### **Resumo sobres conceitos de bases de dados**

### **Introdução aos Sistemas de Banco de Dados**

Um sistema de banco de dados é um sistema computadorizado que mantém registros, funcionando como um repositório para arquivos de dados. Ele permite operações como:

* Adição, busca, exclusão e alteração de dados.
* Gerenciamento de arquivos dentro do banco de dados.

Exemplo: o banco de dados ADEGA contém informações sobre vinhos, onde requisições como SELECT, INSERT, DELETE e UPDATE são feitas em SQL (Structured Query Language), uma linguagem padrão internacional usada para manipulação de dados.

### **Componentes de um Sistema de Banco de Dados**

O sistema é composto por quatro elementos principais:

1. **Dados:** armazenados no banco, podem ser simples (texto, números) ou complexos (fotos, mapas, gravações).
2. **Hardware:** varia desde computadores pequenos até grandes mainframes.
3. **Software:** o sistema gerenciador do banco de dados (DBMS).
4. **Usuários:** os indivíduos ou organizações que interagem com o sistema.

### **Conceitos Importantes**

* **Tabelas e Estrutura de Dados:** Os dados são organizados em tabelas. Colunas são chamadas de campos, e as linhas, registros. Em SQL, são chamadas de atributos, relações e tuplas.
* **Chaves Primárias:** Garantem que cada registro seja único (ex.: DEP# na tabela ADEGA).

### **Operações no Banco de Dados**

SQL é a linguagem usada para interagir com bancos de dados, com os principais comandos sendo:

* **SELECT:** Busca de dados.
* **INSERT:** Inserção de novos registros.
* **DELETE:** Exclusão de registros existentes.
* **UPDATE:** Alteração de registros.

Essas operações permitem a manipulação eficiente das informações armazenadas.

### **Observações sobre SQL**

1. É amplamente adotado comercialmente e essencial para bancos de dados modernos.
2. Utiliza uma estrutura padrão que facilita a manipulação de dados complexos.
3. Oferece versatilidade, com suporte a dados simples e avançados como imagens e áudios.

### **Finalidade dos Bancos de Dados**

Um banco de dados existe para armazenar e organizar informações de forma acessível, auxiliando indivíduos e organizações em suas atividades. Ele transforma dados em informações significativas.

### **Vantagens da Abordagem de Banco de Dados**

1. **Compartilhamento de Dados:** Aplicações existentes e novas podem compartilhar informações sem a necessidade de duplicação.
2. **Redução de Redundância:** Combina dados compartilhados para evitar duplicidade e economizar espaço. Quando redundância é necessária, deve ser controlada (propagação de atualizações).
3. **Prevenção de Inconsistências:** A centralização reduz discrepâncias entre entradas duplicadas, garantindo dados corretos e coerentes.
4. **Suporte a Transações:** Garante que operações relacionadas sejam realizadas por completo ou revertidas em casos de falha (ex.: atomicidade em transferências bancárias).
5. **Manutenção de Integridade:** Restrições de integridade evitam erros como dados incorretos ou incoerentes e protegem usuários que compartilham informações.
6. **Reforço de Segurança:** DBA controla acesso aos dados, garantindo segurança para informações confidenciais e prevenindo riscos associados à centralização.
7. **Equilíbrio de Requisitos Contraditórios:** O DBA estrutura o sistema para atender às prioridades da empresa sem comprometer aplicações menos críticas.
8. **Imposição de Padrões:** Garante consistência na representação, nomenclatura e documentação de dados, facilitando compartilhamento e migração.

### **Independência de Dados**

* **Física:** Refere-se à habilidade de alterar o armazenamento físico dos dados sem impactar o acesso lógico pelos usuários. Isso aumenta a flexibilidade no gerenciamento dos bancos de dados.
* A independência de dados é fundamental para manter a adaptabilidade e eficiência do sistema.

### **1. Nível Interno (Físico):**

Este é o nível mais baixo da arquitetura e está relacionado ao armazenamento real dos dados. Aqui, você está lidando com a forma como os dados são armazenados no hardware (discos, memória, etc.), como são organizados e acessados.

**Detalhamento:**

* O nível **interno** descreve a **estrutura física** do banco de dados. Ou seja, como os registros de dados, índices e outros objetos são organizados fisicamente. A ideia principal desse nível é otimizar o desempenho de acesso e armazenamento.
* No seu exemplo, o tipo de registro **EMP\_ARMAZENADO** (com 20 bytes de comprimento) é um **registro físico** que contém dados brutos e prefixos de controle (como ponteiros e flags), usados para garantir que o banco de dados funcione corretamente.
* **Índice EMPX**: É uma estrutura adicional usada para otimizar a busca de registros. O índice mapeia rapidamente os números de empregado (EMP#) para os registros de dados armazenados, sem a necessidade de percorrer toda a tabela. Isso acelera operações de busca, como encontrar um empregado com um número específico.

### **2. Nível Conceitual (Lógico):**

Este é o nível intermediário entre os usuários e o armazenamento físico. Ele descreve o banco de dados de forma lógica e unificada, sem se preocupar com detalhes de armazenamento. A visão conceitual é projetada para refletir as necessidades de todos os usuários do banco de dados, sem se concentrar em implementações específicas ou nos detalhes físicos.

**Detalhamento:**

* O nível **conceitual** fornece uma **representação global** de todo o banco de dados, incluindo todos os dados e suas inter-relações, sem focar em como ou onde eles são armazenados fisicamente.
* Aqui, o banco de dados é representado por entidades como **empregados**, **departamentos**, **fornecedores** e **peças**. O relacionamento entre esses dados, como um empregado trabalhando em um departamento, ou um fornecedor fornecendo peças, é descrito nesse nível.
* O exemplo mencionado fala de uma visão conceitual em que os dados dos **empregados** são representados por campos como **NÚMERO\_EMPREGADO**, **NÚMERO\_DEPARTAMENTO** e **SALÁRIO**. No entanto, as especificidades de como esses dados estão armazenados (como o índice ou a estrutura física do registro) não são descritas no nível conceitual.

O **esquema conceitual** não deve incluir qualquer informação sobre **detalhes físicos de armazenamento**, como o tipo de índice ou a maneira como os dados são acessados. Ele descreve apenas os dados de forma lógica, como os registros de "empregado" e "departamento", sem vincular isso a como ou onde essas informações são realmente armazenadas.

### **3. Nível Externo (Visão do Usuário):**

Este nível descreve a forma como diferentes usuários interagem com o banco de dados. Cada usuário pode ter uma **visão personalizada** de acordo com suas necessidades, e essas visões não precisam refletir o banco de dados completo.

**Detalhamento:**

* No nível **externo**, o banco de dados é **apresentado** de maneira personalizada para cada usuário ou aplicação, de modo que o usuário veja apenas os dados relevantes para ele. Um usuário pode, por exemplo, ver apenas dados sobre empregados ou departamentos, mas não precisa saber sobre fornecedores ou peças.
* A **visão externa** é uma forma de "abstract" do banco de dados, ou seja, uma **representação adaptada** das informações no banco de dados, com base nas permissões e necessidades de um usuário específico.
* Por exemplo, no seu caso, um usuário PL/I pode estar interessado apenas no **número de empregado** e **salário**, enquanto outro usuário, no Departamento de Compras, pode estar mais interessado nos **fornecedores** e **peças**. Essa **visão externa** é implementada usando o **esquema externo**, que é uma **definição da visão personalizada** de cada usuário. O esquema externo é escrito usando a **DDL externa**, uma sublinguagem de dados que permite criar essas visões específicas.

A visão externa é importante porque ela garante que o usuário só tenha acesso aos dados relevantes e ocultos os detalhes desnecessários ou confidenciais.

### **Mapeamento Entre os Níveis:**

O mapeamento entre os três níveis é essencial para garantir que o banco de dados funcione de forma eficiente e consistente, independentemente de como os dados são armazenados ou como cada usuário os vê. A seguir, detalho esse mapeamento:

1. **Esquema Conceitual e Esquemas Externos**: O mapeamento entre o **esquema conceitual** (que representa o banco de dados de maneira geral) e os **esquemas externos** (que representam a visão do banco de dados para cada usuário) é feito por um mecanismo chamado de **mapeamento de visões externas para o modelo conceitual**. Isso garante que as visões externas estejam corretas em relação à visão global dos dados.
2. **Esquema Conceitual e Esquema Interno**: O mapeamento entre o **esquema conceitual** e o **esquema interno** é o mapeamento de dados entre o modelo lógico (conceitual) e o modelo físico (interno). Isso ocorre quando as definições de dados conceituais (como atributos de registros de empregados) são mapeadas para uma estrutura física de armazenamento (como registros de dados com prefixos de controle, campos e índices). Esse mapeamento é geralmente feito pelo SGBD de forma transparente para os usuários.

### **Sublinguagens de Dados (DSL):**

* **DDL (Data Definition Language)**: No seu exemplo, a **DDL externa** e a **DDL conceitual** são responsáveis por definir as estruturas de dados e as visões de usuários. A DDL externa descreve como os dados serão vistos por cada usuário, enquanto a DDL conceitual descreve as definições lógicas do banco de dados sem se preocupar com detalhes de armazenamento físico.
* **DML (Data Manipulation Language)**: A **DML** descreve as operações de manipulação de dados, como inserções, atualizações e consultas. Em PL/I, essas operações podem ser feitas usando instruções que interagem com o SGBD para modificar ou consultar os dados armazenados.

### **PL/I:**

PL/I (Programming Language One) é uma linguagem de programação desenvolvida na década de 1960 pela IBM. Ela foi projetada para ser versátil, combinando características de linguagens como COBOL (voltada para aplicações empresariais) e Fortran (utilizada em cálculos científicos). Por isso, a PL/I é bem adequada tanto para tarefas administrativas quanto para cálculos científicos complexos.

Algumas das principais características da PL/I incluem:

- Multifuncionalidade: Pode ser usada para manipulação de dados comerciais e científicos.

- Suporte a processamento paralelo:\*\* Oferece construções para facilitar a execução simultânea de processos.

- Flexibilidade:Admite tipos de dados robustos e uma ampla gama de instruções de controle.

A linguagem **PL/I** serve como uma linguagem de programação para a interação com o banco de dados. A PL/I pode ser usada para:

* **Declarar tipos de dados** (no contexto da DDL).
* **Manipular dados** (usando a DML, por exemplo, invocando funções ou procedimentos do SGBD).
* **Interagir com o banco de dados** de forma eficiente, utilizando tanto a parte conceitual quanto a externa de forma abstrata.

### **Visão Conceitual e Esquema Conceitual**

A visão **conceitual** é a representação global do banco de dados, englobando todos os dados e suas inter-relações, mas sem se preocupar com o formato de armazenamento físico. Ela reflete como os dados são organizados de maneira lógica e como as informações fluem dentro de um sistema ou empresa.

#### **Esquema Conceitual:**

O **esquema conceitual** é a definição dessa visão. Ele descreve todos os dados que o banco de dados irá conter e as relações entre esses dados, mas de forma mais abstrata. Embora seja uma representação lógica, ela não é simples como as definições de um programa COBOL. O esquema conceitual vai muito além da mera definição de campos e registros: ele pode incluir **restrições de segurança**, **restrições de integridade** e até as **regras de negócio** que regem como os dados podem ser usados e manipulados dentro da organização.

Em muitos sistemas atuais, o esquema conceitual é basicamente uma junção de esquemas externos (visões personalizadas de diferentes usuários) junto com as restrições de integridade e segurança. No entanto, idealmente, ele deveria ser uma descrição mais completa de toda a empresa, incluindo a forma como os dados são utilizados, controlados e auditados. Isso é algo que, no futuro, pode se tornar mais sofisticado, mas que ainda não é amplamente adotado.

### **Visão Interna (Nível Físico)**

O **nível interno** está relacionado à **representação física** do banco de dados. Enquanto o nível conceitual descreve os dados de forma lógica e sem considerar o armazenamento físico, o nível interno está preocupado com a forma como os dados são realmente armazenados e acessados no dispositivo físico.

#### **Visão Interna:**

* A **visão interna** é uma representação de baixo nível do banco de dados, onde se especifica como os dados são armazenados fisicamente no sistema (por exemplo, como os registros são organizados e quais índices são usados para otimizar o acesso).
* A **visão interna** está **separada do nível físico** real, ou seja, ela não trata de detalhes específicos do armazenamento como **tamanho de cilindros**, **trilhas** ou qualquer outra característica de dispositivos de armazenamento. Ela vê os dados como se estivessem armazenados em um espaço de endereçamento linear infinito.
* Ela é descrita por meio do **esquema interno**, que não apenas define os tipos de registros armazenados, mas também determina como esses registros estão organizados fisicamente, a ordem dos registros, como os índices são usados e outros aspectos da implementação física.

### **1. Definir o Esquema Conceitual**

Uma das primeiras responsabilidades do DBA é criar o **esquema conceitual**, que é a implementação técnica das decisões do **administrador de dados (DA)**. O DA, como mencionado, decide **quais dados** são necessários para a empresa e define a **estrutura lógica** do banco de dados, enquanto o DBA converte essas decisões em um esquema executável no sistema, utilizando a **DDL conceitual**.

* **Esquema Conceitual**: Refere-se à estrutura lógica do banco de dados, identificando as entidades e os dados que devem ser armazenados.
* **Modelo de Dados**: O DBA usa a DDL para criar essa estrutura no banco de dados, garantindo que os dados estejam organizados de forma eficaz e que o sistema possa fornecer as respostas corretas de acordo com as necessidades da empresa.

### **2. Definir o Esquema Interno**

O **esquema interno** é a tradução do esquema conceitual para a forma física de armazenamento no sistema. O DBA é responsável por garantir que os dados sejam armazenados de maneira eficiente e que o banco de dados tenha bom desempenho. Isso envolve o **projeto físico**, que é uma parte do planejamento do banco de dados, relacionado à maneira como os dados serão fisicamente organizados e indexados.

* **Esquema Interno**: Descreve como os dados são fisicamente armazenados no sistema, incluindo detalhes como a forma dos registros e os índices.
* **Mapeamento Conceitual/Interno**: É uma definição crítica que liga o nível conceitual ao nível interno, garantindo que mudanças no armazenamento não afetem a visão conceitual do banco de dados.

### **3. Contacto com os Usuários**

O DBA tem o papel crucial de ser o **ponto de contato entre o sistema e os usuários**. Ele precisa garantir que os dados necessários estejam disponíveis de forma adequada para as operações diárias e, além disso, ele também ajuda os usuários a desenvolver as **visões externas** e os mapeamentos correspondentes.

* **Mapeamentos Externos/Conceituais**: O DBA deve definir como os dados serão vistos por diferentes usuários, criando esquemas personalizados para diferentes grupos dentro da empresa.
* **Consultoria e Treinamento**: O DBA também deve ajudar os usuários no desenvolvimento de aplicativos, treinando-os e resolvendo problemas de banco de dados.

### **4. Definir Restrições de Segurança e Integridade**

Uma das funções do DBA é garantir que o banco de dados esteja protegido e que os dados mantidos nele sejam consistentes. O DBA define e implementa **restrições de segurança** e **restrições de integridade** para evitar acesso não autorizado e garantir que os dados sigam regras lógicas de consistência.

* **Segurança**: Garantir que apenas usuários autorizados possam acessar ou modificar os dados.
* **Integridade**: Garantir que os dados sejam consistentes, seguindo regras definidas para a validação de entradas no banco de dados.

### **5. Definir Normas de Descarga e Recarga**

Outro aspecto essencial da administração de banco de dados é o **controle de falhas e backup**. O DBA deve implementar um sistema de **descarga (dump)** e **recarga (restore)** de dados para garantir que, caso ocorra um erro ou falha de hardware, o banco de dados possa ser rapidamente restaurado a um estado consistente, minimizando o impacto para os usuários.

* **Backup e Recuperação**: O DBA deve planejar e implementar procedimentos regulares de backup, assegurando que a recuperação de dados seja rápida e eficiente.
* **Sistemas de Alta Disponibilidade**: Para grandes sistemas de banco de dados, como bancos de dados multiterabytes, a recuperação rápida e sem perda de dados torna-se ainda mais crítica.

### **6. Monitorar o Desempenho e Responder a Requisitos de Mudanças**

O DBA é responsável por garantir que o banco de dados esteja operando com o melhor **desempenho possível**. Isso envolve o processo de **tuning**, ou afinação do banco de dados, para otimizar a velocidade de acesso aos dados e garantir que ele atenda aos requisitos da empresa.

* **Tuning de Banco de Dados**: O DBA monitora a performance do banco de dados e ajusta a organização dos dados (por exemplo, reorganizando o banco de dados ou ajustando índices) para garantir que ele permaneça eficiente à medida que as demandas da empresa mudam.
* **Adaptação a Mudanças**: O DBA deve garantir que qualquer mudança na estrutura interna do banco de dados seja refletida nos mapeamentos conceituais, mantendo a **independência lógica de dados**.

### **Conceito e Processo Geral**

O SGBD é um software responsável por gerenciar o acesso e as operações no banco de dados. De forma resumida, ele segue estas etapas:

1. **Pedido de acesso:** O usuário utiliza uma sublinguagem de dados (exemplo: SQL) para fazer a solicitação.
2. **Análise do pedido:** O SGBD intercepta e analisa o pedido.
3. **Inspeção de esquemas e mapeamentos:** Ele consulta o esquema externo, o mapeamento externo/conceitual, o esquema conceitual, o mapeamento conceitual/interno e a definição do banco de dados armazenado.
4. **Execução:** Realiza as operações necessárias no banco de dados.

Para operações mais complexas, como buscar registros externos, o SGBD faz múltiplas conversões e construções em etapas: registros armazenados → registros conceituais → registros externos. Isso pode incluir conversões de tipo de dados.

### **Funções Principais do SGBD**

#### **1. Definição de Dados (Data Definition):**

* O SGBD aceita definições de dados (esquemas externos, conceituais, internos, mapeamentos) e converte-as em formatos apropriados.
* Inclui componentes como o **processador de DDL**, que interpreta e compila definições feitas em linguagens de definição de dados (DDLs).
* Exemplo: Para registros externos de "EMPREGADO", o SGBD "entende" campos como "SALÁRIO" e pode processar consultas como "obter empregados com salário < R$ 50.000,00".

#### **2. Manipulação de Dados (Data Manipulation):**

* Processa requisições de busca, atualização, exclusão ou inserção de dados usando a linguagem de manipulação de dados (DML).
* As requisições podem ser:
* **Planejadas:** Previstas antecipadamente e otimizadas pelo administrador de banco de dados (DBA).
* **Não Planejadas:** Consultas ad hoc feitas de forma interativa, sem planejamento prévio.
* Exemplos de uso incluem sistemas operacionais (requisições planejadas) e suporte à decisão (requisições não planejadas).

#### **3. Otimização e Execução:**

* Um otimizador determina a forma mais eficiente de executar as requisições.
* Após otimizar, o gerenciador em tempo de execução (run time) executa os pedidos, possivelmente chamando um gerenciador de arquivos para acessar os dados armazenados.

#### **4. Segurança e Integridade de Dados:**

* Monitora e bloqueia ações que violam restrições de segurança ou integridade definidas pelo DBA. Pode fazer isso tanto em tempo de compilação quanto de execução.

#### **5. Recuperação de Dados e Controle de Concorrência:**

* Garantia de consistência e recuperação em caso de falhas. Normalmente é gerenciado por um componente adicional chamado **gerenciador de transações**.

#### **6. Dicionário de Dados (Metadados):**

* Funciona como um banco de dados especial contendo informações sobre os dados (metadados), incluindo esquemas, mapeamentos e restrições.
* Permite consultas para saber, por exemplo, quais programas ou usuários serão afetados por alterações no sistema.

#### **7. Desempenho:**

* Todas as funções do SGBD devem ser executadas com eficiência, minimizando atrasos e sobrecargas.