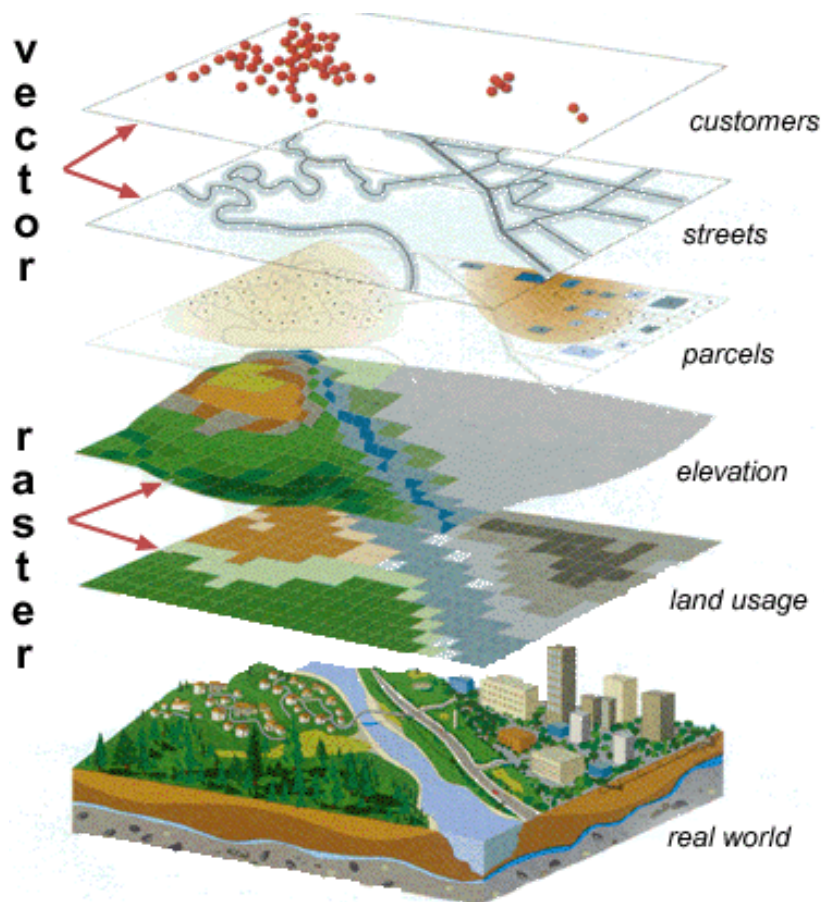




SIG: Sistemas de Informações Geográficas



Adaptado de
Gilberto
Câmara



Banco de Dados Geográficos

Comandos SQL

Adaptado de

Karine Reis Ferreira – karine@dpi.inpe.br

Gilberto Câmara – gilberto@dpi.inpe.br

Gilberto Ribeiro de Queiroz – gribeiro@dpi.inpe.br

Ronaldo Celso Messias Correia - ronaldo@fct.unesp.br



Faixa de Valores e Comparação de Strings

- Exemplo1 - Selecionando os nomes dos clientes com código entre 2 e 5:

```
SELECT nome FROM cliente
      WHERE cod_cli BETWEEN 2 AND 5;
SELECT nome FROM cliente
      WHERE cod_cli >= 2 AND cod_cli <= 5;
```

- Exemplo2 - Selecionando os nomes de clientes que começam com a letra G:

```
SELECT nome
      FROM cliente
      WHERE nome LIKE 'G%';
```



Faixa de Valores e Comparação de Strings

- Exemplo3 – Selecionando os nomes de clientes que contenham a letra E:

```
SELECT nome  
      FROM cliente  
     WHERE nome LIKE '%E%';
```

- Exemplo4 – Selecionando os nomes de clientes que não comecem com a letra G:

```
SELECT nome  
      FROM cliente  
     WHERE nome NOT LIKE 'G%';
```



Consultas com Junção

- É uma consulta onde os dados são recuperados de mais de uma tabela.
- A cláusula FROM define o produto cartesiano entre as relações.

- Ex:

```
sig=#SELECT cliente.*, passagem.*  
      FROM cliente, passagem;
```

Ambíguo

É necessário que os atributos selecionados sejam distinguíveis.

```
sig=#SELECT cod_cli  
      FROM cliente, passagem;
```



Consultas com Junção

- É uma consulta onde os dados são recuperados de mais de uma tabela.
- A cláusula FROM define o produto cartesiano entre as relações.

- Ex:

```
sig=#SELECT cliente.cod_cli  
      FROM cliente, passagem;
```

```
sig=#SELECT c.nome, p.num  
      FROM cliente AS c, passagem AS p;
```



Consultas com Junção

- Dados: Relação: voo

Atributo	Tipo	Tamanho	Restrição	descrição
Num_voo	Varchar	30	PK	Identificador de cada vôo previsto
origem	varchar	30	-	Nome da cidade de origem
Aeroporto_o	varchar	40	-	Nome do aeroporto de origem
destino	varchar	30	-	Nome da cidade de destino
Aeroporto_d	varchar	40	-	Nome do aeroporto de destino
Horário partida	timestamp	-	-	Horário previsto do vôo
Horário	timestamp			Horário previsto de



Consultas com Junção

- Dados: Relação: cred_card

atributo	tipo	tamanho	restrição	Descrição
Id_card	Integer	-	PK	Identificador do cartão de crédito
nome	varchar	20		Nome da bandeira

Timestamp: '2017-10-23 14:19:00'

```
ALTER TABLE passagem ADD COLUMN num_voo varchar(30)DEFAULT '12345';
```




Consultas com Junção

- Exemplo1 - Obter o nome do cliente e seu respectivo número da passagem e poltrona:

```
SELECT nome, passagem.num, poltrona  
FROM cliente, passagem  
WHERE cliente.cod_cli = passagem.cod_cli  
ORDER BY nome, num;
```

```
ALTER TABLE voo ADD CONSTRAINT numvoo_pk PRIMARY KEY  
(num_voo);
```

```
ALTER TABLE passagem ADD CONSTRAINT numvoo_fk FOREIGN KEY  
(num_voo) REFERENCES voo(num_voo);
```



Consultas com Junção

- Exemplo2 - Obter o nome do cliente e seu respectivo número da passagem, poltrona e número do vôo:
- Tudo

```
SELECT c.nome, p.num, p.poltrona, v.num_voo  
FROM cliente c, passagem p, voo v;
```

```
SELECT c.nome, p.num, p.poltrona, v.num_voo  
FROM cliente c, passagem p, voo v  
WHERE c.cod_cli = p.cod_cli  
AND p.num_voo = v.num_voo  
ORDER BY c.nome, v.num_voo;
```



Consultas com Junção

- Exemplo2 - Obter o nome do cliente e seu respectivo número da passagem, poltrona e número do vôo:
- Usando Junção

```
SELECT c.nome, p.num, p.poltrona, v.num_voo
FROM cliente c
JOIN passagem p ON (c.cod_cli = p.cod_cli)
JOIN voo v ON (p.num_voo = v.num_voo)
ORDER BY c.nome, v.num_voo;
```



Consultas com Junção

- Exemplo3 - Obter o número da passagem, poltrona, rota e número do vôo do cliente 'JONAS':

```
SELECT c.nome, p.num, p.poltrona, v.num_voo
FROM cliente c, passagem p, voo v
WHERE c.cod_cli = p.cod_cli
      AND p.num_voo = v.num_voo
      AND c.nome = 'JONAS'
ORDER BY c.nome, v.num_voo;
```



Consultas com Junção

- Exemplo4 - Obter o número da passagem, poltrona, rota e número do vôo do cliente 'JONAS' cujo valor seja maior que R\$ 150,00 :

```
SELECT c.nome, p.num, p.poltrona, v.num_voo
FROM cliente c, passagem p, voo v
WHERE c.cod_cli = p.cod_cli
      AND p.num_voo = v.num_voo
      AND c.nome = 'JONAS'
      AND valor > '150.00'
ORDER BY c.nome, v.num_voo
```



Removendo Linhas Duplicadas

- Exemplo1 - Obter o nome de todos os clientes que compraram passagens:

```
sig=#SELECT c.nome
      FROM cliente c, passagem p
      WHERE c.cod_cli = p.cod_cli
      ORDER BY c.nome;
```

- Exemplo2 - Usando DISTINCT:

```
sig=#SELECT DISTINCT c.nome
      FROM cliente c, passagem p
      WHERE c.cod_cli = p.cod_cli
      ORDER BY c.nome;
```



Removendo Linhas Duplicadas

- Exemplo3 – Obter o nome e número de passagens de todos os clientes que compraram passagens:
- Exemplo4 – Usando DISTINCT:



Removendo Linhas Duplicadas

- Exemplo3 - Obter o nome e número de passagens de todos os clientes que compraram passagens:

```
SELECT c.nome, p.num
      FROM cliente c, passagem p
     WHERE c.cod_cli = p.cod_cli
     ORDER BY c.nome;
```

- Exemplo4 - Usando DISTINCT:

```
SELECT DISTINCT c.nome, p.num
      FROM cliente c, passagem p
     WHERE c.cod_cli = p.cod_cli
     ORDER BY c.nome;
```




Funções Agregadas

- Exemplo1 - Qual o número total de passagens vendidas:

```
sig=# SELECT *, COUNT (*)  
      FROM passagem;
```

- Exemplo2 - Qual o valor total das passagens vendidas:

```
sig=# SELECT SUM(valor)  
      FROM passagem;
```

- Exemplo3 - Qual é a passagem mais cara:

```
sig=# SELECT MAX(p.valor)  
      FROM passagem p;
```



Funções Agregadas

- Exemplo4 - Qual é a passagem mais barata:

```
sig=# SELECT MIN (p.valor)  
      FROM passagem p;
```

- Exemplo5 - Qual é o valor médio das passagens:

```
sig=# SELECT AVG (p.valor)  
      FROM passagem p;
```



Funções Agregadas

- Exemplo6 - Quantas passagens foram vendidas para o vôo RJ-SP:

```
sig=#SELECT COUNT (*)  
        FROM passagem p, voo v  
        WHERE p.num_voo = v.num_voo  
        AND v.rota = 'RJ-SP';
```

- Exemplo7 - Quantas Estados estão representados pelos clientes:

```
sig=#SELECT COUNT (DISTINCT uf)  
        FROM cliente;
```

- Para uma lista mais detalhada de funções agregadas consulte \da no psql.



Group By

- Exemplo1 - Número de pessoas por UF:

```
sig=#SELECT uf, COUNT(*) AS TOTAL_PESSOAS
      FROM cliente
      GROUP BY uf;
```

- Exemplo2 - Quanto foi arrecadado em cada vôo:

```
sig=#SELECT num_voo, SUM(valor)
      FROM passagem
      GROUP BY num_voo
      ORDER BY num_voo;
```



Group By

- Exemplo3 - Possíveis fontes de erro:

```
sig=#SELECT num_voo, cod_cli, SUM(valor)
      FROM passagem
      GROUP BY num_voo
      ORDER BY num_voo;
```

- Obs.: Podemos agrupar por mais de uma coluna.

INSERT com Query

- Adicionar dados:
 - Comando: **INSERT INTO**
`INSERT INTO tabela (A1, A2, A3...) QUERY;`
 - Exemplo:
`sig=#INSERT INTO cliente (codigo, nome,
telefone, sexo) SELECT * FROM cliente3;`
INSERT



Números Sequenciais (Sequences)

- Exemplo3 – Utilizando um campo serial:

```
CREATE TABLE passagem (  
    num      SERIAL,  
    ...)
```

- Exemplo4 – Destruindo uma sequence:

```
DROP SEQUENCE contador_cli;
```



Visões (Views)

- São pseudo-tabelas.
- Podem representar um subconjunto de uma tabela:
 - Selecionando certas colunas
 - Selecionando certas linhas
- Empregadas para restringir o acesso a colunas da tabela.
- Ex: Visão sobre a junção cliente e passagem

```
CREATE VIEW cliente_passagem AS
SELECT nome, passagem.num, poltrona
      FROM cliente, passagem
      WHERE cliente.cod_cli =
            passagem.cod_cli
      ORDER BY nome, num;
SELECT * FROM cliente_passagem;
```




Visões (Views)

```
--Eliminando a view  
DROP VIEW cliente_passagem;
```



Quais linhas retornar de um resultado

- Exemplo1: Selecionando as 2 passagens mais caras

```
SELECT num, valor
FROM passagem
ORDER BY valor DESC
LIMIT 2;
```

- Exemplo2: Selecionando as 2 cliente a partir do terceiro

```
SELECT nome
FROM cliente
ORDER BY nome
LIMIT 2
OFFSET 2;
```



Exercícios

Criar uma tabela chamada estados para armazenar informações sobre as Unidades Federativas do Brasil:

- Atributos: cod_uf, nome, sigla, populacao.
- Colocar as restrições de integridade necessárias.
- Inserir pelo menos os estados dos clientes já cadastrados.
- Alterar a tabela clientes adicionando uma chave estrangeira para esta nova tabela.
- Remover a coluna antiga.