
Database Systems 2

Lecture 12

Database Design Methodology

Logical Design 2

Based on Connolly & Begg: Database Systems (3 ed)

Overview of the Methodology

Logical Database Design

- 2 Build and validate local logical data model for each user view
- 3 Build and validate global logical data model

Logical Database Design

3 Build and validate global logical data model

- 3.1 Merge local logical data models into global model
- 3.2 Validate global logical data model
- 3.3 Check for future growth
- 3.4 Draw final Entity-Relationship diagram
- 3.5 Review global logical data model with users

Step 3.1 – Merge Local data models into Global Model

- 3.1.1 Review names of entities and their primary keys
- 3.1.2 Review the names of the relationships
- 3.1.3 Merge entities from the local views
- 3.1.4 Include (without merging) entities unique to each local view
- 3.1.5 Merge relationships from the local views
- 3.1.6 Include (without merging) relationships unique to each local view
- 3.1.7 Check for missing entities and relationships
- 3.1.8 Check foreign keys
- 3.1.9 Check integrity constraints
- 3.1.10 Draw the global ERD
- 3.1.11 Update the documentation

3.1.1 Review names of entities and keys

List the entities and primary keys from each model and look for overlaps.

Supervisor	Manager
Property_For_Rent	Property_For_Rent
Owner (Sub) Private_Owner (Sub) Business_Owner	Owner (Sub) Business_Owner
Lease_Agreement	Rental_Agreement
Inspection	

3.1.2 Review names of relationships

List the entities and relationships from each model and look for matches.

Supervisor			Manager		
Entity	Relationship	Entity	Entity	Relationship	Entity
Staff	takes part in	Staff_Property_Link	Staff	takes part in	Overseer
	carries out	Inspection		assigned to	Allocated_Staff
	joins	Allocated_Staff		related to	Next_of_Kin
Supervisor	supervises	Allocated_Staff	Supervisor	supervises	Allocated_Staff

3.1.2 Review names of relationships

Look out for entities or relationships that have the same name but represent different concepts (homonyms)

Look out for entities or relationships that have different names but which represent the same concept (synonyms).

Compare the attributes, especially the keys, of each entity and their relationships with other entities. Also the domains of each attribute.

Be aware that entities or relationships in one view may be represented simply as attributes in another view.

Eg The Branch entity may have an attribute called Manager_Name in one view, which maps onto an entire entity called Manager in another view.

3.1.3 Merge entities from the local views

(Manager's View)

Staff (Staff_No, FName, LName, Address, Tel_No, Sex, DOB (Date_of_Birth),
Position, **Salary**, **Date_Joined**, **NIN (National Insurance Number)**,
Typing_Speed, Branch_No)

Primary Key Staff_No

Alternate Key NIN

Foreign Key Branch_No **references** Branch(Branch_No)

(Supervisor's View)

Staff (Staff_No, FName, LName, Address, Tel_No, Sex, DOB (Date_of_Birth),
Job_Title, Typing_Speed, Branch_No)

Primary Key Staff_No

Foreign Key Branch_No **references** Branch(Branch_No)

(Global View)

Staff (Staff_No, FName, LName, Address, Tel_No, Sex, DOB (Date_of_Birth),
Position, Salary, Date_Joined, NIN (National Insurance Number),
Typing_Speed, Branch_No)

Primary Key Staff_No

Alternate Key NIN

Foreign Key Branch_No **references** Branch(Branch_No)

Merge entities with the same name and same primary keys

- Easy

Merge entities with the same name and different primary keys

- Choose one of the two candidate keys for the merged entity

Merge entities with different names

- Do the entities have the same purpose?
- If they have the same primary key, that is a good clue
- Check with the users.

3.1.4 Include (without merging) entities unique to each local view

Easy –

But on a large scale project, all of these would need to be documented carefully.

- List of shared entities.
- List of distinct entities.
- List of possibly duplicated entities for further analysis.

Merge Entities Table

Decide on Global Entities

Supervisor	Manager		Global
Property_For_Rent	Property_For_Rent		
Owner (Sub) Private_Owner (Sub) Business_Owner	Owner (Sub) Business_Owner		
Lease_Agreement	Rental_Agreement		
Inspection			

3.1.5 Merge relationships from the local views

Merge relationships with same name and same purpose

- Easy, but ensure that the cardinality and optionality match.
- What do you do if they don't?
- Beware relationships with same name and different purpose.

Merge relationships with different names but the same purpose.

- If they link the same entities, with the same cardinality and optionality, are they actually the same relationship?
- What do you do if you aren't sure?

3.1.6 Include (without merging) relationships unique to each local view

Easy

Really quite an easy step

Merge Relationships Table

List the entities and relationships from each model and look for matches.

Supervisor			Manager			Global		
Entity	Relationship	Entity	Entity	Relationship	Entity	Entity	Relationship	Entity
Staff	takes part in	Staff_Property_Link	Staff	takes part in	Overseer			
Staff	carries out	Inspection	Staff	assigned to	Allocated_Staff			
Staff	joins	Allocated_Staff	Staff	related to	Next_of_Kin			
Supervisor	supervises	Allocated_Staff	Supervisor	supervises	Allocated_Staff			

3.1.7 Check for missing entities and relationships

This, on the other hand, is really difficult.

How can entities and relationships get left out of all the local views?

Where does it usually happen?

No hard and fast rules – requires a detailed knowledge of the enterprise being modelled.

It is likely that gaps in the data model will become apparent at a later stage.

3.1.8 Check foreign keys

3.1.9 Check integrity constraints

Check that all child entities still contain the correct foreign keys and integrity constraints.

Which entities should you pay particular attention to?

3.1.10

Draw the global ERD

3.1.11

Update the documentation

Make sure you have an up to date ERD and Data Dictionary.

If they are not right, all of the future work you do on this database might have to be redone.

Logical Database Design

3 Build and validate global logical data model

3.1 Merge local logical data models into global model

3.2 Validate global logical data model

3.3 Check for future growth

3.4 Draw final Entity-Relationship diagram

3.5 Review global logical data model with users

Step 3.2 - Validate global logical data model

How is this done?

The same way as we validated the local models.

Use normalisation

Map the transactions onto the ERD

Step 3.3 - Check for future growth

How can you do this?

Talk to the (eventual) users of the database?

Talk to the managers of the company?

Always allow for some expansion anyway.

Step 3.4 - Draw final global ERD

Incorporating any changes made as a result of validating the model, and allowing for future growth.

Step 3.5 - Review global logical data model with users

Make sure that all the user groups sign off on the global data model.