

Chave: Conceitualmente, entidades e relacionamentos individuais são distintos, entretanto, na perspectiva do banco de dados, a diferença entre ambos deve ser estabelecida em termos de seus atributos. O conceito de chave permite-nos fazer tais distinções.

Super chave: é um atributo ou um conjunto de atributos que identificam univocamente um tuplo numa relação

Chave candidata: é uma super chave tal que nenhum seu subconjunto é uma super chave na relação

Chave primária: chave candidata seleccionada para permitir identificação unívoca de um tuplo numa relação

Chave Estrangeira: É um atributo ou conjunto de atributos cujos valores aparecem necessariamente na chave primária de uma tabela. Este mecanismo permite a implementação de relacionamentos no modelo relacional.

Restrições de integridade: restrições impostas na BD para impedir que fique incompleta, incorrecta ou inconsistente

Tipos de restrições de integridade:

- **Dados requeridos:** alguns atributos não permitem valores nulos
- **Domínio de atributos:** cada atributo pode ter apenas valores definidos num dado conjunto
- **Restrições às entidades:** restrições colocadas nos relacionamentos entre entidades na BD
- **Integridade de entidades:** a chave primária não pode ser nula
- **Integridade referencial:** se a chave estrangeira possui um valor, esse valor deve referir-se a um tuplo existente na relação pai

Dependência funcional: descreve o relacionamento entre atributos de uma relação. Por exemplo, se A e B são atributos de R, B é funcionalmente dependente em A se cada valor de A está associado com exactamente um valor de B. Uma dependência funcional designa-se de restrição entre os atributos

Normalização: técnica usada para produzir um conjunto de relações com propriedades desejadas dados requisitos. É uma forma de garantir que a BD está livre de certas características indesejáveis (anomalias no insert, update e delete), que poderão levar à perda da integridade de dados. Diz-se que uma BD está normalizada quando atingir a 3ª forma normal (3FN).

1FN – não há tuplos repetidos numa relação, atomicidade (ou: A intersecção de cada linha e de cada coluna numa tabela contém um e só um valor atómico.)

2FN – não há dependências parciais (ou: Se cada atributo não participante na chave primária for totalmente dependente desta.)

3FN – não há dependências transitivas (ou: Se cada atributo não participante na chave primária for dependente de forma não transitiva da chave primária.)

Ao remover estas anomalias, a BD torna-se mais flexível, poderosa e menos complexa

DBMS/SGBD: conjunto de aplicações para criação, manutenção e utilização de BDs, e não necessita de aplicações exteriores para extrair dados

Funções de um SGBD: definição e criação de BDs; consultas e relatórios em BDs; actualização de conteúdos das BDs; facilitar a criação de aplicações pelo utilizador; revisões e reestruturações de BDs; controlo de integridade de BDs; monitorização de performance; providenciar segurança; recuperação de dados.

Tarefas de um administrador de um SGBD: definição dos esquemas, definição de métodos de acesso e estrutura de armazenamento, modificação do esquema e da organização física, definição de autorizações de acesso aos utilizadores, especificação de restrições de integridade, manutenção do sistema

Desenho da BD:

- **Conceptual:** construção do modelo independente de qualquer consideração do ponto de vista físico
- **Lógico:** refinamento do modelo conceptual, definição de acordo com o modelo a utilizar, mas são ainda ignorados factores físicos, com estrutura de armazenamento, índices
- **Físico:** última fase do desenho, onde é decidido como implementa a BD. Apesar de ser independente de um DBMS é necessário primeiro escolhê-lo, de forma a tomar decisões quanto à performance é necessário criar tabelas relacionais e restrições vindos do modelo lógico; identificar estruturas de armazenamento e métodos de acesso; desenho da protecção de segurança

Níveis de abstracção:

- **Vistas/externo:** consiste na forma como cada utilizador “vê” a BD. Cada utilizador vê apenas um conjunto de entidades, atributos e relacionamentos, tudo o resto não lhe é visível
- **Conceptual:** nível intermédio; contém toda a estrutura lógica da BD. É a vista completa de toda a BD e cumpre os requisitos de organização de dados, mas é ainda independente de requisitos de armazenamento físico; contém todas as entidades, atributos e relacionamentos da BD
- **Físico/interno:** implementação física da BD num DBMS, já com requisitos de armazenamento, etc....

Bases de dados distribuídas: É uma colecção de informação logicamente relacionada, distribuída fisicamente numa rede

- **vantagens:** organização estrutural (fragmentos localizam-se nos departamentos onde pertencem), autonomia local, protecção de dados importantes, maior performance, economia, modularidade, transacções confiáveis
- **desvantagens:** complexidade, gastos (devido à complexidade), segurança, dificuldade em manter integridade, inexperiência (área nova), falta de standards, complexidade do desenho da BD, necessário software adicional, sistema operativo deve comportar ambiente distribuído, controlo de concorrência

Transacção: consiste numa unidade de trabalho numa BD. Conjunto de operações que produz sempre um resultado positivo. Caso alguma das operações falhe, o bloco falha por inteiro e todas as operações anteriores voltam ao estado anterior ao início da transacção (rollback)

Uma transacção deve sempre transformar a base de dados de um estado consistente para outro estado também consistente, embora durante a transacção possa haver inconsistência.

Uma transacção possui dois estados:

- Committed: executou com sucesso
- Aborted - se a transacção não executou com sucesso, é preciso fazer rollback.

Transacções possuem a propriedade **ACID** (**A**tomicidade, **C**onsistência, **I**solamento, **D**urabilidade)

Locking

: Procedimento usado para controlar o acesso aos dados. Quando uma transacção esta a aceder a algum conjunto de dados, um lock bloqueia o acesso a qualquer outra transacção que necessite aceder a esses dados.

-> Shared Lock

Pode ler mas não modificar.

-> Exclusive Lock

Pode ler e modificar.

-> 2 Phase-Locking

Todas as operações de Locking precedem o primeiro Unlock.

Cada transacção está dividida em 2 fases:

- Growing phase: os locks precisos são obtidos mas nenhum é libertado
- Shrinking phase: todos os locks são libertados e não se pode obter mais nenhum lock

-> Cascading Rollback

Quando varias transacções estão a executar e são dependentes umas das outras, se uma falha e faz rollback, todas as outras dependentes fazem rollback

-> Deadlock

Ocorre quando duas(ou mais transacções) estão à espera que um lock seja libertado

- > Timeouts: espera um período de tempo, e depois aborta e reinicia
- > Prevention: prevê que vá acontecer deadlocks
- > Detection: deixa ocorrer deadlocks mas reconhece as ocorrências e separa-as

Two-phase commit: assegura um mecanismo automático de recuperação caso algo falhe durante uma transacção, permitindo a base de dados continuar sincronizada como antes do início da transacção.

Uncommitted dependency problem: ocorre quando uma transacção consegue ver os resultados intermédios de uma outra transacção que ainda não terminou (committed). Pode ser evitado se não se permitir a leitura dos registos envolvidos nessa transacção por outras, enquanto a primeira não terminar (com ou sem sucesso)

Cursors: buscar, guardar, apagar registos. Permite a navegação transversal da BD. São usados para operar numa colecção de registos e iterar sobre eles ou sobre toda a BD (Declarar, Abrir, Posicionar/Manipular, Fechar, Retirar da Memória).

Trigger: é um procedimento que é executado em resposta a certos eventos numa dada tabela ou view da BD. Normalmente são utilizados para manter a integridade da BD; outros usos incluem a prevenção de escrita de dados, logs, etc.

Catálogo: é uma parte da base de dados que contem a definição de todos os objectos presentes na base de dados (tabelas, views, sinónimos, índices, utilizadores, grupos, etc.), bem como a definição da base de dados.

View: é uma tabela virtual construída a partir do resultado de uma query. Não faz parte do esquema físico da BD. (Se for definida como actualizavel, qualquer alteração nos registos da view provocam alterações nas tabelas da

BD.) As views podem ser utilizadas para representar uma parte da BD numa única tabela; podem fazer a junção e simplificar múltiplas tabelas numa única tabela virtual; ocupam pouco espaço (apenas possui a definição da view), podem ajudar a esconder a complexidade dos dados e podem providenciar maior segurança.

-> Relação

: É uma tabela com linhas e colunas

-> Atributo: É o nome da coluna de uma relação

-> Tuplo

: É a linha de uma relação

-> Grau

: O grau de uma relação é o numero de atributos que contem

-> Cardinalidade

: Numero de tuplos que possui