


Banco de Dados

Fabiano Baldo

Fabiano Baldo 1

1




Banco de Dados

- Banco de Dados;
- Sistema de Gerenciamento de Banco de Dados;
- Banco de Dados Relacional;

Fabiano Baldo 2

2




Banco de Dados

- "Banco de dados é uma coleção de unidades de dados físicos que são relacionados entre si de uma maneira específica" (GUIDE-SHARE, 1970);
- "Uma banco de dados consiste de todas as ocorrências de registros controlados por um esquema específico" (CODASYL, 1971);
- "Um banco de dados é uma coleção de dados, organizados e integrados, que constituem uma representação natural de dados, e que possa ser utilizada por todas as aplicações relevantes sem duplicação de dados" (PALMER, 1975).

Fabiano Baldo 3

3



Sistema de Gerenciamento de Banco de Dados (SGBD)

- Tem como objetivo prover um ambiente que seja adequado e eficiente para recuperar, armazenar e manter as informações pertencentes a um banco de dados;
- Justificativa: Necessidade de haver a separação entre os problemas específicos de aplicações e problemas de armazenamento de dados, ou seja, independência entre dados e programas;
- São projetados para gerenciar grandes grupos de informações;
- As primeiras implementações de SGBDs datam do final da década de 60.

Fabiano Baldo 4

4

Banco de Dados Relacional

- Um banco de dados relacional consiste em uma coleção de tabelas, cada qual designada por um nome único;
- Uma linha numa tabela representa um relacionamento entre um conjunto de valores;
- Uma vez que uma tabela é uma coleção de relacionamentos, existe uma correspondência íntima entre o conceito de tabela e o conceito matemático de relação;
- É a partir desta correlação que o modelo de dados relacional tira seu nome.

Fabiano Baldo

5

5

Exemplo

Nome da relação		Atributos					
ALUNO	Nome	SSN	FoneResidencia	Endereco	FoneEscritorio	Idade	MPG
Tuplas	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21
	Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	null	18	2.89
	Dick Davidson	422-11-2320	null	3452 Elgin Road	749-1253	25	3.53
	Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
	Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	null	19	3.25

Fabiano Baldo

6

6

Chave Primária e Estrangeira

- As noções de superchave, chave candidata, chave primária também são aplicáveis ao modelo relacional;
- Podemos estender este conceito incluindo a noção de chave estrangeira que serve como forma de transformar um relacionamento fraco entre entidades em um relacionamento forte.

Fabiano Baldo

7

7

Tabela Agência

número-agência	nome-agência
10	Downtown
13	Mianus
15	Redwood
21	Pownal

Chave Primária e Estrangeira

■ Chave Primária
■ Chave Estrangeira




Tabela Empréstimo


número-agência	número-empréstimo	número-cliente	quantia
10	17	Jones	1000
13	29	Curry	500
21	29	Williams	1200
15	23	Smith	2000

Chave Estrangeira + Chave Candidata = Chave Primária

Fabiano Baldo

8

8




SQL

- SQL (Structured Query Language);
- Usa uma combinação da álgebra relacional e construções de cálculo relacional;
- Foi desenvolvida pela IBM no início dos anos 70 e mais tarde se tornou um padrão ANSI;
- Se estabeleceu como a linguagem padrão para banco de dados relacional;
- Embora seja chamada de "linguagem de consulta" ela contém outras capacidades além de consultas a banco de dados;
- Inclui recursos para definição de estruturas, modificação e restrições de dados.

Fabiano Baldo 9

9




Comandos de Definição

- São utilizados para descrever / definir o esquema do banco de dados;
- Estão definidos na **Data Definition Language (DDL)**;
- Contém comandos para criar, modificar e excluir definições do esquema do banco de dados;
- Também define restrições de integridade, direitos de acesso e privilégios para tabelas e visões.

Fabiano Baldo 10

10



Comando de Criação

- Especificado pela palavra reservada **CREATE**.
- Usada para:
 - CREATE DATABASE
 - CREATE TABLE
 - CREATE VIEW
 - CREATE INDEX
 - CREATE PROCEDURE

```
CREATE TABLE deposito (
    numero_agencia integer not null,
    numero_conta integer not null,
    nome_cliente string,
    PRIMARY KEY (numero_agencia, numero_conta));
```

11

11



Comando de Alteração

- Especificado pela palavra reservada **ALTER**.
- Usada para:
 - ALTER TABLE
 - ALTER VIEW
 - ALTER PROCEDURE
 - ALTER TRIGGER


```
ALTER TABLE deposito
    ADD saldo float,
    DROP nome_cliente;
```

```
ALTER TABLE deposito
    ADD CONSTRAINT fk_deposito FOREIGN KEY
    (numero_agencia) REFERENCES agencia
    (numero_agencia) ON DELETE CASCADE;
```

Fabiano Baldo

12

12



Comando de Remoção


- Especificado pela palavra reservada **DROP**.
- Usada para:
 - DROP DATABASE
 - DROP TABLE
 - DROP VIEW
 - DROP INDEX
 - DROP PROCEDURE

DROP TABLE deposito;

13

Fabiano Baldo

13




Comandos de Manipulação

- Inserção: `insert into conta (nome_agencia, numero_conta, nome_cliente, saldo)
values (Central, 9732, Smith, 1200)`
- Atualização: `update conta
set saldo = saldo * 1,05
where saldo > 1000`
- Remoção: `Delete from conta
where nome_cliente = "Smith"`

14

Fabiano Baldo

14




Comandos de Consulta

- A estrutura básica de uma expressão SQL para consulta consiste em três cláusulas: **Select**, **From** e **Where**.
 - **Select**: Corresponde à operação *projeção* da álgebra relacional. É usada para listar os atributos desejados no resultado de uma consulta;
 - **From**: Corresponde à operação *produto cartesiano* da álgebra relacional. Ela lista as relações a serem examinadas na avaliação da expressão;
 - **Where**: Corresponde ao predicado da *seleção* da álgebra relacional. Consiste em um predicado envolvendo atributos de relações que aparecem na cláusula From.

Fabiano Baldo 15

15




Projeção

- Utilizada para recuperar colunas específicas de dados;
- SELECT é o comando que executa essa operação;
- Exemplo:


```
select  c.nome_cliente,
        c.cidade_cliente
from cliente c
```

Fabiano Baldo 16

16




Seleção

- Utilizada para selecionar linhas específicas de dados;
- Esta operação é possível através da cláusula WHERE;
- Exemplo:

```
select nome_cliente, cidade_cliente, idade  
from cliente  
where nome_cliente = "Smith"
```

Fabiano Baldo 17

17




Junção (Join)

- Retorna os dados da consulta de uma ou mais tabelas em um único conjunto;
- Principais tipos de Junção
 - Junção Interna:
 - Inner Join, Natural Join
 - Junção Externa:
 - Left Outer Join, Right Join, Full Outer Join
- Inner Join também é possível através da cláusula WHERE;

Fabiano Baldo 18

18



Junção Interna


- Junção usando a cláusula WHERE:


```
select  cliente.nome_cliente, idade, saldo
from  conta, cliente
where  conta.nome_cliente = cliente.nome_cliente
```
- Junção usando Inner Join:


```
select  cliente.nome_cliente, idade, saldo
from
conta join cliente on conta.nome_cliente = cliente.nome_cliente
```

Fabiano Baldo 19

19



Operadores Aritméticos

- Executam operações matemáticas em duas expressões (ou campos) de tipo numérico;
- Operações suportadas:

+	: adição	*	: multiplicação
-	: subtração	/	: divisão
		%	: módulo
- Exemplo:


```
update conta
set saldo = saldo * 1,05
where nome_cliente = "Smith"
```

Fabiano Baldo 20

20

Operadores de Comparação

- Testa se duas expressões (ou campos) são iguais ou não;
- Operações suportadas:

= : igual	>= : maior igual
> : maior que	<= : menor igual
< : menor que	<> : diferente

```
select nome_cliente, saldo
from conta
where saldo >= 1000
```

Fabiano Baldo

21

21

Operadores Lógicos


- São comumente usados na cláusula WHERE para testar alguma condição;
- Operadores lógicos retornam um valor booleano **TRUE** ou **FALSE**;
- Operações Suportadas:

ALL	IN
AND	LIKE
ANY	NOT
IS NULL	SOME
BETWEEN	OR
EXISTS	

Fabiano Baldo

22

22



Operadores Lógicos (Exemplos)

- **AND:**


```
select cliente.nome_cliente, idade, saldo
from conta, cliente
where conta.nome_cliente = cliente.nome_cliente and
      saldo > 1000
```
- **OR:**

```
select cliente.nome_cliente, saldo
from conta
where saldo < 100 or saldo > 1000
```
- **LIKE:**

```
select cliente.nome_cliente, idade
from cliente
where nome_cliente like "S%"
```

Fabiano Baldo 23

23



Cláusula ORDER BY

- Utilizada para ordenar os valores das colunas do resultado da consulta;
- Na ausência do ORDER BY os valores do resultado são ordenados de forma crescente, a partir do primeiro campo projetado;
- Exemplo:


```
select cliente.nome_cliente, idade, saldo
from conta, cliente
where conta.nome_cliente = cliente.nome_cliente
order by
  cliente.nome asc,
  saldo desc,
  idade desc
```

Fabiano Baldo 24

24