

# System Design

## Design of Data Model

**Aphrodice Rwagaju**

SAD

Y2 Term 3

# Data Modeling

- Goal – make sure all data objects required by a database are completely and accurately represented
- Data model design – the blueprint for creating a physical implementation of a database

# Data Modeling Terms

- **Entity** – a class of real world objects having common attributes (e.g., sites, variables, methods).
- **Attribute** – A characteristic or property of an entity (site name, latitude, longitude)
- **Relationship** – an association between two or more entities
- **Cardinality** – the number of entities on either end of a relationship (one-to-one, one-to-many, many-to-many, etc.)

# Data Modeling Exercise

- Consider:
  - What is the “entity”?
  - What are the “attributes” of the entity?



# Data Modeling Exercise

- What is the entity?
- What are the attributes?





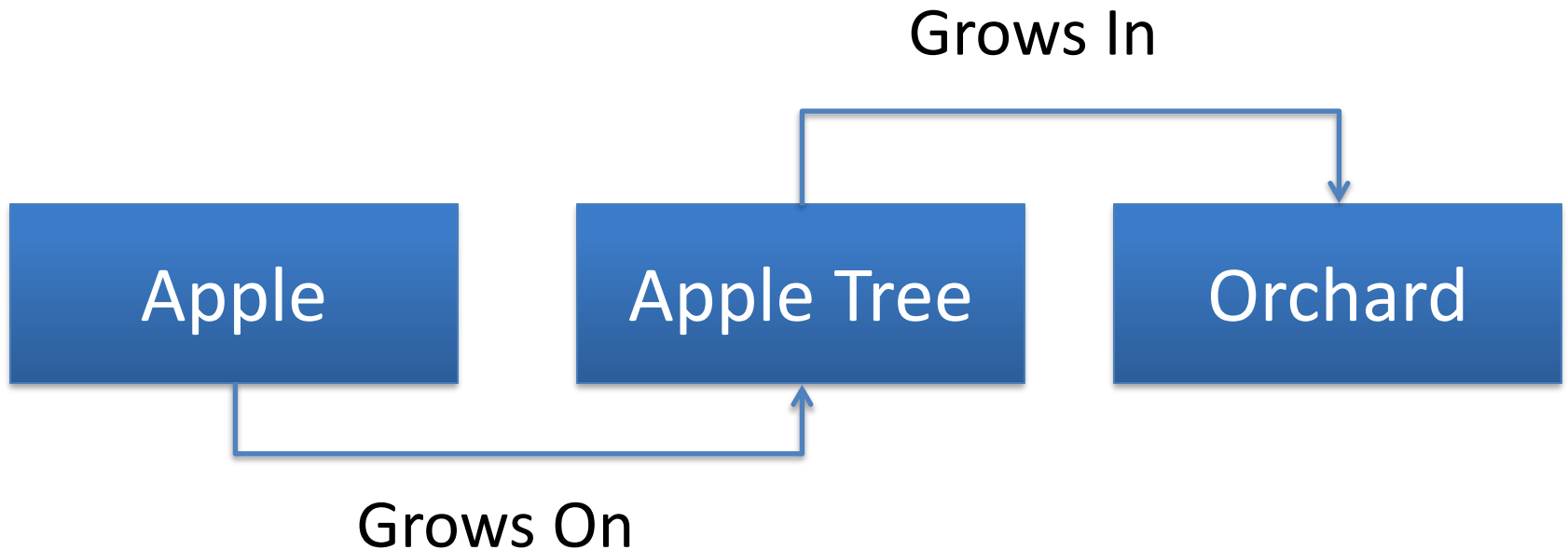
# Data Modeling Exercise

- What is the entity?
- What are the attributes?



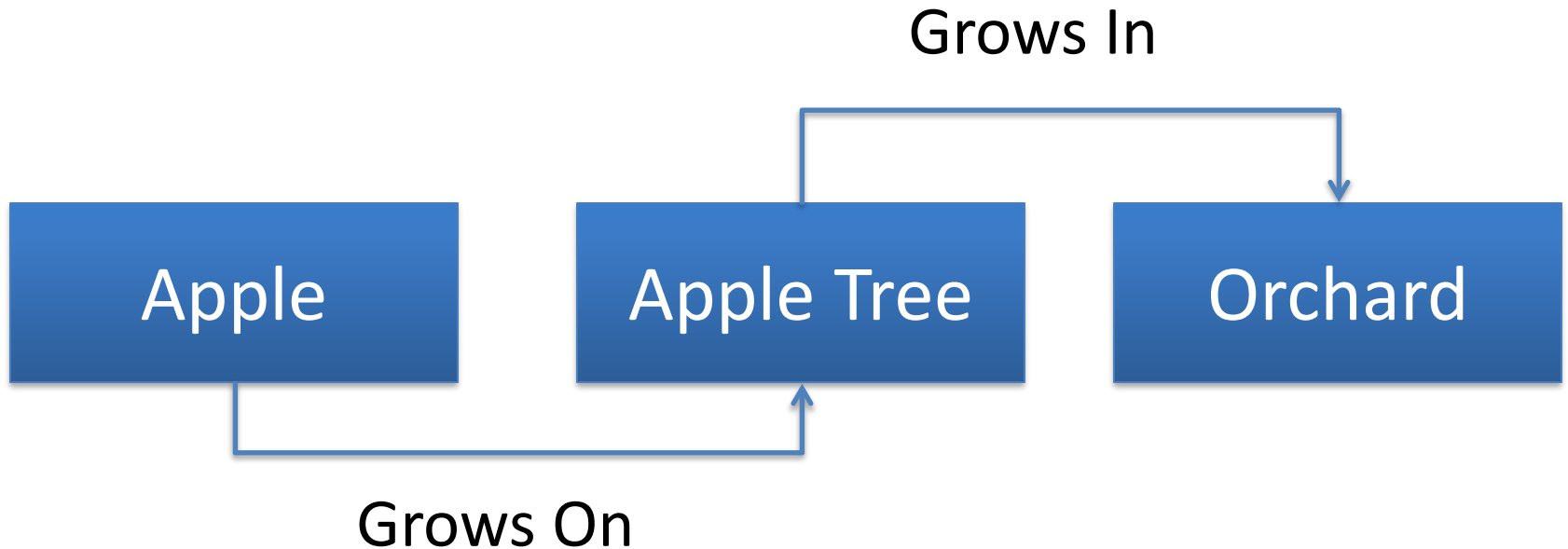
# Data Modeling Exercise

- What are the relationships?



# Data Modeling Exercise

- What are the relationships?



What about the business rules?



# Data Model Requirements

- What is the information/data domain that you are modeling?
- What are the queries that you want to do?
- What software do you want (have) to use?
- How do you want to share the data?

# Data Model Design

- Our focus – relational data model design
- Three stages:
  - Conceptual data model
  - Logical data model
  - Physical data model

# Conceptual Data Model

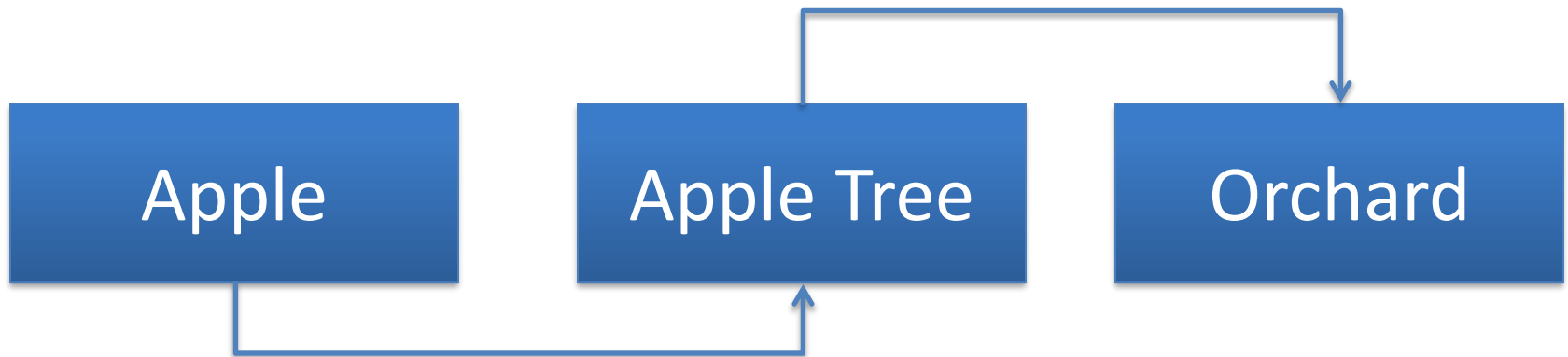
## (AKA – The Information Model)

- High-level description of the data domain
- Does not constrain how that description is mapped to an actual implementation in software
- There may be many mappings
  - Relational database
  - Object model
  - XML schema, etc.

# Apple/Tree/Orchard Conceptual Model



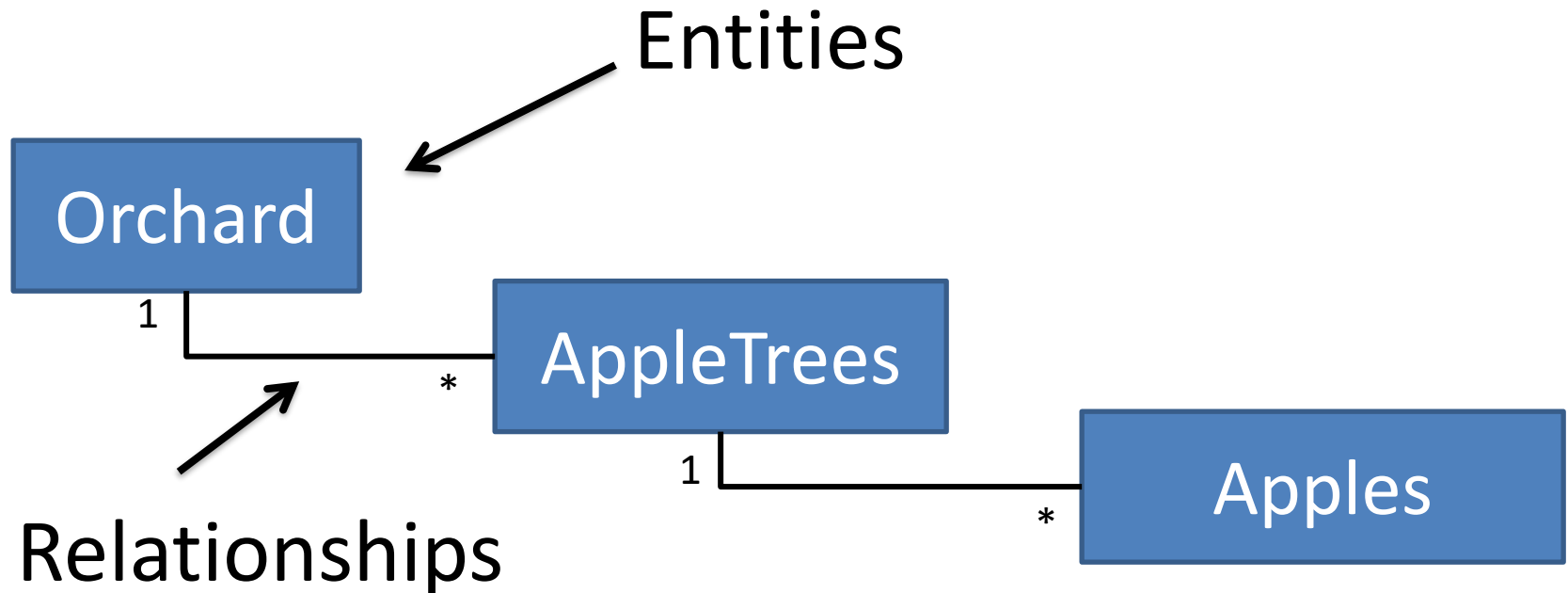
Grows In



Grows On

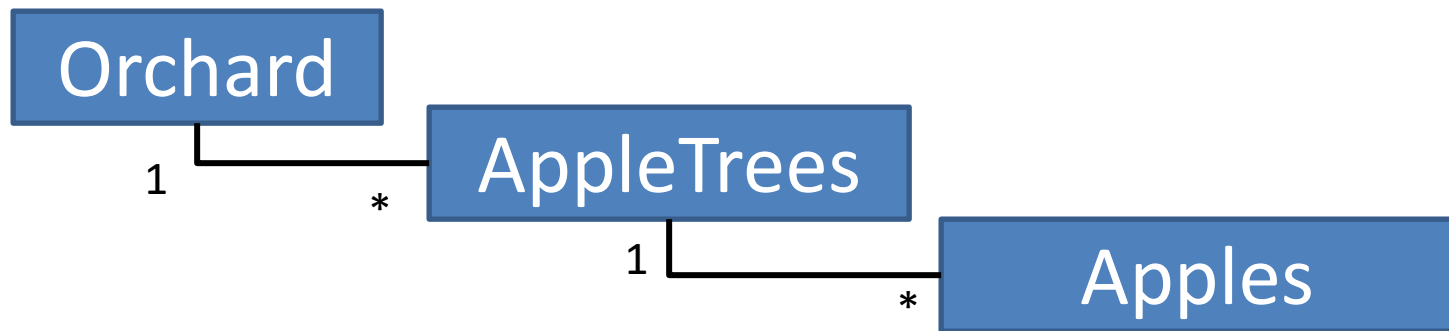
# Conceptual Data Model

- Defines scope of the domain
- Defines and organizes data requirements
- Defines **entities** and **relationships** among them



# Defining Entities and Relationships

- Instead of beginning with this:



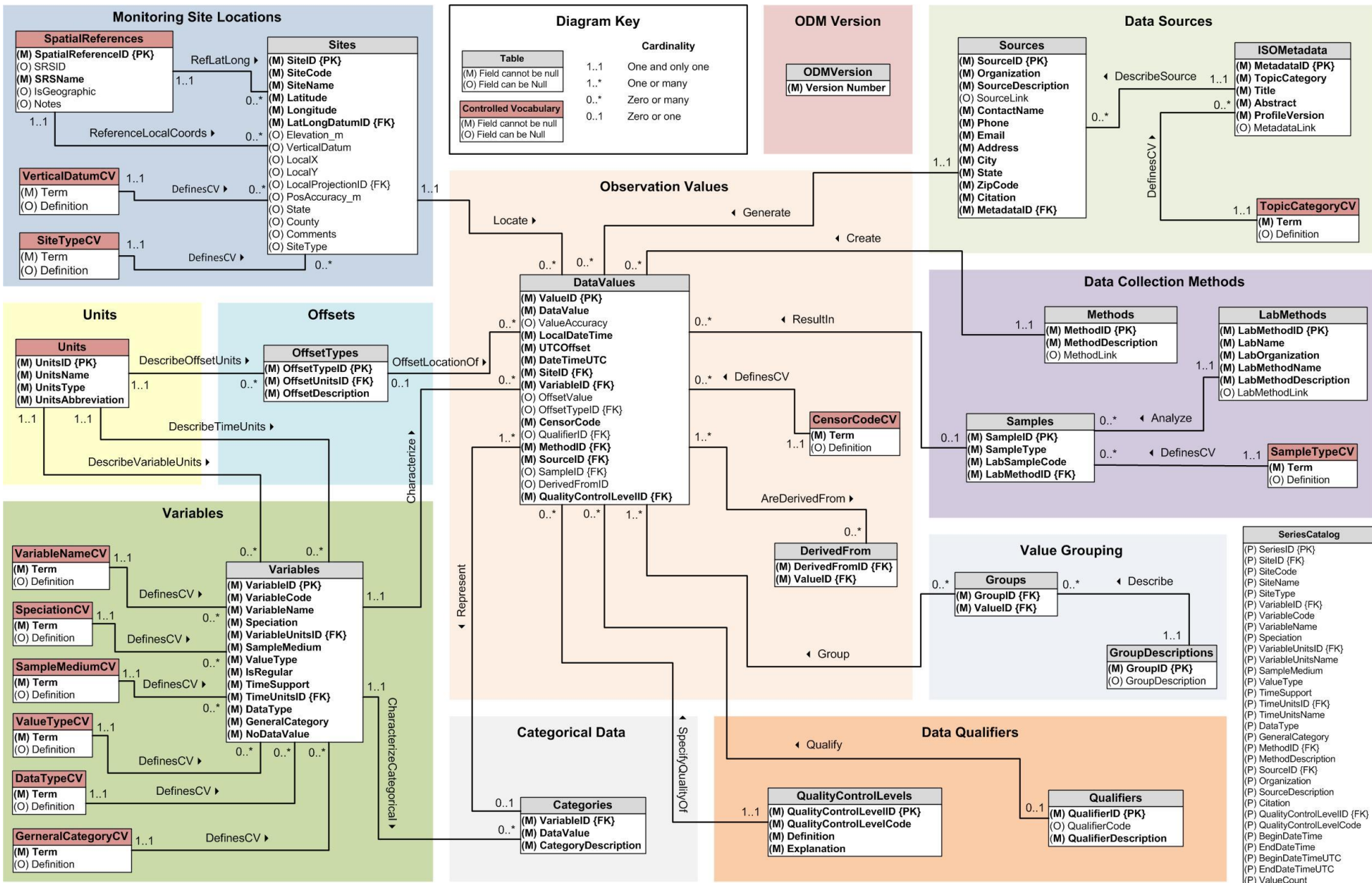
- Sometimes its easier to write statements:
  - Many **AppleTrees** *are grown* in an **Orchard**.
  - Each **AppleTrees** *contains* one or more **Apples**.
- The **nouns** become entities and the **verbs** become relationships.

# Logical Data Model

- Technology independent
- Contains more detail than the Conceptual Data Model
- Considered by many to be just an expanded conceptual data model
- Defines
  - Entities AND their attributes
  - Relationships AND cardinality
  - Constraints
- Generally completed as a documented Entity Relationship (ER) Model or diagram

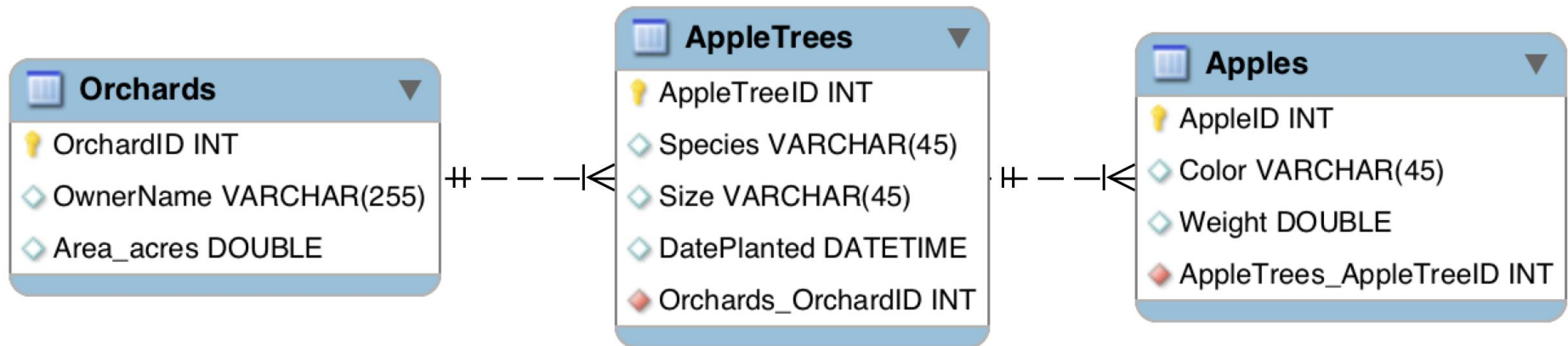


# Example: Logical Data Model ER Diagram



# Entity Relationship Diagram

- Documentation of the structure of the data
- Used to communicate the design
- Serve as the basis for data model implementation



# Entity Relationship Diagram

## (Relation Database Context)

- Entities effectively become tables
- Attributes describe entities and become fields (columns) in tables
- Relationships link tables on a common attribute or “key” and become formal constraints (part of the business rules)

# Entities and Attributes

- What are the attributes?



## Attributes

- Tree Name:
- Tree Code:
- Tree Type:
- Description:

## Values

Apple  
APP-01  
Hybrid  
Descriptions to be added here.

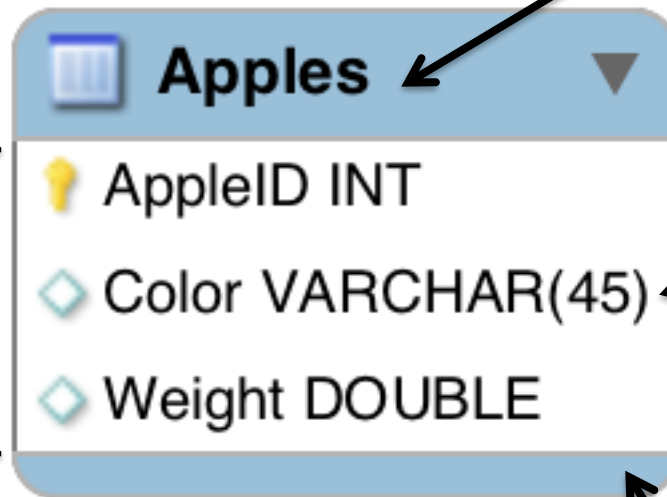
Entity = ApplTree

**Relational Model: All instances of an entity have the same attributes**

# ER Diagram Entity Notation



Attributes







Entity Name

Data Type

Entity

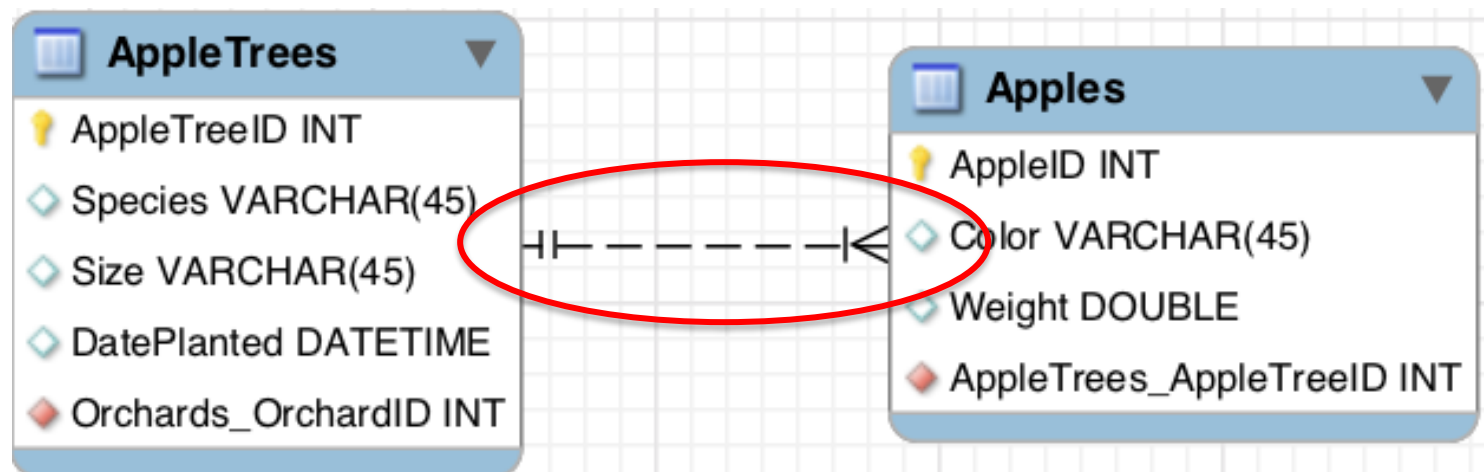
# ER Diagram Relationship Notation

- Multiple notation systems are used
- Each software program is a little different
- Most common is “Crows Foot”

	Crows Foot	Alternative
from Zero to Many		0 .. *
from One to Many		1 .. *
from One to One i.e., one and only one		1 .. 1
from Zero to One		0 .. 1

# Relationship Cardinality and Participation

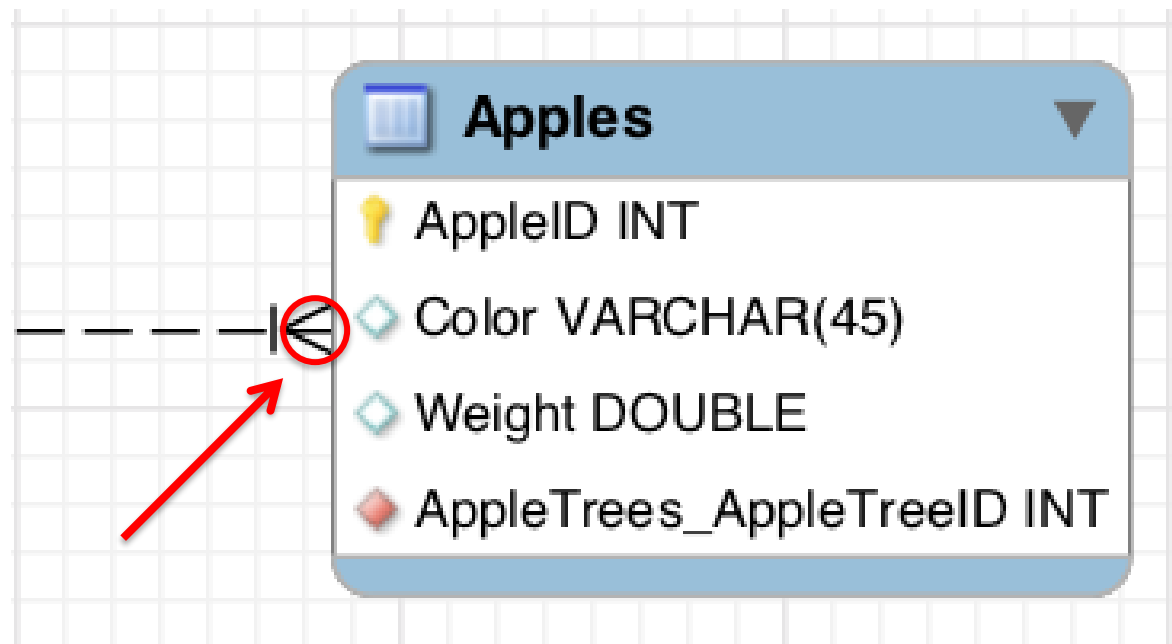
- **Cardinality:** maximum number of times an instance in one entity can be associated with instances in the related entity
- **Participation:** minimum number of times an instance in one entity can be associated with instances in the related entity





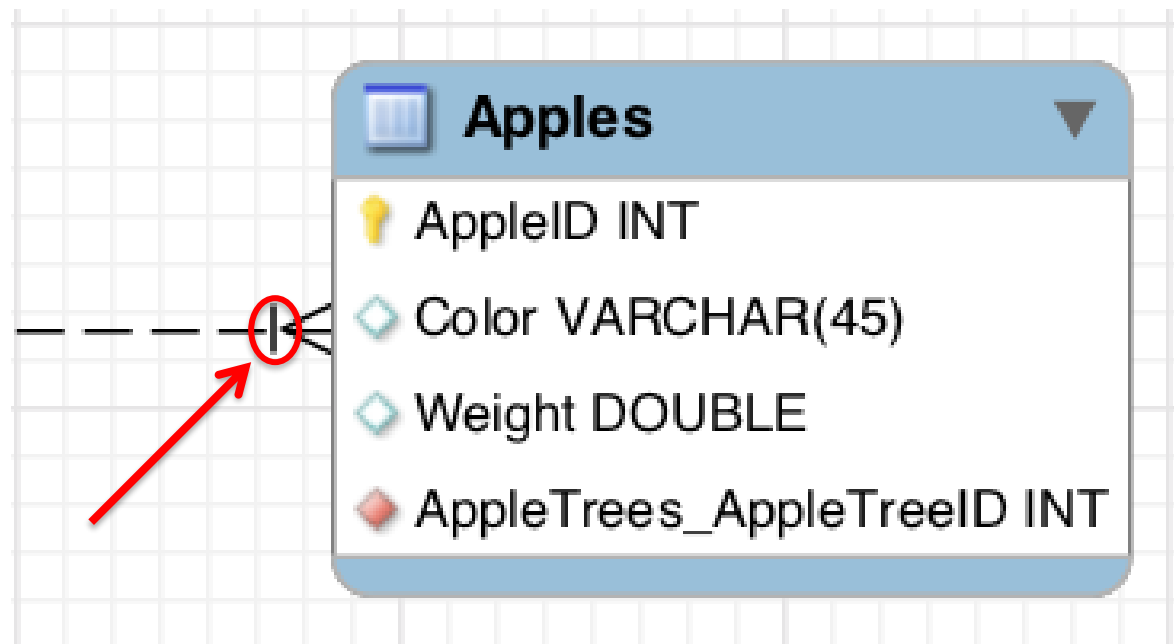
# Cardinality

- Maximum number of times an instance of an entity can be associated with instances of a related entity
- Can ONLY have values of 1 or many
- Located closest to the entity in Crows Foot notation



# Participation

- Minimum number of times an instance in one entity can be associated with instances in the related entity
- Indicates whether an instance of an entity **MUST** participate in the relationship
- Can **ONLY** have the values of 0 or 1

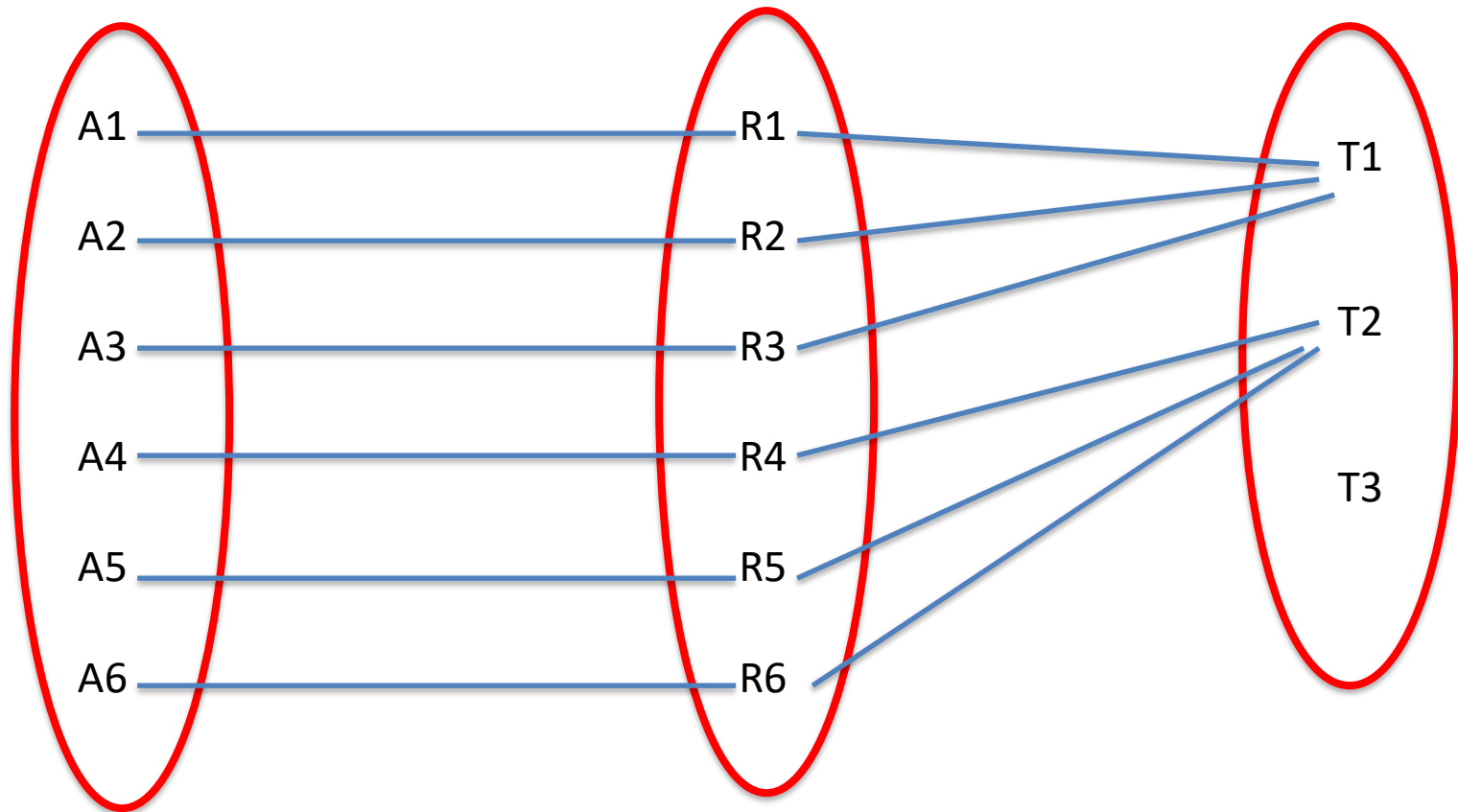


# Mapping Cardinality and Participation

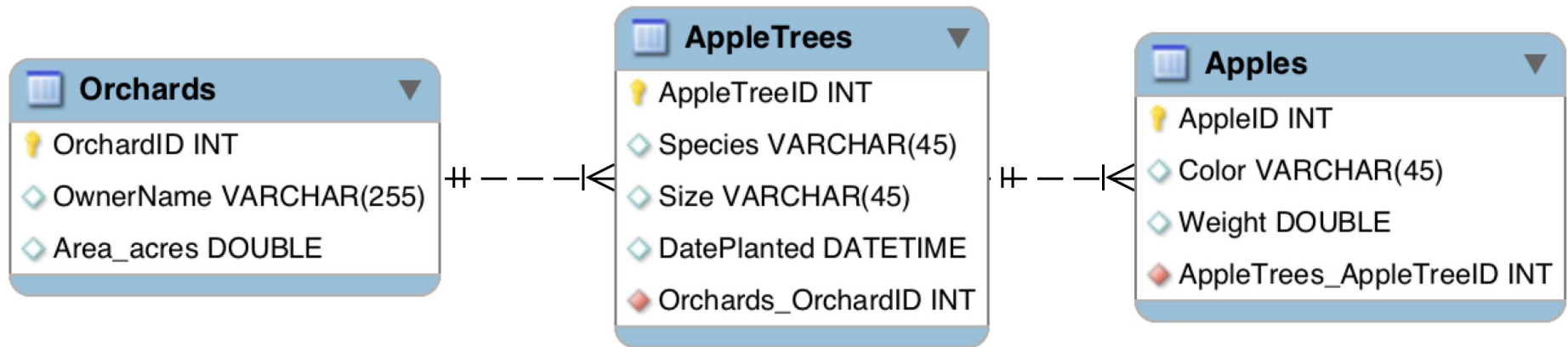
Apple

Grows\_on

AppleTree



# Reading Cardinality and Participation



from Zero to Many



from One to Many



from One to One

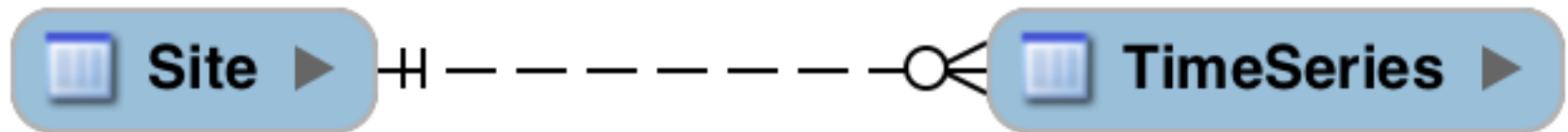
i.e., one and only one



from Zero to One

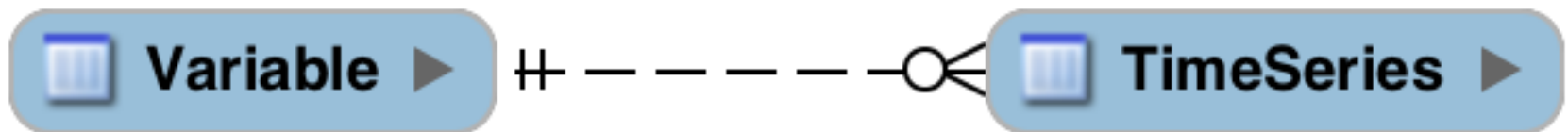


# Relationship Examples



Left to Right: A site has 0 or more time series of data.

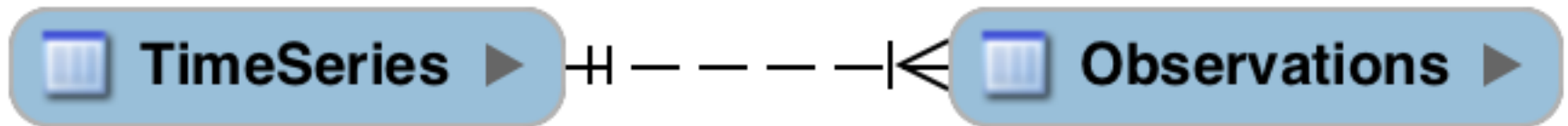
Right to Left: A time series is measured at 1 and only 1 site.



Left to Right: A variable has 0 or more time series of data.

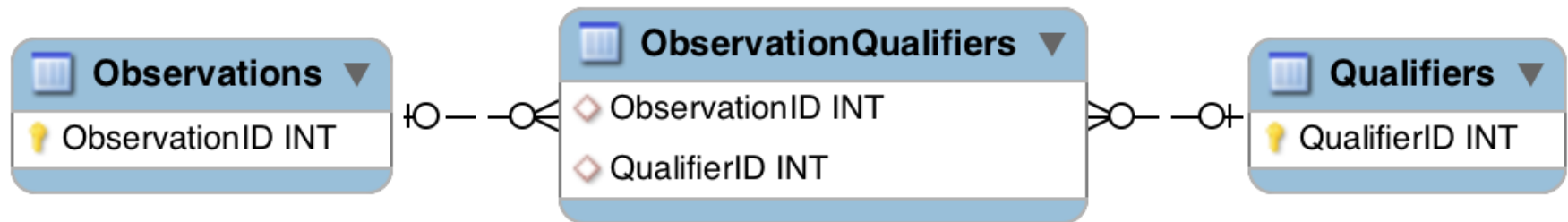
Right to Left: A time series can have 1 and only 1 variable.

# Relationship Examples



Left to Right: A TimeSeries has 1 or more Observations.

Right to Left: An Observation belongs to 1 and only 1 TimeSeries.

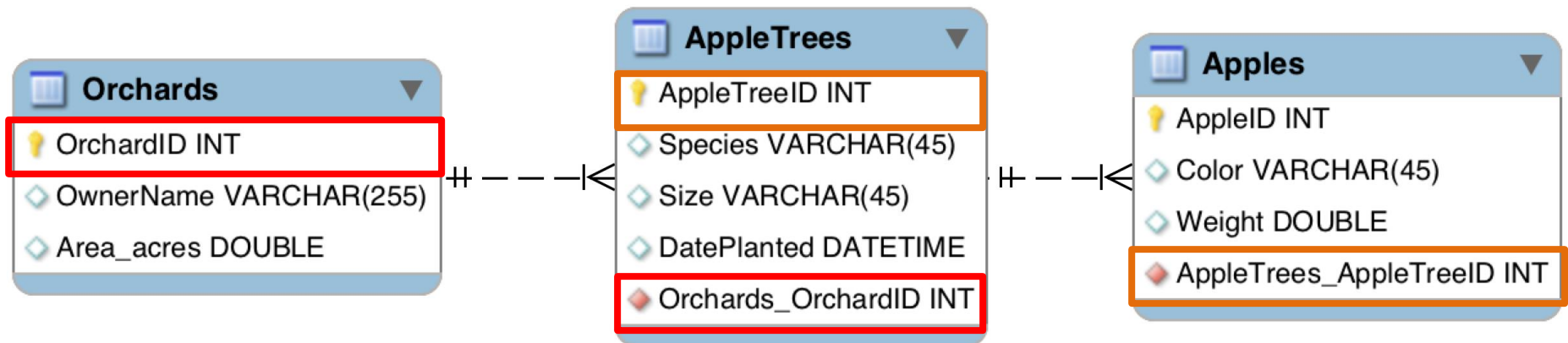


Left to Right: An Observation has 0 or more Qualifiers.

Right to Left: A Qualifier describes 0 or more Observations.

# Primary and Foreign Keys

- Each row in a table should have an attribute that is a persistent, unique identifier – the “Primary Key”
- Primary key in “parent” table
- Foreign key in “child” table





# Primary and Foreign Key Example

OrchardID	OwnerName	Area_acres
1	Mahoro Peace	120.5
2	Mucyo David	20

Orchards

Apple  
Trees

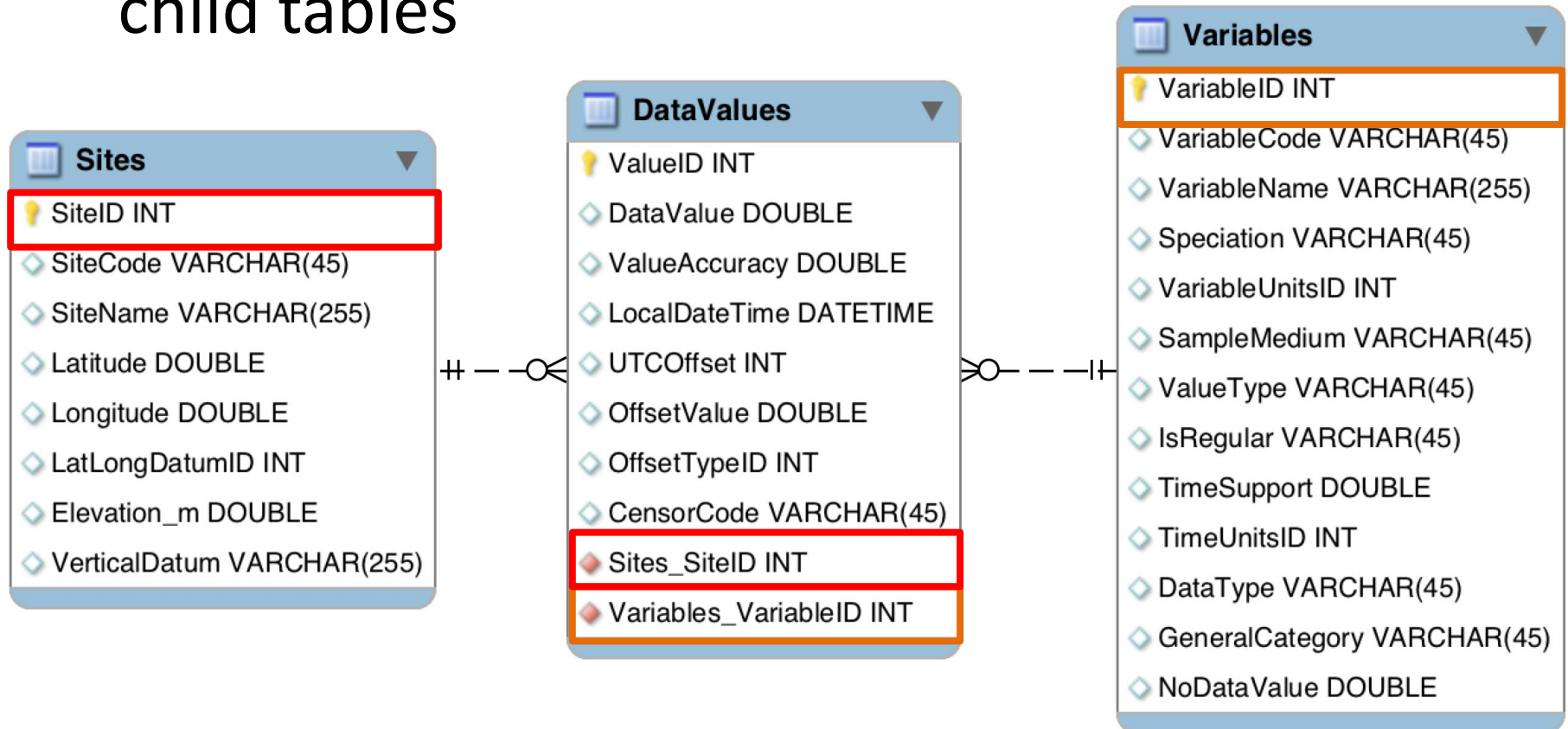
AppleTreeID	Species	Size	DatePlanted	OrchardID
1	Honeycrisp	5	9/22/2015	1
2	Honeycrisp	6	9/22/2015	1
3	Honeycrisp	3	9/22/2015	1

AppleID	Color	Weight	AppleTreeID
1	Green	200	2
2	Green	180	2
3	Green	195	2

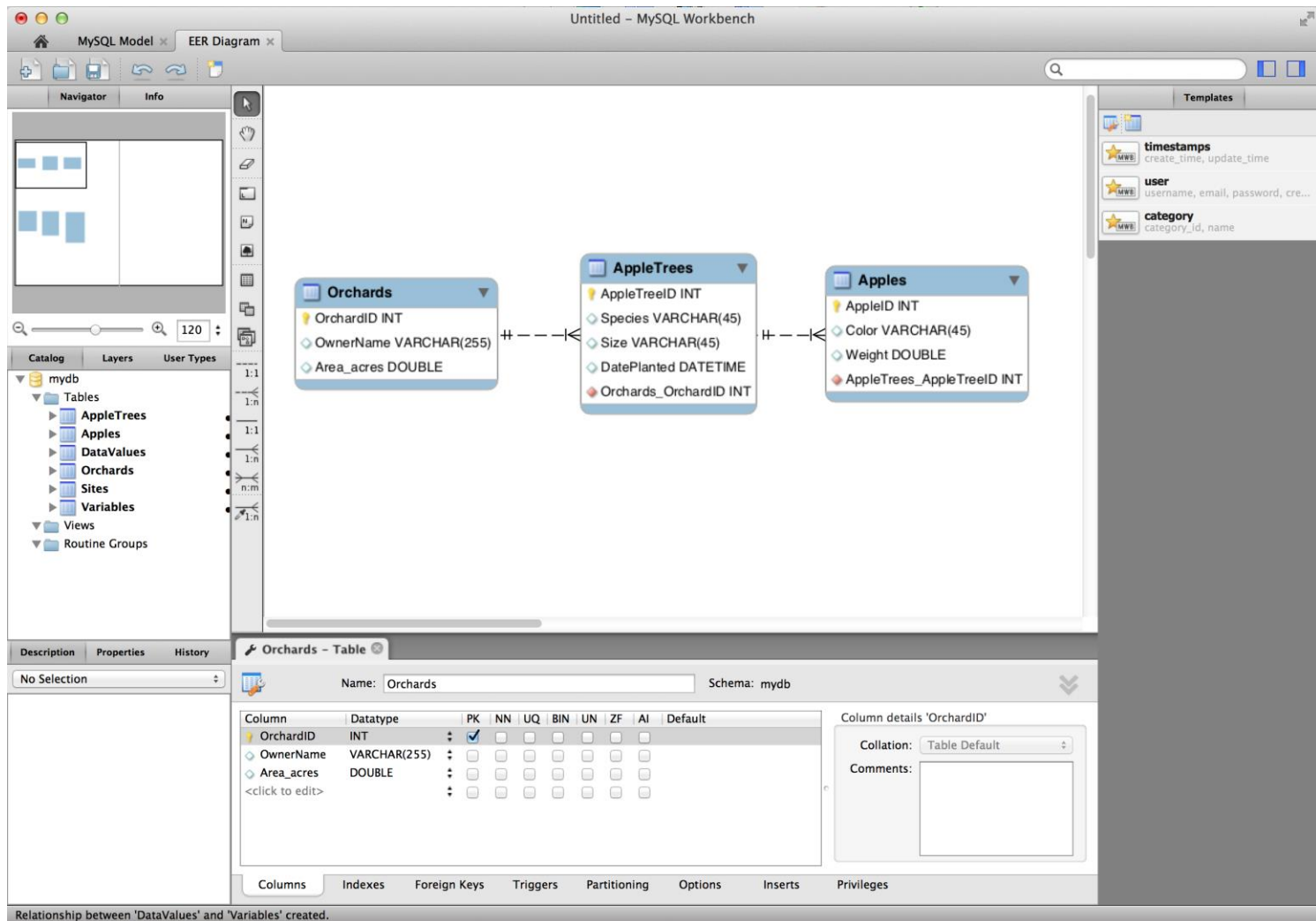
Apples

# Primary and Foreign Keys

- Usually consist of integer values
- MUST be the same data type in parent and child tables



# Creating an ER Diagram Using MySQL Workbench



# Summary

- Data model design is a 3 step process – conceptual, logical, physical (next time)
- Conceptual and logical data models can be expressed using Entity Relationship (ER) diagrams
- ER diagrams capture the entities, attributes, and relationships to model your information domain
- ER diagrams are a powerful way to document the design of your data model

# Steps in Data Model Design

1. Identify entities
2. Identify relationships among entities
3. Determine the cardinality and participation of relationships
4. Designate keys / identifiers for entities
5. List attributes of entities
6. Identify constraints and business rules

# Exercise

- Work in groups of 5 students
- Use MySQL Workbench to begin creating an Entity Relationship diagram
  - Identify 5 entities of school management information system
  - Specify attributes
  - Create relationships