# Trabalho sobre Álgebra Relacional

## Responda as seguintes questões:

1. Defina o que é Álgebra Relacional.

2. Defina o que é uma Relação.

3. Defina o que é um Atributo.

4. Defina o que é um Domínio.

5. Defina o que é uma Tupla.

6. Defina o que é Grau.

7. Defina o que é Cardinalidade.

8. Defina o que é uma Base de Dados Relacional.

9. Defina o que é a operação unária de seleção.

10. Defina o que é uma operação unária de projeção.

11. Defina o que é a operação entre conjuntos, união.

12. Defina o que é a operação entre conjunto, set difference.

13. Defina o que é a operação entre conjuntos interseção.

14. Defina o que é o produto cartesiano.

15. Define o que é Theta Join.

16. Defina o que EquiJoin.

17. Defina o que é Natural Join.

18. Defina o que é Outer Join.

19. Defina o que é Semi Join.

1. A Álgebra Relacional é uma linguagem de consulta formal, porém procedimental, ou seja, o usuário dá as instruções ao sistema para que ele realize uma seqüência de operações na base de dados para calcular o resultado desejado. A álgebra relacional é uma forma de cálculo sobre conjuntos ou relações. A álgebra relacional recebia pouca atenção até a publicação do modelo relacional de dados de E.F Codd, em 1970. Codd propôs tal álgebra como uma base para linguagens de consulta em banco de dados.
2. Os dados são tratados como tabelas bidimensionais, chamadas relações.
3. O cabeçalho da coluna é chamado de atributo. Cada relação tem um número fixo de colunas, chamados atributos.
4. O tipo de dados que descreve os tipos de valores que podem aparecer em cada coluna é chamado de domínio.
5. Uma linha é chamada de tupla. Cada registro e chamado de tupla.
6. O grau e o número de atributos da relação.
7. E o número máximo e mínimo de ocorrências de uma entidade que estão associadas as ocorrências de outra entidade que participa do relacionamento. Ou seja, a cardinalidade e importante para ajudar a definir o relacionamento, pois ela define o número de ocorrências em um relacionamento.
8. O modelo relacional se destaca pela simplicidade e flexibilidade, utilizando como base os mesmos conceitos dos arquivos de dados. Os dados são armazenados em tabelas que, por sua vez, são compostas de campos e registros. O banco de dados relacional foi criado com base na teoria matemática de conjuntos, e, por isso, seus componentes principais supracitados também podem ser conhecidos por seus correspondentes em linguagem matemática. Os campos podem ser chamados de atributos, os registros são chamados de tuplas, e as tabelas são conhecidas como relação.
9. A operação unária de seleção é uma operação que seleciona tuplas (linhas) de uma relação que satisfazem a uma determinada propriedade.
10. A operação unária de projeção seleciona colunas específicas numa relação, isto é, efetua um corte vertical na relação.
11. A união é uma operação retorna a união das tuplas de duas relações R1 e R2 com eliminação automática de duplicatas. Produz como resultado uma Relação que contém todas as linhas da primeira Relação seguidas de todas as linhas da segunda tabela. A Relação resultante possui a mesma quantidade de colunas que as relações originais, e tem um número de linhas que é no máximo igual à soma das linhas das relações fornecidas como operandos, já que as linhas que são comuns a ambas as relações aparecem uma única vez no resultado. A operação de união é comutativa   
    => R1 U R2 = R2 U R1. Obs: As relações devem possuir o mesmo número de atributos.
12. A diferença retorna as tuplas presentes em R1 e ausentes em R2. É uma operação que requer como operandos duas relações união-compatíveis, ou seja, estruturalmente idênticas. O resultado é uma relação que possui todas as linhas que existem na primeira relação e não existem na segunda. A diferença não é comutativa.
13. A intersecção retorna as tuplas comuns a R1 e R2. Esta é uma operação adicional que produz como resultado uma tabela que contém, sem repetições, todos os elementos que são comuns às duas tabelas fornecidas como operandos. As tabelas devem ser união-compatíveis. Existem operadores de álgebra que são deriváveis de outros. A operação de intersecção é derivável de união e diferença.
14. O produto cartesiano retorna todas as combinações de tuplas de duas R1 e R2. O resultado do produto cartesiano de duas relações é uma terceira relação contendo todas as combinações possíveis entre os elementos das relações originais. Essa relação resultante possuirá um número de colunas que é igual à soma das quantidades de colunas das duas relações iniciais, e um número de linhas igual ao produto do número de suas linhas. Portanto, se fizermos o produto cartesiano de uma relação A que possua 5 colunas e 10 linhas com uma relação B onde existem 3 colunas e 8 linhas, a relação resultante terá 5+3= 8 colunas e 10\*8= 80 linhas.   
    Total de colunas do produto cartesiano: Número colunas da primeira tabela + número de colunas da segunda tabela. Número de linhas do produto cartesiano: Número de linhas da primeira tabela x número de linhas da segunda tabela. Assim, cada linha dessa relação corresponderá à concatenação de uma linha da primeira relação com uma linha da segunda.
15. Theta Join = esta es la combinación general que todos usan porque le permite especificar la condición (la cláusula ON en SQL). Puede unirse en prácticamente cualquier condición que desee, por ejemplo, en Productos que tienen las 2 primeras letras similares o que tienen un precio diferente. En la práctica, esta rara vez es el caso: en el 95% de los casos, se unirá en una condición de igualdad.
16. Equi Join = el más común utilizado en la práctica. El ejemplo anterior es una combinación equi. ¡Las bases de datos están optimizadas para este tipo de uniones! Lo contrario de una combinación equi es una combinación no equi, es decir, cuando se une en una condición distinta de "=". ¡Las bases de datos no están optimizadas para esto! Ambos son subconjuntos de la combinación theta general. La combinación natural también es una combinación theta, pero la condición (theta) está implícita.
17. Unión natural = la combinación (la cláusula ON) se realiza en todas las columnas con el mismo nombre; elimina las columnas duplicadas del resultado, a diferencia de todas las demás combinaciones; la mayoría de los DBMS (sistemas de bases de datos creados por varios proveedores, como SQL Server de Microsoft, MySQL de Oracle, etc.) ni siquiera se molestan en admitir esto, es solo una mala práctica (o decidieron no implementarlo a propósito). Imagine que llega un desarrollador y cambia el nombre de la segunda columna en Producto de Precio a Costo. Luego, todas las uniones naturales se realizarían en PName Y en Cost, lo que daría como resultado 0 filas ya que ningún número coincide.
18. El tipo de JOIN más común del modelo de base de datos relacional es el INNER JOIN de SQL, que, en la práctica, se utiliza cuando se quieren conectar dos tablas de una base de datos a partir de una columna en común. Cada registro de una tabla se fusiona con el registro correspondiente de la otra y quedan ocultos aquellos para los que el Sistema Gestor de Bases de Datos (SGBD) no puede encontrar una coincidencia.  
    Frente a esta sentencia, el OUTER JOIN no solo vuelca los registros de datos que cumplen la condición de selección en las dos tablas (por ejemplo, valores iguales en dos columnas), sino también las tuplas restantes de una u otra tabla.  
    Siguiendo la dirección de lectura de la sintaxis de SQL se habla de una tabla izquierda y una derecha. Las operaciones respectivas se denominan en consecuencia LEFT OUTER JOIN y RIGHT OUTER JOIN. Si, en una consulta, además de los registros que cumplen la condición de selección, también deseas obtener todos los registros de datos de la tabla izquierda y de la derecha, entonces se habla de un FULL OUTER JOIN.  
    Los diagramas de conjuntos permiten explicar el principio que hay tras las diferentes sentencias de JOIN:  
      
    Tipos de OUTER JOIN de SQL: Cada OUTER JOIN se ejecuta como LEFT, RIGHT o FULL OUTER JOIN.
19. Existem alguns mitos envolvendo o que conceitualmente é conhecido como semi join e, infelizmente, tais mitos acabam por prejudicar a escrita de consultas, causando problemas de performance. Mas, o que são semi joins? São joins que retornam linhas de uma tabela A baseado na existência de linhas correlacionadas em uma tabela B. Se a consulta retorna apenas atributos (campos) da tabela da esquerda, então o join é chamado de Left Semi Join; se retorna apenas atributos da tabela da direita, então é chamado de Right Semi Join. Um semi join pode ser produzido usando INNER JOINS, EXISTS, IN e também com o INTERSECT.