Database Design and Practice 1
Lecture 06

Mapping the conceptual model into a logical model: essential guide!

Lecture 06 - Objectives

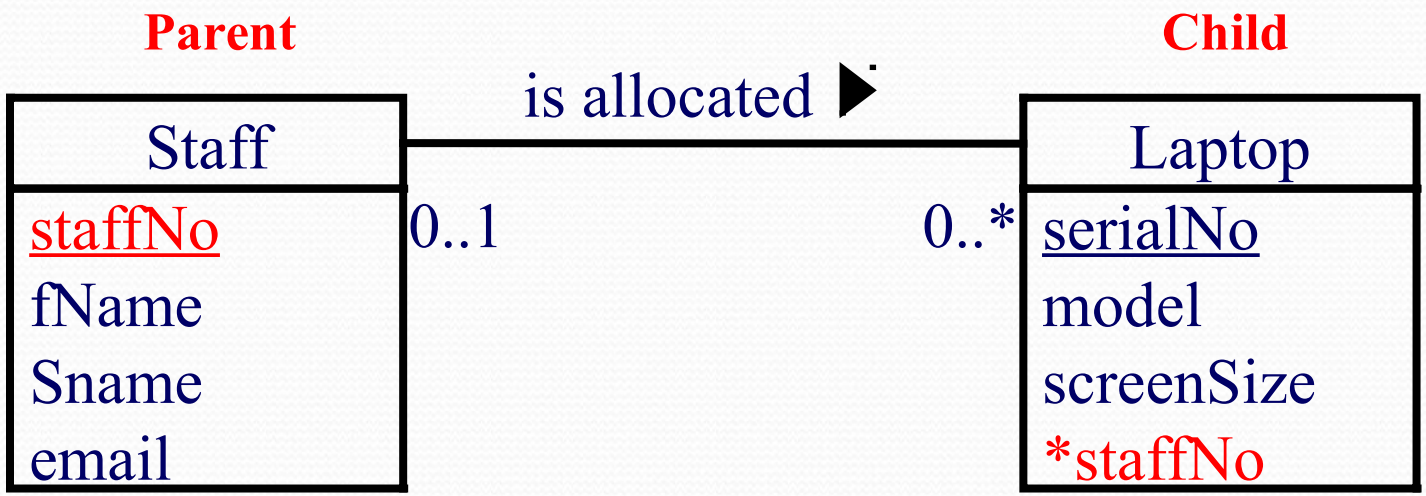
- How to derive a set of relations from a conceptual data model to a logical data model.
- How to resolve following relationships:
  - One-to-many
  - One-to-one mandatory on both sides
  - One-to-one optional on one side
  - One-to-one optional on both sides
  - Many-to-many
  - Complex relationships: ternary and quaternary
  - Generalisation with {Mantatory, And}
  - Generalisation with {Optional, And}
  - Generalisation with {Mandatory, Or}
  - Generalisation with {Optional, Or}

One-to-many (1:M) relationship

- Parent on “one” side, child on the “many” side
- Create FK on Child side which references PK on Parent side

Conceptual

Logical



Tables

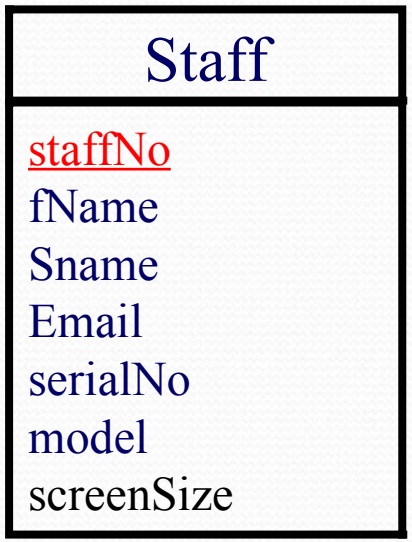
Staff (staffNo, fName, sName, email)
Laptop (serialNo, model, screenSize, *staffNo)

One-to-one (1:1) mandatory on both sides

- Merge two tables in one
- Choose one PK from the two PKs, other one is AK

Conceptual

Logical



Tables

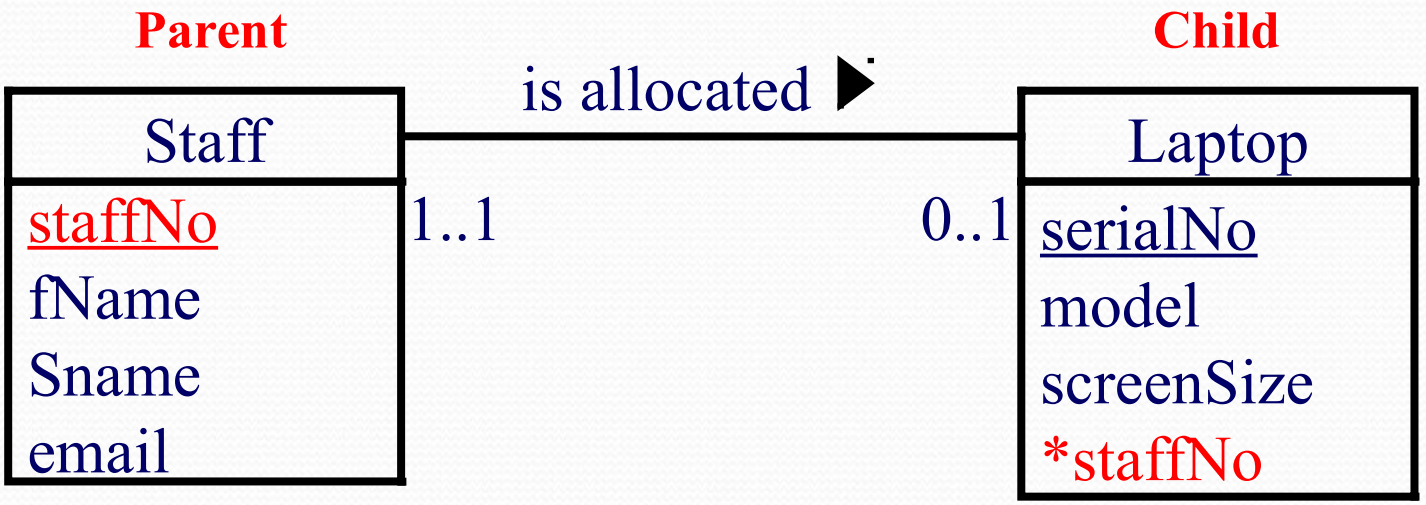
Staff (staffNo, fName, sName, email, serialNo, model, screenSize)

One-to-one (1:1) optional on one side

- Parent on “mandatory” side, child on the “optional” side
- Create FK on Child side which references PK on Parent side

Conceptual

Logical



Tables

Staff (staffNo, fName, sName, email)
Laptop (serialNo, model, screenSize, *staffNo)

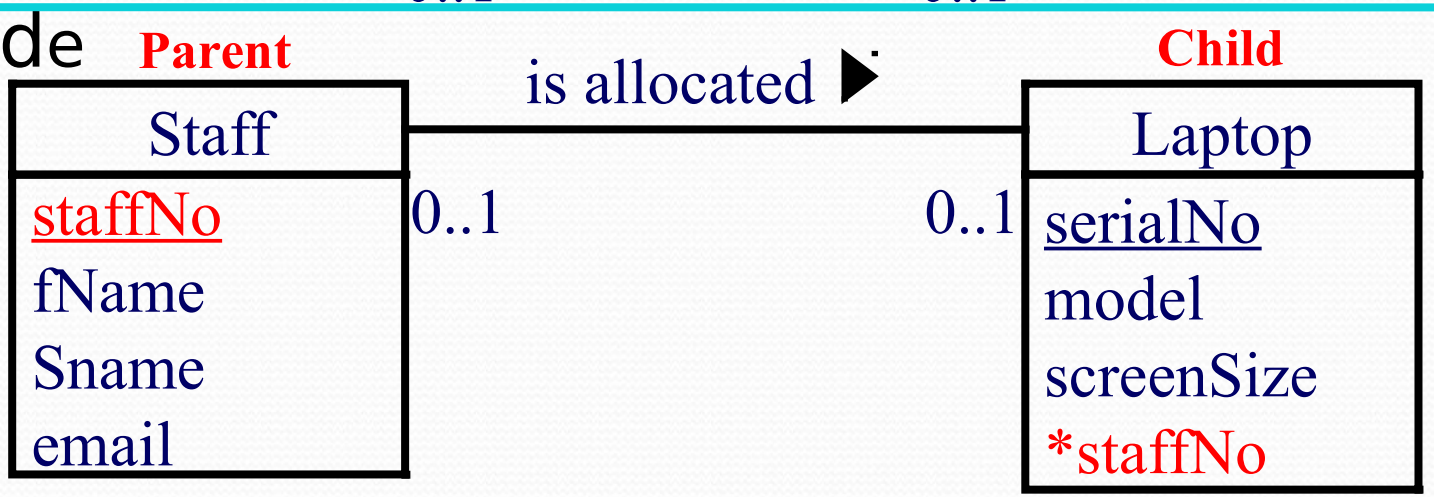
One to one (1:1) optional on both sides

- If no info, choose parent and choose child
- If info, which one is more mandatory than optional:

  Parent on “More mandatory”, Child on “more optional”
- Create FK on Child side which references PK on Parent side

Conceptual

Logical



Tables

Staff (staffNo, fName, sName, email)

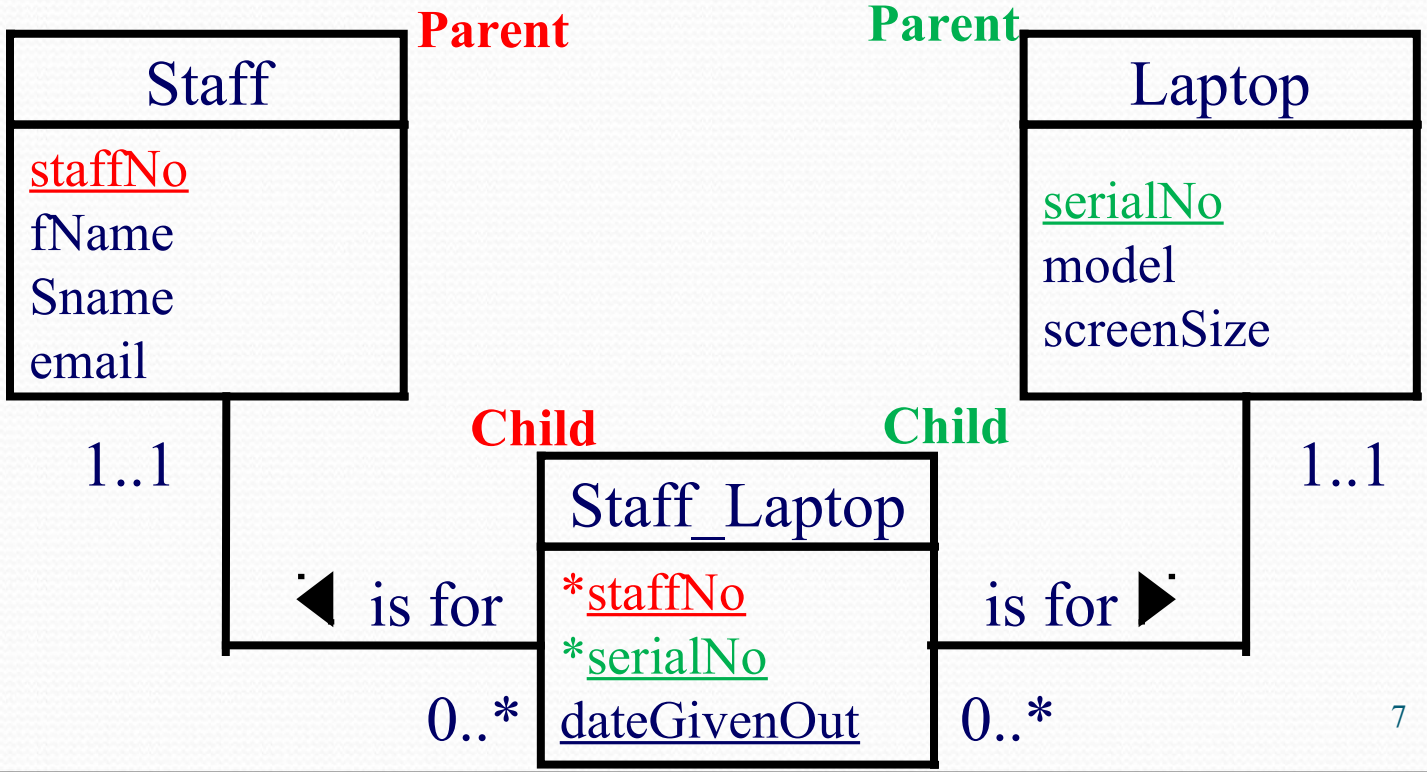
Laptop (serialNo, model, screenSize, *staffNo)

Many-to-Many

- Create a Link Table. Original tables have 1:M with new table.
- PK of Link Table: combination of other two PKs
- FKs of Link Table: reference PK of other two tables

Conceptual

Logical



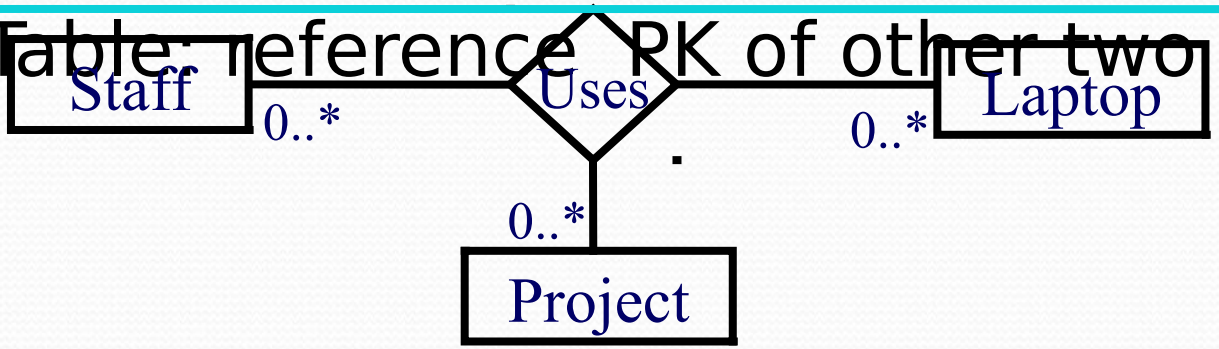
Many-to-Many (continued)

Tables	Staff (staffNo, fName, sName, email) Laptop (serialNo, model, screenSize) Staff_Laptop (*staffNo, *serialNo, dateGivenOut)

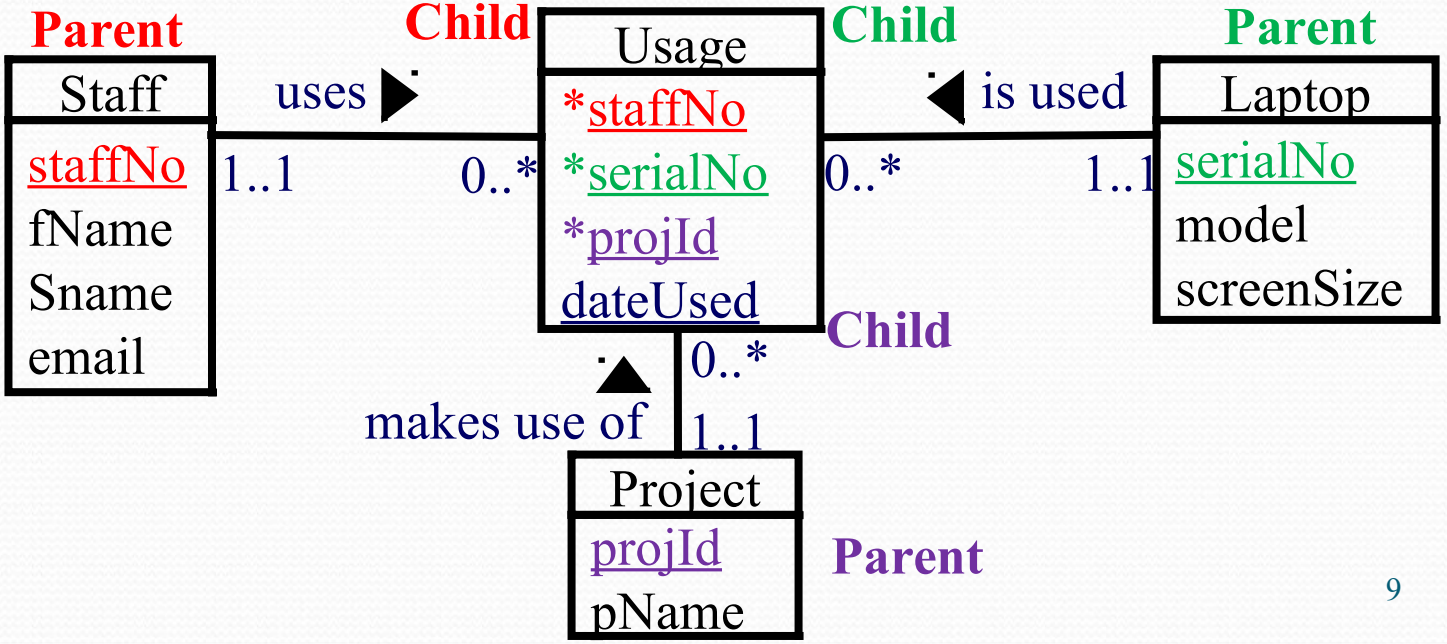
quaternary

- Create a Link Table. Original tables have 1:M with new table.
- PK of Link Table: combination of other two PKs
- FKs of Link Table: reference PK of other two tables

Conceptual



Logical



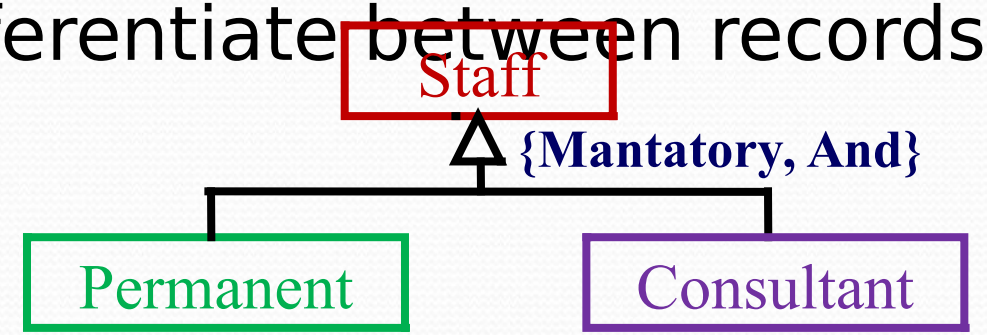
Complex relationships (continued)

Tables	Staff (staffNo, fName, sName, email) Laptop (serialNo, model, screenSize) Project (projId, pName) Usage(*staffNo, *serialNo, *projId, dateUsed)

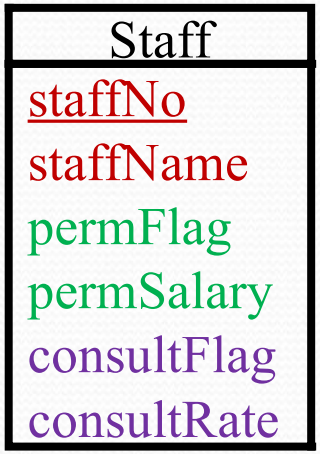
And}

- Create a single table which combines attributes of all entities.
- PK of single table: PK of general entity.
- Use flags to differentiate between records of ex-subentities.

Conceptual



Logical



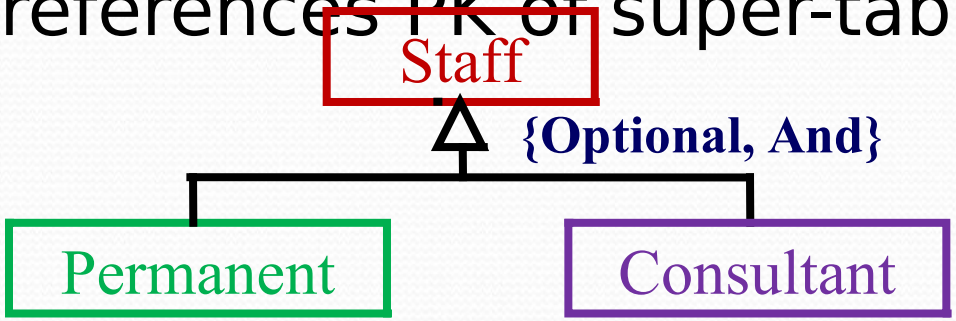
Tables

Staff (staffNo, staffName, permSalary, consultRate, permFlag, consultFlag)

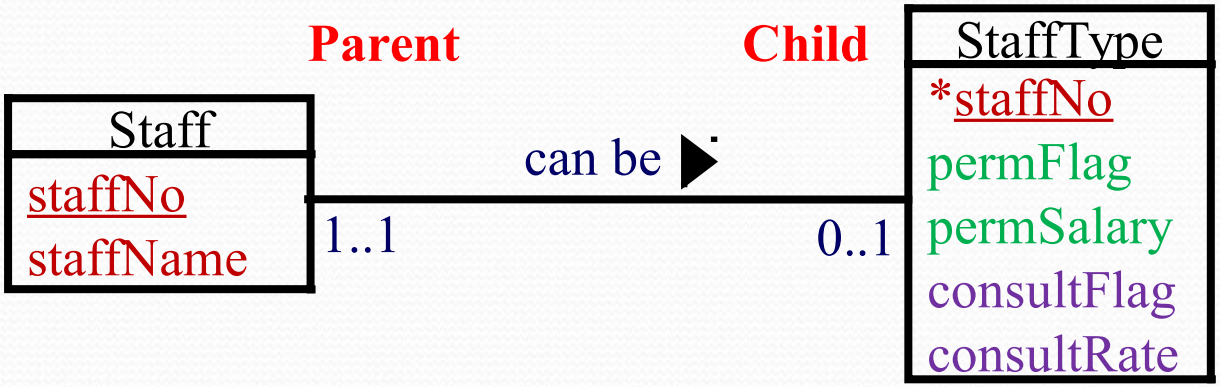
And}

- Create a table for super-entity & a table for both sub-entities.
- PK of both tables: PK of original super entity.
- FK of sub-table: references PK of super-table.

Conceptual



Logical



Tables

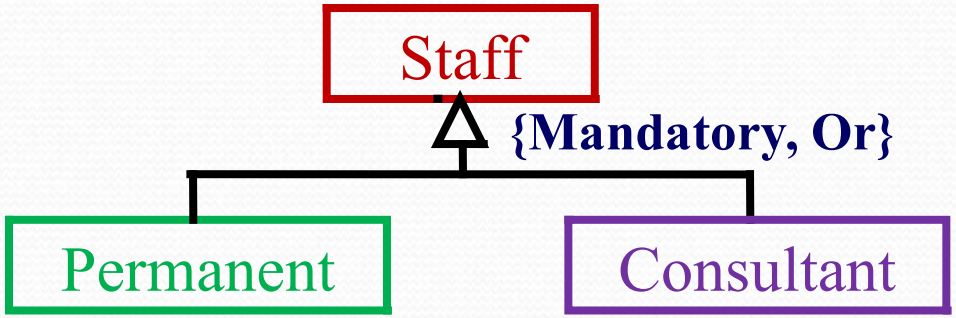
Staff (staffNo, staffName)

StaffDetails (*staffNo, permFlag, permSalary, consultFlag, consultRate)

Or}

- Create a table for each of the sub-entities.
- PK of both tables: PK of original super entity.
- Each table have own relationships with the rest of the schema.

Conceptual



Logical

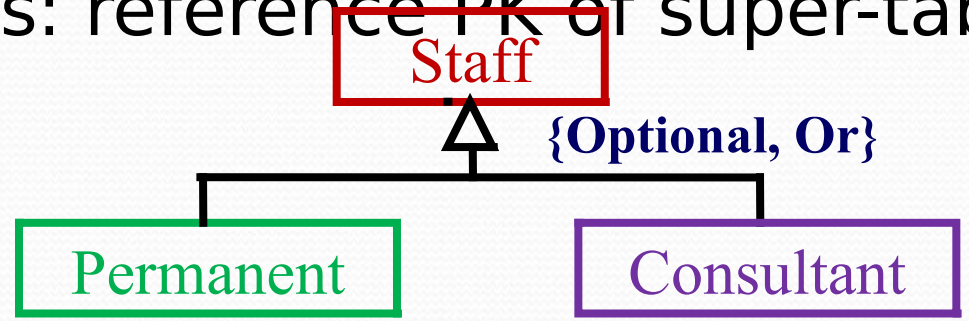
Tables

PermStaff (staffNo, staffName, permSalary)
ConsutlStaff(staffNo, staffName, consultRate)

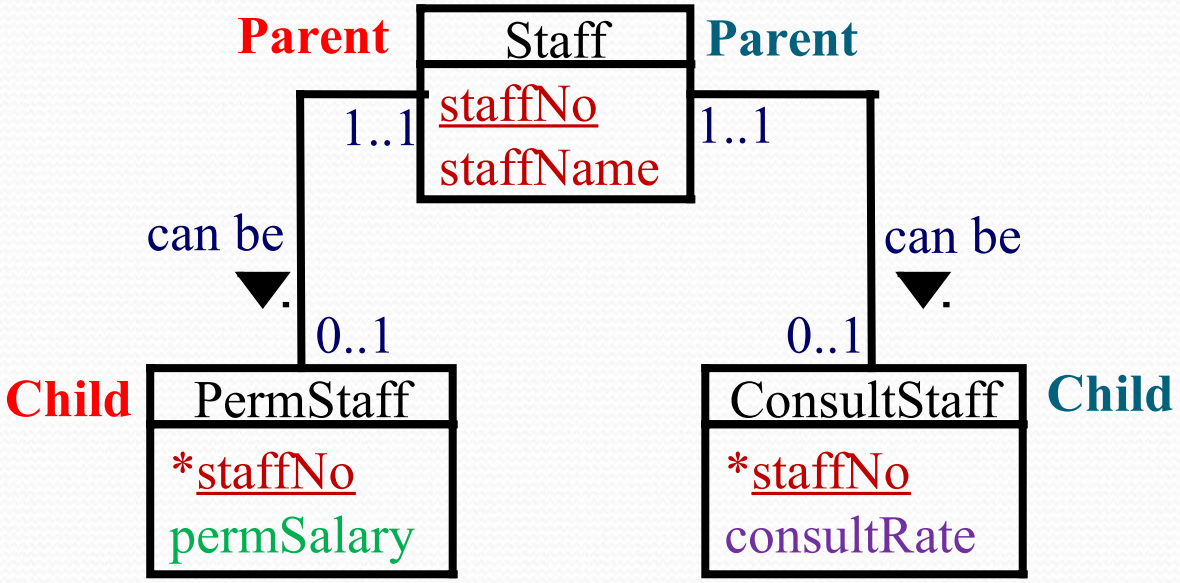
Or}

- Create a table for super-entity & 2 tables for sub-entities.
- PK of all 3 tables: PK of original super entity.
- FKs of sub-tables: reference PK of super-table.

Conceptual



Logical



Generalisation With {Optional, Or} (continued)

Tables	Staff (staffNo, staffName) PermStaff (*staffNo, permSalary) ConsutlStaff(*staffNo,consultRate)