Banco de Dados

Robot Eva pelo Mundo SQL

01

FUNDAMENTOS SQL

Desvendando Bancos de Dados: SQL e NoSQL

Principais Comandos SQL

Bancos de dados são a espinha dorsal de praticamente todas as aplicações modernas, armazenando e organizando dados de forma eficiente. Do SQL (Structured Query Language) ao NoSQL, há uma variedade de opções disponíveis para atender às diversas necessidades de armazenamento e manipulação de dados. Neste ebook, exploraremos apenas os fundamentos e comandos do banco de dados, SQL, capacitando você a conhecer informações sobre esta tecnologia.

02

BANCO DE DADOS SQL

SQL (Structured Query Language)

SQL (Structured Query Language) é uma linguagem de programação padrão utilizada para gerenciar e manipular bancos de dados relacionais. Ela foi desenvolvida inicialmente pela IBM nos anos 1970 e desde então tornou-se a linguagem dominante para interação com sistemas de gerenciamento de banco de dados relacionais (RDBMS), como MySQL, PostgreSQL.

Bancos de Dados Relacionais: SQL é um banco de dados relacional. Principais sistemas de gerenciamento de banco de dados relacionais (RDBMS), como MySQL, PostgreSQL e SQL Server.

Transações e controle de concorrência: garantindo a integridade dos dados em ambientes multiusuário.

Linguagem de Definição de Dados (DDL) e Linguagem de Manipulação de Dados (DML).

Consultas avançadas com JOINs, subconsultas e funções agregadas.

03

COMANDOS SQL

Comandos SQL

Manipulação de dados com comandos SQL básicos: SELECT, INSERT, UPDATE, DELETE.
Chaves primárias, estrangeiras e índices: otimizando a eficiência do banco de dados.
Modelagem de dados: normalização e denormalização.
Principais comandos SQL fundamentais para interagir com bancos de dados relacionais.

SELECT: Usado para selecionar dados de uma tabela. É o comando mais comum em SQL.

```
SELECT coluna1, coluna2 FROM tabela WHERE condição;
```

INSERT: Insere novos registros em uma tabela.

```
INSERT INTO tabela (coluna1, coluna2) VALUES (valor1, valor2);
```

UPDATE: Atualiza registros existentes em uma tabela.

UPDATE tabela SET coluna1 = valor1, coluna2 = valor2 WHERE condição;

DELETE: Remove registros de uma tabela.

DELETE FROM tabela WHERE condição;

CREATE TABLE: Cria uma nova tabela no banco de dados.

```
CREATE TABLE tabela (
    coluna1 tipo_de_dado,
    coluna2 tipo_de_dado,
    ...
);
```

ALTER TABLE: Modifica a estrutura de uma tabela existente.

ALTER TABLE tabela ADD coluna tipo_de_dado;

DROP TABLE: Remove uma tabela do banco de dados.

DROP TABLE tabela;

JOIN: Combina registros de duas ou mais tabelas com base em uma condição relacionada.

SELECT colunas FROM tabela1 JOIN tabela2 ON tabela1.chave = tabela2.chave;

GROUP BY: Agrupa registros com base em valores semelhantes em uma ou mais colunas sql.

SELECT coluna, FUNÇÃO_AGGREGATE(coluna) FROM tabela GROUP BY coluna;

ORDER BY: Ordena os resultados em ordem ascendente ou descendente.

SELECT coluna FROM tabela ORDER BY coluna ASC|DESC;

HAVING: Filtra grupos de resultados de acordo com uma condição especificada.

SELECT coluna, FUNÇÃO_AGGREGATE(coluna) FROM tabela GROUP BY coluna HAVING condição;

COUNT, SUM, AVG, MAX, MIN: São funções de agregação usadas com GROUP BY para realizar cálculos em conjuntos de dados.

COUNT: Conta o número de linhas em um conjunto de resultados.

SELECT COUNT(*) FROM tabela;

SUM: Calcula a soma de valores em uma coluna.

SELECT SUM(coluna) FROM tabela;

AVG: Calcula a média de valores em uma coluna.

SELECT AVG(coluna) FROM tabela;

MAX: Retorna o valor máximo em uma coluna.

SELECT MAX(coluna) FROM tabela;

MIN: Retorna o valor mínimo em uma coluna.

SELECT MIN(coluna) FROM tabela;

Você pode usar esses comandos em conjunto com outros comandos SQL, como SELECT e GROUP BY, para realizar operações mais complexas em seus dados. Por exemplo:

Contar o número de produtos em cada categoria:

CSELECT categoria, COUNT(*) FROM produtos GROUP BY categoria;

Calcular o total de vendas por mês:

SELECT MONTH(data_venda), SUM(valor_venda) FROM vendas GROUP BY MONTH(data_venda);

Encontrar a idade média dos funcionários:

SELECT AVG(idade) FROM funcionarios;

Encontrar a data mais recente em que um produto foi atualizado:

SELECT MAX(data_atualizacao) FROM produtos;

Encontrar o produto mais barato em estoque:

SELECT MIN(preco) FROM produtos WHERE quantidade_estoque > 0;

Esses são exemplos simples de como usar esses comandos em consultas SQL para obter informações úteis de um banco de dados.