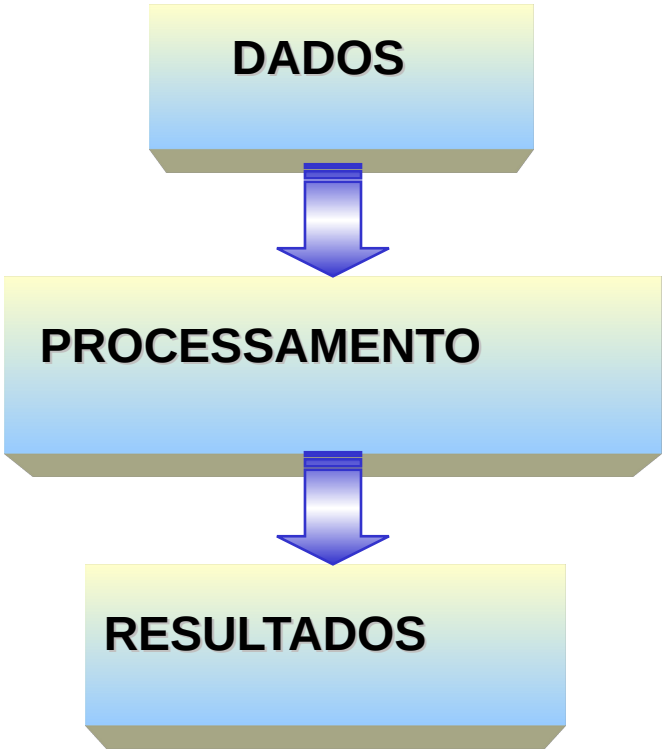

UNITAU
Universidade de Taubaté

Introdução

Introdução



No Princípio

- Sistemas operacionais monousuário e monotarefa
- Execução de programas em lote
- Pequena capacidade de processamento e de armazenamento
- Alto custo do hardware
- Inexistência ou grande dificuldade e custos para conexão entre computadores

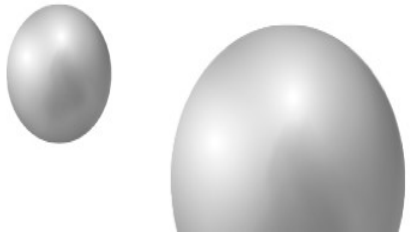
Conseqüências:

- Poucos computadores, quase sempre isolados
- Pequeno número de usuários
- Pouca ou nenhuma integração entre informações de diferentes aplicativos

No Princípio

No Princípio

- Falta de mecanismos que garantam a integridade física dos dados em caso de problemas com o aplicativo
- A consistência dos dados depende da lógica do aplicativo
- Dificuldade para permitir o acesso simultâneo aos dados por diferentes aplicativos ou mesmo várias instâncias do mesmo aplicativo
- Dependência da linguagem para acesso aos dados. Sendo frequentemente obrigatório o uso da mesma linguagem para desenvolver outros aplicativos que acessem os dados
- Falta de ferramentas para administração/manutenção dos dados



Atualmente

- Sistemas operacionais multiusuário e multitarefa, interativos
- Grande aumento da capacidade de processamento e armazenamento
- Enorme redução de custo do hardware
- Acesso à Internet com conexões de alta velocidade e disponibilidade praticamente constante
- Crescimento dos dispositivos móveis e da IoT

Atualmente

Conseqüências:

- Grande número de computadores conectados em redes de alta velocidade
- Aumento exponencial do volume e integração das informações
- Aumento da dependência da informação
- Crescimento do número de aplicações e integração entre as aplicações
- Multiplicação do número de usuários e meios de acesso às informações
- Integração da Informática a quase todos os momentos e aspectos da vida cotidiana

Banco de Dados

Um banco de dados, ou base de dados (BD), é uma coleção de dados interligados, representando informações sobre um domínio específico. É uma coleção lógica e coerente de dados com um significado inerente.

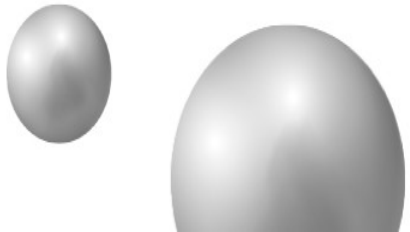
Normalmente é um conjunto de informações com uma estrutura regular, armazenada em um computador.

Sistema Gerenciador de Banco de Dados

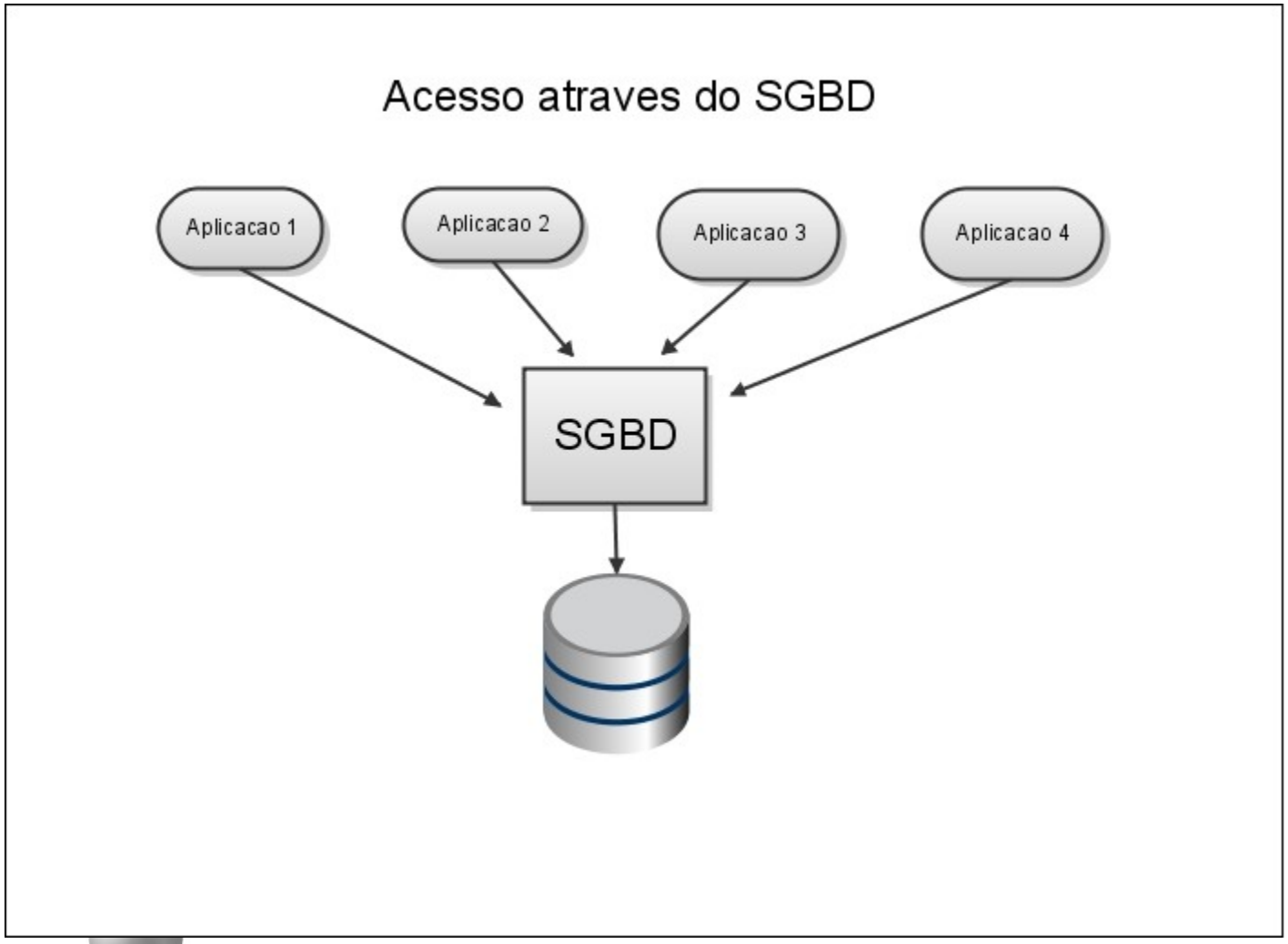
Um Sistema Gerenciador de Banco de Dados (SGBD) ou em inglês Database Management System (DBMS) é um conjunto de programas responsáveis pelo gerenciamento de bases de dados. Permite ao usuário definir, construir e manipular bases de dados para as mais diversas finalidades.

Objetivos de um SGBD:

- **Prover um ambiente adequado, seguro e eficiente para recuperar e armazenar informações da base de dados**
- **Prover os usuários com uma visão abstrata dos dados**



Sistema Gerenciador de Banco de Dados



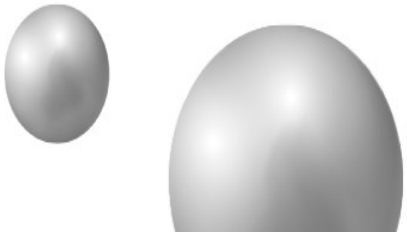
SGBD x GA

Algumas características são necessárias para que um sistema de manipulação de dados seja considerado um SGBD. Caso não possua alguma das características a seguir, o mesmo não pode ser classificado como um SGBD mas apenas como um Gerenciador de Arquivos (GA).

Auto-Contenção

Um SGBD não contém apenas os dados em si, mas armazena completamente toda a descrição dos dados, seus relacionamentos e formas de acesso. Em um GA, em algum momento ao menos, os programas aplicativos declaram estruturas, ou geram os relacionamentos entre os arquivos. Quando se define a forma do registro dentro do próprio programa, não está se lidando com um SGBD.

As informações sobre uma base de dados são chamadas de metadados e as tabelas que armazenam os metadados no SGBD formam o Dicionário de Dados do SGBD.



Independência dos Dados

As aplicações devem ser completamente imunes a alterações na estrutura de armazenamento ou estratégia de acesso aos dados. A criação de um novo método de acesso ou um índice não deve causar mudanças no código da aplicação.

Abstração dos Dados

Um SGBD fornece ao usuário somente uma representação conceitual dos dados, o que não inclui maiores detalhes sobre sua forma de armazenamento real. O chamado Modelo de Dados é um tipo de abstração utilizada para fornecer esta representação conceitual. Neste modelo, um esquema das tabelas, seus relacionamentos e suas chaves de acesso é exibido ao usuário, porém nada é afirmado sobre a criação dos índices, ou como serão mantidos, ou então quais as relações existentes entre as tabelas.

Visões

Um SGBD deve permitir que cada usuário visualize os dados de forma diferente daquela existente previamente no BD. Uma visão consiste de um subconjunto de dados, necessariamente derivados dos existentes no BD, mas que podem não estar explicitamente armazenados no BD. Portanto, uma replicação de uma estrutura, para fins de acesso de forma diferenciada por outros aplicativos, não caracteriza o uso de um SGBD.

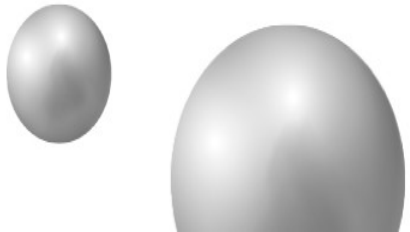
Transações

Um SGBD deve gerenciar completamente a integridade referencial definida em seu esquema, sem precisar em tempo algum, do auxílio do programa aplicativo. Desta forma exige-se que o banco de dados tenha ao menos uma instrução que permita a gravação de uma série modificações simultâneas e uma instrução capaz de cancelar um série modificações.

Por exemplo, imaginemos que estejamos cadastrando um pedido para um cliente, que este deseje reservar 5 itens de nosso estoque, que estão disponíveis e portanto são reservados, porém existe um bloqueio financeiro (duplicatas em atraso) que impede a venda. A transação deverá ser desfeita com apenas uma instrução ao Banco de Dados, sem qualquer modificações suplementares nos dados. Qualquer acesso complementar para a correção da reserva não caracteriza a utilização de um SGBD.

Acesso Automático

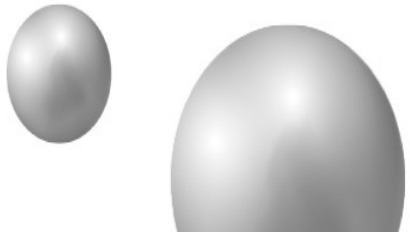
Um SGBD deve gerenciar o acesso aos dados com o intuito de evitar DEADLOCKS sem a necessidade de auxílio da aplicação. Um exemplo de DEADLOCK é, um usuário A está esperando a liberação de um registro que está sendo bloqueado por um usuário B e usuário B espera a liberação de um registro que está sendo bloqueado pelo usuário A, esse bloqueio mútuo é chamado de “abraço mortal” pois tanto o usuários A quanto B não irá conseguir acessar o registro bloqueado pelo outro e não vai liberar o registro que mantém bloqueado até conseguir acessar o outro registro.



Características Gerais de um SGBD

Controle de Redundâncias – A redundância consiste no armazenamento de uma mesma informação em locais diferentes, provocando inconsistências. Em um SGBD as informações só se encontram armazenadas em um único local, não existindo duplicação descontrolada dos dados. Quando existem replicações dos dados, estas são decorrentes do processo de armazenagem típica do ambiente Cliente-Servidor, totalmente sob controle do Banco de Dados.

Compartilhamento dos Dados – O SGBD deve incluir controle de concorrência ao acesso dos dados, garantindo em qualquer tipo de situação a escrita/leitura de dados sem erros.

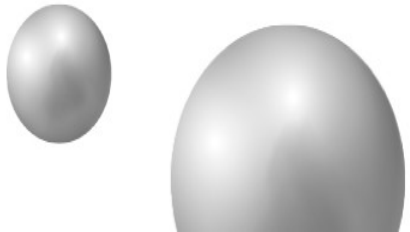


Características Gerais de um SGBD

Controle de Acesso – O SGDB deve dispor de recursos que possibilitem selecionar a autoridade de cada usuário. Assim, enquanto um usuário poderá realizar qualquer tipo de acesso, outros poderão ler alguns dados e atualizar outros e outros ainda poderão somente acessar um conjunto restrito de dados para escrita e leitura.

Interfaceamento – Um SGBD deverá disponibilizar formas de acesso gráfico, em linguagem natural, em SQL ou ainda via menus de acesso, não sendo uma "caixa-preta" somente sendo passível de ser acessada por aplicações.

Esquematização – Um SGBD deverá fornecer mecanismos que possibilitem a compreensão do relacionamento existentes entre as tabelas e de sua eventual manutenção.



Características Gerais de um SGBD

Controle de Integridade – Um SGBD deverá impedir que aplicações ou acessos pelas interfaces possam comprometer a integridade dos dados

Backups – O SGBD deverá apresentar facilidade para recuperar falhas de hardware e software, através da existência de arquivos de pré-imagem ou outros recursos automáticos, exigindo a mínima intervenção de pessoal técnico

Abstração de Dados

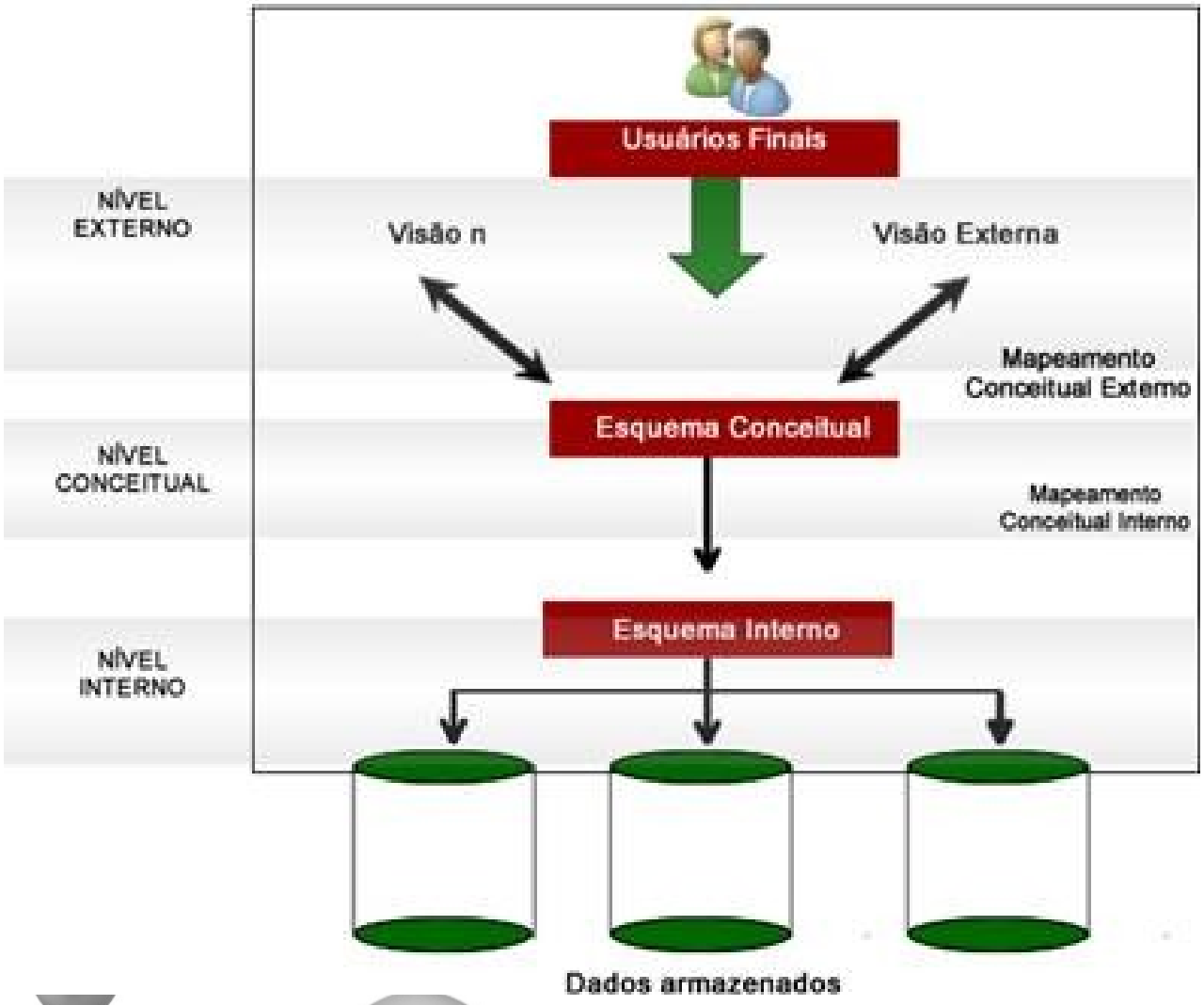
O SGBD deve prover uma visão abstrata de dados para os usuários, isolando, desta forma, detalhes mais internos do BD. A abstração se dá em três níveis:

Nível físico: também chamado de “Esquema interno”, é o nível mais baixo de abstração. Descreve como os dados estão realmente armazenados, englobando estruturas complexas de baixo nível.

Nível conceitual: (ou lógico): conhecido também como “Esquema Conceitual”, descreve quais os dados estão armazenados e seus relacionamentos. Neste nível, o BD é descrito através de estruturas relativamente simples, que podem envolver estruturas complexas no nível físico.

Nível de visões do usuário: é o nível externo, descrevendo partes do BD que serão visualizadas pelos usuários de acordo com suas necessidades.

Abstração de Dados



https://www.devmedia.com.br/arquitetura-de-um-sgbd/25007

Abstração de Dados

Esta arquitetura de três níveis, além de prover a abstração de dados, provê também a independência lógica e física dos dados.

A independência lógica é a capacidade de mudar o esquema conceitual sem a necessidade de modificar programas da aplicação e esquemas externos.

A independência física é a capacidade de mudar o esquema interno sem a necessidade de alterar os esquemas conceitual e externo. O acréscimo de uma informação num esquema conceitual é um exemplo de independência lógica e a reorganização física dos arquivos e a criação de estruturas de acesso adicionais são exemplos de independência física.

Transações

Para um SGBD, uma transação e um conjunto de operações que são tratadas como um bloco único e indivisível e também deve prover isolamento entre acessos concorrentes na mesma massa de dados.

Transações

Atomicidade – a transação é indivisível, todas suas operações devem ser executadas em caso de sucesso ou nenhuma alteração deve ser refletida na base de dados em caso de falha.

Consistência – a base de dados deve estar consistente antes do início da transação e após o final da transação. Se alguma alteração da transação violar a consistência da base de dados, toda a transação deve ser descartada e a base deve voltar a um estado consistente.

Isolamento – cada transação deve ser executada isolada das demais, uma transação não deve acessar ou alterar dados intermediários de outras transações concorrentes.

Durabilidade – após o final da transação, as alterações realizadas pela transação devem persistir mesmo em caso de falha.

Linguagens de Banco de Dados

Um sistema de banco de dados proporciona linguagens para definição dos esquemas do BD e para consultas e atualizações dos dados.

- **Linguagem de Definição de Dados: a DDL (Data Definition Language) é utilizada para definição da estrutura da base de dados, a DDL atua sobre o dicionário de dados da base de dados.**
- **Linguagem de Manipulação de Dados: a DML (Data Manipulation Language) é utilizada para recuperação, inclusão, alteração e deleção dos dados da base de dados. Pode ser procedural, que especifica como os dados devem ser obtidos do banco ou pode também ser declarativa (não procedural), em que os usuários não necessitam especificar como os dados serão obtidos.**

Linguagens de Banco de Dados

- **Linguagem de Controle de Dados: a DCL (Data Control Language) é utilizada para controlar o acesso aos dados da base de dados.**

Linguagens de Banco de Dados

Structured Query Language, ou Linguagem de Consulta Estruturada ou SQL, é uma linguagem de pesquisa declarativa para banco de dados relacional. Muitas das características originais do SQL foram inspiradas na álgebra relacional.

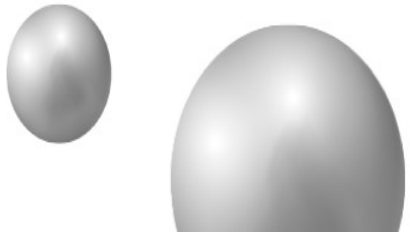
O SQL foi desenvolvido originalmente no início dos anos 70 nos laboratórios da IBM em San Jose, dentro do projeto System R, que tinha por objetivo demonstrar a viabilidade da implementação do modelo relacional proposto por E. F. Codd.

A linguagem SQL é um grande padrão de banco de dados. Isto decorre da sua simplicidade e facilidade de uso. Ela se diferencia de outras linguagens de consulta a banco de dados no sentido em que uma consulta SQL especifica a forma do resultado e não o caminho para chegar a ele.

Linguagens de Banco de Dados

Embora o SQL tenha sido originalmente criado pela IBM, rapidamente surgiram vários "dialetos" desenvolvidos por outros produtores. Essa expansão levou à necessidade de ser criado e adaptado um padrão para a linguagem. Esta tarefa foi realizada pela American National Standards Institute (ANSI) em 1986 e ISO em 1987.

O SQL foi revisto em 1992 e a esta versão foi dado o nome de SQL-92 ou SQL2. Foi revisto novamente em 1999 e 2003, 2006, 2008, 2011, 2016 e 2019 para se tornar SQL:1999 (SQL3), SQL:2003, SQL:2006, SQL:2008, SQL:2011, SQL:2016 e SQL:2019 respectivamente.

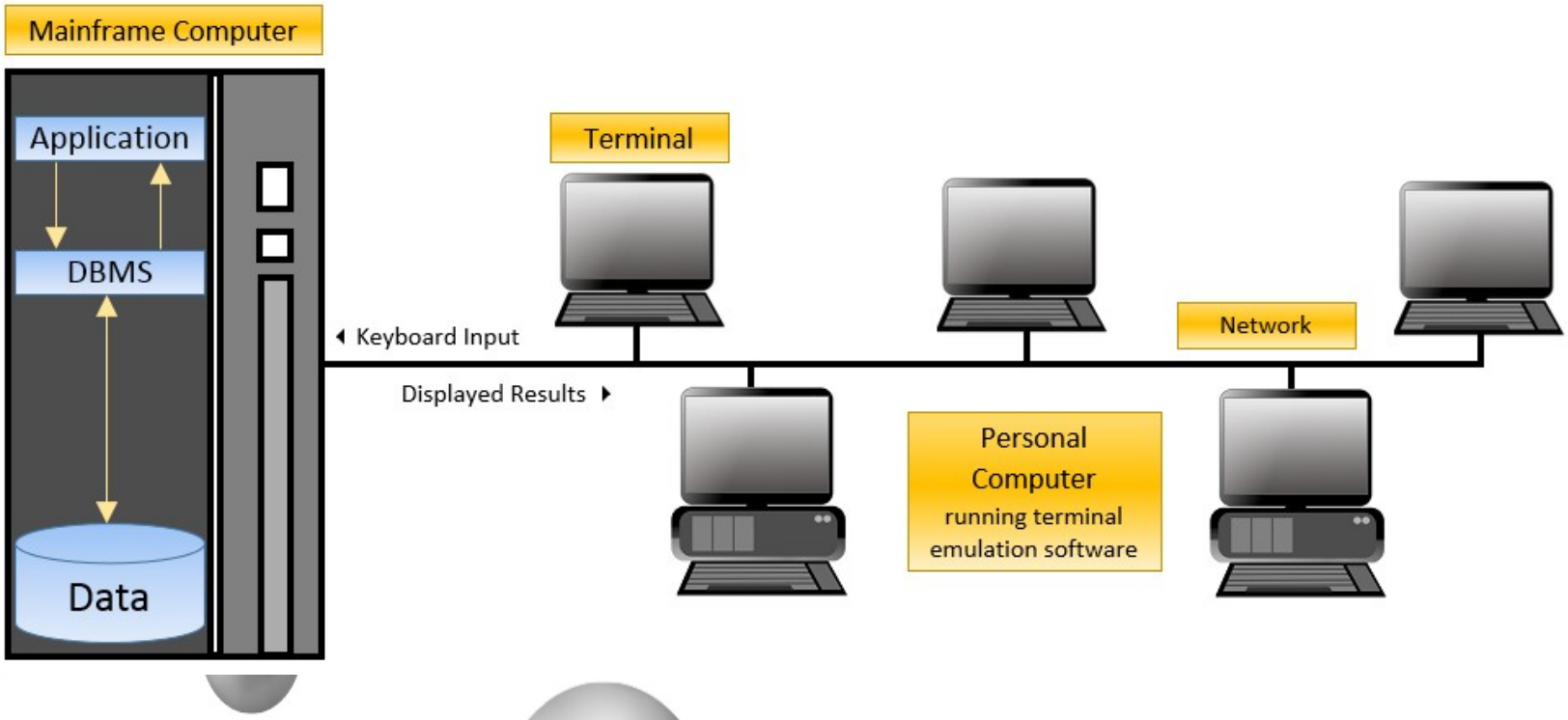


Arquitetura

A arquitetura dos SGBD tem acompanhado as tendências das arquiteturas dos sistemas computacionais. As principais arquiteturas para sistemas de banco de dados são:

Arquitetura Centralizada

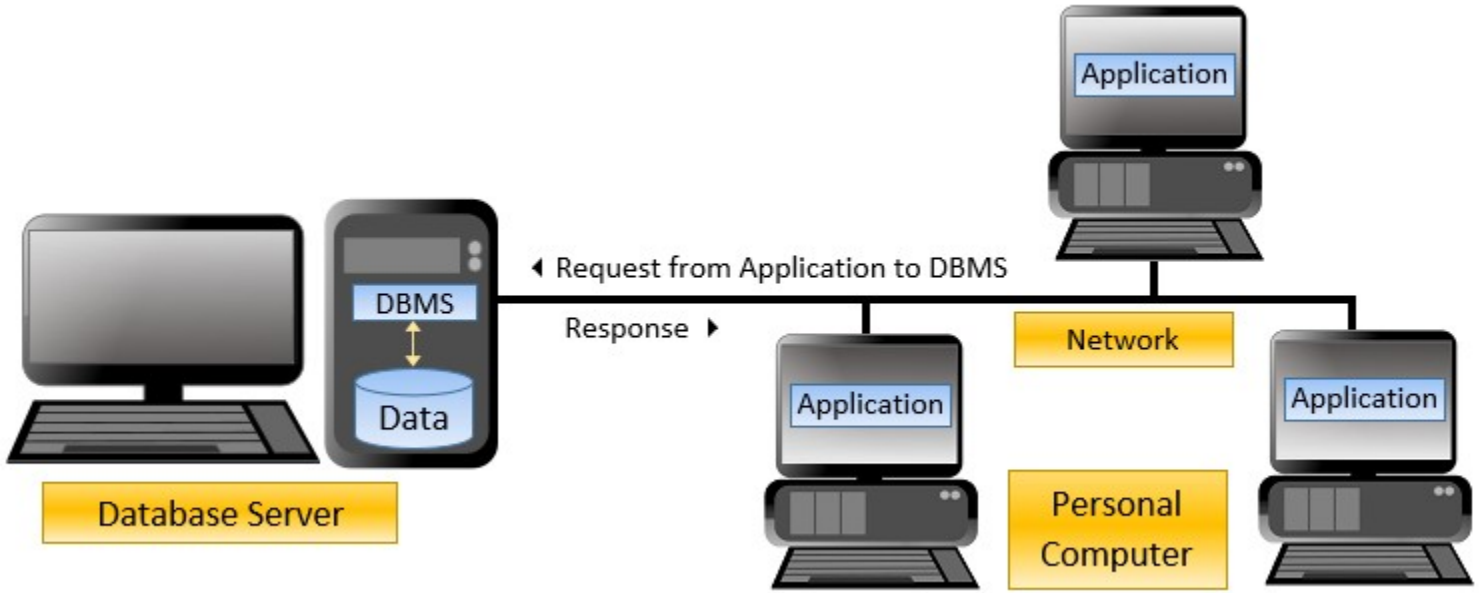
As primeiras arquiteturas utilizavam mainframes para executar todo o processamento principal, incluindo as aplicações, programas de interface com os usuários e o SGBD.



https://app.myeducator.com/reader/web/617b/chapter01/gj149/

Arquitetura Cliente/Servidor

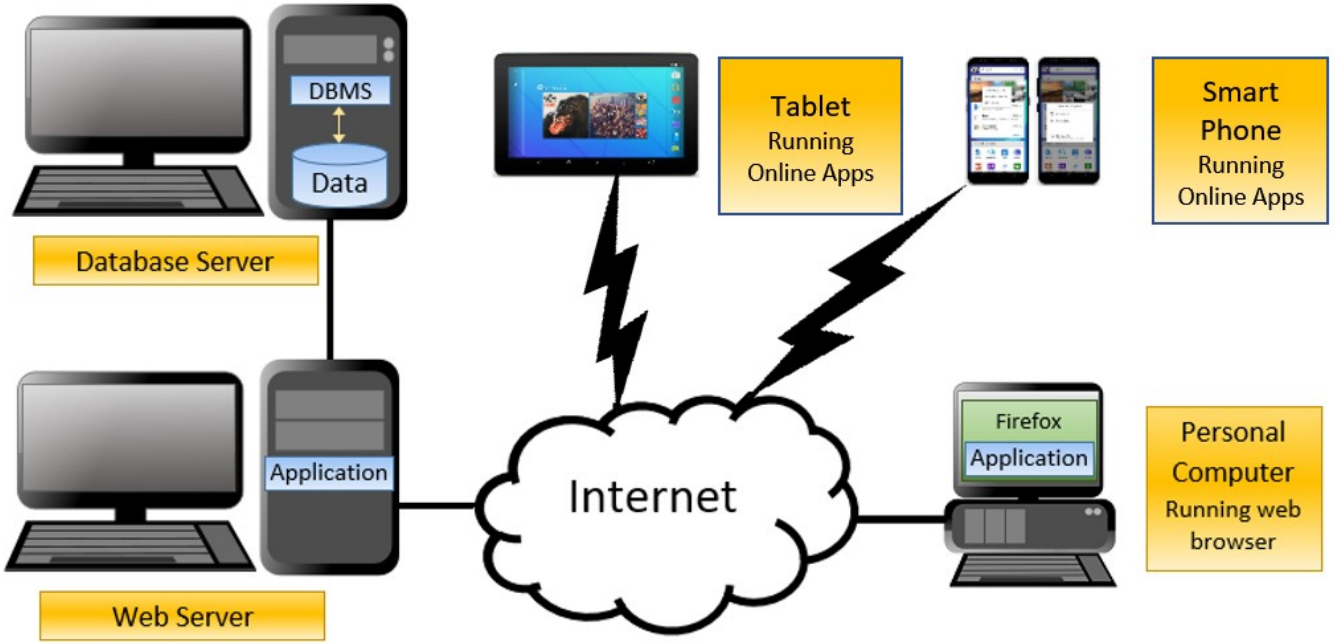
Com o aumento da capacidade de processamento dos microcomputadores e estações de trabalho e disseminação das redes, as aplicações passaram a ser executadas em maquinas clientes e o SGBD em um servidor que atende aos diversos clientes.



https://app.myeducator.com/reader/web/617b/chapter01/gj149/

Arquitetura Cliente/Servidor de 3 Camadas

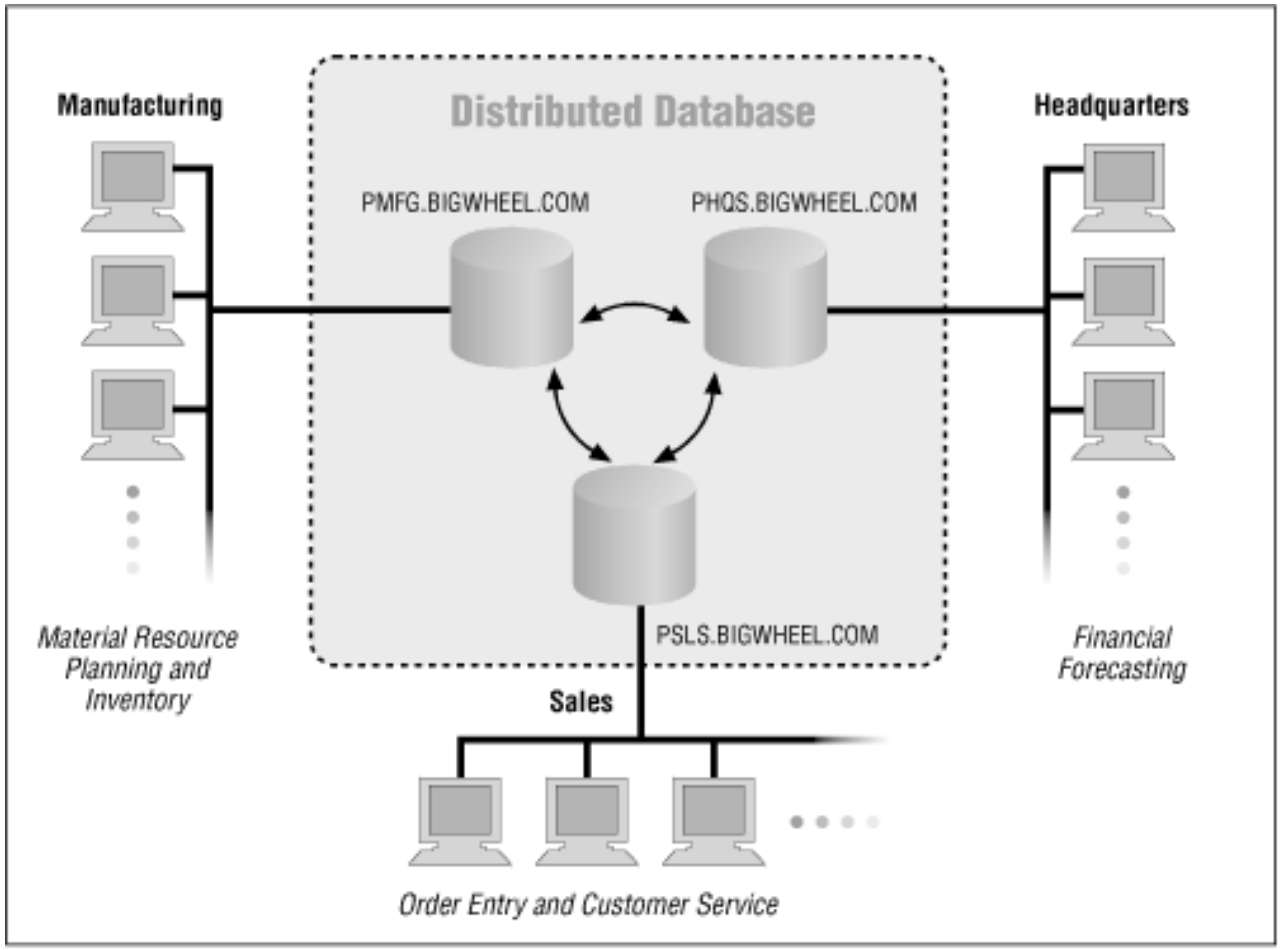
Principalmente com o crescimento da Internet e Intranets, a aplicação passou a ser dividida entre um servidor de aplicações que é responsável por parte da lógica da aplicação conhecida como regras do negócio e o equipamento do usuário que é responsável pela apresentação dos dados para o usuário.



https://app.myeducator.com/reader/web/617b/chapter01/gj149/

Arquitetura Distribuida

Para distribuir a carga de processamento do SGBD, o banco de dados passou a ser divido entre diversos servidores.

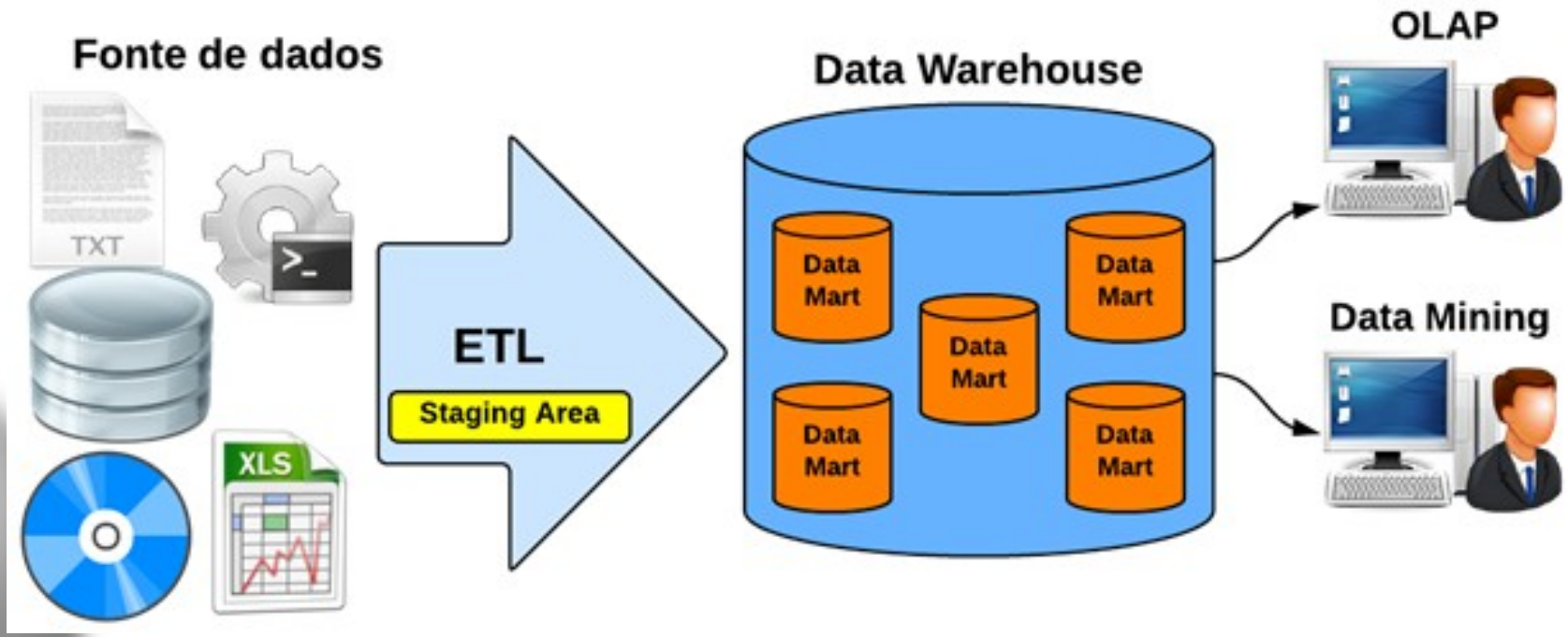


https://www.oreilly.com/library/view/oracle-distributed-systems/1565924320/ch01s02.html

Web Applications

Com o crescimento exponencial da Internet, da capacidade de armazenamento e de processamento, as aplicações tem se tornado cada vez mais diversificadas e complexas. Data Warehouse, Big Data, Cloud Computing, IoT e outros conceitos, trazem mudanças nas arquiteturas das aplicações que demandam novos produtos e novos modelos de banco de dados como bancos NoSQL, banco de dados massivos, cache de banco de dados e outras tecnologias.

Data Warehouse

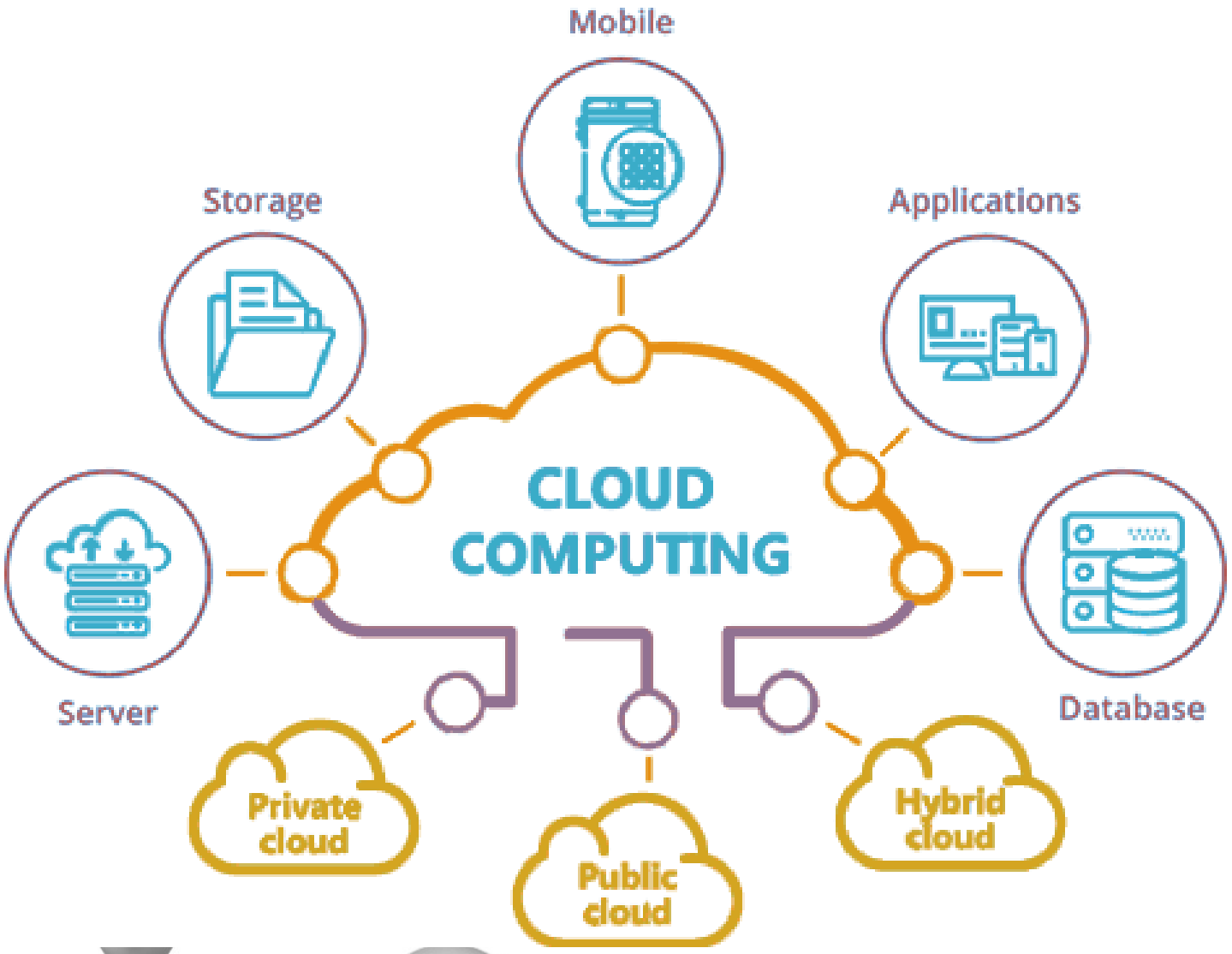


https://canaltech.com.br/business-intelligence/conhecendo-a-arquitetura-de-data-warehouse-19266/

Big Data

https://www.ibm.com/blogs/cloud-computing/tag/big-data-mining/

Cloud Computing

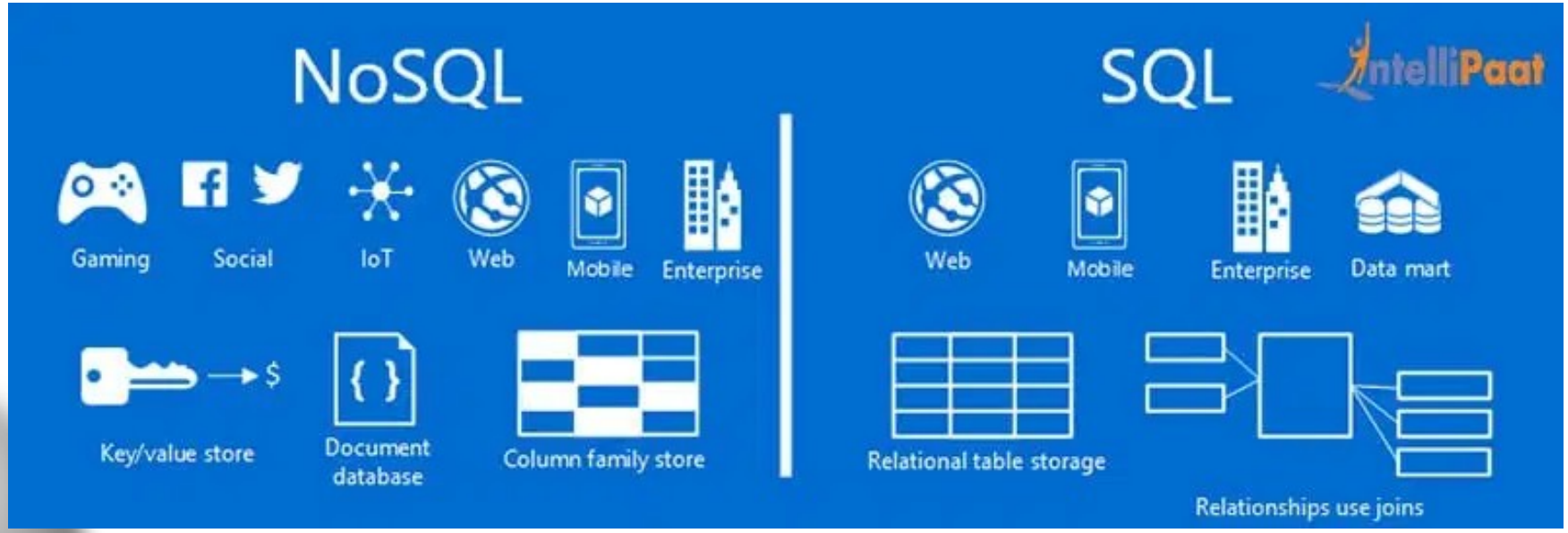


https://networkencyclopedia.com/cloud-computing/

IoT

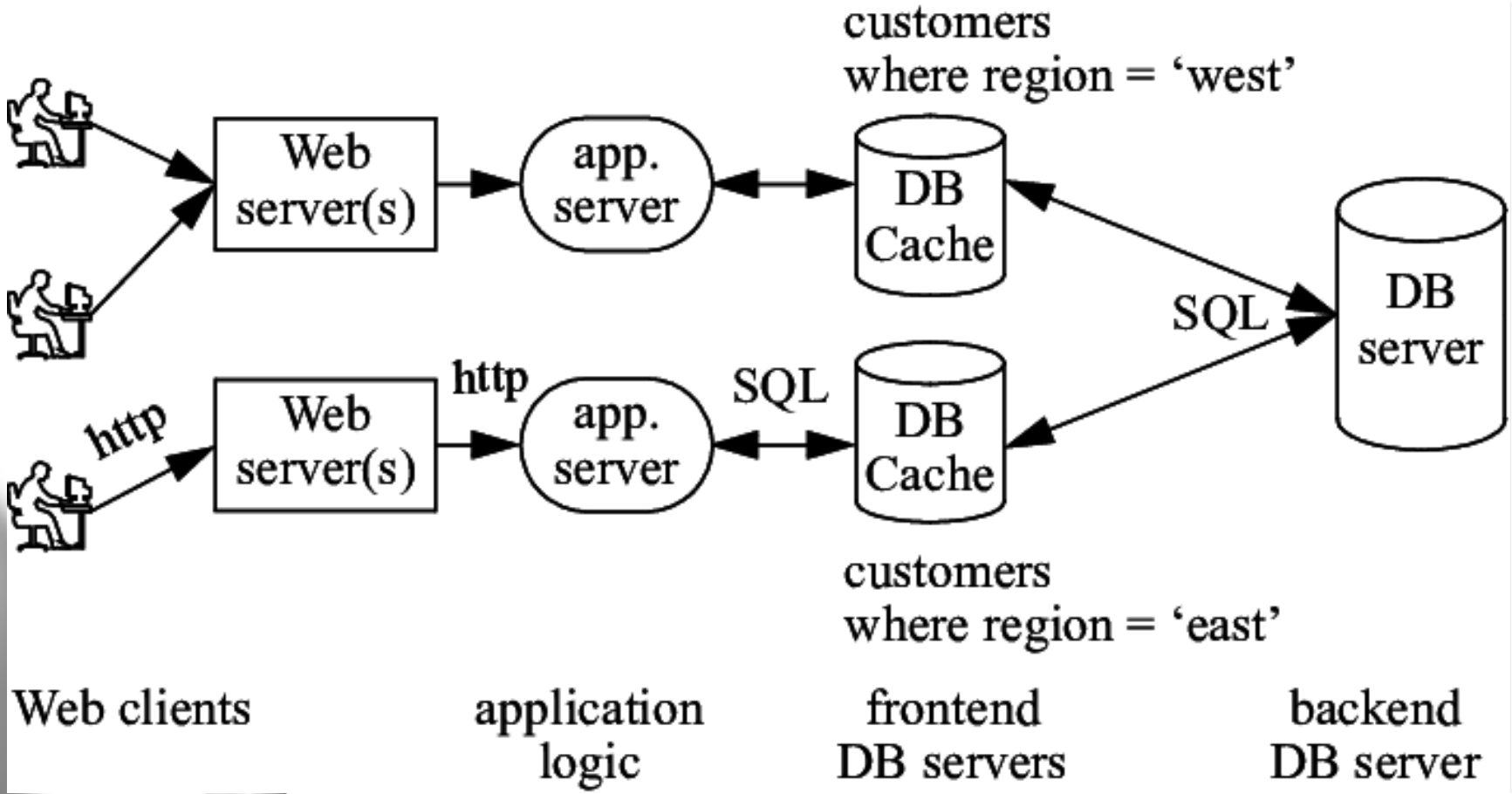
https://www.strate.education/gallery/news/iot-definition

NoSQL



https://softwareengineeringdaily.com/2018/12/14/cloud-database-performance-scaling-mongodb-atlas/

Database Caching



https://www.researchgate.net/figure/Database-caching-for-Web-applications_fig1_220473614