Lecture 9

Mapping an Entity Relationship Schema to a Relational Schema (Enhanced Entity Relationship)

Week 5

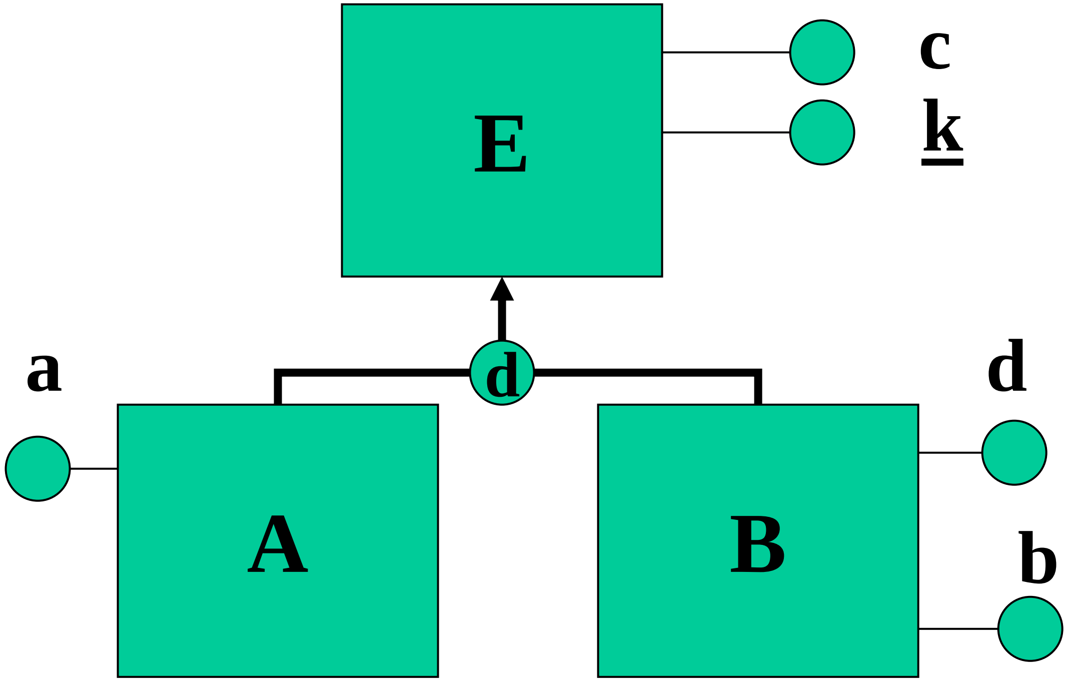
Overview

- Mapping generalisation hierarchies: there are several strategies; we shall study four
- The choice depends on the database population and usage patterns

© 2010 Griffith University

Choice A)

- **Map each entity to a separate schema**
- **Include primary key for each**
- **Include foreign key constraints**



E(<u>k</u>,c)

B(<u>k</u>,d,b) fk: k is k in E

A(<u>k</u>,a) fk: k is k in E

© 2010 Griffith University

E(k,c)

B(k,d,b) fk: k is k in E

A(k,a) fk: k is k in E

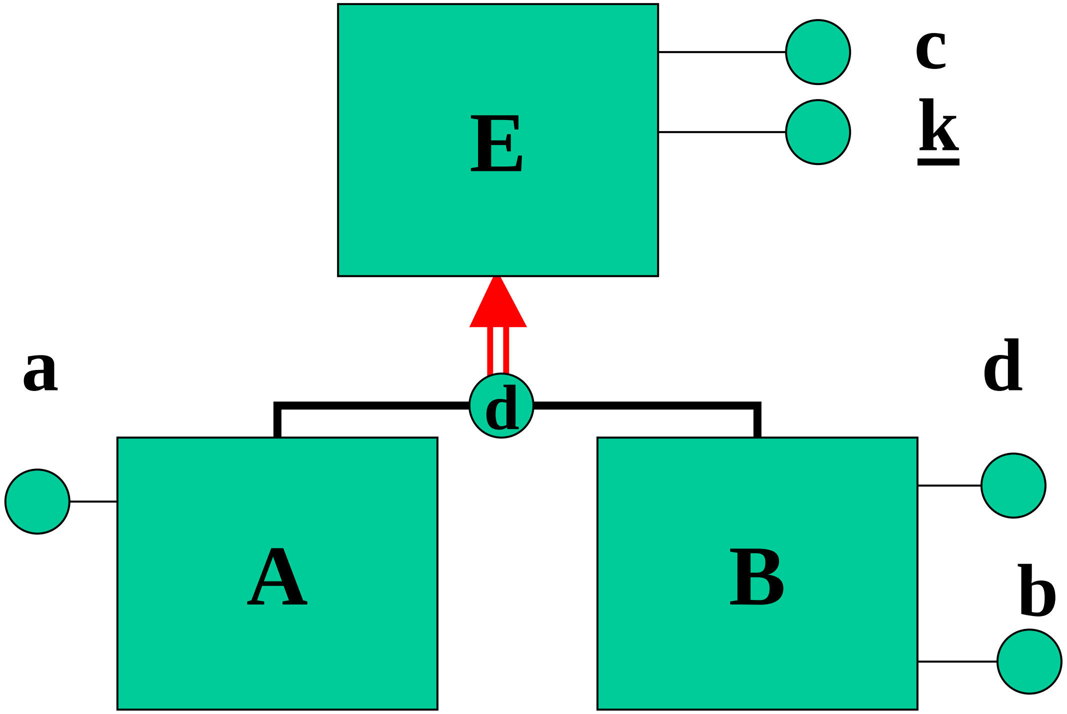
- Advantage: there are fewer null valued attributes (relative to the other mapping choices) and all instances of E (including As and Bs) are identifyable in one schema

- Disadvantage: information about one entity is fragmented, an instance of B is partly stored in E and partly in B - frequently querying all needs many joins

© 2010 Griffith University

Choice B)

- **If E is a total generalisation then map A and B separately**
- **Include disjointness constraint**



B(<u>k</u>,d,b,c) fk: k in B is not a k in A

A(<u>k</u>,a,c) fk: k in A is not a k in B

© 2010 Griffith University

B(k,d,b,c) k in B is not a k in A

A(k,a,c) k in A is not a k in B

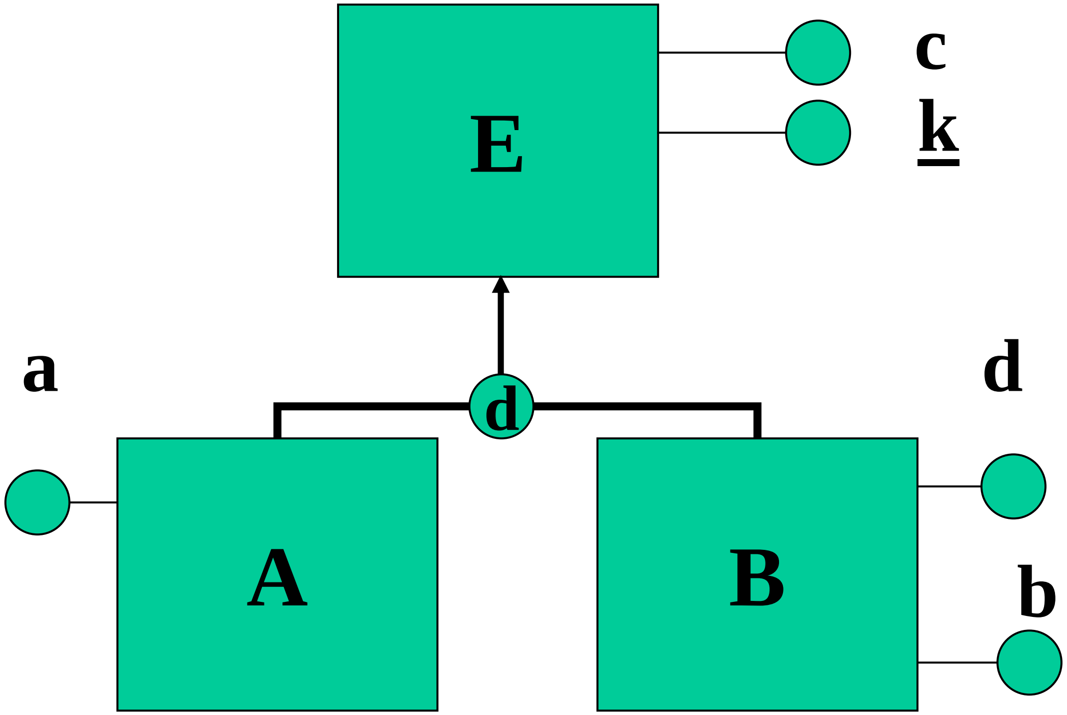
Disjointness constraints

- Advantage: non-redundant storage, all attributes of an entity are stored in one place
- Disadvantage: there is no single relation to retrieve the identifiers (primary keys) of all Es. (Basically the notion of E is not represented)

© 2010 Griffith University

Choice C)

- **If space is not a problem**
- **Must be disjoint types**



E(a, b, c, d, <u>k</u>, type)

where type is one of {‘A’, ‘B’, ‘E’}

© 2010 Griffith University

E(a, b, c, d, <u>k</u>, type)

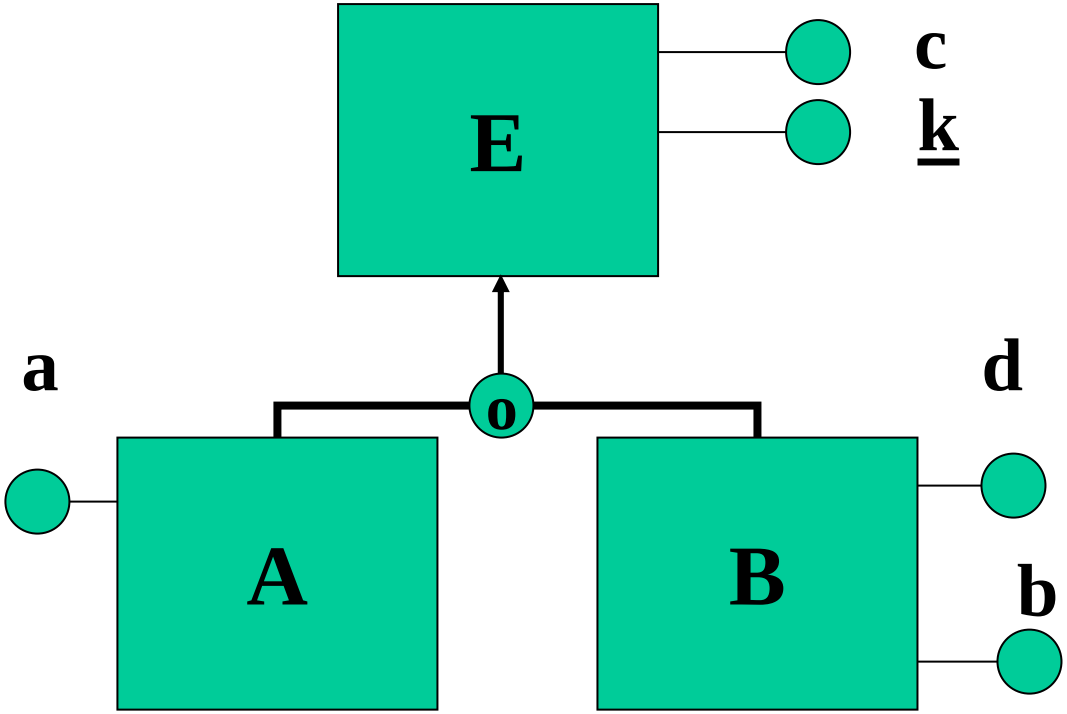
where type is one of {‘A’, ‘B’, ‘E’}

- Advantage: one schema for all instances of E,A and B
- Disadvantage: does not work for overlapping generalisations (although they are rare); always has null values (non-applicable attributes)

© 2010 Griffith University

Choice D)

- **Slight variant of choice C)**
- **Works for overlapping generalisations**



E(a, b, c, d, <u>k</u>)

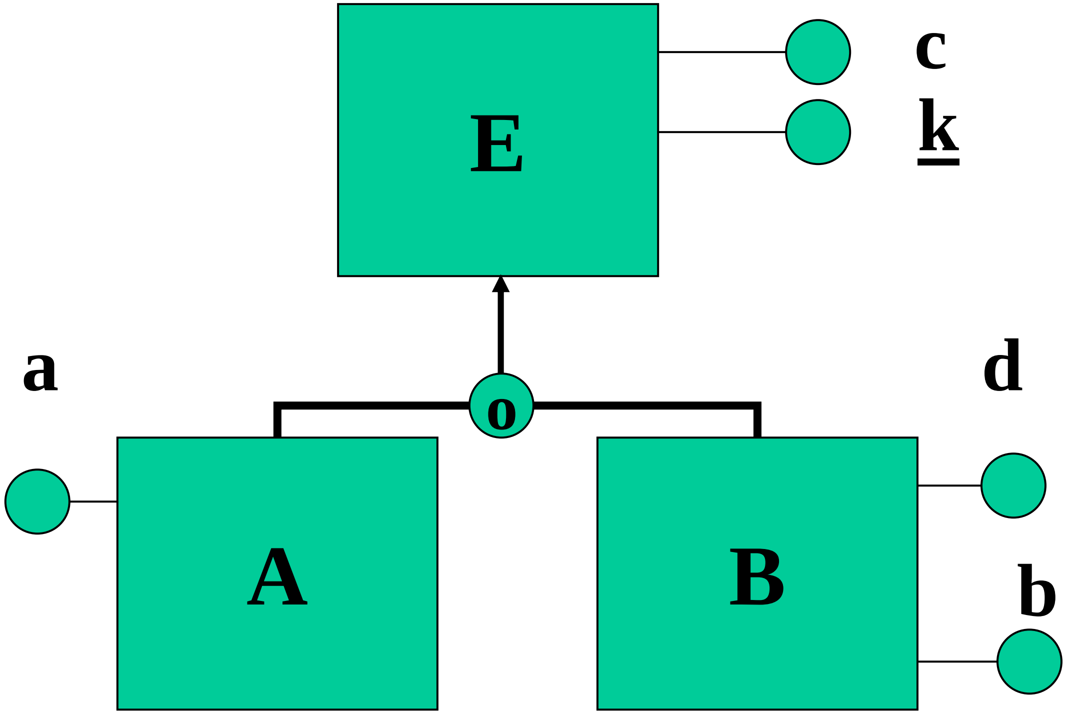
Not quite correct yet!

Loses information (e.g. how to interpret that the value for **a** is NULL?); therefore...

© 2010 Griffith University

Choice D)

- **Therefore…to rectify this problem we add two boolean attributes:** ***isA*** **and** ***isB***
- **Slight variant of choice C)**
- **Works for overlapping generalisations**

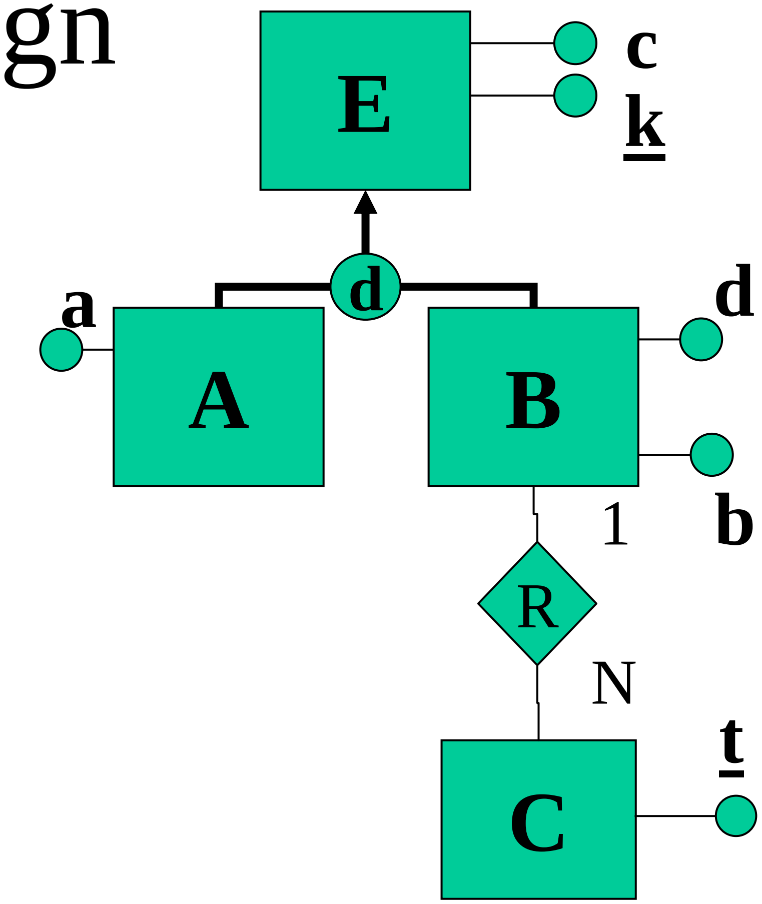


E(a, b, c, d, <u>k,</u> bool: isA, bool: isB**)**

Works also with null values

© 2010 Griffith University

Important note on foreign key constraints



- Suppose we mapped E, B, and A as follows
- Also we have R between C and B (mapped as an attribute of C).

E(k,c)

A(k,a) **fk: k is k in E**

B(k,d,b) **fk: k is k in E**

C(t,k) **fk: k is k in B**

- Notice the fk constraint on relation C! It says that ‘k’ must be an actual k-value in relation B. **It is *not* enough** for this k-value to be a k-value in relation E, because Cs are only related to Bs !

© 2010 Griffith University

Conclusion

- We propose that the first mapping choice is the cleanest one, because it has no redundancy

- However, if efficiency dictates, then one of the other three options may have to be considered

© 2010 Griffith University

The end

© 2010 Griffith University