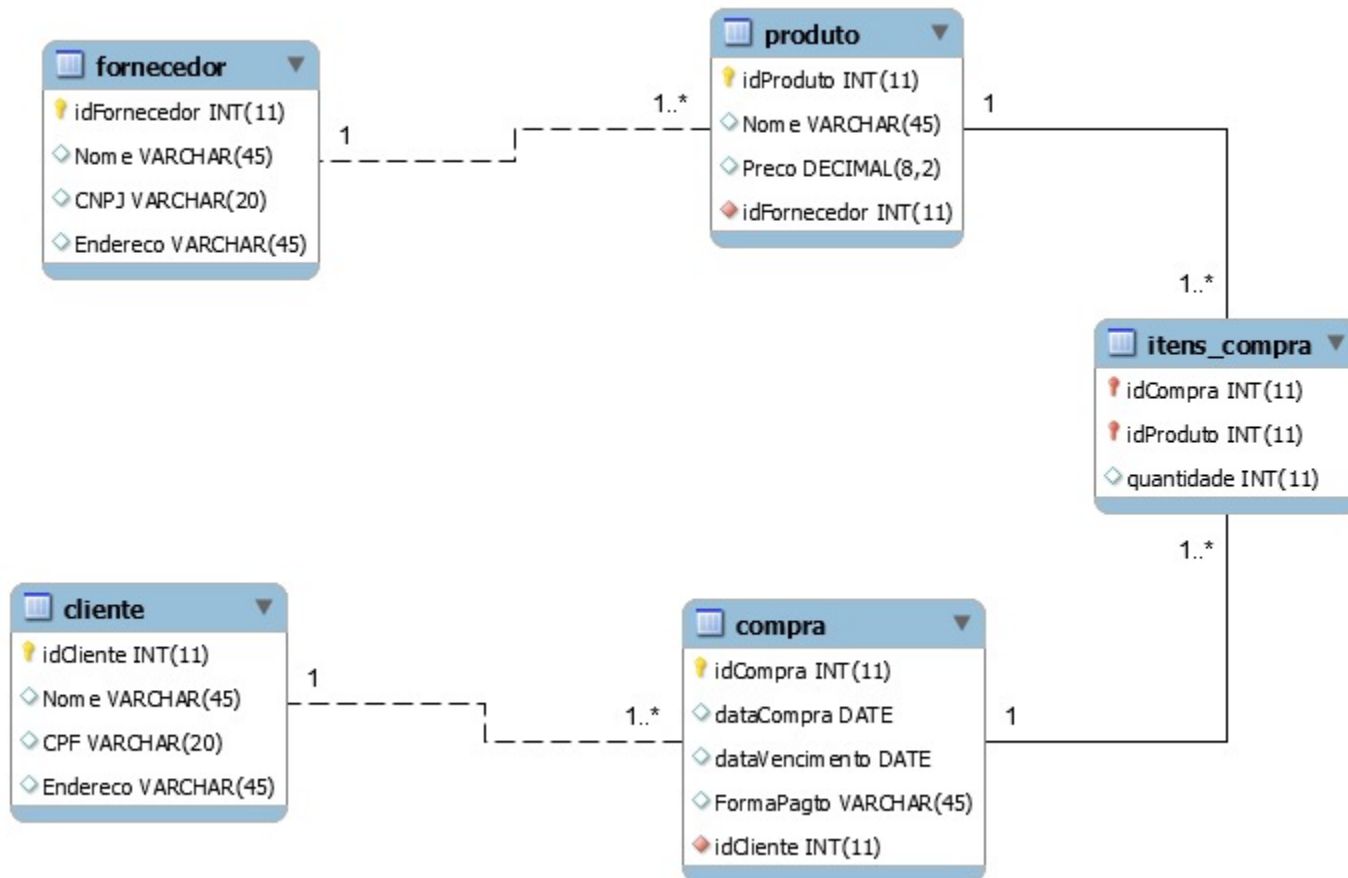


SQL PARA MYSQL

Subconsultas, views e joins

Prof. Bianca

SISTEMA DE COMPRAS



Workbench (UML Notation)



AS TABELAS COM DADOS

Cliente

idCliente	Nome	CPF	Endereco
201	Pedro Oliveira	231213	Rua das Hortências, 100 - Bragança Pta
222	João Souza	243456	Rua das Orquideas, 300 - Bragança Pta
321	Maria Souza	121234	Rua das Pedras, 20 - Bragança Pta
421	Ana Freire	198716	Rua das Hortências, 200 - Campinas
545	Silvia Oliveira	878765	Rua das Hortências, 300 - Bragança Pta

Fornecedor

idFornecedor	Nome	CNPJ	Endereco
1	CHAMEX	123123123	Rua Mogi das Cruzes, 101 Sao Paulo
2	HP	321123444	Rua da Fecilidade, 121 Sao Paulo
3	Elgin	667162432	Rua Fuxuco, 232 Rio de janeiro

Compra

idCompra	dataCompra	dataVencimento	FormaPagto	idCliente
10	2016-04-01	2016-05-02	CC	201
20	2016-04-01	2016-05-02	CC	321
30	2016-04-05	2016-05-02	CC	421
100	2016-04-10	2016-05-02	CC	421
112	2016-04-10	2016-05-02	CC	201

Produto

idProduto	Nome	Preco	idFornecedor
1	Papel A4	14.00	1
2	Cartucho HP	45.00	2
3	Etiqueta CD	17.00	1
4	Cd RW	5.00	2
5	Cd R	2.00	2

Itens_compra

idCompra	idProduto	quantidade
10	1	1
10	2	1
10	3	1
20	1	20
20	4	1
30	1	10
30	2	1
100	1	10
112	5	1

SUBCONSULTAS

SELECT <LISTA DE COLUNAS >

FROM <TABELAS>

WHERE <NOME DA COLUNA> <OPERADOR> <**SUBCONSULTA**>



OPERADORES LÓGICOS

OPERADOR RESULTADO

ALL	TRUE se tudo em um conjunto de comparações for TRUE.
ANY	TRUE se qualquer conjunto de comparações for TRUE.
EXISTS	TRUE se uma subconsulta tiver qualquer linha.
IN	TRUE se o operando for igual a um de uma lista de expressões.
NOT	Inverte o valor de qualquer outro operador booliano.
SOME	TRUE se algum conjunto de comparações for TRUE.



EXEMPLO DE CONSULTA COM IN

Seleciona tuplas a partir de uma lista de valores

Consulta: Selecione os produtos dos fornecedores 1 ou 2

SQL: **SELECT** * **FROM** produto **WHERE** idFornecedor **IN** (1,2);

idProduto	Nome	Preco	idFornecedor
1	Papel A4	14.00	1
2	Cartucho HP	45.00	2
3	Etiqueta CD	17.00	1
4	Cd RW	5.00	2
5	Cd R	2.00	2

Também pode ser usada a expressão:

*select * from produto where idfornecedor=1 or idfornecedor=2;*

EXEMPLO DE SUBCONSULTA

Seleciona tuplas a partir de uma lista de valores

Consulta: Selecione os dados dos produtos da compra 10

SELECT * **FROM** produto

WHERE idproduto **IN**

(**SELECT** Produto_idProduto **FROM** itens_compra

WHERE Compra_idCompra=10

);

idProduto	Nome	Preco	idFornecedor
1	Papel A4	14.00	1
2	Cartucho HP	45.00	2
3	Etiqueta CD	17.00	1

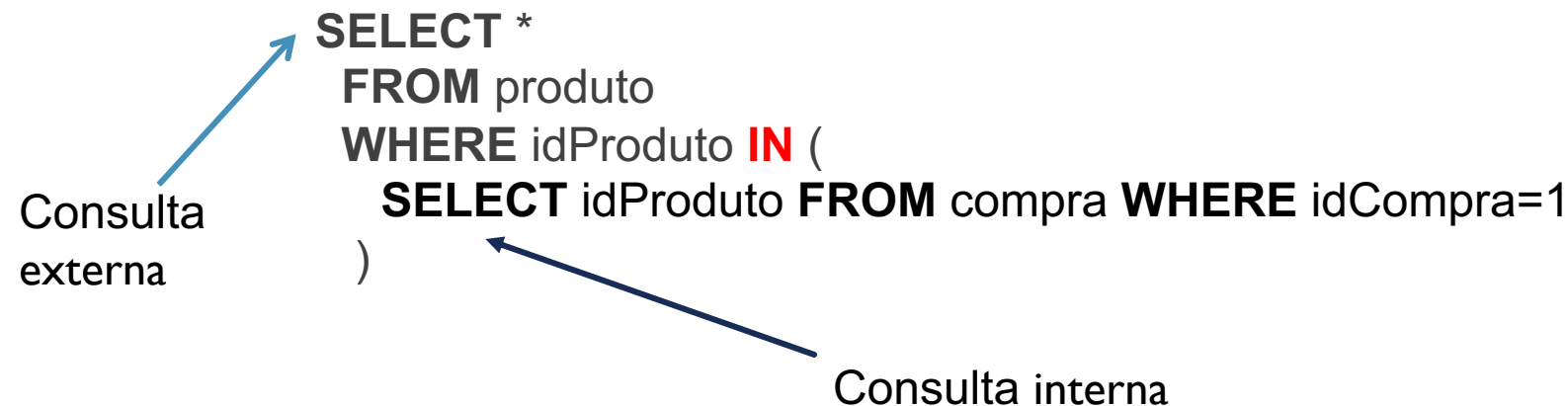
SUBCONSULTA

- A subconsulta é apenas uma instrução **SELECT** dentro de uma outra instrução **SQL**. Os resultados de uma subconsulta são usados na consulta externa, por isto, o tipo de dados da coluna retornada deve ser compatível com o tipo de dados da coluna utilizada na expressão de comparação da consulta externa.
- Exemplo: Obter os dados dos produtos da compra 1:

Consulta externa

```
SELECT *  
FROM produto  
WHERE idProduto IN (  
    SELECT idProduto FROM compra WHERE idCompra=1  
)
```

Consulta interna



ALL ,ANY, SOME

- Os operadores de comparação que introduzem uma subconsulta podem ser modificados pelas palavras-chave **ALL** ou **ANY**. **SOME** é um padrão ISO equivalente a ANY.
- As subconsultas introduzidas por um operador de comparação modificado retornam uma lista com zero ou mais valores e podem incluir uma cláusula GROUP BY ou HAVING.
- Para uma linha em uma subconsulta com **>ALL** satisfazer a condição especificada na consulta externa, o valor na coluna que introduz a subconsulta deve ser **maior do que cada valor na lista de valores retornada pela subconsulta**.
- Por exemplo, **>ALL (1, 2, 3)** quer dizer **maior do que 3**. **>ANY** quer dizer maior do que pelo menos um valor, isto é, maior do que o mínimo. Assim **>ANY (1, 2, 3)** quer dizer maior do que 1.

JUNÇÕES

OBJETIVO DA JUNÇÃO

- Combinar linhas de uma tabela com as linhas de outra tabela, com base na igualdade de valores de duas colunas (PK e FK) das tabelas envolvidas.

idProduto	Nome	Preco	idFornecedor	idFornecedor	Nome	CNPJ	Endereco
1	Papel A4	14.00	1	1	CHAMEX	123123123	Rua Mogi das Cruzes, 101 ...
2	Cartucho HP	45.00	2	2	HP	321123444	Rua da Felicidade, 121 ...
3	Etiqueta CD	17.00	1	3	Elgin	667162432	Rua Fuxuco, 232 Rio de ...
4	Cd RW	5.00	2				
5	Cd R	2.00	2				

idProduto	Nome	Preco	idFornecedor	idFornecedor	Nome	CNPJ	Endereco
1	Papel A4	14.00	1	1	CHAMEX	123123123	Rua Mogi das Cruzes, 101 ...
2	Cartucho HP	45.00	2	2	HP	321123444	Rua da Felicidade, 121 ...
3	Etiqueta CD	17.00	1	1	CHAMEX	123123123	Rua Mogi das Cruzes, 101 ...
4	Cd RW	5.00	2	2	HP	321123444	Rua da Felicidade, 121 ...
5	Cd R	2.00	2	2	HP	321123444	Rua da Felicidade, 121 ...



CONDIÇÃO NECESSÁRIA PARA JUNÇÃO

Para fazer uma junção é necessário que as tabelas tenham relacionamento.

idProduto	Nome	Preco	idFornecedor
1	Papel A4	14.00	1
2	Cartucho HP	45.00	2
3	Etiqueta CD	17.00	1
4	Cd RW	5.00	2
5	Cd R	2.00	2



idFornecedor	Nome	CNPJ	Endereco
1	CHAMEX	123123123	Rua Mogi das Cruzes, 101 Sao Paulo
2	HP	321123444	Rua da Facilidade, 121 Sao Paulo
3	Elgin	667162432	Rua Fuxuco, 232 Rio de janeiro

EXPRESSÃO SQL COM JOIN

```
SELECT *  
FROM    produto p, fornecedor f  
WHERE   p.idFornecedor = f.idfornecedor
```

Ou

```
SELECT *  
FROM    produto p INNER JOIN fornecedor f  
        ON p.idFornecedor = f.idfornecedor
```

RESULTADO DA JUNÇÃO

- Uma tabela com todas as colunas das tabelas participantes da junção, o número de linhas, será igual ao número de linhas de uma das tabelas.

idProduto	Nome	Preco	idFornecedor	idFornecedor	Nome	CNPJ	Endereco
1	Papel A4	14.00	1	1	CHAMEX	123123123	Rua Mogi das Cruzes, 101 ...
2	Cartucho HP	45.00	2	2	HP	321123444	Rua da Felicidade, 121 ...
3	Etiqueta CD	17.00	1	1	CHAMEX	123123123	Rua Mogi das Cruzes, 101 ...
4	Cd RW	5.00	2	2	HP	321123444	Rua da Felicidade, 121 ...
5	Cd R	2.00	2	2	HP	321123444	Rua da Felicidade, 121 ...

idProduto	Nome	Preco	idFornecedor	idFornecedor	Nome	CNPJ	Endereco
1	Papel A4	14.00	1	1	CHAMEX	123123123	Rua Mogi das Cruzes, 101 ...
2	Cartucho HP	45.00	2	2	HP	321123444	Rua da Felicidade, 121 ...
3	Etiqueta CD	17.00	1	1	CHAMEX	123123123	Rua Mogi das Cruzes, 101 ...
4	Cd RW	5.00	2	2	HP	321123444	Rua da Felicidade, 121 ...
5	Cd R	2.00	2	2	HP	321123444	Rua da Felicidade, 121 ...

LEFT JOIN

- A cláusula **LEFT JOIN** ou **LEFT OUTER JOIN** permite obter não apenas os dados relacionados de duas tabelas, mais também os dados não relacionados encontrados na tabela à esquerda da cláusula **JOIN**. Caso não existam dados relacionados entre as tabelas à esquerda e a direita do **JOIN**, os valores resultantes de todas as colunas da lista de seleção da tabela à direita serão nulos.



Exemplo: Obter fornecedores que não tem produtos associados

- **SELECT** Fornecedor.nome
FROM Fornecedor **LEFT JOIN** Produto
ON Fornecedor.idFornecedor=Produto.idFornecedor
WHERE produto.id_produto **is NULL**;

nome
Elgin

RIGHT JOIN

- Faz o mesmo que o LEFT JOIN, entretanto, obtém todos os dados da tabela à direita do JOIN, relacionados ou não relacionados com os dados da outra tabela.
- EXEMPLO: Para testar o RIGHT JOIN basta inverter a ordem das tabelas na cláusula FROM do slide anterior e trocar o LEFT por RIGHT, para produzir o mesmo resultado. Tente !

VIEW

VISÕES

- Uma visão é uma representação dos dados armazenados em uma ou mais tabelas.
- Uma visão não corresponde necessariamente à forma como os dados estão armazenados
- Usa-se visões para armazenar consultas que pretendemos repetir com frequência ou compartilhar com outros usuários, .
- Outro uso da visão é a possibilidade de restringir o acesso aos dados por alguns usuários.
- Uma visão pode ser utilizada na cláusula from de outras consultas SQL
- Visões são criadas com o comando `CREATE VIEW`

VANTAGENS DAS VIEWS

- **Reuso:** as views são objetos de caráter permanente. Pensando pelo lado produtivo isso é excelente, já que elas podem ser lidas por vários usuários simultaneamente.
- **Segurança:** as views permitem que ocultemos determinadas colunas de uma tabela. Para isso, basta criarmos uma view com as colunas que achamos necessário que sejam exibidas e as disponibilizarmos para o usuário.
- **Simplificação do código:** as views nos permitem criar um código de programação muito mais limpo, na medida em que podem conter um SELECT complexo. Assim, criar views para os programadores a fim de poupá-los do trabalho de criar SELECT's é uma forma de aumentar a produtividade da equipe de desenvolvimento.

VIEW

- **CREATE VIEW** name [(col₁, col₂, ..., col_n)]
- **AS** <select> [WITH CHECK OPTION];
- obs:
- (col₁, col₂, ..., col_n) (opcional): corresponde às colunas a serem adicionadas à visão
- <select> (obrigatório) corresponde a uma expressão de consulta qualquer
- **WITH CHECK OPTION** (opcional): evita a atualização e inclusão de dados que não satisfaçam a cláusula select

EXEMPLO DE VIEW

- **CREATE VIEW** Vproduto **AS**

SELECT Produto.*, Fornecedor.Nome

FROM Produto **INNER JOIN** Fornecedor **ON**

Produto.idFornecedor=Fornecedor.IdFornecedor

- Exemplo de uso da view:

SELECT * FROM Vproduto;