

VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui



Dimension Hierarchies

Lecturer : Dr. Pavle Mogin

SWEN 432
*Advanced Database Design and
Implementation*

Plan for Dimension Hierarchies

- Motives
- Attributes and attribute hierarchies
- Declaring hierarchies for 1NF (or 2NF) Dimension tables
- Declaring hierarchies for 3NF (or BCNF) Dimension tables
- Influence of constraints on query rewrite

Motives - General

- General remarks regarding attribute hierarchies:
 - Dimensions are organized in hierarchies
 - Going up a hierarchy is called roll-up
 - Going down a hierarchy is called drill-down
 - The top level of a hierarchy is ALL
 - ALL years, ALL countries, ALL employees,...
 - Data analysis typically starts at a higher hierarchy level and gradually drills down if the situation warrants such analysis
 - Decision to drill down makes a data analyst

Motives – Query Rewrite

- Attribute hierarchies play a very important roll in rewriting a query with the aim to use a materialized view
- Typically a materialized view contains data on a lower level of granularity (finer granularity)
- Also, a query which potentially can use a view requires data on a higher level of granularity (coarser granularity)
- In the query rewrite process, data of a lower granularity level is used automatically by a query processor to produce data of a higher granularity level

Motives – Automated Query Rewrite

- A query processor can automatically use an attribute hierarchy in a query rewrite process only if it is explicitly informed about the existence of the hierarchy
- Also it is extremely important to guaranty that data at different hierarchy levels satisfy certain constraints
- Otherwise the results are unreliable
- ROLAP engines offer a syntax for declaring attribute hierarchies

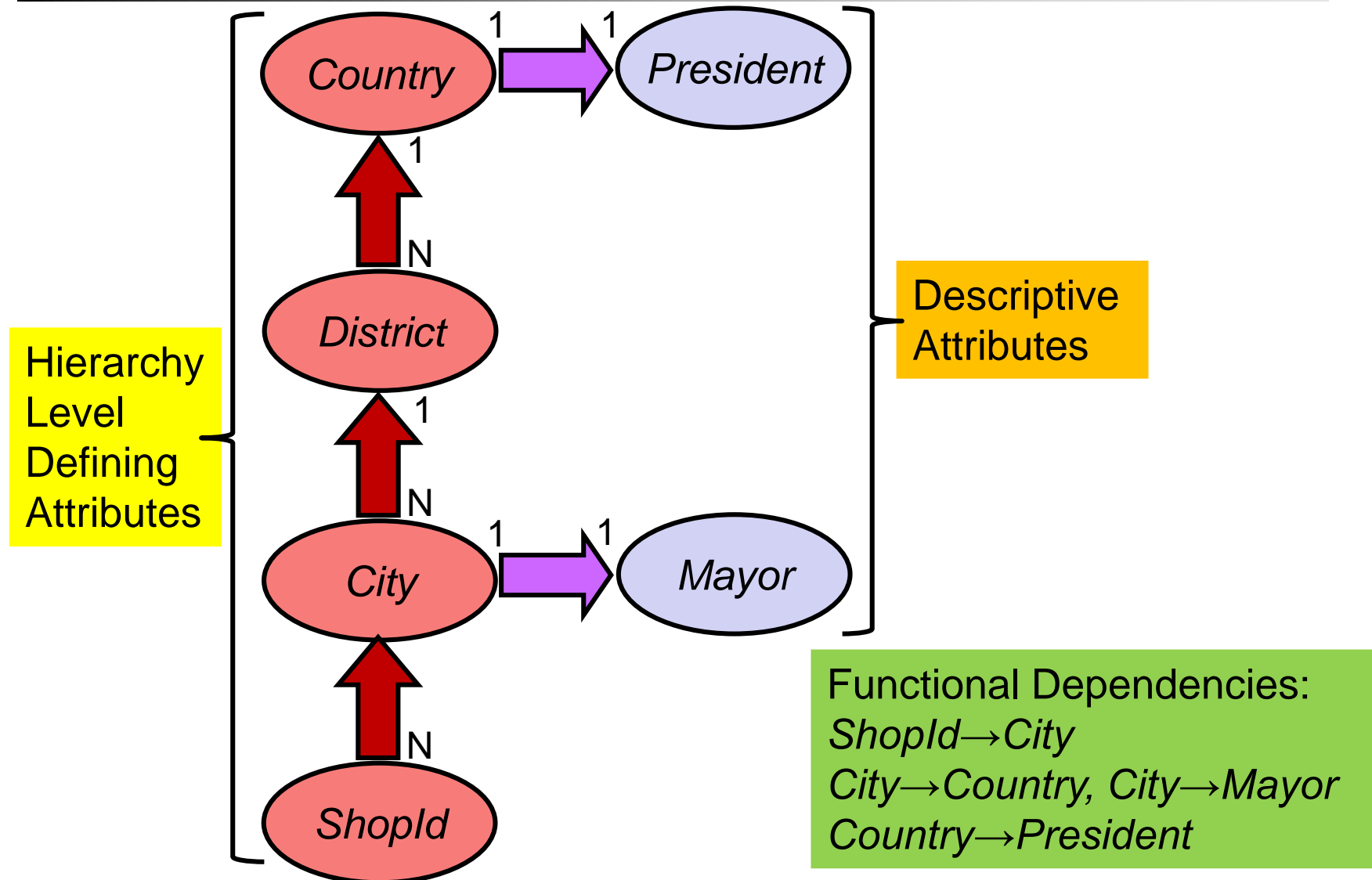
Statements for Declaring Hierarchies

- OLAP engines offer statements to declare:
 - Attributes of a dimension at different hierarchy levels,
 - Relationships between attributes at different levels, and
 - Relationships between attributes at the same level
- These statements are applied on attributes from already existing dimension tables
- These statements have two rolls:
 - To inform ROLAP engine about the existence of hierarchies, and
 - To enforce checking of constraints implied by these declarations

A 2NF Relation Schema

```
CREATE TABLE Location (  
  ShopId int PRIMARY KEY,  
  City varchar(20) NOT NULL,  
  District varchar(20) NOT NULL,  
  Country varchar(20) NOT NULL  
  Mayor varchar(20),  
  President varchar(20) );
```

Location Dimension



CREATE DIMENSION

- Syntax:

```
CREATE DIMENSION <dimension_name>  
LEVEL <column_name> IS  
<table_name.column_name>  
[...]
```

- Example:

```
CREATE DIMENSION location_dim  
LEVEL ShopId IS Location.ShopId  
LEVEL City IS Location.City  
LEVEL District IS Location.District  
LEVEL Country IS Location.Country
```

HIERARCHY (A Clause of Dimension)

- `HIERARCHY` clause defines a relationship of the cardinality `N:1` between levels of a `DIMENSION` statement
- If the attributes on the `N` side are declared to be `NOT NULL`, a `N:1` relationship defines a functional dependency
- There may be more than one `HIERARCHY` statement defined for the same `DIMENSION` statement
- This happens when a dimension table contains more than one hierarchy (multiple hierarchy)
 - e.g. Time

HIERARCHY Clause

- Syntax:

```
HIERARCHY <hierarchy_name> (  
  <column_name1> CHILD OF  
  <column_name2>  
  [...])
```

- Example:

```
HIERARCHY location_rollup (  
  ShopId CHILD OF  
  City CHILD OF  
  District CHILD OF  
  Country)
```

- Fd's: ShopId → City, City → District, District → Country

ATTRIBUTE... DETERMINES Clause

- `ATTRIBUTE...DETERMINES` clause is used to define a N:1 or a 1:1 relationship (and a functional dependency) between a column declared as a `LEVEL` column and another dimension table column that belongs to the same level of hierarchy and depends on the level column
- `ATTRIBUTE...DETERMINES` clause is the last clause of a `CREATE DIMENSION` statement

ATTRIBUTE...DETERMINES Clause

- Syntax:

```
ATTRIBUTE <column_name1> DETERMINES  
<column_name2> [...];
```

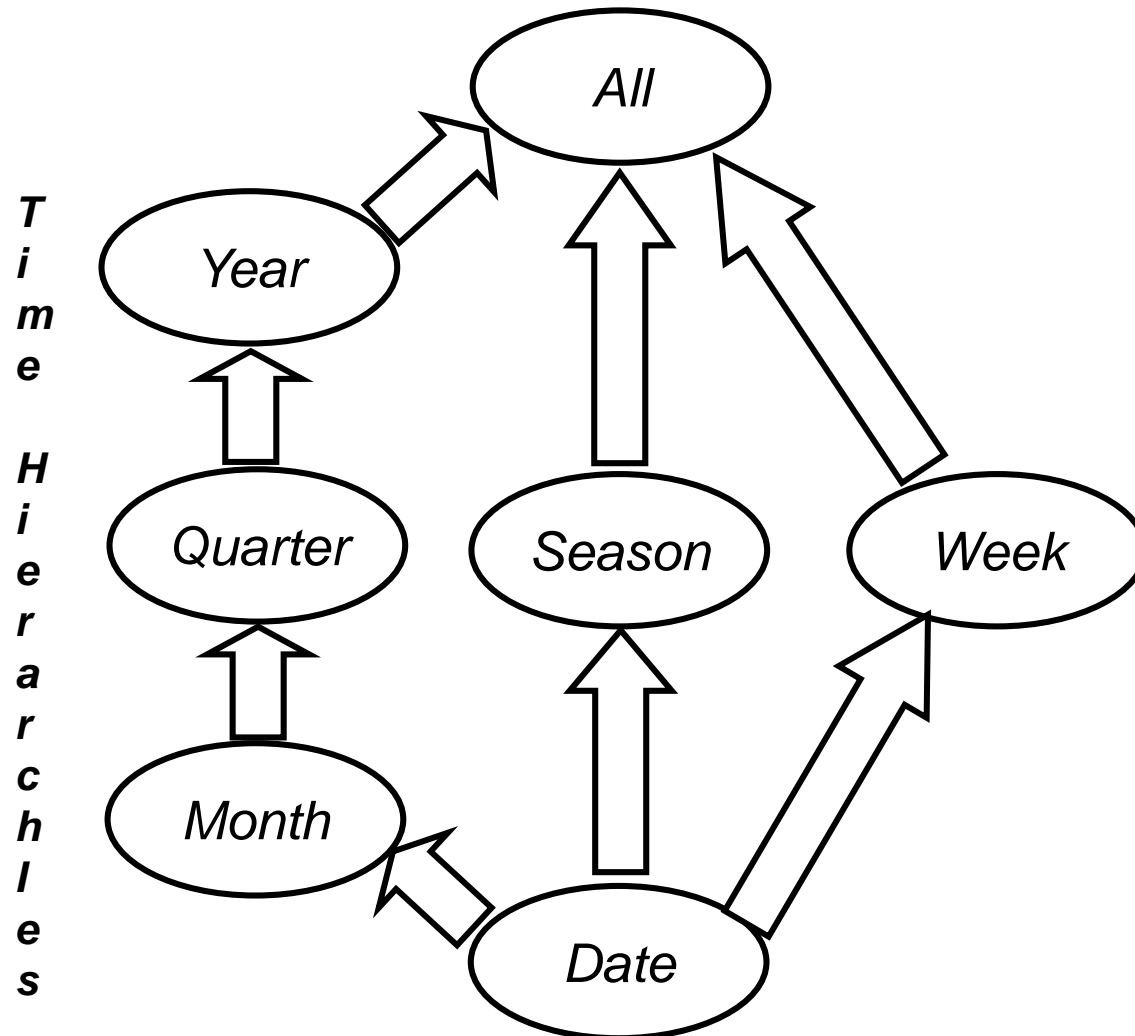
- Example:

```
ATTRIBUTE City DETERMINES Location.Mayor  
ATTRIBUTE Country DETERMINES  
Location.President;
```

Determines the end
of the DIMENSION
statement

- Fd's: City → Mayor, Country → President

Time Dimension with Three Hierarchies



Time Dimension Table

```
CREATE TABLE Time (  
  CurrentDate Date PRIMARY KEY,  
  Month int NOT NULL,  
  Quarter int NOT NULL,  
  Year int NOT NULL,  
  Season int NOT NULL,  
  Week int NOT NULL,  
  DayOfWeek varchar (10),  
  MonthName varchar (10));
```

DIMENSION of the Time Hierarchies

```
CREATE DIMENSION time_dim
    LEVEL CurrentDate IS Time.CurrentDate
    LEVEL Month IS Time.Month
    LEVEL Quarter IS Time.Quarter
    LEVEL Year IS Time.Year
    LEVEL Season IS Time.Season
    LEVEL Week IS Time.Week
HIERARCHY weekly_rollup (
    CurrentDate CHILD OF
    Week )
```


HIERARCHY of the Time Hierarchy

```
HIERARCHY seasonal_rollup (  
    CurentDate CHILD OF  
    Season )
```

```
HIERARCHY calendar_rollup (  
    CuurentDate CHILD OF  
    Month CHILD OF  
    Quarter CHILD OF  
    Year)
```

```
ATTRIBUTE CurrentDate DETERMINES  
Time.DayOfWeek
```

```
ATTRIBUTE Month DETERMINES  
Time.MonthName;
```

3NF Dimension Tables

```
CREATE TABLE Product (  
    ProductId int PRIMARY KEY,  
    ProductName varchar(20),  
    TypeId int NOT NULL REFERENCES  
    ProdType);
```

```
CREATE TABLE ProdType(  
    TypeId int PRIMARY KEY,  
    TypeDesc varchar(30),  
    IndustryId int NOT NULL REFERENCES  
    Industry);
```

```
CREATE TABLE Industry (  
    IndustryId int PRIMARY KEY,  
    IndustryName varchar);
```

CREATE DIMENSION product_dim

```
CREATE DIMENSION product_dim
    LEVEL ProductId IS Product.ProductId
    LEVEL TypeId IS ProdType.TypeId
    LEVEL IndustryId IS Industry.IndustryId
HIERARCHY merchandise_rollup (
    ProductId CHILD OF
    TypeId CHILD OF
    IndustryId
    JOIN KEY Product.TypeId REFERENCES TypeId
    JOIN KEY ProdType.IndustryId REFERENCES IndustryId
)
ATTRIBUTE ProductId DETERMINES Product.ProductName
ATTRIBUTE TypeId DETERMINES ProdType.TypeDesc
ATTRIBUTE IndustryId DETERMINES Industry.IndustryName;
```

Dimensions and Constraints

- The reliability of the information that dimensions convey to the query optimizer depends on the satisfaction of the following constraints:
 - Primary Key (of the fact table and dimension tables),
 - Referential Integrity (between fact and dimension tables, and between snow flaked dimension tables),
 - Not Null, and
 - Functional Dependency (between elements of an attribute hierarchy)
- By the rule, checking satisfaction of these constraints is very expensive in a DW
- ROLAP engines offer commands to disable constraint checking during refreshing and to relay on the quality of ECTL tools
- Also, a periodical checking by special utility programs is provided

Summary

- Dimension tables contain dimension attribute hierarchies, but do not convey a direct information about them
- Dimension hierarchies are crucial for query rewrite
- The statement:
 - `CREATE DIMENSION ... LEVEL`

with clauses:

- `HIERARCHY and`
- `ATTRIBUTE... DETERMINES`

are used to define attribute hierarchies and functional dependencies between their attributes

- Checking satisfaction of dimension constraints is expensive, but constraints are crucial