INFO20003 Database Systems

Dr Renata Borovica-Gajic

Lecture 02
Database Development Process

First: Let's vote!

Student representative candidates:

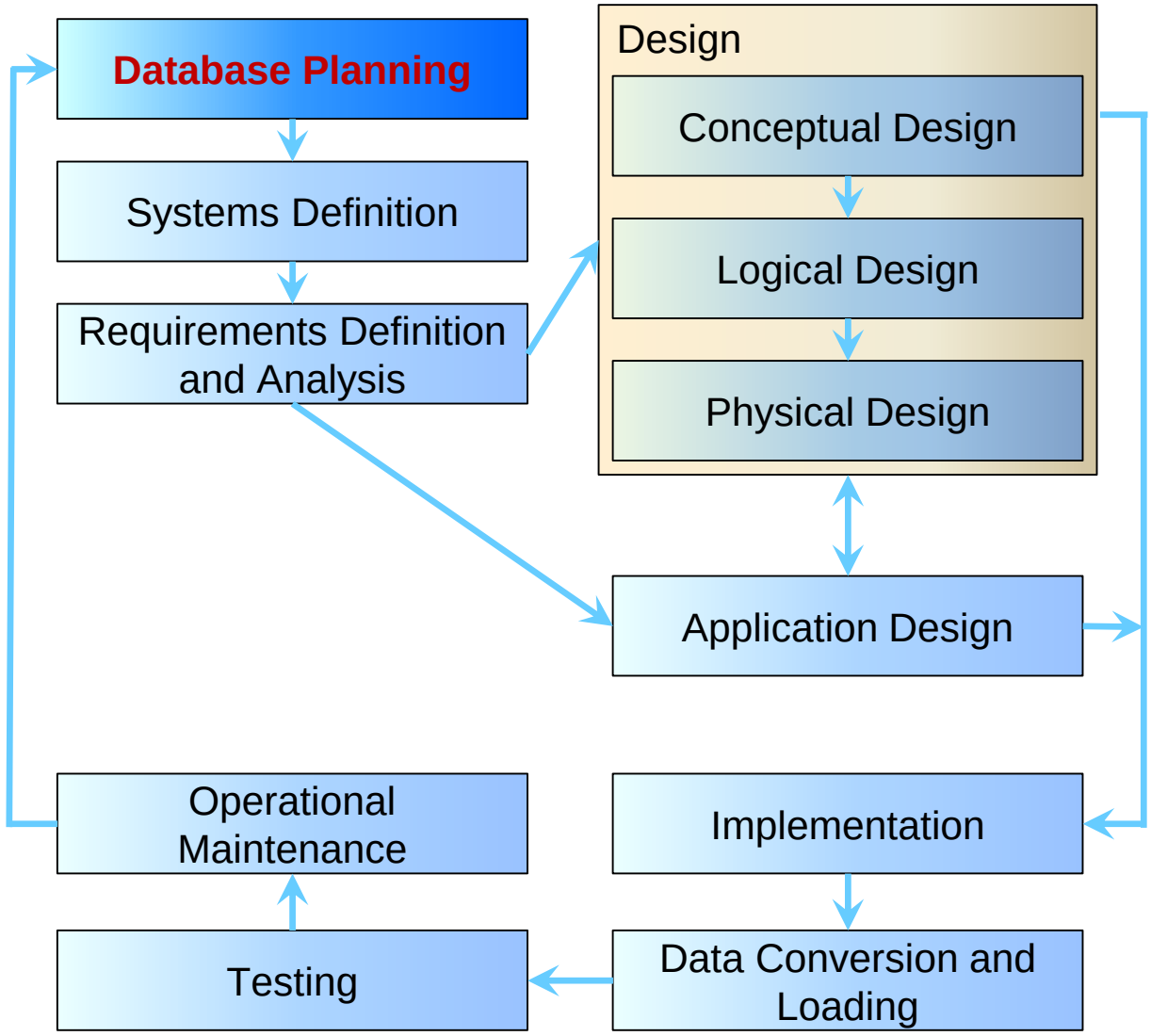
- Weijia Wang
- Ilan Brian Rosen
- Ujashkumar Rakeshkumar Patel
- Manindra Arora
- Tarnvir Singh Grewal Grewal
- Jeremy Yee Lun Tee
- Abdiaziz Farah

https://fbeunimelb.au1.qualtrics.com/jfe/form/SV_37V8e5XKSU07hJz

Coverage

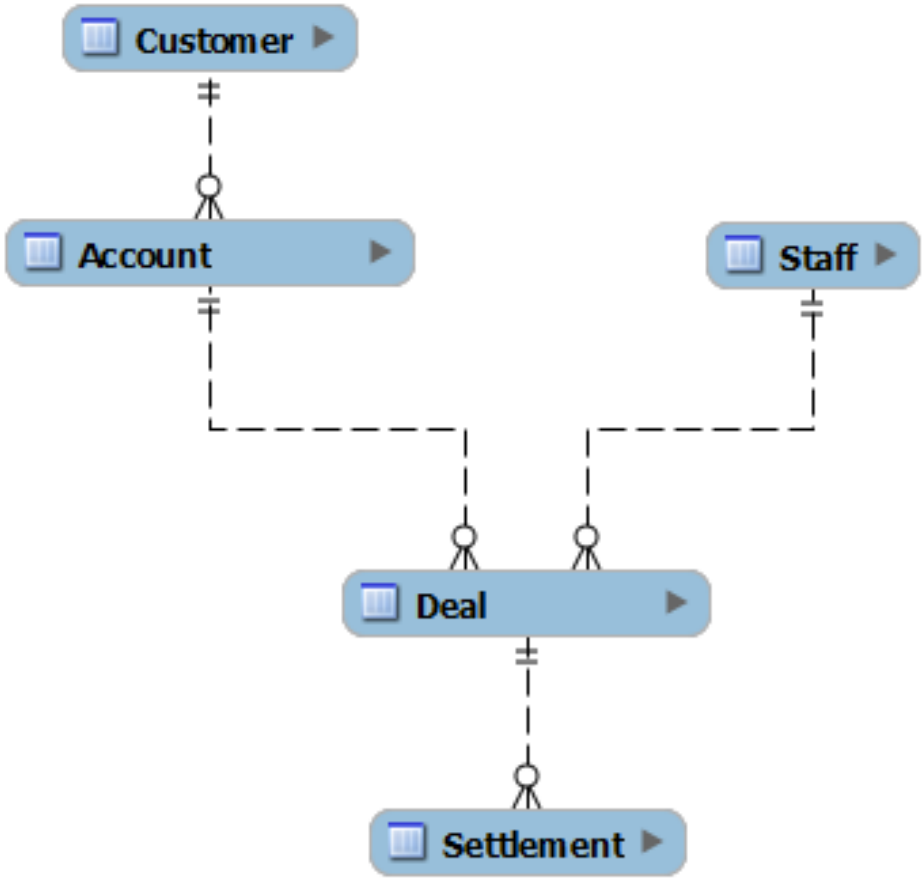
- How database applications are developed
  - The development lifecycle
  - Database design
    - Conceptual design
    - Logical design
    - Physical design

Database Development Lifecycle (Simplified)



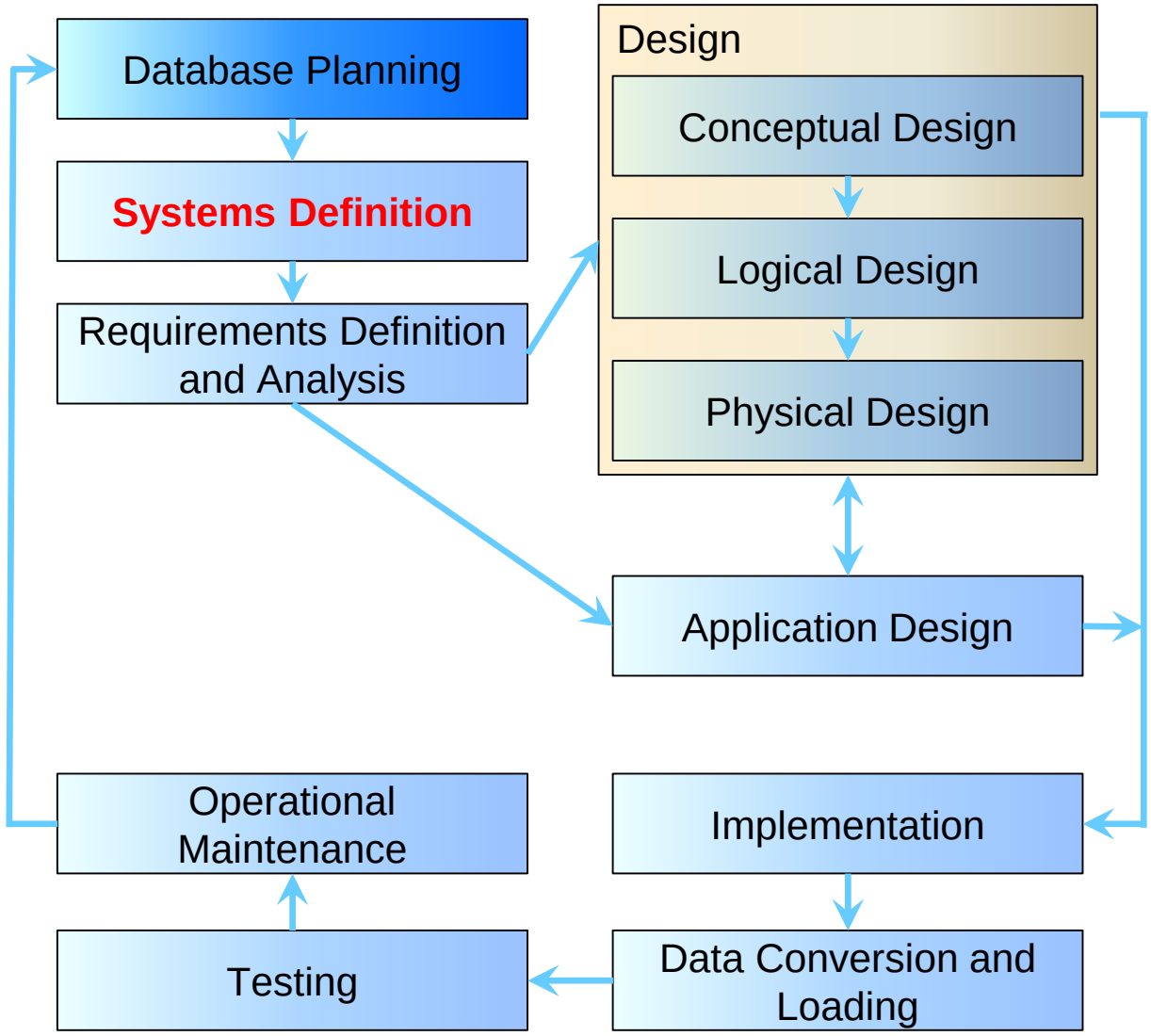
- Planning how to do the project.
  - How does the enterprise work
  - Enterprise data model
- How can the stages be completed efficiently and effectively.
- Outside scope of the course

Example Enterprise Data Model – Investment Banking



- A top level perspective on the data requirements
- Each box (subject area) would have a data model

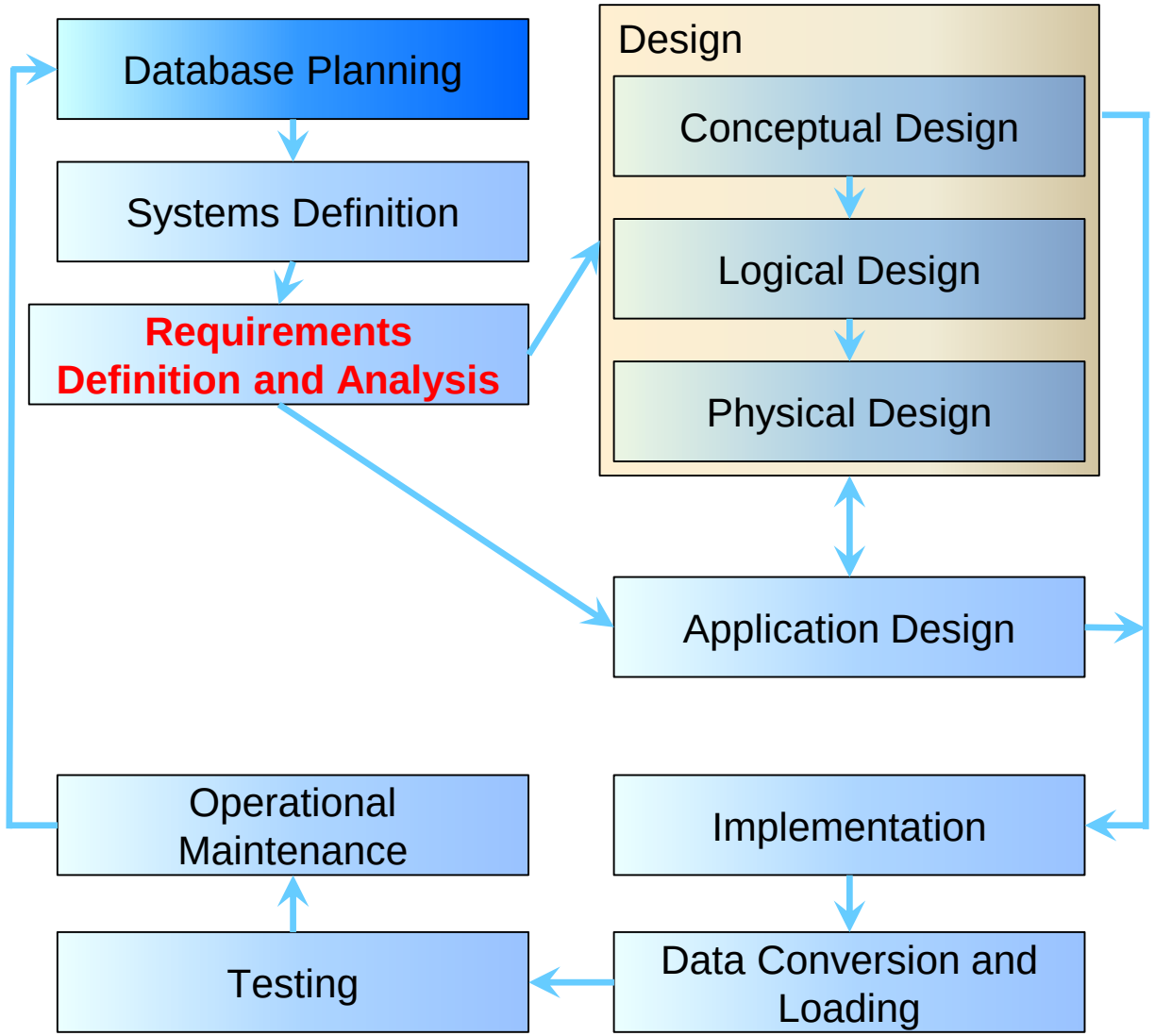
Database Development Lifecycle (Simplified)



- Specifying scope and boundaries
  - Major user views
  - Users
  - Application areas
- How does it interfere with other organisational systems
- User views – how the system operates from differing perspectives

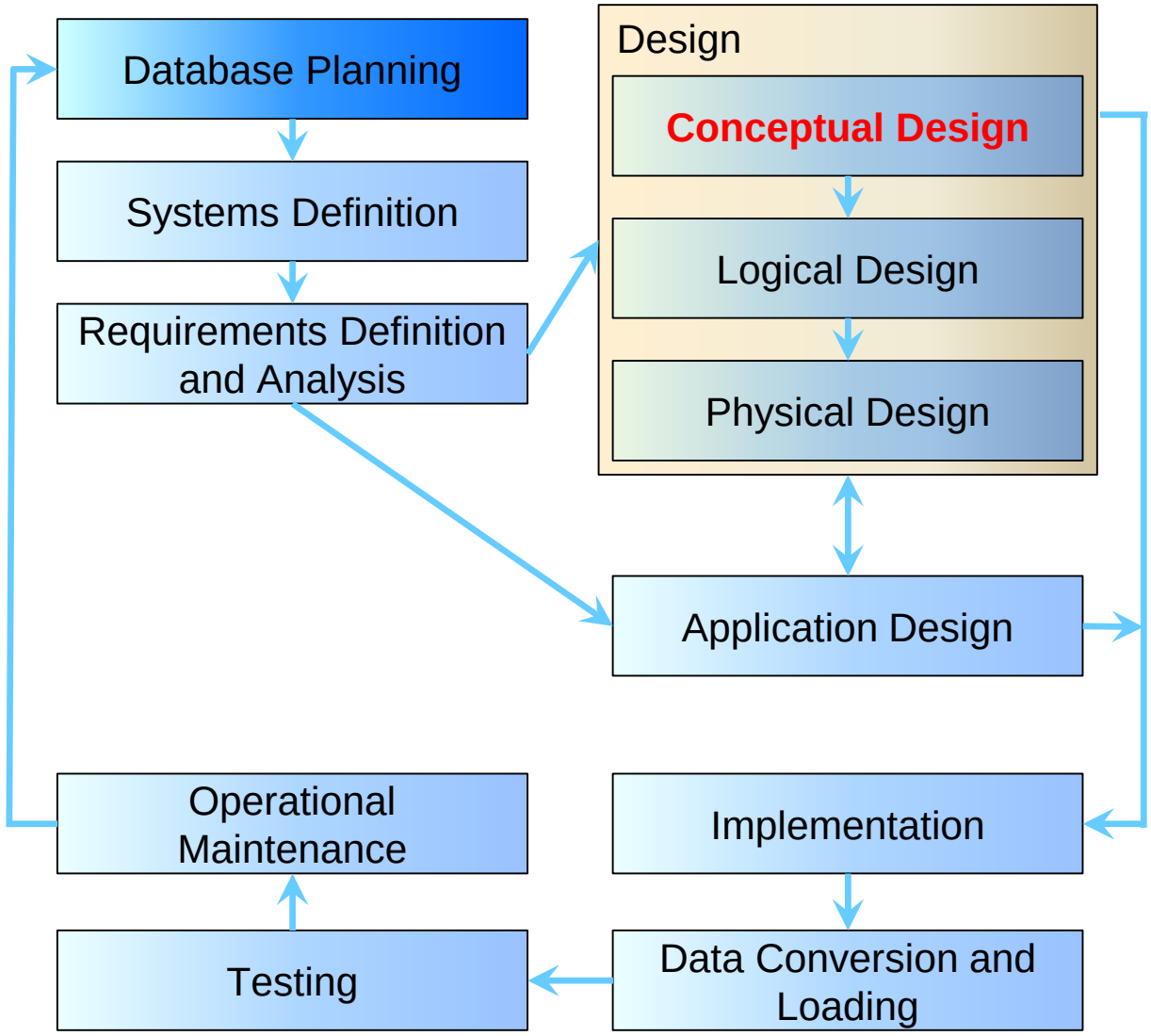
- Outside scope of the course (slightly)

Database Development Lifecycle (Simplified)



- Collection and analysis of requirements for the new system
- You will be given the requirements, but you will need to understand these!
- You may need to ask requirement questions about what you are given (especially for the assignment)
- You should develop a data dictionary here and extend it in design

Database Development Lifecycle (Simplified)

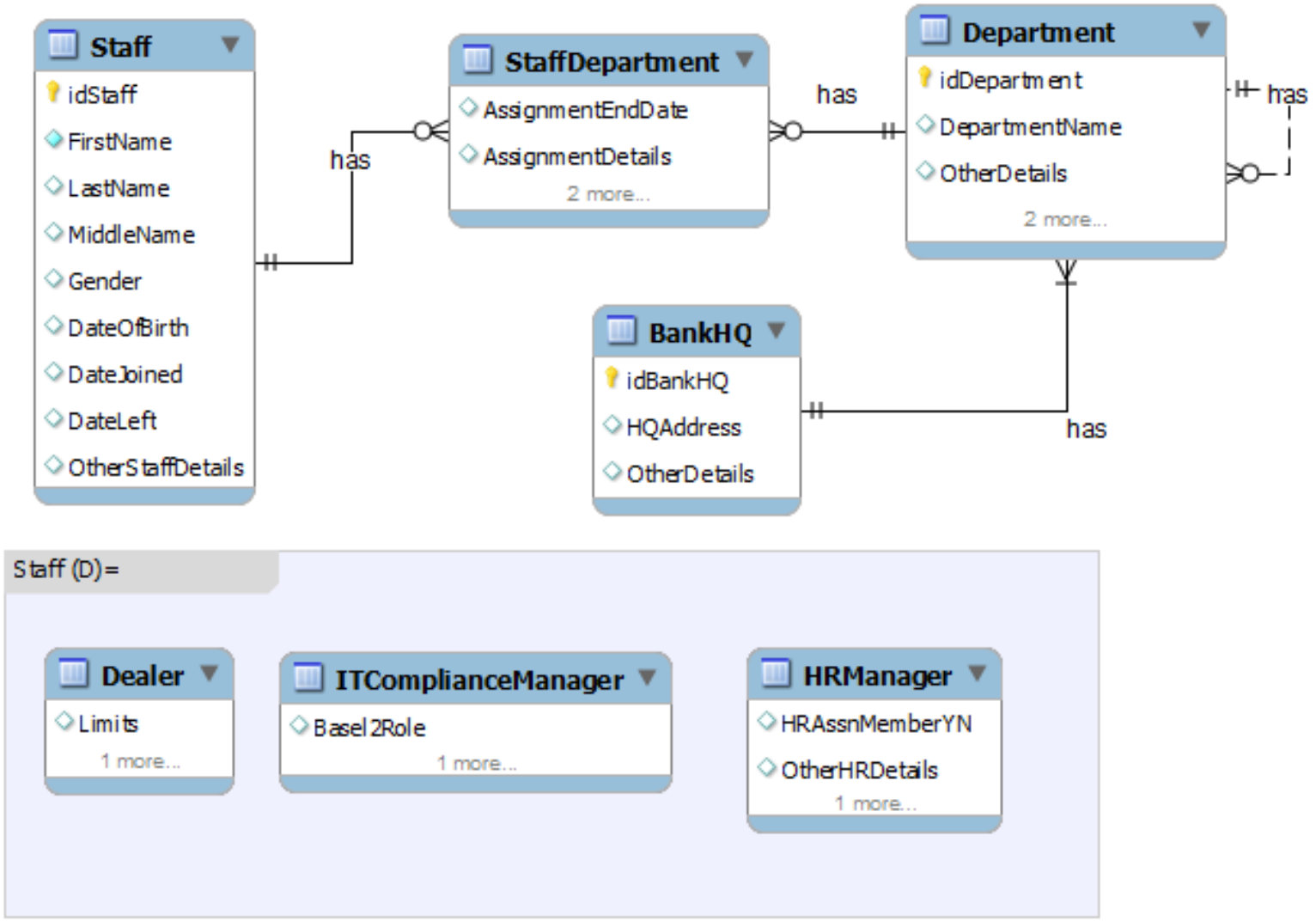


- Construction of a model of the data used in the database – independent of all physical considerations
- Data Models
  - ER, EER Diagrams

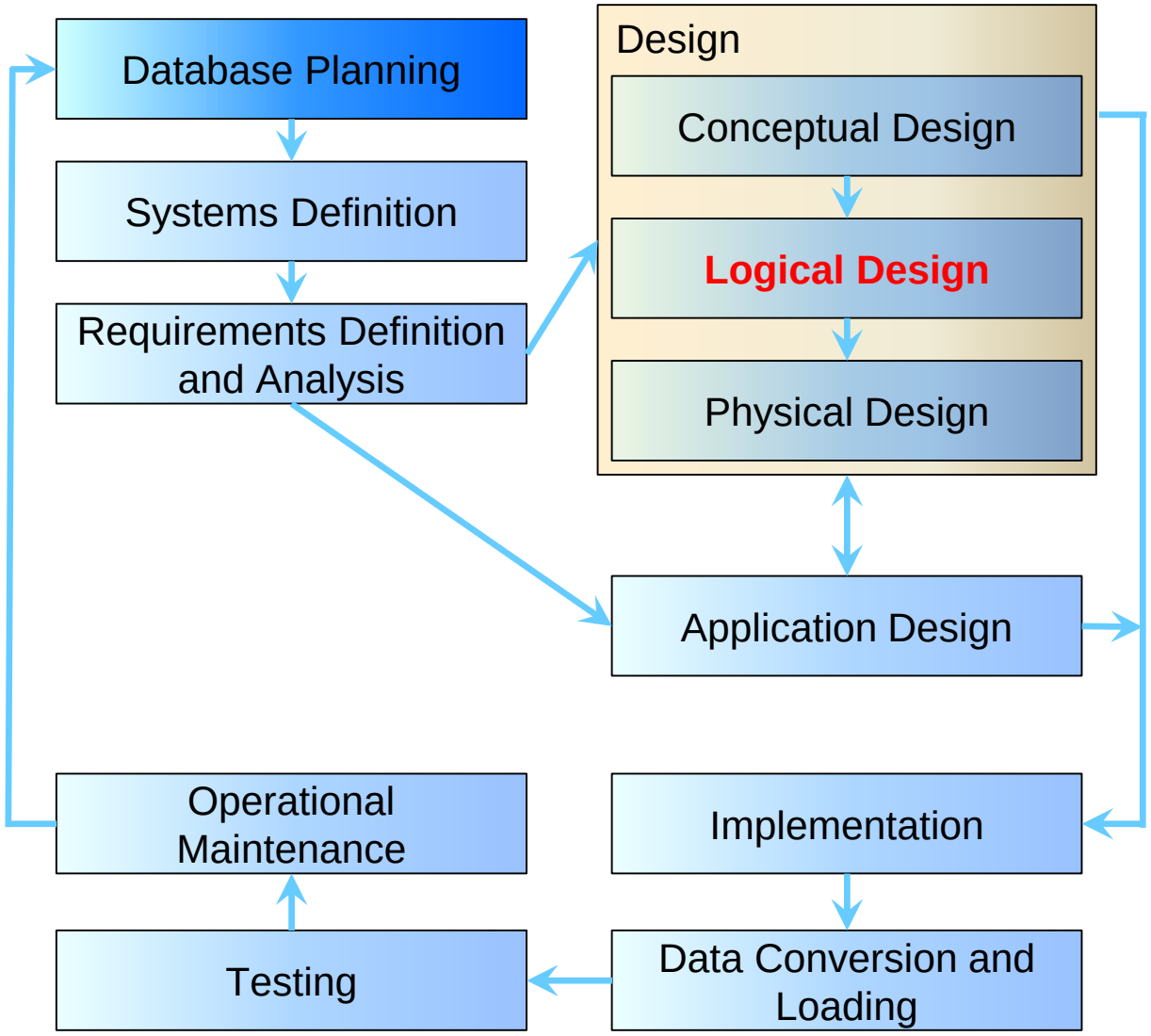
Analysis of the problem

- An investment bank has a number of branches. Within *each branch a number of departments* operate and are *structured in a hierarchical manner*. The bank employs around 3000 staff *who are assigned to work in the* various departments across the branches. There are essentially *three types of special employees* where extra details required by the system. There are dealers who *carry out investments* who *have limits imposed upon them* for how much they can spend. There are IT compliance managers who's *Basel2 role* is required to be stored and there are HR managers that need have their *assessment number* recorded (along with other details not specified here).
- We need a database to record staff details including which department and branch they are assigned...

Example Conceptual Data Model (EER) – Investment Banking

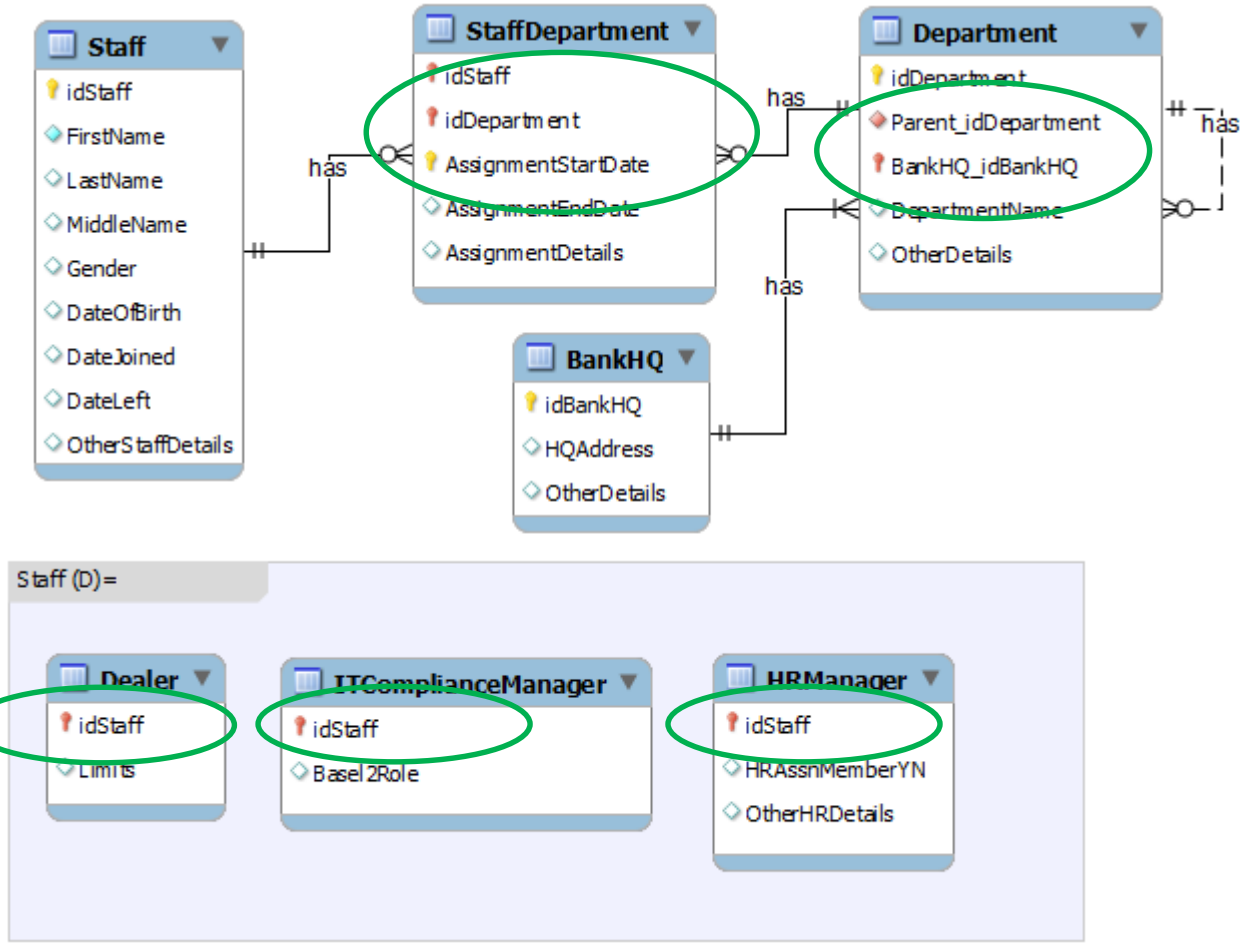


Database Development Lifecycle (Simplified)



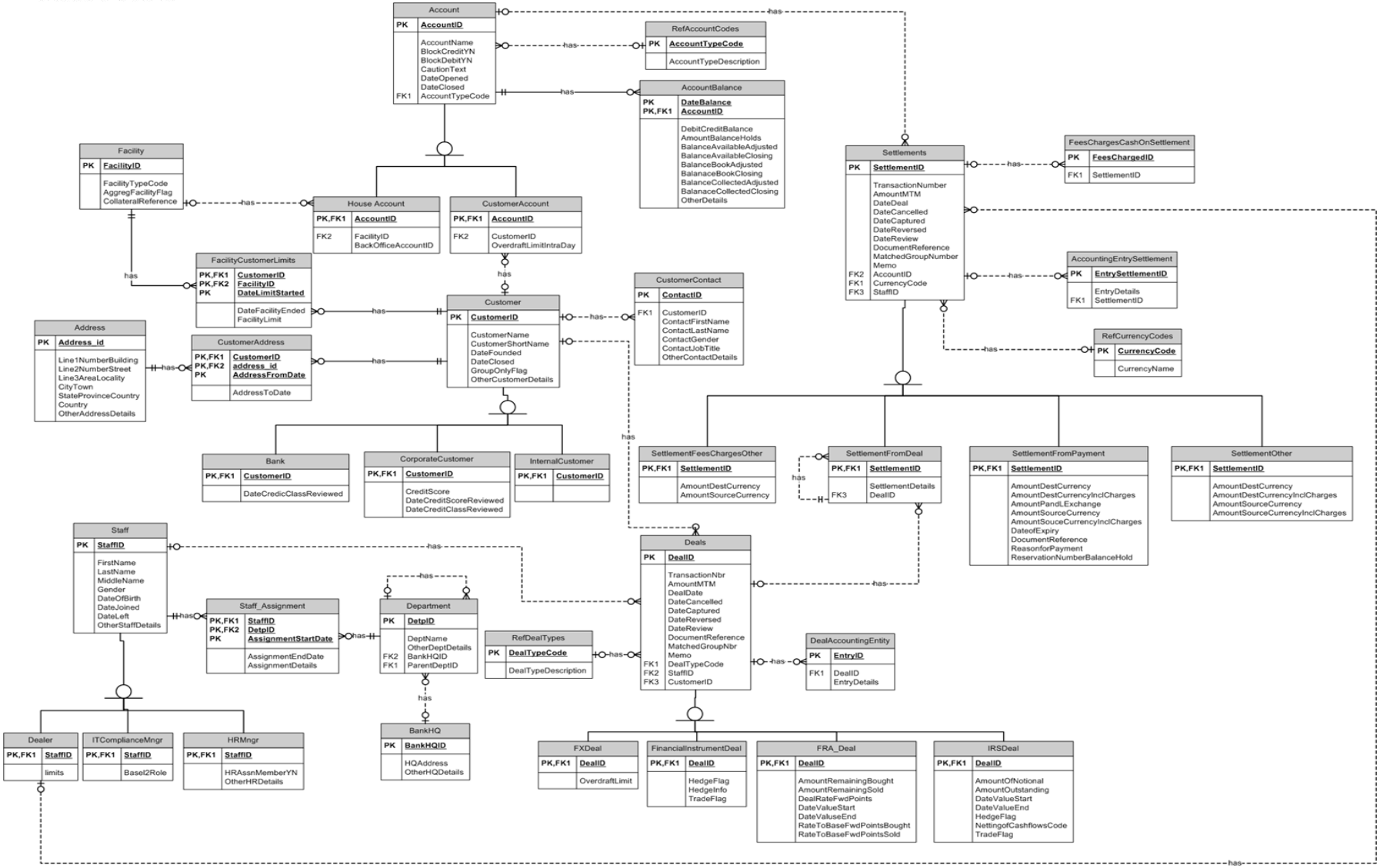
- Construction of a model of the data based on the conceptual design (based on the ER, EER models) – Independent of a specific database and other physical considerations

Example Logical Data Model – Investment Banking (Staff)

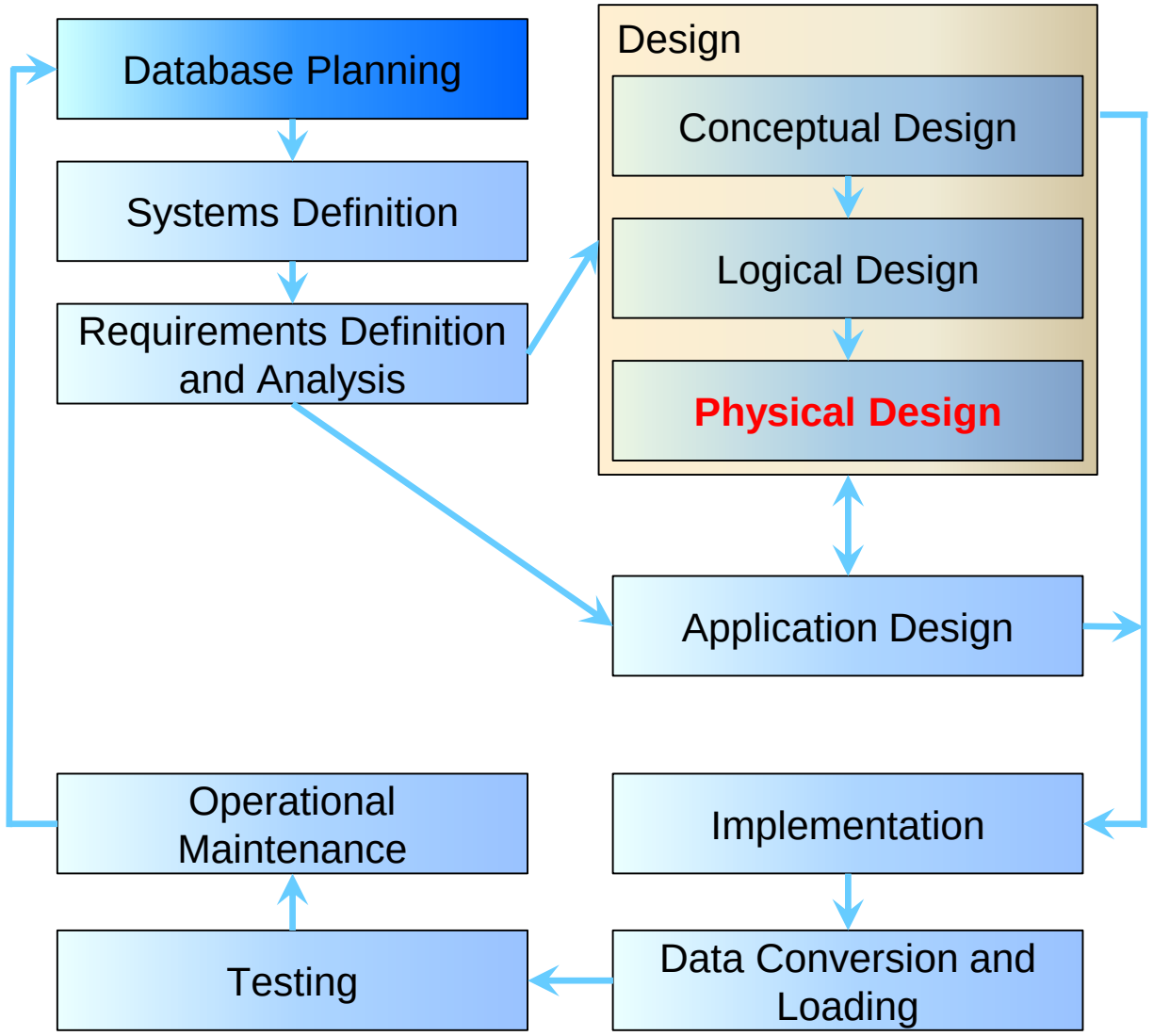


Changes from Conceptual Model (EER)

Example Logical Data Model – Investment Banking (Complete)

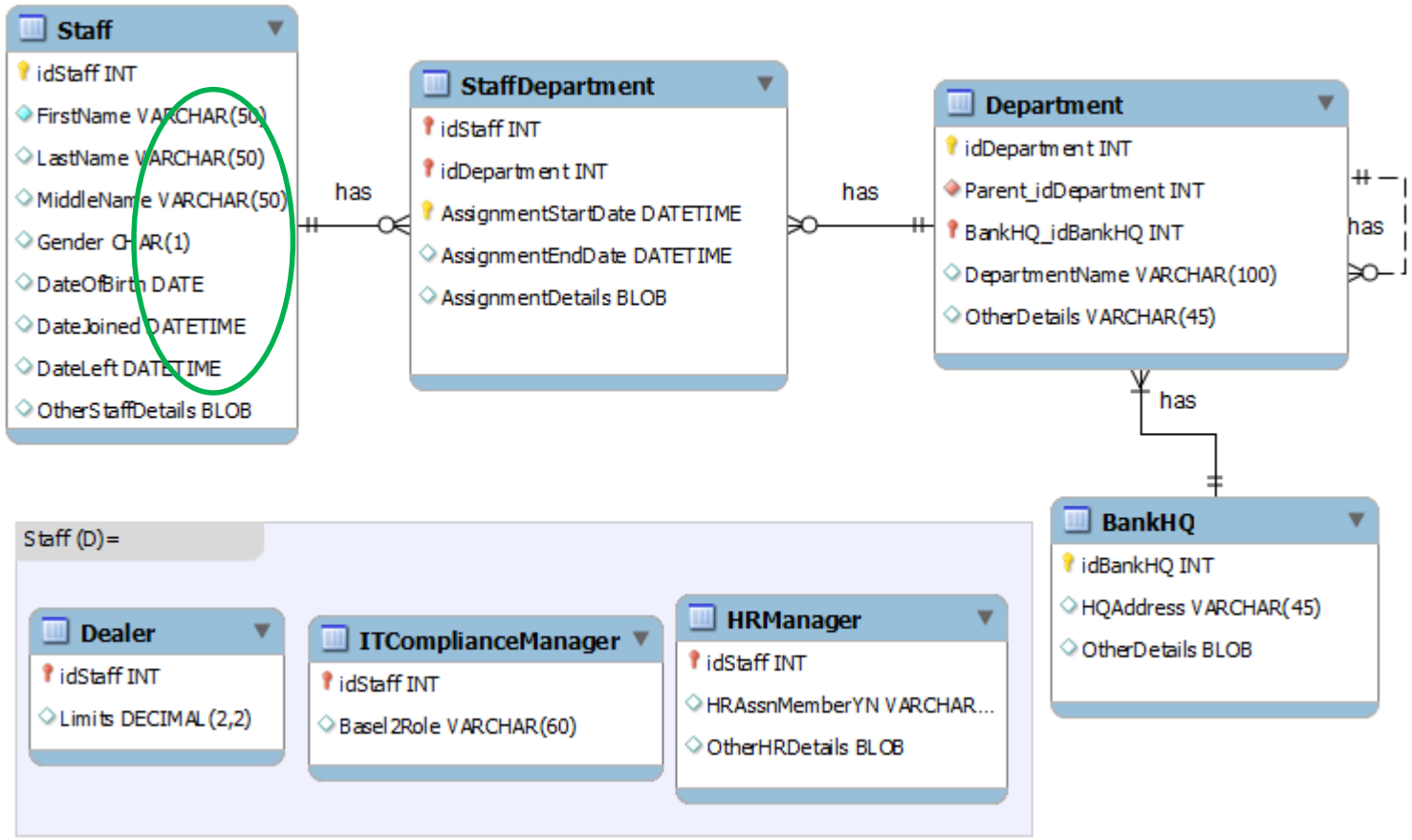


Database Development Lifecycle (Simplified)



- A description of the implementation of the logical design – for a specific DBMS.
- Describes:
  - Basic relations
  - File organisation
  - Indexes
  - Integrity constraints
  - Security measures
- Although we don't do physical design, we discuss some of the issues

Example Physical Model – Investment Banking (Staff)



Choosing Data Types

- Types helps the DBMS store and use information efficiently
  - Can make assumptions in computation
  - Consistency is guaranteed
- Minimise storage space
- Need to consider
  - Can you store all possible values
  - Can the type you choose support the data manipulation required
- Selection of types may improve data integrity

Example of Data Dictionary

- We do the data dictionary as an ongoing process during analysis and design of the database
- Example of what is required

Key	Attribute	Data Type	Not Null	Unique	Description
Type of key Is it a primary key or a foreign key (leave blank if neither)	Name of Attribute	Data type of attribute	If the field is required or is optional	Must the value in the field be unique for that field	A description of the attribute giving any information that could be useful to the database designers or to the application developers. This would include things like attribute sizes, valid values for an attribute, information about coding for this attribute etc.

Example of Partial Data Dictionary

Key	Attribute	Data Type	Not Null	Unique	Description
PK	StaffID	Integer	Y	Y	ID number of the staff member, should be 5 in length. This is the primary identifier (key) of the table.
	FirstName	VarChar			The first given name of the staff member, up to 100 characters.
	LastName	VarChar	Y		The family name of the staff member, up to 100 characters. This must exist for every staff member
	Gender	ENUM	Y		The gender of the staff member, valid values are only “Male” or “Female” (???). An enumerated data type should be used if possible. This should be limited in applications using this field also.
	DateOfBirth	DateTime	Y		This is when the staff member was born. Needs dd/mm/yyyy format.
	...				

MYSQL Data Types (some)

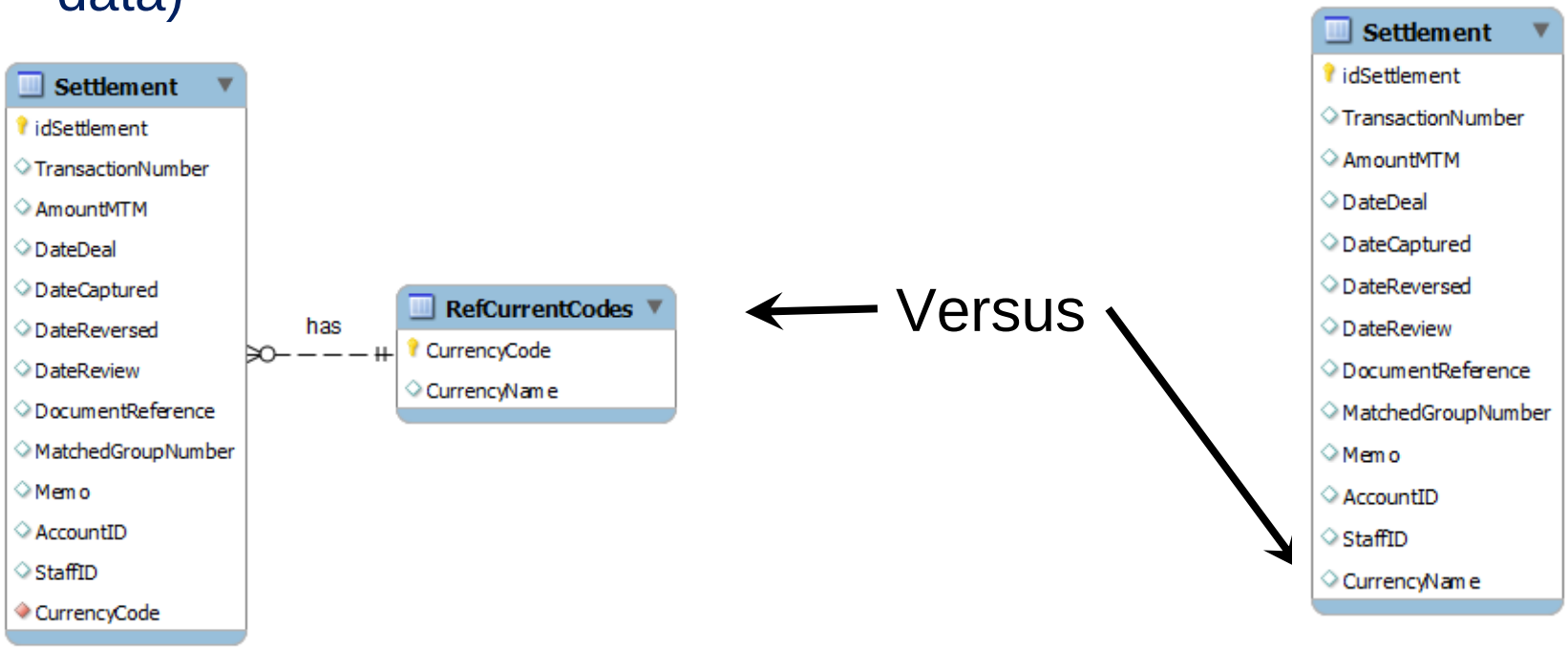
- Character Types
  - CHAR(M): A fixed-length string, right-padded with spaces. The range of M is 0 to 255.
  - VARCHAR(M): A variable-length string. The range of M is 1 to 65535. (its 255 max. in MySQL 4).
  - BIT, BOOL, CHAR: CHAR(1).
  - BLOB, TEXT: up to 65535 bytes (for blob) or characters (for text).
  - ENUM ('value1','value',…) up to 65,535 members.
  - SET ('value1','value2', …) up to 64 members.
- Integer Types
  - TINYINT[(M)]: Signed (-128 to 127) Unsigned(0 to 255)
  - SMALLINT[(M)]: Signed (-32768 to 32767) Unsigned (0 to 65535)
  - MEDIUMINT[(M)]: Signed (-8388608 to 8388607) Unsigned (0 to 16777215)
  - INT[(M)] / INTEGER[(M)]: Signed (-2147483648 to 2147483647) Unsigned (0 to 4294967295)
  - BIGINT[(M)]:Signed(-9223372036854775808 to 9223372036854775807) Unsigned(0 to 18,446,744,073,709,551,615)

MYSQL Data Types (some)

- Real Types
  - FLOAT[(M,D)]: single-precision, allowable values: -3.402823466E+38 to -1.175494351E-38, 0, and 1.175494351E-38 to 3.402823466E+38. M = display width, D = number of decimals.
  - DOUBLE[(M,D)] / REAL[(M,D)]: double-precision, allowable values: -1.7976931348623157E+308 to -2.2250738585072014E-308, 0, and 2.2250738585072014E-308 to 1.7976931348623157E+308.
  - DECIMAL[(M[,D])]: fixed-point type. An unpacked floating-point number. Stored as string. Good for MONEY!
- Time and Date Types
  - DATE 1000-01-01 to 9999-12-31
  - TIME -838:59:59 to 838:59:59
  - DATETIME 1000-01-01 00:00:00 to 9999-12-31 23:59:59
  - TIMESTAMP 1970-01-01 00:00:00 - ~ 2037 Stored in UTC, converted to local
  - YEAR[4] 1901 to 2155 - A useful function in MySQL: NOW();

Other Physical Design Decisions

- How to store “Look Up”
  - Trade off between speed and space (and possibly integrity of data)

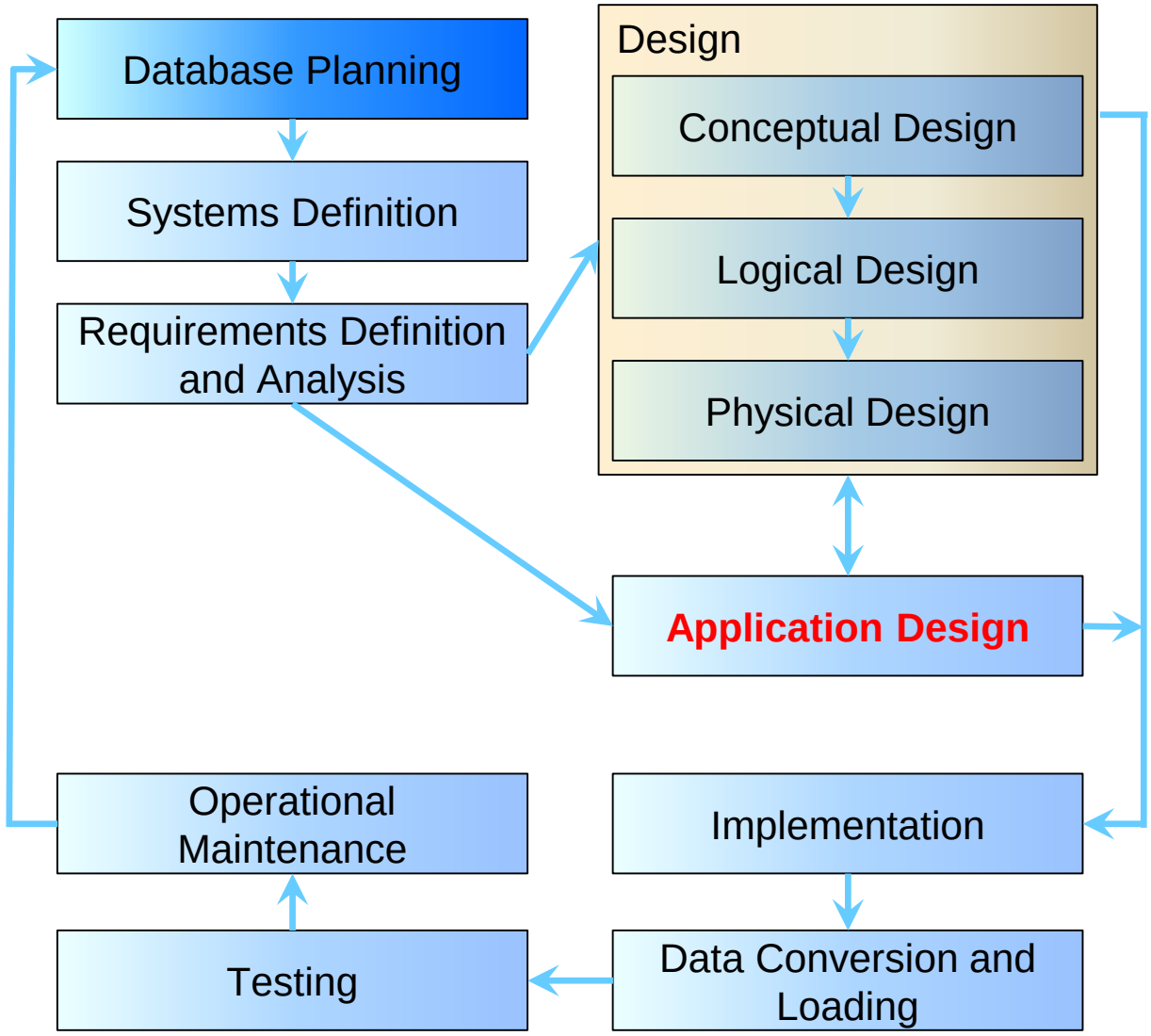


- Data field integrity (ensure fields only contain correct data)
- Handling missing data (concept of NULL data)

Other Physical Design Decisions (2)

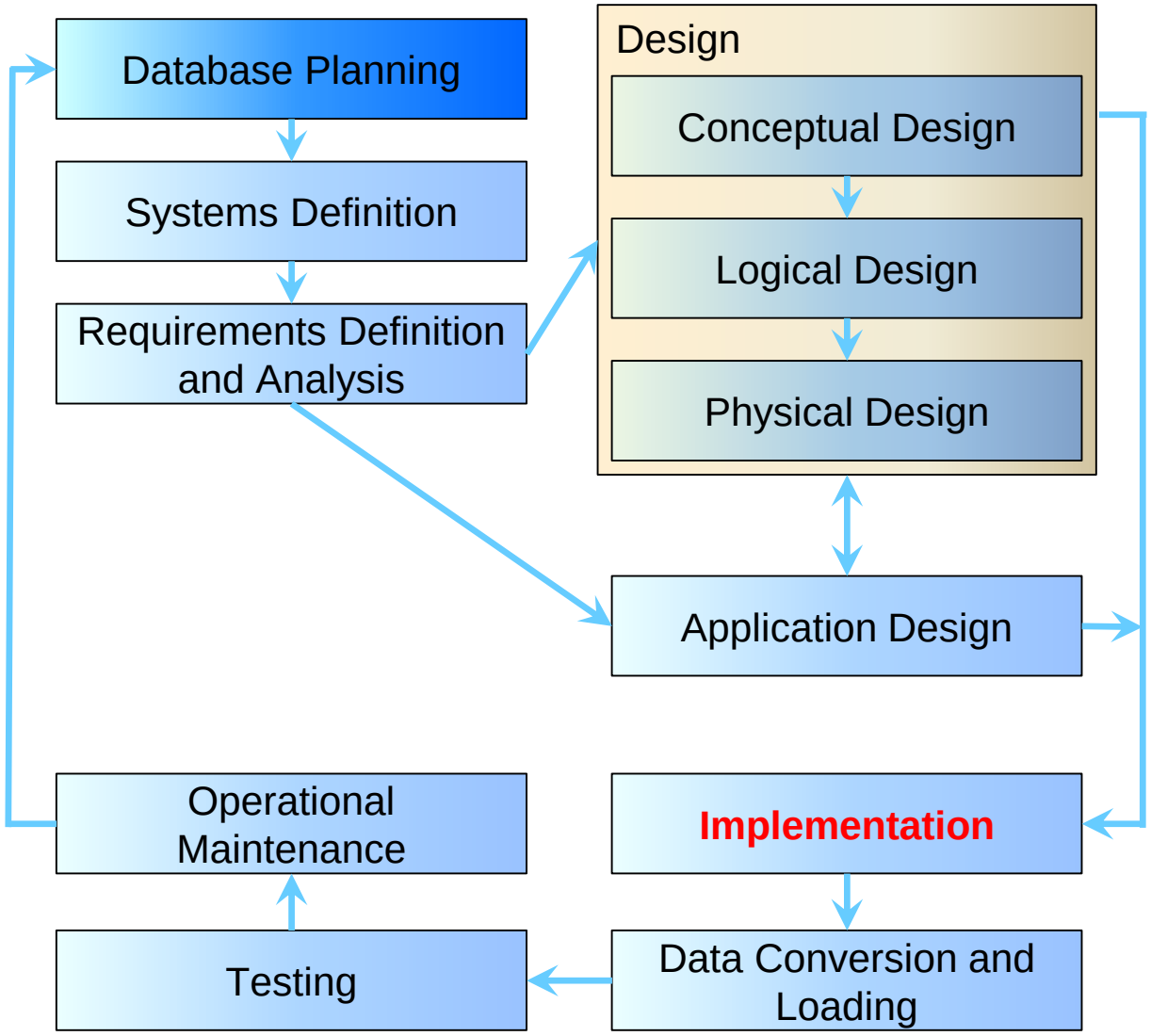
- To De-Normalise or Not (That is the Question)
  - Normalisation
    - A formal method used to validate and improve upon the logical design thus far (which attributes should be grouped together), before proceeding with the physical design.
    - Taught later in the semester
  - De-Normalisation
    - At physical design time need to decide how to implement the design – including removing some of the normalisation steps...
    - Benefits
      - Improved database performance
    - Costs
      - Wasted storage space
      - Data integrity / consistency threats

Database Development Lifecycle (Simplified)



- Done in conjunction with design
- Design of the interface and application programs that use and process the database

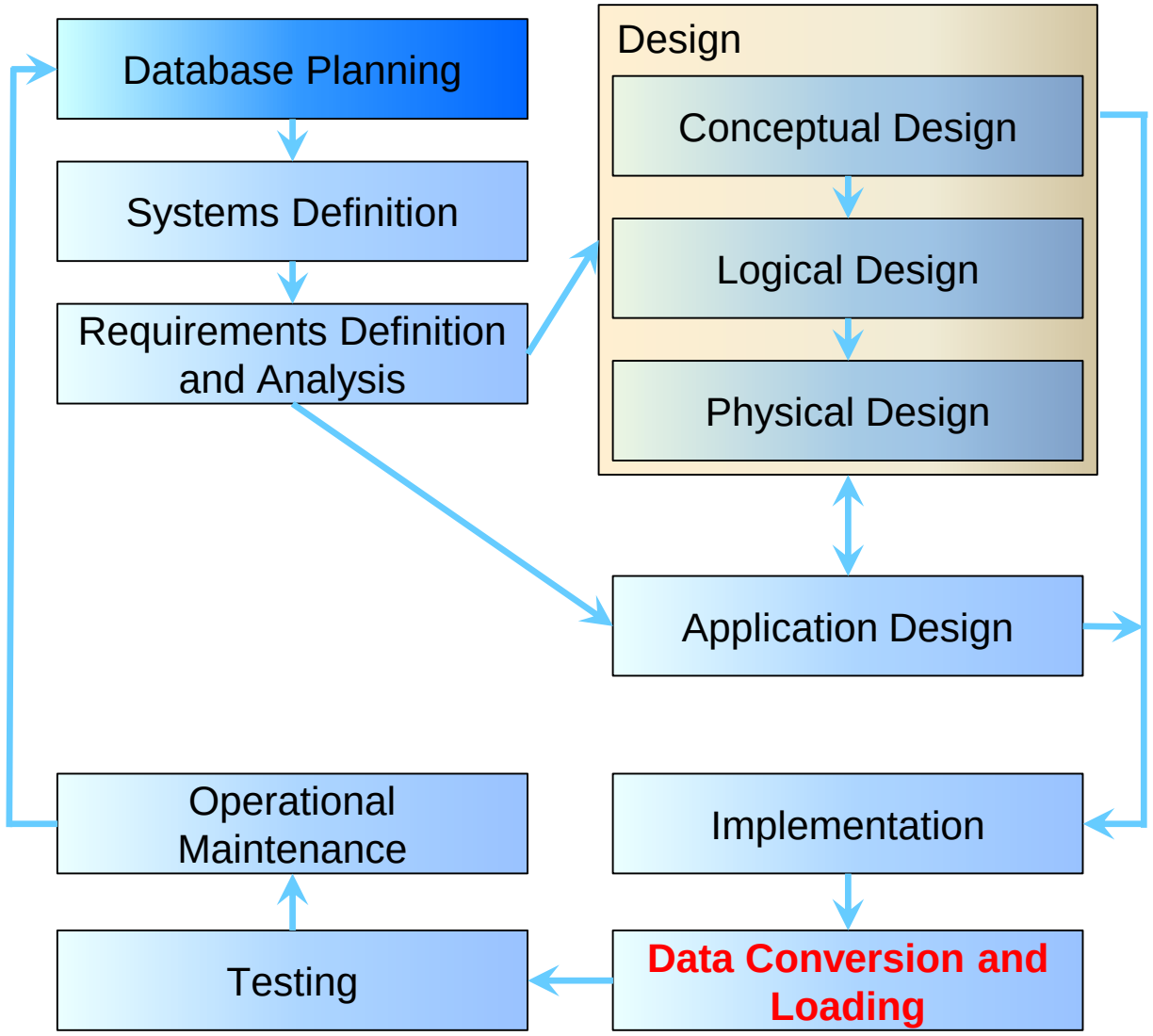
Database Development Lifecycle (Simplified)



- The physical realisation of the database
- Implementation of the design

- Some of the things in implementation are covered as you will be implementing database tables and generating a working application

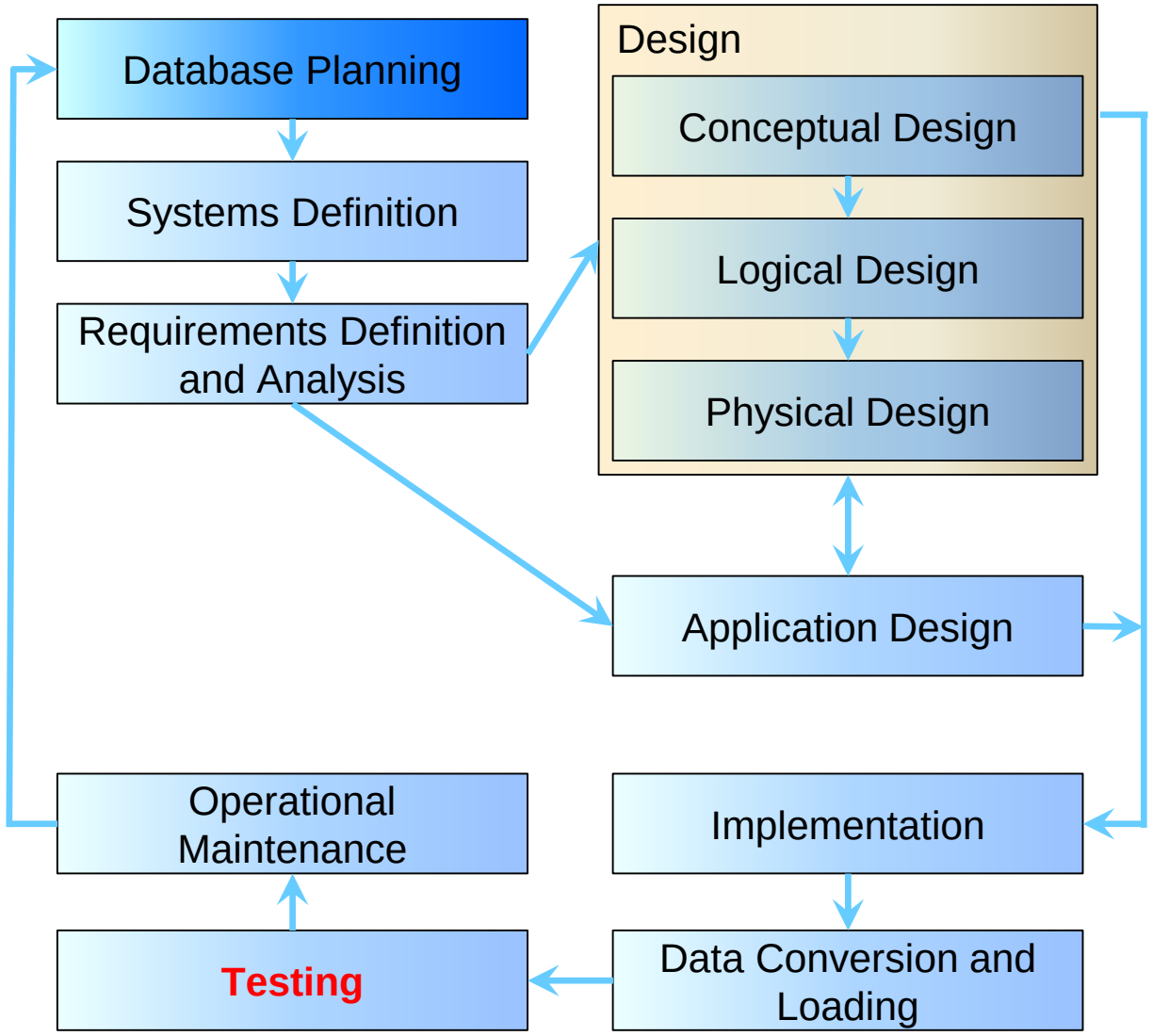
Database Development Lifecycle (Simplified)



- Transfer existing data into the database
- Conversion from old systems
- Non trivial task

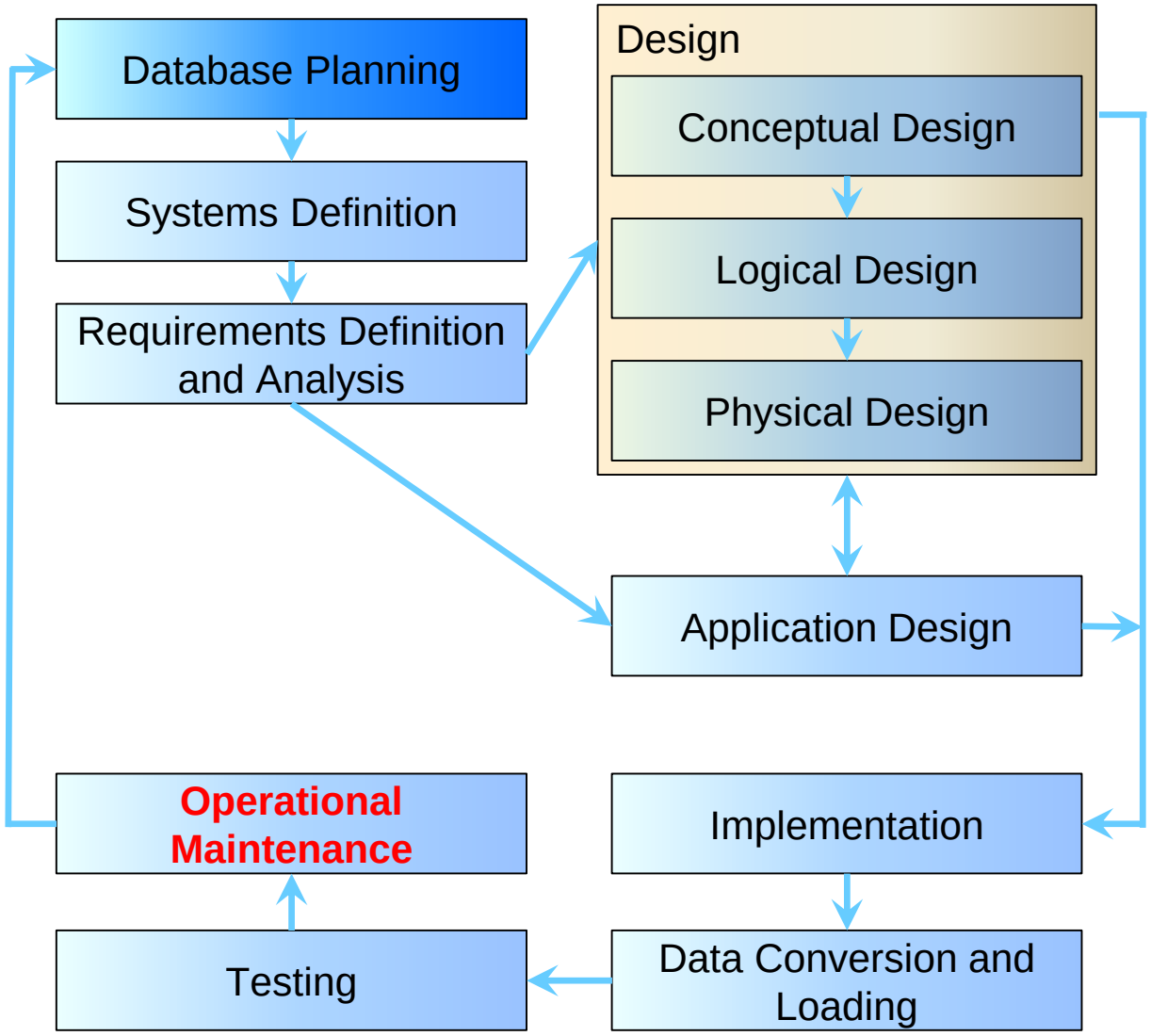
- We give you the data / you make data up. In a real world situation you would have to do this step – very carefully, very time consuming... Lots of issues around this

Database Development Lifecycle (Simplified)



- Running the database to find errors in the design / setup (both at a physical level and at a logical level)
- Other issues also
  - Learnability
  - Performance
  - Robustness
  - Recoverability
  - Adaptability
- Outside scope of the course, although you will be testing your assignment!

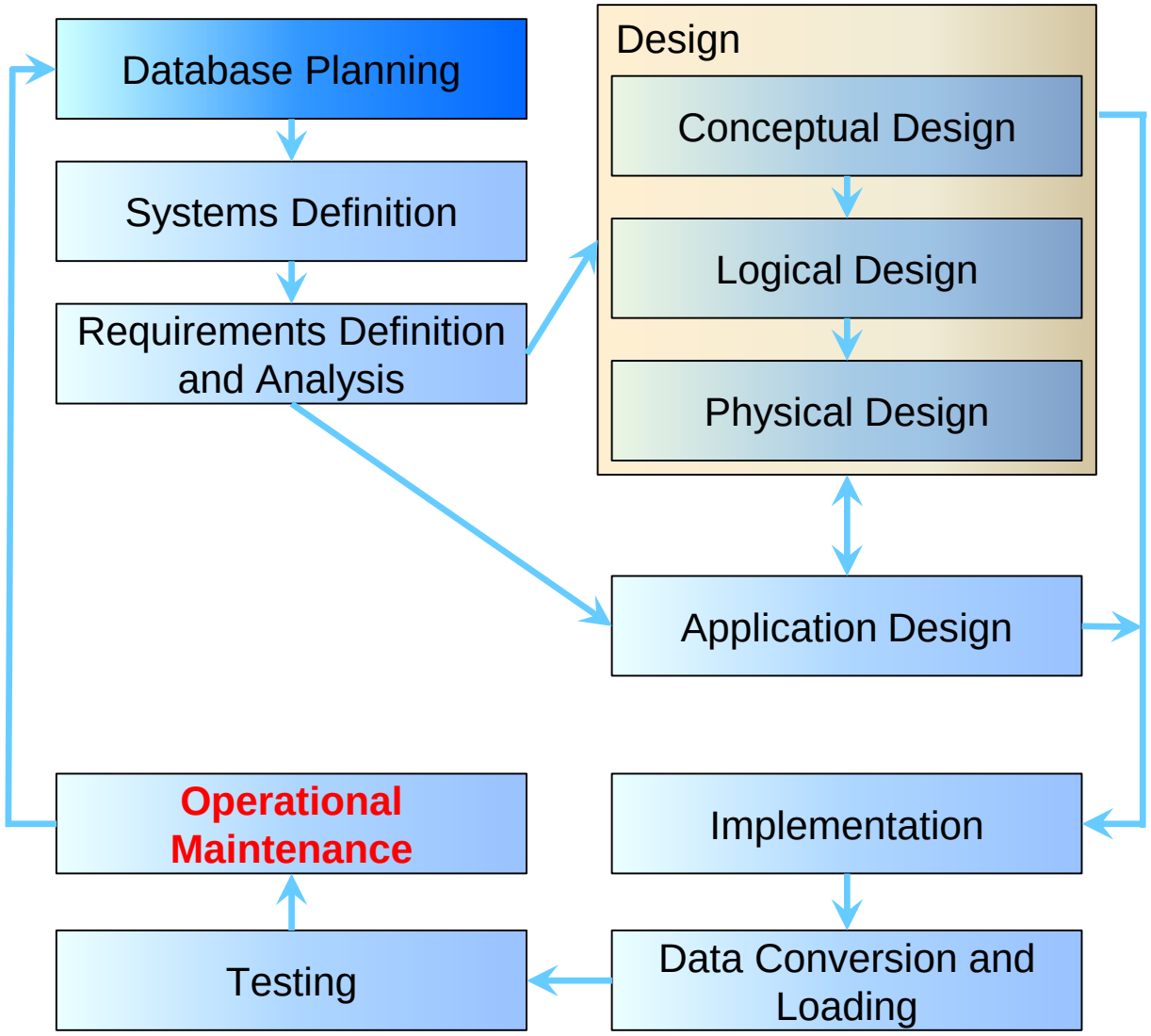
Database Development Lifecycle (Simplified)



- The process of monitoring and maintaining the database system following its commissioning
- Handling new requirements
- Handling changes to requirements

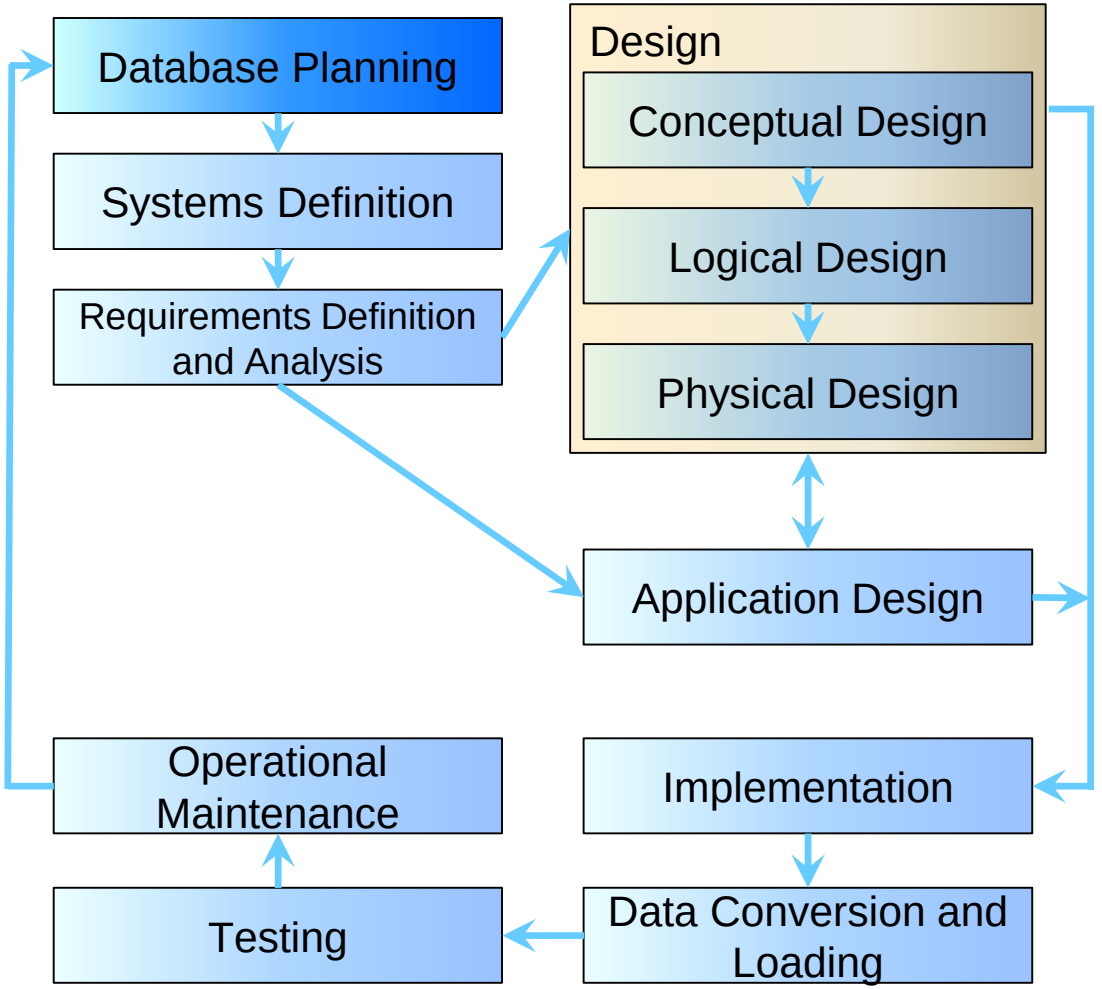
- Outside scope of the course

Database Development Lifecycle (Simplified)



Summary

- Discussed the lifecycle of Database Development
- Showed detail of the Modelling stages



What's Examinable?

- Can you discuss and draw the Database Development Lifecycle?
- This lecture also introduced you to the things we will be doing
  - Data Modelling
    - Conceptual, Logical and Physical Diagrams
  - Any of these could also be on an exam

Next Lecture

- Introduction to Database Design
  - Conceptual design (ER diagrams)