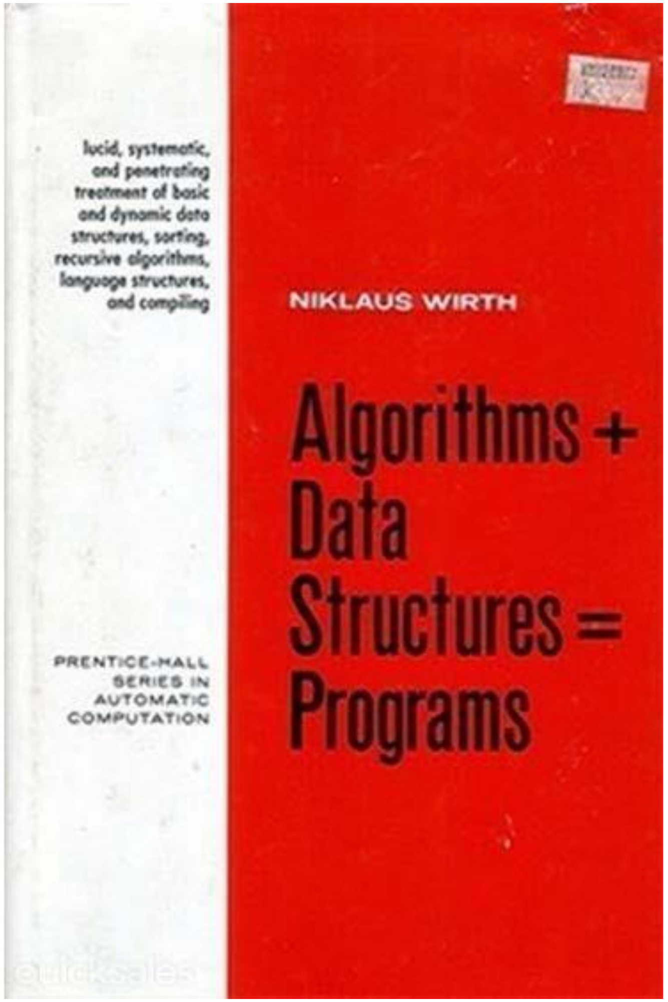
Introduction to Databases

Database Systems & Information Modelling
INFO90002

Week 1 – Databases
Dr Tanya Linden

Context: software and data

- Computer systems consist of *software* (algorithms) working to process *data*.
- You will learn about creating software and algorithms in other subjects (COMP90059)
- This subject is about *data*.
- Focus away from *computation* to *data*
  Processed data = information



The Modern Data Challenge

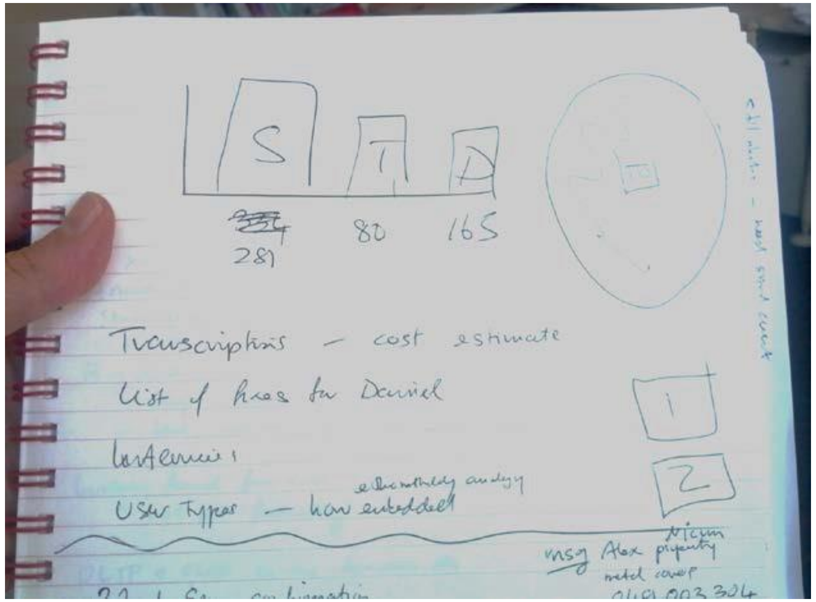
Modern organisations need to store and retrieve large amounts of data

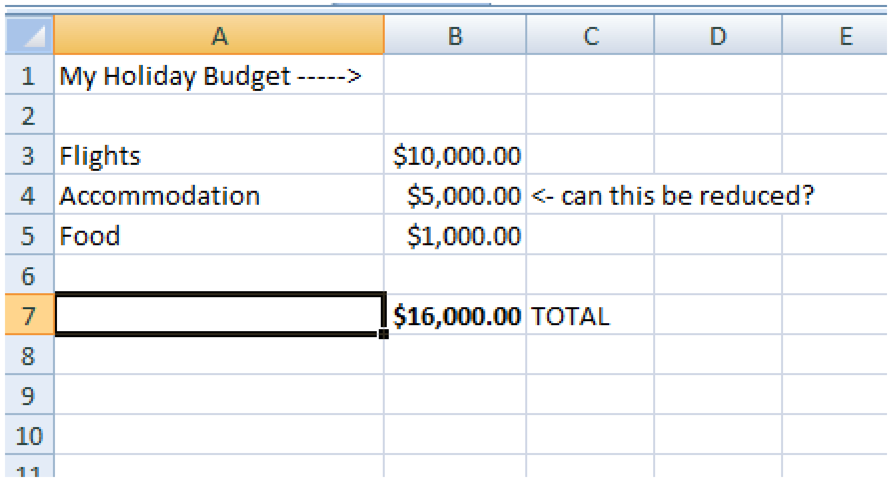
Data can be divided into three major **categories**

- **Structured Data**
- **Semi-Structured Data**
- **Unstructured Data**

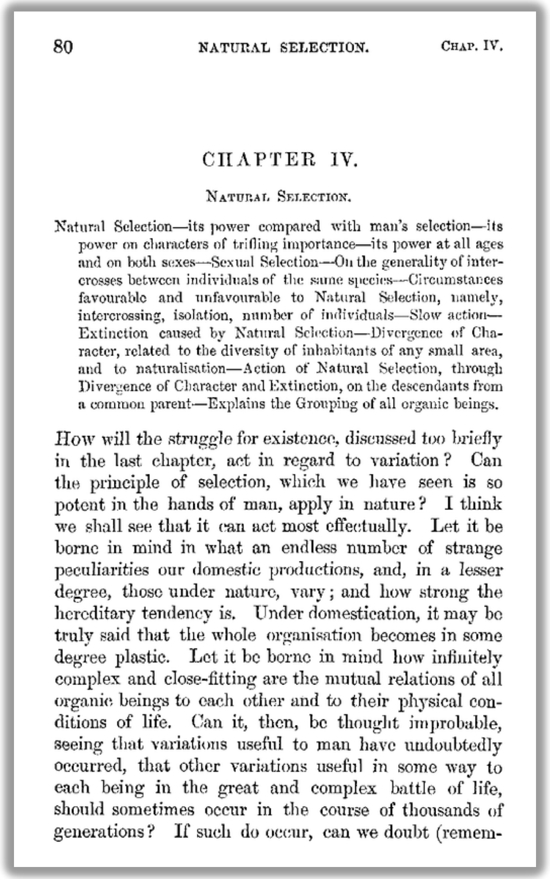
Structured Data is typically used by Relational Database Management Systems (RDBMSs) such as Access, Oracle, SQL Server, MySQL.

Unstructured data





	A	B	C	D	E
1	My Holiday Budget ----->				
2					
3	Flights	$10,000.00			
4	Accommodation	$5,000.00	<- can this be reduced?		
5	Food	$1,000.00			
6					
7		**$16,000.00**	TOTAL		
8					
9					
10					



80 NATURAL SELECTION. CHAP. IV.

CHAPTER IV.

NATURAL SELECTION.

Natural Selection—its power compared with man's selection—its power on characters of trifling importance—its power at all ages and on both sexes—Sexual Selection—On the generality of intercrosses between individuals of the same species—Circumstances favourable and unfavourable to Natural Selection, namely, intercrossing, isolation, number of individuals—Slow action—Extinction caused by Natural Selection—Divergence of Character, related to the diversity of inhabitants of any small area, and to naturalisation—Action of Natural Selection, through Divergence of Character and Extinction, on the descendants from a common parent—Explains the Grouping of all organic beings.

How will the struggle for existence, discussed too briefly in the last chapter, act in regard to variation? Can the principle of selection, which we have seen is so potent in the hands of man, apply in nature? I think we shall see that it can act most effectually. Let it be borne in mind in what an endless number of strange peculiarities our domestic productions, and, in a lesser degree, those under nature, vary; and how strong the hereditary tendency is. Under domestication, it may be truly said that the whole organisation becomes in some degree plastic. Let it be borne in mind how infinitely complex and close-fitting are the mutual relations of all organic beings to each other and to their physical conditions of life. Can it, then, be thought improbable, seeing that variations useful to man have undoubtedly occurred, that other variations useful in some way to each being in the great and complex battle of life, should sometimes occur in the course of thousands of generations? If such do occur, can we doubt (remem-

- handwritten notes
- printed books
- spreadsheets
- etc

Unstructured Data

Unstructured data is not organised in a pre-defined manner.

The organisation **does not know the format,** nor the content of the data in advance.

Consider data sourced from **social media, email,** etc. The contents are unpredictable.

- The data may contain text, audio, video, links, images. One item may include many data about many organizational functions.

How do organizations store such data so that it can be retrieved, collated, analysed?

We will deal with the topic of unstructured data later in **future weeks**.

Semi-Structured Data

Semi-structured data is information that doesn't match the requirements of a relational database.

The data is organized / arranged that makes it easier to analyze.

Examples of semi-structured data include **XML** documents and **NoSQL** databases.

We will briefly deal with the topic of semi-structured data in future weeks

Structured Data

Relational Database Management Systems require data to be stored in a very structured way.

These systems deal with data that has a **repetitive pattern or format**.

Consider **Student** data stored in a University. While every student is different, the university want to store data in the same format for every student. Data Types are also specified for each piece of information

- Student ID – Numeric/Digits
- Student Name – Alpha
- Gender – Alpha
- DateOfBirth – Date

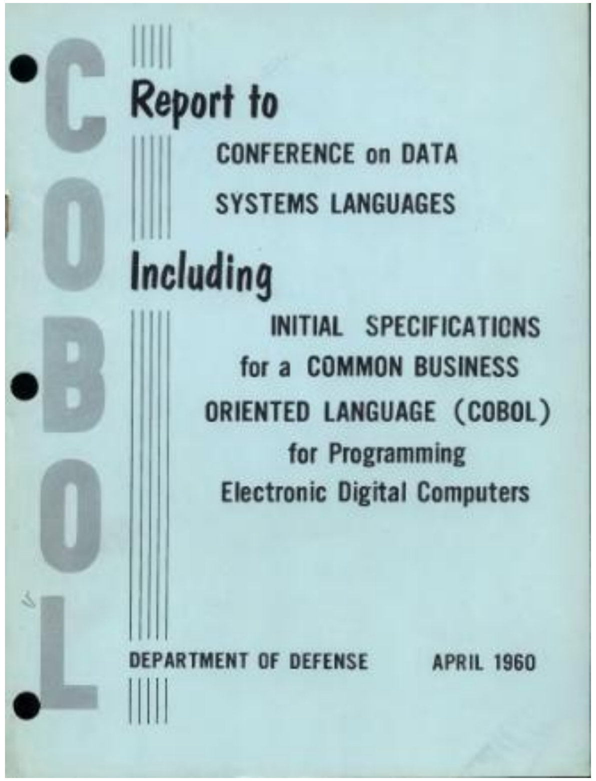
HomeAddress - Alphanumeric

PhoneNo – Numeric or Digits+space+ brackets

NextOfKin - Alpha

File Processing

```
DATA DIVISION.
FILE SECTION.
FD StudentFile.
01 StudentRec.
   88  EndOfStudentFile  VALUE HIGH-VALUES.
   02  StudentId         PIC 9(7).
   02  StudentName.
       03 Surname        PIC X(8).
       03 Initials       PIC XX.
   02  DateOfBirth.
       03 YOBirth        PIC 9(4).
       03 MOBirth        PIC 9(2).
       03 DOBirth        PIC 9(2).
   02  CourseCode        PIC X(4).
   02  Gender            PIC X.
```



Problems with flat-files:

- Data access routines must be programmed in detail
- Each program must include full detail of data structure
- Multiple users cannot simultaneously access data
- Multiple copies of data - not centrally managed

http://groups.engin.umd.umich.edu/CIS/course.des/cis400/cobol/cobol.html

Why use a DBMS?

DBMS = **D**ata**B**ase **M**anagement **S**ystem

Data independence

- Applications should not be exposed to data representation and storage
- DBMS provides an abstract view that hides representation & storage

Efficient access

- More efficient data storage and retrieval than flat files

Data integrity and security

- DBMS enforces data integrity constraints, access controls and govern user access
- Not reliant on just the operating system

Why use a DBMS?

Uniform data administration

- Specialist skills in data management and administration
- Layer of expertise reduces risk to data and data owners

Concurrent access and crash recovery

- Schedules concurrent access. Protects data from system failures

Datasets increasing in diversity and volume

Data Independence

- Logical data independence: Protection from changes in logical structure of data.
- Physical data independence: Protection from changes in physical structure of data

Reduced Application Development time*

Ramakrishnan and Gehrke 3rd Edition, p.9

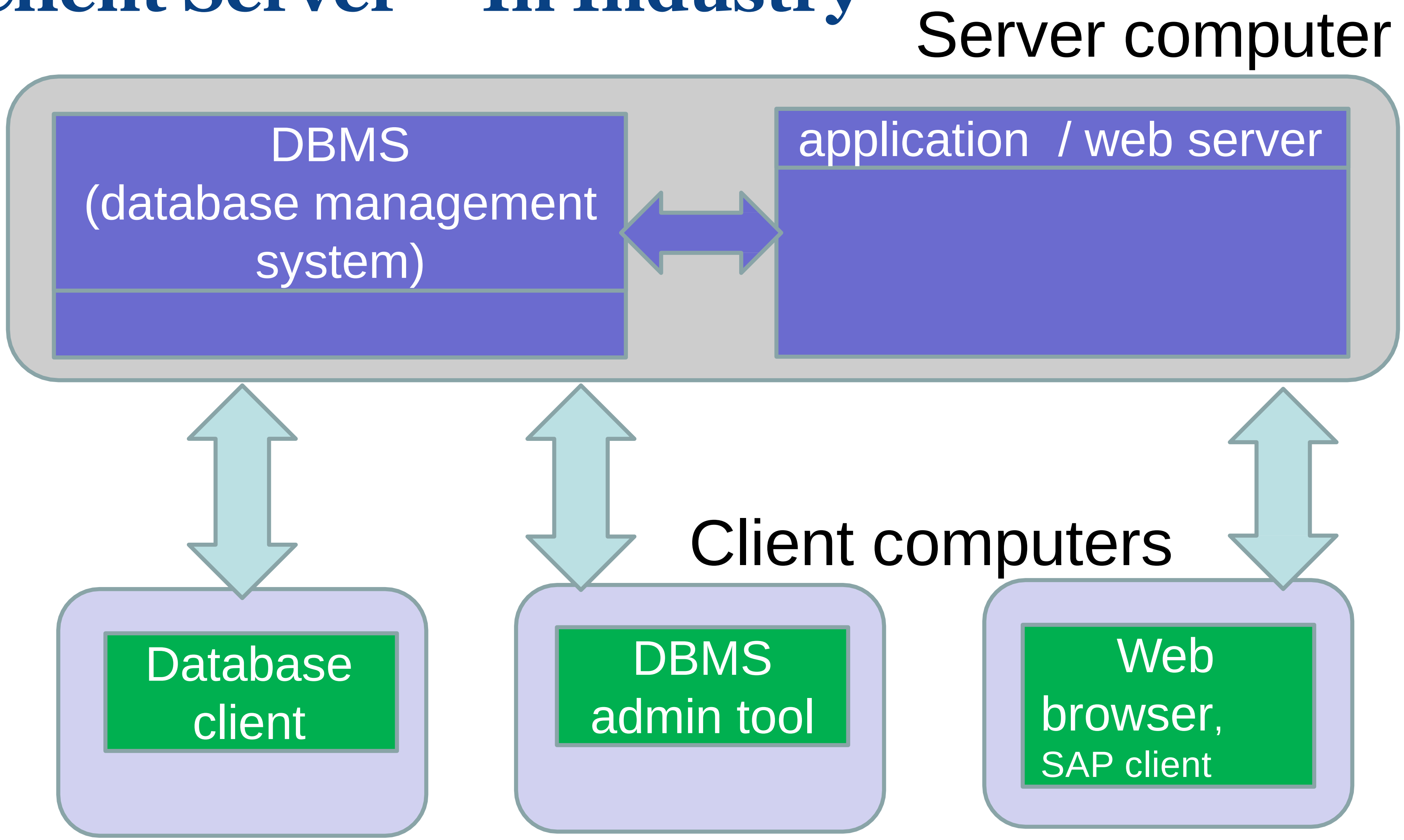
Relational Databases

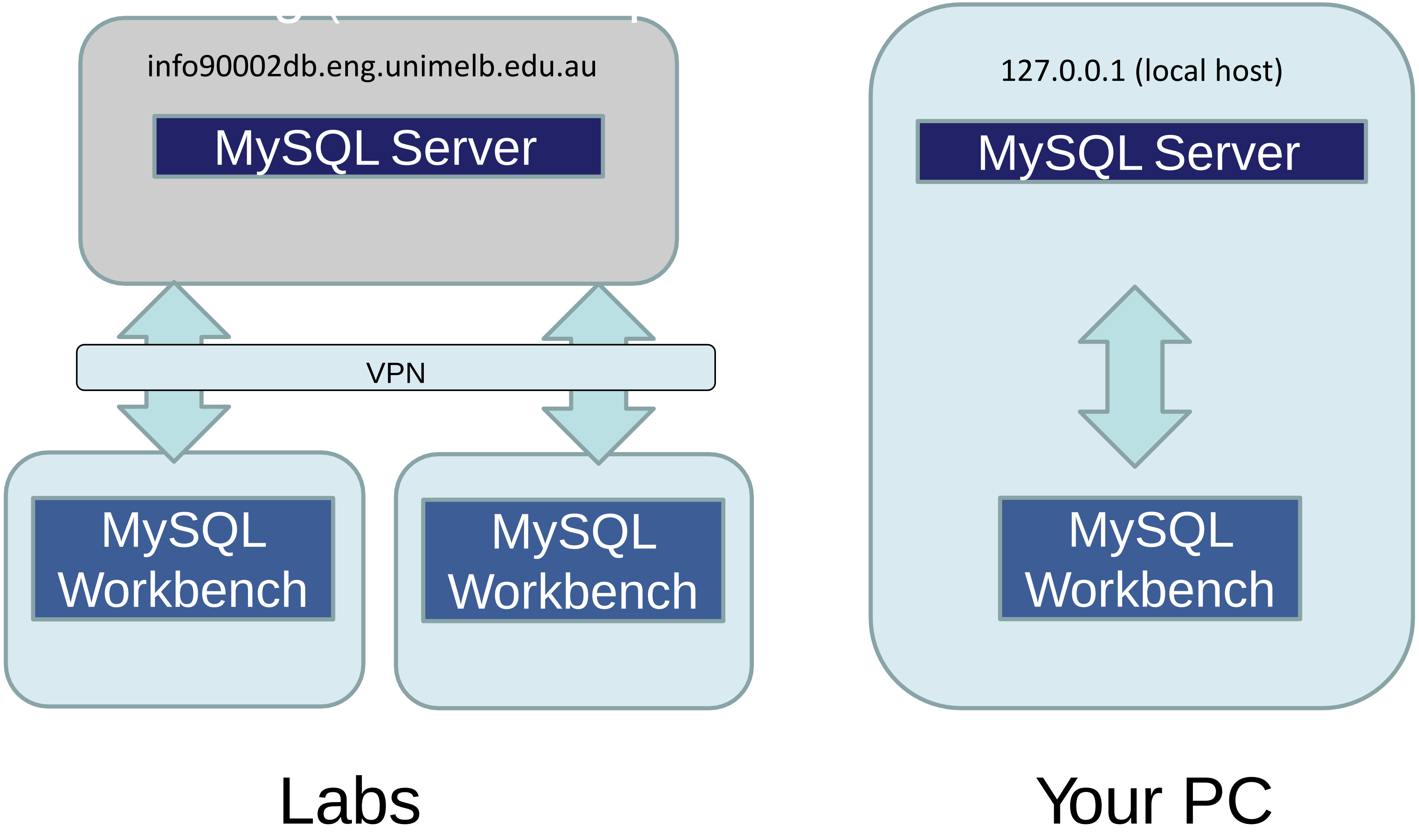
- The first relational databases from Oracle and IBM appear around 1980
- Others appear later

Client-Server Architecture

Introduction to MySQL Server and MySQL Workbench

Client Server – In Industry




info90002db.eng.unimelb.edu.au
MySQL Server
VPN
MySQL Workbench
MySQL Workbench
127.0.0.1 (local host)
MySQL Server
MySQL Workbench
Labs
Your PC

Relational databases

Basic concepts

Relational Data Model

RDBMSs are based on the Relational Data Model

- Developed by Ted Codd in 1970.
- Data is represented in the form of two-dimensional **tables**.

Each two-dimensional table has the following properties:

- A set of uniquely named **Columns / Attributes**
- A list of unnamed/unnumbered **Rows**
- The **order** of the rows is **irrelevant**.

A **Row** consists of a sequence of **Attributes**

- **One cell** for each Attribute
- Only **one value per cell** is allowed.

Relational Data Model (cont.)

Table

Uniquely named columns called Attributes

Attribute

Unnamed and
unnumbered Rows

Row

ID	StName	Degree	Enrolled
1001	Fred Blogs	615	20/01/2020
1004	Emma Jevs	308	03/02/2019
1006	Dave Rigg	234	10/07/2020

The order of the rows is irrelevant.

Only one value per cell allowed.

Relational Databases

A relational database is a collection of related tables

Example:

- **Student Table** (stores data about students)
- **Subject Table** (stores data about university subjects)
- **Enrolment Table** (stores data about a student has enrolled into a specific subject)

RDBMS

A RDBMS is a collection of programs that allow developers / users to store & retrieve data from relational databases

It allows users to perform CRUD (create, read, update and delete) operations on data in the tables. E.g.

- **Create** a student record
- **Retrieve** the student's details
- **Update** the student's details
- **Delete** the student from a table

Setting up a RDBMS

Tables

- Follow a 2 dimensional structure
- Each row of data is identified by a unique Primary Key
- No duplicates, e.g. Student ID

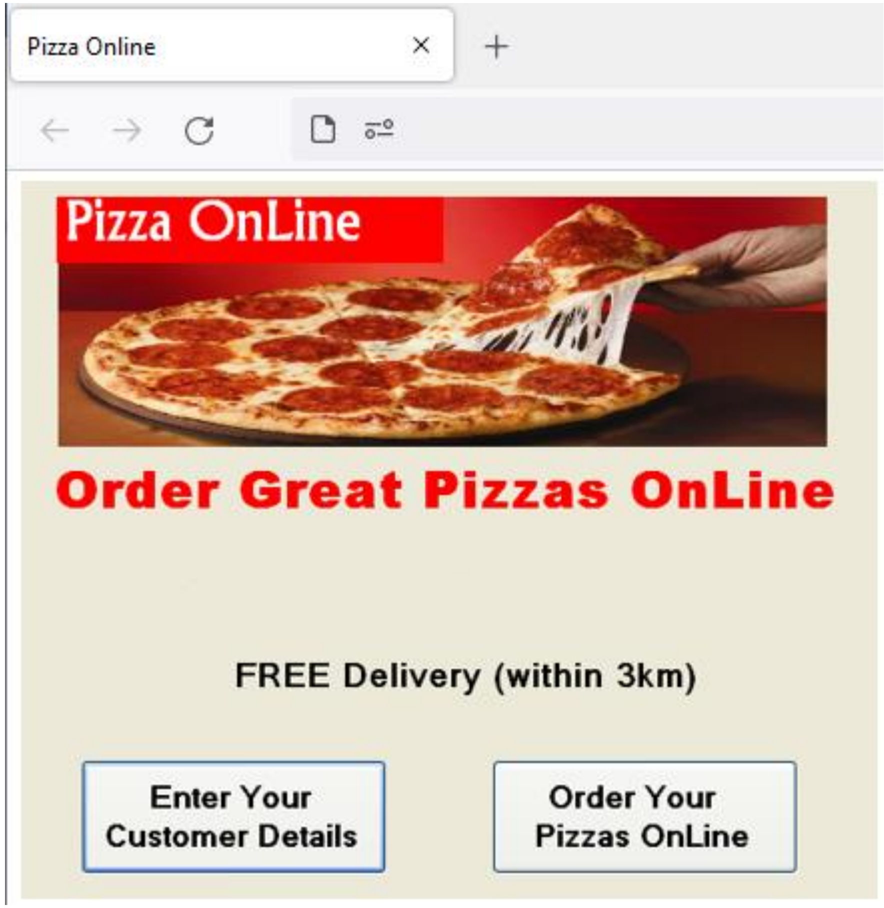
Constraints can be added to validate data

- Student ID is correct length
- Student type is PG or UG (post or undergraduate)
- Student is enrolled in a degree that actually exists

Case Study - Overview

Let's consider a business called **Pizza OnLine**

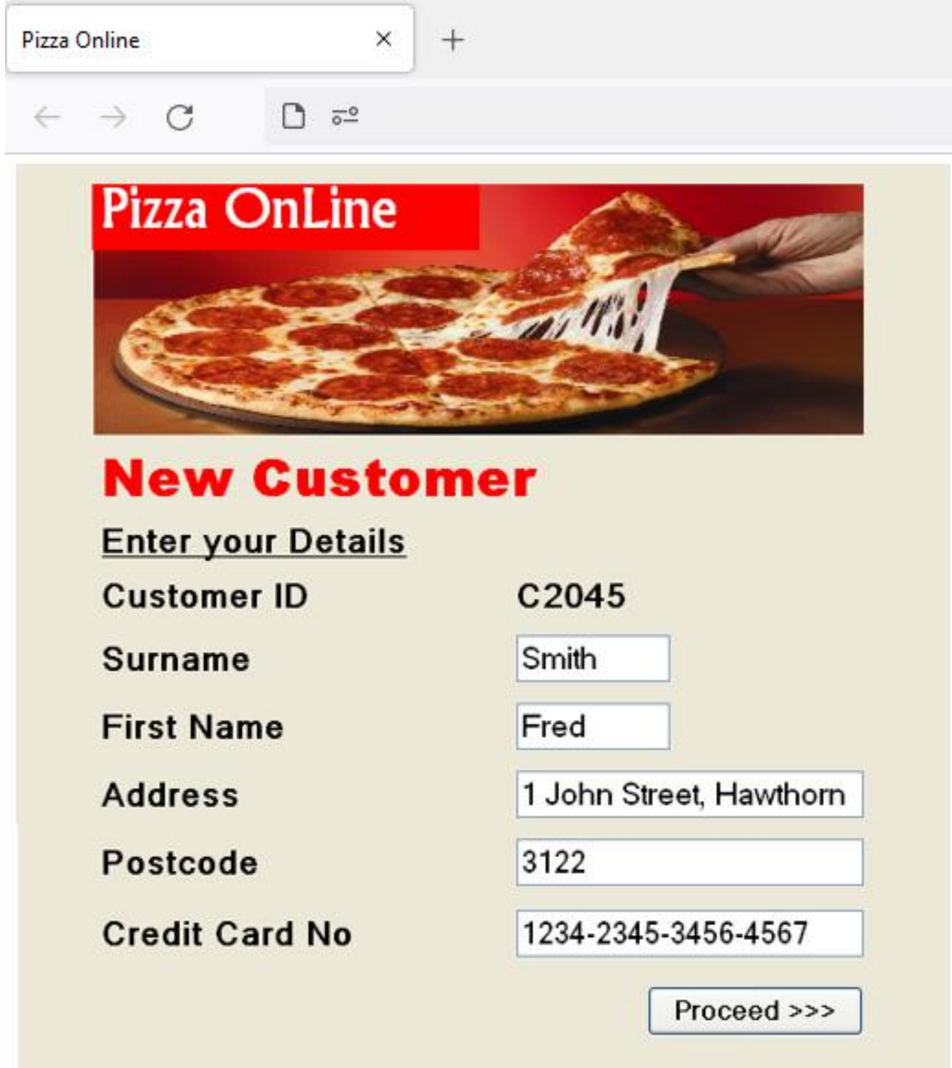
- It allows customers to **order pizzas** via the online portal.
- Customers have pizzas delivered.
- Only people who register with a credit card number can be an on-line customer.



Case Study - Customer Details

Customers need to supply personal details:

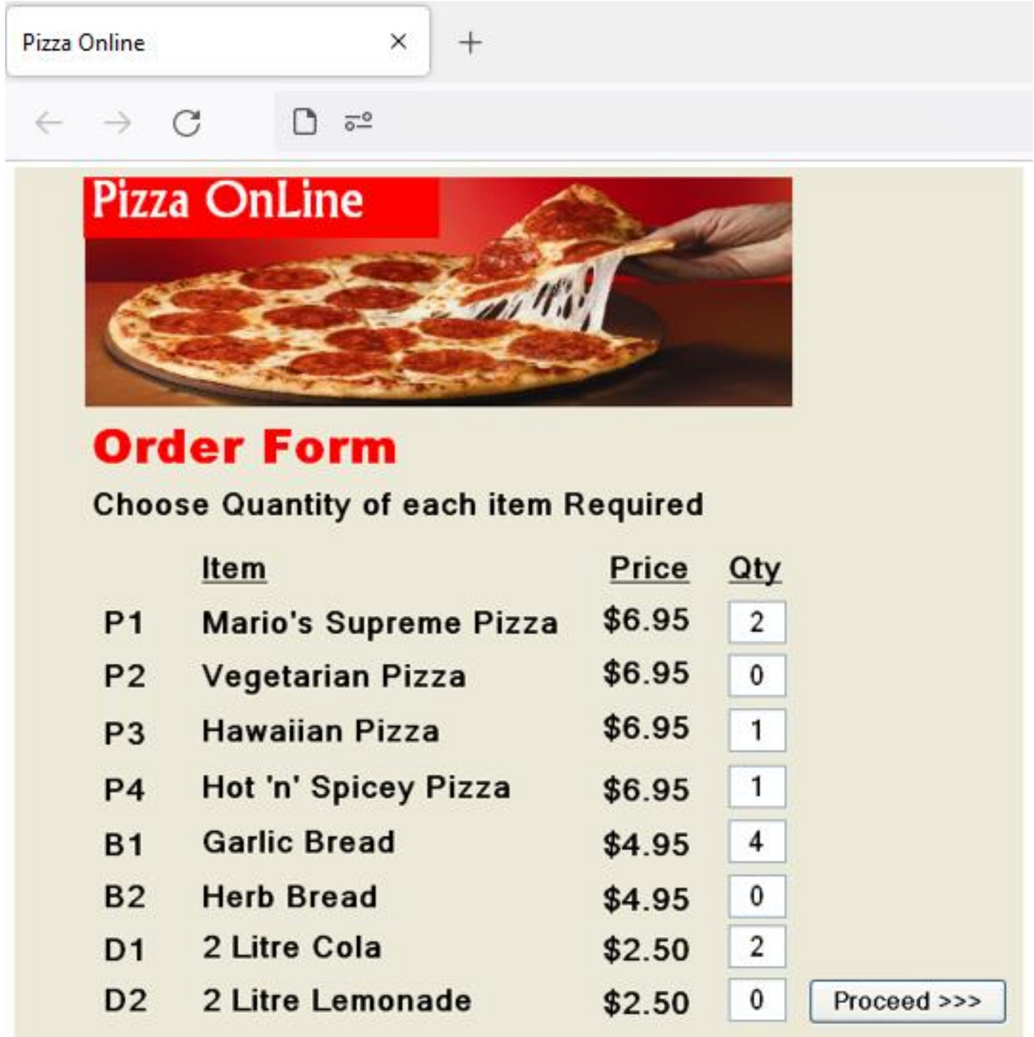
- Name
- Address
- Credit Card No



Case Study – Order Details

The Order Form

- This is **One order**.
- The customer may add **multiple items** and **different** quantities.


Pizza Online

Pizza OnLine

Order Form

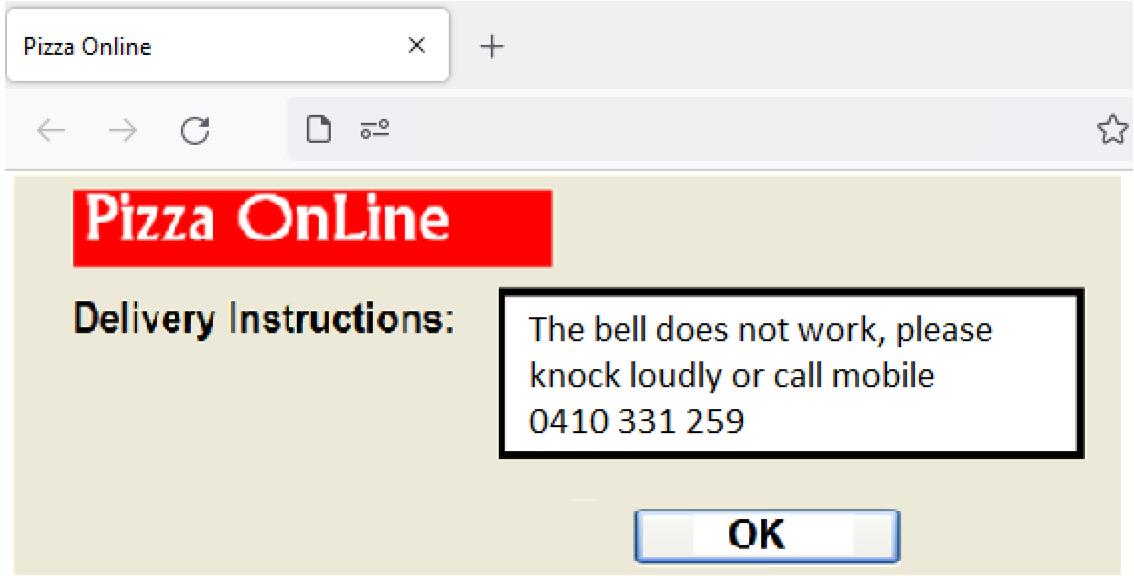
Choose Quantity of each item Required

	Item	Price	Qty
P1	Mario's Supreme Pizza	$6.95	2
P2	Vegetarian Pizza	$6.95	0
P3	Hawaiian Pizza	$6.95	1
P4	Hot 'n' Spicey Pizza	$6.95	1
B1	Garlic Bread	$4.95	4
B2	Herb Bread	$4.95	0
D1	2 Litre Cola	$2.50	2
D2	2 Litre Lemonade	$2.50	0

Proceed >>>

Case Study - Customer Details

Add any **delivery instructions** that are required for the order:



Case Study – The completed Order

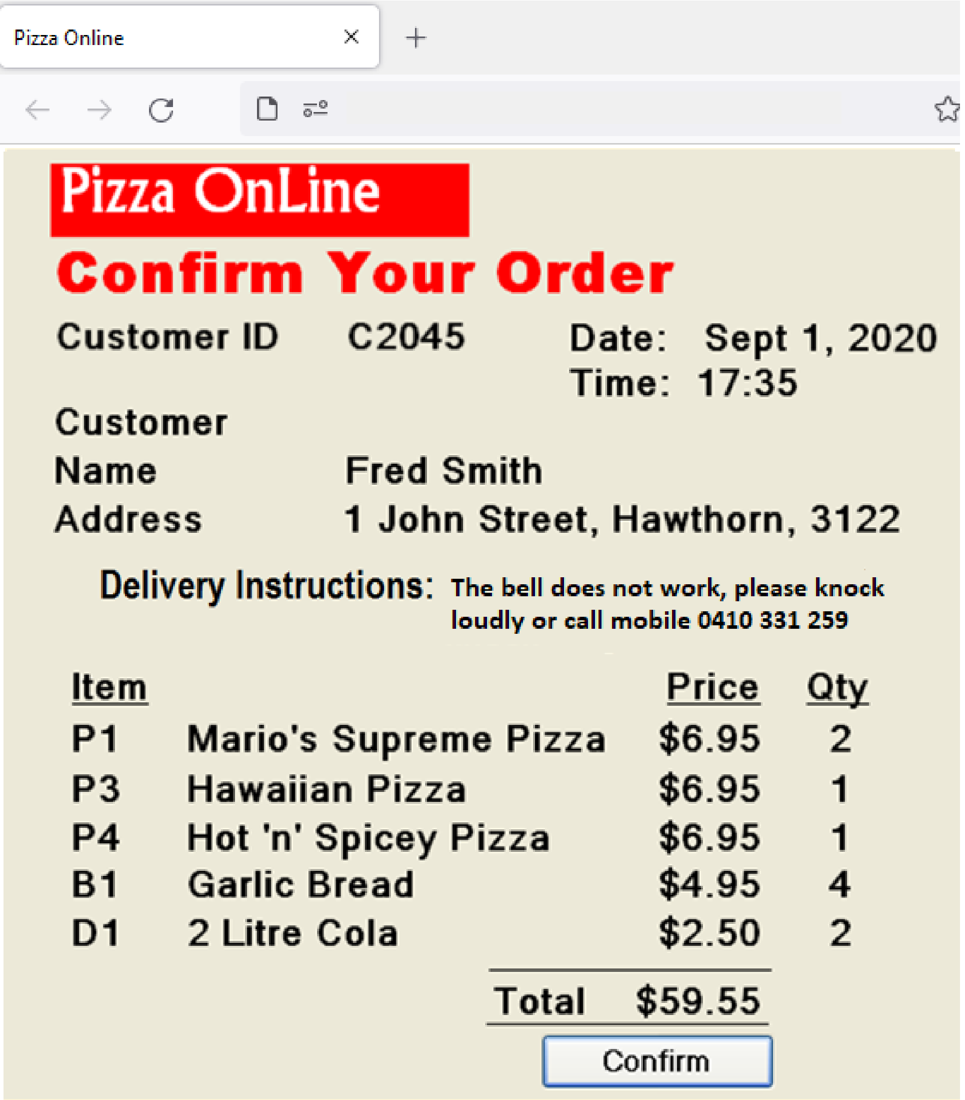
This Order is for Customer **C2045**.

Order Date & Time is **Sept 1, 2020 17:35**

This order has special **delivery instructions**

The order is for many **different Items.**

Each item ordered has a **Quantity** value



Tables in the Pizza Online Database

Customer ID	Surname	First name	Address	Postcode	Credit card
C2045	Smith	Fred	1 John St, Hawthorn	3122	1234234534564567
C2048	Nguyen	Vincent	2/7 Oak Ave, Altona	3018	4554123423457899
C2146	Davis	Liz	32 Lyle St, Toorak	3142	4564564578970022

Code	Pizza name	Price
P1	Mario's Supreme Pizza	6.95
P2	Vegetarian Pizza	6.95
P3	Hawaiian Pizza	6.95
P4	Hot 'n' spicy Pizza	6.95
B1	Garlic Bread	4.95
B2	Herb Bread	4.95
D1	2 Litre Cola	2.50
D2	2 Litre Lemonade	2.50

Order No	Line Item	Quantity
3224	P1	2
3224	P3	1
3224	P4	1
3224	B1	4
3224	D1	2

Order No	Customer ID	Date	Time	Delivery Instructions	Total
3224	C2045	1/09/2021	17:35	The bell does not work,	59.55

Manipulating Tables Structure

Customer ID	Surname	First name	Address	Postcode	Credit card
C2045	Smith	Fred	1 John St, Hawthorn	3122	1234234534564567
C2048	Nguyen	Vincent	2/7 Oak Ave, Altona	3018	4554123423457899
C2146	Davis	Liz	32 Lyle St, Toorak	3142	4564564578970022

Code	Pizza name	Price
P1	Mario's Supreme Pizza	6.95
P2	Vegetarian Pizza	6.95
P3	Hawaiian Pizza	6.95
P4	Hot 'n' spicy Pizza	6.95
B1	Garlic Bread	4.95
B2	Herb Bread	4.95
D1	2 Litre Cola	2.50
D2	2 Litre Lemonade	2.50

Working with tables is like working with files – there are 4 things you can do:

CREATE a table

DROP (i.e. delete) a table

ALTER a table (e.g. add a column)

RENAME a table

Manipulating Table Contents

Customer ID	Surname	First name	Address	Postcode	Credit card
C2045	Smith	Fred	1 John St, Hawthorn	3122	1234234534564567
C2048	Nguyen	Vincent	2/7 Oak Ave, Altona	3018	4554123423457899
C2146	Davis	Liz	32 Lyle St, Toorak	3142	4564564578970022

Code	Pizza name	Price
P1	Mario’s Supreme Pizza	6.95
P2	Vegetarian Pizza	6.95
P3	Hawaiian Pizza	6.95
P4	Hot ‘n’ spicy Pizza	6.95
B1	Garlic Bread	4.95
B2	Herb Bread	4.95
D1	2 Litre Cola	2.50
D2	2 Litre Lemonade	2.50

For each table you need to be able to:

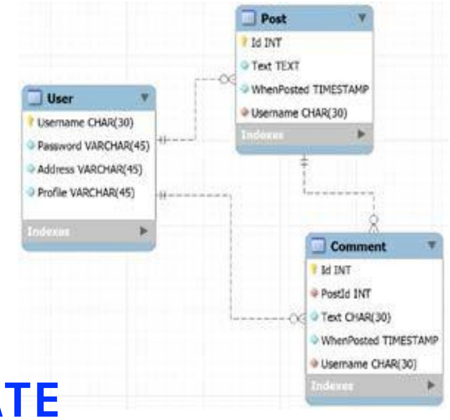
SELECT, or read, data from the table

INSERT new rows into the table

DELETE existing rows from the table

UPDATE existing rows in the table

Database lifecycle



- Design the database
  - data modelling, E-R diagrams
- Implement the database
  - data definition language DDL

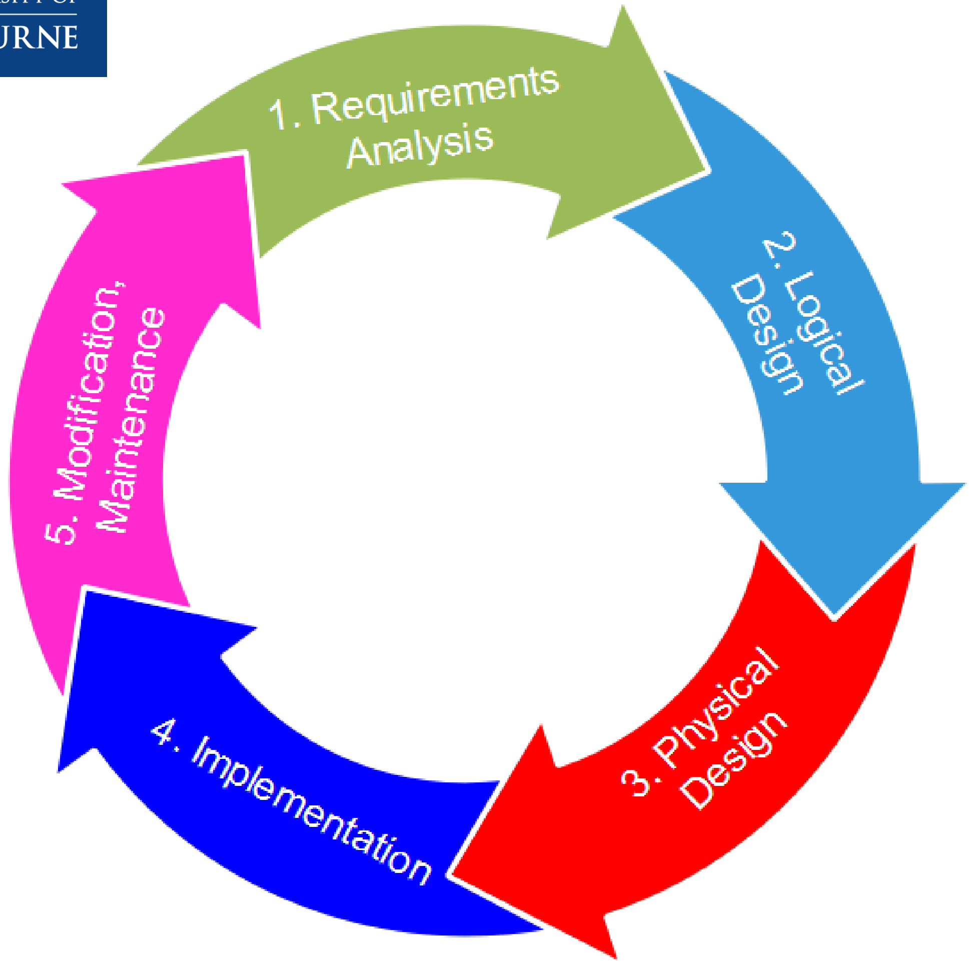
- CREATE
- DROP
- ALTER
- RENAME

- Data access / programming
  - data manipulation language DML

- SELECT
- INSERT
- UPDATE
- DELETE

- Database administration
  - data control language DCL

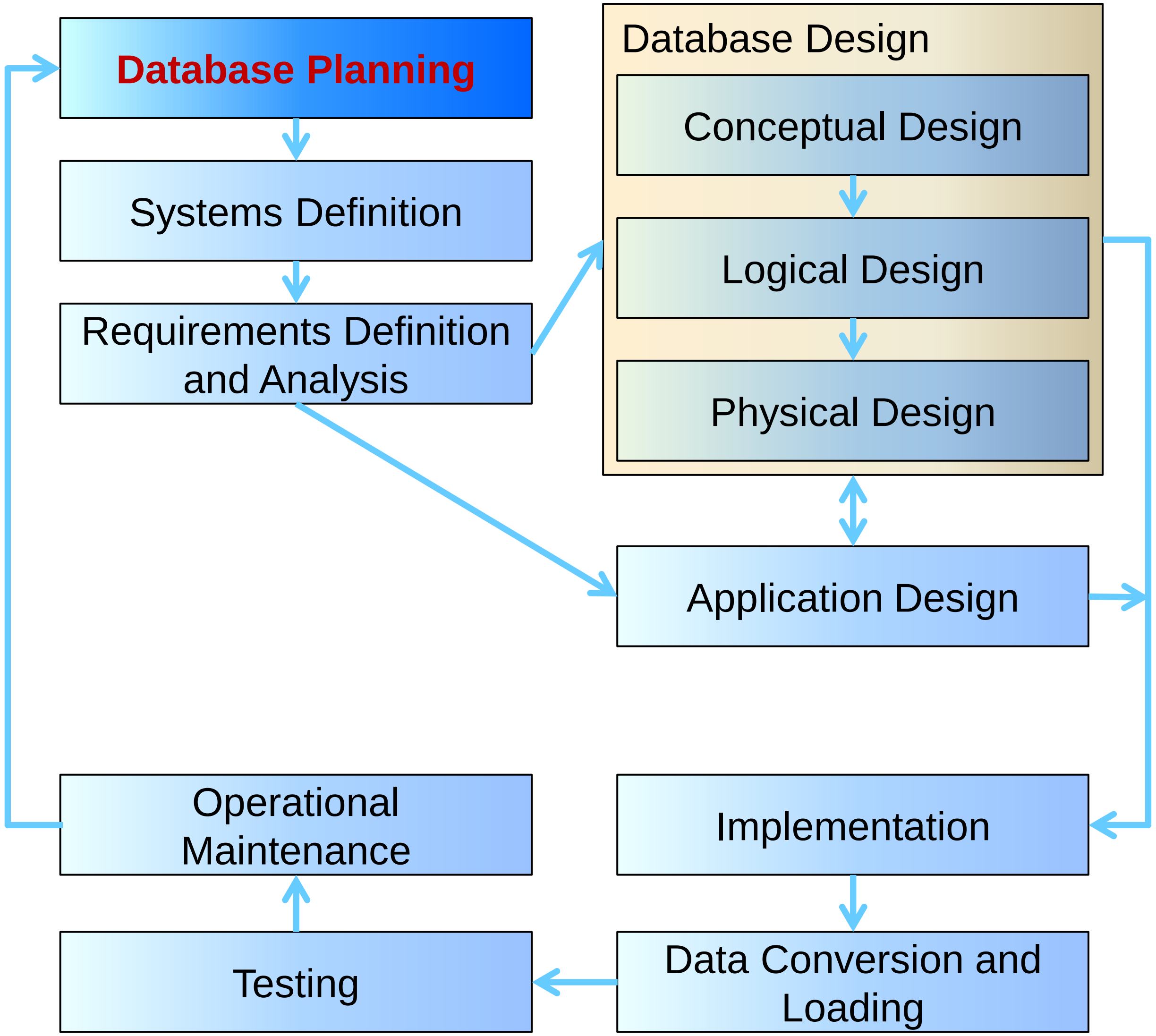
- GRANT
- REVOKE



Database Development Lifecycle

Part of system development lifecycle

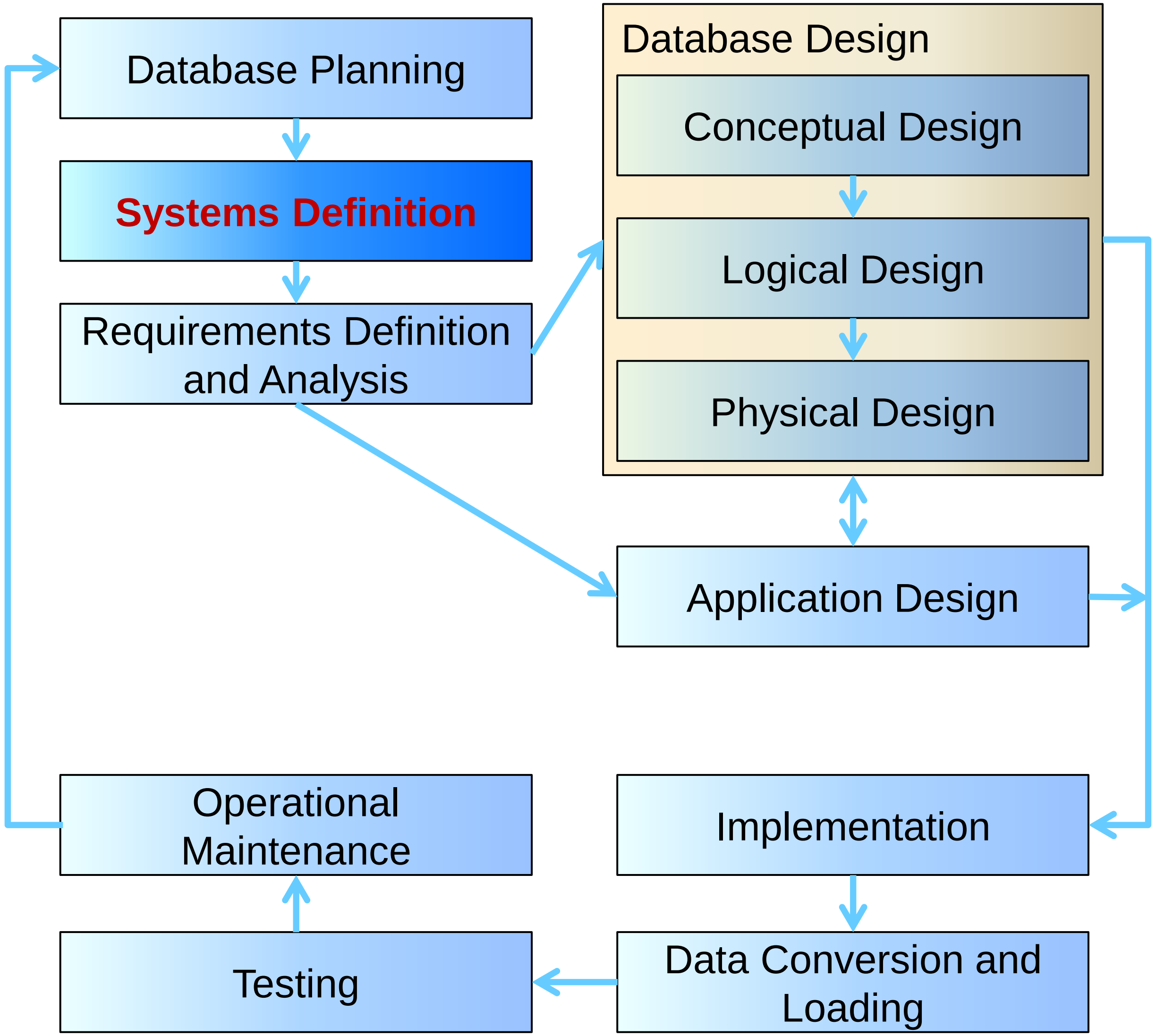
Database Development Lifecycle



- 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

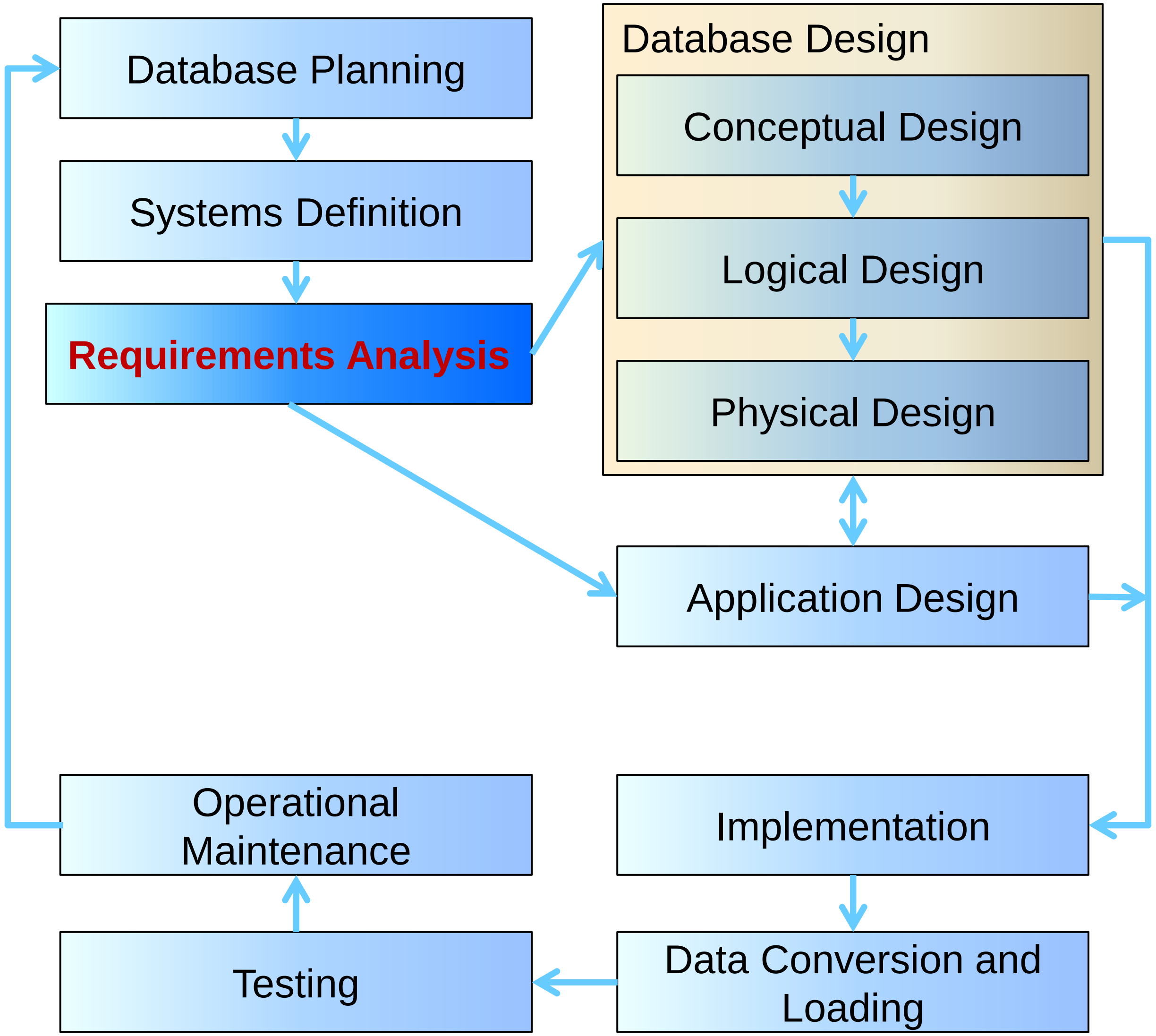
Database Development Lifecycle



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

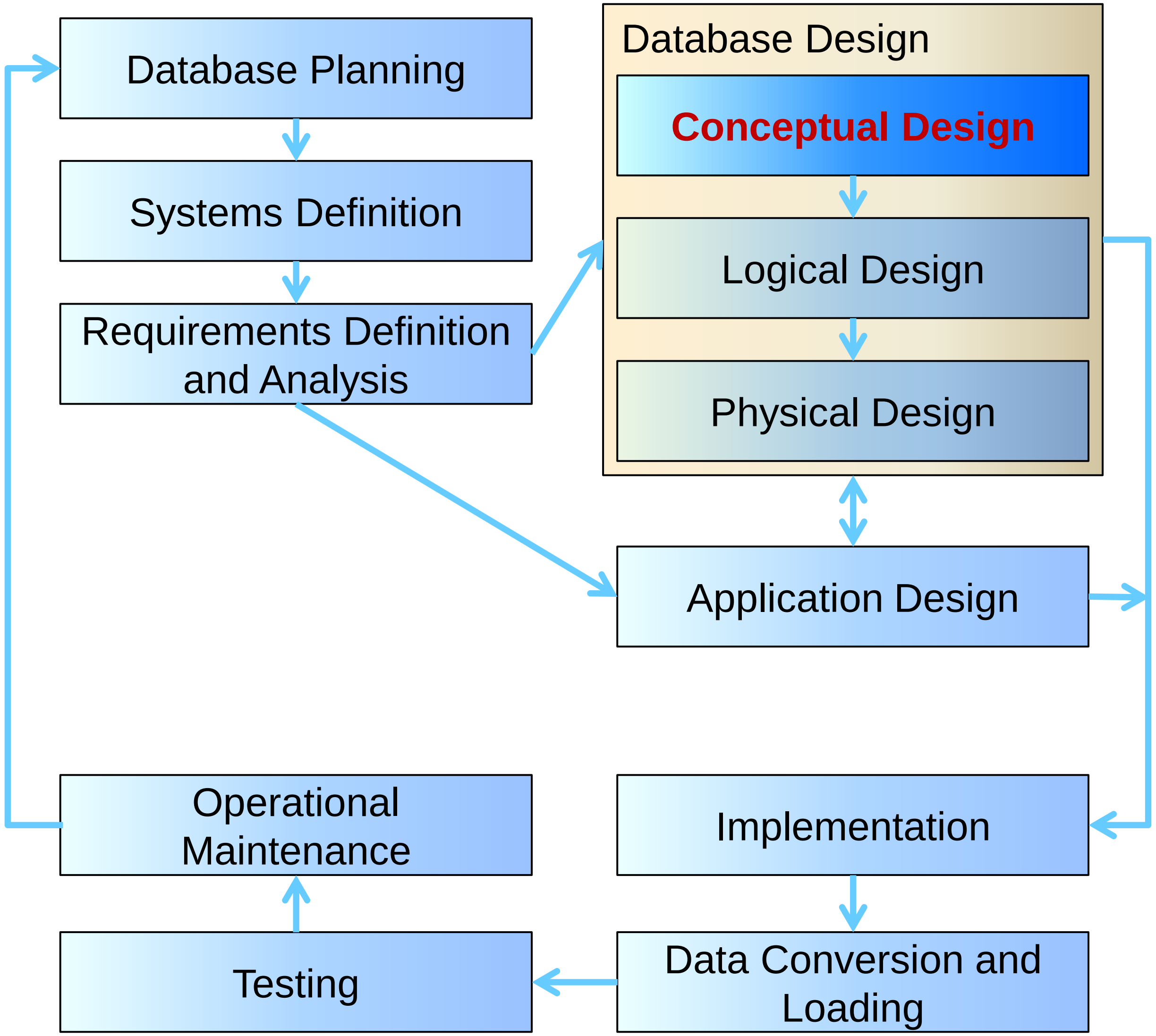
- Outside scope of the course (slightly)

Database Development Lifecycle



- Collection and analysis of requirements for the new system

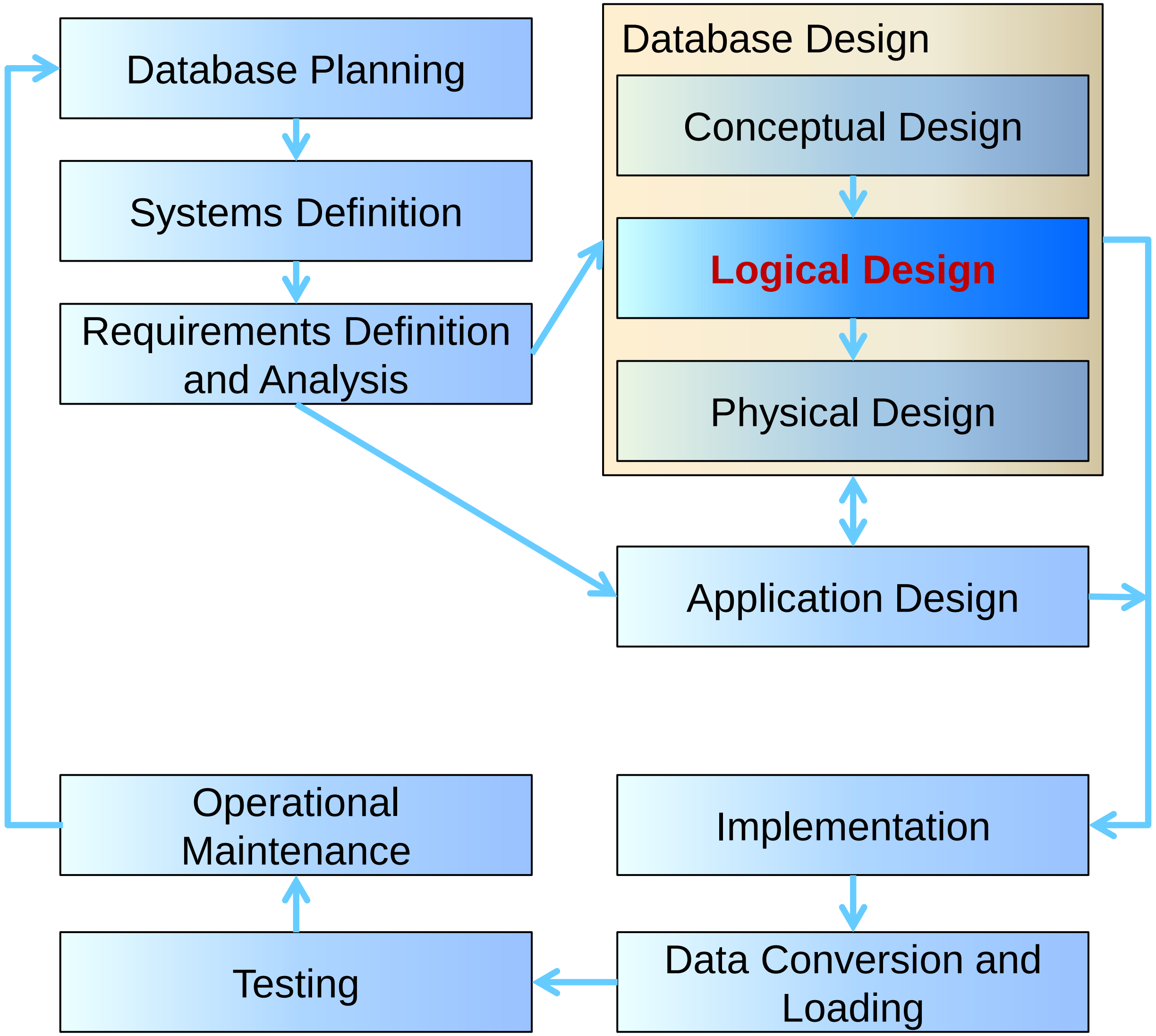
Database Development Lifecycle



- High-level, first-pass model of entities and their connections
- Typically omits attributes*
- Could potentially be implemented in a non-relational database
- Thus can include many-to-many relationships, repeating groups, composite attributes

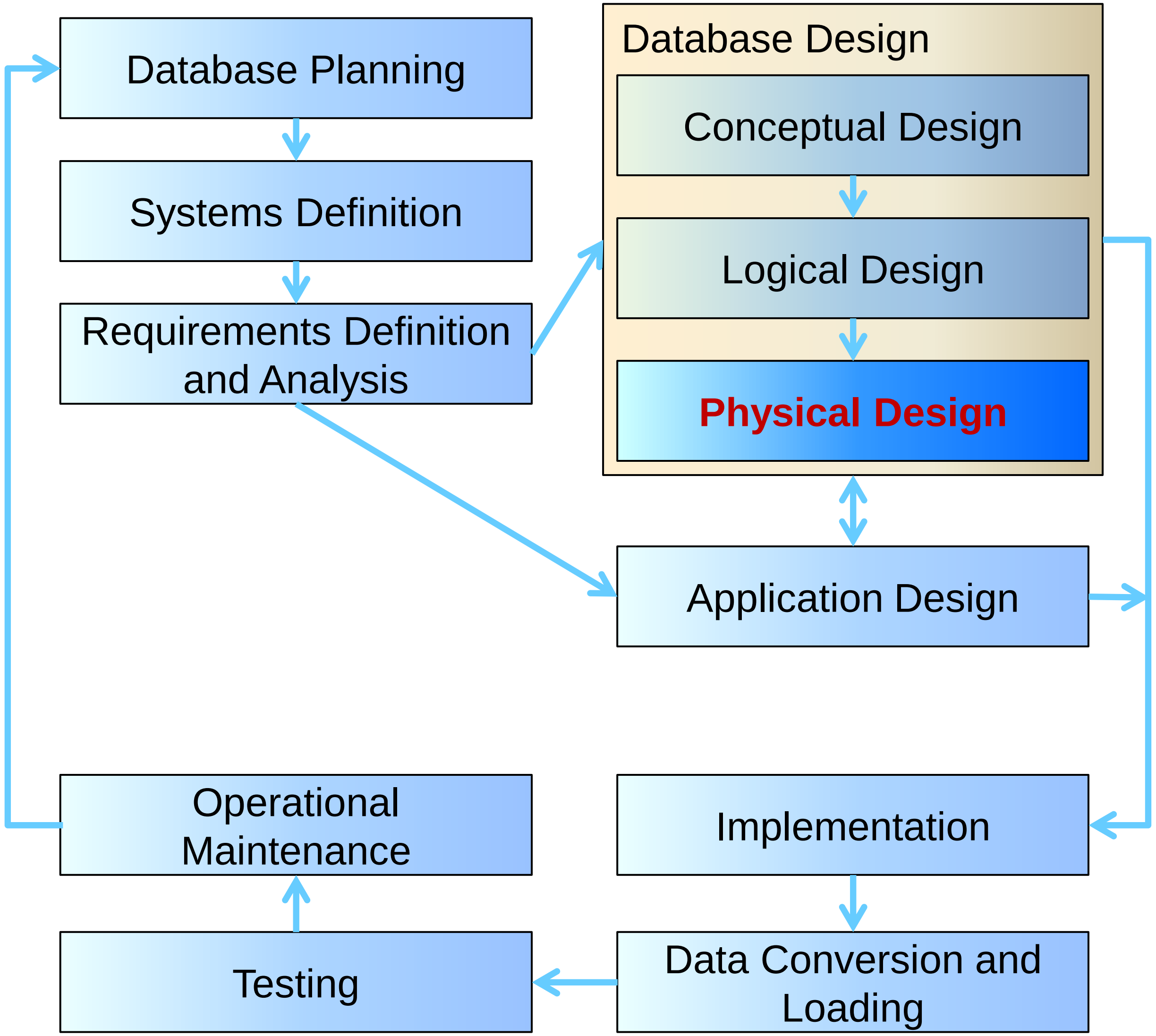
* Typically we list only attributes in the case study

Database Development Lifecycle



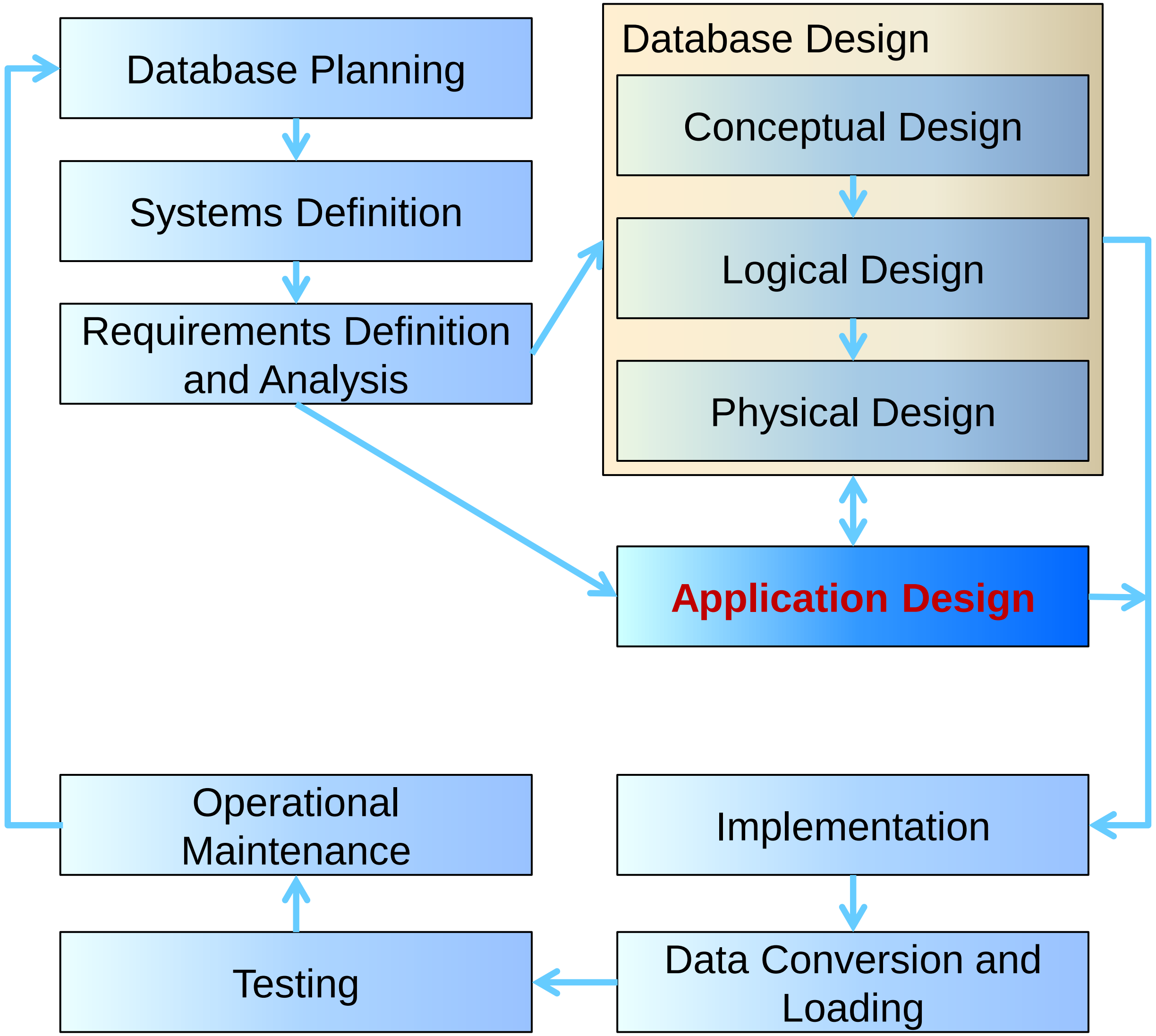
- Builds on the conceptual design
- Designing now for a relational database
- Includes columns and keys
- Independent of a specific vendor and other physical considerations

Database Development Lifecycle



- Implements the logical design for a specific DBMS.
- Describes:
  - Base tables
  - Data types
  - Indexes
  - Integrity constraints
  - File organisation
  - Security measures
- We will cover some aspects of physical design

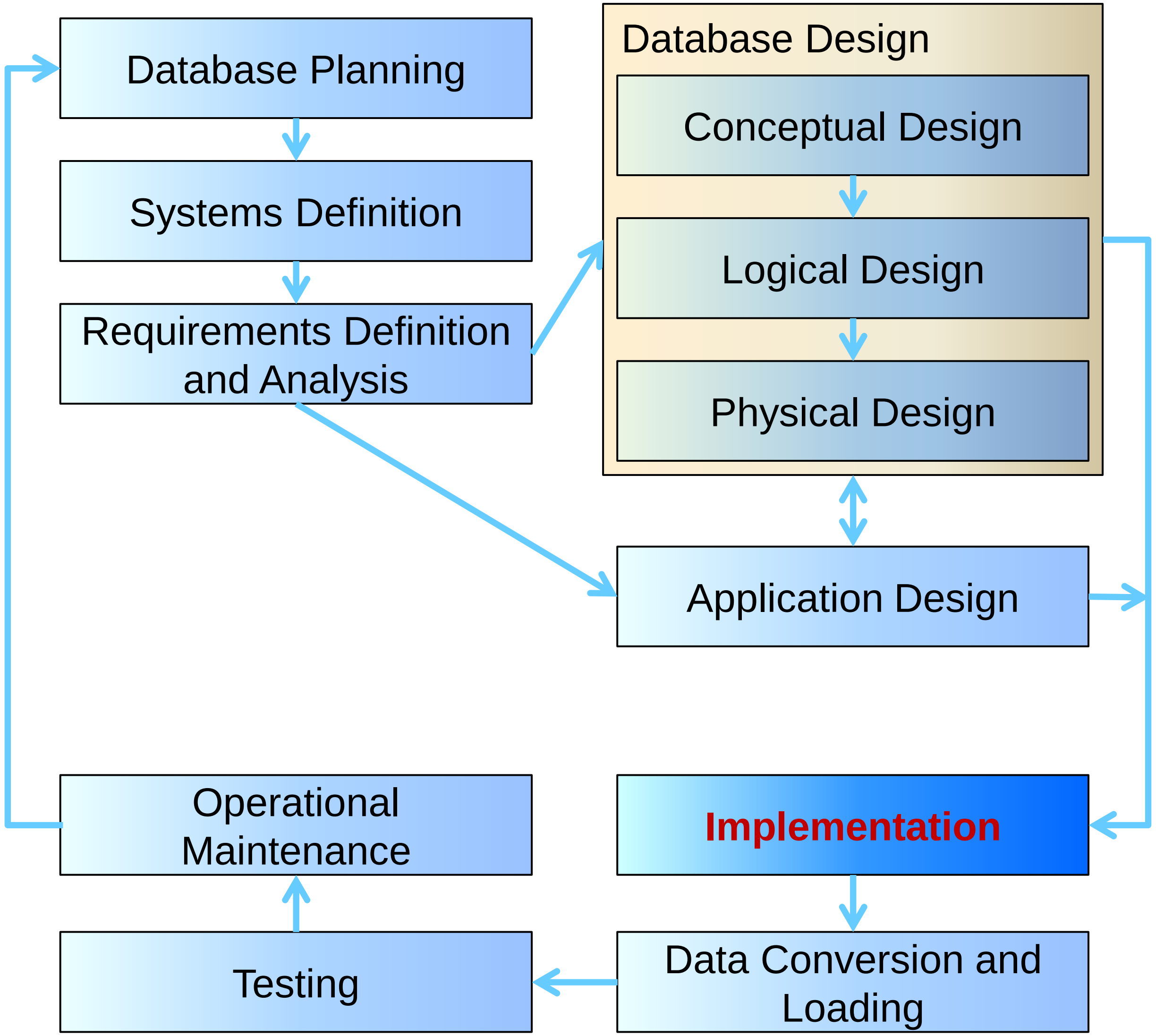
Database Development Lifecycle



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

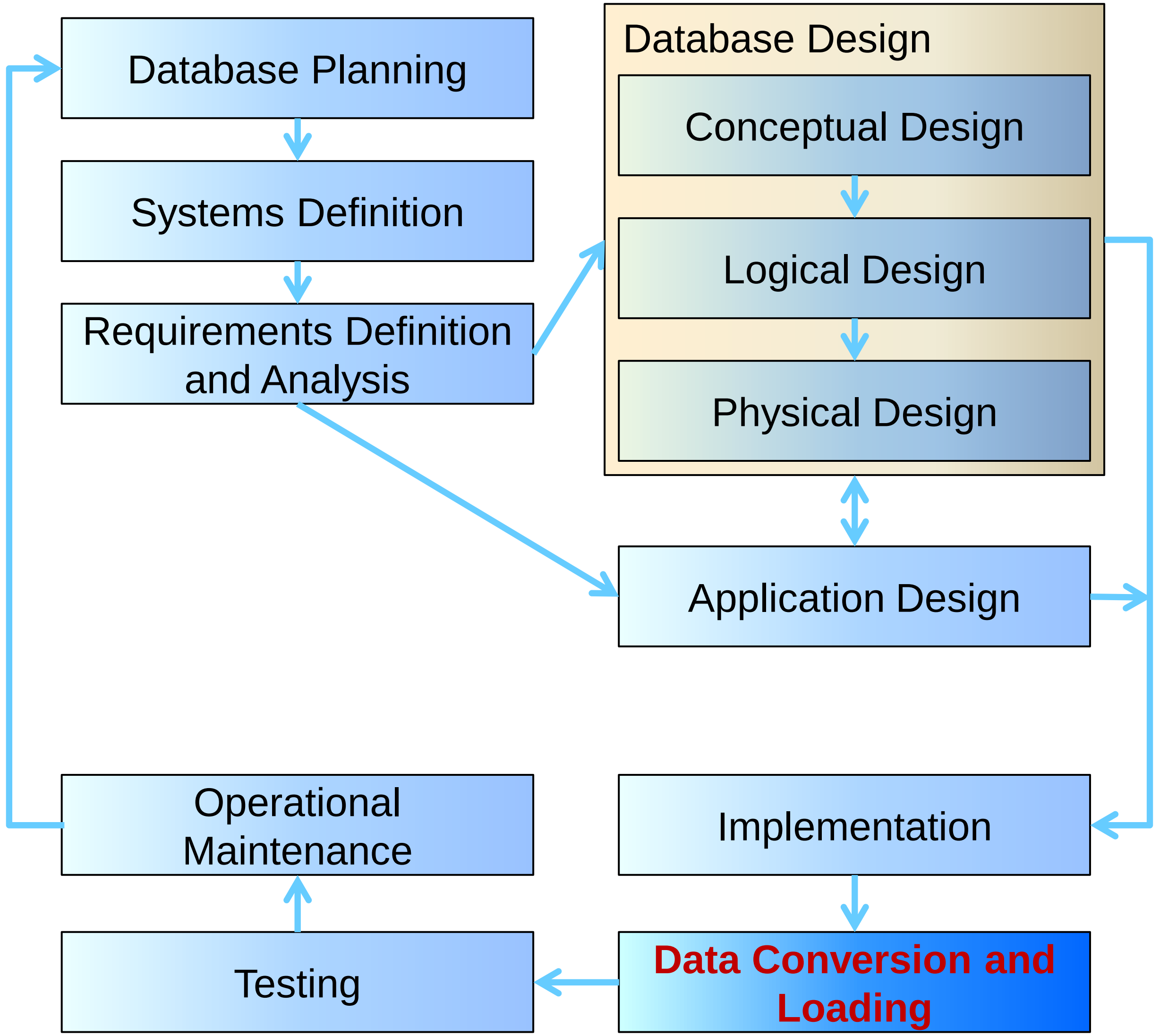
- Mostly outside scope of the course, but discussed in lecture 11

Database Development Lifecycle



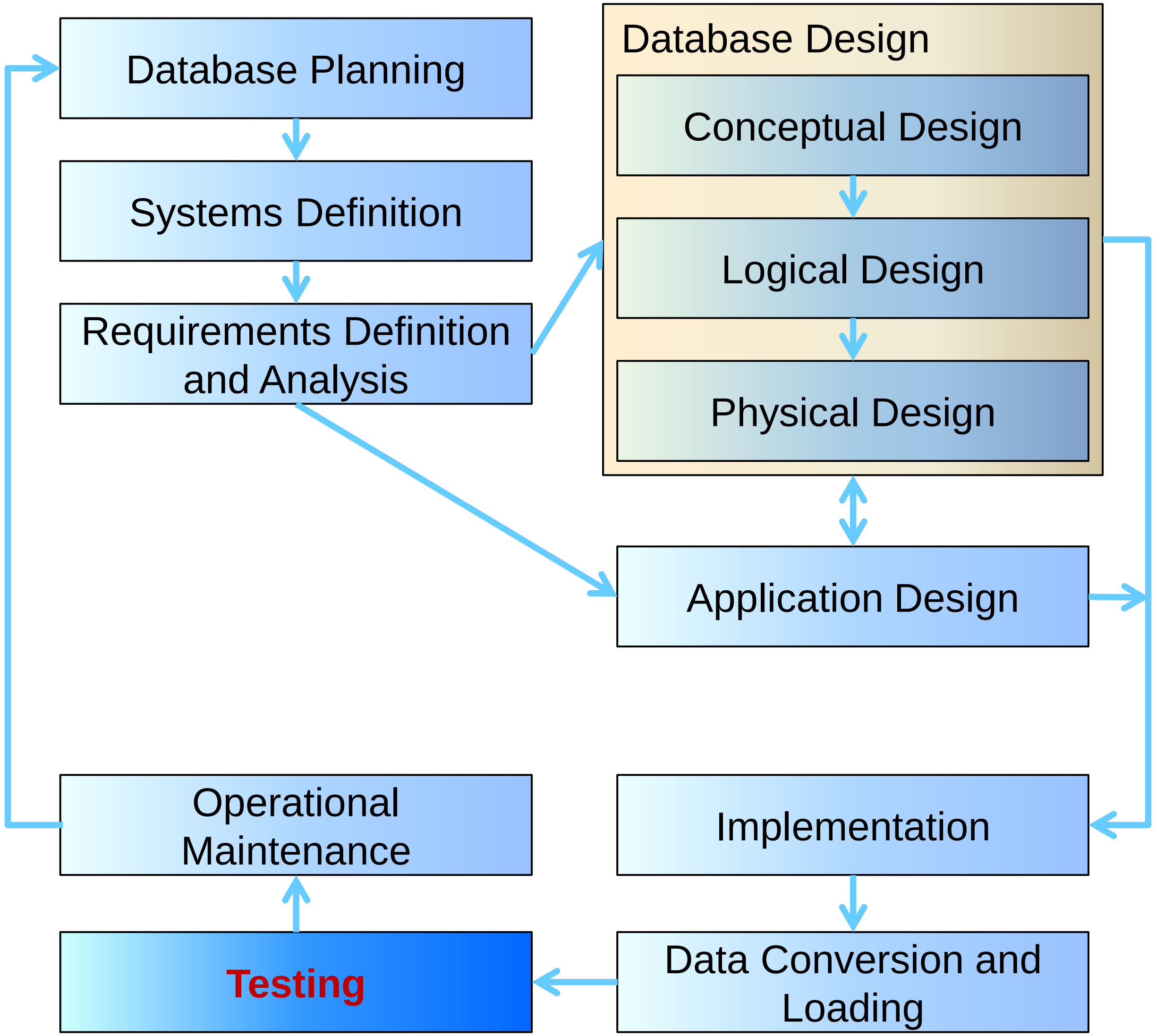
- Implementation of the design as a working database

Database Development Lifecycle



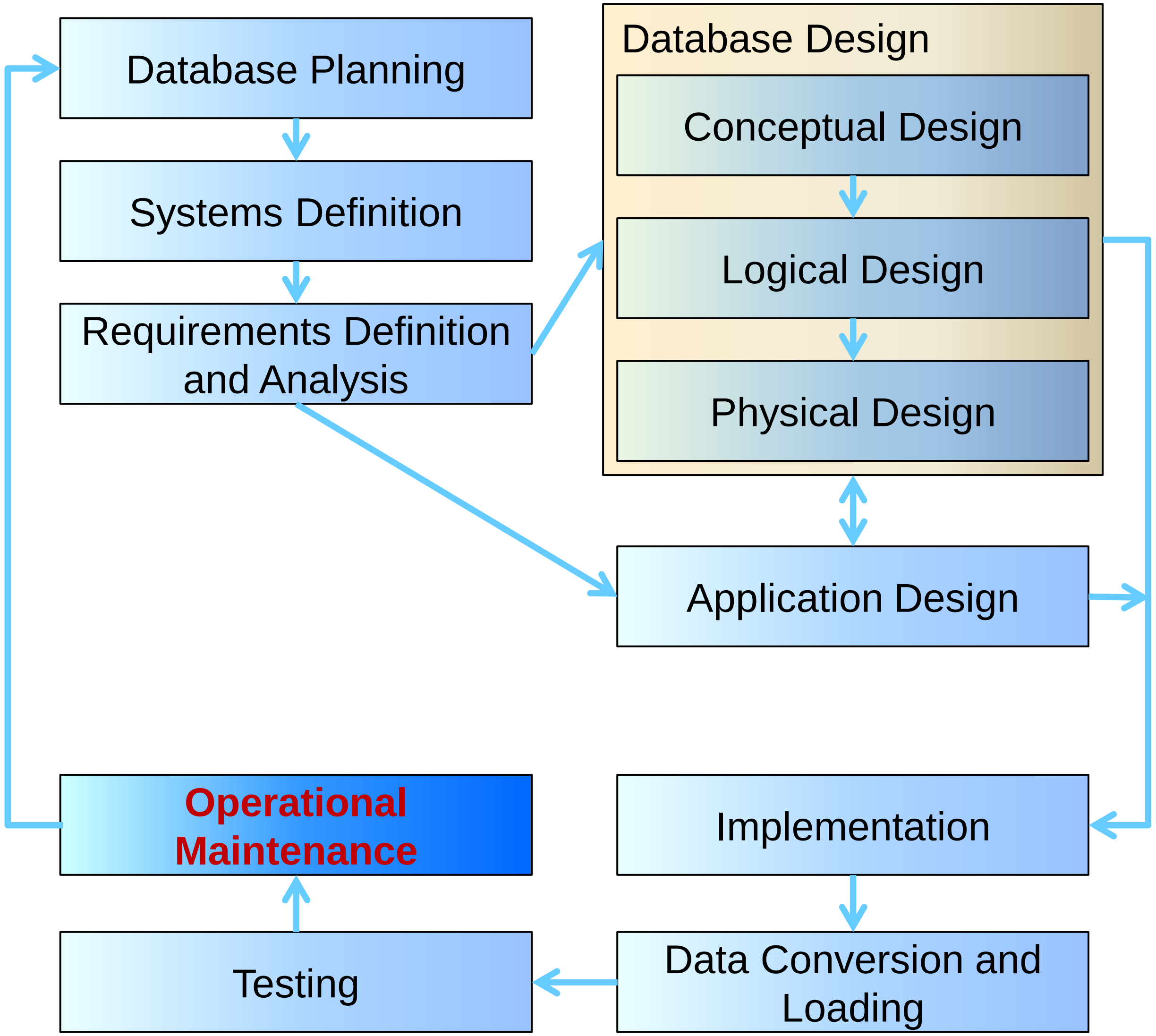
- Transfer existing data into the database
- Conversion from old systems
- Non trivial task
- Mostly outside scope of the course (concepts covered in the Data Warehouse lecture)

Database Development Lifecycle



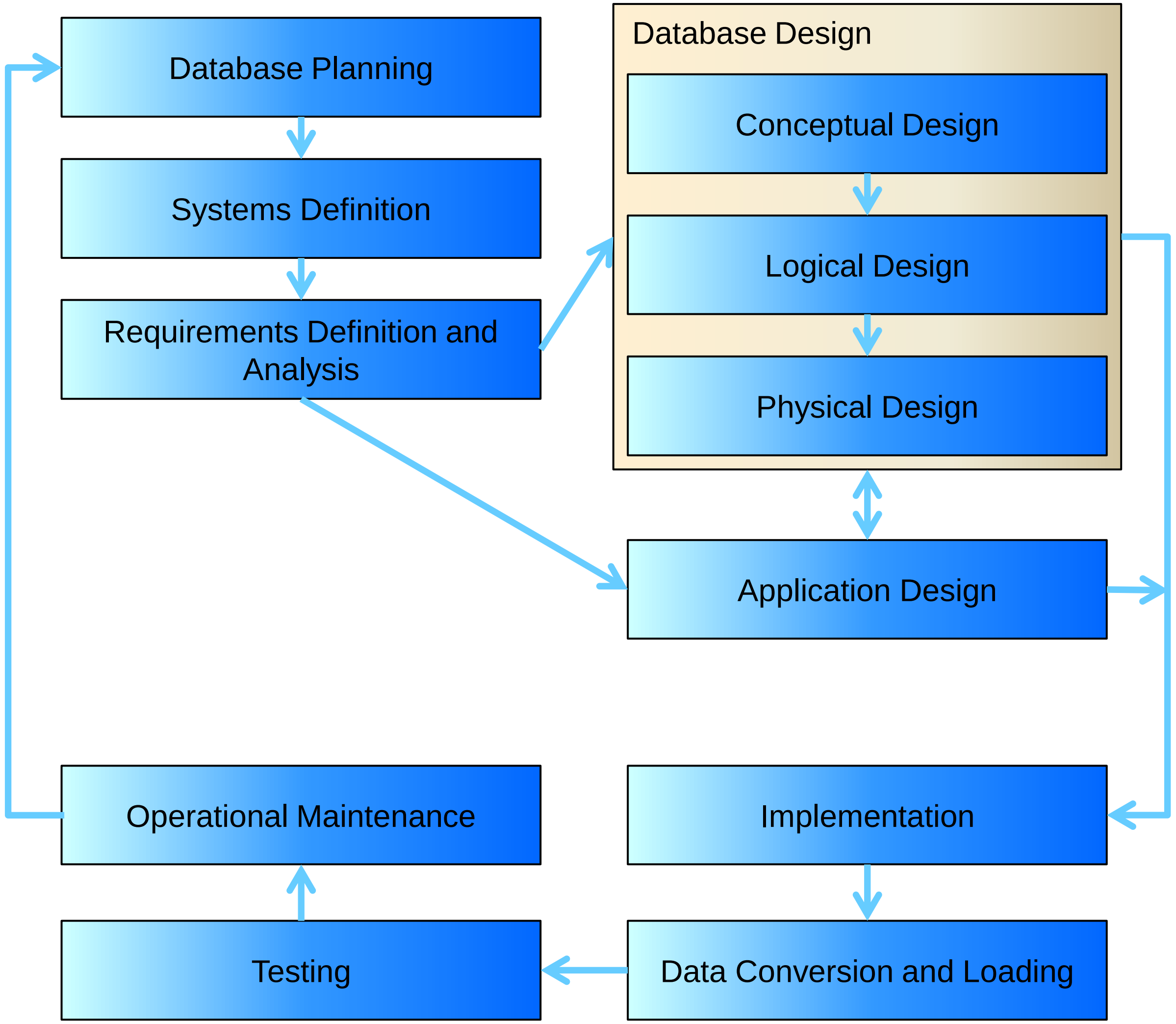
- Running the database to find errors in the design / setup
- Other issues also
  - Performance
  - Robustness
  - Recoverability
  - Adaptability
  - Security
- Mostly outside scope of the course (see ISYS90086 Data Warehousing)

Database Development Lifecycle



- The process of monitoring and maintaining the database following its commissioning
- Monitoring and improving performance
- Handling changes to requirements
- We will touch on some of these topics later in lectures 15, 20

Database Lifecycle (Summary)



Thank you