

INSERT → insere os dados numa tabela

UPDATE → atualiza os dados

DELETE → apaga dados.

Sistema de Gestão de Base de dados Relacional

O SGBD aceita instruções SQL, produz a consulta e mostra os resultados.

Otimizador de Consultas (query optimizer)

- Usa a informação como os dados estão guardados para produzir um plano de execução.

Plano de Execução (execution plan)

- Avalia a consulta e geralmente apresenta-se como uma árvore de operações relacionais.

Gestor de Memórias (buffer manager)

- Sendo um SGBD uma coleção de páginas/ficheiros, é o gestor de memórias que traz as páginas do disco para a memória principal, conforme necessário, em resposta às solicitações de leitura.

Gestor de Espaço em Disco (disk space manager)

- É a camada mais baixa do SGBD, as camadas superiores alocam e desalocam, leem e escrevem páginas através de funções fornecidas por esta camada.

Gestor de Transações (transaction manager)

- É um gestor que garante que cada transação solicitada e liberta os recursos segundo um protocolo de transações apropriado e agenda a execução das transações.

Gestor de Trancos (lock manager)

- Guarda toda a informação sobre os registos de pedidos e de concessões de trancos sobre os objetos da base de dados e quando estes ficam disponíveis.

Gestor de Recuperações (recovery manager)

- Mantém um registo e permite restaurar para um estado consistente após uma falha.

2.3.2 Processamento de Consultas

Transformar a consulta numa estratégia de execução, correta e eficiente, sendo o foco obter a melhor maneira de avaliar essa consulta.

A descrição de dados, ou metadados, guarda em tabelas especiais designadas por catálogos do sistema, é usada para encontrar a melhor forma de avaliar uma consulta.

• **Índice**: é uma estrutura de acesso rápido aos dados e está associado a uma ou mais colunas de uma tabela.

2.3.3 Catálogo

Nomeado também de tabelas do catálogo, dicionário de dados, catálogo do sistema, possui dados globais ao sistema, tais como o tamanho do conjunto dos referências e tamanho das páginas, assim como dados sobre as tabelas, índices e vistas:

Para cada tabela:

- Nome da tabela;
- Nome e estrutura do ficheiro onde está armazenada;
- Nome e tipo de cada atributo;
- Nome de cada índice da tabela;
- Restrições de integridade.

Para cada índice:

- Nome e estrutura do índice;
- Atributos da chave de pesquisa.

Para cada vista:

- Nome e definições.

Adicionalmente, contém informação acerca dos utilizadores e privilégios como ainda estatísticas sobre tabelas e índices atualizadas periodicamente, contendo:

- Cardinalidade da ~~tabela~~ tabela;
- Quantidade de páginas de cada tabela;
- Cardinalidade dos índices;
- Quantidade de páginas de cada índice;
- Quantidade de níveis não-folha de cada índice;
- Grau do índice.

2.3.4 Fases do processamento de consultas

Decomposição (análise e validação)

- Transforma consulta de alto nível em álgebra relacional;
- Verifica se esta conexão sintática e semanticamente.

Estágios de decomposição:

Análise

- Analisa sintaticamente com técnicas dos compiladores de linguagens de programação.

Normalização

- Converte a consulta para algo facilmente manipulável.

Análise semântica

- Rejeita consultas incorretamente formuladas ou contraditórias.

Simplificação

- Deteta e remove redundâncias, eliminando subexpressões comuns para a consulta sem mais fácil e eficiente. É visto ainda as restrições de acesso, definições de vistas e restrições de integridade.

Reestruturação de Consulta

- ~~Para~~ Para fornecer uma implementação mais eficiente.

Otimização

Objetivo de encontrar um bom plano de execução, normalmente representados em álgebra relacional. Para otimizar envolve 2 passos:

- Encontrar planos alternativos;
- Estimar o custo de cada plano encontrado e escolher o ~~melhor~~ de menor custo estimado.

Genção de Código

Execução de código

2.3.6 As 12 Regras de Codd

Conjunto de regras que um SGBD tem que obedecer para considerar-se como relacional.

1. Todos os dados, incluindo o próprio lecionário de dados, são representados numa só forma, em tabelas bidimensionais;
2. Cada elemento de dados fica bem determinado pela combinação do nome da tabela onde está guardado, valor da chave primária e respetiva coluna (atributo);

3. Valores nulos (Nulls) são suportados para representar informações não disponíveis ou não aplicável, independentemente da dominância dos respectivos atributos.

4. Os metadados são representados e acessados da mesma forma que os próprios dados;

5. Tem de existir pelo menos uma linguagem com as seguintes características:

- Manipular dados, com possibilidade de utilização interativa ou em programas de aplicação;
- Definição de dados;
- Definição de vistas;
- Definição de restrições de integridade;
- Definição de acessos (autorizações);
- Manipular transações.

6 - Numa vista, todos os dados atualizados modificados devem ver essas modificações traduzidas nas tabelas base;

7 - Capacidade de tratar uma tabela como se fosse uma simples operação, tanto em operações como de atualização;

8 - Alterações na organização física dos ficheiros de base de dados ou nos métodos de acesso a esses não devem afetar o nível conceptual;

9 - Alterações no nível conceptual, que não envolvam remoção de elementos, não devem afetar o nível externo;

10 - Restrições de integridade devem poder ser especificadas numa linguagem relacional, independentemente dos programas de aplicação, e armazenadas no catálogo;

11 - O facto de uma base de dados centralizar-se numa máquina ou distribuir-se por várias, não deve repercutir-se no nível da manipulação dos dados;

12 - Se existir no sistema uma linguagem de nível baixo-nível (tipo orientada a ~~base~~ registos, record-oriented), esta não deverá permitir ultrapassar as restrições de integridade e segurança.

Modelo de Dados

Modelagem Conceptual

- O objetivo aqui é criar um modelo de forma gráfica, sendo este chamado de Diagrama Entidade e Relacionamento (DER), que identifica todas as entidades e relacionamentos de uma forma global.

Aqui é evitado qualquer detalhamento específico do modelo de base de dados.

A sua principal finalidade é capturar os requisitos de informação e regras de negócio sob o ponto de vista do negócio.

No desenvolvimento de soluções é o primeiro modelo que deve ser desenvolvido.

É independente de Hardware ou Software, ou seja, não depende de nenhum tipo de servidor de base de dados (SQL).

Por tanto qualquer alteração no Software ou hardware, não terá efeito no nível conceptual.

Modelagem Lógica

- A modelagem lógica é necessária para compilar os requisitos de negócio e representar os requisitos como um modelo.

Esta principalmente associada à coleta de necessidades de negócios, e não ao design do banco de dados.

As informações não precisam ser coletadas.

Descreve como os dados são armazenados na base de dados e também os seus relacionamentos.

Os modelos lógicos basicamente determinam se todos os requisitos do negócio foram reunidos. Ele é revisado pelos desenvolvedores.

É mais complexo que o modelo conceptual em que os tipos de coluna são definidos.

Modelagem Física

- A modelagem física lida com o design da base de dados real com base nos requisitos reunidos durante a modelagem lógica da base de dados.

Todas as informações disponíveis são convertidas em modelos relacionais.

Durante a modelagem física, os objetos são definidos em um nível denominado nível de esquema.

A modelagem física depende do Software que já está a ser usado. (SQL).

	Modelo de Dados		
	Conceptual	Lógico	Físico
Normas das entidades	✓		
Relacionamentos	✓		
Atributos	✓		
Chaves Primárias		✓	
Chaves Estrangeiras		✓	
Nome das tabelas			✓
Nome das colunas			✓
Tipo de dados das colunas			✓

Modelo Bachman (1969)

- Vem da área de avaliação da linguagem dentro da aquisição de uma segunda língua e tenta estabelecer as diferentes dimensões da competência comunicativa. Neste modelo, competência organizacional e competência pragmática são distinguidas.

- Charles Bachman (11/12/1924 - 13/07/2017)

Modelo Entity - Relationship (1976)

- É uma maneira sistemática de descrever e definir um processo de negócio. O processo é modelado como componentes (entidades) que são ligadas umas as outras por relacionamentos que expressam as dependências e exigências entre elas.

- Peter Chen (31/01/1947 - Atual)

Modelo Information Engineering (1981)

- Desenvolve um sistema de informação integrado baseando no compartilhamento de dados comuns, com ênfase nas necessidades de suporte à decisão, bem como os requisitos de processamento de transações.

- Clive Finkelstein (1939 - 12/09/2021)

Modelo IDEF1X (1985)

- É uma metodologia para a modelagem da informação cujo propósito inclui oferecer meios para definir uma visão de dados independente de aplicação, que pode ser validada por usuários e transformada em um projeto físico de Base de dados.

- Mary Loomis

Thomas Bruce

Modelo Banken (1995)

- É usado para projetar ou depurar bases de dados relacionais nas áreas de engenharia de software ~~etc~~. Usam um conjunto definido de símbolos, tais como retângulos, diamantes, ovais e linhas de conexão para representar a interconectividade de entidades, relacionamentos e os seus atributos.

- Richard Banken (18/10/1948 - Atual.)

Modelo Object-Role Modelling (1995)

- É usado para modelar a semântica de um universo de discurso. Usa símbolos gráficos baseados na lógica de predicados de primeira ordem e na teoria dos conjuntos para permitir que o modelador crie uma definição de um universo abstrato do discurso.

- G.M. Nijssen (18/10/1938 - Atual.)

- Tenny Halpin (1950 - Atual.)

Modelo Unified Modelling Language (UML) (1997)

- É uma linguagem-padrão para a elaboração da estrutura de projetos de software. Basicamente, a UML permite que desenvolvedores visualizem os produtos dos seus trabalhos em diagramas padronizados.

- Booch (27/02/1955 - Atual.)

- Jacobson (02/09/1939 - Atual.)

- Rumbaugh (22/08/1947 - Atual.)