

Linguagem SQL

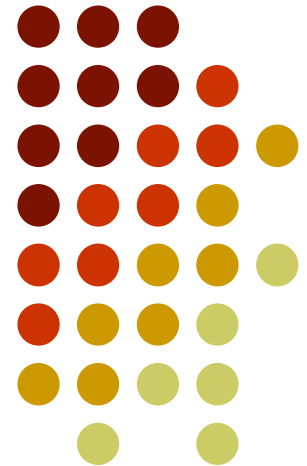
SQL2 ou SQL92



Prof. Jugurta Lisboa Filho

Universidade Federal de Viçosa

Departamento de Informática



Básico da Linguagem SQL



- SQL (*Structured Query Language*)
 - Desenvolvida como a linguagem de consulta do protótipo de SGBD Sistema R (IBM, 1976).
- Adotada como linguagem padrão de SGBD relacional
 - SQL1 - padrão ANSI em 1986.
 - SQL2 - padrão revisado em 1992.
 - SQL3/SQL99 – inclui conceitos de orientação a objetos

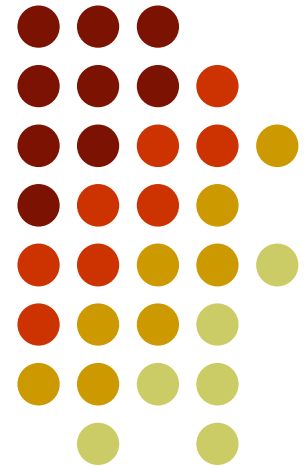
Básico da Linguagem SQL



- Compreende:
 - Linguagem de consulta (QL)
 - Recuperação de dados
 - Linguagem de definição de dados (DDL)
 - Criação e remoção de esquema
 - Definição de visões
 - Linguagem de manipulação de dados (DML)
 - Inserção, remoção e modificação de tuplas no BD
 - Linguagem de manipulação de dados embutida
 - Linguagens hospedeiras como Cobol, Pascal e C
 - Autorização de acesso
 - Restrições de integridade
 - Controle de transações

DDL

Linguagem de Definição de Dados



Definição de Esquemas em SQL



- SQL utiliza os termos *Tabela*, *Linha* e *Coluna* para representar *Relação*, *Tupla* e *Atributo*.
- Componentes de um esquema
 - Tabelas, visões, domínios, restrições de integridade e autorização de acesso.
CREATE SCHEMA GAMBIARRA AUTHORIZATION JUGURTA;
- Catálogo
 - Coleção de esquemas-SQL em um ambiente.
 - Esquemas em um mesmo catálogo podem compartilhar elementos.

Definição de Esquemas em SQL



- Definição de Tabelas
 - Colunas são especificadas como tendo um tipo básico (ou domínio) associado.
 - Podem apresentar restrições e/ou valores *default*.
 - Exemplo:

```
CREATE TABLE APARELHOS
    (NumAp          INTEGER          NOT NULL,
     Tipo           CHAR (15),
     Idade          INTEGER,
     Dono           CHAR (20),
     PRIMARY KEY (NumAp),
     FOREIGN KEY (Tipo) REFERENCES TIPOS );
```

Definição de Esquemas em SQL



- Principais Cláusulas:
 - NOT NULL é requerido para colunas obrigatórias.
 - Atributos da chave primária não podem ser nulos.
 - PRIMARY KEY - especifica um ou mais atributos da chave primária.
 - UNIQUE - especifica chaves alternativas
 - FOREIGN KEY - integridade referencial
- Observação:
 - Colunas são ordenadas pela sua ordem de definição (linhas não tem ordenação)

Definição de Esquemas em SQL



- Definição de domínios

```
CREATE DOMAIN TIPO-NOME AS CHAR(20);
```

```
CREATE DOMAIN TIPO-NUM AS INTEGER;
```

```
CREATE TABLE TÉCNICOS
```

```
    ( NumTec          TIPO-NUM          NOT NULL,  
      Nome            TIPO-NOME,  
      Cargo           TIPO-NOME,  
      PRIMARY KEY (NumTec)  
    );
```


Definição de Esquemas em SQL



- Restrições básicas em SQL
 - Exercícios:
 - Pesquisa sobre as restrições
 - CONSTRAINT
 - CHECK
 - Quais os tipos de domínios de atributos definidos na SQL2?

Definição de Esquemas em SQL



- Eliminando tabelas

DROP TABLE TIPOS CASCADE;

Elimina tabela e mantém integridade referencial.

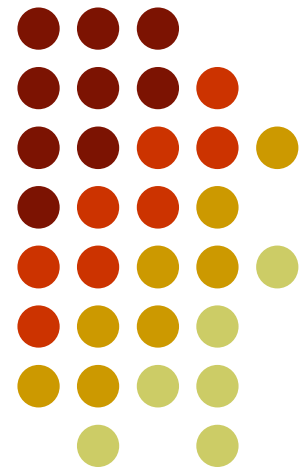
- Alterando tabelas

ALTER TABLE TIPOS ADD Prazo-Médio DATE;

Altera esquema de uma tabela existente.

QL

Linguagem de Consulta a Dados



Consultas em SQL



- Estrutura básica:

```
SELECT      <lista de colunas>
FROM        <lista de tabelas>
[WHERE      <condição>];
```

Uma consulta típica:

```
SELECT      A1, A2, A3, ..., An
FROM        R1, R2, R3, ..., Rn
WHERE       P;
```

é equivalente em álgebra relacional a:

$$\Pi_{A_1, A_2, A_3, \dots, A_n} (\sigma_P (R_1 \times R_2 \times R_3 \times \dots \times R_n))$$

Exemplos de consultas SQL



- Seleção de tuplas (inclui todos os atributos)

- “Obter os aparelhos do tipo Lavadora”.

```
SELECT      *  
FROM        APARELHOS  
WHERE       Tipo = 'Lavadora';
```

- “Obter o nome de todos os técnicos”.

```
SELECT nome FROM TECNICOS
```

Exemplos de consultas SQL



- Seleção de tuplas (inclui todos os atributos)
 - “Obter os aparelhos do tipo Lavadora”.

```
SELECT      *  
FROM        APARELHOS  
WHERE       Tipo = 'Lavadora';
```

- “Obter o nome de todos os técnicos”.

```
SELECT      Nome  
FROM        TÉCNICOS;
```

Consultas em SQL



- Seleção com projeção
 - “Obter o número, tipo e idade de todos os aparelhos cujos donos são Paulo ou Pedro”.

```
SELECT  numAp, tipo, idade FROM APARELHOS  
WHERE  dono = 'Paulo' OR dono = 'Pedro';
```

Consultas em SQL



- Seleção com projeção
 - “Obter o número, tipo e idade de todos os aparelhos cujos donos são Paulo ou Pedro”.

```
SELECT      NumAp, Tipo, Idade
FROM        APARELHOS
WHERE       dono = 'Pedro' OR
            dono = 'Paulo';
```


Consultas em SQL



- Seleção com Junção
 - “Obter os nomes dos técnicos com experiência em secadora”

```
SELECT nomeTecnico FROM EXPERIENCIA, TECNICOS  
WHERE numTec = numTecnico AND tipo = 'secadora';
```

Consultas em SQL



- Seleção com Junção
 - “Obter os nomes dos técnicos com experiência em secadora”

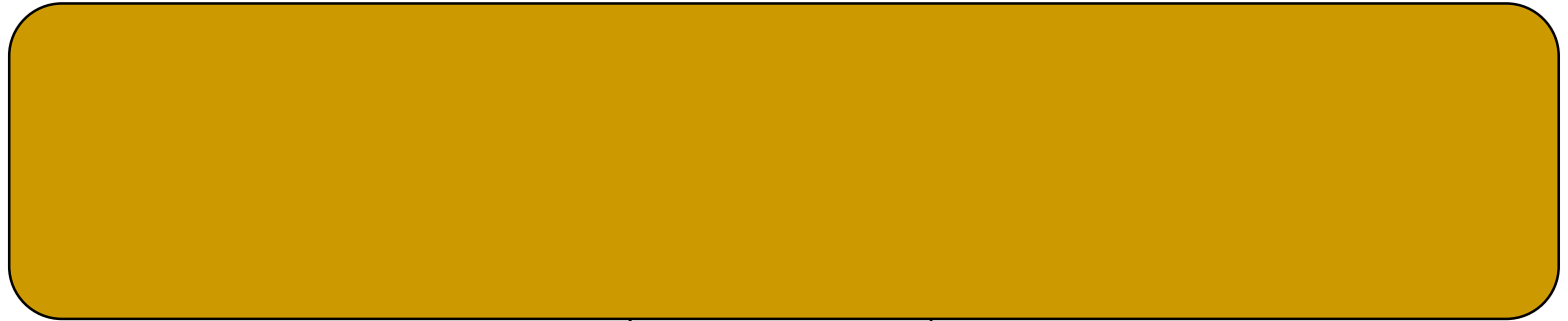
```
SELECT      Nome
FROM        TÉCNICOS, EXPERIÊNCIA
WHERE       NumTec = NumTecnico AND
            Tipo = 'Secadora';
```

Consultas em SQL



- Seleção com Junção

“Obter o número dos aparelhos da categoria 2”



👉 Nomes duplicados: usar qualificação quando o mesmo atributo aparecer em mais de uma tabela.

Consultas em SQL



- Seleção com Junção

“Obter o número dos aparelhos da categoria 2”

```
SELECT          NumAp
FROM            TIPOS, APARELHOS
WHERE           Tipos.Tipo = Aparelhos.Tipo AND
               Categoria = 2;
```

👉 Nomes duplicados: usar qualificação quando o mesmo atributo aparecer em mais de uma tabela.

Consultas em SQL

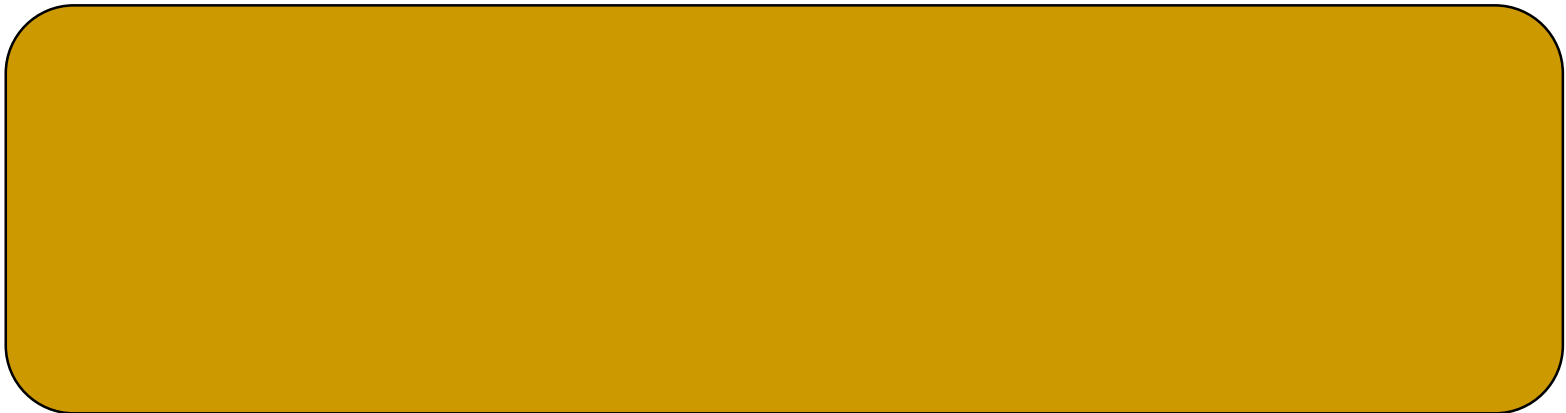


- Seleção com Junção
 - Exercício (sugestão: livro Navathe)
 - Pesquisar sobre as operações:
 - JOIN ... ON
 - NATURAL JOIN
 - INNER JOIN
 - OUTER JOIN (LEFT / RIGHT / FULL)
 - Escreva exemplos de consultas usando o BD Gambiarra

Consultas em SQL



- Uso de Alias (pseudônimo / variável de tupla)
 - Permite associar um “nome de variável” para cada relação, simplificando os comandos SQL.
 - “Liste o nome dos técnicos e sua experiência em aparelhos da categoria 1”



Consultas em SQL



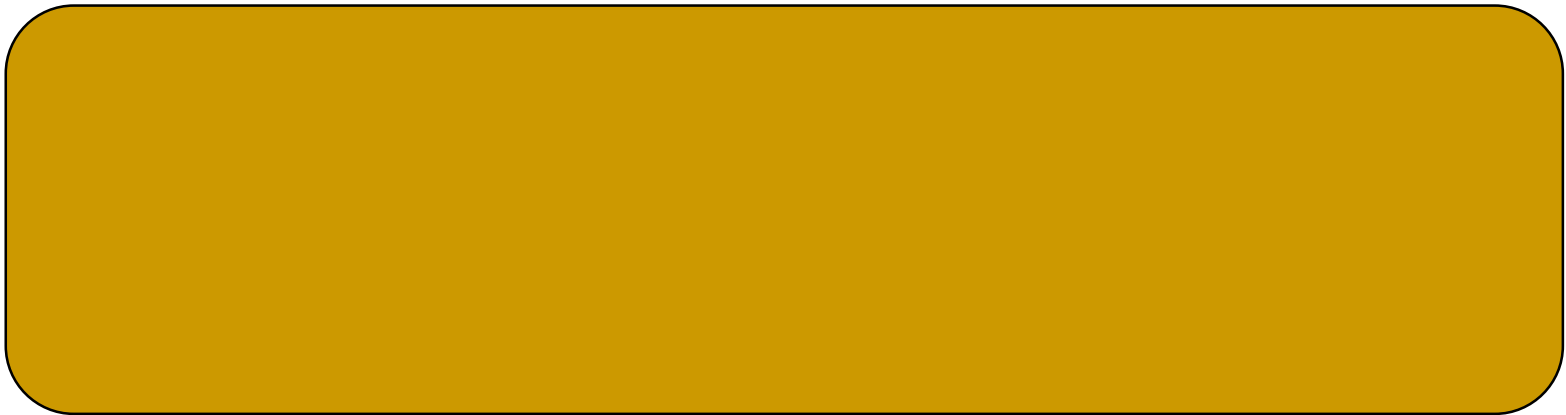
- Uso de Alias (pseudônimo / variável de tupla)
 - Permite associar um “nome de variável” para cada relação, simplificando os comandos SQL.
 - “Liste o nome dos técnicos e sua experiência em aparelhos da categoria 1”

```
SELECT    C.Nome, E.AnosExp, T.Tipo
FROM      TECNICOS C, TIPOS T, EXPERIENCIA E
WHERE     T.Tipo = E.Tipo AND
          E.NumTecnico = T.NumTec AND
          T.Categoria = 1;
```

Consultas em SQL



- Uso de Alias (pseudônimo / variável de tupla)
 - Permite relacionar uma mesma relação 2 vezes.
 - “Liste os donos de aparelhos que possuem mais de um aparelho em conserto”



Consultas em SQL



- Uso de Alias (pseudônimo / variável de tupla)
 - Permite relacionar uma mesma relação 2 vezes.
 - “Liste os donos de aparelhos que possuem mais de um aparelho em conserto”

```
SELECT      A.Dono
FROM        APARELHOS A, APARELHOS B
WHERE       A.Dono = B.Dono AND
            A.NumAp != B.NumAp;
```

Consultas em SQL



- Tabelas como Conjuntos
 - Em geral, SQL não trata a relação como conjunto.
 - Tuplas duplicadas podem aparecer nas relações.
 - Motivos:
 - 1) Eliminar duplicatas é uma operação demorada. Uma solução seria ordenar as tuplas e depois eliminar as duplicatas.
 - 2) O usuário pode querer ver as tuplas duplicadas.
 - 3) Funções agregadas normalmente consideram as tuplas duplicadas (ver funções agregadas).

Consultas em SQL



- Cláusula DISTINCT
 - Força a eliminação de duplicatas.
 - “Obter o nome de todos os donos de aparelhos”.

```
SELECT DISTINCT Dono  
FROM            APARELHOS;
```

Consultas em SQL



- Operações com Conjuntos
 - UNION - INTERSECT - EXCEPT (diferença)
 - Relações resultantes são conjuntos (sem duplicatas).
 - Cláusula ALL permite manter tuplas duplicatas.
 - Ex.: `Relação UNION ALL Relação;`
 - Relações devem ser “compatíveis para união”.

Consultas em SQL



- Exemplos de Operações com Conjuntos
 - “Número dos técnicos que sejam sênior ou que possuam mais de 10 anos de experiência em algum tipo de aparelho”

```
NumTec
(SELECT FROM TECNICOS WHERE CARGO='senior')
UNION
(SELECT numTecnico FROM EXPERIENCIA WHERE
anosExp > 10)
```

Consultas em SQL



- Exemplos de Operações com Conjuntos
 - “Número dos técnicos que sejam sênior ou que possuam mais de 10 anos de experiência em algum tipo de aparelho”

```
(SELECT      NumTec
FROM        TECNICOS
WHERE       Cargo = 'senior')
```

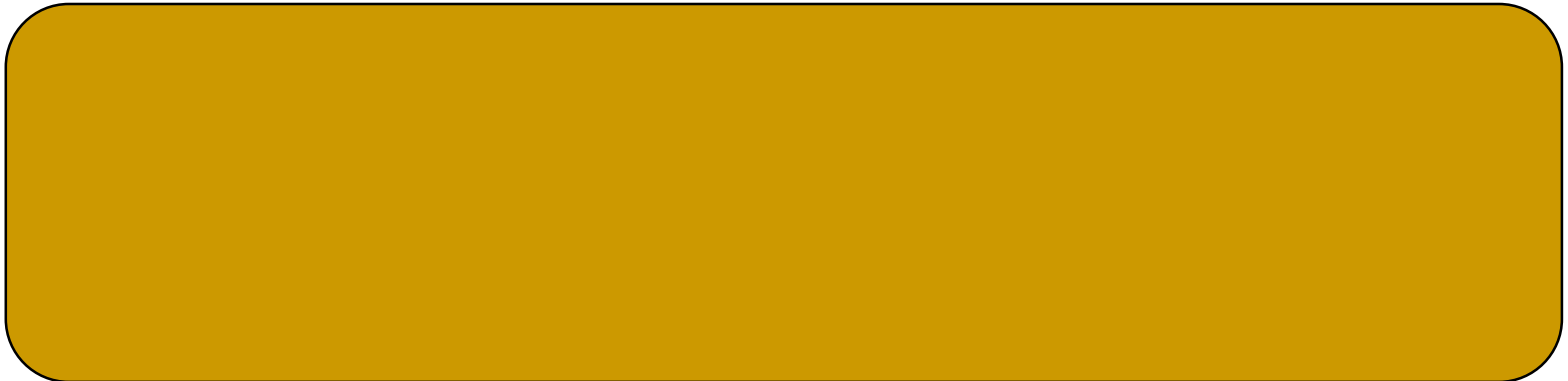
UNION

```
(SELECT      NumTecnico
FROM        EXPERIÊNCIA
WHERE       AnosExp > 10);
```



Consultas em SQL

- Conjuntos Explícitos
 - Conjuntos de valores explícitos na cláusula WHERE
 - “Tipo e categoria de aparelhos com taxa igual a 10, 20 ou 25%”



Consultas em SQL



- Conjuntos Explícitos
 - Conjuntos de valores explícitos na cláusula WHERE
 - “Tipo e categoria de aparelhos com taxa igual a 10, 20 ou 25%”

```
SELECT      A.NumAp, A.tipo, T.categoria
FROM        TIPOS T, APARELHOS A
WHERE       T.tipo = A.tipo AND
            T.taxa IN (10, 20, 25);
```


Consultas em SQL



● Nomes Alternativos de Atributos

- Nomes alternativos podem ser especificados usando o qualificador AS, para atributos ou tabelas
- Estes nomes aparecerão no resultado da consulta

```
SELECT      E.NumTecnico AS NT, T.nome AS N,  
            E.tipo AS T, E.AnosExp AS EX  
FROM        TECNICOS AS T, EXPERIENCIA AS E  
WHERE       T.NumTec = E.NumTecnico;
```

NT		N		T		EX
----	--	---	--	---	--	----

297		Marco		Secadora		15
-----	--	-------	--	----------	--	----

.....

Consultas em SQL



- Consultas Aninhadas
 - “Nome e cargo dos técnicos com experiência acima de 5 anos em aparelhos do tipo Lavadora”

```
SELECT  Nome, cargo
FROM    TECNICOS

WHERE IN (SELECT NumTecnico
           FROM    EXPERIENCIA
           WHERE   anosExp > 5 AND tipo = 'lavadora');
```

Consultas em SQL



- Consultas Aninhadas
 - “Nome e cargo dos técnicos com experiência acima de 5 anos em aparelhos do tipo Lavadora”

```
SELECT    Nome, Cargo
FROM      TECNICOS
WHERE     NumTec IN (SELECT    NumTecnico
                        FROM      EXPERIÊNCIA
                        WHERE     AnosExp > 5 AND
                                tipo='Lavadora');
```

- Operador IN avalia um valor v em um conjunto V
 - Se $v \in V$ o resultado é TRUE (Verdadeiro)

Consultas em SQL



- Consultas Aninhadas
 - Outros operadores de comparação entre elementos e conjuntos.
 - = ANY (= SOME)
 - retorna TRUE se $v =$ algum valor de V (\approx IN)
 - Pode-se usar $>$, \geq , $<$, \leq e \neq ANY
 - $>$ ALL retorna TRUE se $v >$ todos elementos em V

Consultas em SQL



- Funções EXISTS e NOT EXISTS
 - Verifica se o resultado de uma subseleção é vazia.
 - “Liste os técnicos sem experiência”



Consultas em SQL



- Funções EXISTS e NOT EXISTS
 - Verifica se o resultado de uma subseleção é vazia.
 - “Liste os técnicos sem experiência”

```
SELECT          T.Nome
FROM            TECNICOS T
WHERE NOT EXISTS
        (SELECT      *
         FROM          EXPERIÊNCIA E
         WHERE         T.NumTec = E.NumTecnico) ;
```

Consultas em SQL



- Funções EXISTS e NOT EXISTS
 - “Liste os técnicos com nome igual ao nome de algum dono de aparelho”



Consultas em SQL



- Funções EXISTS e NOT EXISTS
 - “Liste os técnicos com nome igual ao nome de algum dono de aparelho”

```
SELECT          T.Nome
FROM            TECNICOS T
WHERE EXISTS
      (SELECT      *
       FROM        APARELHOS A
       WHERE       T.Nome = A.Dono) ;
```




Consultas em SQL

- Consulta com substrings
 - Operador **LIKE** compara padrões de cadeias
 - Caracteres reservados
 - ‘%’ : zero ou mais caracteres
 - ‘_’ : substitui um único caractere

Consultas em SQL



- Comparações entre substrings

- “Nome dos técnicos da família Silva”

```
SELECT      nome
FROM        TECNICOS
WHERE       nome LIKE '%Silva%';
```

- “Número dos aparelhos cujo dono é Paulo ou Paula”

```
SELECT      NumAp
FROM        APARELHOS
WHERE       dono LIKE 'Paul%';
```

Consultas em SQL



- Funções aritméticas em consultas
 - Operadores +,-,*,/ podem ser aplicados com valores ou atributos numéricos

```
SELECT      NumTecnico, AnosExp+1
FROM        EXPERIENCIA
WHERE       tipo = 'Secadora';
```

Consultas em SQL



- Comparações envolvendo **NULL**
 - NULL é usado para representar valores inexistentes
 - Valor desconhecido (mulheres não informam data de nascimento)
 - Valor indisponível (não publique a data do meu nascimento)
 - Valor não aplicável (nunca registre a data de nascimento de uma senhora)



Consultas em SQL

- Comparações envolvendo NULL e os 3 valores lógicos
 - `NULL <> NULL`
 - Quando NULL é comparado, o resultado é UNKNOWN
 - SQL usa 3 valores lógicos TRUE, FALSE e UNKNOWN
 - “Selecione os aparelhos sem informação do dono”

```
SELECT    NumAp
FROM      APARELHOS
WHERE     dono IS NULL;
```

Consultas em SQL



- Conjunto completo de cláusulas no comando **SELECT-FROM**

```
SELECT      <lista de colunas>
FROM        <lista de tabelas>
[WHERE      <condição>]
[GROUP BY   <colunas para agrupamento>]
[HAVING     <condição de agrupamento>]
[ORDER BY   <lista de atributos>]
```

- Apenas as cláusulas **SELECT** e **FROM** são obrigatórias.
- Quando existentes, as cláusulas devem aparecer na ordem especificada acima.

Consultas em SQL



- Funções Agregadas
 - Permitem calcular valores sobre grupos de tuplas, definidas pela cláusula **GROUP BY**.
 - Funções de agregação permitem computar valores:
 - COUNT : contagem de linhas ou valores
 - SUM : soma
 - MAX : máximo
 - MIN : mínimo
 - AVG : média
 - Podem ser aplicadas sobre toda a tabela ou sobre um grupo de linhas.

Exemplos de uso de Funções Agregadas



- Contagem de tuplas
 - “Obter o número de aparelhos em conserto”

```
SELECT          COUNT  (*)  
FROM            APARELHOS;
```


Exemplos de uso de Funções Agregadas



- Contagem por grupos
 - “Obter o número de aparelhos por tipo”

```
SELECT      Tipo, COUNT (NumAp)
FROM        APARELHOS
GROUP BY    Tipo;
```

Exemplos de uso de Funções Agregadas



- Funções Valor Mínimo e Valor Máximo
 - “Obter o menor e o maior valor de anos de experiência dos técnicos”

```
SELECT          MIN (AnosExp) , MAX (AnosExp)  
FROM            EXPERIÊNCIA;
```

Exemplos de uso de Funções Agregadas



- Função Média
 - “Obter a média dos anos de experiência dos técnicos por tipo de aparelho”

```
SELECT      Tipo,  AVG (AnosExp)
FROM        EXPERIÊNCIA
GROUP BY    Tipo;
```

Exemplos de uso de Funções Agregadas



- Ordenando o resultado da consulta - ORDER BY
- “Listar os aparelhos do tipo lavadora ou secadora, ordenados pelo nome do dono”

```
SELECT      *
FROM        APARELHOS
WHERE       Tipo = 'Lavadora' OR
            Tipo = 'Secadora'
ORDER BY    Dono;
```

Exemplos de uso de Funções Agregadas



- Ordenando o resultado da consulta - ORDER BY
 - Ordenação default é ascendente
 - Ordenação pode ser feita por mais de um atributo

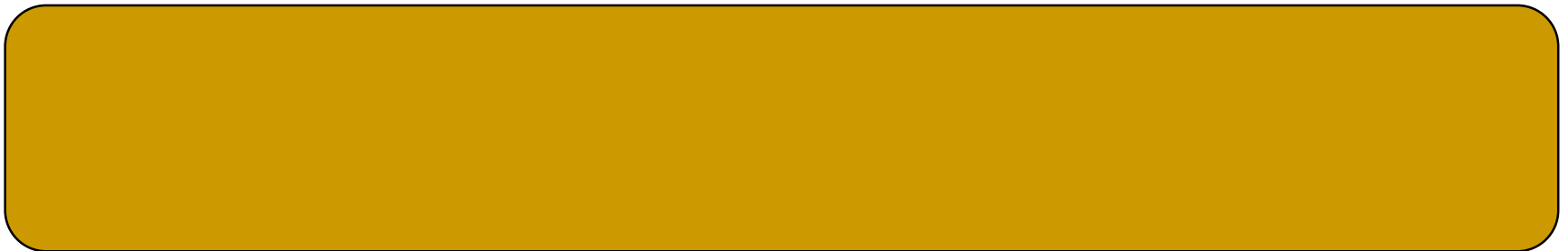
ORDER BY Idade **DESC**, NumAp **ASC**

ORDER BY Dono, Idade, NumAp

Exemplos de uso de Funções Agregadas



- Condição sobre o grupo - HAVING
 - GROUP BY - aplica-se para definir grupos
 - WHERE - condição aplicada às tuplas
 - HAVING - condição aplicada ao grupo
- “Tipo e média de idade de cada tipo de aparelho, cuja média é maior que 5 anos”



Exemplos de uso de Funções Agregadas

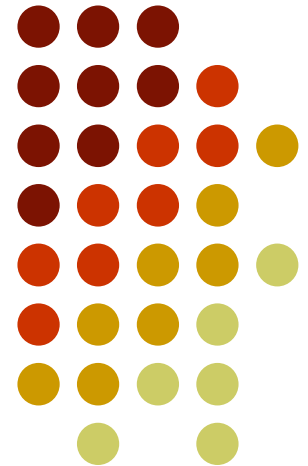


- Condição sobre o grupo - HAVING
 - GROUP BY - aplica-se para definir grupos
 - WHERE - condição aplicada às tuplas
 - HAVING - condição aplicada ao grupo
- “Tipo e média de idade de cada tipo de aparelho, cuja média é maior que 5 anos”

```
SELECT      Tipo, AVG( Idade )
FROM        APARELHOS
GROUP BY    Tipo
HAVING     AVG (Idade) >5;
```

DML

Linguagem de Manipulação de Dados



Manipulação do BD com SQL



- SQL possui três comandos para modificar as tabelas:
 - INSERT (insere tuplas em uma tabela)
 - DELETE (remove tuplas em uma tabela)
 - UPDATE (modifica dados de tuplas de uma tabela)

Manipulação do BD com SQL



- Comando INSERT

- Permite inserir tuplas em uma tabela

```
INSERT INTO TÉCNICOS
```

```
VALUES (425, 'Sérgio', 'estagiário');
```

- Os valores devem ser compatíveis com os domínios dos atributos da tabela.

Manipulação do BD com SQL



- Comando INSERT
 - Permite inserir um conjunto de tuplas, a partir de uma consulta

```
CREATE TABLE          ESTAGIÁRIOS
    ( NumEst            INTEGER      NOT NULL
      NomeEst           CHAR(15) ) ;

INSERT INTO            ESTAGIÁRIOS
SELECT                 NumTec, Nome
FROM                   TÉCNICOS
WHERE                   Cargo = 'estagiário';
```

Manipulação do BD com SQL



- Comando DELETE

- Permite eliminar tuplas de uma tabela

```
DELETE FROM      TÉCNICOS
WHERE            Nome = 'Ronaldo' ;
```

- Eliminar todas as tuplas de uma tabela

```
DELETE FROM      APARELHOS ;
```

- Exclusão não poderá violar as restrições de integridade referencial (chave estrangeira)

Ex.: não é permitido eliminar um tipo de aparelho quando existe um aparelho deste tipo em conserto.

Manipulação do BD com SQL



- Comando DELETE
 - Alguns SGBDs permitem exclusões “em cascata”
 - A exclusão de uma tupla força a exclusão de todas as tuplas que contêm uma chave estrangeira com o valor da tupla eliminada.

Obs: o comando DELETE elimina tuplas e o comando DROP TABLE elimina o esquema da tabela.

Manipulação do BD com SQL



- Comando UPDATE

- Permite alterar um valor em uma única tupla

```
UPDATE      TÉCNICOS
SET         Cargo = 'trainee'
WHERE       NumTec = 718;
```

- Permite alterar valores de atributos em várias tuplas

```
UPDATE      EXPERIÊNCIA
SET         AnosExp = AnosExp + 1;
```

Visões em SQL



- Uma visão é uma tabela derivada de outras tabelas.
 - Outras tabelas englobam tabelas base ou visões.
 - Uma visão não necessariamente existe fisicamente, isto é, ela é considerada uma ***tabela virtual***.
- Visões permitem:
 - aumentar a segurança, quando combinadas com mecanismos de *controle de acesso*
 - aumentar a *independência de dados*

Visões em SQL



- Criando Visões
 - **CREATE VIEW** nome [lista de atributos] **AS** consulta
 - se a lista de atributos não for especificada, as colunas terão os mesmos nomes da tabela base.
 - Criar uma view com número e nome dos técnicos que são seniores.

```
CREATE VIEW      SENIORS  AS  
      SELECT      NumTec, Nome  
      FROM        TÉCNICOS  
      WHERE       Cargo = 'senior';
```


Visões em SQL



- Eliminando Visões do esquema
 - **DROP VIEW** nome
 - elimina uma visão do esquema do BD.

```
CREATE VIEW      TOT-TIPOS  AS (Tipo, Qtde)
      SELECT      Tipo, COUNT (*)
      FROM        APARELHOS
      GROUP BY Tipo;
```

```
DROP VIEW TOT-TIPOS;
```

Visões em SQL



- Exercícios:
 - Pesquise sobre ASSERTIONS
 - Como declarar?
 - Quando se deve usar?
 - Pesquise sobre Atualização de Visões
 - Quais os problemas que podem aparecer?

Exercício SQL



- Dado o seguinte esquema Relacional:

Companhia(c-nome, CNPJ, cidade, ramoAtiv)

Empregado(e-nome, sexo, cidade, chefe*)

chefe referencia Empregado

Trabalha(c-nome*, e-nome*, salário)

c-nome referencia Companhia

e-nome referencia Empregado

Exercício SQL



- **Elabore as seguintes consultas em SQL:**
 - a. Liste os nomes de todos os empregados que trabalham para a “SóTrampo”.
 - b. Nomes e cidades de residência de todos os empregados da “SóTrampo”.
 - c. Nome, sexo e cidade de todos os empregados que trabalham na “SóTrampo” e ganham mais de R\$380,00.
 - d. Encontre todos os empregados que moram e trabalham na mesma cidade.
 - e. Encontre todos os empregados que moram na mesma cidade de seu gerente.
 - f. Liste os empregados que não trabalham na “SóTrampo”.
 - g. Liste o número de empregados de cada Companhia.
 - h. Qual companhia possui o maior número de empregados?
 - i. Altere o BD para informar que o “José da Silva” mudou-se para Viçosa.
 - j. Adicione R\$15,00 ao salário de todos os empregados da “SóTrampo”.
 - k. Adicione R\$30,00 ao salário de todos os gerentes da “SóTrampo”.
 - l. Elimine todos os contratos de trabalho da empresa “Bicos Temporários S.A.”.
 - m. Especifique a visão (EquipeZeSilva) na qual o gerente “José da Silva” só terá acesso aos dados (incluindo salário) de seus subordinados.
 - n. Elimine a visão EquipeZeSilva.

fim



jugurta@ufv.br

