Banco de Dados - SQL

Eduardo Arruda

Eduardo Arruda

- SQL
  - SELECT (projeção de dados)
    - Funções
    - Agrupamento de Dados;

DML - Consultando Dados em Tabelas

LAUREATE BRASIL

- Estrutura Básica

SELECT ➡ PROJEÇÃO

FROM ➡ TABELA OU PRODUTO CARTESIANO DELAS

WHERE ➡ SELEÇÃO

$$\Pi_{Coluna1[,Coluna2[,...]]}(\sigma_{Condição}(Tabela1\ [X\ Tabela2\ [X\ ...\ ]]))$$

SELECT *Coluna1*[,*Coluna2* [, ...]]
FROM *Tabela1*,[*Tabela2* [, ...]]
WHERE *Condição*

DML - Consultando Dados em Tabelas

- Estrutura Genérica

```
SELECT [DISTINCT | ALL] { * | [Tabela.]Coluna1 [AS Alias1]
        [ [Tabela.]Coluna2 [AS Alias2] [, ...]]}
FROM Tabela1 [, Tabela2 [, ... ] ]
[WHERE {Condição Simples | Condição de Sub-consulta} ]
[ORDER BY Coluna1 [ASC | DESC] [,Coluna2 [ASC | DESC] [, ... ]]]
[GROUP BY Coluna1 [,Coluna2 [, ... ]] [HAVING Condição ] ]
[ {UNION | IMTERSECT | EXCEPT} SELECT ... ]
```

DML - Consultando Dados em Tabelas

- **SELECT/FROM** - Projeta os dados da(s) tabela(s), de acordo com os critérios especificados.
  - A projeção do resultado é em uma estrutura tipo tabela
- Basta informar o que se quer, sem se preocupar como fazer isto (SQL Não é procedural).
- Na cláusula SELECT, pode-se utilizar operadores aritméticos e funções de agregações, para projetar cálculos.

Oper. Aritméticos	
+	Adição
-	subtração
*	Multiplicação
/	Divisão

Funções de Agregação	
AVG	Média
MIN	Mínimo
MAX	Máximo
COUNT	Contar
SUM	Somar

DML - Consultando Dados em Tabelas

- **Exemplos**:

```
/* Projetar todas as informações dos autores */
SELECT CodAutor, Nome, Nascimento
FROM AUTOR ;
        OU
SELECT *
FROM AUTOR ;
```

O * projeta todas as colunas de todas as tabelas especificadas na cláusula **FROM**

```
/* Projetar a média dos valores dos livros */
SELECT AVG (Valor)
FROM LIVRO;

/*Projetar todos os livros(títulos) e seus valores com 10% de desconto*/
SELECT Titulo, Valor - (Valor * 0.1)
FROM LIVRO;

/* Projetar a quantidade de autores cadastrados */
SELECT COUNT (*) AS QUANTIDADE, 'ud' AS UNIDADE
FROM AUTOR;
```

DML - Consultando Dados em Tabelas

- **GROUP BY** - agrupa os resultados por valores idênticos.
  - Utiliza-da com as funções de agregação, mas pode ser usada sem estas.
  - OBS: Os campos do GROUP BY devem aparecer no SELECT!

- **Exemplos**:

```
/* Projetar a média dos valores dos livros por editora */
SELECT EDITORA.Razao, AVG (LIVRO.Valor)
FROM LIVRO, EDITORA
WHERE LIVRO.CodEditora=EDITORA.CodEditora
GROUP BY EDITORA.Razao ;

/* Projetar todas as editoras (razão) que publicaram livros */
SELECT EDITORA.Razao
FROM LIVRO, EDITORA
WHERE LIVRO.CodEditora=EDITORA.CodEditora
GROUP BY Editora.Razao;
```

DML - Consultando Dados em Tabelas

- **HAVING** - Utilizada para filtrar o resultado dos grupos
  - Só é atendida depois do Agrupamento !
  - Só existe se associada à cláusula **GROUP BY** (mas o oposto não)
  - Vem depois do GROUP BY e antes do ORDER BY
  - A condição só pode envolver os campos/funções do SELECT
- **Exemplos**:

```
/* Projetar as editoras (código), cujo o preço médio dos livros é > 60*/
SELECT CodEditora, AGV(Valor) AS MediaValor
FROM LIVRO
GROUP BY CodEditora
HAVING MediaValor > 60;

/* Projetar os autores (nome) que publicaram mais de 3 livros*/
SELECT AUTOR.Nome, COUNT(DISTINCT LIVRO.Titulo)
FROM LIVRO, AUTOR
WHERE AUTOR.CodAutor = LIVRO.CodAutor
GROUP BY AUTOR.Nome
HAVING COUNT(DISTINCT LIVRO.Titulo) > 3;
```

Obrigado!

Eduardo Arruda