

Aula 9 – BD 1

SQL Parte 1

Profa. Elaine Faria

UFU - 2018

Visão Geral

- SQL
 - DML (*Data Manipulation Language*)
 - Formular consultas, inserir, excluir e modificar tuplas
 - DDL (*Data Definition Language*)
 - Criar, excluir e modificar as definições de tabelas e visões
 - Gatilhos e Restrições de Integridade
 - Gatilhos (triggers) → ações executadas pelo SGBD sempre que alterações no BD satisfazem condições específicas

Sobre os exemplos

- Todos os exemplos usarão a seguinte base

```
Marinheiros(id_marin: integer, nome_marin:  
string, avaliação: integer, idade: real)
```

```
Barcos(id_barco: integer, nome_barco: string,  
cor: string)
```

```
Reservas(id_marin: integer, id_barco: integer,  
dia: date)
```

id_marin	nome_marin	avaliação	idade
22	Daniel	7	45
29	Bruno	1	33
31	Lucas	8	55,5
32	Alceu	8	25,5
58	Raul	10	35
64	Homero	7	35
71	Ze	10	16
74	Honorato	9	35
85	Americo	3	25,5
95	Bob	3	63,5

Instância M3 de Marinheiros

id_marin	id_barco	dia
22	101	10/10/96
22	102	10/10/98
22	103	10/08/98
22	104	10/07/98
31	102	11/10/98
31	103	11/06/98
31	104	11/12/98
64	101	09/05/98
64	102	09/05/98
74	103	09/08/98

Instância R2 de Reservas

id_barco	nome_barco	cor
101	Interlagos	azul
102	Interlagos	Vermelho
103	Clipper	Verde
104	Marinha	Vermelho

Instância B1 de Barcos

Consulta SQL Básica

SELECT [DISTINCT] lista_selecao

FROM lista_from

WHERE qualificação

- Toda consulta deve ter uma cláusula SELECT e uma cláusula FROM
- A cláusula WHERE é opcional
- Uma consulta intuitivamente corresponde a uma expressão da álgebra envolvendo projeções, seleções e produto cartesiano

Exemplo

- C15: Encontre os nomes e as idades de todos os marinheiros

```
SELECT DISTINCT M.nome_marin,  
               M.idade  
FROM Marinheiros M
```

Exemplo

- C11: Encontre todos os marinheiros com uma avaliação acima de 7

```
SELECT M.id_marin, M.nome_marin,  
       M.avaliação, M.idade  
FROM Marinheiros as M  
WHERE M.avaliação > 7
```

É possível usar SELECT *

Detalhes do comandos SQL

- Lista-from
 - Lista de nomes de tabelas
 - O nome da tabela pode ser seguido por uma variável
- Lista-seleção
 - Lista de (expressões envolvendo) nomes de coluna das tabelas nomeadas na lista-from
 - Nomes das colunas podem ser prefixados por uma variável

Detalhes do comandos SQL

- Qualificação
 - Combinação booleana de condições no formato expressão *op* expressão
 - *Op* é um dos operadores de comparação
 - {<, <=, =, <>, >=, >}
 - Expressão é um nome de coluna, uma constante ou uma expressão
- *Distinct*
 - Palavra reservada opcional
 - Indica que a tabela computada como resposta não deve conter duplicatas

Detalhes do comandos SQL

- A resposta a uma consulta é uma relação
- Estratégia de avaliação de uma consulta
 - Computar o produto cartesiano das tabelas da lista *from*
 - Excluir linhas no produto cartesiano que não satisfazem as condições de *qualificação*
 - Excluir colunas que não aparecem na *lista-seleção*
 - Eliminar as linhas duplicadas (se *DISTINCT*)

Detalhes dos comandos SQL

- Estratégia direta de avaliação conceitual
 - Torna explícita as linhas que devem ser apresentadas na resposta à consulta
 - É bem provável que seja ineficiente

Um SGBD avalia as consultas de uma forma diferente a fim de torná-las mais eficientes!

Exemplo

- C1: Encontre os nomes de marinheiros que reservaram o barco 103.

```
SELECT M.nome_marin  
FROM Marinheiros M, Reservas R  
WHERE M.id_marin=R.id_marin AND  
       R.id_barco = 103
```

Exemplo

id_marin	nome_marin	avaliação	idade
22	Daniel	7	45
31	Lucas	8	55,5
58	Raul	10	35

id_marin	id_barco	dia
22	101	10/10/96
58	103	09/08/98

Compute o resultado da consulta C1 usando essas duas instâncias!

Exemplos de consultas SQL Básicas

- Consulta C1

```
SELECT nome_marin  
FROM Marinheiros M, Reservas R  
WHERE M.id_marin = R.id_marin AND  
       R.id_barco = 103
```

OU

```
SELECT nome_marin  
FROM Marinheiros, Reservas  
WHERE Marinheiros.id_marin = Reservas.id_marin AND  
       id_barco = 103
```

Exemplos de consultas SQL

Básicas

- C16: Encontre os ids dos marinheiros que reservaram um barco vermelho

```
SELECT R.id_marin  
FROM Barcos B, Reservas R  
WHERE B.id_barco = R.id_barco AND  
       B.cor = 'vermelho'
```

Exemplos de consultas SQL Básicas

- C2: Encontre os nomes dos marinheiros que reservaram um barco vermelho

```
SELECT M.nome_marin  
FROM Marinheiros M, Reserva R, Barcos B  
WHERE M.id_marin = R.id_marin AND  
R.id_barco = B.id_barco AND B.cor='vermelho'
```


Exemplos de consultas SQL

Básicas

- C3: Encontre as cores dos barcos reservados por Lucas

```
SELECT B.cor  
FROM Marinheiros M, Reservas R, Barcos B  
WHERE M.id_marin = R.id_marin AND  
       R.id_barco = B.id_barco AND  
       M.nome_marin = 'Lucas'
```

Exemplos de consultas SQL Básicas

- C4: Encontre os nomes dos marinheiros que reservaram pelo menos um barco

```
SELECT DISTINCT M.nome_marin  
FROM Marinheiros M, Reservas R  
WHERE M.id_marin = R.id_marin
```

Expressões e Strings no Comando SELECT

- Cada item em uma lista-seleção pode ser

expressão AS nome-coluna

- expressão: qualquer expressão aritmética ou de string envolvendo os nomes de colunas e constantes
- nome-coluna: novo nome para essa coluna na saída da consulta

Exemplos

- C17: Compute incrementos das avaliações de pessoas que manobraram dois barcos diferentes no mesmo dia

```
SELECT M.nome_marin, M.avaliacao+1 AS avaliação
FROM Marinheiros M, Reservas R1, Reservas R2
WHERE M.id_marin = R1.id_marin AND
      M.id_marin = R2.id_marin AND
      R1.dia = R2.dia AND
      R1.id_barco <> R2.id_barco
```

Exemplos

```
SELECT M1.nome_marin AS nome1,  
       M2.nome_marin AS nome2  
FROM Marinheiros M1, Marinheiros M2  
WHERE 2*M1.avaliação = M2.avaliação
```

String

- Para comparações de strings pode-se usar

=, <, >, etc.

- Ordem das strings determinada alfabeticamente

- Like

- Símbolo Curinga % → zero ou mais caracteres arbitrários

- Símbolo _ → exatamente um caractere arbitrário

Exemplo

- C18: Encontre as idades dos marinheiros cujos nomes começam e terminam com B e têm no mínimo três caracteres

```
SELECT M.idade  
FROM Marinheiros M  
WHERE M.nome_marin LIKE 'B_%B'
```

Union, Intersect e Except

- SQL fornece 3 construtores de manipulação de conjuntos
 - União → UNION
 - Interseção → INTERSECT
 - Diferença → EXCEPT

Exemplo

- C5: Encontre os nomes dos marinheiros que reservaram um barco vermelho ou um barco verde

```
SELECT M.nome_marin  
FROM Marinheiros M, Reservas R, Barcos B  
WHERE M.id_marin=R.id_marin AND  
      R.id_barco = B.id_barco AND  
      (B.cor='vermelho' OR B.cor='verde')
```

Exemplo

- C6: Encontre os nomes dos marinheiros que reservaram um barco vermelho e um barco verde

```
SELECT M.nome_marin
FROM Marinheiros M, Reservas R1, Barcos B1,
     Reservas R2, Barcos B2
WHERE M.id_marin = R1.id_marin AND
      R1.id_barco = B1.id_barco AND
      M.id_marin = R2.id_marin AND
      R2.id_barco = B2.id_barco
      B1.cor='vermelho' AND B2.cor = 'verde'
```

Exemplo

- Consultas C5 e C6 podem ser melhor escritas usando UNION e INTERSECT

Exemplo

- C5

```
SELECT M.nome_marin
FROM Marinheiros M, Reservas R, Barcos B
WHERE M.id_marin=R.id_marin AND
       R.id_barco = B.id_barco AND B.cor = 'vermelho'
UNION
SELECT M2.nome_marin
FROM Marinheiros M2, Barcos B2, Reservas R2
WHERE M2.id_marin=R2.id_marin AND
       R2.id_barcos=B2.id_barco AND B2.cor='verde'
```

Exemplo

- C6

```
SELECT M.nome_marin
FROM Marinheiros M, Reservas R, Barcos B
WHERE M.id_marin=R.id_marin AND
      R.id_barco = B.id_barco AND B.cor = 'vermelho'
INTERSECT
SELECT M2.nome_marin
FROM Marinheiros M2, Barcos B2, Reservas R2
WHERE M2.id_marin=R2.id_marin AND
      R2.id_barcos=B2.id_barco AND B2.cor='verde'
```

Exemplo

A solução proposta para a consulta C6 usando INTERSECT contém um erro

Qual é esse erro?

Exemplo

- C19: Encontre os id_marins de todos os marinheiros que reservaram barcos vermelhos, mas não barcos verdes

```
SELECT R.id_marin
FROM Reservas R, Barcos B
WHERE R.id_barco= B.id_barco AND
      B.cor ='vermelho'
EXCEPT
SELECT R2.id_marin
FROM Barcos B2, Reservas R2
WHERE R2.id_barco = B2.id_barco AND
      B2.cor='verde'
```

Exemplo

- Observação
 - A consulta C19 baseia-se na integridade referencial, ou seja, não há reservas para marinheiros não existentes
- UNION, INTERSECT e EXCEPT
 - podem ser usados em duas tabelas quaisquer que sejam compatíveis à união

Exemplo

- C20: Encontre todos os ids dos marinheiros que têm uma avaliação 10 ou reservaram o barco 104

```
SELECT M.id_marin
FROM Marinheiros M
WHERE M.avaliação = 10
UNION
SELECT R.id_marin
FROM Reservas R
WHERE R.id_barco = 104
```

- Observações

- UNION, INTERSECT E EXCEPT eliminam duplicatas

- Para reter as duplicatas use

- UNION ALL

- O número de cópias de uma linha no resultado é $m+n$, onde m e n são os números de vezes que a linha aparece nas duas partes da união

- INTERSECT ALL

- O número de cópias de uma linha no resultado é $\min(m,n)$

- EXCEPT ALL

- O número de cópias de uma linha no resultado é $m - n$ onde m corresponde à primeira relação

Referências

- R. Ramakrishnan e J. Gehrke, *Database Management Systems*, 3a Edição, McGraw-Hill, 2003.