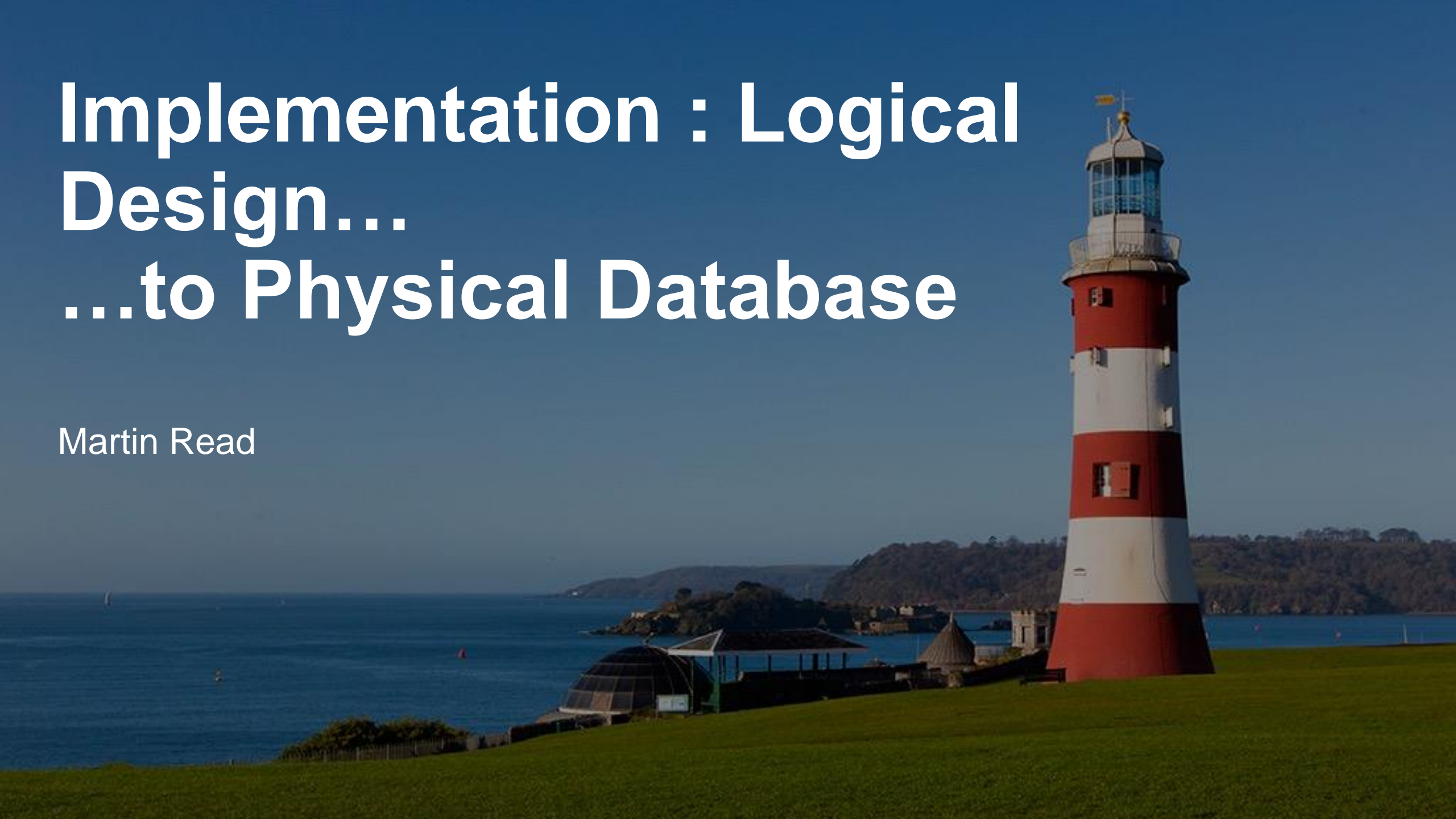


# Implementation : Logical Design.... ...to Physical Database

Martin Read



# This Lecture

Aim:

- Implementation of the database design

Learning Outcomes:

- To be able to create tables from the entities.

# Database Advantages

- Removal of redundancy
- Shareability
  - applications can share databases
  - Users can share data, perhaps remotely
- Security
  - Data should be more secure, access rights can be imposed
- Reliability
  - Protection of data against loss/damage/destruction
- Speed
  - Database tools allow speedy application development

# Database Advantages

- Data independence
  - Structure separate from programmes
- Integrity
  - Maintenance of high quality, correct and consistent data
- Cost
  - Reduced costs for systems development and maintenance
- Query language support
  - Ability to examine information from a number of angles
  - enables ad hoc queries
  - greatly improved Management access to Information

# Database Disadvantages

- Complexity
  - Requires experts to design, develop and maintain them, and some training for users
- Initial cost
  - Retraining, expensive software, specialised hardware
- Complexity
  - May require multiple databases to satisfy requirements
- Still all or nothing
  - If system crashes, none of the applications may run
- Not intelligent - Cannot interpret results

# Analysis issues

- Choosing an initial key
  - choose what is obvious
  - add an attribute(s) if necessary to establish a straightforward (Simple) key for a given relation
- Use samples as pointers to
  - processing requirements
  - “hidden” attributes
    - particularly when studying manual documents, or computerised documents with manual additions
- Documentation

# The Data Dictionary

- A data dictionary should specify as many of the properties as is considered relevant
- This should be applicable in any software or programming language
- Particular care should be used to indicate primary & foreign keys

# Documenting each Entity

- Name
- ID [optional]
- Expected number of occurrences (records) [optional]
- Description
- Synonyms
- Relationship Link Phrases
- Attributes (listed below entity)



# Documenting each Attribute

- Name
- ID [optional]
- Description
- Synonyms
- Data type
- Size/length of field
- Possible data values - ranges
- Required at creation? i.e. optionality
- Validation rules
- Key usage

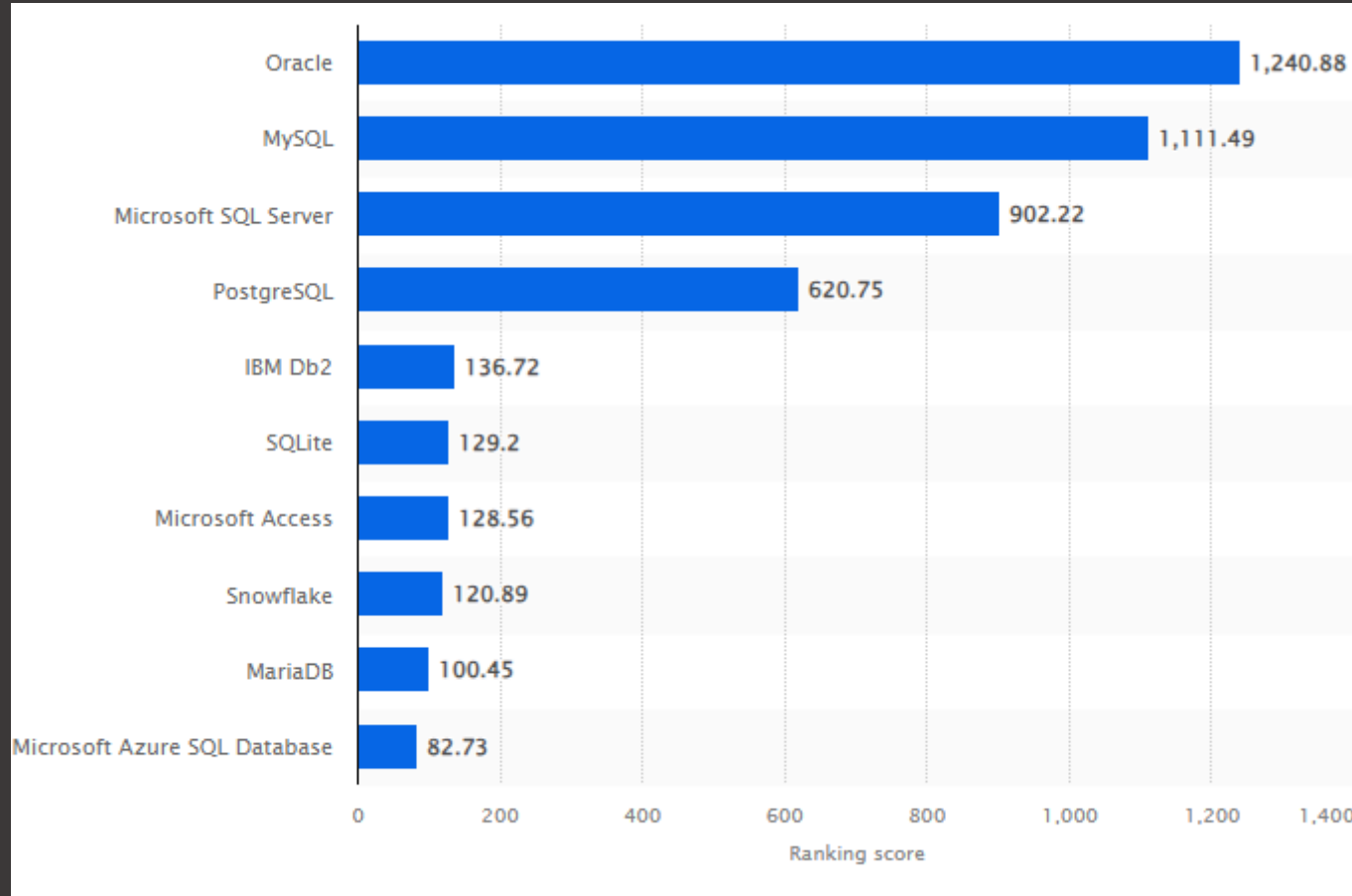
# Implementation – the issues!

- How far can you take the design independently of the eventual implementation platform?
- What are the features or constraints of the implementation platform?

The reality is that design is almost always carried out with an implementation platform in view

- e.g. DB2, Oracle, Access, SYBASE, SQL-Server...

# The most popular RDBMS in 2023



# Constraining the design

How does the implementation platform deal with:

- Entities?
  - Tables
- Attributes?
  - Fields
- Relationships?
  - Keys
- Referential Integrity?

# SQL Server

A database must be created as a first step

- Entities are implemented as TABLES
- Attributes are implemented as FIELDS
- Relationships (including keys) are implemented via constraints

# Field Definitions

including all parameters necessary for applying the field to a database

- **Key usage**
- **Data Type**
- **Field size**
- Default value?
- Required?
- Allow Zero Length?
- Indexed?
- Validation Rule/s?

# Determining the Data type

- **Questions to ask**

- any character?

- all alphabetic?

- numbers only?

- whole numbers?

- fractions/decimals?

- accounting figures?

- date/time?

- indicator/flag?

- other?

**Generic**

Alpha-numeric

Alpha

Numeric

Integer - various

Real - various

Currency/Real

Date/Time

Logical/Boolean

as appropriate

e.g. Hyperlink, Image, etc

# Other factors

- How big must the field be in order to hold the required data?  
=> Size/length of field
- Precision & Scale of numbers (number of significant digits/digits after decimal point)
- What are the possible data values?
- Must a value always be present => Optionality
- Is the attribute (part of) a Key (Primary/Foreign)
- What rules could/should be applied on data entry => Validation



# SQL Server Table definition

- Meaningful table name used – length up to 128
- Meaningful column names used
- Data types are appropriate
- Primary key is defined
- Foreign keys are defined (where applicable)
  - explicit reference to parent table & column given
- Appropriate not null constraints defined (where applicable)
- All constraints named

# How to proceed

Create a database!

- Define tables with their key structure
- Define relationships according to the LDS/ERD
- Develop tables with the rest of the fields & their properties according to the design
- Consider test requirements
- Produce test data
- 'Prove' the database by applying test data

# Summary

- Introduction to the implementation of the database design