

Matéria: Tecnologia da Informação

Assunto: Banco de Dados

Resumo Teórico do Assunto

Para ter sucesso nas questões sobre Banco de Dados, é fundamental compreender os conceitos do modelo relacional, a linguagem SQL, os diferentes tipos de bancos de dados NoSQL e os princípios de modelagem de dados.

1. Introdução ao Banco de Dados

Um **Banco de Dados** é um conjunto organizado de informações (dados) armazenadas e acessadas eletronicamente. Ele é essencial para gerenciar grandes volumes de dados de forma eficiente, segura e consistente.

2. Modelo Relacional e SQL

O **Modelo Relacional** é o tipo mais comum de banco de dados, onde os dados são organizados em **tabelas** (também chamadas de relações).

- **Tabelas:** São compostas por:

- * **Colunas** (ou atributos/campos): Representam as características dos dados.

- * **Linhas** (ou registros/tuplas): Representam uma ocorrência única dos dados.

- **Chaves:**

- * **Chave Primária (PK):** Uma coluna ou conjunto de colunas que identifica unicamente cada linha em uma tabela. Seus valores não podem ser nulos e devem ser únicos.

- * **Chave Estrangeira (FK):** Uma coluna ou conjunto de colunas em uma tabela que referencia a **Chave Primária** de outra tabela. Ela estabelece um **relacionamento** entre as tabelas, garantindo a **integridade referencial** (ou seja, que os dados relacionados existam na tabela referenciada).

- **SQL (Structured Query Language):** É a linguagem padrão para interagir com bancos de dados relacionais.

- * **`SELECT`:** Usado para recuperar dados do banco de dados.

- * **`SELECT coluna1, coluna2 FROM tabela;`** - Seleciona colunas específicas.

- * **`SELECT * FROM tabela;`** - Seleciona todas as colunas.

- * **`FROM`:** Indica de qual(is) tabela(s) os dados serão recuperados.

- * **`WHERE`:** Filtra as linhas com base em uma condição especificada.

- * **`JOIN`:** Usado para combinar linhas de duas ou mais tabelas com base em uma coluna relacionada (geralmente uma FK). Quando você lista múltiplas tabelas na cláusula **`FROM`** e especifica a condição de junção na cláusula **`WHERE`** (ex: **`TabelaA.coluna =**

TabelaB.coluna`), isso é uma forma de **junção implícita** (ou **produto cartesiano** seguido de filtro). A forma mais recomendada e explícita é usar `INNER JOIN`, `LEFT JOIN`, etc.

3. Bancos de Dados NoSQL

NoSQL (Not only SQL) é uma categoria de bancos de dados que oferece alternativas ao modelo relacional, sendo mais flexíveis e escaláveis para certos tipos de dados e cargas de trabalho. Eles não usam SQL como linguagem principal e não seguem o esquema rígido das tabelas relacionais.

- **Tipos Comuns de NoSQL:**

- * **Chave/Valor:** Armazena dados como um dicionário, onde cada item tem uma chave única e um valor associado. Simples e rápido para acesso direto.

- * **Orientado a Documentos:** Armazena dados em documentos (geralmente JSON, BSON ou XML), que podem ter estruturas complexas e aninhadas. Flexível para dados semi-estruturados.

- * **Orientado a Colunas (ou Família de Colunas):** Armazena dados em famílias de colunas, otimizado para grandes volumes de dados e consultas analíticas.

- * **Orientado a Grafos:** Ideal para representar e consultar dados com **relacionamentos complexos e interconectados**, como redes sociais, sistemas de recomendação ou cadeias de suprimentos. Os dados são armazenados como **nós** (entidades) e **arestas** (relacionamentos) entre eles, permitindo consultas eficientes sobre as conexões.

4. Modelagem de Dados e Criação de Tabelas

A **Modelagem de Dados** é o processo de criar um modelo abstrato e conceitual de como os dados são armazenados, processados e acessados.

- **Diagrama Entidade-Relacionamento (E-R):** Uma ferramenta visual para modelar dados, representando:

- * **Entidades:** Objetos ou conceitos do mundo real (ex: Empresa, CNAE).

- * **Atributos:** Características das entidades (ex: CNPJ, razaoSocial).

- * **Relacionamentos:** Associações entre entidades (ex: Empresa *tem* um CNAE).

- **Mapeamento E-R para o Modelo Relacional:**

- * Cada **entidade** geralmente se torna uma **tabela**.

- * Cada **atributo** se torna uma **coluna** na tabela correspondente.

- * **Relacionamentos** são representados por **chaves estrangeiras (FKs)** nas tabelas. A chave primária da entidade "pai" é adicionada como uma chave estrangeira na tabela da entidade "filho".

- **Comando `CREATE TABLE`:** Usado para definir a estrutura de uma nova tabela no SQL.

- * **Sintaxe Básica:**

```
```sql
```

```
CREATE TABLE NomeTabela (
 NomeColuna1 TipoDados1 [NOT NULL],
 NomeColuna2 TipoDados2,
```

```
PRIMARY KEY (NomeColunaPK),
FOREIGN KEY (NomeColunaFK) REFERENCES
TabelaReferenciada(ColunaPKReferenciada)
);
...
```

\* **`NOT NULL`**: Restrição que impede que uma coluna tenha valores nulos.

## 5. Tipagem de Dados e Inferência (Contexto de Linguagens de Programação)

Embora a questão 60 seja sobre TypeScript, o conceito de **tipagem** é fundamental para a integridade dos dados, tanto em linguagens de programação quanto em bancos de dados (onde cada coluna tem um tipo de dado definido).

- **Tipagem Estática:** Em linguagens com tipagem estática (como Java, C#, e **TypeScript**), os tipos das variáveis são verificados em tempo de **compilação**. Isso significa que, uma vez que uma variável é declarada com um tipo (explicitamente ou por **inferência de tipo**), ela só pode armazenar valores desse tipo. Tentar atribuir um valor de um tipo diferente resultará em um **erro de compilação**, pois a linguagem garante a consistência dos tipos.
- **Inferência de Tipo:** É a capacidade de um compilador ou interpretador de determinar automaticamente o tipo de uma variável com base no valor que lhe é atribuído, sem que o programador precise declará-lo explicitamente. Mesmo com inferência, em tipagem estática, o tipo é "fixado" após a inferência inicial.
- **Impacto:** A tipagem estática ajuda a prevenir erros em tempo de execução, garantindo a consistência dos dados e a robustez do código.

## Questões de Provas Anteriores

Fonte: [escriturario\\_agente\\_de\\_tecnologia.pdf](#), Página: 17

pcimarkpci MjgwNDowMTRkOjE0YTU6OTI1ODozOGQ2OjNhMGM6NTM0MzplZmI1:U3V  
uLCAyNyBKdWwgMjAyNSAyMzo0NzozMSAtMDMwMA==

[www.pciconcursos.com.br](http://www.pciconcursos.com.br)

AGENTE DE TECNOLOGIA - Microrregião 158 -TI

17

GABARITO 1

BANCO DO BRASIL

58

A CNAE (Classificação Nacional de Atividades Econômicas), de responsabilidade do IBGE, possui códigos que são utili -

zados para caracterizar as atividades econômicas das empresas no Brasil. Por exemplo: empresas da área de construção

de edifícios utilizam o CNAE de código 4120-4/00 para caracterizar a sua atividade

**econômica principal. Considere que**

**existe um banco de dados em uma empresa, que concede empréstimos a outras empresas, com as seguintes tabelas:**

**Empresa (CNPJ, razaoSocial, endereco, atividade)**

**CNAE (codigo, descricao)**

**A coluna "atividade" da tabela Empresa é uma chave estrangeira que referencia a coluna "codigo" da tabela CNAE.**

**Que comando SELECT do SQL retorna apenas o CNPJ e a razão social das empresas cuja atividade econômica principal**

**é a construção de edifícios (código 4120-4/00)?**

(A) SELECT \*

FROM Empresa E, CNAE C

WHERE E.atividade = C.codigo

(B) SELECT \*

FROM Empresa

WHERE atividade = 'construção de edifícios'

(C) SELECT CNPJ, razaoSocial

FROM CNAE

WHERE codigo = '4120-4/00'

(D) SELECT CNPJ, razaoSocial

FROM Empresa E, CNAE C

WHERE E.atividade = C.codigo

AND C.codigo = 'construção de edifícios'

(E) SELECT CNPJ, razaoSocial

FROM Empresa

WHERE atividade = '4120-4/00'

59

Considere um cenário no qual uma empresa de investimentos financeiros tenha realizado uma campanha para que seus colaboradores indicassem outras pessoas que pudessem vir a estar interessadas em seus produtos e serviços. Esses colaboradores também foram estimulados a informar a sua ligação com as pessoas indicadas (ex: amigo, irmão, primo) e as eventuais ligações entre as pessoas indicadas. Estes relacionamentos são importantes para a definição da próxima campanha dessa empresa.

O banco de dados NoSQL mais indicado para representar esses dados é o que utiliza o modelo

(A) chave/valor

(B) orientado a colunas

(C) orientado a documentos

(D) orientado a grafos

(E) relacional

60

O código abaixo está em linguagem TypeScript, cuja tipagem é estática.

```
let fraseLegal = 'Bom dia!';
```

```
fraseLegal = 9.5;
```

Analisando o código apresentado, tem-se que a TypeScript

(A) não realiza a inferência de tipo, por isso, o código apresentado vai rodar sem erro.

(B) realiza a inferência de tipo, por isso, o código apresentado vai rodar sem erro.

(C) realiza a inferência de tipo, por isso, aceita a atribuição do tipo number para um tipo string sem gerar erro de compilação do código apresentado.

(D) realiza a inferência de tipo, por isso, não aceita a atribuição do tipo number para um tipo string, o que vai gerar um erro de compilação do código apresentado.

(E) não diferencia valores ponto flutuante (decimal) de valores inteiros, por isso, o código apresentado vai rodar sem erro.

---

Fonte: [escrituario\\_agente\\_de\\_tecnologia.pdf](#), Página: 25

pcimarkpci MjgwNDowMTRkOjE0YTU6OTI1ODozOGQ2OjNhMGM6NTM0MzplZmI1:U3VuLCAYNyBKdWwgMjAyNSAyMzo0NzozMSAtMDMwMA==  
[www.pciconcursos.com.br](http://www.pciconcursos.com.br)

25

**BANCO DO BRASIL**

**AGENTE DE TECNOLOGIA - Microrregião 158 -TIGABARITO 1**

**Qual transformação da relação Rel irá preservar a semântica do diagrama E-R apresentado?**

(A) CREATE TABLE REL (  
X1 TEXT NOT NULL,  
Y1 TEXT NOT NULL,  
R1 TEXT NOT NULL,  
R2 TEXT NOT NULL,  
PRIMARY KEY (X1, Y1),  
FOREIGN KEY (X1)  
REFERENCES TX (X1),

```

FOREIGN KEY (Y1)
REFERENCES EY (Y1));
(B) CREATE TABLE REL (
X1 TEXT NOT NULL,
Y1 TEXT NOT NULL,
R1 TEXT NOT NULL,
R2 TEXT NOT NULL,
PRIMARY KEY (X1, R1),
FOREIGN KEY (X1)
REFERENCES TX (X1),
FOREIGN KEY (Y1)
REFERENCES EY (Y1));
(C) CREATE TABLE REL (
X1 TEXT NOT NULL,
Y1 TEXT NOT NULL,
R1 TEXT NOT NULL,
R2 TEXT NOT NULL,
PRIMARY KEY (Y1, R1),
FOREIGN KEY (X1)
REFERENCES TX (X1),
FOREIGN KEY (Y1)
REFERENCES EY (Y1));
(D) CREATE TABLE REL (
X1 TEXT NOT NULL,
Y1 TEXT NOT NULL,
R1 TEXT NOT NULL,
R2 TEXT NOT NULL,
PRIMARY KEY (X1, Y1, R1),
FOREIGN KEY (X1)
REFERENCES TX (X1),
FOREIGN KEY (Y1)
REFERENCES EY (Y1));
(E) CREATE TABLE REL (
X1 TEXT NOT NULL,
Y1 TEXT NOT NULL,
R1 TEXT NOT NULL,
R2 TEXT NOT NULL,
PRIMARY KEY (R1),
FOREIGN KEY (X1)
REFERENCES TX (X1),
FOREIGN KEY (Y1)
REFERENCES EY (Y1));

```