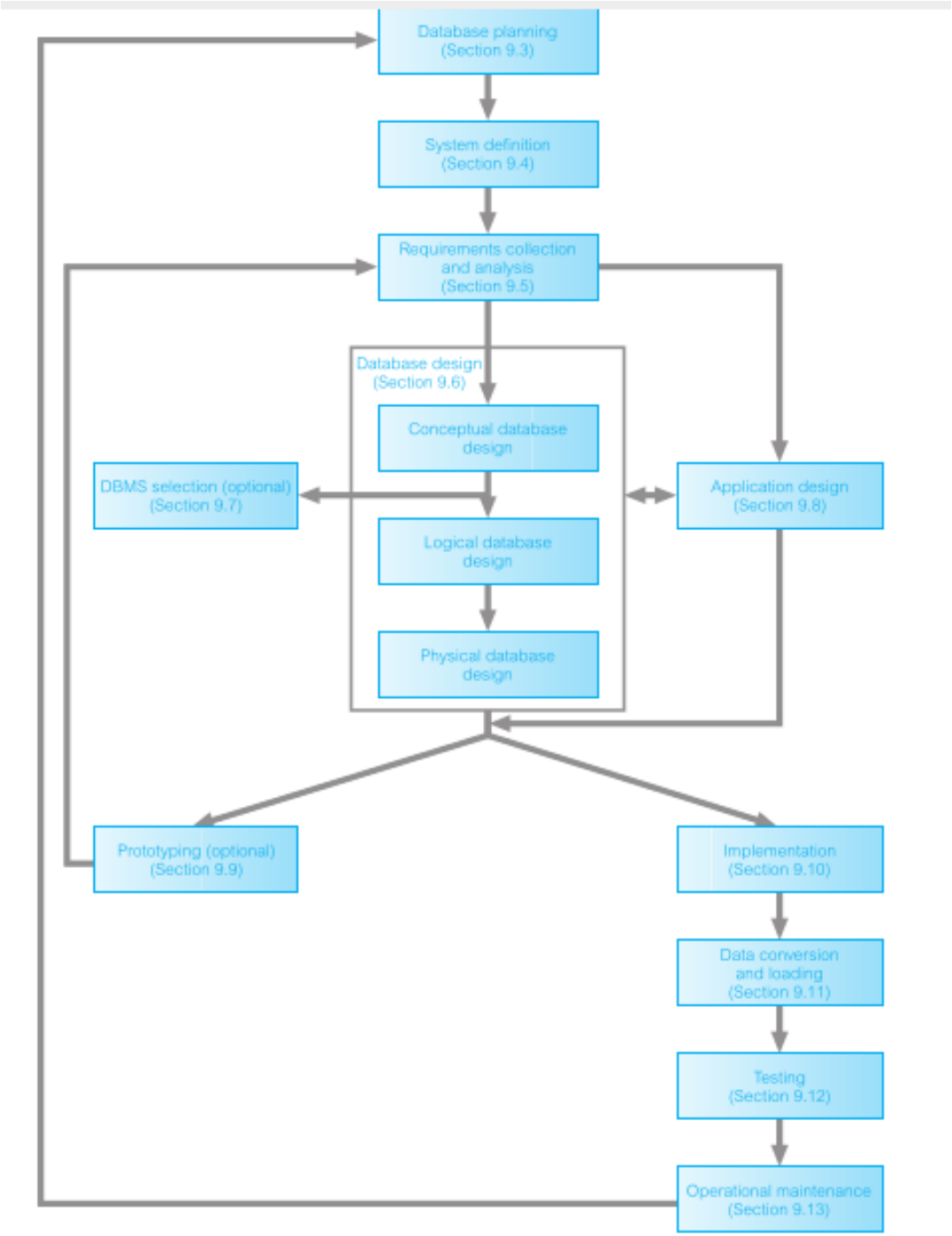
Ciclo de vida de uma base de dados

Para bases de dados pequenas com um pequeno número de utilizadores, não é mt complexo.
Para bases de dados grandes, com centenas de queries e milhares de utilizadores, pode ser extremamente complexo



Fases:

****Planeamento:**** Primeiro passo deve ser definir o mission statement (define os principais objetivos da base de dados). Depois deve-se definir os objetivos. Podem ser acompanhados de info extra como os recursos, o trabalho que deve ser feito e o dinheiro envolvido. Devem ser definidos standards que dizem como os dados vão ser recolhidos, a documentaçao necessária e como se deve proceder nos próximos passos

****Definicao do sistema**:** Descreve o contexto e limites da base de dados e as maiores vistas de utilizadores

****Recolha e análise de requisitos**:** Recolher info sobre o cliente e utiliza-la no design da base de dados. A informaçao deve ser recolhida para cada vista. Há duas abordagens: **centralizada** e de **integraçao de vistas**. Centralizada recolhe os requisitos de cada vista e faz o merge de tudo antes de passar para a fase de modelação. Esta é melhor quando as vistas sao quase iguais e a base de dados não é suficientemente complexa que justifique. A de integração mantém as vistas separadas e modela de acordo com isso. Esta é melhor quando há diferenças significativas entre cada vista e a complexidade da informação o justifica.
Sistemas MUITO complexos usam uma combinação de ambas.

****Design da base de dados**:** Processo de criar o design de uma base de dados que vai suportar o sistema pretendido.

Duas abordagens: **bottom-up e **top-down**.**
Bottom-up analisa os atributos e como estes se relacionam, dando origem a entidades e relacionamentos entre estas. É melhor para bases de dados simples com poucos atributos. Em bases de dados complexas seria dificil a análise de dependencias funcionais.
Top-down é melhor para bases de dados complexas. Desenvolvem-se algumas entidades de alto nível e vai-se refinando sucessivamente até termos uma série de entidades, relacionamentos e atributos entre elas (leccionada nas aulas). Há ainda inside-out que identifica as primeiras entidades como top-bottom e depois vai vendo os atributos e trabalhando como a bottom-up e a estratégia mista que usa top-bottom numa parte do problema e bottom-up noutra.
Assim, vamos criando modelos de dados que servem para perceber:

- a perspetiva de cada utilizador sobre os dados.
- a natureza dos dados, independente da sua representação física
- o uso dos dados ao longo das diferentes vistas

Fases do design da base de dados: conceptual, lógico e físico.

Conceptual: criaçao de um modelo de dados independente da arquitetura fisica, do DBMS e de tudo resto e baseado puramente nos requirimentos do utilizador e na compreensao que obtemos a partir da análise de requisitos.

Logico: construçao de um modelo dos dados que apesar de já se traduzir em relações, ainda é independente do DBMS e de qualquer outra consideração física. Este modelo irá ser mapeado para um DBMS específico, mas deverá ser construído de tal forma que seja independente deste. No entanto, já tem algumas noções de como vai ser: relacional, hierarquico, baseado em objetos. No entanto, parte física como índices ou estruturas de storage. Ao longo deste processo, o modelo vai sendo validade contra os requirimentos do utilizador e vai sofrendo um processo de normalização.

Fisico: Processo de produzir uma implementaçao do armazenamento dos dados, tendo em conta índices, transacoes, etc

Escolha do DBMS:

1. Definir os termos de referencia de estudo

Indica o problema, os criterios de escolha, uma lista inicial de produtos e as restriçoes necessarias.
2. Reduzir a lista a 2 ou 3 produtos
3. Avaliar os mesmos

O ideal seria fazer uma lista de features dos DBMS, atribuir um peso a cada uma e um resultado a cada DBMS. Assim, a escolha seria feita de forma relativamente automatica.
4. Recomendar a escolha e produzir o relatorio

Table 9.2 The criteria to produce an optimal data model.

Structural validity	Consistency with the way the enterprise defines and organizes information.
Simplicity	Ease of understanding by IS professionals and non-technical users.
Expressibility	Ability to distinguish between different data, relationships between data, and constraints.
Nonredundancy	Exclusion of extraneous information; in particular, the representation of any one piece of information exactly once.
Shareability	Not specific to any particular application or technology and thereby usable by many.
Extensibility	Ability to evolve to support new requirements with minimal effect on existing users.
Integrity	Consistency with the way the enterprise uses and manages information.
Diagrammatic representation	Ability to represent a model using an easily understood diagrammatic notation.