

# JOINS

- outer joins - utilizados para combinar dados de duas tabelas e manter também os registros que não têm correspondência nenhuma das tabelas  
Utéis para identificar registros incompletos ou sem correspondência; todos os registros de uma tabela mesmo que não tenham relação direta com a outra  
↳ left outer join , right outer join , full outer join
- inner join - usado para combinar todos os dados de duas ou mais tabelas mas apenas retorna os registros que têm correspondência em ambas

- self-join - operação de função que utiliza apenas uma tabela, isto é, combina os dados de uma tabela com os dados da mesma tabela → utilizada para consultas hierárquicas ou compara registros da mesma tabela
- semi-join - permite reduzir o número de registros envolvidos numa operação de função aplicando sobre o resultado da função uma projeção envolvendo todos os atributos de uma das tabelas
- anti-join: dá-nos um resultado que contém todos os registros que não satisfazem a condição de função de uma operação de função entre duas tabelas.

# Sistema de Gestão de Bases de Dados

Sistema de Gestão de Bases de Dados (SGBD) - sistema de software que executa a execução e o controlo da realização de todas as operações que permitem a realização sobre o conteúdo da BD, como a gestão do acesso, a sua persistência e manipulação, entre outros.

→ ferramenta para a descrição e controlo de dados de forma segura e integrada capaz de suportar simultaneamente diferentes utilizadores e fornecer opções de backup e recuperação de dados

- gestão eficiente, armazenamento de dados e recuperação otimizada
- garante a consistência dos dados e a simultaneidade com os controlos de transacção

- aplicação de políticas de segurança e protocolos
- Comportamentos do SGBD
  - processamento de linguagens (DML)
  - processamento de queries
  - gestão de catálogo (dicionário do sistema)
- Funcionalidades:
  - Gestão e consultas com SQL
  - Integridade dos dados
- Gestor de Base de Dados:
  - trata das queries
  - interpreta queries
  - optimiza queries
  - transaction Manager

## Ciclo de vida do desenvolvimento de uma BD

1. Definição do Sistema - compreensão e enumeração dos objetos, análise da viabilidade e planejamento inicial do projeto
2. Levantamento de Requisitos - levantamento analise e organização dos requisitos
3. Modelação Conceptual - criação do modelo ER que é validado com o cliente
4. Modelação Lógica - conversão do modelo conceptual para modelo lógico
5. Implementação Física - criação efetiva da base de dados de acordo com o modelo lógico
6. Exploração - testes e avaliação do cumprimento dos parâmetros definidos anteriormente
7. Monitorização - administração da SBD . Tarefas de rotina

## Base de dados relacional

Base de dados relacional - conjunto compatilhado de dados logicamente estruturados/relacionados entre si e uma descrição desses dados desenvolvida para satisfazer as necessidades de informação de uma organização.

As BDs relacionais utilizam tabelas para organizar os dados em linhas e colunas facilitando consultas complexas e relacionamentos entre informações.

O modelo de base representado é o modelo relacional no qual os dados são estruturados em relações e os ligações entre tabelas são estabelecidas através de chaves primárias e estrangeiras.

# ACID

- Conjunto de propriedades fundamentais para garantir confiança e integridade na transação de bases de dados
- Atomicidade - garante que todos os operações são concluídos com sucesso ou nenhuma é aplicada
- Consistência - assegura que a base de dados passa de um estado válido para outro também válido
- Isolamento - evita que transações simultâneas infiramuns uns nos outros
- Durabilidade - garante que as alterações realizadas por uma transação confirmada persistem mesmo em caso de falha do sistema
- Previne corrupção de dados, inconsistências e garante a confiabilidade do sistema em ambientes multiutilizadores

# Integridade de uma Base de Dados

- A integridade uma BD relacional é garantida através de restrições que controlam como os dados são inseridos, atualizados e mantidos:

1. Restrição de chave primária: garante que cada registo na tabela é único e não nulo. O valor chave primária não pode ter uma linha vazia.
2. Restrição chave estrangeira: garante a integridade referencial ou seja que os valores de uma tabela estão corretamente relacionados com os valores de outra.

## Chave Primária

Chave primária - atributo que identifica univocamente cada registo de uma BD relacional (não pode ser nula, nem ter valores repetidos)

Para identificar a chave primária devem analisar os dados que garantem a unicidade em cada registo

## Super-chave

Super-chave - qualquer conjunto de um ou mais atributos que identifica de forma única cada tuplo de uma tabela.

Se não houver duas linhas da tabela com os mesmos valores para todos os atributos da super-chave ela é válida. Toda a chave primária é super-chave, mas nem todo a superchave é uma chave primaria.

↳ resultam de uma má escolha de atributos identificadores  
aparecem por excesso de atributos quando a chave primaria não é bem definida

## Chave Candidata

Chave-Candidata - atributo ou mesmo conjunto que identifica de forma única uma instância dessa entidade. Podemos selecionar vários chaves candidatas, mas apenas uma delas é chave primária. As restantes permanecem como chaves alternativas para a identificação.

A identificação cometa de chaves candidatas é essencial para garantir a integridade dos dados e a aplicação dos regras de normalização. Padrões de seleção: unicidade (identifica de forma única) irreduzibilidade (se se pode remover um atributo e ainda identificar unicamente então não é uma chave candidata válida (é uma superchave)) simplicidade.

## Chave estrangeira

Chave estrangeira - atributo ou conjunto de atributos numa tabela que faz referência à chave primária de outra tabela. A principal função é garantir e estabelecer a relação entre os dados das duas tabelas.

Asegura que os valores inseridos na tabela que a contém correspondem a valores existentes na tabela referenciada garantindo assim a integridade dos dados. Está associada à integridade referencial que assegura que não existem referências inválidas entre tabelas.

## Normalização

Normalização - análise das chaves primárias e das dependências funcionais de todos os atributos

- Atributo mais primo - atributo que mais faz parte da chave primária
- Dependência funcional - quando o valor de um atributo depende univocamente de outro atributo.

$A_1 \rightarrow A_2$  significa que: - para cada valor de  $A_1$  existe apenas um valor possível de  $A_2$ ; diz-se que  $A_2$  é funcionalmente dependente de  $A_1$  ou que  $A_1$  determina funcionalmente  $A_2$ ; valores iguais para  $A_1$  determinam valores iguais para  $A_2$

# 1FN

• Diz-se que uma relação está na **1FN** se:

- Possui chave primária
- Todos os seus atributos forem atômicos (não são permitidos atributos compostos ou multivalores)
- Não possuem grupos de dados repetitivos

• Aplicações da 1FN:

1. Uma das chaves candidatas é escolhida para chave primária
2. Atributos multiválor não convertidos em novas relações com FKs a referir a PK
3. Cada atributo composto é mapeado em novos <sup>atômicos</sup> atributos atômicos

## 2 FN

Diz-se que uma relação está em 2FN se:

- também estiver em 1FN (atributos atômicos e chave primária)
  - todos os seus atributos não primários forem totalmente dependentes da sua chave primária, isto é, não podem existir dependências parciais
- Ex: Tabela com os atributos IdAluno, IdDisciplina, nomeAluno, nomeDisciplina. Sendo PK(IdAluno, IdDisciplina) nomeAluno apenas depende de IdAluno e nomeDisciplina apenas depende de IdDisciplina X não pode acontecer ( $\rightarrow$  dividir em duas tabelas Aluno e Disciplina)
- Anomalias de não estiverem 2.FN: redundância de dados, anomalias de atualização, de inserção e de eliminação.

# 3FN

Diz-se que uma relação está na **3FN** se:

- também estiver na 1FN e na 2FN

- todos os atributos que não sejam chaves primárias sejam mutuamente independentes mas havendo assim dependências funcionais transitivas. Isto é, todos os atributos dependem exclusivamente da chave primária