Database Design

Methodology

Database Design Methodology

A structured approach that uses procedures, techniques, tools, and documentation aids to support and facilitate the process of design

Database Design Methodology

Three phases

Conceptual database design

Logical database design

Physical database design

Constructing a model of the information used in an enterprise, independent of *all* physical considerations

Database Design Methodology

Three phases

Conceptual database design

Logical database design

Physical database design

Constructing a model of the information used in an enterprise on a specific data model but independent of a particular DBMS and other physical considerations

Database Design Methodology

Three phases

Conceptual database design

Logical database design

Physical database design

Producing a description of the implementation of the database on secondary storage

Critical Success Factors

Work interactively with users	Use diagrams for the data model
Follow a structured methodology	Use a database design language
Employ a data driven approach	Build a data dictionary
Incorporate structural and integrity considerations in the model	
Be willing to repeat steps	

Conceptual database design

Logical database design

Physical database design

Conceptual database design

Logical database design

Physical database design

Conceptual database design

Logical database design

Physical database design

Build local conceptual model for each user view

Conceptual database design

Logical database design

Physical database design

Build and validate local logical data model

Conceptual database design

Logical database design

Physical database design

Build local conceptual data model for each user view

1.1	Identify entity types
1.2	Identify relationships
1.3	Identify and associate attributes with entity and relationship types
1.4	Determine attribute domains
1.5	Determine SK/CK/PK attributes
1.6	Specialise/generalise entity types (optional)
1.7	Check model for redundancy
1.8	Validate model against user transactions
1.9	Review data model with user

Conceptual database design

Logical database design

Physical database design

Build and Validate Local Logical Data Model

2.1	Remove features not compatible with the relational data model
2.2	Derive relations
2.3	Validate relations using normalisation
2.4	Validate relations against user transactions
2.5	Define integrity constraints
2.6	Review local logical data model with user

Conceptual database design

Logical database design

Physical database design

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	Review logical data Models with users

Conceptual database design

Logical database design

Physical database design

Translate Global LDM for target DBMS

4.1	Design Base Relations for Target DBMS
4.2	Derive representations of derived data
4.3	Design Integrity Rules for Target DBMS

Design Physical Representation

5.1	Analyze Transcations
5.2	Choose File Organisations
5.3	Choose Indexes
5.4	Estimate Disk Space

6	Design User Views
7	Design security mechanisms
8	Consider redundancy
9	Monitor and fine tune

Conceptual database design

Logical database design

Physical database design

Conceptual database design

1.1 Identify entity types

Identify and document the main entity types in users view

Look for objects, nouns or noun phrases

Customer	Invoice	Address
Salesperson	Vehicle	Staffmember

Conceptual database design

Logical database design

Physical database design

Producing a description of the implementation of the database on secondary storage

Conceptual database design

1.1 Identify entity types

Do not confuse with relationship types

Manager

Do not confuse with attributes

Marriage

Watch out for synonyms (same meaning)

Office Branch

Conceptual database design

1.1 Identify entity types

Could use a data dictionary

Repository of defined common terms

Entity Name	Description	Aliases	Occurance
Employee	General term for all employees in the system	staff_member	Each employee works in one particular branch
Branch	Term for each physical site on a highstreet	shop	There are 12 branches

Conceptual database design

1.2 Identify relationships

Determine the multiplicity of the relationships

Entity Name	Multiplicity	Description	Aliases	Occurance	Relationship	Entity Name	Multiplicity
Employee	1..1	General term for all employees in the system	staff_member	Each employee works in one particular branch	works at	Branch	1..8
Branch	1..1	Term for each physical site on a highstreet	shop	Each branch is in a geographical region	situated in	Region	1..*

Conceptual database design

1.2 Identify relationships

Identify and document the relationships that exist between the entity types previously identified

Employee Works at Branch

Conceptual database design

1.2 Identify relationships

Generally use a modelling technique

Current standard is UML (Unified Modelling Language)

Many different model types in UML

Class diagrams Close to old style entity-relationship diagrams

Conceptual database design

1.2 Identify relationships

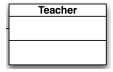
Could redraw the data dictionary at this point to include the relationships

Entity Name	Description	Aliases	Occurance	Relationship	Entity Name
Employee	General term for all employees in the system	staff_member	Each employee works in one particular branch	works at	Branch
Branch	Term for each physical site on a highstreet	shop	Each branch is in a geographical region	situated in	Region

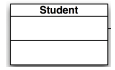
Conceptual database design

UML Class diagrams

Use one box for each object / entity / class



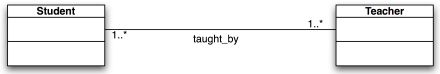
Subdivide box into three, name of class at top



Conceptual database design

UML Class diagrams

Indicate a relationship with a named line between them



Multiplicity indicated by x .. y notation

x is smallest number (0, 1, or n) y is largest (1, n or *)

Conceptual database design

1.3 Identify and associate attributes with entity and relationship types

Determine if derived attribute

Not stored but calculated when required

age	should always be calculated
total cost	could be calculated

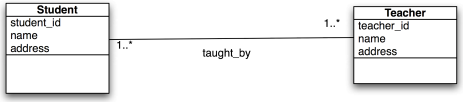
Conceptual database design

1.3 Identify and associate attributes with entity and relationship types

Identify and associate attributes with appropriate entities or relationships

Identify the names and types of values that make up an entity

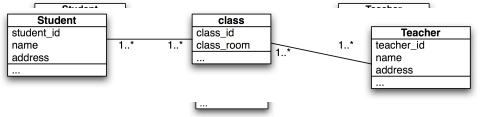
Written into the second part of each UML class



Conceptual database design

1.3 Identify and associate attributes with entity and relationship types

Determine attributes on relationships



The importance of the relationship attributes may force a rethink - promoting the relationship attributes into a new class

Conceptual database design

1.3 Identify and associate attributes with entity and relationship types

Determine if single or composite

Determine if single or multi-valued

student_id	single, single valued
name	composite, single valued
first name	single, single valued
phone number	single, multi valued

Conceptual database design

1.4 Determine attribute domains

Determine types of attributes

Could be placed in the data dictionary

Attribute Name	Description	Aliases	Valued	Composite	Calculated	type
Name	Name of employee	employee_name	single	yes	No	char
Address	Address of employee	employee_address	single	yes	No	char
Phone	Phone number		multi	no	no	char
Age	Age of employee		single	no	yes	number

Conceptual database design

1.5	Determine SK/CK/PK attributes

Determine superkeys, candidate keys and primary keys

New attributes may have to be created if no appropriate key can be found

Remember that keys can be combinations of attributes

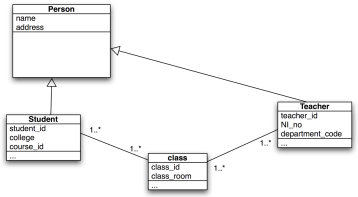
Conceptual database design

1.6	Specialise/generalise entity types (optional)

Object data model allows for superclasses and subclasses

Parent of / Child of type relationships

Sometimes called inheritance



Conceptual database design

1.7	Check model for redundancy

Recheck 1..1 relationships to see if efficiency can be gained by merging

Remove redundant relationships

Check for loops in the diagram

Remove relationships that are parts of loops that can be generated

Conceptual database design

1.8	Validate model against user transactions

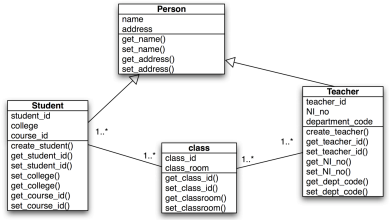
Find the transactions / actual activities that take place

Place actions in third sub box of class diagram

Conceptual database design

1.9	Review data model with user

Check the model to ensure its 'truth'



Conceptual database design

Logical database design

Physical database design

Build and Validate Local Logical Data Model

Build a logical data model based on the conceptual data model of the user's view of the enterprise - then validate using normalization and against required transactions

Build and Validate Local Logical Data Model

Conceptual database design	2.1 Remove features not compatible with the relational data model
Logical database design	2.2 Derive relations
Physical database design	2.3 Validate relations using normalisation
	2.4 Validate relations against user transactions
	2.5 Define integrity constraints
	2.6 Review local logical data model with user

Logical database design

2.2 Derive relations

To derive relations from the local logical data model and to document the composition of each relation including identifying any foreign keys

student (SID, firstname, lastname, address1, address2, address3, address4, postcode ...)

Logical database design

2.1 Remove features not compatible with the relational data model

To refine the local conceptual data model to remove features not compatible with the relational data model

Map this model to a local logical data model

Logical database design

2.3 Validate relations using normalisation

Apply standard normalisation tests

Most organisations aim for 3rd Normal Form or Boyce-Codd Normal Form

Logical database design

2.1 Remove features not compatible with the relational data model

- Remove M:N relationships
- Remove complex relationships
- Remove recursive relationships
- Remove relationships with attributes
- Re-examine 1:1 relationships
- Remove redundant relationships

Logical database design

2.4 Validate relations against user transactions

Ensure that the logical data model supports the transactions that are required by the user view

Logical database design

2.4	Validate relations against user transactions

Ensure that the logical data model supports the transactions that are required by the user view

Conceptual database design

Logical database design

Physical database design

Build and Validate Global Logical Data Model

Objective: Combine the individual local logical data models into a single global logical data model

Logical database design

2.5	Define integrity constraints

To identify and document the integrity constraints given in the user's view of the enterprise. This includes identifying:

Required data
Referential integrity (foreign key validity)
Attribute domain constraints (restricted values like Y/N)
Enterprise constraints
Entity integrity (PKs not null)

Conceptual database design

Logical database design

Physical database design

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	Review logical data Models with users

Logical database design

2.6	Review local logical data model with user

To ensure that the local logical data model is a true representation of the user's view

Logical database design

3.1	Merge Local Logical Data Models into Global Model

Review the names of entities and their primary keys
Review the names of relationships
Merge entities from the local views
Include (without merging) entities unique to each local view
Merge relationships from the local views
Include (without merging) relationships unique to each local view
Check for missing entities and relationships
Check foreign keys
Check Integrity Constraints
Draw the global logical data model
Update the documentation

Logical database design

3.2	Validate Global Logical Data Model

To validate the global logical data model using normalization and against the required transactions, if necessary

Logical database design

3.3	Check for future growth

To determine whether there are any significant changes likely in the foreseeable future and to assess whether the global logical data model can accommodate these changes

Logical database design

3.4	Review logical data Models with users

To ensure that the global logical data model is a true representation of the enterprise

Conceptual database design

Translate Global LDM for target DBMS

4.1	Design Base Relations for Target DBMS
4.2	Derive representations of derived data
4.3	Design Integrity Rules for Target DBMS

Logical database design

Design Physical Representation

5.1	Analyze Transactions
5.2	Choose File Organisations
5.3	Choose Indexes
5.4	Estimate Disk Space

Physical database design

6	Design User Views
7	Design security mechanisms
8	Consider redundancy
9	Monitor and fine tune