Banco de dados II

Aula 2

Setembro de 2024

Agenda Revisão SQL

- JOIN
- Agrupamentos
- Subqueries
- Exercícios

01

Relembrando os JOINS

Junção interna (INNER JOIN)

É uma relação binária que irá receber duas relações(tabelas) e retornará uma terceira relação atendendo às condições de junção, veja o exemplo de duas relações abaixo.

Professor

Nome	Area	Disciplina
Claudio	Enfermagem	Laboratorios
Nathalia	Psicologia	Psicologia infantil
Antonio	Engenharia	Calculo II
Beatriz	Redes	Segurança
Bernardo	Redes	Servidores

Turma

Disciplina	Codigo
Segurança	AB1D
Criptografia	CPD1F
Segurança	EB3F
Psicologia infantil	PI4M
Calculo II	CCBD2
Calculo II	CCBD4

Junção interna (INNER JOIN)

Resultado:

Nome	Area	Disciplina	Codigo
Beatriz	Redes	Seguranca	AB1D
Nathalia	Psicologia	Psicologia infantil	PI4M
Antonio	Engenharia	Calculo II	CCBD2
Beatriz	Redes	Seguranca	EB4F
Antonio	Engenharia	Calculo II	CCBD4

Junção interna (INNER JOIN)

Até agora vimos como realizar pesquisas em uma única tabela. Contudo, na montagem do nosso modelo de dados sempre temos diversas tabelas. Logo é necessário sabermos como vincular a informação dessas tabelas de forma a mostrar a informação de maneira correta. A isto é dado o nome de união de tabelas (join).

Como visto anteriormente, a união entre as entidades do nosso modelo lógico se dá por meio de chaves primárias e estrangeiras. Essas chaves são, na representação física do modelo, as colunas que as tabelas têm em comum. No decorrer apresentarei as diversas formas de unir colunas e como implementá-las em SQL.

Junção interna (INNER JOIN)

Para realizar a união de tabelas, basta acrescentarmos após a cláusula FROM do comando SELECT as tabelas que queremos unir. Devemos colocar na cláusula WHERE a condição de união das tabelas, ou seja, as respectivas chaves primária e estrangeira. Sintaxe:

Junção interna (INNER JOIN)

```
SELECT
Tabela1.coluna1,
Tabela1.coluna2,
Tabela2.coluna1,
Tabela2.coluna2
FROM Tabela1, Tabela2
WHERE Tabela1.chave_primaria = Tabela2.chave_estrangeira
```

Obs.: Sempre que utilizamos mais de uma tabela em um select, é realizado um produto cartesiano entre as tabelas, isso é, uma combinação das linha de uma tabela com todas as linhas da outra tabela.

Junção interna (INNER JOIN)

Pelo que vimos em aulas passadas, o nome das chaves primárias e o nome das chaves estrangeiras que são correspondentes são iguais, por esse motivo, a comparação entre as colunas, que garante a vinculação correta entre as linhas, deve ser possível diferença as tabelas ais quais as colunas pertencem. Veja como ficaria o select apresentado anteriormente

Junção interna (INNER JOIN)

```
SELECT
Tabela1.coluna1,
Tabela1.coluna2,
Tabela2.coluna1,
Tabela2.coluna2
FROM Tabela1 T1, Tabela2 T2
WHERE T1.chave_primaria = T2.chave_estrangeira
```

Junção externa (OUTER JOIN)

Levam em consideração os casos que não tem correspondência, podendo ser:

- Junção externa a esquerda(left outer Join)
- Junção externa a direita(right outer Join)
- Junção externa total(full outer Join)

A junção externa será estudada no futuro.

02

Funções agregadas

Como funcionam

São funções que funcionam como funções convencionais de qualquer linguagem de programação estrutural, mas no caso os comandos de SQL retornam uma parte do banco de dados, estas funções agregadas retornarão uma informação única da parte do banco de dados consultado. Segue as funções abaixo:

AVG(parâmetros)	Retorna a média dos valores
MAX(parâmetros)	Retorna o valor máximo
MIN(parâmetros)	Retorna o valor mínimo
SUM(parâmetros)	Retorna a soma de todos os parâmetros
COUNT(parâmetros)	Retorna o número de linhas com o parâmetro

Funções de Agrupamento

Cláusula GROUP BY:

```
SELECT COLUNAS
FROM TABELA
WHERE  CONDIÇÃO
GROUP BY CAMPOS
```

Funções de Agrupamento

Grupamento de Colunas: A cláusula GROUP BY permite que você crie sumários a partir de grupos de linhas, agrupando valores iguais em colunas especificadas. Por exemplo, se você agrupar por uma coluna "Cidade", você terá um grupo de resultados para cada cidade diferente na sua tabela.

Sumários: As funções de agregação são aplicadas a cada grupo separadamente e resumem os dados de alguma forma. Por exemplo, você pode contar o número de linhas em cada grupo usando COUNT(), ou calcular a média de uma coluna numérica com AVG().

OBS.: Não é possível selecionar colunas que não estão incluídas na cláusula GROUP BY ou que não estão encapsuladas em uma função de agregação, pois isso resultaria em um erro. O SQL não saberia qual valor retornar para a coluna não agrupada, já que haveria múltiplos valores possíveis para cada grupo.

Exempos de Agrupamentos

Liste a quantidade de Clientes por país;

```
Select pais, count(*)
From clientes
Group by pais
```

Liste a quantidade de pedidos no mês de Janeiro de 2007;

```
Select count(*), date_format(datadopedido, '%m/%y')
From clientes
Group by date_format(datadopedido, '%m/%y')
```

Exempos de Agrupamentos

Media de fretes por país;

Select avg(frete), pais

From clientes

Group by pais;

03

Usando subqueries

O que são subqueries

Subqueries, também conhecidas como subconsultas ou consultas aninhadas, são consultas SQL dentro de outra consulta SQL. Elas são usadas para realizar operações que normalmente requerem várias etapas em uma única consulta, permitindo uma forma de agrupar dados que não podem ser agregados de outra forma. Subqueries podem ser usadas em diversas cláusulas, incluindo SELECT, FROM, WHERE, e HAVING.

São usadas em:

- Predicados de comparação
  - Predicado IN
  - Predicados ALL ou ANY
  - Predicado EXISTS

Observe exemplo

Qual o código e nome dos Clientes que moram nas mesma cidades que os Fornecedores?

```
SELECT codigodocliente, nomedocliente FROM clientes;
SELECT cidade FROM fornecedores;
```

Agora a integração das duas consultas:

```
SELECT codigodocliente, nomedaempresa FROM clientes WHERE cidade
in ( SELECT cidade FROM fornecedores);
```

Observe exemplo

Qual o nome dos clientes que tem pedidos cadastrados;

```
SELECT nomedaempresa
    FROM              clientes
    WHERE EXISTS
    (SELECT  * FROM  pedidos
    WHERE clentes.codigodopedido = pedidos.codigodocliente                    );
```

Observe exemplo

Qual o nome dos clientes que não tem pedidos cadastrados;

```
SELECT nomedaempresa
FROM                clientes
WHERE NOT EXISTS
(SELECT  * FROM  pedidos
WHERE clentes.codigodopedido = pedidos.codigodocliente);
```

04

Exercícios Síncronos

Crie o seguinte Banco de dados

Funcionario(id, nome, sexo, cargo, salario);

Fornecedor(id, nome, cidade, produto);

Pedido(id, id_funcionario, id_cliente, descricao, observação, cidade, produto, data);

Cliente(id, nome, idade, sexo, renda, cidade);

Faça as consultas abaixo

Liste a quantidade de pedidos feitos por funcionários;

Liste a quantidade de clientes por país e cidade;

Liste o Código do Fornecedor, o código da Categoria, a média de preço, o preço máximo e mínimo da tabela de produtos grupados por fornecedor e categoria;

Liste a quantidade de pedidos da tabela pedidos solicitados por ano,mês da data do pedido;

Liste a quantidade de pedidos feitos por funcionários de 20 à 30 anos;

Obrigado!

Alguma dúvida?

jheymesso.Cavalcanti@unicap.br