

# BASE DE DADOS



SQL

Teóricas- Práticas  
Ano Lectivo 2018/2019

Nuno Escudeiro/Melo e Castro/  
Rui Coentro/Silva Pereira

## Conceitos base

- **Universo ou Domínio de Discurso, UoD:** parte do mundo real que é relevante para o sistema, conjunto das entidades relevantes
- **Base de dados:** conjunto de ficheiros que armazenam/persistem os dados necessários à operação de um sistema informático organizados/estruturados/modelados de forma a tornar a sua manipulação eficiente atendendo à utilização esperada
- **SGBD – Sistema de Gestão de Bases de Dados:** aplicação que permite gerir/manipular os ficheiros da base de dados (Oracle, MySQL, MS SQL Server, PostgreSQL, ...)
- **Base de dados relacional:** base de dados que organiza os dados em tabelas/relações e associações entre elas. Particularmente adequado a sistemas transacionais caracterizados pelo elevado número de acessos pontuais, i.e., taxas de atualização elevadas manipulando um número reduzido de registos em cada transação/operação

## Conceitos base

- **SQL – Structured Query Language:** linguagem padrão para acesso e manipulação de uma base de dados relacional
- SQL é composto, no essencial, por **10 instruções base** que normalmente se agrupam em três conjuntos: DDL, DML e DCL
  - DDL – Data Definiton Language (create, alter, drop)
  - DML – Data Manipulation Language (insert, update, delete, select)
  - DCL – Data Control Language (grant, revoke, deny)
- A **DML** inclui quatro instruções que permitem manipular os dados:
  - insert - inserir registos na base de dados
  - update – atualizar atributos de registos
  - delete – remover registos
  - **select** – consultar a base de dados, selecionar dados

## Consultas de dados numa BDR

- Dado o UoD/domínio “inscrições em disciplinas” representado pelo seguinte modelo de dados relacional:

**curso**(codigo, nome)

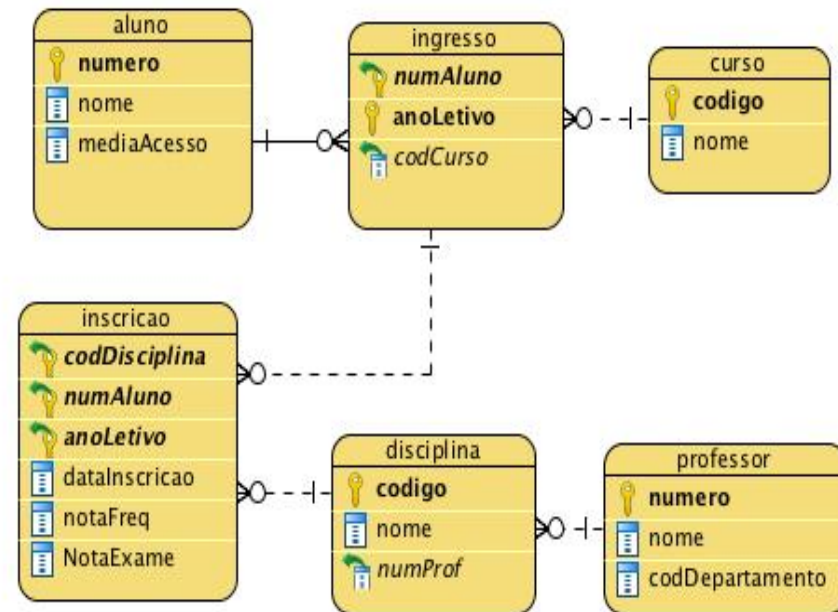
**aluno**(numero, nome, mediaAcesso)

**disciplina**(codigo, nome, numProf)

**professor**(numero, nome, codDepartamento)

**ingresso**(numAluno, anoLetivo, codCurso)

**inscricao**(codDisciplina, numAluno, anoLetivo, dataInscricao, notaFreq, notaExame)



## Consultas de dados numa BDR

1. Nome dos cursos
2. Código e nome das disciplinas
3. Nome dos alunos com media de acesso superior ou igual a 14 valores
4. Nota do aluno 1164233 na disciplina BDDAD em 2016-17; a nota de frequência vale 60% da nota final e a nota de exame o restante
5. Nota do aluno Isabel Antónia na disciplina BDDAD em 2016-17
6. Nota do aluno Isabel Antónia na disciplina Bases de Dados em 2016-17
7. Lista de alunos e respetivas classificações de frequência, exame e final, na disciplina Bases de Dados em 2016-17. As notas que ainda não tenham sido atribuídas a um determinado aluno devem ser indicadas com “---”

# Modelação Logica BDR

## NULL

- Representa algo que não se conhece

- Ex.1

Select \* from Alunos

Where Telefone = NULL -----→ Errado

- Ex.2

Select \* from Alunos

Where Telefone IS NULL -----→ Correcto

- Função IsNull / NVL Sqlserver / Oracle
  - Select IsNull(Telefone,'Não tem') From alunos
  - Select NVL(Telefone,'Não tem') From alunos

## SQL-DML (Data Manipulation Language)

```
SELECT [ ALL | DISTINCT ] <select_list>  
FROM <table_list>  
[WHERE <where_expression>]  
[GROUP BY <groupby_list>]  
[HAVING <having_expression>]  
[ORDER BY <orderby_list> [DESC|ASC]]
```

<select\_list> ::= { \* | { *table\_name* | *view\_name* | *table\_alias* }. \* | { *column\_name expression* }, [, ...n] }

<table\_list> ::= { *table\_name* | *view\_name* | *table\_alias* } [, ...n] }

<where\_expression> ::= Qualquer expressão booleana envolvendo expressões, colunas ou constantes

<groupby\_list> ::= { *table\_name* | *view\_name* | *table\_alias* }. \* | { *column\_name expression* }, [, ...n] }

<having\_expression> ::= Qualquer expressão booleana envolvendo expressões, colunas ou constantes, mas que estejam envolvidas na <groupby\_list>

<orderby\_list> ::= { *table\_name* | *view\_name* | *table\_alias* }. \* | { *column\_name* | <index> }, [, ...n] }

<index> ::= Pode tomar valores de 1 até ao nº de colunas na select\_list

## Agrupamentos

SELECT lista-atributos-a-retornar

FROM lista-tabelas-envolvidas-na-query

WHERE condições-a-verificar-em-cada-registo

**GROUP BY** lista-atributos-que-definem-agrupamento

**HAVING** condições-a-verificar-em-cada-grupo

- A lista-atributos da cláusula SELECT consiste(1) numa lista de nomes e (2) uma lista de termos de agrupadores.  
**Cada coluna que apareça em (1) tem também de aparecer na lista-atributos-do-grupo**
- As expressões que aparecem nas condições-do-grupo na cláusula HAVING servem para filtrar os valores dos grupos  
**Não é possível usar diretamente os operadores de agregação na cláusula WHERE**



## Operações de conjuntos

- ★ As operações de conjunto **Union**, **Intersect** e a **Except** (**Minus em Oracle**) operam em relações;
- ★ Eliminam os registos repetidos;
  - ★ se desejarmos obter as repetições, devemos explicitar através da forma **union all**, **intersect all** e **except all**.
- ★ Tem de existir compatibilidade entre os conjuntos:
  - ★ Mesmo número de campos
  - ★ Tipos de dados compatíveis

Logicamente, uma consulta usando INTERSECT pode ser escrita usando MINUS

## Consultas de dados numa BDR

1. Mostre o número de alunos inscritos em cada disciplina
2. Histograma das notas de frequência de todos os alunos do curso de Licenciatura em Engenharia Informática
3. Liste o nome do aluno com maior média de acesso por curso e ano letivo ordenado por ordem crescente do ano letivo e decrescente da média de acesso;
4. Liste todos os alunos ou professores;
5. Liste os alunos que são de informática e estão inscritos na base de dados;
6. Liste os alunos que são de informática e estão inscritos na base de dados, ordenados alfabeticamente;
7. Mostre todos os cursos que não têm ingressos;