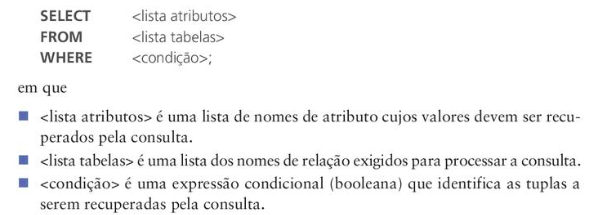
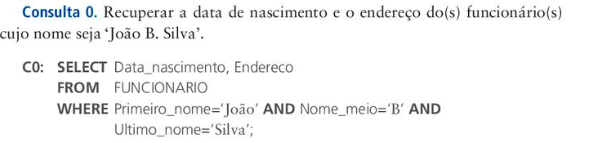
**6.3 Consultas de recuperação básica em SQL**

A instrução SELECT é a instrução básico para a recuperação de dados de um banco de dados. É importante ressaltar que a operação SELECT não é igual ao select da álgebra relacional, visto que existem diversos tipos de instruções SELECT.

**6.3.1 A Estrutura SELECT-FROM-WHERE das consultas SQL básicas**

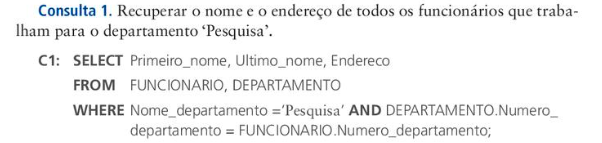
****

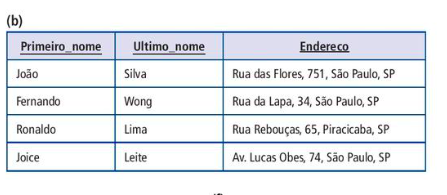
Em SQL, os operadores básicos de comparação lógica para comparar valores de atributos entre si e com constantes literais são: =, <=, >, >= e <>(diferente)





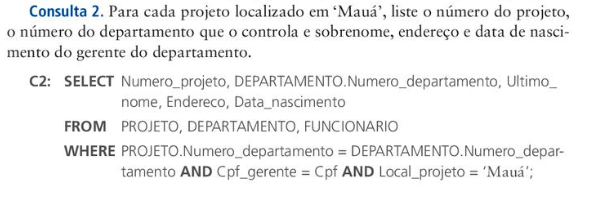
Pode-se pensar em uma váriavel de tupla implícita na consulta SQL variando ou repetindo sobre cada tupla individual na tabela FUNCIONARIO e avaliando a condição na cláusula WHERE. Somente as tuplas que satifsfazem a condição, ou seja, aquelas tuplas para as quais a condição é avaliada como TRUE após substituir seus valores de atributo correspondente.





Quando se iguala um nome de atributo a outro nome de atributo a operação de junção está sendo feita, pois combina duas tuplas, uma de DEPARTAMENTO e uma de FUNCIONARIO.

Uma consulta que envolve apenas condições de seleção e junção mais atributos de projeção é conhecida como uma consulta **seleção-projeção-junção**.





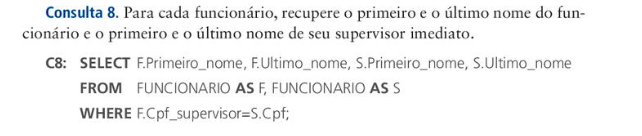
A condição de junção PROJETO.Numero\_departamento = DEPARTAMENTO.Numero\_departamento relaciona uma tupla de projeto à sua tupla de departamento que o controla, enquanto a condição Cpf\_gerente = Cpf relaciona a tupla do departamento que o controla à tupla de funcionario que gerencia esse departamento.

**6.3.2 Nomes de atributos ambíguos, apelido, renomeação e variáveis de tupla**

Em SQL, o mesmo nome pode ser usado para dois ou mais atributos, desde que estejam em relações diferentes. Caso isso aconteça, e uma consulta em múltiplas relações se referer a dois ou mais atributos com o mesmo nome, é **NECESSÁRIO** qualificar o nome do atributo com o nome da relação para evitar ambiguidade. Isso é realizado prefixando o nome da relação ao nome do atributo e separando os dois por um ponto.

Nomes de relações podem ser renomeadas para evitar repetição na digitação.

Isso também pode acontecer, caso a consulta seja feita na mesmsa relação, para isso é necessário realizar:



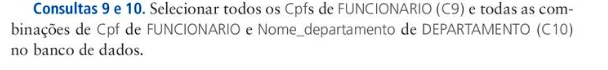
Neste caso, é necessário declarar nomes de relação alternativos F e S, chamados apelidos, para a relação FUNCIONARIO. Um apelido pode vir após a palavra-chave AS, ou diretamente após o nome da relação, por exemplo, escrevendo FUNCIONARIO F na cláusula FROM. Também é possível renomear os atributos da relação dentro da consulta em SQL.

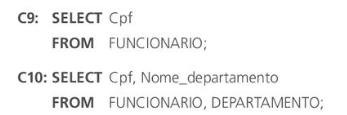


Com essa instrução, Pn torna-se um apelido para Primeiro\_nome, Nm para Nome\_meio, e assim por diante. Na consulta 8, podemos pensar F e S como duas cópias diferente das relação FUNCIONARIO, a primeira,F, representa funcionário no papel de supervisionados, a segunda,S, representa os funcionarios no papel de supervisores.

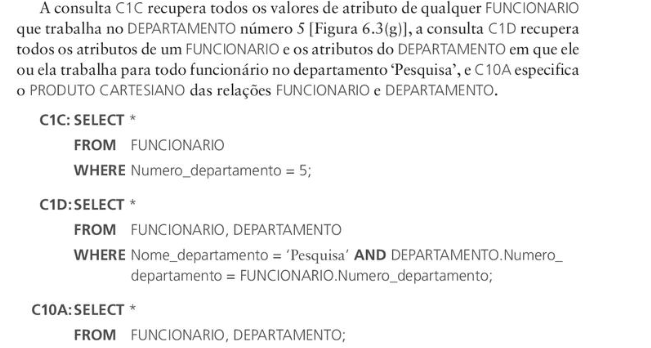
**6.3.3 Cláusula WHERE não especificada e uso do asterisco**

A falta de uma cláusula WHERE indica que não há condição sobre a seleção de tuplas, logo, todas as tuplas da relação especificada na cláusula FROM se qualificam e são selecionadas para o resultado da consulta. Se mais de uma relação for especificada na cláusula FROM e não houver uma cláusula WHERE, então o PRODUTO CARTESIANO dessas relações será selecionado.



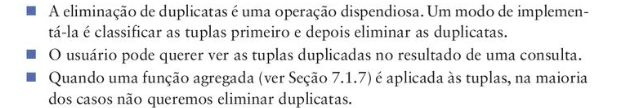


Dessa forma, é extremamente importante especificar cada condição de seleção e junção na cláusula WHERE. Caso alguma condição desse tipo for esquecida, o resultado poderá ser relações incorretas e muito grandes.

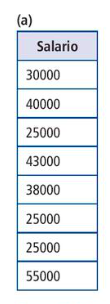
Para recuperar todos os valores de atributos das tuplas selecionadas, não precisamos listas os nomes de atributo explicitamente em SQL, basta usar o asterisco(\*), que significa todos os atributos.

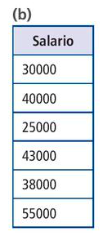
**6.3.4 Tabelas como conjuntos em SQL**

A SQL normalmente trata uma tabela não como um ocnjunto, mas como um multiconjunto, tuplas duplicadas podem aparecer mais de uma vez em uma tabela e no resultado de uma consulta. A SQL não elimina automaticamente tuplas duplicadas nos resultados das consultas pelos motivos:

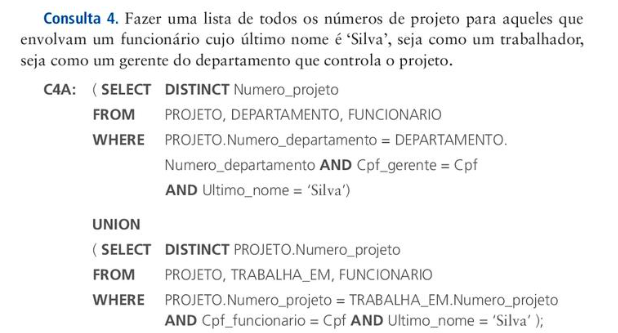


Uma tabela SQL com uma chave é restrita a ser um conjunto, uma vez que o valor de chave precisa ser distinto em cada tupla. Caso quisermo eliminar as tuplas duplicadas do resultado de uma consulta SQL, usamo a palavra-chave DISTINCT na clásula SELECT, significando que apenas as tuplas distintas deverão permanecer no resultado. Em geral uma consulta SELECT DISTINCT elimina duplicatas enquanto uma consulta com SELECT ALL não elimina. A C11 recuperar o salário de cada funcionário, se vários funcionários tiverem o mesmo salário esse valor de salário aparecerá muitas vezes no resultado da consulta. Se estivremos interessados apenas em valores de salários dinstintos queremos que cada valor apareça uma vez. Usando a palavra DISTINCT isso é possível.





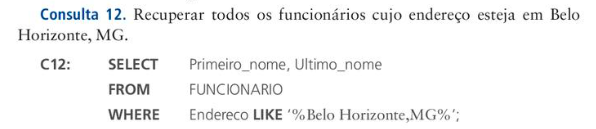
A SQL incorporou algumas das operações de conjunto da teorica dos conjuntos. A união de conjuntos UNION, diferença de conjunto EXCEPT e interesecção de conjunto INTERSECT. As relações resultantes dessas operações de conjunto são conjuntos de tuplas, ou seja, tuplas duplicadas são eliminadas do resultado. Essas operações se aplicam apenas a relações compatíves com o tipo, de modo que é necessário garantir que as duas relações em que aplicamos a operação tenham os mesmos atributos e que os atributos apareçam na mesma ordem nas duas relações.



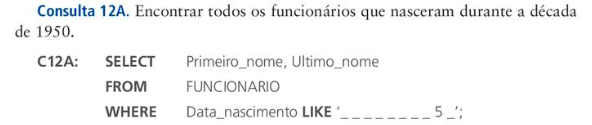
A primeira consulta SELECT recupera os projetos que envolvem Silva, como um gerente do departamento que controla o projeto e a segunda recupera os projetos que envolvem um Silve com um trabalhador no projeto.

**6.3.5 Combinação de padrão de subcadeias e operadores ariméticos**

O primeiro recurso permite condições de comparação apenas sobre parte de uma cadeia de caracteres usando o operador de comparção LIKE. Isso pode ser usado para combinação de padrão de cadeia de caracteres. Cadeia de caracteres parciais são especificados com o uso de dois caracteres reservados: % substitui um número qualquer de zero ou mais caracteres, e o sublinhado, substitui um único caractere.

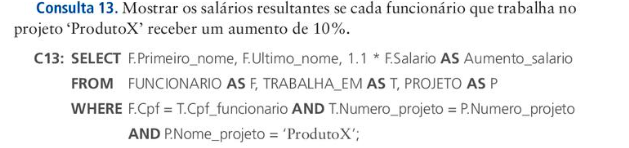
****

Para recuperar toos os funcionários que nasceram durante a década de 1950, podemos usar a consulta C12A. Aqui ‘5’ precisa ser o nono caractere cada cadeia, de acordo com o nosso formato de data.

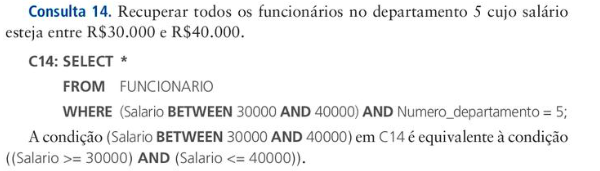


Caso um sublinhado ou % for necessário como um caractere literal na cadeia, este deve ser precedido por um caractere de escape, que é especificado após a cadeia usando a palavra-chave ESCAPE. Por exemplo, ‘AB\\_CD\%EF’ESCAPE’\’ representa a cadeia literal ‘AB\_CD%EF’, pois \ é especificado como caractere de escape.

Caso queira aplicar um operador arimético a relação podemos realizar:

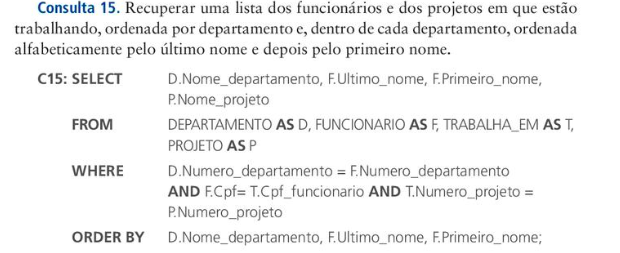


O operador de comparação que pode ser usado por conveniência é BETWEEN que está ilustrado abaixo:



**6.3.6 Ordenação dos resultados da consulta**

A SQL Permite que o usuário ordene as tuplas no resultado de uma consulta pelos valores de um ou mais atributos que aparecem no resultado da consulta, usando a cláusula ORDER BY.



A ordem padrão está em ordem crescente de valores. Podemos especificar a palvra DESC se quisermos ver o resultado em uma ordem decrescente de valores. A palavra ASC pode ser usada para especificar a ordem crescente explicitamente. Por exemplo, se quisermos a ordem alfabetica decrescente de NOme\_departamento e a ordem crescente de Ultimo\_nome, Primeiro\_nome, a cláusula ficaria:



**6.4 Instruções INSERT, DELETE e UPDATE em SQL**

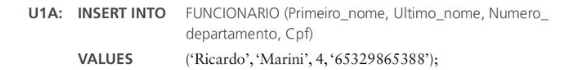
Em SQL, três comandos podem ser usados para modificar o banco de daods: Insert, Delete e UPDATE.

**6.4.1 O comando INSERT**

Em sua forma mais simples INSERT é usado para acrescentar uma única tupla a uma relação. Temos de especificar o nome da relação e uma lista de valores para a tupla. Os valores devem ser listados na mesma ordem em que os atributos correspondentes foram especificados no comando CREATE TABLE. Por exemplo, para acrescentar uma nova tupla à relação FUNCIONARIO podemos utilizar o comando:

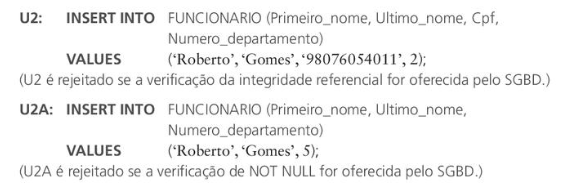
****

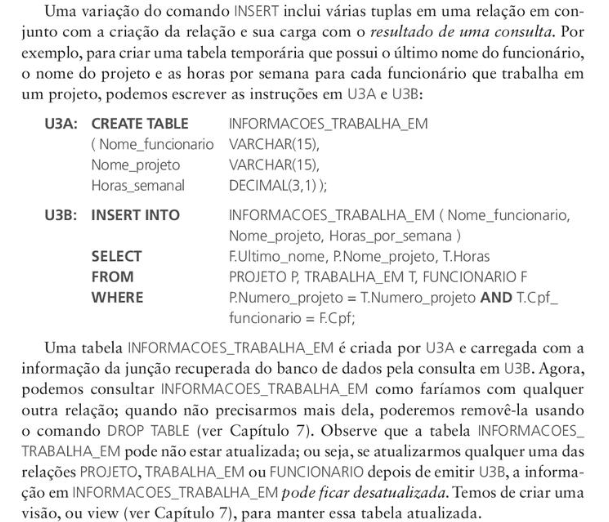
Uma segunda forma da instrução INSERT permite que o funcionário especifique nomes de atributos explícitos que correspondem aos valores fornecidos no comando INSERT. Isso é útil se uma relação tiver muitos atributos, mas apenas alguns deles recebem valores em uma nova tupla. Porém os valores precisam incluir todos os atributos com a especificação NOT NULL e nenhum valor padrão. Os atributos com NULL permitido ou com valores DEFAULT são aqueles que podem ser omitidos. Por exemplo, para inserir uma tupla para um novo FUNCIONARIO do qual conhecemos apenas os atributos Primeiro\_nome, Ultimo\_nome, Numero\_departamento e Cpf, podemos fazer:

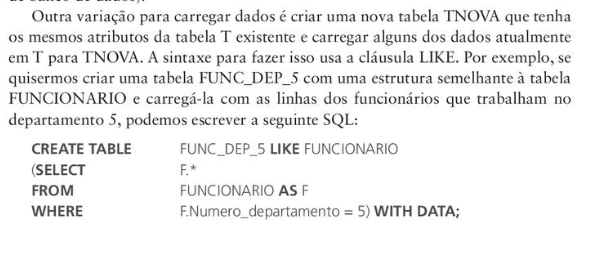


Os atributos não especificados em U1A são definidos como seu valor DEFAULT ou NULL, e os valores são listados na mesma ordem que os atributos são listados no próprio comando INSERT.

O SGBD pode rejeitar a inserção caso essa viole alguma restrição do BD, como por exemplo, não informar a chave primária, ou informar algum valor inválido para um atributo. COmo:

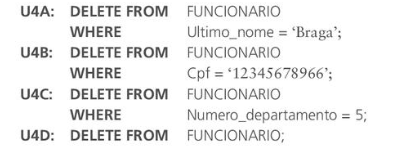






**6.4.2 O comando DELETE**

O comando DELETE remove tuplas de uma relação. Ele inclui uma cláusula WHERE, semelhante à que é usada em uma consulta SQL, para selecionar as tuplas a serem excluídas. As tuplas são explicitamente excluídas de apenas uma tapela por vez. No entanto a exclusão pode ser propagar para as tuplas em outras relações se ações de disparo referencial forem especificadas nas restrições de integridade referencial. Uma cláusula WHERE inexistente especifica que todas as tuplas da relação deverão ser excluídas, porém a tabela permanece no banco de dados. Para removê-la deve-se usar o comando DROP TABLE.



**6.4.3 O comando UPDATE**

O comando UPDATE é usado apra modificar os valores de atributo de uma ou mais tuplas selecionadas. Assim como o comando DELETE, uma cláusula WHERE no comando UPDATE, seleciona as tuplas a serem modificadas em uma única relação. No entanto, a atualização de uma chave primária pode ser propagada para os valores de chave estrangeira das tuplas em outras relação se tal ação de disparo referencial, for especificada nas restrição de integridade referencial da DDL.

Uma cláusula SET adicional no comando UPDATE especifica os atributos a serem modificados e seus novos valores. Por exemplo, para alterar o local e o numero de departamento que controla o número de projeto 10 para santo andré e 5 respectivamente, usamos:

