

Introdução à Banco de Dados

Consultas SQL

Prof. Ulpio Netto.

Baixem o BD da aula de Hoje

- Link: bit.ly/oxetechBD
- Eu criei um banco de dados com 45 linhas de inserção, o banco de dados irá armazenar os dados de alunos, que possuem número de matrícula, nome, idade e data da nascimento.
- Um trecho desse banco de dados:

```
1023|Rafael Lima|19|2005-02-08  
1024|Vanessa Almeida|23|2001-09-02  
1025|Anderson Silva|20|2004-07-14  
1026|Laura Oliveira|22|2002-11-28  
1027|Lucas Costa|21|2003-06-03  
1028|Patrícia Rocha|19|2005-03-17  
1029|Gustavo Pereira|23|2001-12-09  
1030|Camila Alves|20|2004-09-21
```



Consultas

- Uma consulta é uma instrução usada para interagir com bancos de dados. O SQL permite que os usuários realizem diversas operações em um banco de dados. As consultas são essenciais para manipulação eficiente
- Como vimos antes, podemos consultar informações de tabelas através do comando **SELECT**.
- Mas além disso, é possível fazer mais coisas do que só ver todos as inserções feitas no banco de dados.



Estrutura de uma consulta

- A estrutura de uma consulta segue a regra:

```
sqlite> SELECT COLUNAS  
...> FROM TABELA  
...> WHERE CONDIÇÃO  
...> ;
```

- Então pode-se ler que: Selecione as colunas da tabela tal usando a condição que foi passada.
- Caso queira consultar todas as colunas, pode-se usar o * para representar todas as colunas.



Filtrando Consultas

- Usar o **Where** é opcional, usado para filtrar os resultados com base em uma condição

```
sqlite> select * from aluno where idade <20;  
1004|Ana Pereira|19|2005-01-08  
1009|Roberto Lima|19|2005-03-05  
1014|Amanda Costa|19|2005-07-17  
1019|Pedro Rocha|19|2005-04-05  
1023|Rafael Lima|19|2005-02-08  
1028|Patrícia Rocha|19|2005-03-17  
1033|José Souza|19|2005-01-01  
1038|Luana Rocha|19|2005-05-29  
1043|Marcelo Lima|19|2005-06-13
```



Operadores do Where - Relacional

- O Where realiza uma comparação da condição que foi utilizada.

Operador	Significado	Exemplo
=	Igual	5 = 5
>	Maior que	5>3
<	Menor que	3<5
>=	Maior ou Igual	5>=2
<=	Menor ou igual	10<= 10
<>	Diferente	1 <> 2



Operadores do Where - Lógico

- Geralmente usados para relacionar duas comparações Relacionais entre si.

Operador	Significado	Exemplo
And	E	True and True
Or	Ou	True or False
Not	Não	Not False



Order BY

- O Termo **ORDER BY** é usado para ordenar os resultados em ordem crescente ou decrescente.

```
select idade from aluno order by idade desc;
```



Funções de Agregação

- As funções de agregação são utilizadas em SQL para realizar cálculos em conjuntos de dados. Essas funções operam em várias linhas de uma tabela e retornam um único valor resultante



Funções de Agregação - Count

- A função **COUNT** conta o número de linhas em um conjunto de resultados,. Pode ser usada com * para contar todas as linhas ou com uma coluna especifica para contar as linhas não nulas dessa coluna.

```
sqlite> select count(*) from aluno;  
45
```



Funções de Agregação - SUM

- A função **SUM** calcula a soma dos valores em uma coluna

```
sqlite> select sum(idade) from aluno;  
945
```

```
sqlite> select sum(nome) from aluno;  
0.0
```



Funções de Agregação - AVG

- A função **AVG** ou average calcula a média dos valores em uma coluna.

```
sqlite> select avg(idade) from aluno;  
21.0
```



Funções de Agregação - MAX

- A função **MAX** retorna o valor máximo em uma coluna

```
sqlite> select max(idade) from aluno;  
23
```



Funções de Agregação - MAX

- A função **MAX** retorna o valor máximo em uma coluna

```
sqlite> select max(idade) from aluno;  
23
```

```
sqlite> select matricula,nome,max(idade) from aluno;  
1005|Paulo Rocha|23
```



Funções de Agregação - min

- A função **MIN** retorna o valor mínimo em uma coluna

```
sqlite> select min(idade) from aluno  
19
```

```
sqlite> select matricula,nome,min(idade) from aluno;  
1004|Ana Pereira|19
```



Group BY

- O termo **GROUP BY** é utilizada em conjunto com funções de agregação para agrupar resultados com base em uma ou mais colunas.
- Exemplo:

```
sqlite> select nome,count(*)  
...> from aluno  
...> group by nome;
```



Usando o AS

- O Termo **AS** é usado para “Renomear” uma tabela em uma consulta.
- Como o SQLite não mostra o nome das colunas durante uma consulta, não será muito usado para isso, porém, podemos armazenar informações usando **as** como se fosse uma variável.

Exemplo:

```
sqlite> select nome,count(*) as quantidade
...> from aluno
...> group by nome
...> having quantidade >1;
Gustavo Pereira|2
Patrícia Rocha|2
```



Explicando o exemplo:

```
sqlite> select nome,count(*) as quantidade  
...> from aluno  
...> group by nome  
...> having quantidade >1;  
Gustavo Pereira|2  
Patrícia Rocha|2
```

- **Count(*)** conta o numero de registros com o mesmo nome
- **as quantidade** atribui um **alias** à coluna resultante para facilitar a leitura
- **Group by** agrupa os registros pelo campo “Nome” permitindo que a função **COUNT** seja aplicada a cada grupo de registros com o mesmo nome.
- **Having quantidade** filtra apenas os grupos que tem mais de uma ocorrência



Exercícios:

- Usando o mesmo DB que você baixou:
 - Recupere todos os alunos da tabela.
 - Recupere apenas o nome e a idade de todos os alunos.
 - Recupere os alunos que têm 20 anos de idade.
 - Recupere os alunos cuja data de nascimento é anterior a '2003-01-01'.
 - Recupere os alunos ordenados por idade, do mais novo ao mais velho.
 - Recupere os alunos que tem 21 anos ou mais.
 - Recupere os alunos que têm 20 anos ou menos e cujos nomes começam com 'A'.
 - Recupere o total de alunos na tabela.
 - Recupere a idade média dos alunos.
 - Recupere os alunos com idade superior à média





SECTI

 **OXETECH**