Linguagem SQL

Aula 01 – Apresentação e Histórico de Banco de Dados

Gustavo Bianchi Maia
gustavo.maia@faculdadeimpacta.com.br

Sumário

- Apresentação da Disciplina
  - Apresentação: Professor / Aluno
  - Competências, Habilidades, Critérios de Avaliação, Bibliografia, Bases Tecnológicas
- Histórico de Banco de Dados
- O Modelo Relacional
- Histórico da Linguagem SQL
- Álgebra Relacional
- Categorias de Declarações SQL

Apresentação

- Mestrado em Ciência da computação (IME-USP)
- Bacharel em Análise de Sistemas (Unaerp)

- Experiência Profissional
  - Pesquisa científica
    - Projeto Genoma
    - Ministério da saúde (distribuição coquetel AIDS)
  - Especialista em Banco de Dados, com 21 anos de atuação
    - Distribuidoras de bebidas - Ambev
    - Bittime / Grassroots
    - Protege
    - Vagas.com

Competências

- Arquitetar um Bancos de dados capaz de atender às necessidades especificadas.
- Desenvolver rotinas de definição, manipulação e recuperação de dados.
- Garantir a integridade dos dados armazenados utilizando-se de restrições estruturais e funcionais.
- Criar relatórios para análise e consolidação das informações armazenadas.

Habilidades

- Conhecimento das regras de mapeamento dos modelos lógico/conceitual para o físico.
- Conhecimento da sub-linguagem SQL de Definição de dados (DDL): Criação, alteração e remoção de estruturas e regras de armazenamento.
- Conhecimento da sub-linguagem SQL de Manipulação de dados (DML): Inserção, remoção e atualização de dados.
- Conhecimento da sub-linguagem SQL de Pesquisa de dados (DQL): Consulta de dados, Predicados, Funções built-in, Agrupamento e Junções.
- Conhecimento sobre conexões, sessões e transações em banco de dados.

Critérios de Avaliação

Nota Final = 50% MAC + 50% Prova

~~**ou**~~

~~**Nota Final = [40% MAC + 30% Prova + 30% MPAI] + (Bônus PAI Livro)**~~

Se (Nota Final ≥ 6,0 e Frequência ≥ 75%) ⇒ Aprovado

Senão ⇒ Reprovado

Onde:

MAC (Média de Atividades Contínuas): Média das 07 melhores médias de cada AC semanal em um total de 10 ACs.

Prova = Avaliação Semestral

MPAI = Média das provas do PAI para Disciplinas Incidentes do PAI

- O aluno tem direito a uma Prova Substitutiva, com todo o conteúdo do semestre letivo, para substituir a Prova Semestral.

- A Prova Substitutiva somente será utilizada se for maior que a Prova.

Introdução

História de Banco de dados

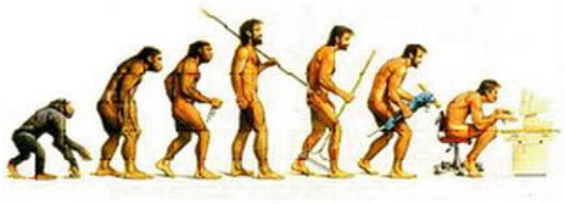
e

Linguagem SQL

Histórico de Banco de Dados

Antigamente as empresas armazenavam dados em fichas de papel que eram organizadas em arquivos físicos através de pastas. Extrair informações e manter esses arquivos organizados era uma tarefa muito custosa.

Além disso o acesso à informação dependia da localização geográfica dos arquivos. Enfim esses arquivos físicos evoluíram para arquivos digitais.

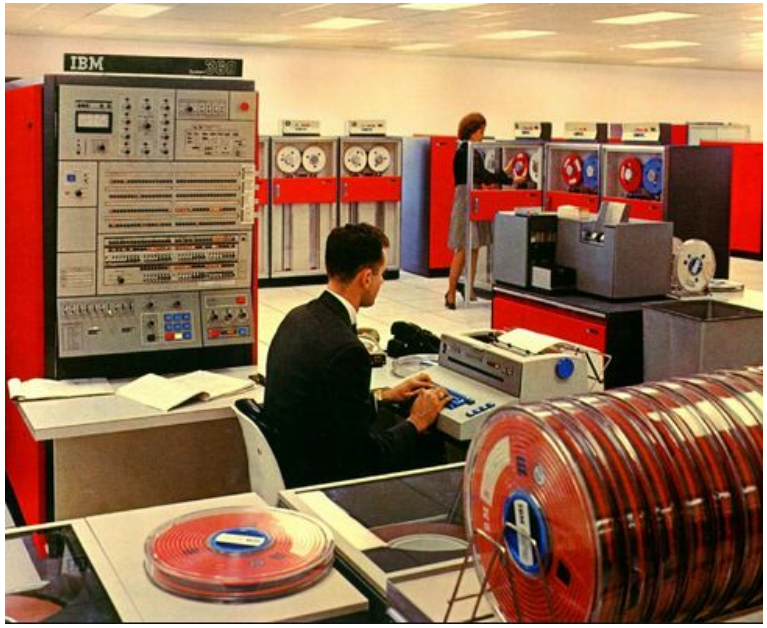


No início cada entidade (clientes, funcionários, produtos, etc.) eram um arquivo de dados acompanhados de um “software simples” para manipular os dados dos arquivos, esses softwares permitiam realizar operações de cadastro, alteração, exclusão e consulta nos arquivos digitais. Apesar de melhorar a tarefa de consulta de informações, os arquivos digitais eram ainda uma versão melhorada dos arquivos físicos.

Histórico de Banco de Dados

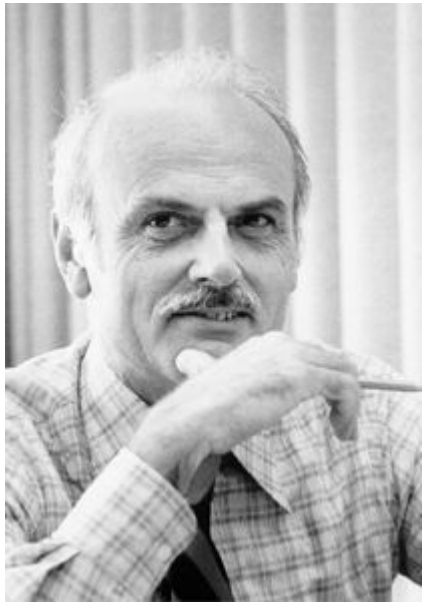
As entidades precisavam relacionar-se, por exemplo um produto é fornecido por um fornecedor, e com os arquivos digitais relacioná-las não era uma tarefa muito trivial, os “softwares simples” para manipular os arquivos digitais começaram a ficar “complexos” para permitir os relacionamentos entre as entidades.

Na década de 60 a empresa IBM investiu fortemente em pesquisas para solucionar estes problemas nos bancos de dados digitais primitivos. Vários modelos de bancos de dados surgiram nesta época, dentre eles os modelos hierárquico e rede (ADABAS, IMS, MUMPS, DBMS/10, ...).



Histórico de Banco de Dados

Em junho de 1970, o pesquisador Edgar Frank “Ted” Codd da IBM, mudou a história dos bancos de dados apresentando o modelo relacional no artigo intitulado “*A Relational Model of Data for Large Shared Data Banks*”, onde o autor apresentou uma forma de usuários sem conhecimento técnico armazenarem e extraírem grandes quantidades de informações de um banco de dados. Esse artigo foi o grande impulso para a evolução dos bancos de dados. A partir do artigo do Dr. Codd os cientistas aprofundaram a ideia de criar o modelo de banco de dados relacional.



Histórico de Banco de Dados

Em 1974 Donald D. Chamberlin e Raymond F. Boyce, também da IBM, publicaram a linguagem de consulta SEQUEL (Structured English Query Language), que seria o nome original da linguagem de consultas, mas que por violações de direitos autorais (o acrônimo já havia sido registrada pela companhia aérea Hawker Siddeley), o nome foi trocado para SQL (Structured Query Language).

Histórico de Banco de Dados

Apesar de ter sido o marco dos bancos de dados relacionais, o artigo de Codd não foi muito explorado no início e somente no final da década de 70 que os primeiros sistemas comerciais foram lançados.

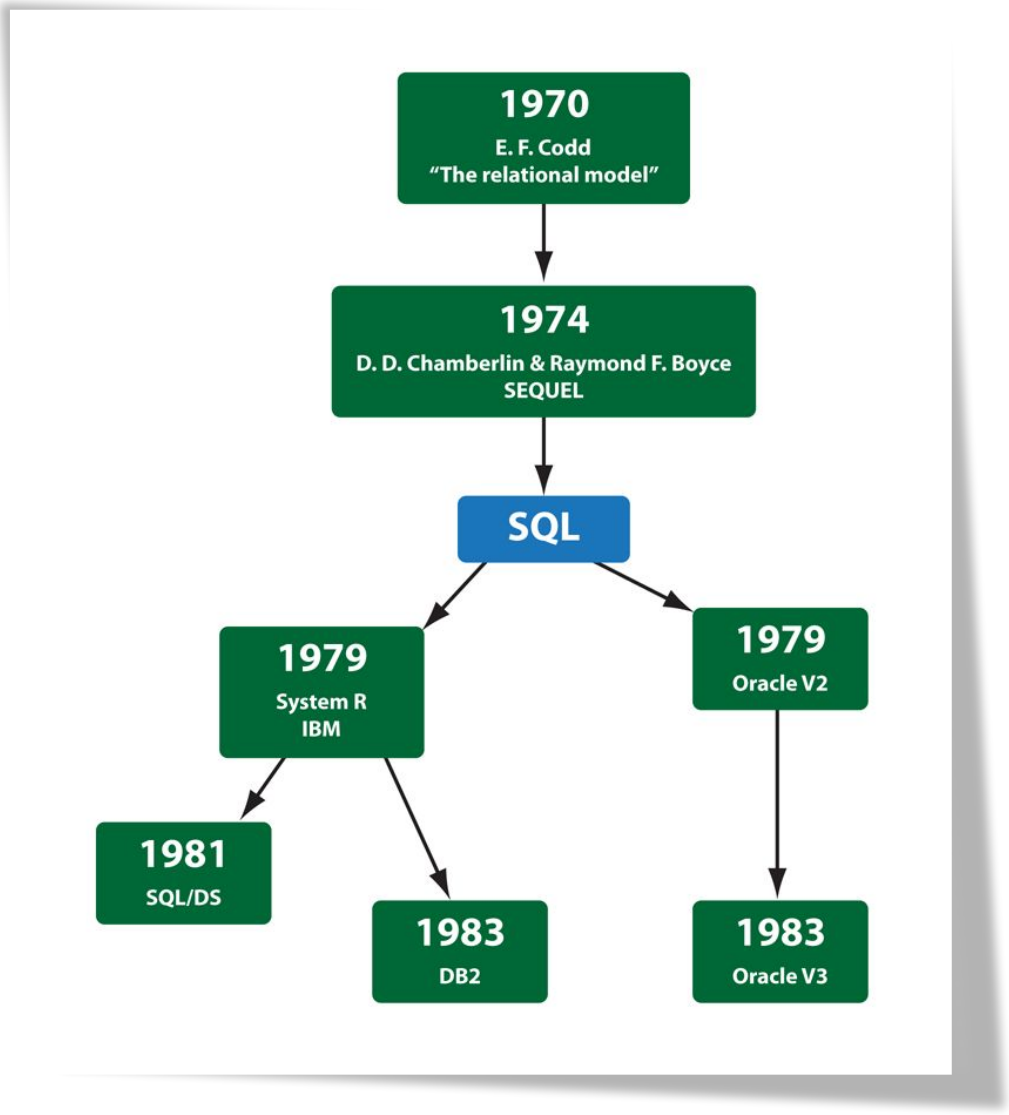
- Em 1979 o primeiro produto SQL, **ORACLE V2**, foi lançado pela Relational Software (posteriormente alterado para Oracle Corporation).

- Algumas semanas depois a IBM lança seu protótipo, **System R**, já utilizando a linguagem SQL (Structured Query Language), que posteriormente se tornaria a linguagem padrão para bancos de dados relacionais. Embora tenha contribuído para a evolução dos bancos de dados relacionais, o “Sistema R” não foi muito bem sucedido comercialmente, mas os sistemas de banco de dados seguintes foram baseados nele.

- Nos anos seguintes vários produtos forma lançados como SQL/DS em 1981 eo DB2 em 1983, ambos da IBM.

- Na sequencia vieram SYBASE, SQL Server, MySQL, DBase III, Paradox, …

Histórico de Banco de Dados



Histórico de Banco de Dados

O SQL é uma linguagem de pesquisa declarativa (cada conjunto de instruções deve permitir a fácil identificação de seu objetivo e/ou resultado) para banco de dados relacional.

Foi desenvolvido na década 70 por pesquisadores dos laboratórios da IBM em San Jose, CA. O objetivo era demonstrar a viabilidade da implementação do modelo relacional proposto por E. F. Codd. Muitas das características originais do SQL foram inspiradas na **álgebra relacional**.

Essa facilidade de identificação e utilização foi percebida pelo mercado e embora o SQL tenha sido originalmente criado pela IBM, rapidamente surgiram vários "dialetos" desenvolvidos por outros produtores, se tornando um “padrão de fato” da indústria de banco de dados.

“Dialetos”: T-SQL (Transact SQL, Microsoft), PL/SQL (Oracle), Procedure Language (IBM), ...

Histórico de Banco de Dados

Essa expansão levou à necessidade de ser criado e adaptado um padrão para a linguagem. Esta tarefa foi realizada em 1986 pelo instituto americano *American National Standards Institute* (ANSI) e em 1987 pelo instituto europeu *International Organization for Standardization* (ISO).

O SQL foi revisto em 1992 e a esta versão foi dado o nome de SQL-92.

Foi revisto novamente em 1999 e 2003 para se tornar SQL:1999 (SQL3) e SQL:2003

O Modelo Relacional

O modelo relacional foi idealizado pelo Dr. Codd, primeiro modelo de dados descrito teoricamente. É um modelo adequado e projetado para um Sistema Gerenciador de Banco de Dados (SGBD) / Relational Database Managment System (RDBMS), que se baseia no princípio em que todos os dados estão armazenados em tabelas (ou, matematicamente falando, relações). Toda sua definição é teórica e baseada na **lógica de predicados** e na **teoria dos conjuntos**.

Subsequentemente foi mantido e aprimorado por Chris Date e Hugh Darwen como um modelo geral de dados. Em 1995 eles mostraram como o modelo relacional pode ser estendido com características de orientação a objeto sem comprometer os seus princípios fundamentais.

O Modelo Relacional

Para melhor entender o modelo relacional e a linguagem SQL, precisamos introduzir alguns conceitos e relacionar com sua origem matemática. Para isso iremos apresentar rapidamente as ideias centrais que envolvem esses elementos.

Álgebra Relacional

A Álgebra Relacional introduz os conceitos para uma linguagem de consulta procedural, onde:

- É a forma teórica de se manipular o banco de dados relacional
- Uma linha é chamada de ***tupla***
- O cabeçalho da coluna é chamado de ***atributo***
- Tabela é chamada de ***relação***
- O tipo de dados que descreve os tipos de valores que podem aparecer em cada coluna é chamado de ***domínio***
- Consiste de um conjunto de operações:
  - ✔ Entrada: uma ou duas relações
  - ✔ Saída: uma nova relação resultado

Álgebra Relacional

Existem as seguintes operações na álgebra relacional:

Operações Fundamentais:

- Seleção
- Projeção
- Produto Cartesiano
- Renomear
- União
- Diferença de Conjuntos

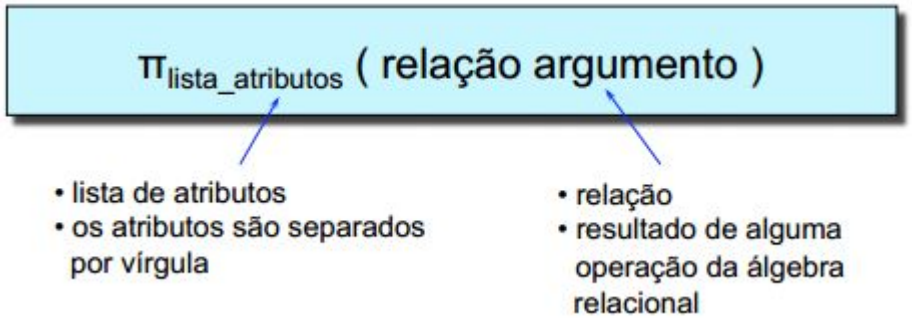
Operações Adicionais:

- Interseção de Conjuntos
- Junção Natural
- Divisão
- Agregação

Álgebra Relacional

Projeção

Geralmente indicada pela letra grega ΠΠ (pi), projeta as colunas solicitadas, isto é, produz um subconjunto vertical



cliente (nro_cli, nome_cli, end_cli, saldo, cod_vend)

nro_cli	nome_cli	end_cli	saldo	cod_vend
1	Márcia	Rua X	100,00	1
2	Cristina	Avenida 1	10,00	1
3	Manoel	Avenida 3	234,00	1
4	Rodrigo	Rua X	137,00	2

Álgebra Relacional

Projeção

Seja a seguinte relação denominada FUNCIONARIO:

Matricula	Nome	Cargo
11	Roberto Silva	DBA Sênior
12	Adriana Mendes	Gerente de Projetos
13	João dos Santos	Desenvolvedor Júnior
14	Helen Almeida	Analista de Sistemas Pleno

Aplicando a projeção, temos:

π_{Nome} (FUNCIONARIO)

Nome
Roberto Silva
Adriana Mendes
João dos Santos
Helen Almeida

Álgebra Relacional

Seleção

Representado pela letra grega σ (sigma), seleciona tuplas que satisfaçam à condição de seleção.

$$\sigma_{\text{condição\_seleção}} (\text{relação argumento})$$

- pode envolver operadores de comparação (=, >, ≥, <, ≤, ≠)
- pode combinar condições usando-se ∧, ∨, ¬

- relação
- resultado de alguma operação da álgebra relacional

Operadores lógicos	Operadores relacionais	
^ (and)	= (Igual a)	
v (or)	< (Menor que)	<= (Menor ou igual a)
¬ (not)	> (Maior que)	>= (Maior ou igual a)

cliente (nro_cli, nome_cli, end_cli, saldo, cod_vend)

nro_cli	nome_cli	end_cli	saldo	cod_vend
1	Márcia	Rua X	100,00	1
2	Cristina	Avenida 1	10,00	1
3	Manoel	Avenida 3	234,00	1
4	Rodrigo	Rua X	137,00	2

Álgebra Relacional

Seleção

É um dos operadores fundamentais da álgebra relacional e tem como resultado um subconjunto estruturalmente idêntico a de um conjunto inicial fornecido como argumento, mas apenas com os elementos do conjunto original que atendem a uma determinada condição (também chamada de predicado).

Se temos a Relação **R**

C1	C2	C3
1	A	10,00
2	A	15,00
3	B	5,00

As operações de Seleção abaixo resultarão em:

σ C2 = "A" (R)

C1	C2	C3
1	A	10,00
2	A	15,00

σ C2 = "A" ^ C3 >= 15,00 (R)

C1	C2	C3
2	A	15,00

Álgebra Relacional

Produto Cartesiano

Combina tuplas de duas relações (relações não precisam ter atributos comuns), gerando como resultante um conjunto de todas as tuplas possíveis entre as relações participantes.

Álgebra Relacional

Produto Cartesiano

cliente (nro_cli, nome_cli, end_cli, saldo, cod_vend)

nro_cli	nome_cli	end_cli	saldo	cod_vend
1	Márcia	Rua X	100,00	1
2	Cristina	Avenida 1	10,00	1
3	Manoel	Avenida 3	234,00	1
4	Rodrigo	Rua X	137,00	2

vendedor (cod_vend, nome_vend)

cod_vend	nome_vend
1	Adriana
2	Roberto

Cliente × Vendedor

nro_cli	nome_cli	end_cli	saldo	cliente. cod_vend	vendedor. cod_vend	nome_vend
1	Márcia	Rua X	100,00	1	1	Adriana
1	Márcia	Rua X	100,00	1	2	Roberto
2	Cristina	Avenida 1	10,00	1	1	Adriana
2	Cristina	Avenida 1	10,00	1	2	Roberto
3	Manoel	Avenida 3	234,00	1	1	Adriana
3	Manoel	Avenida 3	234,00	1	2	Roberto
4	Rodrigo	Rua X	137,00	2	1	Adriana
4	Rodrigo	Rua X	137,00	2	2	Roberto

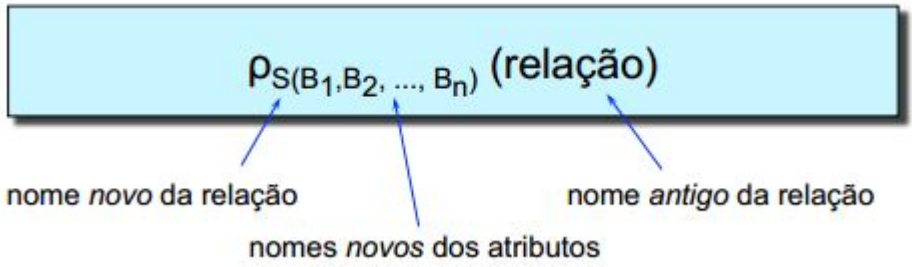
grau: número de atributos de cliente + número de atributos de vendedor

número de tuplas: número de tuplas de cliente * número de tuplas de vendedor

Álgebra Relacional

Renomear

Renomeia o nome da relação, nomes dos atributos da relação ou ambos. Representada pela letra P (rho), é indicada para ser utilizada quando uma relação é usada mais do que uma vez para responder à consulta.



- Exemplos
  - $\rho_{\text{comprador}}$ (cliente)
  - $\rho_{(\text{código, nome, rua, saldo, vendedor})}$ (cliente)
  - $\rho_{\text{comprador (código, nome, rua, saldo, vendedor)}}$ (cliente)

Álgebra Relacional

Linguagem SQL X Álgebra Relacional

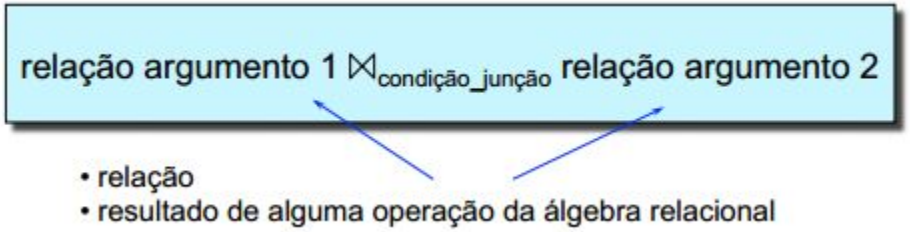
```
SELECT <lista de atributos>
FROM <lista de tabelas>
[WHERE condições de seleção]
```

SQL	Álgebra Relacional
SELECT	projeção
FROM	produto cartesiano
WHERE	seleção

Álgebra Relacional

Junção

Concatena tuplas relacionadas de duas relações. Faz um produto cartesiano das relações, forçando igualdade sobre os atributos que aparecem nas relações.



Álgebra Relacional

Junção - $R \bowtie_{i \theta j} S$

- Passo 1:
  - formar um produto cartesiano das relações

nro_cli	nome_cli	end_cli	saldo	cliente. cod_vend	vendedor. cod_vend	nome_vend
1	Márcia	Rua X	100,00	1	1	Adriana
1	Márcia	Rua X	100,00	1	2	Roberto
2	Cristina	Avenida 1	10,00	1	1	Adriana
2	Cristina	Avenida 1	10,00	1	2	Roberto
3	Manoel	Avenida 3	234,00	1	1	Adriana
3	Manoel	Avenida 3	234,00	1	2	Roberto
4	Rodrigo	Rua X	137,00	2	1	Adriana
4	Rodrigo	Rua X	137,00	2	2	Roberto

- Passo 2:
  - fazer uma seleção forçando igualdade sobre os atributos que aparecem nas relações

nro_cli	nome_cli	end_cli	saldo	cliente. cod_vend	vendedor. cod_vend	nome_vend
1	Márcia	Rua X	100,00	1	1	Adriana
1	Márcia	Rua X	100,00	1	2	Roberto
2	Cristina	Avenida 1	10,00	1	1	Adriana
2	Cristina	Avenida 1	10,00	1	2	Roberto
3	Manoel	Avenida 3	234,00	1	1	Adriana
3	Manoel	Avenida 3	234,00	1	2	Roberto
4	Rodrigo	Rua X	137,00	2	1	Adriana
4	Rodrigo	Rua X	137,00	2	2	Roberto

Cliente $\bowtie$ Vendedor

cod_vend = cod_vend

nro_cli	nome_cli	end_cli	saldo	cliente. cod_vend	vendedor. cod_vend	nome_vend
1	Márcia	Rua X	100,00	1	1	Adriana
2	Cristina	Avenida 1	10,00	1	1	Adriana
3	Manoel	Avenida 3	234,00	1	1	Adriana
4	Rodrigo	Rua X	137,00	2	2	Roberto

Álgebra Relacional

Junção Natural

É uma operação binária que é escrita como ($R \bowtie S$) onde R e S são relações. O resultado da junção natural é uma tabela com todas as combinações das tuplas em R e S que seus atributos em comum, são iguais.

Se temos a Relação **MARCAS**

CodigoMarca	NomeMarca
1	Honda
2	Toyota
3	Nissan

e a Relação **MODELOS**

CodigoModelo	NomeModelo	CodigoMarca
1	Civic	1
1	Hilux	2
1	370Z	3

A Junção Natural abaixo, resulta em:

MODELOS **MARCAS**: $\bowtie$

CodigoModelo	NomeModelo	CodigoMarca	NomeMarca
1	Civic	1	Honda
1	Hilux	2	Toyota
1	370Z	3	Nissan

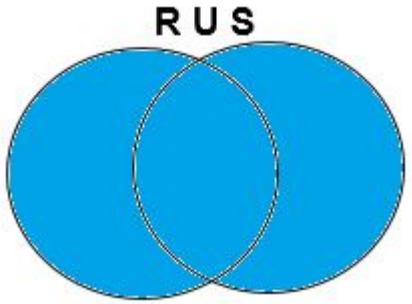
Álgebra Relacional

Operações Sobre Conjuntos

Unem ou eliminam tuplas de duas relações compatíveis (mesmos atributos).

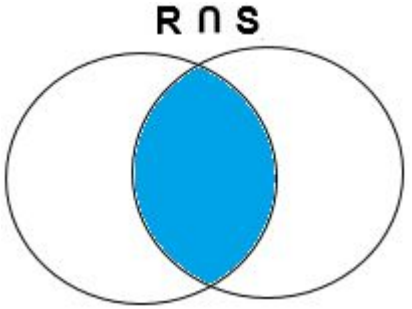
- **União (R U S)**

Gera uma relação que contém todas as tuplas pertencentes a R e S.



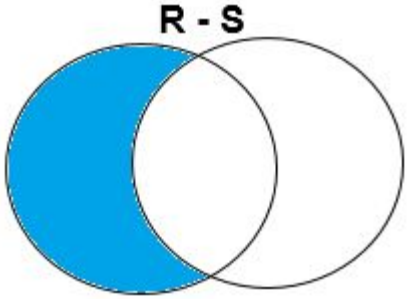
- **Interseção(R ∩ S)**

Gera uma relação que contém todas as tuplas pertencentes tanto a R quanto a S.



- **Diferença (R - S)**

Gera uma relação que contém todas as tuplas pertencentes a R que não pertençam a S.



Álgebra Relacional

Linguagem SQL x Operação Sobre Conjuntos

SQL	Álgebra Relacional
UNION	∪
INTERSECT	∩
EXCEPT	–

Categorias de Declarações SQL

DDL – Data Definition Language

Todas as instruções que criam novas estruturas para armazenamento de dados (CREATE), alteram estas estruturas (ALTER) ou as removem (DROP) fazem parte da DDL. Também fazem parte instruções para criar, atualizar e remover outros objetos (índices, visões, gatilhos, ...).

Exemplos:

- Criar a tabela de Cliente com os atributos (campos), nome (texto de 40 letras), idade (número inteiro) e data de nascimento – dt_nasc (data).

  create table Cliente (nome varchar(40), idade int, dt_nasc datetime)

- Remover a tabela de cliente

  drop table Cliente

Categorias de Declarações SQL

DML – Data Manipulation Language

Todas as instruções que alteram (UPDATE), inserem (INSERT) ou removem (DELETE) dados em uma tabela fazem parte do sub-conjunto de instruções denominadas DML.

Exemplos:

- Inserir o cliente de nome jose, idade 23 e id_cliente 1235
  insert into cliente (nome, idade, id_cliente) values ('jose', 23, 1235)

- Remover da tabela cliente a linha cujo id_cliente seja 1234
  delete from cliente where id_cliente = 1234

- Atualizar a tabela cliente, substitua o nome para 'Paulo' onde o id_cliente seja 1235
  update cliente set nome = 'Paulo' where id_cliente = 1235

Categorias de Declarações SQL

DCL – Data Control Language

Quaisquer instruções que envolvam o controle de acesso aos dados, sejam estas instruções para dar permissão (GRANT), remover (REVOKE) ou negar (DENY), fazem parte da DCL.

Exemplos:

- Garanta ao usuário Marcelo, o direito de selecionar dados na tabela de Cliente
  grant select on Cliente to Marcelo

- Retire do usuário Marcelo as permissões de atualização, inserção e remoção da tabela Cliente
  revoke update, delete, insert on Cliente to Marcelo

Categorias de Declarações SQL

DTL - Data Transactional Language / TCL - Transactional Control Language

Quaisquer comandos que envolvam a declaração de inicio de um bloco transacional (BEGIN TRAN, BEGIN WORK, START TRANSACTION), ou os dois possíveis finais deste um bloco, seja para o sucesso (COMMIT) ou falha (ROLLBACK), fazem parte da DTL.

Exemplos:

- Iniciar explicitamente uma transação com o nome AlterarSaldo
  begin transaction AlterarSaldo

- Abortar uma transação, voltando ao estado original dos dados
  rollback transaction AlterarSaldo

- Confirmar a transação, fazendo com que os dados fiquem alterados
  commit transaction

Categorias de Declarações SQL

DQL - Data Query Language

Qualquer comando que envolva a declaração de recuperação de dados (SELECT).

Exemplo:

Cliente (IDCliente, IDTaxa, Nome, Endereço, Cidade, Estado, CEP, Telefone)

- Retornar as colunas Nome, Telefone da tabela Cliente
  select Nome, Telefone from Cliente

- Recuperar o atributo nome da tabela de clientes onde o id_cliente seja igual a 1234
  select nome from Cliente where idCliente = 1234

Obrigado !

Gustavo Bianchi Maia
gustavo.maia@faculdadeimpacta.com.br