|  |  |  |
| --- | --- | --- |
| **Professor: LUIS CARLOS DOS SANTOS** | | **Curso: INFORMÁTICA PARA INTERNET INTEGRADO AO ENSINO MÉDIO** |
| **Disciplina: MODELAGEM E DESENVOLVIMENTO DE BANCO DE DADOS** | | **Turma: 2Y** |
| **Aluno: Guilherme Galante** | | **Número: 11** |
| **Aluno: Vinicius Rodrigues** | | **Número: 28** |
| **Data: 23/08/2017** | **Exercícios de Fixação** | **Conceito:** |

**Responda as questões abaixo de acordo com as notas de aulas:**

**1-) O que são JOINS em Banco de Dados?**

**O JOIN é uma cláusula da linguagem SQL, que permite criar consultas combinando resultados de uma ou duas tabelas através de valores comuns entre uma ou varias colunas de cada tabela.**

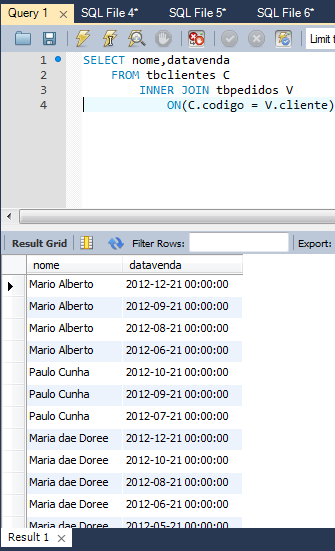
**2-) Explique como funcionam o INNER JOIN, LEFT JOIN e RIGHT JOIN.**

**O INNER JOIN é considerado o tipo de JOIN ‘’padrão’’. Eliminando o INNER obteremos o mesmo resiltado;**

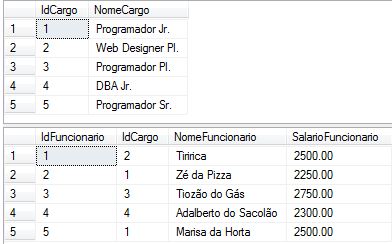
**Os comandos LEFT JOIN e RIGHT JOIN fazem intersecção entre as tabelas, porem, cada um faz a intersecção baseada em sua posição.**

***3-) Leia o artigo abaixo e desenvolva os exemplos no WorkBench. Atenção quanto aos scripts, pois os mesmos foram gerados e executados no MS-SQL Server 2008. Você deverá realizar os devidos ajustes quanto a sintaxe e escrever abaixo de cada um dos scripts, como executou em MySql no SGBD WorkBench***

**INNER JOIN –** A cláusula **INNER JOIN** permite usar um operador de comparação para comparar os valores de colunas provenientes de tabelas associadas. Por meio desta cláusula, os registros de duas tabelas são usados para que sejam gerados os dados relacionados de ambas. Usamos as cláusulas **WHERE** e **FROM** para especificar esse tipo de associação.



**4-) Crie as tabelas Cargo e Funcionario e insira os registros como se segue. Depois você deve registrar os script que utilizou para inserir os registros, imagem da estrutura da tabela criada no WorkBench e a consulta dos dados cadastrados no WorkBench conforme abaixo:**



Repare que temos a coluna **IdCargo** nas duas tabelas, porém, ela possui finalidades distintas: enquanto na tabela **Cargo**, ela é chave primária, na tabela **Funcionario** ela é chave estrangeira.

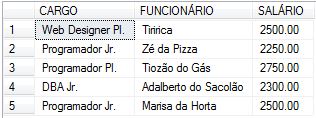
Assim, a associação entre as tabelas é feita pela coluna **IdCargo** e podemos identificar os cargos existentes e o nome dos funcionários que desempenham cada um deles. Usamos então a cláusula **INNER JOIN** para obtermos os dados relacionados das duas tabelas, para que sejam retornados todos os cargos ocupados pelos funcionários, bem como todos os funcionários que desempenham algum cargo. Veja como isso é feito no script abaixo com a sintaxe do SQL Server da Microsoft

SELECT C.NOMECARGO [CARGO], F.NOMEFUNCIONARIO AS [FUNCIONÁRIO], F.SALARIOFUNCIONARIO AS [SALÁRIO] FROM CARGO AS C

INNER JOIN FUNCIONARIO AS F

ON C.IDCARGO = F.IDCARGO

**5-) Desenvolva o script acima para o MySql no SGBD WorkBench conforme abaixo:**



Podemos usar também a cláusula **WHERE** e termos o mesmo resultado. O script ficará assim:

SELECT C.NOMECARGO [CARGO], F.NOMEFUNCIONARIO AS [FUNCIONÁRIO], F.SALARIOFUNCIONARIO AS [SALÁRIO] FROM CARGO AS C, FUNCIONARIO AS F

WHERE C.IDCARGO = F.IDCARGO

Independente da cláusula usada, o relacionamento entre as tabelas será feito, como dito anteriormente, por meio da chave primária da tabela **Cargo** e da chave estrangeira da tabela**Funcionario**. Note que, no retorno de nossas consultas, o único cargo que não é exibido pra nós é o **Programador Sr.** já que não há nenhum funcionário relacionado a este cargo. Para que exibamos também este cargo usamos a cláusula que será explicada a seguir.

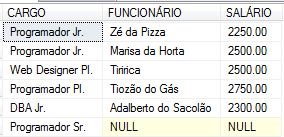
**LEFT JOIN –** A cláusula **LEFT JOIN** ou **LEFT OUTER JOIN** permite obter não apenas os dados relacionados de duas tabelas, mais também os dados não relacionados encontrados na tabela à esquerda da cláusula **JOIN**. Caso não existam dados relacionados entre as tabelas à esquerda e a direita do **JOIN**, os valores resultantes de todas as colunas da lista de seleção da tabela à direita serão nulos. Para exemplificar melhor, vejamos os exemplos das tabelas **Cargo** e **Funcionário**. Como dito anteriormente, o único cargo que não contém funcionário vinculado a ele é o **Programador Sr.** Para obtermos mesmo assim esse cargo, usamos a cláusula **LEFT JOIN** à esquerda do sinal de igual (**=**), como no script abaixo:

**6-) Desenvolver o scrit abaixo no MySql no SGBD WorkBench**

SELECT C.NOMECARGO [CARGO], F.NOMEFUNCIONARIO AS [FUNCIONÁRIO], F.SALARIOFUNCIONARIO AS [SALÁRIO] FROM CARGO AS C

LEFT JOIN FUNCIONARIO AS F

ON C.IDCARGO = F.IDCARGO



**RIGHT JOIN –** Ao contrário do **LEFT JOIN**, a cláusula **RIGHT JOIN** ou **RIGHT OUTER JOIN**retorna todos os dados encontrados na tabela à direita de **JOIN**. Caso não existam dados associados entre as tabelas à esquerda e à direita de **JOIN**, serão retornados valores nulos.

Suponhamos que a posição das tabelas usadas nos exemplos anteriores foi trocada. Se mesmo assim desejamos obter o mesmo resultado obtido anteriormente, podemos usar a cláusula **RIGHT JOIN**, assim iremos conseguir tanto os dados relacionados como os não relacionados disponíveis na tabela à direita da cláusula **JOIN**.

**7-) Desenvolver o scrit abaixo no MySql no SGBD WorkBench**

SELECT C.NOMECARGO [CARGO], F.NOMEFUNCIONARIO AS [FUNCIONÁRIO], F.SALARIOFUNCIONARIO AS [SALÁRIO] FROM FUNCIONARIO AS F

RIGHT JOIN CARGO AS C

ON F.IDCARGO = C.IDCARGO

Que terá o seguinte retorno:

+----------------+--------------+--------+

| cargo | func | salario|

+----------------+--------------+--------+

| Web Designer PI| tiririca | 2500 |

| Programador Jr | ze da pizza | 2250 |

| programador PI | Tiozao do gas| 2750 |

| DBA Jr | Adalberto | 2300 |

| Programador Jr | Marisa | 2500 |

+----------------+--------------+--------+

Perceba que somente inverti a ordem das tabelas nas cláusulas **FROM** e **RIGHT JOIN** em relação ao exemplo com **LEFT JOIN**.