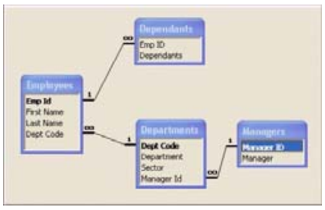
IT2201 / IT2601 / IT2564 / IT2621 / IT2521 / IT2323

Database Management Systems



Unit 5
Conceptual & Logical Database design

1

Unit Objectives

- At the end of this topic, you should be able to
  - Use ER modeling to build a conceptual data model
  - Derive a set of relations from a conceptual data model
  - Validate these relations

Mapping ERD to Relations

2

Database Design Methodology

We separate database design into three main phases:

➔ **Conceptual database design**

- The **process of constructing a model of the data used in an enterprise**, independent of *all* physical considerations.

➔ **Logical database design**

- The **process of constructing a model of the data used in an enterprise based on a specific data model**, but independent of a particular DBMS and other physical considerations.

3

Database Design Methodology

➔ **Physical database design**

- The **process of producing a description of the implementation of the database** on secondary storage; It describes the base relations, file organizations, and indexes used to achieve efficient access to the data, and any associated integrity constraints and security measures.

4

Database Design Methodology (Summary slide)

Conceptual database design

Step 1
Build conceptual data model

→

Logical database design

Step 2
Build logical data model

→

Physical database design

Step 3
Translate logical data model for target DBMS

Deliverables:

ER Model

Data dictionary

Deliverables:

Relational schema

Deliverables:

Base relations
Indexes
User views
Security mechanisms
De-normalized relations

5

Data dictionary, Relational Schema

- Example of a data dictionary showing documentation of entities and attributes.

~~Entity name~~	Attributes	Description	Data Type & Length	Nulls	Multi-valued	...
Staff	**staffNo**	Uniquely identifies a member of staff	5 variable characters	No	No	
	name					
	fName	First name of staff	15 variable characters	No	No	
	lName	Last name of staff	15 variable characters	No	No	
	position	Job title of member of staff	10 variable characters	No	No	
	sex	Gender of member of staff	1 character (M or F)	Yes	No	
	DOB	Date of birth of member of staff	Date	Yes	No	
PropertyForRent	**propertyNo**	Uniquely identifies a property for rent	5 variable characters	No	No	

- Relational Schema
  - Staff (staffNo, fName, lName, position, sex, DOB)
  - PropertyforRent (propertyNo, address, type, room, rent, ownerNo, staffNo)

FK FK

6

Conceptual Database Design

- Step 1: Build conceptual data model

  Steps

  1.1 Identify entity types
  1.2 Identify relationship types
  1.3 Identify and associate attributes with entity or relationship types
  1.4 Determine attribute domains
  1.5 Determine candidate, primary, and alternate key attributes
  1.6 Consider use of enhanced modeling concepts (optional step)
  1.7 Check model for redundancy
  1.8 Validate conceptual data model against user transactions
  1.9 Review conceptual data model with user

7

Step 1.7 Check model for redundancy

- Objective
  - To check for the presence of any redundancy in the model.
  - Tasks
    - Re-examine one-to-one (1:1) relationships
    - Remove redundant relationships
    - Consider time dimension

8

Step 1.8 Validate conceptual data model against user transactions

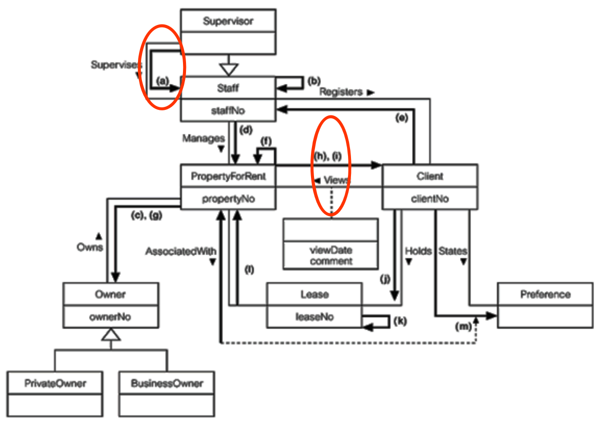
- Objective
  - To ensure that the conceptual data model supports the required transactions
  - Tasks
    - Use the data model and the data dictionary to perform the operations manually. This involves checking that
      - The required attributes are present in the data model.
      - Where attributes have to be taken from more than one entity, that there is a pathway between the two entities; that is, there is an identified relationship, either direct or indirect, between the two entities.

9

Check model supports user transactions

a) List details of staff supervised by a named supervisor at the branch.

b) List details of clients who had viewed the properties.

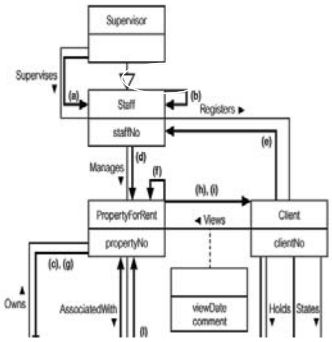


10

Exercise 1

Identify the transaction paths for the following transactions:

- List the details of comments made by clients viewing a given property
- List the clients registering at the branch and the names of the members of staff who registered the clients



11

Logical Database Design

- Step 2: Build logical data model

  Steps

  2.1 Derive relations for logical data model
  2.2 Validate relations using normalization
  2.3 Validate relations against user transactions
  2.4 Check integrity constraints
  2.5 Review logical data model with users

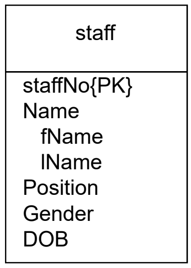
12

Step 2.1 Derive relations for logical data model

- Examine the structures present in the data model:

(1) Strong entity types

- Create a relation that includes all simple attributes of that entity. For composite attributes, include only constituent simple attributes.



Staff (staffNo, fName, lName, position, gender, DOB)
Primary Key staffNo

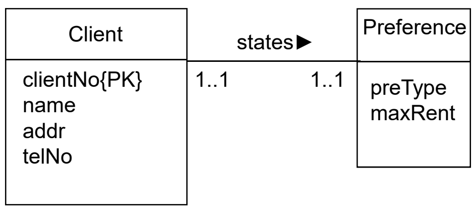
13

Step 2.1 Derive relations for logical data model

(2) Weak entity types

- Create a relation that includes all simple attributes of that entity.
- Primary key is partially or fully derived from each owner entity.

Preference (clientNo, prefType, maxRent)
Primary Key clientNo

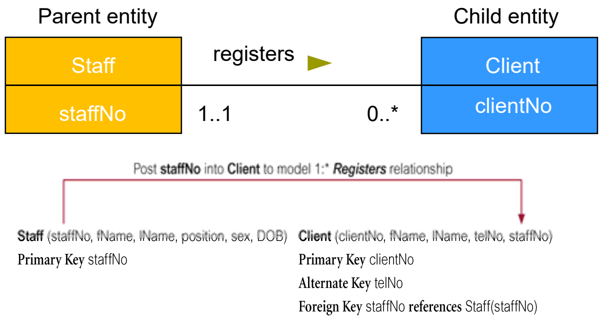


14

Step 2.1 Derive relations for logical data model

(3) 1:* binary relationship types

- Entity on 'one side' is designated the parent entity and entity on 'many side' is the child entity.
- Post copy of the primary key attribute(s) of parent entity into relation representing child entity, to act as a foreign key.



15

Step 2.1 Derive relations for logical data model

(4) 1:1 binary relationship types

- More complex as cardinality cannot be used to identify parent and child entities in a relationship.
- Instead, **participation used to decide whether to combine entities into one relation or to create two relations** and post copy of primary key from one relation to the other. Consider the following:
- *(a) mandatory* participation on *both* sides of 1:1 relationship;
- *(b) mandatory* participation on *one* side of 1:1 relationship;
- *(c) optional* participation on *both* sides of 1:1 relationship.

16

Step 2.1 Derive relations for logical data model

(a) *Mandatory* participation on *both* sides of 1:1 relationship

- **Combine entities involved into one relation** and choose one of the primary keys of original entities to be primary key of new relation, while other (if one exists) is used as an alternate key.

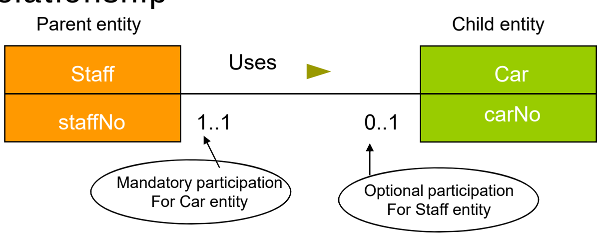
Columns of staff   Columns of car

StaffCar (<u>StaffNo</u>, name, position, salary, branchNo, supervisorStaffNo, CarNo, make, model)
Primary Key staffNo
Alternate key carNo

17

Step 2.1 Derive relations for logical data model

(b) *Mandatory* participation on *one* side of a 1:1 relationship



staffNo is posted to represent the *Uses* relationship

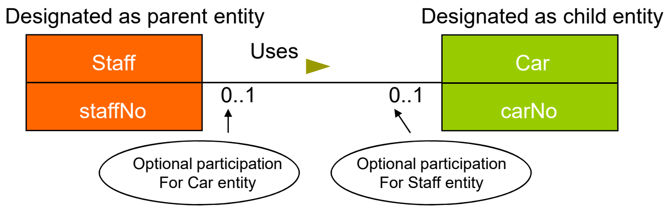
Staff (<u>staffNo</u>, name, position, salary, branchNo, supervisorStaffNo)
Primary Key staffNo
Foreign key supervisorStaffNo **references** staff(staffNo)

Car (<u>carNo</u>, make, model, staffNo)
Primary Key carNo
Alternate key staffNo
Foreign key staffNo **references** staff(staffNo)

18

Step 2.1 Derive relations for logical data model

(c) *Optional* participation on *both* sides of a 1:1 relationship (Choice of parent entity is arbitrary)



staffNo is posted to represent the *Uses* relationship

Staff (<u>staffNo</u>, name, position, salary, branchNo, supervisorStaffNo)
Primary Key staffNo
Foreign key supervisorStaffNo **references** staff(staffNo)

Car (<u>carNo</u>, make, model, staffNo)
Primary Key carNo
Alternate key staffNo
Foreign key staffNo **references** staff(staffNo)

19

Step 2.1 Derive relations for logical data model

(5) Superclass/subclass relationship types

- Identify superclass as parent entity and subclass entity as child entity.
- There are various options on how to represent such a relationship as one or more relations.
- Most appropriate option dependent on number of factors such as:
  - disjointness and participation constraints on the superclass/subclass relationship,
  - whether subclasses are involved in distinct relationships,
  - number of participants in superclass/subclass relationship.

20

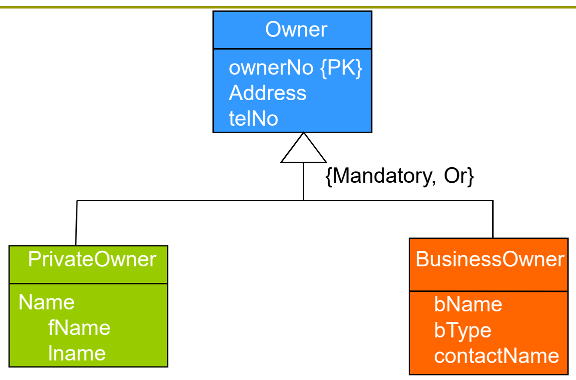
Guidelines for Representation of Superclass / Subclass Relationship

Table 15.1 Guidelines for the representation of a superclass/subclass relationship based on the participation and disjoint constraints.

Participation constraint	Disjoint constraint	Relations required
Mandatory	Nondisjoint {And}	Single relation (with one or more discriminators to distinguish the type of each tuple)
Optional	Nondisjoint {And}	Two relations: one relation for superclass and one relation for all subclasses (with one or more discriminators to distinguish the type of each tuple)
Mandatory	Disjoint {Or}	Many relations: one relation for each combined superclass/subclass
Optional	Disjoint {Or}	Many relations: one relation for superclass and one for each subclass

21

Examples of Superclass / Subclass Relationship

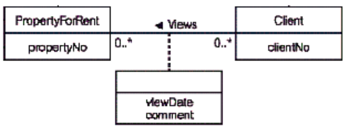


PrivateOwner (ownerNo, fName, lName, address, telNo)
Primary key ownerNo
BusinessOwner (ownerNo, bName, btype, contactName, address, telNo)
Primary Key ownerNo

22

Step 2.1 Derive relations for logical data model

(6) *:* binary relationship types - Example



Client (clientNo, fName, lName, telNo, prefType, maxRent, staffNo)

Primary Key clientNo **Foreign Key** staffNo **references** Staff(staffNo)

PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent)

Primary Key propertyNo

Viewing (clientNo, propertyNo, dateView, comment)
Primary Key clientNo, propertyNo
Foreign Key clientNo **references** Client(clientNo)
Foreign Key propertyNo **references** PropertyForRent(propertyNo)

23

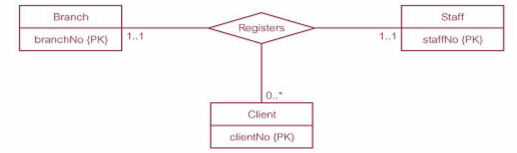
(6) *:* binary relationship types – cont'd

- For each *:* binary relationship, create a relation to represent the relationship (Views) and include any attributes that are part of relationship.
- Post a copy of the primary key attribute(s) of the entities that participate in relationship into new relation, to act as foreign keys.
- These foreign keys will also form primary key of the new relation, possibly in combination with some of the attributes of the relationship.

24

Step 2.1 Derive relations for logical data model

(7) Complex relationship types



Staff (staffNo, fName, lName, position, sex, DOB, supervisorStaffNo)
Primary Key staffNo
Foreign Key supervisorStaffNo **references** Staff(staffNo)
Branch (branchNo, street, city, postcode)
Primary Key branchNo
Client (clientNo, fName, lName, telNo, prefType, maxRent, staffNo)
Primary Key clientNo
Foreign Key staffNo **references** Staff(staffNo)

Registration (clientNo, branchNo, staffNo, dateJoined)
Primary Key clientNo
Foreign Key branchNo **references** Branch(branchNo)
Foreign Key clientNo **references** Client(clientNo)
Foreign Key staffNo **references** Staff(staffNo)

25

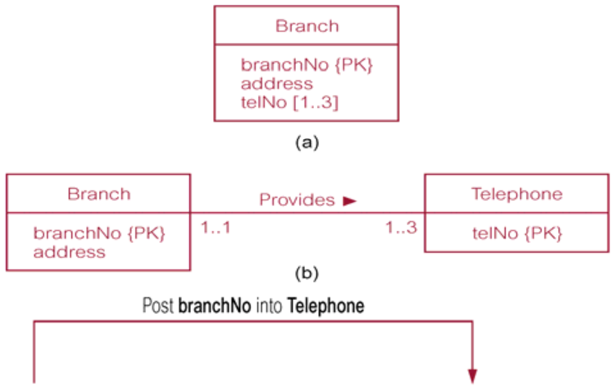
(7) Complex relationship types – cont'd

- For each complex relationship, create a relation to represent the relationship and include any attributes that are part of the relationship.
- Post a copy of the primary key attribute(s) of entities that participate in the complex relationship into the new relation, to act as foreign keys.

26

Step 2.1 Derive relations for logical data model

(8) Multi-valued attributes - Example



Branch (branchNo, street, city, postcode)
Primary Key branchNo

Telephone (telNo, branchNo)
Primary Key telNo
Foreign Key branchNo **references** Branch(branchNo)

27

(8) Multi-valued attributes – cont'd

- For each multi-valued attribute in an entity, create a new relation to represent the multi-valued attribute and include the primary key of the entity in the new relation, to act as a foreign key.

28

Summary of How to Map Entities and Relationships to Relations

Table 15.2 Summary of how to map entities and relationships to relations.

Entity/Relationship	Mapping
Strong entity	Create relation that includes all simple attributes.
Weak entity	Create relation that includes all simple attributes (primary key still has to be identified after the relationship with each owner entity has been mapped).
1:* binary relationship	Post primary key of entity on 'one' side to act as foreign key in relation representing entity on 'many' side. Any attributes of relationship are also posted to 'many' side.
1:1 binary relationship:	
(a) Mandatory participation on both sides	Combine entities into one relation.
(b) Mandatory participation on one side	Post primary key of entity on 'optional' side to act as foreign key in relation representing entity on 'mandatory' side.
(c) Optional participation on both sides	Arbitrary without further information.
Superclass/subclass relationship	See Table 15.1.
: binary relationship, complex relationship	Create a relation to represent the relationship and include any attributes of the relationship. Post a copy of the primary keys from each of the owner entities into the new relation to act as foreign keys.
Multi-valued attribute	Create a relation to represent the multi-valued attribute and post a copy of the primary key of the owner entity into the new relation to act as a foreign key.

29

Step 2.2 Validate relations against normalization

- Validate the relations in the logical data model using normalization
- Ensure that the relations have minimal data redundancy to avoid the problems of update anomalies

30

Step 2.3 Validate relations against user transactions

- Ensure that the relations support the transactions.
- Ensure that no error has been introduced while creating relations.
- Perform the operations manually using the relations, the primary key/foreign key, the ER diagram, and the data dictionary.

31

Step 2.4 Define integrity constraints

- We consider 5 types of integrity constraints
  - Required data – No null value allowed
  - Attribute Domain Constraints – Set of allowable values
  - Entity integrity – Primary key of an entity cannot hold nulls
  - Referential integrity – value of foreign key must refer to an existing tuple in the parent relation
  - Enterprise constraints – business rules

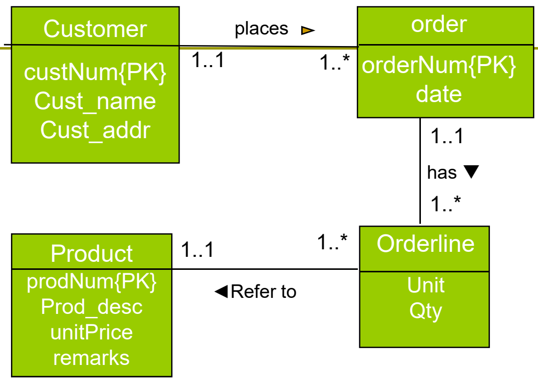
32

Step 2.5 Review logical data model with user

- To ensure that the logical data model and supporting documentation that describes the model is a true representation of the data requirements of the enterprise.

33

Exercise 2 - Derive relational schema for the following ERD.



34

Summary

- The database design methodology includes three main phases: conceptual, logical and physical database design.
- **Logical database design** is the **process of constructing a model of the data used in an enterprise** based on a specific data model, but independent of a particular DBMS and other physical considerations.
- **A logical data model includes ER diagram(s), relational schema, and supporting documentation**, such as the data dictionary, which is produced throughout the development of the model.
- The **main steps** of the logical database design methodology for the relational model include: **building logical data model** and **deriving and validating relations.**

35

Reference Materials

1. Database Systems, Connolly, Ch 16, 17
2. Database Solutions, Connolly, Ch 7-10

36