

SENAC - SERVIÇO NACIONAL DE APRENDIZAGEM COMERCIAL
UNIDADE EDUCACIONAL : CEP TALAL ABU-ALLAN – CEILÂNDIA

ALUNO: SÉRGIO KLANOVICH

PROJETO INTEGRADOR SENAC
IMPLEMENTAÇÃO DE BANCO DE DADOS PARA MERCADO

TRABALHO APRESENTADO NO CURSO DE
ADMINISTRADOR DE BANCO DE DADOS,
DA INSTITUIÇÃO SENAC.

ORIENTADOR: PROF. HUDSON NEVES E
SILVA

BRASÍLIA-DF

2024

INTRODUÇÃO

A implementação de um banco de dados em um mercado é fundamental devido à sua importância e relevância para várias áreas-chave do negócio. Aqui estão alguns pontos que destacam a importância de um banco de dados para um mercado:

Controle de Estoque e Inventário: Um banco de dados permite que o minimercado mantenha registros precisos de todos os produtos em estoque. Isso inclui informações detalhadas sobre quantidades disponíveis, datas de vencimento, fornecedores e histórico de movimentação de estoque. Esses dados são cruciais para evitar escassez ou excesso de produtos, garantindo que os clientes sempre encontrem o que precisam.

Gestão de Vendas e Transações: Com um banco de dados, o mercado pode registrar todas as transações de vendas de forma eficiente. Isso inclui detalhes como itens comprados, preços, horários de venda e métodos de pagamento. Esses dados são valiosos para análises de vendas, previsão de demanda, identificação de tendências de compra e elaboração de estratégias de marketing direcionadas.

Perfil do Cliente e Programas de Fidelidade: Um banco de dados permite ao minimercado manter registros detalhados sobre os clientes, incluindo histórico de compras, preferências de produtos e informações de contato. Esses dados são essenciais para a implementação de programas de fidelidade, personalização de ofertas e comunicação direcionada com os clientes.

Análise e Tomada de Decisão: Com acesso a dados precisos e atualizados, os gestores do minimercado podem realizar análises detalhadas sobre o desempenho do negócio. Isso inclui análises de lucratividade de produtos, identificação de padrões de compra, avaliação de eficácia de promoções e identificação de áreas de melhoria operacional. Essas análises embasam a tomada de decisões estratégicas para otimizar o desempenho e a rentabilidade do minimercado.

Eficiência Operacional: Um banco de dados bem projetado e gerenciado permite que o minimercado automatize processos operacionais, reduzindo a necessidade de trabalho manual e minimizando erros. Isso inclui tarefas como atualização de estoque, geração de relatórios, processamento de pedidos de compra e gerenciamento de inventário. A automação desses processos aumenta a eficiência operacional, reduz custos e melhora a experiência do cliente.

Em resumo, um banco de dados é essencial para um minimercado porque proporciona uma base sólida para gerenciar estoques, transações, clientes e operações de forma eficiente e eficaz. Ele permite que o minimercado tenha acesso a informações valiosas para tomar decisões informadas, impulsionar as vendas, aumentar a fidelidade do cliente e garantir o sucesso a longo prazo do negócio.

CONCEITO DE BANCO DE DADOS E SUA IMPORTÂNCIA

Um banco de dados é um sistema organizado de armazenamento de dados que permite a criação, consulta, atualização e gerenciamento eficiente das informações. Ele consiste em uma coleção estruturada de dados inter-relacionados, geralmente armazenados em tabelas, que podem ser acessados e manipulados por meio de consultas.

A importância de um banco de dados é significativa em muitos aspectos, especialmente nos ambientes empresariais. Aqui estão algumas razões que destacam sua importância:

Centralização de Dados: Um banco de dados oferece um local centralizado para armazenar todos os dados relevantes de uma organização. Isso elimina a necessidade de múltiplos sistemas de armazenamento dispersos e facilita o acesso às informações necessárias.

Consistência dos Dados: Ao armazenar dados em um banco de dados, é possível garantir a integridade e consistência dos dados. Isso é alcançado por meio de restrições de integridade, como chaves primárias, chaves estrangeiras e regras de validação, que ajudam a manter a precisão e confiabilidade dos dados.

Acesso Rápido e Eficiente: Com um banco de dados, os usuários podem acessar rapidamente as informações necessárias por meio de consultas. Índices podem ser usados para melhorar o desempenho das consultas, permitindo recuperações rápidas de dados, mesmo em conjuntos de dados grandes.

Segurança dos Dados: Os bancos de dados oferecem recursos robustos de segurança para proteger os dados contra acesso não autorizado, garantindo que apenas usuários autorizados tenham permissão para visualizar ou modificar as informações sensíveis.

Suporte a Transações: Bancos de dados transacionais suportam transações, garantindo que as operações de banco de dados sejam consistentes e confiáveis, mesmo em ambientes concorrentes. Isso é essencial para aplicativos onde a precisão dos dados é crítica, como sistemas de processamento de pedidos e sistemas de reservas.

Análise de Dados: Os bancos de dados fornecem uma base para análise de dados e geração de relatórios. Com ferramentas de consulta avançadas, é possível extrair insights valiosos dos dados armazenados, identificando padrões, tendências e oportunidades de negócios.

Escalabilidade: Os bancos de dados são escaláveis, o que significa que podem lidar com grandes volumes de dados e aumentar sua capacidade conforme necessário à medida que a organização cresce.

Em resumo, um banco de dados é essencial para organizar, armazenar e gerenciar eficientemente os dados de uma organização, proporcionando acesso rápido, consistente e seguro às informações críticas para o funcionamento do negócio.

SISTEMAS DE BANCO DE DADOS

Os sistemas de banco de dados são os softwares responsáveis por gerenciar e facilitar o acesso, armazenamento, manipulação e segurança dos dados em um ambiente computacional. Existem diferentes tipos de sistemas de banco de dados, cada um projetado para atender a diferentes necessidades e cenários de uso. Aqui estão alguns dos principais tipos de sistemas de banco de dados:

Banco de Dados Relacional (RDBMS):

Este é um dos tipos mais comuns de sistemas de banco de dados. Ele organiza os dados em tabelas relacionadas umas às outras por meio de chaves estrangeiras.

Utiliza a linguagem SQL (Structured Query Language) para realizar consultas e manipulações nos dados.

Exemplos populares incluem MySQL, PostgreSQL, Oracle Database, SQL Server e SQLite.

Banco de Dados NoSQL:

Este tipo de banco de dados é projetado para armazenar e recuperar grandes volumes de dados não estruturados ou semi-estruturados.

Não segue o modelo relacional tradicional e pode usar diferentes modelos de dados, como documentos, grafos, chave-valor e colunas.

Exemplos incluem MongoDB, Cassandra, Redis, Couchbase e Neo4j.

Banco de Dados Orientado a Objetos:

Neste tipo de banco de dados, os dados são armazenados na forma de objetos, que consistem em dados e métodos associados.

É especialmente útil para linguagens de programação orientadas a objetos, como Java e C++.

Exemplos incluem db4o e ObjectDB.

Banco de Dados In-Memory:

Estes bancos de dados armazenam os dados diretamente na memória principal do computador, em vez de em discos.

Isso proporciona acesso mais rápido aos dados, ideal para aplicativos que exigem baixa latência.

Exemplos incluem Redis, Memcached e Oracle TimesTen.

Banco de Dados de Grafos:

Este tipo de banco de dados é projetado para armazenar e consultar dados em forma de grafos, que consistem em nós e arestas.

É especialmente útil para modelar e analisar redes complexas, como redes sociais e sistemas de recomendação.

Exemplos incluem Neo4j, Amazon Neptune e ArangoDB.

Banco de Dados Espacial:

Este tipo de banco de dados é projetado para armazenar e consultar dados relacionados a objetos espaciais, como mapas e coordenadas geográficas.

É especialmente útil em aplicações de geolocalização e GIS (Sistemas de Informações Geográficas).

Exemplos incluem PostGIS e Oracle Spatial.

Esses são apenas alguns dos tipos mais comuns de sistemas de banco de dados. Cada tipo tem suas próprias características, vantagens e desvantagens, e a escolha do sistema de banco de dados adequado depende das necessidades específicas do aplicativo ou projeto.

SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS - SGBD

Trata-se de um software que permite aos usuários criar, acessar, atualizar e gerenciar um banco de dados. O SGBD atua como uma camada intermediária entre os usuários finais e os dados armazenados, facilitando a interação com o banco de dados por meio de consultas e comandos.

Os principais objetivos de um SGBD são garantir a integridade, segurança, eficiência e confiabilidade dos dados armazenados, além de fornecer mecanismos para realizar consultas complexas e manipulações nos dados de forma eficiente. Para alcançar esses objetivos, os SGBDs oferecem uma variedade de recursos e funcionalidades, incluindo:

Modelagem de Dados: Os SGBDs permitem aos usuários definir a estrutura e o esquema do banco de dados, incluindo a criação de tabelas, definição de tipos de dados, restrições de integridade e relacionamentos entre tabelas.

Linguagem de Consulta: A maioria dos SGBDs suporta uma linguagem de consulta padronizada, como SQL (Structured Query Language), que permite aos usuários realizar consultas complexas nos dados armazenados, recuperando informações específicas conforme necessário.

Gerenciamento de Transações: Os SGBDs oferecem suporte a transações, permitindo que os usuários executem operações de banco de dados de forma consistente e confiável, mesmo em ambientes concorrentes. Isso inclui operações como commit (confirmação) e rollback (desfazer).

Segurança e Controle de Acesso: Os SGBDs fornecem recursos robustos de segurança para proteger os dados contra acesso não autorizado. Isso inclui autenticação de usuários, controle de acesso baseado em funções e permissões, criptografia de dados e auditoria de atividades.

Otimização de Consultas: Os SGBDs utilizam técnicas de otimização de consultas para melhorar o desempenho das consultas, como a seleção de índices apropriados, uso de planos de execução eficientes e cache de resultados.

Backup e Recuperação: Os SGBDs oferecem recursos para realizar backups regulares dos dados e restaurá-los em caso de falhas ou perda de dados.

Alguns exemplos populares de SGBDs incluem MySQL, PostgreSQL, Oracle Database, SQL Server, SQLite, MongoDB e Redis. A escolha do SGBD adequado depende das necessidades específicas do projeto, como volume de dados, requisitos de desempenho, modelo de dados e recursos necessários.

MODELOS DE BANCO DE DADOS

Os modelos de banco de dados descrevem a estrutura lógica e as regras para representar os dados armazenados em um banco de dados. Existem vários modelos de banco de dados, cada um com suas próprias características e formas de organizar os dados. Aqui estão alguns dos modelos mais comuns:

Modelo Hierárquico:

Neste modelo, os dados são organizados em uma estrutura hierárquica de árvore, com um único "nó raiz" conectando vários "nós filhos".

Cada nó pode ter vários nós filhos, mas apenas um nó pai.

É amplamente utilizado em sistemas legados, mas tem limitações em termos de flexibilidade e complexidade.

Modelo de Rede:

Este modelo estende o modelo hierárquico, permitindo que um nó tenha múltiplos nós pais, criando uma estrutura de rede.

Ele introduz o conceito de "conjuntos" (sets) e "associações" (associations) para representar relacionamentos complexos entre os dados.

Embora mais flexível que o modelo hierárquico, ainda é complexo e raramente utilizado em sistemas modernos.

Modelo Relacional:

Este é o modelo mais amplamente utilizado na indústria atualmente.

Os dados são organizados em tabelas relacionadas umas às outras por meio de chaves estrangeiras.

As operações são realizadas usando a linguagem SQL (Structured Query Language), que permite consultas complexas e manipulações de dados.

Modelo Orientado a Objetos:

Este modelo representa os dados na forma de objetos, que consistem em dados e métodos associados.

É especialmente útil em ambientes de programação orientada a objetos, onde os objetos são facilmente mapeados para entidades no banco de dados.

Modelo de Entidade-Relacionamento (ER):

Este modelo é usado para representar a estrutura lógica do banco de dados de forma visual, usando diagramas de entidade-relacionamento.

Entidades representam objetos do mundo real, e os relacionamentos definem como essas entidades estão conectadas umas às outras.

É uma ferramenta útil para projetar e comunicar a estrutura do banco de dados antes de implementá-lo fisicamente.

Modelo Dimensional:

Este modelo é usado principalmente em data warehouses e sistemas de business intelligence.

Os dados são organizados em tabelas de fatos e dimensões, otimizados para facilitar a análise de grandes conjuntos de dados.

Cada modelo de banco de dados tem suas próprias vantagens, desvantagens e casos de uso adequados. A escolha do modelo certo depende das necessidades específicas do projeto, como volume de dados, complexidade dos relacionamentos e requisitos de desempenho.

VANTAGENS DO MER - MODELO ENTIDADE-RELACIONAMENTO

O Modelo Entidade-Relacionamento (MER) é uma técnica de modelagem de dados que oferece uma série de vantagens no projeto e desenvolvimento de bancos de dados. Aqui estão algumas das principais vantagens do MER:

Abstração de Complexidade: O MER oferece uma representação simplificada e abstrata da estrutura do banco de dados, focando nas entidades e nos relacionamentos entre elas. Isso permite aos projetistas e desenvolvedores lidar com a complexidade do sistema de forma mais organizada e compreensível.

Compreensão Visual: O MER é uma ferramenta visual que utiliza diagramas para representar as entidades, atributos e relacionamentos do banco de dados. Isso torna mais fácil para os stakeholders entenderem a estrutura do banco de dados, mesmo sem conhecimento técnico profundo.

Identificação de Requisitos: Ao criar um MER, os analistas podem identificar e documentar os requisitos do sistema de forma clara e concisa. Isso ajuda a garantir que o banco de dados atenda às necessidades dos usuários finais e aos objetivos do negócio.

Flexibilidade no Design: O MER permite que os projetistas experimentem diferentes abordagens de modelagem antes de implementar o banco de dados. Isso proporciona flexibilidade para ajustar o design com base em feedback e mudanças nos requisitos, antes de comprometer recursos na implementação.

Integridade dos Dados: O MER suporta a definição de restrições de integridade, como chaves primárias, chaves estrangeiras e restrições de

integridade referencial. Isso ajuda a garantir a consistência e a qualidade dos dados armazenados no banco de dados.

Manutenção e Evolução: O MER facilita a identificação e resolução de problemas de design durante o processo de desenvolvimento. Além disso, fornece uma base sólida para a manutenção e evolução contínua do sistema à medida que os requisitos mudam ao longo do tempo.

Comunicação Eficaz: O MER serve como uma ferramenta de comunicação eficaz entre os membros da equipe de desenvolvimento, permitindo que eles discutam e colaborem no design do banco de dados de forma clara e concisa.

Essas vantagens tornam o Modelo Entidade-Relacionamento uma técnica valiosa no projeto de bancos de dados, ajudando a garantir a eficácia, eficiência e qualidade dos sistemas de informação.

ARQUITETURAS DE BANCO DE DADOS

As arquiteturas de banco de dados são fundamentais para o design e funcionamento de sistemas de armazenamento de dados. Aqui estão algumas das arquiteturas comumente usadas:

Arquitetura Cliente-Servidor: Nesta arquitetura, o banco de dados é hospedado em um servidor central e os clientes se conectam a ele para acessar e manipular os dados. Essa abordagem permite compartilhar recursos de forma eficiente e facilita a manutenção centralizada.

Arquitetura de N camadas (ou multicamadas): Nesta arquitetura, o sistema é dividido em várias camadas, com cada camada tendo sua própria responsabilidade. Isso pode incluir uma camada de apresentação, uma camada de lógica de negócios e uma camada de armazenamento de dados. Essa abordagem promove a modularidade e a escalabilidade.

Arquitetura Peer-to-Peer (P2P): Nesta arquitetura, cada nó na rede atua como cliente e servidor ao mesmo tempo. Os dados são distribuídos entre os nós e cada nó é responsável por uma parte do armazenamento e processamento. Isso é comumente usado em sistemas distribuídos e descentralizados.

Arquitetura de Data Warehouse: Esta arquitetura é projetada para análise e consulta de grandes conjuntos de dados. Os dados são extraídos de várias fontes, transformados e carregados em um repositório centralizado (o data warehouse) onde podem ser consultados por ferramentas de business intelligence e análise.

Arquitetura de Big Data: Esta arquitetura é usada para lidar com grandes volumes de dados que não podem ser facilmente processados por sistemas tradicionais de banco de dados. Isso geralmente envolve o uso de tecnologias distribuídas, como Hadoop e Spark, para armazenar e processar os dados em clusters de computadores.

Cada uma dessas arquiteturas tem suas próprias vantagens e desvantagens, e a escolha da arquitetura certa depende dos requisitos específicos do sistema, como escalabilidade, disponibilidade, segurança e desempenho.

PADRÕES SQL - STRUCTURED QUERY LANGUAGE

Pode ser traduzido para o português como "Linguagem de Consulta Estruturada". É uma linguagem de programação utilizada para gerenciar e manipular dados em bancos de dados relacionais.

Os padrões SQL, ou Structured Query Language, são conjuntos de regras e diretrizes que especificam como uma linguagem de consulta relacional deve ser escrita e interpretada. Isso garante consistência e interoperabilidade entre diferentes sistemas de gerenciamento de banco de dados que suportam SQL. Aqui estão alguns dos padrões SQL mais comuns:

SQL-86: Também conhecido como SQL-87, foi o primeiro padrão SQL estabelecido pelo ANSI (American National Standards Institute) em 1986. Ele definiu os conceitos básicos da linguagem SQL, como SELECT, INSERT, UPDATE e DELETE.

SQL-89: Este padrão foi uma extensão do SQL-86 e adicionou a funcionalidade de junções e subconsultas.

SQL-92: Também conhecido como SQL2, foi lançado em 1992. Esta versão foi um grande avanço em relação às anteriores e introduziu muitos recursos importantes, como suporte para tipos de dados definidos pelo usuário, consultas recursivas e a cláusula CASE.

SQL:1999 (SQL3): Esta versão trouxe recursos avançados, incluindo suporte para procedimentos armazenados, gatilhos, tipos de dados complexos, hierarquias e tratamento de XML.

SQL:2003: Esta versão concentrou-se na adição de recursos de suporte a objetos, como tipos de dados complexos, herança e métodos.

SQL:2008: Esta versão incluiu melhorias significativas na linguagem, como suporte para tabelas temporárias, expressões regulares e operações OLAP.

SQL:2011: Esta versão foi uma atualização menor e focou principalmente em melhorias na especificação da linguagem e correções de erros.

SQL:2016: Introduziu recursos como expressões JSON, consultas de tabelas temporais e melhorias na instrução MERGE.

SQL:2019: A versão mais recente, introduziu recursos como suporte a aprendizado de máquina, expressões temporais e melhorias na manipulação de gráficos.

Embora os sistemas de banco de dados possam implementar diferentes partes dos padrões SQL de maneiras ligeiramente diferentes e também podem ter extensões proprietárias, seguir os padrões SQL ajuda a garantir a portabilidade das consultas entre diferentes sistemas de banco de dados compatíveis com SQL.

ESTUDO DE CASO - PROJETO

Um estudo de caso de projeto pode abordar uma ampla variedade de tópicos e setores. Aqui está um exemplo genérico de um estudo de caso de projeto:

Título do Projeto:

Implementação de um Sistema de Gerenciamento de Biblioteca Online

Contexto:

Uma biblioteca local deseja modernizar seu sistema de gerenciamento de biblioteca para melhorar a eficiência e a experiência do usuário. Atualmente, o sistema é baseado em papel e lápis para registro de empréstimos, devoluções e pesquisa de catálogo.

Objetivo do Projeto:

Desenvolver e implementar um sistema de gerenciamento de biblioteca online que permita aos usuários pesquisar o catálogo, fazer reservas, renovar empréstimos e gerenciar suas contas.

Atividades:

Levantamento de requisitos: Realizar entrevistas com os usuários da biblioteca para entender suas necessidades e requisitos para o novo sistema.

Análise de sistemas existentes: Avaliar sistemas de gerenciamento de biblioteca online existentes para identificar funcionalidades essenciais e melhores práticas.

Design do sistema: Criar uma arquitetura de sistema e design de banco de dados para suportar as funcionalidades desejadas.

Desenvolvimento de software: Desenvolver o sistema de acordo com os requisitos e o design estabelecidos, utilizando tecnologias web modernas.

Testes: Realizar testes de qualidade e usabilidade para garantir que o sistema atenda aos requisitos e expectativas dos usuários.

Implementação: Lançar o sistema e fornecer treinamento para funcionários da biblioteca e usuários finais.

Suporte e manutenção: Fornecer suporte contínuo e manutenção para garantir o bom funcionamento do sistema e realizar atualizações conforme necessário.

Resultados Esperados:

Melhoria da eficiência no gerenciamento de empréstimos e devoluções.

Aumento da satisfação do usuário com a capacidade de acessar o catálogo da biblioteca online.

Redução de custos operacionais associados ao uso de sistemas baseados em papel.

Aumento do acesso à informação por meio de um sistema mais acessível e fácil de usar.

Considerações Finais:

O sucesso do projeto será medido pelo alcance dos objetivos estabelecidos, bem como pela adoção e satisfação dos usuários com o novo sistema de gerenciamento de biblioteca online.

Este é apenas um exemplo genérico de um estudo de caso de projeto. Os detalhes específicos podem variar dependendo do contexto, dos requisitos e das características do projeto em questão.

AMBIENTE DE DESENVOLVIMENTO USADO E FERRAMENTAS PARA UM BANCO DE DADOS DE UM SUPERMERCADO

Para a implementação de um banco de dados para um mercado, é essencial escolher ferramentas e tecnologias que ofereçam recursos de armazenamento e recuperação de dados eficientes e confiáveis. Abaixo estão algumas opções comuns de ambiente de desenvolvimento e ferramentas para esse tipo de projeto:

Sistema de Gerenciamento de Banco de Dados (SGBD):

MySQL: Um SGBD relacional de código aberto amplamente utilizado, adequado para aplicações de pequeno e médio porte.

PostgreSQL: Outra opção de SGBD relacional de código aberto conhecida por sua robustez e recursos avançados, como suporte a JSON.

SQLite: Um SGBD leve e autônomo, ideal para projetos menores ou para desenvolvimento e teste.

Ferramentas de Modelagem de Dados:

MySQL Workbench: Uma ferramenta oficial da Oracle para modelagem, desenvolvimento e administração de bancos de dados MySQL.

DBeaver: Uma ferramenta de banco de dados universal que oferece suporte a vários SGBDs, incluindo MySQL, PostgreSQL e SQLite.

pgModeler: Especializada em modelagem de dados para PostgreSQL, com recursos avançados e interface intuitiva.

Ferramentas de Implantação e Hospedagem:

Heroku, AWS, Google Cloud Platform: Opções de hospedagem na nuvem para implantar o backend e o banco de dados.

Essas são algumas das ferramentas e tecnologias que podem ser úteis no desenvolvimento e implementação de um banco de dados para um mercado. A escolha específica dependerá das necessidades do projeto, preferências da equipe de desenvolvimento e considerações de infraestrutura.

CONSIDERAÇÕES FINAIS

A implementação de um banco de dados em um mercado é um passo crucial para modernizar e otimizar os processos de negócios, fornecendo uma base sólida para o gerenciamento eficiente de estoque, vendas, clientes e operações. Ao longo deste trabalho, exploramos a importância de um banco de dados em um minimercado, destacando como ele facilita o controle de estoque, a gestão de vendas, a personalização do atendimento ao cliente, a análise de dados e a automação de processos operacionais.

Além disso, discutimos conceitos fundamentais relacionados a bancos de dados, como modelos de dados, sistemas de banco de dados, modelos de banco de dados e padrões SQL. Esses conceitos fornecem uma compreensão mais profunda de como os bancos de dados são projetados, implementados e utilizados em diferentes contextos.

Abordamos também um estudo de caso hipotético sobre a implementação de um sistema de gerenciamento de biblioteca online, destacando os principais passos envolvidos no projeto, desde a análise de requisitos até a implementação e suporte contínuo.

Com base nas informações apresentadas, fica evidente que um banco de dados desempenha um papel fundamental na operação eficiente e no sucesso de um mercado. Ao adotar as melhores práticas de design e implementação de banco de dados, os mercados podem melhorar sua capacidade de atender às demandas dos clientes, tomar decisões estratégicas informadas e manter uma vantagem competitiva no mercado em constante evolução.

B I B L I O G R A F I A

Banco de Dados: Conceitos e Aplicações

Autor(es): Ramez Elmasri e Shamkant Navathe

Título da obra: Fundamentos de Banco de Dados

Local de publicação: São Paulo

Editora: Pearson

Data de publicação: 2021

Programando com SQL

Autor(es): Ivan Fernandes

Título da obra: Programando com SQL: Um Guia Completo para Desenvolvedores e Profissionais de Banco de Dados

Local de publicação: Rio de Janeiro

Editora: Elsevier

Data de publicação: 2018