Introdução a Banco de dados

Banco de dados - Componentes de uma tabela

Entidade: Identifica o objeto de interesse do sistema e tem “vida” própria, ou seja, a representação abstrata de um objeto do mundo real sobre o qual desejamos guardar informações.

Banco de dados - Componentes de uma tabela

Atributos: São propriedades (características) que identificam as entidades. Uma entidade é representada por um conjunto de atributos. Os atributos podem ser simples, composto, multivalorado ou determinante.

Banco de dados - Componentes de uma tabela

Atributo Simples: Não possui qualquer característica especial. A maioria dos atributos serão simples. Quando um atributo não é composto, recebe um valor único como nome, por exemplo e não é um atributo chave, então ele será atributo simples.

Banco de dados - Componentes de uma tabela

Atributo Composto: O seu conteúdo é formado por vários itens menores. Exemplo: Endereço. Seu conteúdo poderá ser dividido em vários como: Rua, Número, Complemento, Bairro, Cep e Cidade.

Banco de dados - Componentes de uma tabela

Atributo Multivalorado: O seu conteúdo é formado por mais de um valor. Exemplo: Telefone. Uma pessoa poderá ter mais de um número de telefone.

SQL

A linguagem SQL surgiu em meados da década de 70, sendo resultado de um estudo de E. F. Codd, membro do laboratório de pesquisa da IBM em San Jose, Califórnia. Este estudo tinha foco em desenvolver uma linguagem que adapta-se ao modelo relacional.

SQL

O primeiro sistema de BD baseado em SQL tornou-se comercial no final dos anos 70 juntamente com outros sistema de BD's relacionais.

SQL

O sucesso da linguagem SQL foi tão grande que obrigou o ANSI (American National Standarts Institute), a padronizar as implementações da linguagem, assim, nos dias de hoje, a maior parte de BD's seguem criteriosamente esta padronização, podendo ter algumas variações, mais mesmo assim não afetando na padronização global da linguagem tornando assim a portabilidade mais fácil, se seguida de forma adequada pelo DBA.

Tipos de dados no SQL Server

Tipo	Descrição	Armazenamento
char(n)	String de caracteres de tamanho fixo, máximo de 8000 caracteres.	n
varchar(n)	String de caracteres de tamanho variável, máximo de 8000 caracteres.	
nchar(n)	Dados Unicode de tamanho fixo, máximo de 4000 caracteres.	
nvarchar(n)	Dados Unicode de tamanho variável, máximo de 4000 caracteres.	
bit	0, 1 ou nulo	
tinyint	Números inteiros de 0 a 255	1 byte
smallint	Números inteiros de -32768 a 32767	2 bytes
int	Números inteiros entre -2,147,483,648 e 2,147,483,647	4 bytes

Tipos de dados no SQL Server

Tipo	Descrição	Armazenamento
bigint	Números entre -9,223,372,036,854,775,808 e 9,223,372,036,854,775,807	8 bytes
real	Números de ponto flutuante entre -3.4×10^{38} e 3.4×10^{38}	4 bytes
datetime	De 01/01/1753 a 31/12/9999, com uma precisão de 3.33 milisegundos.	8 bytes
smalldatetime	De 01/01/1900 a 06/06/2079, com uma precisão de 1 minuto.	4 bytes
date	Data apenas. De 01/01/0001 a 31/12/9999	3 bytes
time	Hora apenas. Precisão de até 100 nanossegundos.	3-5 bytes
text	Cadeia de caracteres de tamanho variável. Até 2GB de dados.	
money	Dados monetários de -922,337,203,685,477.5808 até 922,337,203,685,477.5807	8 bytes

Tipos de dados no SQL Server

https://docs.microsoft.com/pt-br/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver15

Chave primária

A chave primária de um arquivo é um atributo ou um conjunto de atributos que identifica, de forma única, cada registro do arquivo. Todo arquivo deve possuir uma chave. A função da chave é garantir a unicidade dos registros.

Chave primária

Por exemplo, o cadastro de motoristas no Detran tem como chave o número da carteira de motoristas. Não há dois motoristas com o mesmo número de carteira.

A chave de um arquivo deve ser: Única, universal e imutável

Chave estrangeira

Chave estrangeira, ou Foreign Key (FK), ou ainda chave externa é a chave que permite a referência a registros oriundos de outras tabelas. Ou seja, é o campo ou conjunto de campos que compõem a chave primária de uma outra tabela.

Formas Normais

O processo de normalização compreende o uso de um conjunto de regras, chamados de formas normais. Ao analisarmos o banco de dados e verificarmos que ele respeita as regras da primeira forma normal, então podemos dizer que o banco está na “primeira forma normal”.

Formas Normais

Caso o banco respeite as primeiras três regras, então ele está na “terceira forma normal”. Mesmo existindo mais conjuntos de regras para outros níveis de normalização, a terceira forma normal é considerada o nível mínimo necessário para grande parte das aplicações. [Microsoft 2007]

Primeira Forma Normal (1FN)

Um banco de dados dentro dos padrões de normalização reduz o trabalho de manutenção e ajuda a evitar o desperdício do espaço de armazenamento.

Primeira Forma Normal (1FN)

Se tivermos cadastrado no banco um cliente e tivermos o seu telefone registrado em mais de uma tabela, havendo uma alteração no seu número de telefone, teremos que fazer essa atualização em cada tabela. A tarefa se torna muito mais eficiente se tivermos seu telefone registrado em apenas uma tabela.

Primeira Forma Normal (1FN)

A primeira forma normal serve para reprovar atributos multivalorados, compostos e suas combinações. O domínio de um atributo deve incluir apenas valores atômicos (indivisíveis), e o valor de qualquer atributo em uma tupla deve ser único, como valor no domínio desse atributo.

Primeira Forma Normal (1FN)

- Somente possui valores atômicos (simples e indivisíveis)
- Não há grupos de atributos repetidos (há apenas um dado por coluna nas linhas)
- Existe uma chave primária
- Relação não possui atributos multivalorados ou relações aninhadas

Primeira Forma Normal (1FN)

```
mysql> select * from cliente;
+-------------+--------------+----------------------+-----------------------------------------+
| cod_cliente | nome_cliente | tel_cliente          | endereco_cliente                        |
+-------------+--------------+----------------------+-----------------------------------------+
|        2453 | Ana          | 4213-6532/975635632  | Rua Min. Alberto Jorge 1492 - Vila Primavera |
|        2532 | José         | 99653-2145/2865-3212 | Rua das Giestas, 234, ap.45- Vila Bela  |
|        2536 | Marcos       | 2643-5321            | Av. Carlos de Almeida, 459 - Vila das Rosas |
+-------------+--------------+----------------------+-----------------------------------------+
```

Tabela não normalizada

Primeira Forma Normal (1FN)

cod_cliente	nome_cliente	tel1_cliente	tel2_cliente	endereco_cliente	complemento	bairro
2453	Ana	4213-6532	975635632	Rua Min. Alberto Jorge 1492	S/C	Vila Primavera
2532	Jose	99653-2145	2865-3212	Rua das Giestas, 234	Ap.45	Vila Bela
2536	Marcos	2643-5321		Av. Carlos de Almeida, 459	S/C	Vila das Rosas

Tabela 1FN

Segunda Forma Normal (2FN)

- Estar na 1FN
- Todos os atributos que não-chave são funcionalmente dependentes de todas as partes da chave primária.
- Não existem dependências parciais.
- Caso contrário, deve-se gerar uma nova tabela com os dados.

Segunda Forma Normal (2FN)

Deve-se criar uma nova relação para cada chave PK ou combinação de atributos que forem determinantes em uma dependência funcional. Esse atributo será a PK na nova tabela.

Segunda Forma Normal (2FN)

Mova os atributos não-chave dependentes desta PK para a nova tabela. Após aplicar a segunda forma normal, teremos:

Segunda Forma Normal (2FN)

tbl_Peças					
Cod_Peça	Cod_Fornec	Local_Fornecedor	Qtde_Estoque	Tel_Fornecedor	Qtde_Caixas
0009	121	São Paulo	512	2365-6532	52
0023	122	Manaus	263	4465-8632	27
0065	121	São Paulo	196	2365-6532	20
0071	123	Porto Alegre	89	2956-8653	9
0073	122	Manaus	296	4465-8632	30

Tabela apenas na 1FN

Segunda Forma Normal (2FN)

tbl_Peça			
Cod_Peça	Cod_Fornec	Qtde_Estoque	Qtde_Caixas
0009	121	512	52
0023	122	263	27
0065	121	196	20
0071	123	89	9
0073	122	296	30

Tabela Peça – 2FN

Segunda Forma Normal (2FN)

tbl_Fornecedor		
Cod_Fornec	Local_Fornecedor	Tel_Fornecedor
121	São Paulo	2365-6532
122	Manaus	4465-8632
PK 123	Porto Alegre	2956-8653

Tabela Fornecedor – 2FN

Terceira Forma Normal (3FN)

Uma tabela está na 3FN se ela estiver na segunda forma normal e se nenhuma coluna não-chave depender de outra coluna não-chave.

Terceira Forma Normal (3FN)

Para cada atributo não—chave que for um determinante na relação, crie uma nova tabela. Esse atributo será a PK na nova relação.

Terceira Forma Normal (3FN)

Mova então todos os atributos que são dependentes funcionalmente do atributo chave para a nova tabela. O Atributo (PK na nova relação) fica também na tabela original, e servirá como uma chave estrangeira para associaras duas relações.

Terceira Forma Normal (3FN)

tbl_Venda				
Nota_Fiscal	Cod_Vendedor	Nome_Vendedor	Cod_Produto	Qtde_vendida
15326	002	Leila	132	10
15327	006	Ana	153	12
15328	002	Leila	143	11
15329	009	Fábio	132	9
15330	007	Renato	153	12

Tabela fora da – 3FN

Terceira Forma Normal (3FN)

tbl_Venda			
Nota_Fiscal	Cod_Vendedor	Cod_Produto	Qtde_vendida
15326	002	132	10
15327	006	153	12
15328	002	143	11
15329	009	132	9
15330	007	153	12

Tabela Venda – 3FN

Terceira Forma Normal (3FN)

tbl_Vendedor	
Cod_Vendedor	Nome_Vendedor
002	Leila
006	Ana
007	Renato
009	Fábio

Tabela Vendedor – 3FN

Categorias de comandos Linguagem SQL

- **DDL** - Data Definition Language - Linguagem de Definição de Dados. São os comandos que interagem com os objetos do banco.
  - São comandos DDL : CREATE, ALTER e DROP

Categorias de comandos Linguagem SQL

- **DML** - Data Manipulation Language - Linguagem de Manipulação de Dados. São os comandos que interagem com os dados dentro das tabelas.
  - São comandos DML : INSERT, DELETE e UPDATE

Categorias de comandos Linguagem SQL

- **DCL** - Data Control Language - Linguagem de Controle de Dados. São os comandos para controlar a parte de segurança do banco de dados.
  - São comandos DCL : GRANT, REVOKE E DENY.

Categorias de comandos Linguagem SQL

- **DQL** - Data Query Language - Linguagem de Consulta de dados. São os comandos de consulta.

São comandos DQL : SELECT (é o comando de consulta)

Categorias de comandos Linguagem SQL

- **DTL** - Data Transaction Language - Linguagem de Transação de Dados. São os comandos para controle de transação.
  - São comandos DTL : BEGIN TRANSACTION, COMMIT E ROLLBACK