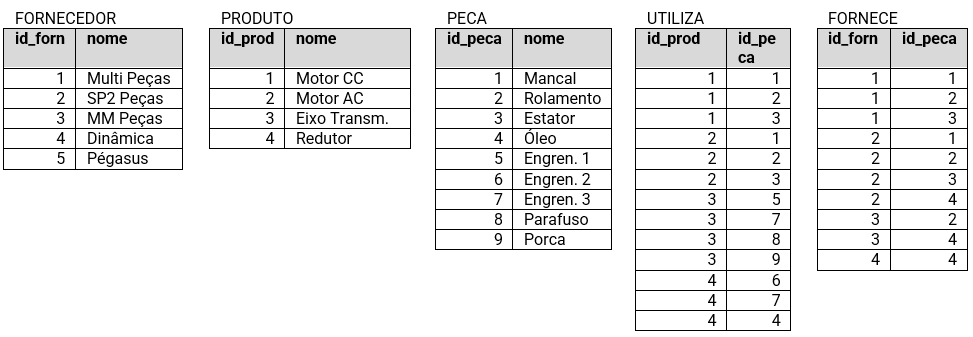
**Consultas Mais Complexas**

### CONSULTAS ENVOLVENDO MAIS DE DUAS TABELAS

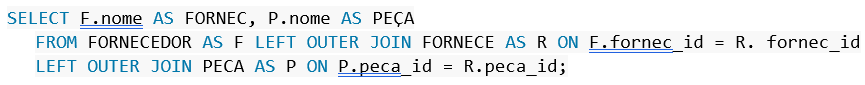
               Consultas envolvendo três ou mais tabelas funcionam da mesma forma que consultas com duas tabelas. O resultado da consulta consiste na seleção de linhas do produto cartesiano de todas as tabelas envolvidas. Para uma consulta envolvendo 3 tabelas com 1.000, 5.000 e 10.000 linhas cada, o produto cartesiano origina uma tabela com 500 **bilhões** de linhas! Certamente os SGBDS não fazem o produto cartesiano antes da seleção das linhas. Eles utilizam algoritmos bastante sofisticados para simultaneamente combinar e selecionar de linhas. De qualquer forma, o desempenho de uma junção depende da quantidade de tabelas envolvidas, do tamanho destas tabelas e das condições impostas nas cláusulas WHERE e ON. Considere o diagrama ER a seguir:

https://paperx-dex-assets.s3.sa-east-1.amazonaws.com/images/1679500856713-ZCw0wfiaSh.png

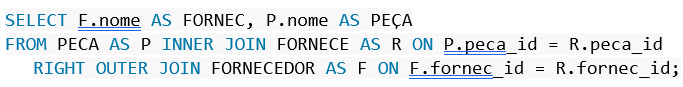
​As entidades FORNECEDOR, PECA e PRODUTO e os relacionamentos FORNECE e UTILIZA são mapeados em 5 tabelas. Os atributos não foram mostrados. Os comandos de criação e inserção de registros estão no material de apoio.



A relação de **todos** os fornecedores e peças que fornecem:



​Observe que foram feitos dois LEFT OUTER JOIN em sequência. Primeiramente, o mais à esquerda é executado e seu resultado e colocado em uma tabela temporária (virtual). Em seguida, o outro é executado entre a tabela temporária e PECA. Pode-se colocar em sequência quantas junções forem necessárias. A execução sempre será da esquerda para a direita e os resultados intermediários são colocados em tabelas temporárias. Na realidade, a consulta acima pode ser feita com um RIGHT OUTER JOIN e um INNER JOIN (por quê? verifique!):

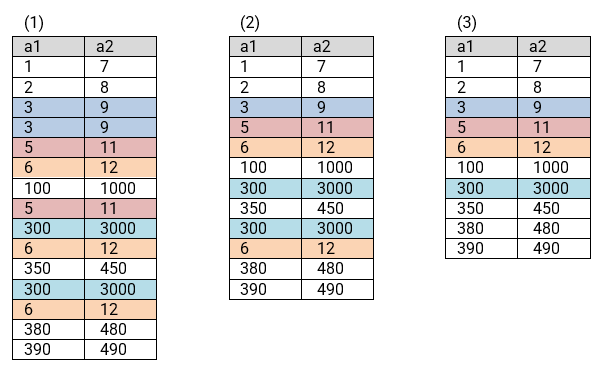
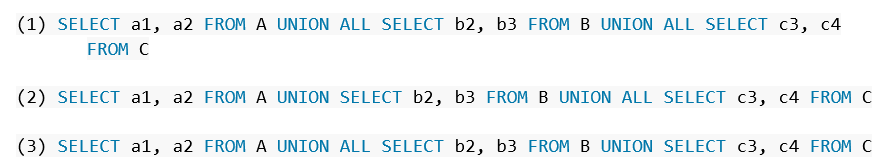
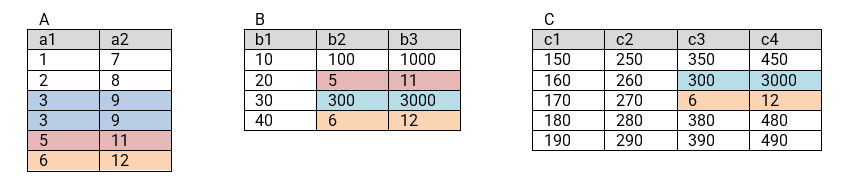


### O OPERADOR UNION

               As operações de junção realizam sempre a seleção de linhas a partir do produto cartesiano das tabelas envolvidas. Estas tabelas possuem, na grande maioria das situações, estruturas distintas. O operador UNION permite que os resultados de duas ou mais consultas (SELECT) sejam unidas em uma só tabela. Algumas restrições: o número de colunas de todas as consultas envolvidas deve ser o mesmo; os tipos das colunas correspondentes em todas as consultas devem ser compatíveis.



​UNION retira os registros duplicados por padrão, sem que seja necessário incluir o modificador DISTINCT. Para forçar a inclusão dos registros duplicados, o qualificador ALL deve ser usado. Um cuidado a ser tomado: o primeiro UNION [DISTINCT] que surgir, faz com que todos os registros duplicados até aquele momento sejam excluídos. Considere as tabelas a seguir:

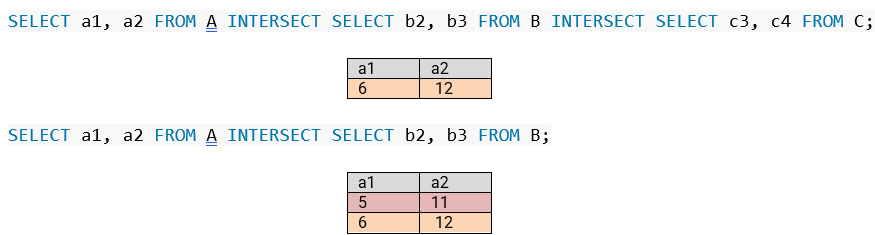


### O OPERADOR INTERSECT

              O operador INTERSECT faz o “oposto” do operador UNION: retorna as linhas que são comuns às consultas envolvidas. Embora alguns dialetos suportem o modificador ALL, o seu comportamento pode variar. Por conta disto, o funcionamento do INTERSECT será apresentado apenas com o modificador DISTINCT, que é o seu padrão.



   Considere as mesmas três tabelas A, B e C mostradas anteriormente.



Quando há mais de um INTERSECT na consulta, eles são executados da esquerda para a direita: A INTERSECT B INTERSECT C é equivalente a (A INERSECT B) INTERSECT C; para alterar a ordem de execução, devem ser usados parênteses. Embora o operador INTERSECT esteja definido na linguagem SQL padrão, ele pode ser substituído por um comando INNER JOIN com as colunas correspondentes sendo comparadas na cláusula ON:

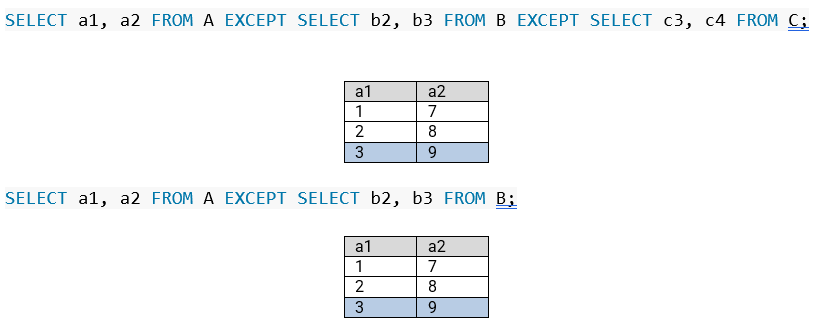
https://paperx-dex-assets.s3.sa-east-1.amazonaws.com/images/1679501649443-7mLRy8vf5E.pnghttps://paperx-dex-assets.s3.sa-east-1.amazonaws.com/images/1679501662869-qed6fF1xnl.png

### O OPERADOR EXCEPT

O operador **EXCEPT** ou **MINUS** seleciona as linhas da primeira tabela que não pertencem à segunda. Os modificadores ALL e DISTINCT têm a mesma função que no operador UNION.



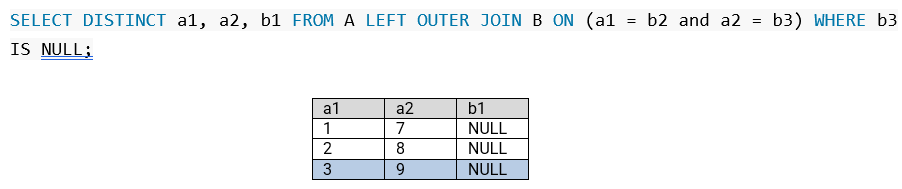
  Considere as mesmas três tabelas A, B e C mostradas anteriormente.



Quando há mais de um EXCEPT na consulta, eles são executados da esquerda para a direita. Para alterar a ordem de execução, devem ser usados parênteses.

De forma similar ao operador INTERSECT, é possível implementar o operador EXCEPT utilizando o comando LEFT OUTER JOIN e um pequeno macete.

Deseja-se selecionar as linhas da primeira tabela (A) que não têm correspondente na segunda (B). Isto é o mesmo que excluir da primeira tabela as linhas que possuem correspondente, isto é, a interseção entre as duas tabelas. Deve-se incluir no SELECT um campo qualquer da segunda tabela. Este campo será NULL apenas nas linhas em que não há correspondência na segunda tabela. Deve-se também incluir o modificador DISTINCT no SELECT (o LEFT OUTER JOIN inclui todas as linhas de A, inclusive as repetidas):



A precedência dos operadores **UNION** e **EXCEPT** é a mesma. Havendo UNION e EXCEPT na mesma consulta, eles são executados da esquerda para a direita. A ordem de execução pode ser alterada usando-se parênteses.

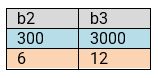
O operador **INTERSECT** tem prioridade sobre **UNION** e **EXCEPT**. Considere a consulta a seguir:

https://paperx-dex-assets.s3.sa-east-1.amazonaws.com/images/1679501965429-lTEYQOaEgX.png

​O operador **INTERSECT** será executado primeiro

https://paperx-dex-assets.s3.sa-east-1.amazonaws.com/images/1679502097765-ELwDaHPGfk.png

​produzindo o resultado abaixo:



​Em seguida, a operação envolvendo **UNION** é executada

https://paperx-dex-assets.s3.sa-east-1.amazonaws.com/images/1679502207629-1S4YaakuJg.png