

3075 - Fundamentos de banco de dados

2204 - Banco de dados

Curso de Sistemas de Informação

Prof. Daniella Vieira

Unidade II

CONCEITOS: MODELAGEM DE DADOS

- Modelos de banco de dados.
- O modelo relacional: domínios, tuplas, atributos, relações, entre outros.
- Diagrama de entidade-relacionamento.
- Mapeamento do modelo conceitual para o modelo relacional.

Referência Bibliográfica

NAVATHE, Elmasri. **Sistema de Banco de Dados**. 6ª Edição. São Paulo: Person Addison Wesley, 2011.



Álgebra relacional

Maneira teórica de se manipular o banco de dados relacional.

Linguagem de consulta procedural

- usuários especificam os dados necessários e como obtê-los

Consiste de um conjunto de operações

- entrada: uma ou duas relações
- saída: uma nova relação resultado

Álgebra relacional

As operações da álgebra relacional são as seguintes:

Seleção (σ) - seleciona um subconjunto de linhas de uma relação

Projeção (π) – apaga colunas desnecessárias de uma relação

Produto cartesiano (\times) – permite combinar duas relações

União ou Union (\cup) - tuplas na relação 1 e na relação 2

Diferença ou Minus ($-$) – tuplas na relação 1 mas não na relação 2

Álgebra relacional

Importante

O resultado de cada operação da álgebra relacional é uma nova relação, uma tabela, que pode ser manipulada por outras operações da álgebra relacional. Assim, as operações da álgebra relacional são realizadas sobre relações inteiras (não em uma tupla (linha) da relação; o resultado dessas operações é sempre uma nova relação, uma tabela.

Álgebra relacional: seleção

$\sigma_{\text{condição_seleção}}$ (relação argumento)

- pode envolver operadores de comparação
(=, >, ≥, <, ≤, ≠)
- pode combinar condições usando-se \wedge , \vee , \neg

- relação
- resultado de alguma operação da álgebra relacional

SQL - DML

```
SELECT <lista de atributos e funções>  
FROM <lista de tabelas>  
[ WHERE predicado ]  
[ GROUP BY <atributos de agrupamento> ]  
[ HAVING <condição para agrupamento> ]  
[ ORDER BY <lista de atributos> ] ;
```

Álgebra relacional: projeção

- Projeta as **colunas** solicitadas (i.e. produz um subconjunto vertical)

$\pi_{\text{lista_atributos}}$ (relação argumento)

- lista de atributos
- os atributos são separados por vírgula

- relação
- resultado de alguma operação da álgebra relacional

Álgebra relacional: produto cartesiano

relação argumento 1 \times relação argumento 2

- relação
- resultado de alguma operação da álgebra relacional

Álgebra relacional: renomear

Renomeia: nome da relação, nomes dos atributos da relação, nome da relação e nomes dos atributos.

$\rho_{S(B_1, B_2, \dots, B_n)} \text{ (relação)}$

nome *novo* da relação

nome *antigo* da relação

nomes *novos* dos atributos

SQL: SELECT-FROM-WHERE

SELECT <atributos>

FROM <lista de tabelas>

[WHERE condições de seleção]

SQL

Álgebra Relacional

SELECT

->

projeção

FROM

->

produto cartesiano

WHERE

->

seleção

SQL: ORDER BY

Ordena as tuplas que aparecem no resultado de uma consulta

- asc (padrão): ordem ascendente
- desc: ordem decendente

Ordenação pode ser especificada em vários atributos

- a ordenação referente ao primeiro atributo é prioritária. Se houver valores repetidos, então é utilizada a ordenação referente ao segundo atributo, e assim por diante

SQL: AS

Renomeia

– atributos

- deve aparecer na cláusula SELECT
- útil para a visualização das respostas na tela

– relações

- deve aparecer na cláusula FROM
- útil quando a mesma relação é utilizada mais do que uma vez na mesma consulta

Sintaxe: nome_antigo AS nome_novo

Álgebra e SQL: Junção

[INNER] JOIN

$R \bowtie S$

Somente as tuplas de R que têm tuplas correspondentes em S – e vice-versa – aparecem no resultado.

R			S		$R \bowtie S$				
A	B	C	A	D	R.A	S.A	B	C	D
1	a	x	1	d	1	1	a	x	d
2	b	y	2	d	2	2	b	y	d
3	a	y							
4	c	y	5	e					

Álgebra e SQL: Junção

LEFT [OUTER] JOIN

$R \bowtie S$

Mantém cada tupla de R na tabela de junção.

Preenche com valores nulos as tuplas de S que não correspondem à coluna de junção em R.

R			S		$R \bowtie S$				
A	B	C	A	D	R.A	S.A	B	C	D
1	a	x	1	d	1	1	a	x	d
2	b	y	2	d	2	2	b	y	d
3	a	y	5	e	3	Null	a	y	Null
4	c	y			4	Null	c	y	Null

Álgebra e SQL: Junção

RIGHT [OUTER] JOIN

$R \bowtie S$

Mantém cada tupla de S na tabela de junção.

Preenche com valores nulos as tuplas de R que não correspondem à coluna de junção em S.

R			S		$R \bowtie S$				
A	B	C	A	D	R.A	S.A	B	C	D
1	a	x	1	d	1	1	a	x	d
2	b	y	2	d	2	2	b	y	d
3	a	y	5	e	Null	5	Null	Null	e
4	c	y							

Álgebra e SQL: Junção

FULL [OUTER]JOIN

$R \bowtie S$

Mantém cada tupla de R e de S na tabela de junção.

Preenche com valores nulos as tuplas que não correspondem à coluna de junção.

R			S		$R \bowtie S$				
A	B	C	A	D	R.A	S.A	B	C	D
1	a	x	1	d	1	1	a	x	d
2	b	y	2	d	2	2	b	y	d
3	a	y			3	Null	a	y	Null
4	c	y			4	Null	c	y	Null
			5	e	Null	5	Null	Null	e

Operações sobre Conjuntos

Unem duas relações

- Operações: união, intersecção, e diferença
- Características
 - atuam sobre relações compatíveis. Duas relações são compatíveis se: possuem o mesmo grau; e, seus atributos possuem os mesmos domínios (os domínios dos i -ésimos atributos de cada relação são os mesmos).
 - eliminam registros duplicados da relação resultado.

Operações sobre Conjuntos

Dado o conjunto de tabelas:

cliente (nro_cli, nome_cli, end_cli, saldo, cod_vend)

vendedor (cod_vend, nome_vend)

Liste os nomes dos clientes que possuem nomes iguais aos nomes de vendedores.

```
SELECT nome_cli FROM cliente
```

```
INTERSECT
```

```
SELECT nome_vend FROM vendedor
```

SQL: Subconsultas Aninhadas

Subconsulta

Expressão `SELECT ... FROM ... WHERE ...` aninhada dentro de outra consulta

- Aplicações mais comuns
 - testes para membros de conjuntos
 - comparações de conjuntos
 - cardinalidade de conjuntos

Membros de um Conjunto

IN

- testa se um atributo ou uma lista de atributos é membro do conjunto.

NOT IN

- verifica a ausência de um membro em um conjunto.

Conjunto:

- coleção de valores produzidos por uma cláusula SELECT ...
FROM ... WHERE ...

Membros de um Conjunto

Dado o conjunto de tabelas:

cliente (nro_cli, nome_cli, end_cli, saldo, cod_vend)

vendedor (cod_vend, nome_vend)

Liste os números dos clientes que tem nome igual ao nome de um vendedor.

```
SELECT nro_cli FROM cliente
WHERE nome_cli IN
    (SELECT nome_vend FROM vendedor)
```

Cardinalidade de Conjuntos

EXISTS: a condição é verdadeira quando a lista (resultado de uma consulta) não for vazia

NOT EXISTS: a condição é verdadeira quando a lista for vazia

Liste os números dos clientes que têm nome igual ao nome de um vendedor.

```
SELECT nro_cli FROM cliente
      WHERE EXISTS
(SELECT * FROM vendedor
 WHERE cliente.nome_cli=vendedor.nome_vend)
```

Álgebra Relacional: Agregação

atributos_agrupamento ξ funções_agregação
(relação argumento)

- lista de atributos de agrupamento
- os atributos são separados por vírgula

- relação
- resultado de alguma operação da álgebra relacional

SQL: GROUP BY-HAVING

```
SELECT <lista de atributos e funções>  
FROM <lista de tabelas>  
[ WHERE predicado ]  
[ GROUP BY <atributos de agrupamento> ]  
[ HAVING <condição para agrupamento> ]  
[ ORDER BY <lista de atributos> ] ;
```

Funções de Agregação

Funções

- Média: AVG()
- Mínimo: MIN()
- Máximo: MAX()
- Total: SUM()
- Contagem: COUNT()

Observação

- DISTINCT: não considera valores duplicados
- ALL: inclui valores duplicados

Funções de Agregação

Dado a tabela:

vinho (vinho_id, nome_vinho, tipo_vinho, preco, vinicola_id)

Dados de simulação

vinho_id	nome_vinho	tipo_vinho	preco	vinicola_id
10	Amanda	tinto	100,00	1
09	Belinha	branco	200,00	1
05	Camila	rosê	300,00	1
15	Daniela	branco	250,00	2
27	Eduarda	branco	150,00	2
48	Fernanda	tinto	7,00	2
13	Gabriela	tinto	397,00	3
12	Helena	branco	333,00	3

Funções de Agregação

Dado a tabela:

vinho (vinho_id, nome_vinho, tipo_vinho, preco, vinicola_id)

Qual a média dos preços?

ξ $AVG(preco)$ (vinho)

```
SELECT AVG (preco) FROM vinho
```

Qual o vinho mais barato e qual o vinho mais caro?

ξ $MIN(preco), MAX(preco)$ (vinho)

```
SELECT MIN (preco), MAX (preco) FROM vinho
```

Funções de Agregação

Dado a tabela:

vinho (vinho_id, nome_vinho, tipo_vinho, preco, vinicola_id)

Quantos vinhos existem na relação vinho?

ξ COUNT (vinho_id) (vinho)

SELECT COUNT (vinho_id) FROM vinho

SQL: Cláusula GROUP BY

Permite aplicar uma função de agregação não somente a um conjunto de registros, mas também a um grupo de conjunto de registros.

Grupo de conjunto de registros: conjunto de tuplas que possuem o mesmo valor para os atributos de agrupamento

SQL: Cláusula GROUP BY

Qual o preço mais alto e a média dos preços por tipo de vinho?

ρ (tipo_vinho, "maior preco", "preço medio")
(tipo_vinho ξ MAX (preco), AVG (preco) (vinho))

```
SELECT  tipo_vinho,  
        MAX (preco) AS "maior preco",  
        AVG (preco) AS "preço medio"  
FROM  vinho  
GROUP BY tipo_vinho
```

As tuplas da tabela vinho são divididas em grupo, cada grupo contendo o mesmo tipo de valor para o atributo de agrupamento tipo_vinho

SQL: Cláusula HAVING

Permite especificar uma condição de seleção para grupos, melhor do que para registros individuais .

Recupera os valores para as funções somente para aqueles grupos que satisfazem à condição imposta na cláusula HAVING

SQL: Cláusula GROUP BY

Qual o preço mais alto e a média dos preços por tipo de vinho, para médias de preços superiores a R\$200,00 ?

```
SELECT tipo_vinho,  
       MAX (preco), AVG (preco)  
FROM vinho  
GROUP BY tipo_vinho  
HAVING AVG (preco) > 200
```

aplica-se o predicado que aparece na cláusula WHERE. Coloca-se as tuplas que satisfazem a cláusula WHERE em grupos por meio da cláusula GROUP BY. Aplica-se a cláusula HAVING a cada grupo. Remove-se os grupos que não satisfazem o predicado da cláusula HAVING. Exibe-se as colunas listadas na cláusula SELECT.

Atividade

