter that we’ve built an elegant solution using best-of-breed products and platforms. If the business community has not embraced the data warehouse and continued to use it actively six months after training, then we have failed the acceptance test. Unlike an operational system rewrite, where business users have no choice but to use the new system, data warehouse usage is sometimes optional. Business user acceptance has more to do with simplicity than anything else.

## Data Warehouse Benefits

**Benefits of a data warehouse:**

* A data warehouse maintains a copy of information from the source transaction systems.
* This architectural complexity provides the opportunity to:
  + Maintain data history, even if the source transaction systems do not.
  + Integrate data from multiple source systems, enabling a central view across the enterprise. This benefit is always valuable, but particularly so when the organization has grown by merger.
  + Improve data, by providing consistent codes and descriptions, flagging or even fixing bad data.
  + Present the organization's information consistently.
  + Provide a single common data model for all data of interest regardless of the data's source.
  + Restructure the data so that it makes sense to the business users.
  + Restructure the data so that it delivers excellent query performance, even for complex analytic queries, without impacting the operational systems.
  + Add value to operational business applications, notably customer relationship management (CRM) systems.

## Dimensional Approach

**Normalized versus Dimensional approach for storage of data.**

There are two leading approaches to storing data in a data warehouse — the dimensional approach and the normalized approach.

The dimensional approach, whose supporters are referred to as “Kimballites”, believe in Ralph Kimball’s approach in which it is stated that the data warehouse should be modeled using a Dimensional Model/Star schema. The normalized approach, also called the 3NF model, whose supporters are referred to as “Inmonites”, believe in Bill Inmon's approach in which it is stated that the data warehouse should be modeled using an E-R ModelNormalized model.

In a dimensional approach, transaction data are partitioned into either "facts", which are generally numeric transaction data, or "dimensions", which are the reference information that gives context to the facts. For example, a sales transaction can be broken up into facts such as the number of products ordered and the price paid for the products, and into dimensions such as order date, customer name, product number, order ship-to and bill-to locations, and salesperson responsible for receiving the order.

A **key advantage of a dimensional approach** is that the data warehouse is easier for the user to understand and to use. Also, the retrieval of data from the data warehouse tends to operate very quickly.

Dimensional structures are easy to understand for business users, because the structure is divided into measurements/facts and context/dimensions. Facts are related to the organization’s business processes and operational system whereas the dimensions surrounding them contain context about the measurement (Kimball, Ralph 2008).

**The main disadvantages of the dimensional approach are:**

In order to maintain the integrity of facts and dimensions, loading the data warehouse with data from different operational systems is complicated, and it is difficult to modify the data warehouse structure if the organization adopting the dimensional approach changes the way in which it does business.

## Normalized Approach

In the normalized approach, the data in the data warehouse are stored following, to a degree, database normalization rules. Tables are grouped together by subject areas that reflect general data categories (e.g., data on customers, products, finance, etc.). The normalized structure divides data into entities, which creates several tables in a relational database. When applied in large enterprises the result is dozens of tables that are linked together by a web of joins. Furthermore, each of the created entities is converted into separate physical tables when the database is implemented (Kimball, Ralph 2008).

The **main advantage of this approach** is that it is straightforward to add information into the database.

A **disadvantage of this approach** is that, because of the number of tables involved, it can be difficult for users both to: join data from different sources into meaningful information and then access the information without a precise understanding of the sources of data and of the data structure of the data warehouse.

It should be noted that both normalized – and dimensional models can be represented in entity-relationship diagrams as both contain jointed relational tables. The difference between the two models is the degree of normalization.

These approaches are not mutually exclusive, and there are other approaches. Dimensional approaches can involve normalizing data to a degree.

# Oracle SQL Developer

## Download and Install

Download the latest version of Oracle SQL Developer from <http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html>

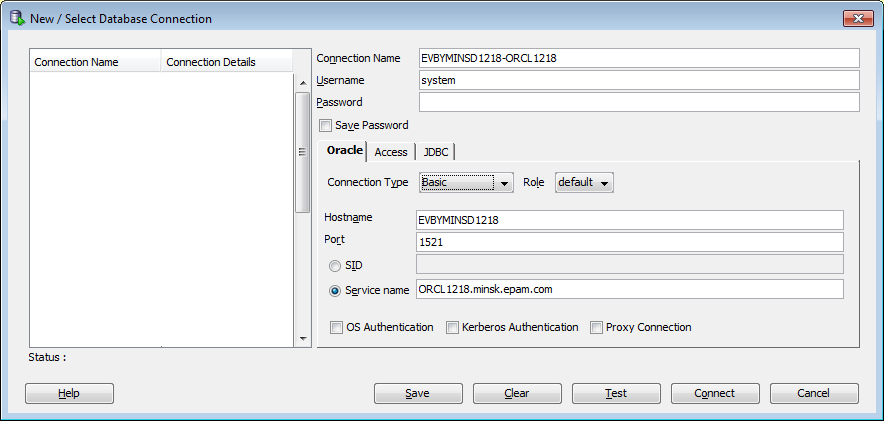
The easiest way is to get binary named “[Oracle SQL Developer for 32-bit Windows](http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html) (This zip file includes the JDK1.6.0\_11)”.

Oracle Learning Library resources:

* [Getting Started with Oracle SQL Developer 3.0](http://apex.oracle.com/pls/apex/f?p=44785:24:896077371487201::NO:24:P24_CONTENT_ID,P24_PREV_PAGE:5234,2)
* [Working with Data Grids using Oracle SQL Developer 3.0](http://apex.oracle.com/pls/apex/f?p=44785:24:896077371487201::NO:24:P24_CONTENT_ID,P24_PREV_PAGE:5249,2)
* [Using Query Builder in Oracle SQL Developer 3.0](http://apex.oracle.com/pls/apex/f?p=44785:24:896077371487201::NO:24:P24_CONTENT_ID,P24_PREV_PAGE:5235,2)

## Establish Connection

It’s strongly recommended to test connection before save.

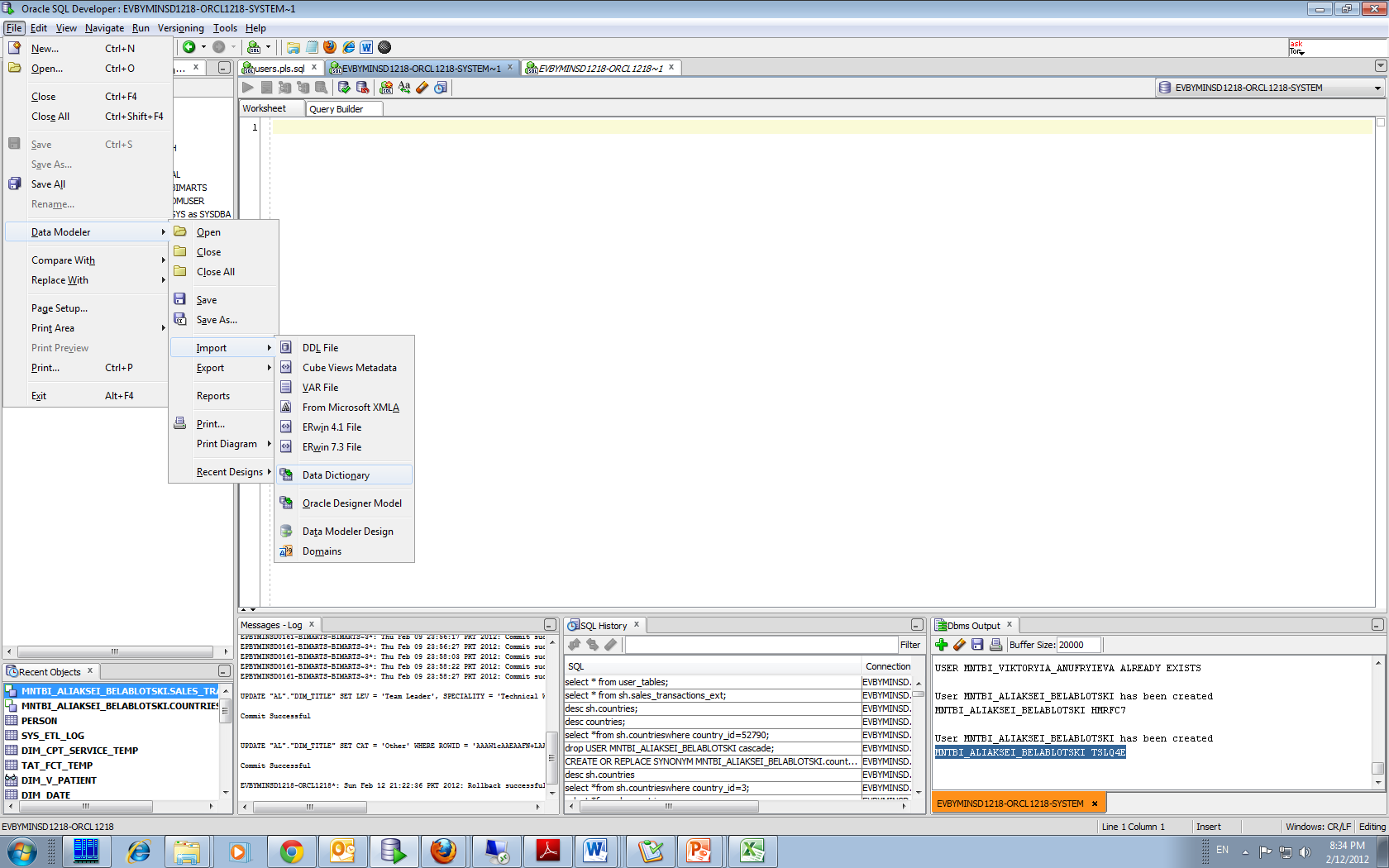


## Oracle Data Modeler

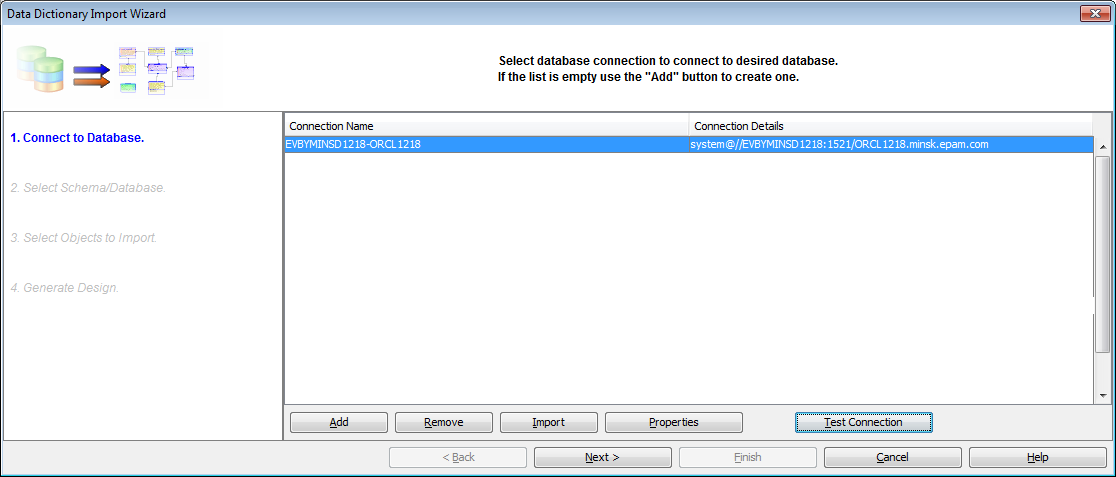
Oracle Learning Library resources:

* [SQL Developer Data Modeler 3.0 New Features Overview](http://apex.oracle.com/pls/apex/f?p=44785:24:896077371487201::NO:24:P24_CONTENT_ID,P24_PREV_PAGE:5221,2)
* [Importing Data Models](http://apex.oracle.com/pls/apex/f?p=44785:24:896077371487201::NO:24:P24_CONTENT_ID,P24_PREV_PAGE:5220,2)
* [Forward and Reverse Engineering Models and Working with Logical Model Diagrams, Displays and Subviews](http://apex.oracle.com/pls/apex/f?p=44785:24:896077371487201::NO:24:P24_CONTENT_ID,P24_PREV_PAGE:5231,2)
* [Generating a Multi-Dimensional Model with SQL Developer Data Modeler 3.0](http://apex.oracle.com/pls/apex/f?p=44785:24:896077371487201::NO:24:P24_CONTENT_ID,P24_PREV_PAGE:5229,2)
* [Adding a Table to An Existing Database for SQL Developer Data Modeler 3.0](http://apex.oracle.com/pls/apex/f?p=44785:24:896077371487201::NO:24:P24_CONTENT_ID,P24_PREV_PAGE:5228,2)

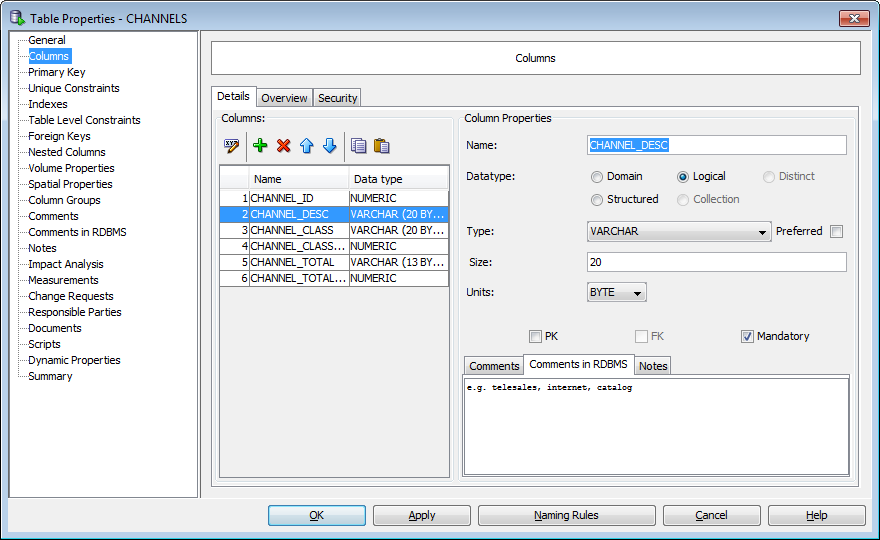
Start Data Modeler with importing metadata from existing schema (import metadata from Sales History).



Use an existing connection to Sales History.



Explore database objects and relationships between them. Notice “Comments in RDBMS” tab in Properties window (these comments are very helpful for understanding object purpose).



## Oracle SQL Developer Reports

Oracle Learning Library resources:

* [Overview of Reports](http://apex.oracle.com/pls/apex/f?p=44785:24:0::NO:24:P24_CONTENT_ID,P24_PREV_PAGE:6011,2)
* [How to Create a Report](http://apex.oracle.com/pls/apex/f?p=44785:24:0::NO:24:P24_CONTENT_ID,P24_PREV_PAGE:6012,2)
* [Adding a Child Report](http://apex.oracle.com/pls/apex/f?p=44785:24:0::NO:24:P24_CONTENT_ID,P24_PREV_PAGE:6013,2)
* [PDF Reports](http://apex.oracle.com/pls/apex/f?p=44785:24:0::NO:24:P24_CONTENT_ID,P24_PREV_PAGE:6014,2)
* [Export a PDF Report](http://apex.oracle.com/pls/apex/f?p=44785:24:0::NO:24:P24_CONTENT_ID,P24_PREV_PAGE:6015,2)

Read about User-defined reports in SQL Developer’s User Guide (p 63, 83).

