

Matéria: Tecnologia da Informação

Assunto: Banco de Dados Relacionais

Resumo Teórico do Assunto

Para dominar as questões sobre **Banco de Dados Relacionais**, especialmente aquelas que envolvem a conversão de modelos conceituais (como o Diagrama Entidade-Relacionamento) para o modelo lógico relacional, é fundamental compreender os seguintes conceitos:

1. Modelo Relacional: A Base dos Bancos de Dados

O **Modelo Relacional** é o modelo de dados mais amplamente utilizado para organizar e gerenciar informações. Ele se baseia na ideia de representar os dados em **tabelas bidimensionais**, também chamadas de **relações**.

- **Tabela (Relação):** Uma estrutura que organiza dados em linhas e colunas.
- * **Tupla (Linha/Registro):** Representa uma única ocorrência de um conjunto de dados. Cada linha em uma tabela é uma tupla.
- * **Atributo (Coluna/Campo):** Representa uma característica ou propriedade dos dados. Cada coluna em uma tabela é um atributo.
- * **Domínio:** O conjunto de valores permitidos para um atributo.

2. Chaves: A Identificação e Conexão dos Dados

As chaves são atributos ou conjuntos de atributos cruciais para a integridade e o relacionamento dos dados.

• Chave Primária (PK - Primary Key):

- * É um atributo (ou um conjunto de atributos) que **identifica unicamente cada tupla (linha)** em uma relação (tabela).
- * **Características Essenciais:**
 - * **Unicidade:** Nenhum valor da chave primária pode se repetir dentro da mesma tabela.
 - * **Não Nula (NOT NULL):** Nenhum atributo que compõe a chave primária pode conter valores nulos.
- * **Notação na Questão:** Colunas sublinhadas (`_coluna_`) indicam a chave primária.

• Chave Estrangeira (FK - Foreign Key):

- * É um atributo (ou um conjunto de atributos) em uma tabela que **referencia a Chave Primária (PK)** de outra tabela (ou, em casos específicos, da mesma tabela).
- * **Propósito:** Estabelecer **relacionamentos** entre tabelas, garantindo a **Integridade**

Referencial.

* **Integridade Referencial:** Garante que, para cada valor de chave estrangeira, exista um valor correspondente na chave primária da tabela referenciada, ou que o valor da chave estrangeira seja nulo (se permitido). Isso evita "links quebrados" entre os dados.

* **Notação na Questão:** Representada pela cláusula `REF: REF`. Ex: `x1 REF XB` significa que `x1` é uma chave estrangeira que referencia a chave primária da tabela `XB`.

3. Atributos Nulos (NULL Values)

- Um atributo pode ser **nulo** quando seu valor é desconhecido, não aplicável ou não fornecido.

- **Notação na Questão:** Colunas que admitem o valor nulo são exibidas entre colchetes (`[coluna]`).

- **Importante:** Atributos que fazem parte de uma Chave Primária **NÃO** podem ser nulos.

4. Diagrama Entidade-Relacionamento (E-R Diagram)

O **Diagrama E-R** é uma ferramenta de modelagem conceitual que representa a estrutura de um banco de dados de forma abstrata e visual, antes de sua implementação.

- **Entidade:** Um objeto ou conceito do mundo real sobre o qual se deseja armazenar informações (ex: Cliente, Produto, Pedido). No modelo relacional, uma entidade geralmente se torna uma **tabela**.

- **Atributo:** Uma propriedade ou característica de uma entidade (ex: nome do cliente, preço do produto). No modelo relacional, um atributo se torna uma **coluna** na tabela.

- **Relacionamento:** Uma associação entre duas ou mais entidades (ex: um Cliente **faz** um Pedido).

5. Mapeamento de Diagramas E-R para o Modelo Relacional

A habilidade de converter um Diagrama E-R em um esquema relacional é crucial. As regras gerais são:

- **Entidades Fortes:** Cada entidade forte no Diagrama E-R é mapeada para uma **tabela** no modelo relacional. Os atributos da entidade se tornam as colunas da tabela, e o identificador da entidade se torna a **Chave Primária** da tabela.

- **Atributos:**

- * **Atributos Simples:** Tornam-se colunas na tabela correspondente.

- * **Atributos Compostos:** Cada componente do atributo composto se torna uma coluna separada.

- * **Atributos Multivalorados:** Geralmente exigem a criação de uma **nova tabela** separada para armazenar os valores múltiplos. A chave primária dessa nova tabela será composta pela chave primária da entidade original (como FK) e pelo próprio atributo multivalorado.

- **Relacionamentos:**

- * **Relacionamentos 1:1 (Um para Um):**

- * A chave primária de uma das entidades é adicionada como **Chave Estrangeira** na tabela da outra entidade.

- * A escolha de qual tabela recebe a FK depende da participação (obrigatória/opcional) ou de decisões de design. Se um lado tem participação total (obrigatória), a FK geralmente vai para o lado com participação parcial (opcional).

- * **Relacionamentos 1:N (Um para Muitos):**

- * A regra mais comum: A chave primária da entidade do lado "um" é adicionada como **Chave Estrangeira** na tabela da entidade do lado "muitos".

- * Exemplo: Um departamento tem muitos funcionários. A PK de Departamento (`id_departamento`) vai como FK para a tabela Funcionário.

- * **Relacionamentos N:M (Muitos para Muitos):**

- * Exigem a criação de uma **nova tabela associativa (ou de ligação)** para representar o relacionamento.

- * A chave primária dessa nova tabela é geralmente uma **chave composta** formada pelas chaves primárias das duas entidades participantes, que também atuam como **Chaves Estrangeiras** referenciando suas respectivas tabelas.

- * Essa nova tabela pode ter seus próprios atributos, se o relacionamento tiver atributos.

- * Exemplo: Muitos alunos podem cursar muitas disciplinas. Cria-se uma tabela `Matricula` com `id_aluno` (FK) e `id_disciplina` (FK) como PK composta.

- * **Entidades Fracas:**

- * Uma entidade fraca depende da existência de uma entidade forte para ser identificada.

- * É mapeada para uma **tabela** separada.

- * Sua **Chave Primária** será composta pela chave primária da entidade forte da qual ela depende (como FK) e por seu próprio atributo discriminador (chave parcial).

6. Preservação da Semântica e Regras de Integridade

A questão busca um esquema relacional que "preserva a semântica desse diagrama E-R, sem a necessidade de criação de regras de integridade adicionais". Isso significa que o próprio design das tabelas, com suas chaves primárias e estrangeiras, e a definição de atributos nulos/não nulos, deve ser suficiente para impor as restrições e o significado do modelo conceitual.

- A correta aplicação das regras de mapeamento (especialmente para relacionamentos e chaves) é o que garante que a **Integridade de Entidade** (PK única e não nula) e a **Integridade Referencial** (FKs válidas) sejam mantidas, refletindo a lógica do Diagrama E-R.

- Se um relacionamento é 1:N, a FK no lado N já impõe que um registro no lado N só pode existir se houver um correspondente no lado 1. Se um relacionamento N:M é mapeado para uma tabela associativa, a PK composta dessa tabela já garante a unicidade das combinações e a integridade referencial.

Ao entender e aplicar esses princípios, você estará apto a analisar um Diagrama E-R e

derivar o esquema relacional correto que reflete sua semântica.

Questões de Provas Anteriores

Fonte: escriturario_agente_de_tecnologia (1).pdf, Página: 19

pcimarkpci MjgwNDowMTRkOjE0YTU6OTI1ODozOGQ2OjNhMGM6NTM0MzplZmI1:U3V
uLCAyNyBKdWwgMjAyNSAyMzo0Nzo0MCAtMDMwMA==

www.pciconcursos.com.br

AGENTE DE TECNOLOGIA - Microrregião 16 DF-TI19

BANCO DO BRASIL

GABARITO 1

52

Esquemas de bancos de dados relacionais podem ser descritos por meio de notações alternativas à linguagem SQL.

Em uma dessas notações, uma tabela (relação) é descrita por meio de um nome e de um conjunto de nomes de colunas (atributos), separadas por vírgulas.

Os demais elementos de uma tabela são os seguintes:

- Colunas sublinhadas compõem a chave primária.
- Colunas que admitem o valor nulo são exibidas entre colchetes.
- Chaves estrangeiras são representadas por meio da cláusula REF:<lista_de_colunas> REF <nome_de_tabela>

Nesse contexto, considere a Figura a seguir, que exibe um diagrama E-R.

Qual esquema relacional preserva a semântica desse diagrama E-R, sem a necessidade de criação de regras de integridade adicionais?

(D) XB(x1, x2)

TA(t1, t2, tipo, [t3], [t4], [t5])

S(x1, t1, s1)

x1 REF XB

t1 REF TA

(E) XB(x1, x2, t1, s1)

t1 REF TA

TA(t1, t2, tipo)

TB(t1, t3, t4)

t1 REF TA

TC(t1, t5)

t1 REF TA

(A) XB(x1, x2, t1, s1)

t1 REF TA

TA(t1, t2, tipo, t3, t4, t5)
(B) XB(x1, x2)
TA(t1, t2, tipo)
TB(t1, t3, t4)
t1 REF TA
TC(t1, t5)
t1 REF TA
S(x1, t1, s1)
x1 REF XB
t1 REF TA
(C) XB(x1, x2)
TA(t1, t2, tipo, [t3], [t4], [t5])
S(x1, t1, s1)
x1 REF XB
t1 REF TA