Data Warehousing Design

Adapted from Chapter 32 Database Systems:

A Practical Approach to Design :Implementation and Management by Connolly Begg 2015 © Pearson Education

Oracle Database Data Warehousing Guide 10*g Release 2*

References:

Chapter 19 Schema Modeling Technique

Chapter 10 Dimensions

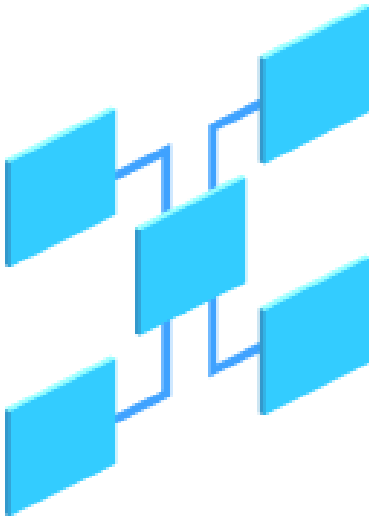
Chapter 3 Physical Design in Data Warehouses

Logical Design

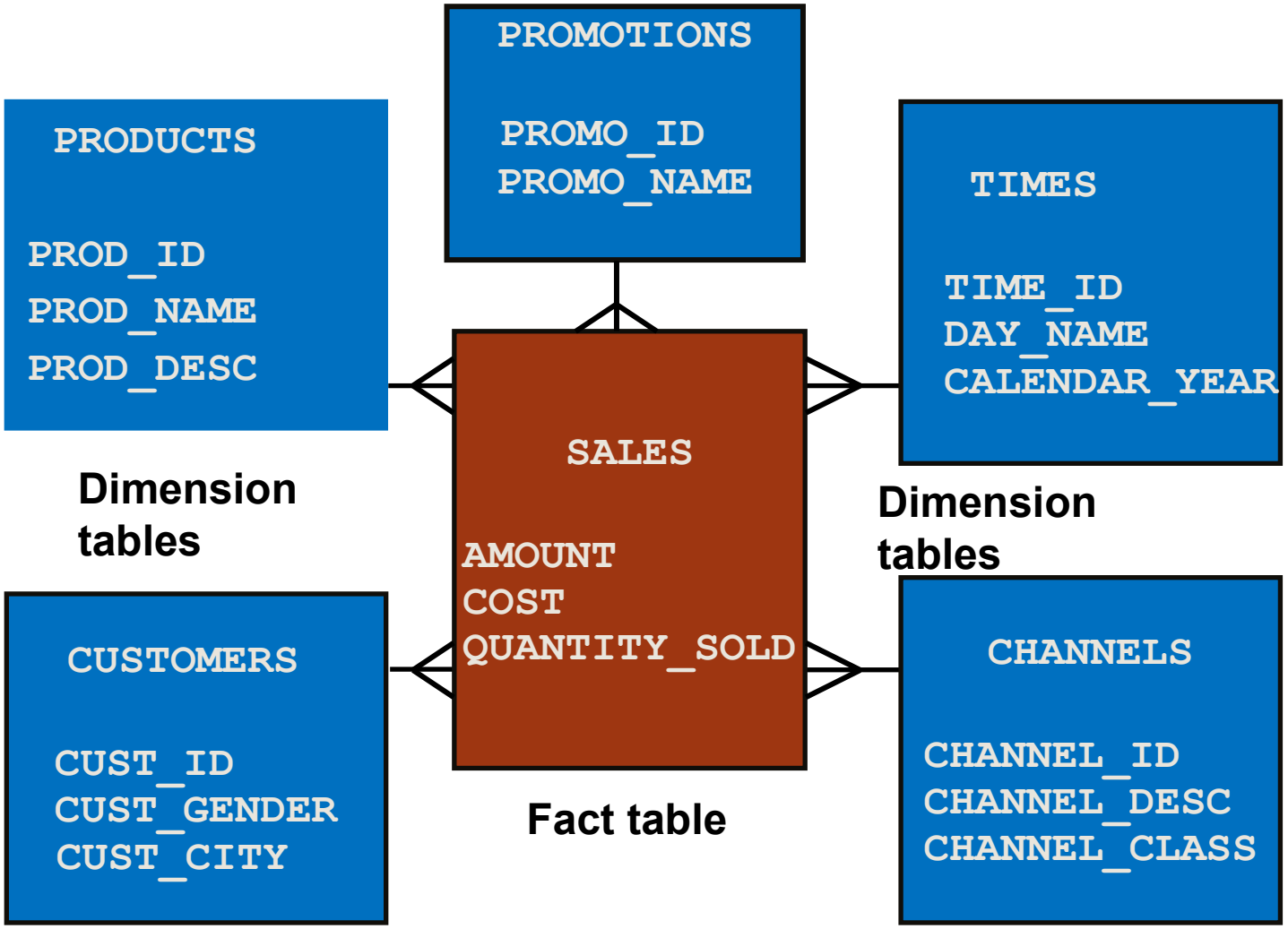
- A logical design is conceptual and abstract.
- Entity-relationship (ER) modeling is useful in identifying logical information requirements.
  - An *entity* represents a chunk of data.
  - The properties of entities are known as *attributes*.
  - The links between entities and attributes are known as *relationships*.
- Dimensional modeling is a specialized type of ER modeling (with some important differences) useful in data warehouse design.

Data Warehousing Schemas

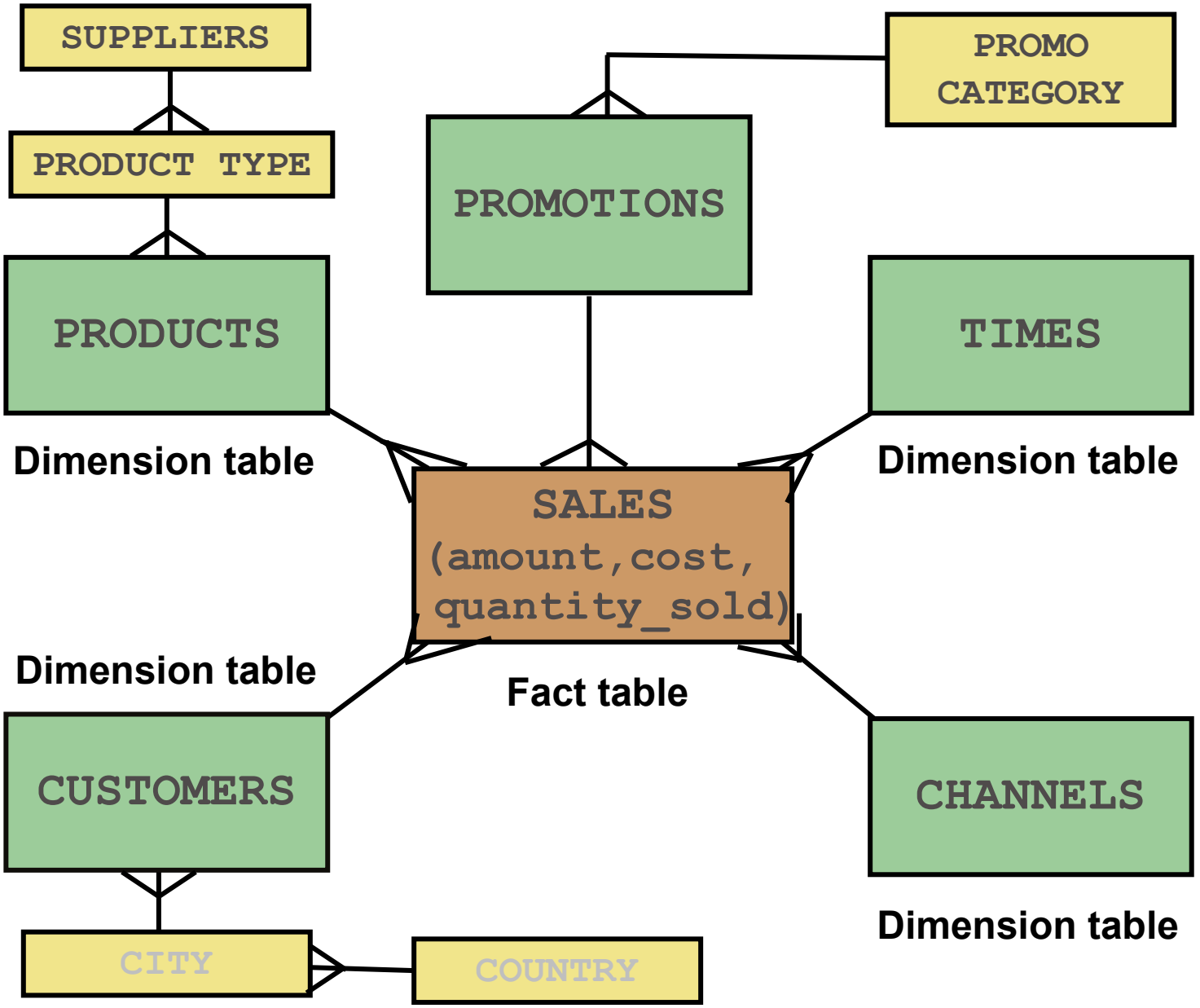
- Objects can be arranged in data warehousing schema models in a variety of ways:
  - Star schema (Kimball)
  - Snowflake schema
  - Third normal form (3NF) schema (Inmon)
  - Hybrid schemas e.g. StarFlakes & Constellations
- The source data model and user requirements should steer the data warehouse schema.
- Implementation of the logical model may require changes to enable you to adapt it to your physical system.



Star Schema Model



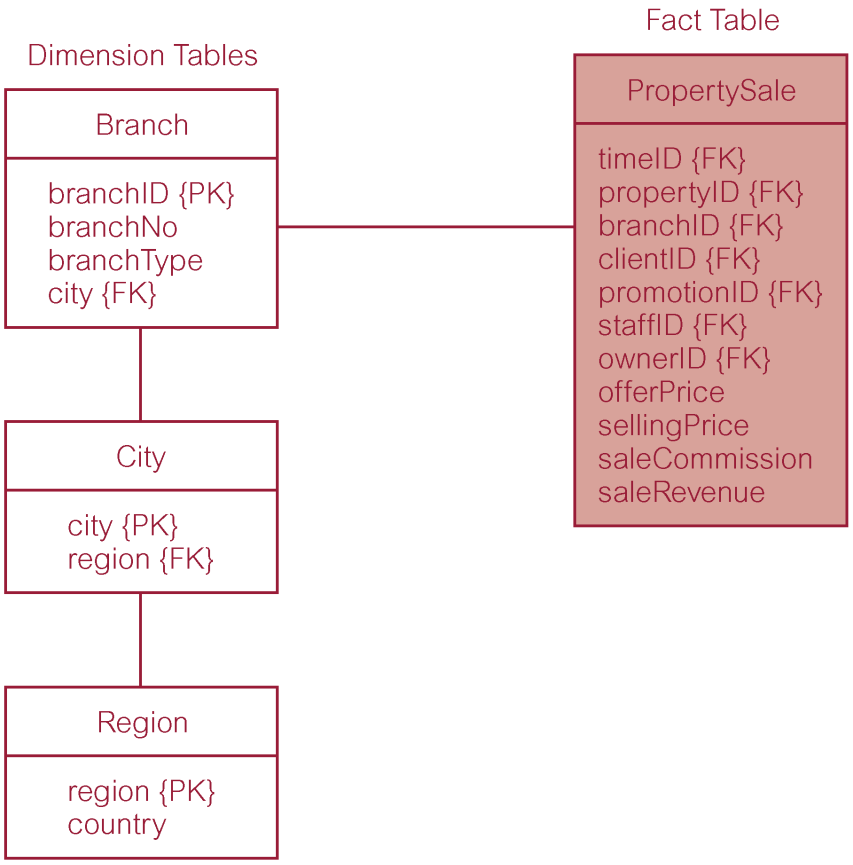
Snowflake Schema Model



Dimensionality Modelling Schema Types

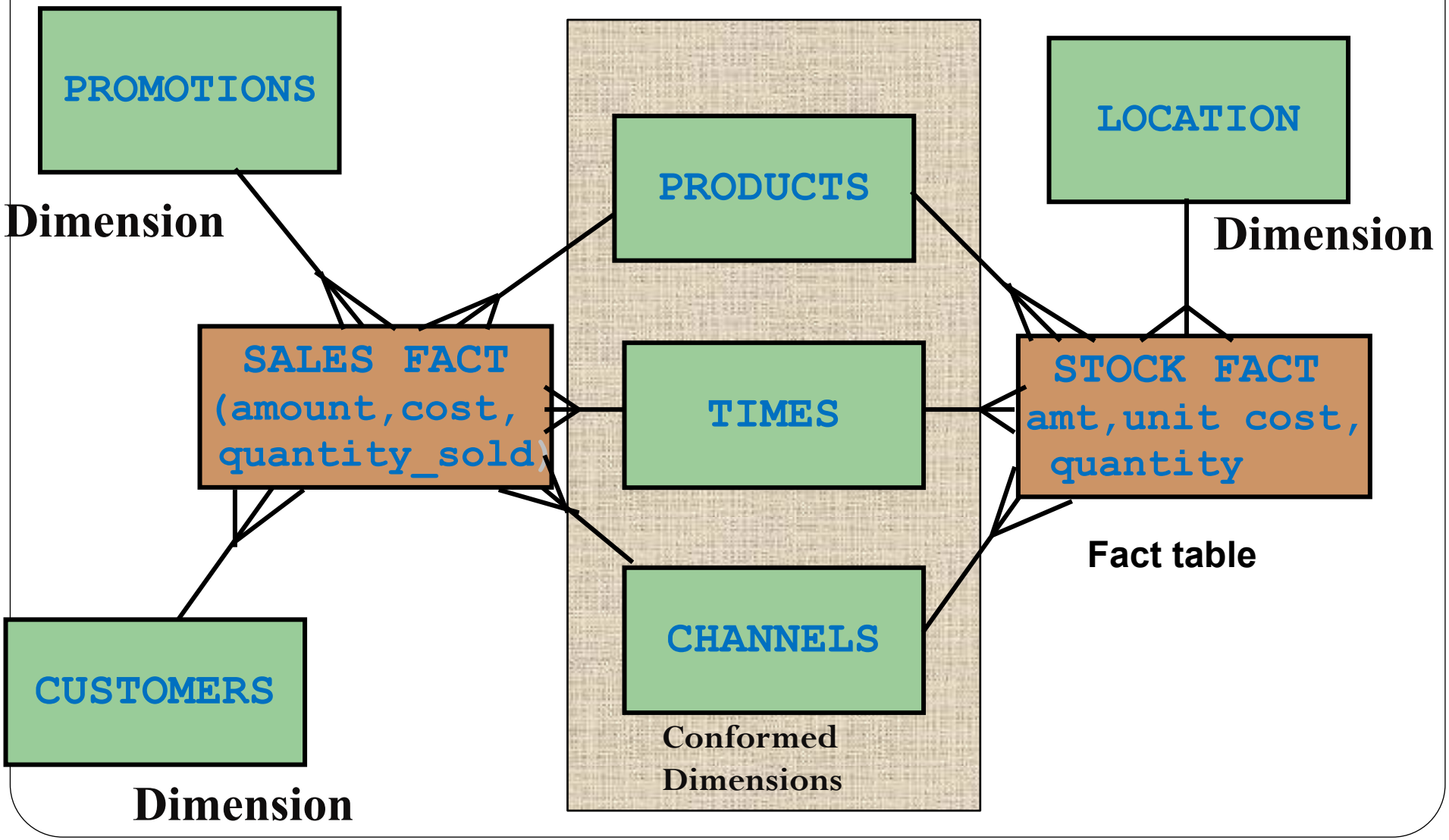
- Snowflake schema is a variant of the star schema where dimension tables contain normalized data.
- Starflake schema is a hybrid structure that contains a mixture of star (denormalized) and snowflake (normalized) schemas. Allows dimensions to be present in both forms to cater for different query requirements.
- **EXERCISE**
  - Draw a high Level Star Schema by interpreting the PropertySale Fact Table
  - Provide a snowflake schema (i.e. normalised) for the dimension "BRANCH" used by the PropertySale Fact Table that contains attributes branchId, branchNo, branchName, branch type, city, country, region
  - Note branchId is a Surrogate Key

Property sales with normalized version of Branch dimension table



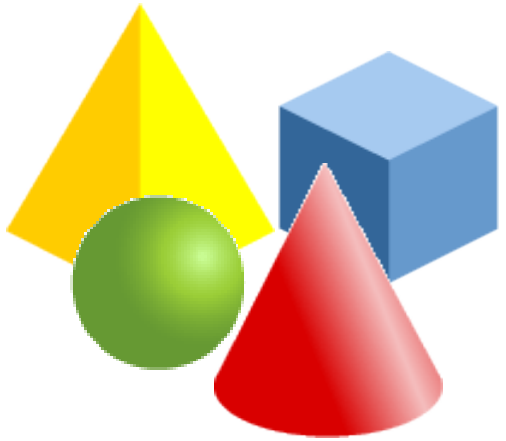
Snowflake

Fact Constellation or Galaxy



Data Warehousing Objects Recap

- Fact tables
  - Fact tables are the large tables that store business measurements.
- Dimension tables
  - A dimension is a structure composed of one or more hierarchies that categorizes data.
  - Unique identifiers are specified for one distinct record in a dimension table.
- Relationships
  - Relationships guarantee integrity of business information.



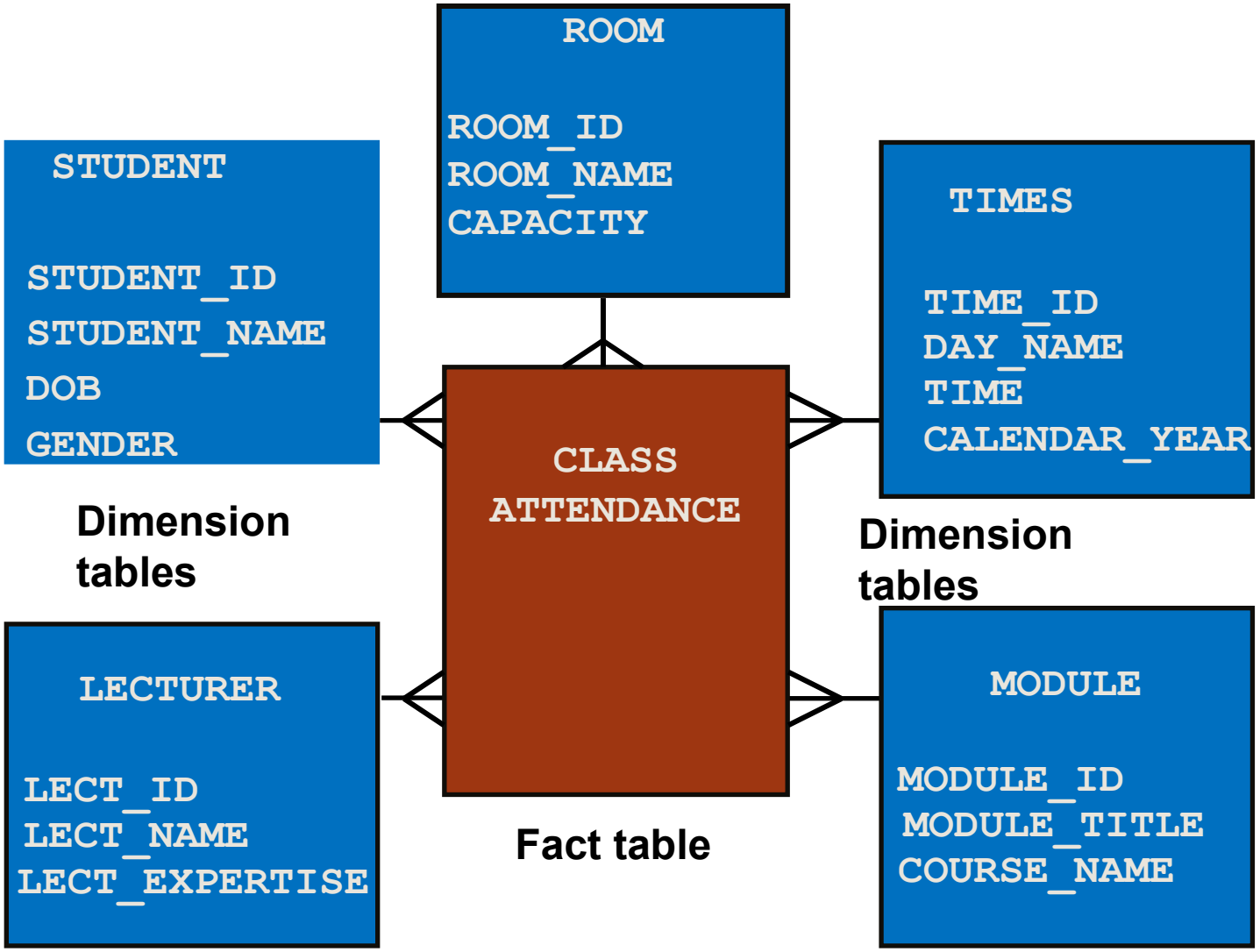
Fact Measures/Key Figures - Additive Nature

- Additive: Facts that can be summed up/aggregated across all of the dimensions in the fact table (e.g., discrete numerical measures of activity, i.e., quantity sold, euros sold)
- Semi-Additive: Facts that can be summed up for some of the dimensions in the fact table, but not the others (e.g., account balances, inventory level, distinct counts)
- Non-Additive: Facts that cannot be summed up for any of the dimensions present in the fact table (e.g., measurement of room temperature, percentages, profit margins)

Fact Table Types

- Different types of fact tables
  - Transactional – Additive facts tracking events over time
  - Snapshot or inventory – Pictures in time of levels or balances
  - Factless – NO Measures! just Dimensionality relationships

Factless Fact Table Schema Model



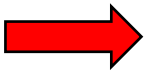
Slowly Changing Dimensions

- Changes to Dimensions usually happen sporadically and less frequently than fact table measurements
- Four Main Types
  1. Type 0 – Attribute values never change. Type 0 applies to most Date Dimension attribute
  2. Type 1 - Overwrite
  3. Type 2 - Add a New Dimension Record
  4. Type 3 - Add a New Field(s)

SCD Type-1 UPDATE

- Update Product Line to Rough Terrain because of a data error

CUSTOMER Dimension	Record/Row
Customer ID	3456
Name	Bike Mart
Customer Type	Retail
Product Line	**ALL**
Account Open	15-09-2002



Reseller Dimension	OverWrite Value
Customer ID	3456
Name	Bike Mart
Customer Type	Retail
Product Line (Type1Update)	**Rough Terrain**
Account Open	15-09-2002

Note: history can be lost if changing correct dimension (old) values to new values

SCD Type 2 UPDATE

Update Model and Class values to **Mountain** and **Heavy Frame**

PRODUCT Dimension	Row\Record
Product ID	BK-M825
Name	Yukon Special
Size	44
Model	All Terrain
Class	Light Frame

Mountain

Heavy Frame

PRODUCT DIM	Original Record	NEW RECORD
Product No (Surrogate Key)	462	477
Product ID (Natural Key)	BK-M825	BK-M825
Name	Yukon Special	Yukon Special
Size	44	44
Model (Type 2)	All Terrain	**Mountain**
Class (Type 2)	Light Frame	**Heavy Frame**
Start Time End Time	1/1/10 9/9/10	10/9/10 NULL

Note: Both Records now exist in the Product Dimension Table

SCD TYPE 3 – UPDATE – Hold some History

Update City value From Dublin to **Galway**

CUSTOMER	Record\Row
Customer Key	W123
Name	John Smith
City	Dublin

Galway

CUSTOMER	DIMENSION
Customer Key	W123
Name	John Smith
Original City	Dublin
Current City	Galway
Effective Date	15-09-2020

SCD TYPE 3 – UPDATE – Alternative Values

Product	Record\Row
Product Key	p9923
Name	Marking Pen
Category	Household Good

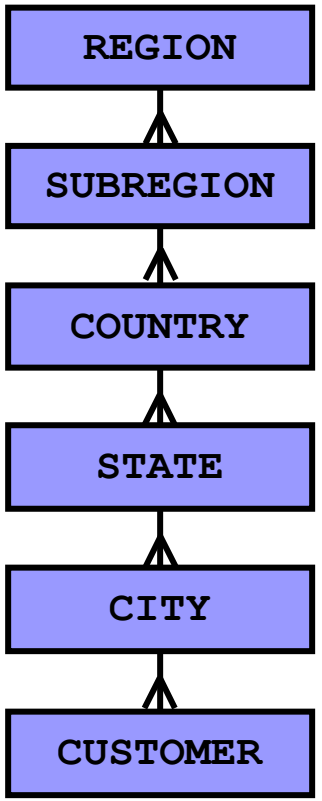
CUSTOMER	DIMENSION
Product Key	p9923
Name	John Smith
Name	Marking Pen
Category	Household Good
Alt. Category	Art Supplies

Update Record to
hold alternative categories

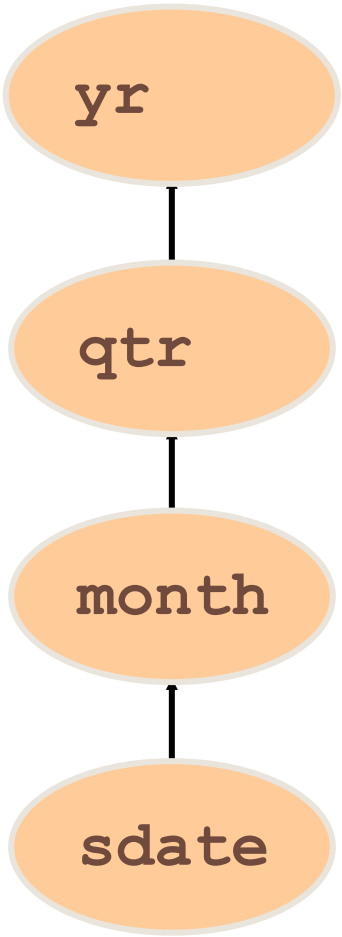
Dimensions and Hierarchies

- A dimension is a structure composed of one or more hierarchies that categorizes data.
- Dimensional attributes help to describe the dimensional value.
- Dimension data is collected at the lowest level of detail and aggregated into higher level totals.
- Hierarchies are structures that use ordered levels to organize data.
- In a hierarchy, each level is connected to the levels above and below it.

***CUSTOMERS* dimension hierarchy (by level)**



Defining Dimensions and Hierarchies




```
CREATE DIMENSION time_dim
 LEVEL sdate IS time.sdate
 LEVEL month IS time.month
 LEVEL qtr   IS time.quarter
 LEVEL yr    IS time.year
HIERARCHY calendar_rollup (
 sdate CHILD OF
 month CHILD OF
 qtr   CHILD OF  yr )
ATTRIBUTE month
  DETERMINES time.month_name;
```

Deriving Dimensional Models from ER Model

- Dimensional Models are derived from query analysis requirements as well as from the **Enterprise ER models**

4 Steps

- Step 1 Classify Entities
- Step 2 Design High-level star schema
- Step 3 Detailed Fact Table Design
- Step 4 Detailed Dimension Table Design

Deriving Dimensional Models from ER Model continued

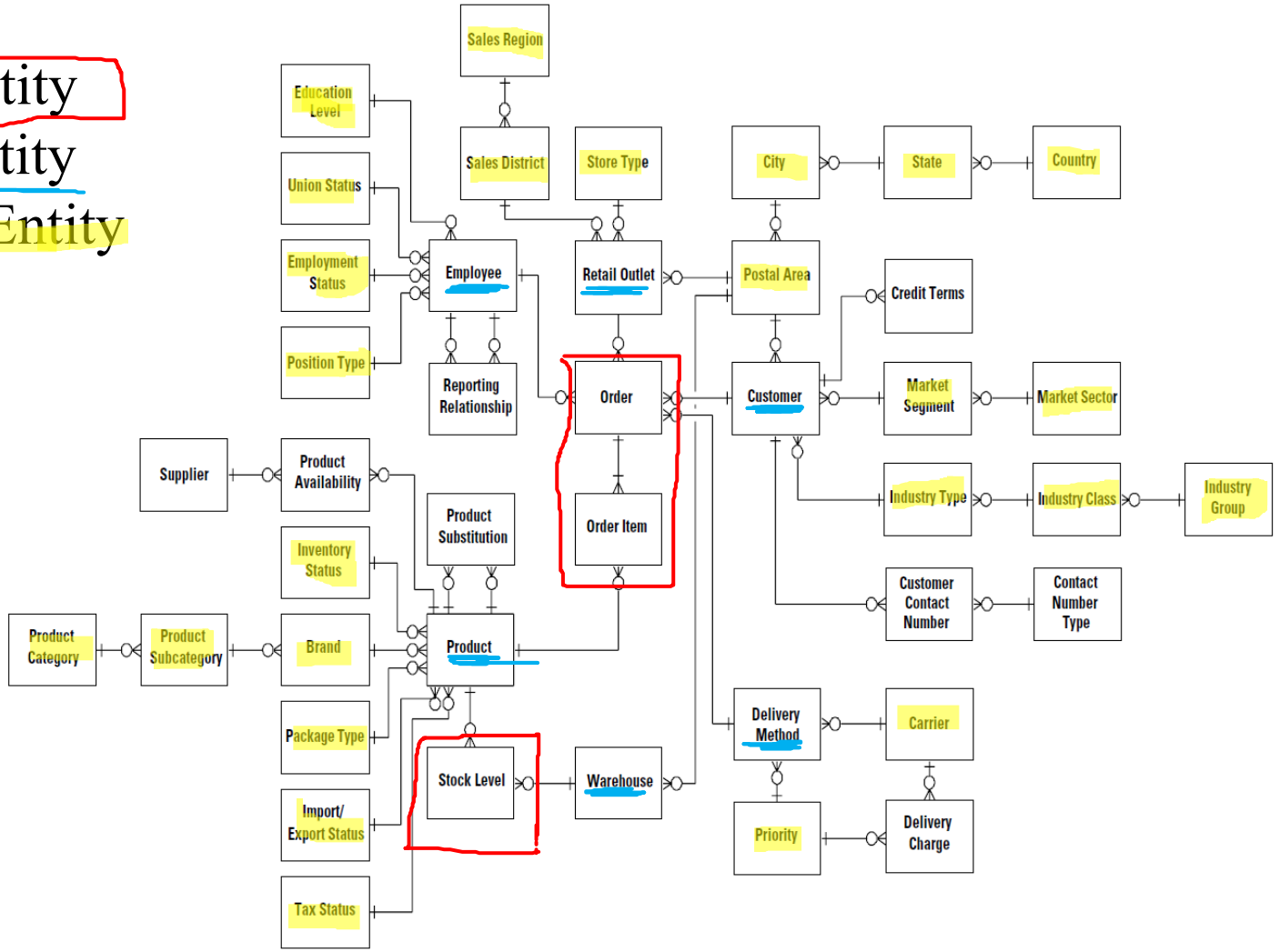
- Step 1 – Classify Entities
  - **Transaction Entities**
    - Entities that record details of business events e.g. orders, bank transactions, insurance claims, hospital attendances, inventory control.
  - **Component Entities**
    - Entities that are directly related to a transaction entity by a 1:m relationship
  - **Classification Entities**
    - These entities are related to a component entity by a chain of 1:m relationships.
- Step 2 Design High-level star schema
  - Identify Star Schemas Required
  - Define Level of Summarisation
  - Identify Relevant Dimensions

Identify the Entities

Transaction Entity

Component Entity

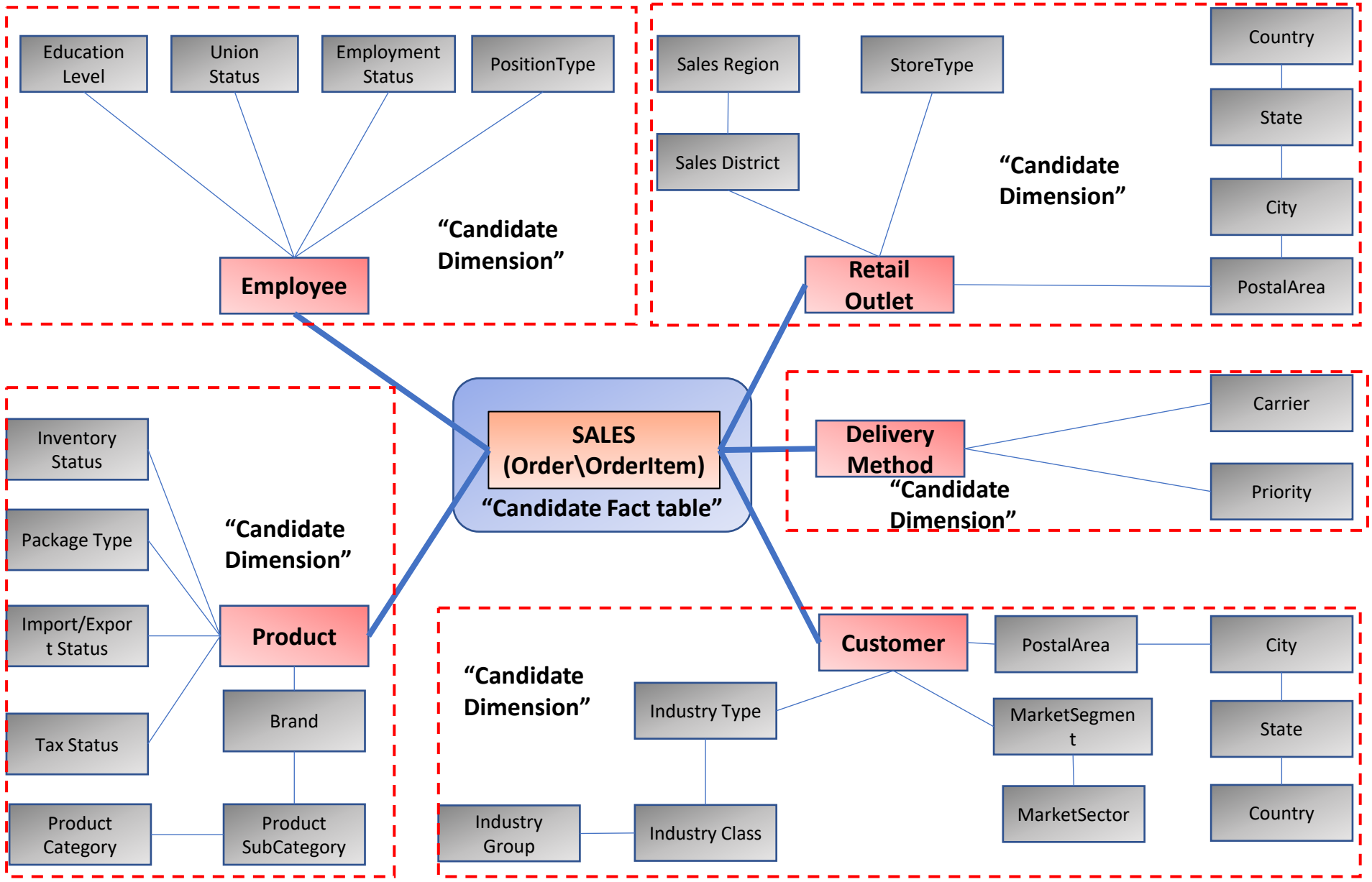
Classification Entity



Deriving Dimensional Models from ER Model continued

- Step 3 – Detailed Fact Table Design
  - Define key
  - Define Facts\ measures
- Step 4 Detailed Dimension Table Design
  - Define Dimension Key
  - Collapse Hierarchies
  - Replace codes and abbreviations by descriptive text

Candidate **SALES** Star Schema



Candidate **STOCK** Star Schema

Where is the time Dimension?