

## Lecture 5

# Conceptual Design with the Entity Relationship Data Model (continued)

Week 3

## Overview

- Relationships
- Participation and cardinality constraints
- Higher order relationships
- Roles

© 2010 Griffith University

2

## Relationship type

- A relationship type defines a property of entities of an entity type

E.g. Sailor *serves on* a Ship

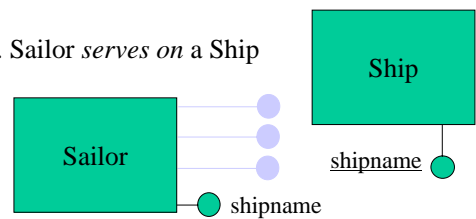
© 2010 Griffith University

3

## Relationship type

- A relationship type defines a property of entities of an entity type

E.g. Sailor *serves on* a Ship



shipname is the name of a 'Ship'. Hence, we do not represent this property of 'Sailor' as an attribute, but instead...

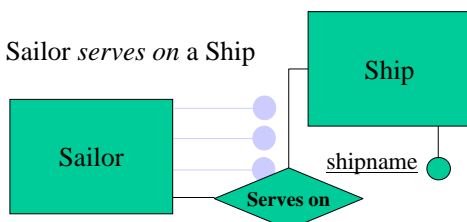
© 2010 Griffith University

4

## Relationship type

- A relationship type defines a property of entities of an entity type

E.g. Sailor *serves on* a Ship

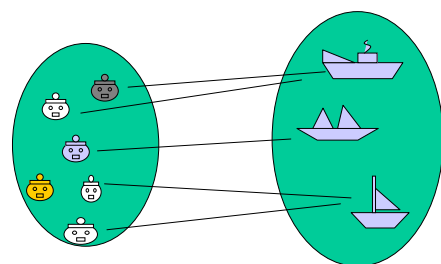


© 2010 Griffith University

We represent this fact as a *relational property*

5

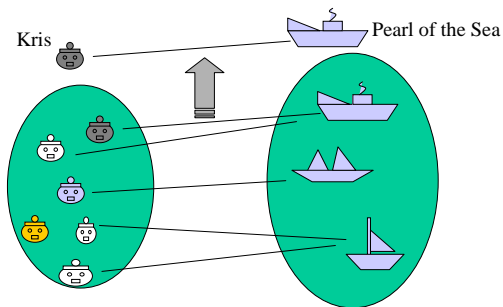
## Relation, Relationship Instances



© 2010 Griffith University

6

## Relationship Instance

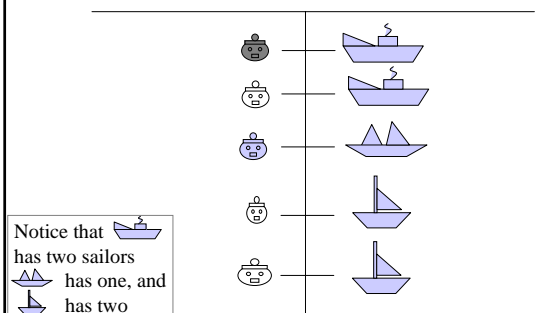


Example relationship instance or relationship) :  
Kris **serves on** the Pearl of the Sea

© 2010 Griffith University

7

## Relation, Relationship Instances

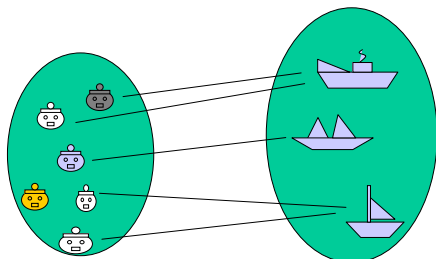


A relation is a set of relationship instances

© 2010 Griffith University

8

## Participation and cardinality constraints



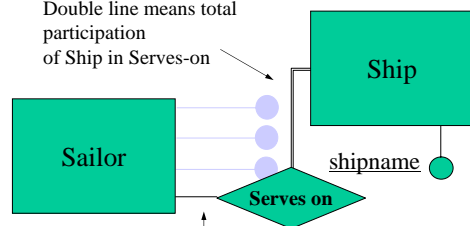
Some sailors do not serve on any ship (**partial** participation)

Every ship has at least one sailor (**total** participation)

© 2010 Griffith University

9

Double line means total participation of Ship in Serves-on



Single line means **partial** participation of Sailor in Serves-on

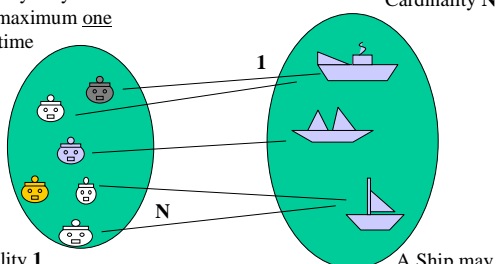
(Note: There are alternative representations)

© 2010 Griffith University

10

## Participation and cardinality constraints

A Sailor may only serve on maximum one Ship at a time



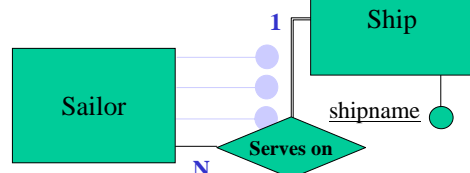
Cardinality 1

A Ship may have more than one Sailor serving on it

© 2010 Griffith University

11

We say that Serves on is an N:1 relationship



A Ship must have minimum one Sailor, but may have *several* (N) Sailors serving on it

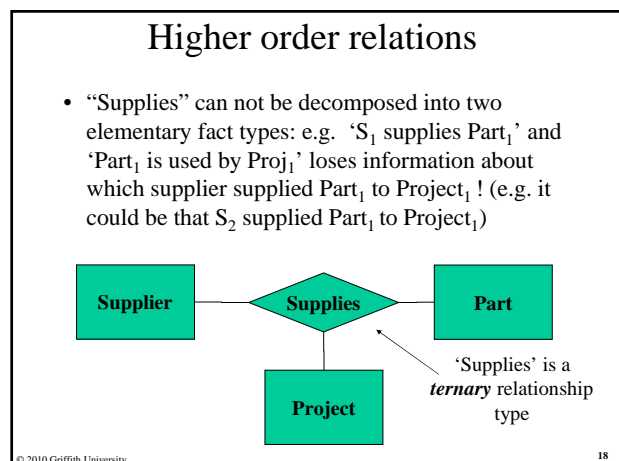
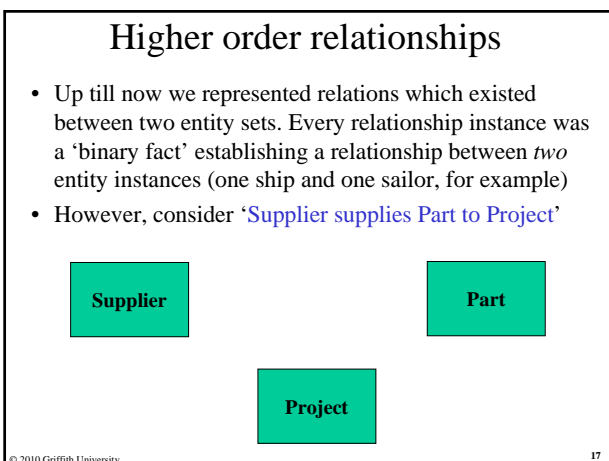
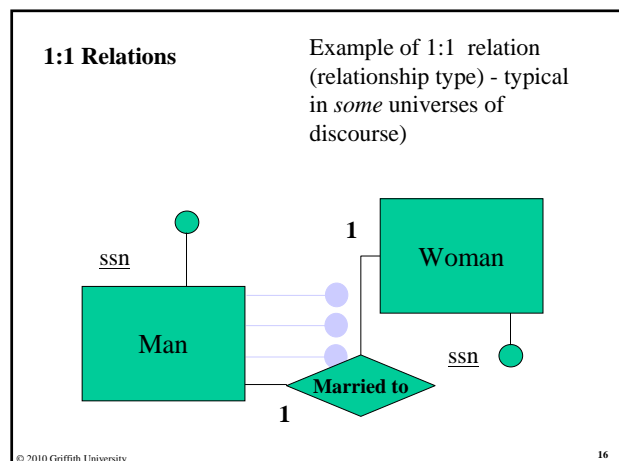
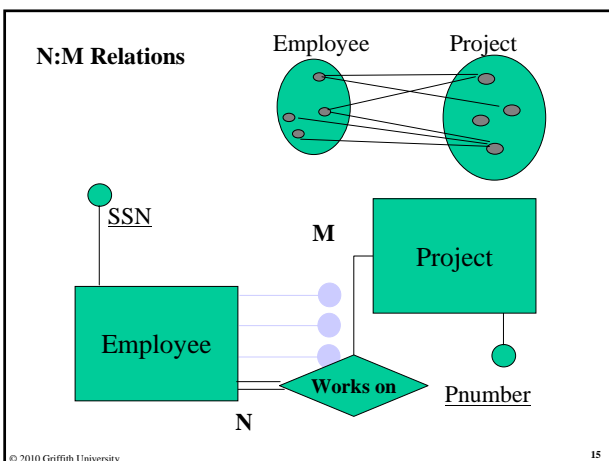
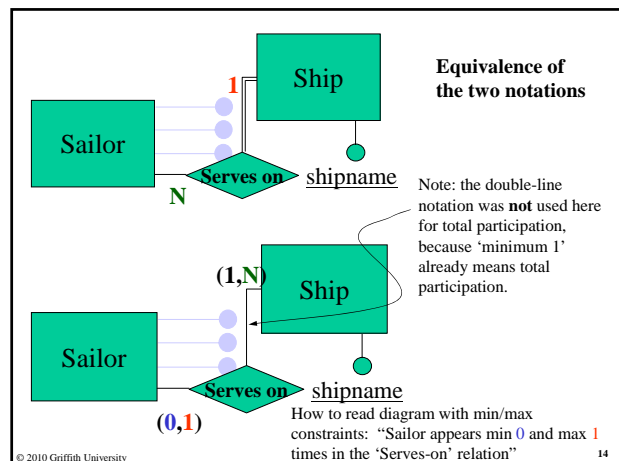
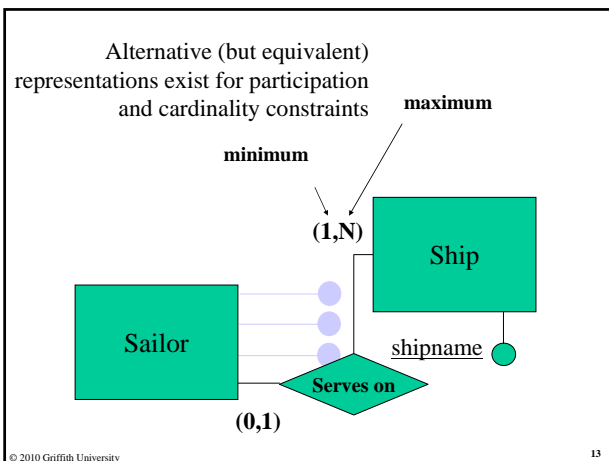
A Sailor may serve on a ship. A Sailor may not serve on more than *one* (1) ship at a time

How to read the diagram: Sailor - Serves on - 1 - Ship

Ship has - serving on it - N - Sailors

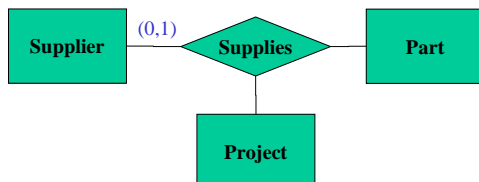
© 2010 Griffith University

12



### Representing constraints on higher order relations...

- Assume that if a supplier supplies a part to a project then it is not allowed to supply anything to any other project. Assume also that every project must have at least one part supplied to it. Parts may be represented in the database even if no project uses them at the moment.
- Represent the participation and cardinality constraints:

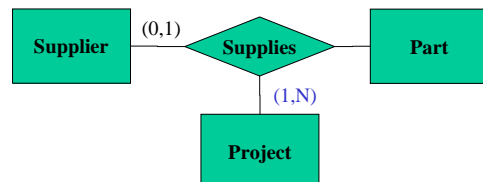


© 2010 Griffith University

19

### Representing constraints on higher order relations...

- Assume that if a supplier supplies a part to a project then it is not allowed to supply anything to any other project. Assume also that every project must have at least one part supplied to it, and Parts may be represented in the database even if no project uses them at the moment.
- Represent the participation and cardinality constraints:

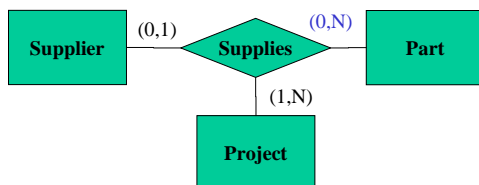


© 2010 Griffith University

20

### Representing constraints on higher order relations...

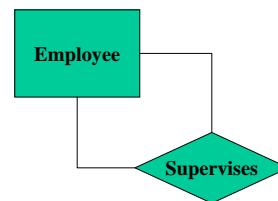
- Assume that if a supplier supplies a part to a project then it is not allowed to supply anything to any other project. Assume also that every project must have at least one part supplied to it, and Parts may be represented in the database even if no project uses them at the moment.
- Represent the participation and cardinality constraints:



© 2010 Griffith University

21

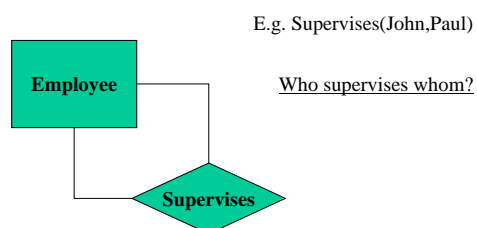
### Roles



© 2010 Griffith University

22

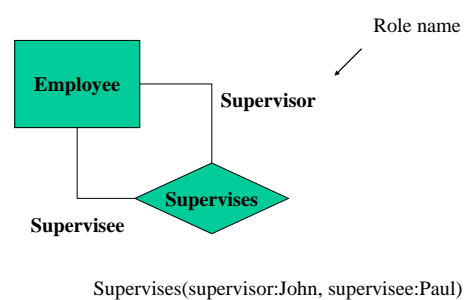
### Roles



© 2010 Griffith University

23

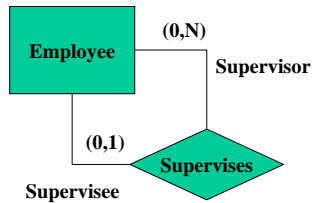
### Roles



© 2010 Griffith University

24

We represent constraints as with any other relation

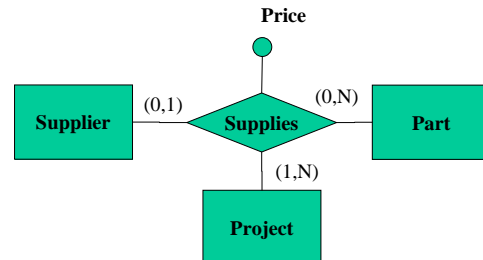


© 2010 Griffith University

25

## Attributes of relationship types

Price is not an attribute of any of the entities, it is an attribute of the relationship type "Supplies". (The price is determined for the 'deal')



© 2010 Griffith University

26

## Summary

- We have introduced the following concepts:
- Entity
- Attribute (simple, set valued, complex)
- Attribute domain
- Key (candidate, primary)
- Relationship
- Participation and cardinality constraints
- Binary, ternary, higher order relationship types
- Roles
- Attributes of relationship types

© 2010 Griffith University

27

The end

© 2010 Griffith University

28