

UNIT 3.1



DATAWAREHOUSING

UNIT 3

CHAPTER 1



1.Designing the Target Structure: Data warehouse design, Dimensional

- design, Cube and dimensions, Implementation of a dimensional model in a database, Relational implementation (star schema), Multidimensional implementation (OLAP), Designing the ACME data warehouse, Identifying the dimensions, Designing the cube, Data warehouse design in OWB, Creating a target user and module, Create a target user, Create a target module, OWB design objects.

Data warehouse design

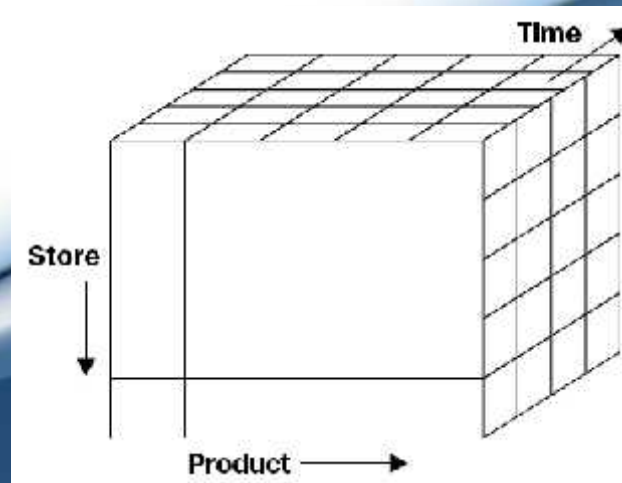
- design of a data warehouse is basically **dimensional model**.
- This is a way of looking at the data from a business perspective
- This makes the data simple, understandable, and easy to query for the business end user.
- dimensional models were introduced to provide the end user with a flattened structure of easily queried tables that he or she can understand from a business perspective

Dimensional design

- A dimensional model takes the business rules of our organization and represents them in the database in a more understandable way.
- EG:::"how many gizmos did I sell last month in all stores in the south and how does that compare to how many I sold in the same month last year?"
- A dimensional model removes the complexity and represents the data in a way that end users can relate to it more easily from a business perspective.

Cube and dimensions

- The **dimensions become the business characteristics about the sales, for example:**
 - A time dimension—users can look back in time and check various time periods
 - A store dimension—information can be retrieved by store and location
 - A product dimension—various products for sale can be broken out



Implementation of a dimensional model in a database

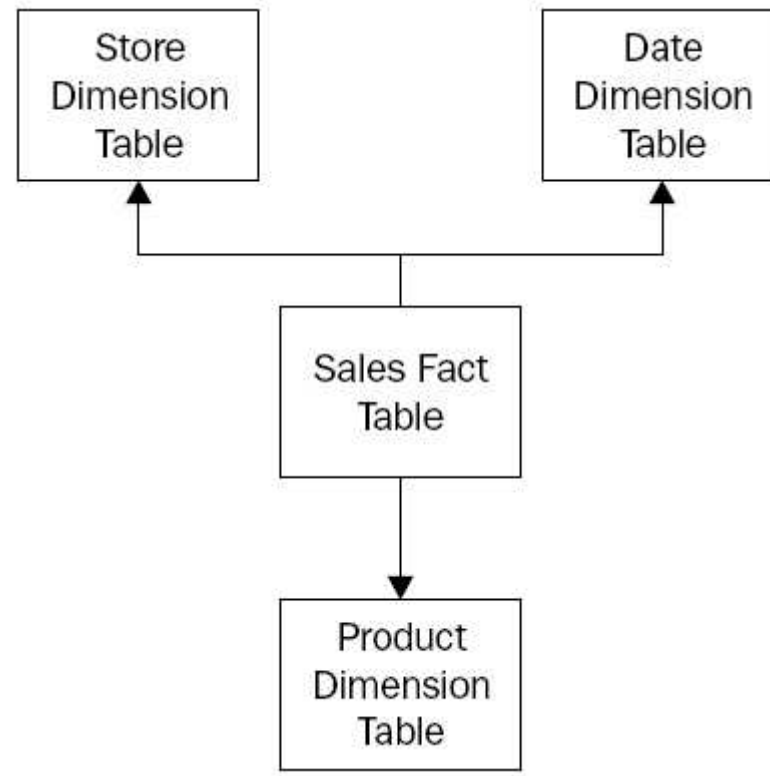
- implementation of the model to see how it gets physically represented in the database
- There are two options:
 - a **relational implementation**(most common) and
 - a **multidimensional implementation**(requires a special feature in a database that allows defining cubes).

Relational implementation (star schema)

- The diagrams presented showed all the tables interconnected, and we discussed the use of foreign keys in a table to refer to a row in another table.
- The ER diagram of such an implementation would be shaped somewhat like a star, and thus the term **star** schema is used to refer to this kind of an implementation.
- The main table in the middle is referred to as the **fact table** because it holds the facts, or measures that we are interested in about our organization.
- The tables surrounding the fact table are known as dimension tables.

For collating records it uses a concept of de-normalization

This is a variation of the star schema referred to as a **snowflake schema** because with this type of implementation, dimension tables are partially normalized to pull common data out into secondary dimension tables.



Multidimensional implementation (OLAP)

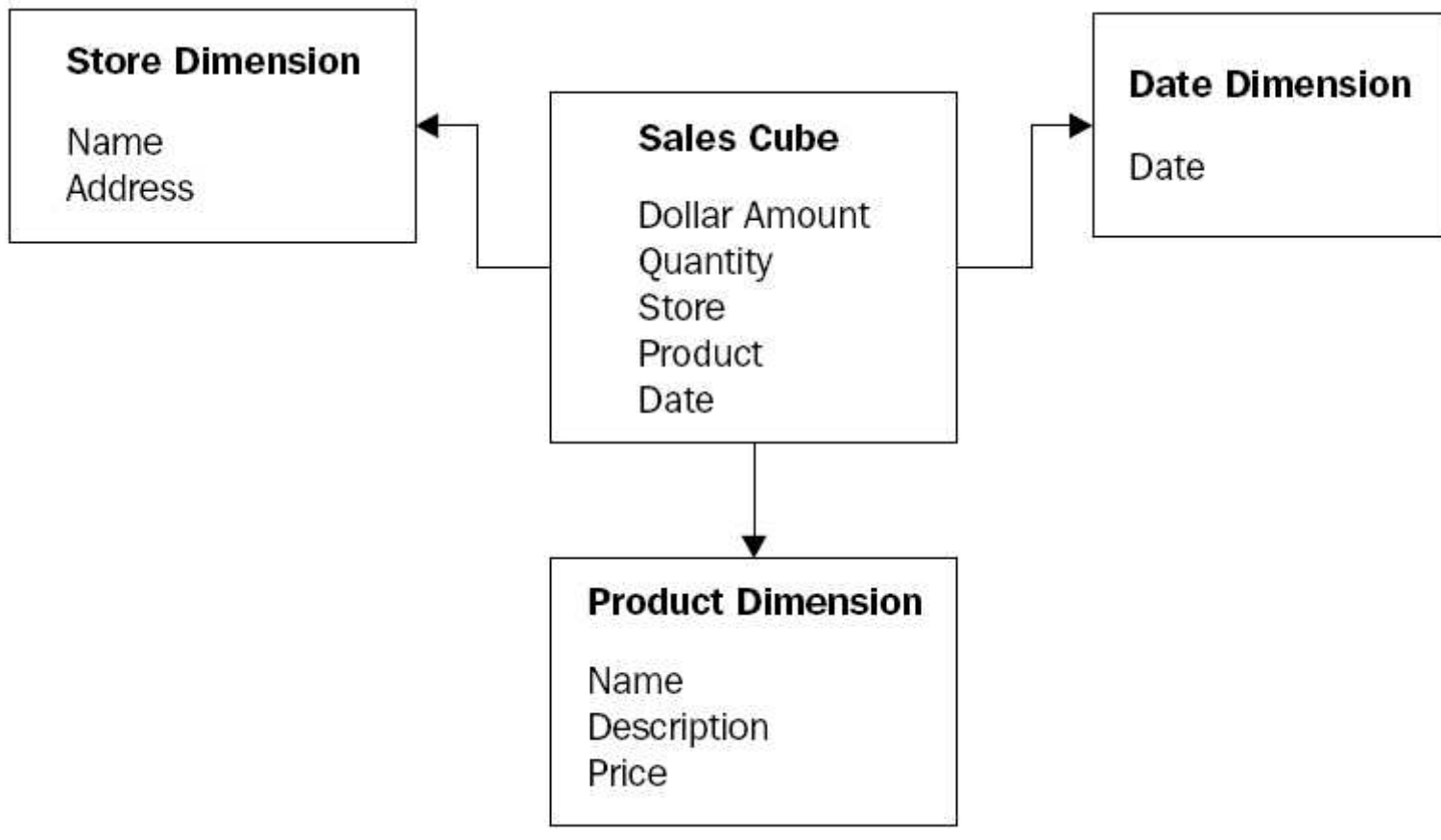
- **OLAP (online analytic or analytical processing)** requires a database with special features that allow it to store cubes as actual objects in the database
- It also provides advanced calculation and analytic content built into the database to facilitate advanced analytic querying

Designing the ACME data warehouse

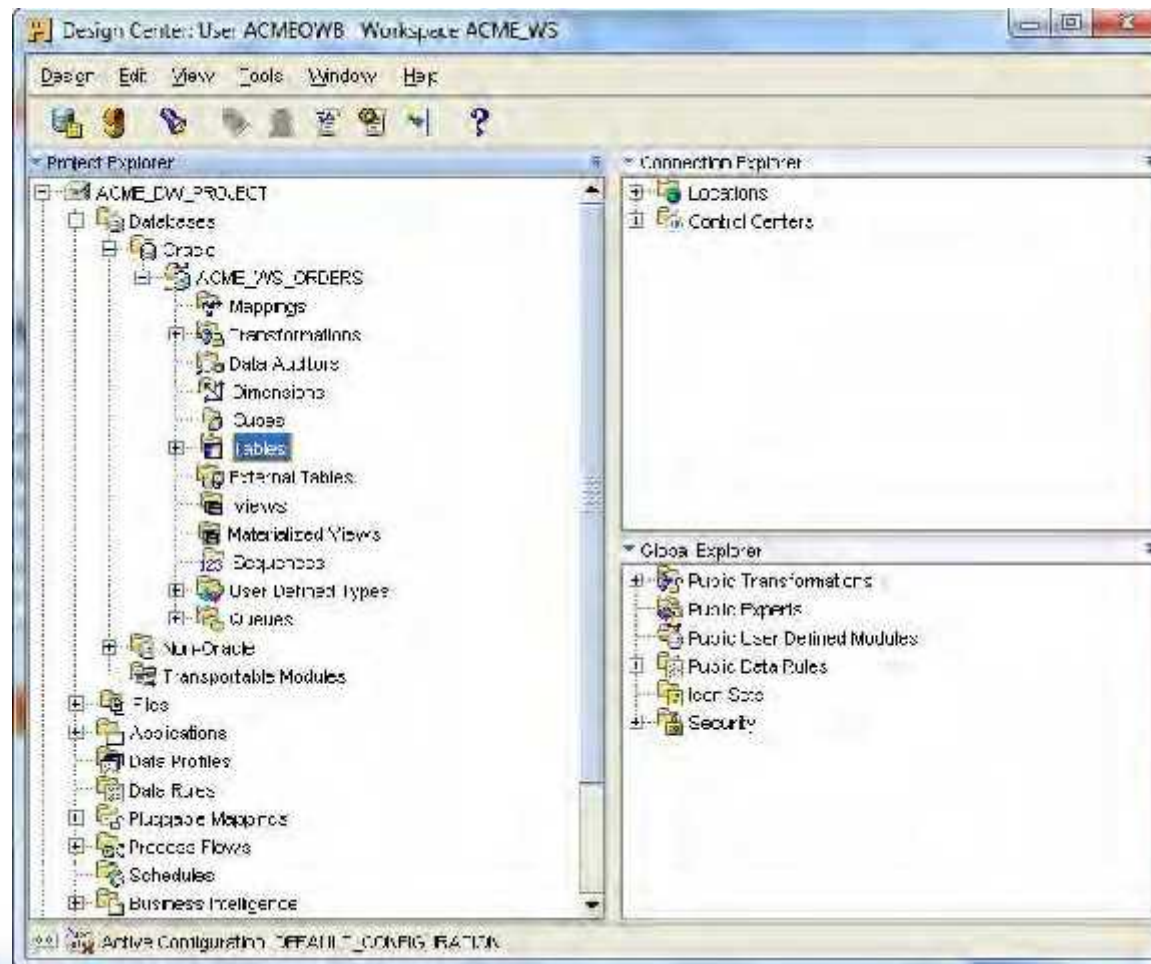
- **Identifying the dimensions**
 - know what business process we're going to be supporting with our data warehouse (management concerned with daily inventory or daily sales volume)
 - guaranteed that we will need a time/date type dimension for any data warehouse

Designing the cube

- In the case of the ACME Toys and Gizmos Company, we have seen that the main measure the management is concerned about is daily sales
- There are other numbers we could consider such as inventory numbers: How much of each item is on hand?
- A very important topic to consider at this point is what will be the **grain of the measure**—the sales data—that we're going to store in our cube? **The grain (or granularity) is the level that the sales number refers to.**



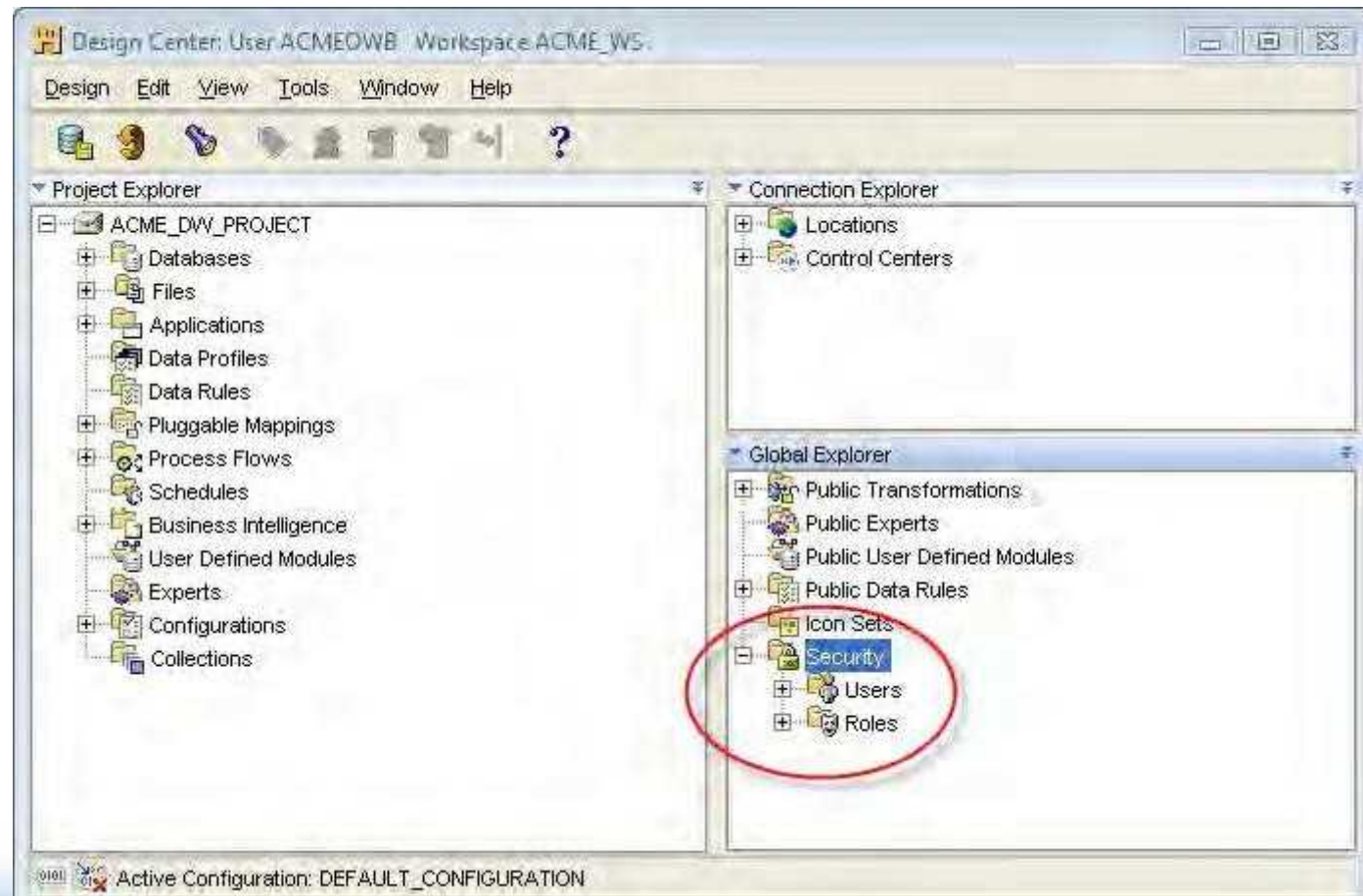
Data warehouse design in OWB



Creating a target user and module

- Every target module must be mapped to a target user schema.
- **acmeowb** user as the repository owner and mentioned that this user can be a deployment target for our data warehouse

- One of those object types is a Users object that exists under the Security node as shown here:



- Right-click on the Users node and select New... to launch the Create User dialog box as shown here:

click on the **Create DB User...** button to create a new database user



we'll see the new username if we expand the Users node under Security in the Global Explorer

Specify user name and password with DBA privilege:

DBA name: system

DBA password: *****

Provide information to create the new DB user:

Name: ACME_DWH

Password: *****

Confirm Password: *****

Table Space:

Default: USERS

Temporary: TEMP

Help OK Cancel

Create a target module

- Right-click on the Oracle object under Databases and select New... from the pop-up menu to launch the Create Module Wizard and step through the process. We'll name this module ACME_DWH for ACME Data Warehouse.

Edit Oracle Database Location: ACME_DWH_LOCATION1

Name: ACME_DWH_LOCATION

Description:

Type: HOST:PORT:SERVICE ▼

User Name: ACME_DWH

Password: *****

Host: localhost

Port: 1521

Service Name: ACMEDW2

Use Global Name:

Schema: ACME_DWH

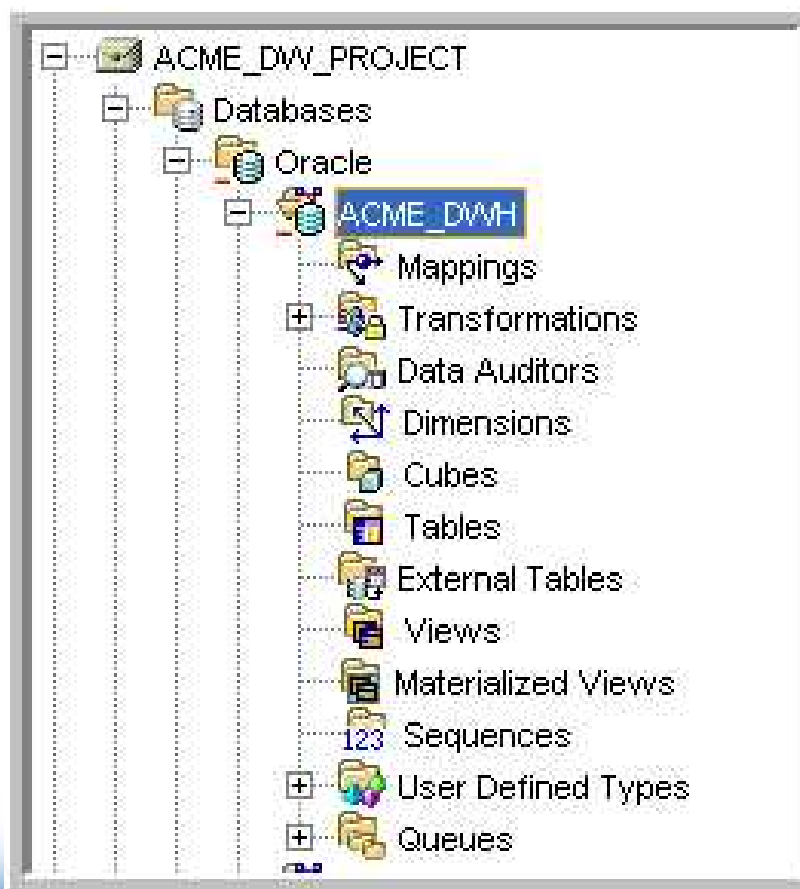
Version: 11.1 ▼

Test Results: Successfull

OWB design objects

- we can see a number of objects that are available to us

There are objects that are relational such as **Tables, Views, Materialized Views, and Sequences**. Also, there are dimensional objects such as **Cubes and Dimensions**.



DATAWAREHOUSING

UNIT 3

END OF CHAPTER 1

