

Banco de Dados - Ciclo 1 - Webaula 6



WEBAULA6

≡ Modelo de banco de dados relacional

≡ Referências de Imagens

QUESTION BANKS

Modelo de banco de dados relacional

Até o momento foram abordados o modelo de banco de dados hierárquico e o modelo de banco de dados em rede, cada qual com suas particularidades e funcionalidades. Mesmo assim, ambos têm características comuns, como a representação dos dados em forma de registros. Isso os torna modelos orientados a registros, pois qualquer acesso à base de dados para manipulação (consulta, inclusão, alteração ou exclusão) é feito em um registro de cada vez. A partir de agora conheceremos as características do modelo mais utilizado atualmente: o modelo relacional.



Inicialmente vamos compreender a utilização do termo “relacional”. A origem dessa denominação está associada à forma como os dados, também representados em forma de registros, estão ligados uns aos outros (ELMASRI; NAVATHE, 2012). Neste modelo não há ponteiros de ligação (links), pois os dados “se relacionam” uns aos outros através de valores próprios. Daí a nomenclatura modelo relacional.

No início da década de 70, após longos estudos tendo como base a teoria dos conjuntos e álgebra relacional, o matemático Edgar Frank Codd, apresenta uma proposta teórica para representação de dados e suas relações (DATE, 2004). Segundo o autor, este foi o marco referencial para o que conhecemos como hoje abordagem ou modelo de banco de dados relacional. Um dos objetivos da abordagem relacional foi de aumentar a independência dos dados em relação às aplicações, mas para isto era necessária a definição de um conjunto de funções para armazenamento e recuperação dos dados. A estrutura fundamental do modelo relacional é a relação, ou seja, a tabela. Uma relação é constituída por um ou mais atributos, também conhecidos como campos, que traduzem o tipo de dado a armazenar. Cada instância do esquema linha é chamado de tupla ou também registro.

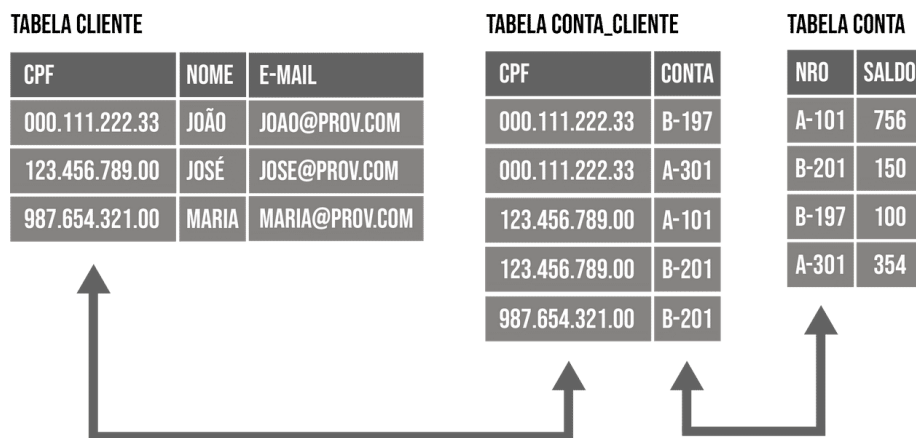


Figura 15: Representação dos dados no modelo relacional

Quanto à representação gráfica, podemos observar na figura 15 a tabela cliente e a tabela conta. Observe que CPF é a chave para a relação com a tabela conta, onde NRO é o atributo chave. Considerando que um cliente pode assumir uma ou mais contas, e por sua vez uma conta pode estar

ligada a um ou mais clientes tornou-se necessária a criação de uma estrutura de ligação: outra tabela, para estabelecer a relação entre os dados.

A seguir apresenta-se o código na linguagem SQL que define as estruturas representadas na figura 15:

```
create database BANCO;
create table CLIENTE (
    CPF      integer not null,
    NOME     varchar(30),
    EMAIL    varchar (50),
    primary key (CPF));

Create table CONTA (
    NRO      integer not null,
    SALDO    float,
    primary key(NRO));

Create table CLIENTE_CONTA (
    CPF      integer not null,
    NRO      integer not null,
    primary key (CPF, NRO),
    foreign key (CPF) references CLIENTE(CPF),
    foreign key (NRO) references CONTA(NRO));
```

Para ELMASRI e NAVATHE (2012), o modelo relacional se apresenta como a abordagem mais flexível e adequada para o armazenamento de dados nas soluções da grande maioria dos problemas que se colocam no nível da concepção e implementação da base de dados. Além disso, as estruturas do modelo para representação dos dados são consideradas simples e com isso apresentam alta performance na manipulação.

O modelo relacional não tem caminhos pré-definidos para se fazer acesso aos dados como nas demais abordagens já apresentadas: hierárquico e rede, ou seja, qualquer atributo de qualquer tabela pode ser objeto de recuperação independente de suas relações. A busca, a recuperação, assim como

qualquer operação de manipulação, inserção, alteração e exclusão são realizadas através do conjunto de comandos definidos na linguagem SQL, portanto a SQL é a linguagem padrão para a abordagem relacional.

Como já citado, a abordagem implementa estruturas de dados organizados em tabelas, porém para trabalhar com essas tabelas algumas restrições precisam ser impostas para evitar aspectos indesejáveis, como repetição e perda de informação por exemplo. Algumas dessas restrições estão relacionadas à integridade referencial e à consistência entre as tabelas, representadas no modelo através das chaves.

Observe a definição das chaves:

```
create database BANCO;
create table CLIENTE (
    CPF        integer not null,
    NOME       varchar(30),
    EMAIL      varchar(50),
    primary key (CPF));

Create table CONTA (
    NRO        integer not null,
    SALDO      float,
    primary key (NRO));

Create table CLIENTE_CONTA (
    CPF        integer not null,
    NRO        integer not null,
    primary key (CPF, NRO),
    foreign key (CPF) references CLIENTE (CPF),
    foreign key (NRO) references CONTA (NRO));
```

Reflita

Mas para que servem estas chaves (keys)?

Aqui podemos ter uma ideia de como ficaria a codificação para definir as estruturas representadas. Facilmente você pode perceber que o script de definição baseada na SQL controla a redundância nos dados não permitindo a duplicidade de determinados atributos, o que garante a unidade ou identidade de cada registro. Você deve estar se perguntando: E como se dá o relacionamento entre os registros? O relacionamento entre os registros se dá através das chaves definidas no momento da definição das estruturas. Mas o que são chaves? E para que servem? É exatamente isso que você vai descobrir em seguida.

É possível observar que há definição de chaves primárias (primary key) e chaves estrangeiras (foreign key). Entenda em que consiste cada uma:

- Chave primária: é a combinação de um ou mais atributos que identificam unicamente a entidade. O valor deve ser único para cada registro (linha).
- Chave estrangeira: é a combinação de um ou mais atributos que referenciam outra tabela. Possui o valor da chave primária da tabela referenciada.

O modelo relacional apresenta inúmeros soluções comerciais e de código aberto. Alguns produtos que na década passada eram estritamente licenciados e com valor agregado significativo, atualmente disponibilizam versões livres com nenhuma ou poucas restrições de funcionalidades, porém com limitação de espaço de gerenciamento de dados. Alguns dos produtos mais conhecidos são:

- MYSQL – Empresa Sun Microsystems
- ORACLE – Empresa Oracle
- MSSQL – Empresa Microsoft
- POSTGRESQL – Projeto Open Source do PostgreSQL Global Development Group



Dica

Você vai conhecer mais sobre cada uma das soluções existentes no mercado na unidade que trata dos Sistemas Gerenciadores de Bancos de Dados (SGBDs).

Referências de Imagens

Referências

BEIGHLEY, Lynn. Use a cabeça SQL. Rio de Janeiro: Alta Books, 2008.

DATE, C. J. Introdução a sistemas de bancos de dados. Rio de Janeiro: Elsevier, Campus, 2004.

ELMASRI, R.; NAVATHE, S. B. Sistemas de banco de dados. 6. ed. São Paulo: Pearson Education do Brasil; Addison Wesley, 2012.

GARCIA-MOLINA, H. Implementação de sistemas de bancos de dados. Rio de Janeiro: Campus, 2001.

KHOSHAFIAN, Setrag. Banco de dados orientado a objeto. Rio de Janeiro: Infobook, 1994.

LISBOA, F. Zend Framework: desenvolvendo em PHP 5 orientado a objetos com MVC. São Paulo: Novatec, 2008.

MULLER, Robert J. Projeto de banco de dados: usando UML para modelagem de dados. São Paulo: Berkeley, 2002.

RAMAKRISHNAN, R.; GEHRKE, J. Sistemas de gerenciamento de banco de dados. São Paulo: McGraw-Hill, 2008.

SILBERSCHATZ, A. Sistema de banco de dados. Rio de Janeiro: Elsevier, 2012.

SILVA, L. Banco de dados para web: do planejamento à implementação. São Paulo: Érica, 2001.

Divisão de Modalidades de ensino (DME), Fundação Universidade Regional de Blumenau (FURB), 2019