

# BANCO DE DADOS I



## **Junção de Tabelas**

Prof: Ricardo Rufino

# Operação de junção



- As operações de junção retornam uma relação com o resultado da combinação de duas outras relações.
- Estas operações adicionais são utilizadas habitualmente em subconsultas na cláusula **from**
- **Condição de junção** – define quais as tuplas que são combinadas nas duas relações, assim como quais os atributos que aparecem no resultado da junção.
- **Tipo de junção** – define como tratar as tuplas que não estão relacionados entre si (basedadas na condição de junção).
- Tipos de Junção: INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN
- Condição da junção: NATURAL , ON, USING

# Operação de junção



- É obrigatória a utilização de uma condição de junção nas junções de tipo RIGHT LEFT e opcional no inner join (na ausência, comporta-se como o produto cartesiano)
- A condição **natural** aparece antes do tipo de junção (por exemplo natural join, retirando o complemento INNER). As restantes condições aparecem após o tipo de junção.
- Nas junções com condição natural, primeiro aparecem os atributos comuns a ambas as relações, na ordem pela qual aparecem na relação do lado esquerdo. Depois, aparecem os restantes atributos da relação do lado esquerdo, seguidos dos restantes atributos da relação do lado direito.
- Nas junções com condição using (A1, A2, ..., An), primeiro aparecem os atributos A1, A2, ..., An. Depois, aparecem os restantes atributos da relação do lado esquerdo, seguidos dos restantes atributos da relação do lado direito.

# Junção versus Produto Cartesiano

- Listar o nome, número de empréstimo e montante de todos os clientes que efectuaram um empréstimo na agência de Perryridge.
- **select** borrower.\*, amount **from** borrower, loan **where** borrower.loan\_number = loan.loan\_number and branch\_name = 'Perryridge'
  - Versus**select \* from** borrower **natural join** loan **where** branch\_name = 'Perryridge'
- A última separa claramente onde se coloca a origem dos dados de onde se colocam as condições “de filtragem” (selecção)
  - Esta separação não só torna a leitura mais fácil, como pode ser aproveitada para implementações.
- Na última consulta não se pode colocar “borrower.\*” após o **select** pois deixa de ser possível utilizar o nome da tabela nas colunas usadas para fazer a junção natural (isto é, as comuns a ambas as tabelas)

# Exemplos

- **select \* from loan inner join borrower on loan.loan\_number = borrower.loan\_number**
- **select \* from loan left join borrower on loan.loan\_number = borrower.loan\_number**
- **select \* from loan natural join borrower**
- **select \* from loan natural right join borrower**

# Utilização típica de USING

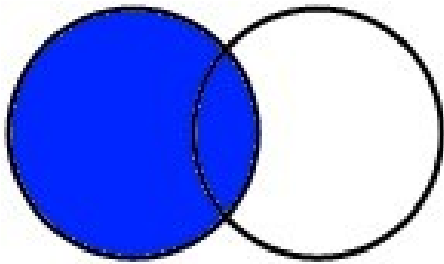
- Considerem-se as tabelas cursos(cod\_curso,nome) e alunos(num\_aluno, nome, cod\_curso)
- A consulta
- **select \* from cursos natural join** alunos
- é equivalente a
- **select** cursos.cod\_curso as cod\_curso, cursos.nome as nome, alunos.num\_aluno **from** cursos, alunos **where** cursos.cod\_curso = alunos.cod\_curso and cursos.nome = alunos.nome
- Enquanto que a consulta
- **select \* from cursos inner join** alunos **using** (cod\_curso)
- corresponde a  
**select** cursos.cod\_curso as cod\_curso, cursos.nome,alunos.num\_aluno, alunos.nome **from** cursos, alunos **where** cursos.cod\_curso = alunos.cod\_curso
- Qual é a correta?

# Aspectos a ter cuidado

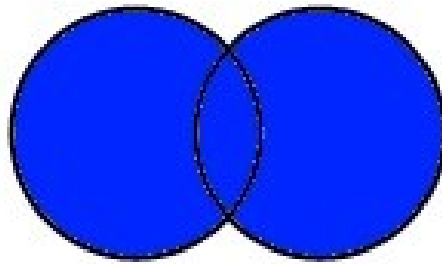
- Ocasionalmente as junções podem ser ambíguas quando estão envolvidas mais do que 2 tabelas:
- **Select \* from a natural left outer join b left outer join c on b.c1 = c.c1**
- Pode ser interpretado como:
- **Select \* from (a natural left outer join b) left outer join c on b.c1 = c.c1**
- ou:
- **Select \* from a natural left outer join (b left outer join c on b.c1 = c.c1)**
- Os operadores associam à esquerda, ou seja, a primeira hipótese é a executada. Na dúvida usar os parêntesis, em particular quando há muitas condições de junção expressas através de ON.

# SQL JOINS

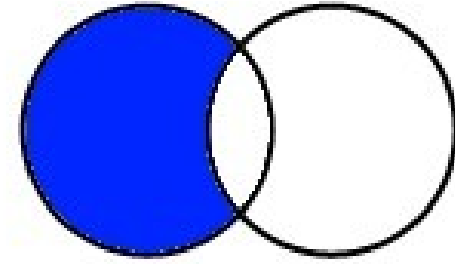
LEFT JOIN



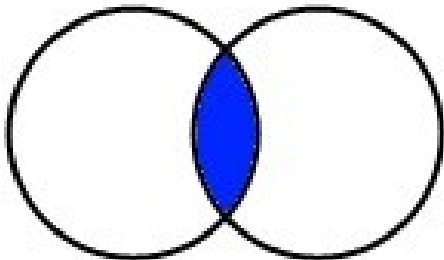
FULL OUTER JOIN



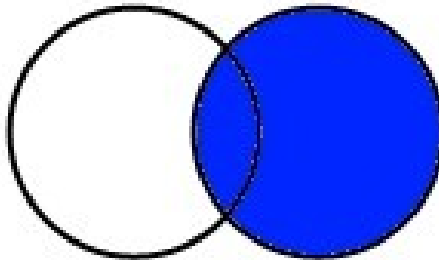
LEFT JOIN  
(if NULL)



INNER JOIN



RIGHT JOIN



RIGHT JOIN  
(if NULL)

