

Operador	Álgebra Relacional	SQL Equivalente	Funcionalidade	Parâmetros	Exemplo (Álgebra Relacional)	Exemplo (SQL)
Seleção (σ)	$\sigma_{condição}(R)$	SELECT * FROM R WHERE condição	Filtra tuplas que satisfazem uma condição	condição : condição de filtragem R : relação (tabela)	$\sigma_{idade > 30}(\text{Clientes})$	SELECT * FROM Clientes WHERE idade > 30;
Projeção (π)	$\pi_{atributos}(R)$	SELECT atributos FROM R	Retorna colunas específicas, eliminando duplicatas	atributos : colunas a selecionar R : relação (tabela)	$\pi_{nome, idade}(\text{Clientes})$	SELECT DISTINCT nome, idade FROM Clientes;
Renomeação (ρ)	$\rho_{novo_nome}(R)$	SELECT * FROM R AS novo_nome	Renomeia a relação ou seus atributos	novo_nome : novo nome para a relação R : relação original	$\rho_C(\text{Clientes})$	SELECT * FROM Clientes AS C;
União ($\cup$)	$R \cup S$	SELECT * FROM R UNION SELECT * FROM S	Combina tuplas de duas relações, eliminando duplicatas	R : primeira relação S : segunda relação (mesma estrutura)	$\text{ClientesBrasil} \cup \text{ClientesEUA}$	SELECT * FROM ClientesBrasil UNION SELECT * FROM ClientesEUA;
Interseção ($\cap$)	$R \cap S$	SELECT * FROM R INTERSECT SELECT * FROM S	Retorna tuplas que estão em ambas as relações	R : primeira relação S : segunda relação (mesma estrutura)	$\text{ClientesBrasil} \cap \text{ClientesEUA}$	SELECT * FROM ClientesBrasil INTERSECT SELECT * FROM ClientesEUA;
Diferença ($-$)	$R - S$	SELECT * FROM R EXCEPT SELECT * FROM S	Retorna tuplas de R que não estão em S	R : primeira relação S : segunda relação (mesma estrutura)	$\text{ClientesBrasil} - \text{ClientesEUA}$	SELECT * FROM ClientesBrasil EXCEPT SELECT * FROM ClientesEUA;
Produto Cartesiano ($\times$)	$R \times S$	SELECT * FROM R, S	Combina todas as tuplas de R com todas as de S	R : primeira relação S : segunda relação	$\text{Clientes} \times \text{Pedidos}$	SELECT * FROM Clientes, Pedidos;
Junção Natural ($\bowtie$)	$R \bowtie S$	SELECT * FROM R NATURAL JOIN S	Combina tuplas de R e S com base em atributos comuns	R : primeira relação S : segunda relação	$\text{Clientes} \bowtie \text{Pedidos}$	SELECT * FROM Clientes NATURAL JOIN Pedidos;
Junção Theta ($\bowtie_{\theta}$)	$R \bowtie_{\theta} S$	SELECT * FROM R JOIN S ON θ	Combina tuplas de R e S com uma condição específica	R : primeira relação S : segunda relação θ : condição de junção	$\text{Clientes} \bowtie_{\text{Clientes.cidade = Pedidos.cidade}} \text{Pedidos}$	SELECT * FROM Clientes JOIN Pedidos ON Clientes.cidade = Pedidos.cidade;
Junção Externa Esquerda ($\ltimes$)	$R \ltimes S$	SELECT * FROM R LEFT JOIN S ON condição	Retorna todas as tuplas de R e tuplas de S que têm correspondência	R : primeira relação S : segunda relação condição : condição de junção	$\text{Clientes} \ltimes \text{Pedidos}$	SELECT * FROM Clientes LEFT JOIN Pedidos ON Clientes.id = Pedidos.cliente_id;
Junção Externa Direita ($\rightrightarrows$)	$R \rightrightarrows S$	SELECT * FROM R RIGHT JOIN S ON condição	Retorna todas as tuplas de S e tuplas de R que têm correspondência	R : primeira relação S : segunda relação condição : condição de junção	$\text{Clientes} \rightrightarrows \text{Pedidos}$	SELECT * FROM Clientes RIGHT JOIN Pedidos ON Clientes.id = Pedidos.cliente_id;
Junção Externa Completa ($\ltimes\bowtie$)	$R \ltimes\bowtie S$	SELECT * FROM R FULL JOIN S ON condição	Retorna todas as tuplas de R e S , correspondentes ou não	R : primeira relação S : segunda relação condição : condição de junção	$\text{Clientes} \ltimes\bowtie \text{Pedidos}$	SELECT * FROM Clientes FULL JOIN Pedidos ON Clientes.id = Pedidos.cliente_id;

Operador	Álgebra Relacional	SQL Equivalente	Funcionalidade	Parâmetros	Exemplo (Álgebra Relacional)	Exemplo (SQL)
Divisão ($\div$)	$R \div S$	(Não tem equivalente direto)	Retorna tuplas de R associadas a todas as de S	R : relação com tuplas para dividir S : relação com valores associados	ClientesProdutos $\div$ ProdutosVendidos	SELECT cliente_id FROM Clientes WHERE NOT EXISTS (SELECT 1 FROM Produtos WHERE Produtos.produto_id NOT IN (SELECT produto_id FROM Compras WHERE ...))
Ordenação (τ)	$\tau_{\text{atributos}}(R)$	SELECT * FROM R ORDER BY atributos	Ordena as tuplas da relação com base nos atributos especificados	atributos : colunas a ordenar R : relação (tabela)	$\tau_{\text{idade}} \text{DESC}(\text{Clientes})$	SELECT * FROM Clientes ORDER BY idade DESC;
Agrupamento (γ)	$\gamma_{\text{atributos}; \text{função_agregada}}(R)$	SELECT atributos, função_agregada FROM R GROUP BY atributos	Agrupar tuplas e aplicar funções agregadas	atributos : colunas para agrupar função_agregada : função aplicada aos grupos R : relação (tabela)	$\gamma_{\text{cidade}; \text{COUNT(*)}; \text{AVG(idade)}}(\text{Clientes})$	SELECT cidade, COUNT(*), AVG(idade) FROM Clientes GROUP BY cidade;