

UNIVERSIDADE DE BRASÍLIA

Faculdade do Gama

Sistemas de Banco de Dados 2

Tecnologias de Banco de Dados (TI-BD)

Bancos de Dados Orientado a Objeto e Objeto-Relacional

Ailton Aires Amado - 180011600

Brasília, DF

2023

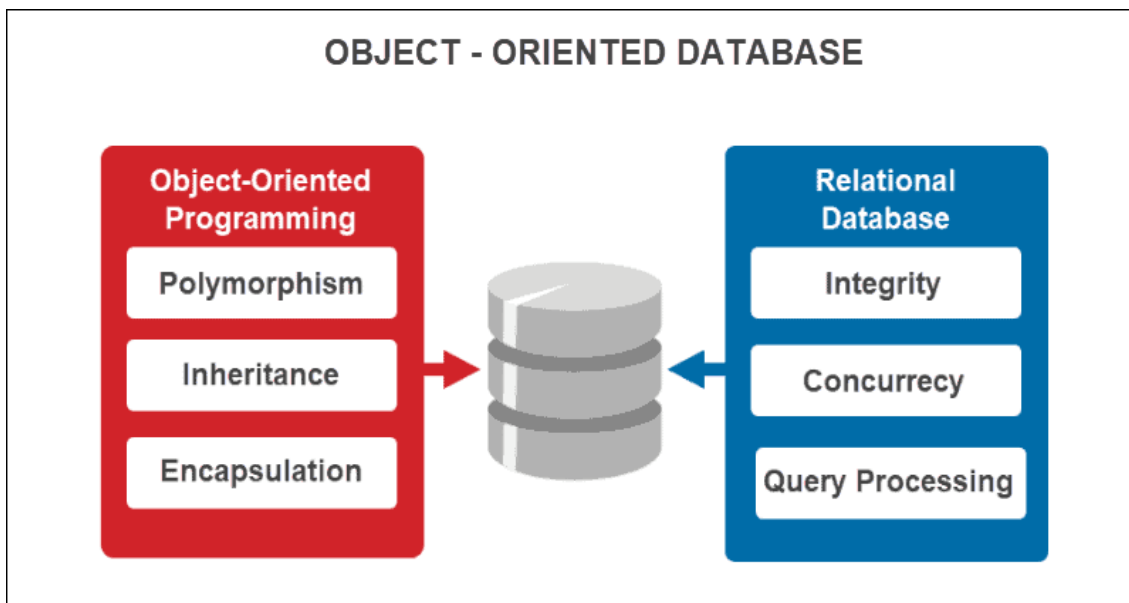
Definição

Como definição para banco de dados, temos que um banco de dados é um conjunto de dados inter-relacionados cuja finalidade é armazenar informações (DATE et al., 2004, p. 5; ELMASRI; NAVATHE, 2011, p. 3; RAMAKRISHNAN; GEHRKE, 2008 p. 3; SILBERSCHATZ et al. 2010, p. 1). Esses dados podem ser compreendidos como fatos capazes de serem registrados e que possuem algum significado associado a eles. Embora essa seja uma definição aceita, ela é comumente considerada genérica. Ainda de acordo com o Elmasri e Navathe (op. cit.), o termo banco de dados tem um uso mais restrito e possui propriedades como:

- representação de algum aspecto do mundo real, onde alterações nesses aspectos são reproduzidos no banco de dados.
- coleção de dados logicamente coerentes, que possui significado e não aleatório.
- são manipulados e gerenciados para uma finalidade específica, e que possui um grupo específico de usuários.

Após definir o que é um banco de dados, podemos falar sobre Object-Oriented Database (OOD), ou banco de dados orientado a objetos (BDOO). Os BDOO ou apenas “bancos de dados de objeto (BDO)” (ELMASRI; NAVATHE, op. cit., p. 236), são um conjunto de dados conectados, relacionados e interligados que possuem grandes volumes de dados e informações logicamente coerentes e que possuem uma estrutura de armazenagem de dados baseado nos paradigmas de orientação a objetos (OO), ou seja, na forma de objetos ao invés de tabelas como nos bancos de dados relacionais. Algumas das principais características de OO e que definem os BDO são apresentadas na figura 1 abaixo.

Figura 1 - Object-Oriented Database



Fonte: DANCUK (2021)

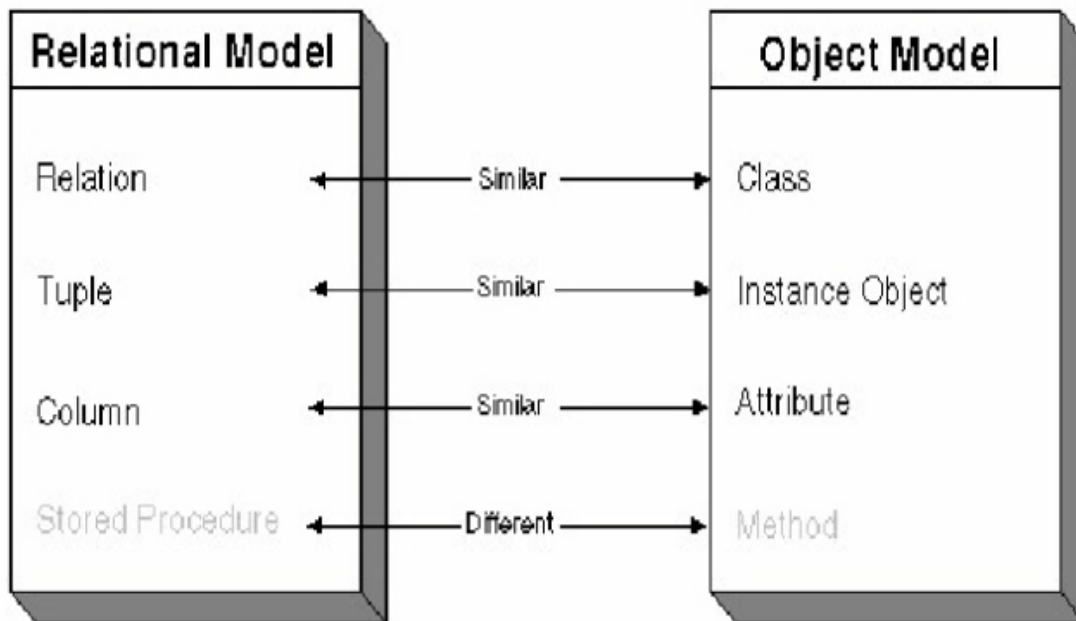
A figura 1 representa algumas das principais diferenças diante das características de um banco de dados orientado a objetos para um banco de dados relacional. Enquanto um BDO possui características de OO (polimorfismo, herança e encapsulamento), o banco de dados relacional (BDR) possui a integridade, concorrência e processamento de requisição, além de outras como sendo aspectos importantes no contexto de BDR. Vale ressaltar que um SGDO (Sistema de Gerenciamento de Dados de Objeto) é responsável por manter a integridade e identidade dos objetos para que sejam facilmente identificados. Isso é feito por meio de uma identidade única definida para cada objeto, que é conhecido como identificador de objeto (IDO), ou em inglês *Object Identifier (OID)* (ELMASRI; NAVATHE, 2011, p.239).

Quando falamos de *Object-Relational Database (ORDB)* ou do português Banco de Dados Objeto-Relacional (BDOR), é um sistema de banco de dados que procura obter benefícios dos modelos relacional e de objeto e usa a abordagem OO e relacional para o armazenamento e manipulação dos dados, combinando características dos BDR e BDO (ELMASRI; NAVATHE, 2011, p.247; RAMAKRISHNAN; GEHRKE, 2008, p. 643; SILBERSCHATZ et al.

2010; p. 257; DEVARAKONDA, 2001, p.3).

Na figura 2 é possível perceber as similaridades e diferenças da modelagem baseadas em objetos e relacionamentos.

Figura 2 - Modelo Relacional x Modelo de Objeto

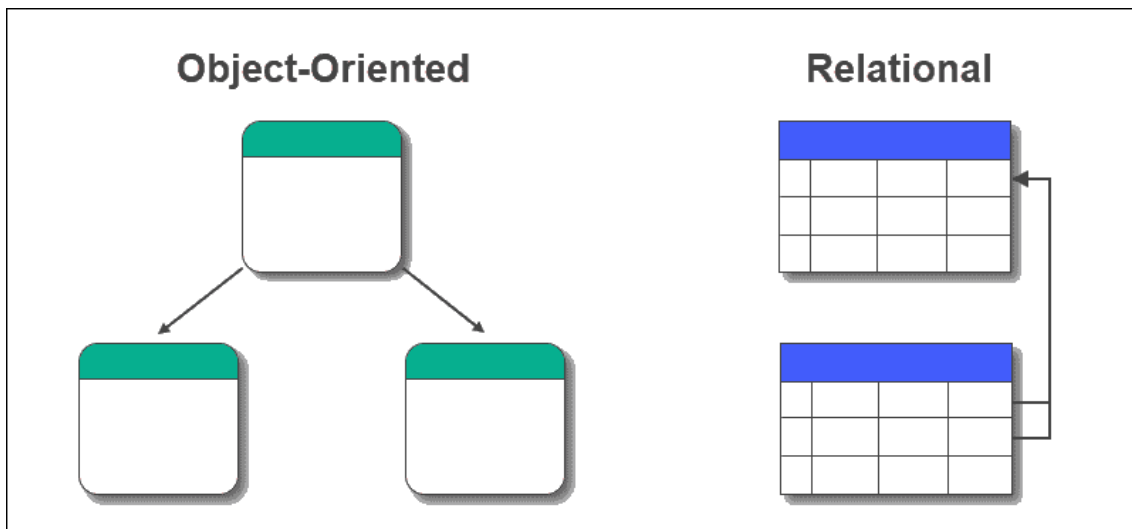


Fonte: Alzahrani (2016)

Basicamente uma *stored procedure* é um procedimento armazenado em BDR e que pode ser executado sem a necessidade de escrever algum código fonte. É composto por instruções SQL pré-compiladas e armazenadas no BD. Já um método de um objeto é um procedimento armazenado dentro do próprio objeto em BDO e que permite que operações sejam executadas diretamente no objeto. Vale ressaltar que o método é específico para um determinado objeto, enquanto a *stored procedure* pode ser utilizada em diferentes situações.

A figura 3 mostra a estrutura e os relacionamentos dos BDO e BDR.

Figura 3 - Relacionamentos dos modelos de objetos e de tabelas



Fonte: Alzahrani (2016)

Basicamente o que a imagem apresenta é um esboço bem simples dos relacionamentos nos BDO e nos BDR. Dentro dos BDO, os objetos se relacionam com outros objetos, enquanto os BDR se relacionam entre tabelas. Ambos os relacionamentos são capazes de estabelecer organização e conexão dos dados no banco de dados, mas de maneiras um pouco diferentes.

Objetivos

Segundo Devarakonda (2001, p.1), o desejo de poder representar objetos de difícil compreensão levou a criação dos sistemas hoje comumente chamados de OO, onde suas principais características são: encapsulamento, herança e polimorfismo. Ainda segundo o autor, BDO trazem um modelo de dados com capacidade de suportar os recursos orientados a objetos juntamente com tipos abstratos de dados.

Dito isso, os objetivos de BDO são semelhantes ao de BDR, que é “persistir dados necessários para a manutenção do negócio para o qual são aplicados, possibilitando a recuperação, comparação e tratamento desses dados a fim de produzir resultados tangíveis” (BOSCARIOLI, 2006, p. 2). Esse tipo de banco foi construído em torno de linguagens de programação persistentes e para atender aplicações mais complexas (ELMASRI; NAVATHE, op. cit., p. 236). Os BDO permitem uma maior integração entre o banco de

dados e as aplicações orientadas a objeto, já que a estrutura dos dados ficou mais parecida com as classes usadas no software (TELES, et al., 2011, p. 1). As principais características de OO (objetos, métodos, herança e polimorfismo), fundamentais nas linguagens OO, caracterizam o BDO (RICARTE, 1998, p. 12).

Já os BDOR tem como principal objetivo “obter os benefícios dos modelos relacional e de objeto, como escalabilidade e suporte para tipos de dados ricos” (DEVARAKONDA, 2001, p. 3). Ou seja, os dados, por exemplo, serão armazenados de diferentes formas, sejam baseados em tabelas ou em uma estrutura de dados mais rica, que são conhecidas como tipos definidos pelo usuário (*UDTs - User-Defined Types*), semelhantes aos tipos de dados abstratos (*ADTs - abstract data types*). A grande diferença é que os ADTs foram desenvolvidos nas linguagens de programação.

Vantagens

Segundo alguns autores (ELMASRI; NAVATHE, 2011; STONEBRAKER; MOORE, 1996; SILBERSCHATZ et al., 2010), cada banco de dados, sejam eles BDR, BDO ou BDOR, têm suas vantagens. No caso dos BDO e BDOR, temos:

- armazenamento direto de objetos persistentes e complexos de forma permanente, permitindo o compartilhamento dos objetos entre vários programas e aplicações. No caso dos BDR, os dados são mapeados em tabelas relacionais para serem armazenados
- suportam tipos de dados complexos, compostos e definidos pelos usuários (*UDTs*). Geralmente os BDR suportam tipos de dados específicos, mas com a crescente dos BDO, os bancos de dados se adaptaram a essa realidade e passaram a suportar (*UDTs*)
- interação com mais de uma linguagem orientada a objetos, com capacidade de oferecer objetos persistentes e seus comportamentos. BDR também possuem esse suporte, porém utilizando a estrutura relacional

- os mecanismos de indexação, para localizar os objetos com eficiência, o controle de concorrência, para permitir o compartilhamento de objeto entre programas concorrentes, e a recuperação de falhas. Todas essas características existem nos BDR, respeitando a ideia das tabelas.
- embora o encapsulamento completo possa ser considerado tanto um ponto negativo quanto positivo em BDO, dependendo do contexto, dentro dos pontos positivos temos a garantia da integridade e segurança dos objetos armazenados. Também podemos destacar que, dependendo do nível de encapsulamento, as operações internas podem ser modificadas sem que programas externos que utilizam esses objetos sejam impactados. Isso também ocorre nos BDR, já que operações como *DML (Data Manipulation Language)* e *DDL (Data Definition Language)* estão dentro do contexto de encapsulamento, principalmente quando falamos de *stored procedures*.
- BDOR é mais rico em termos de recursos e funcionalidades do que um modelo de dados relacional
- BDOR também possuem um aspecto fundamental, que é a capacidade de combinar recursos tanto dos BDO quanto dos BDR, o que torna a modelagem mais flexível

Desvantagens

Quanto às desvantagens, alguns autores (BOSCARIOLI, 2006; ELMASRI; NAVATHE, 2011; WEIDLER, 2004 apud BOSCARIOLI, 2006) abordam os BDOO e BDOR como tendo os seguintes aspectos negativos:

- alto encapsulamento ou encapsulamento completo, onde o estado interno do objeto não pode ser acessado ou modificado diretamente de fora do objeto, ou seja, dificulta o acesso dos usuários do banco de dados, pois eles precisam saber os nomes de atributo para especificar condições de seleção sobre os atributos a fim de buscar e recuperar objetos específicos, além de que uma recuperação simples acaba exigindo uma operação predefinida, tornando assim as consultas

rotineiras difíceis de especificar no ato. Dentro do contexto de BDR, isso ocorre com frequência na *DML* e *DDL*.

- um problema nos primeiros sistemas de banco de dados OO envolvia a representação dos relacionamentos entre objetos. Nos primeiros sistemas de BDO, prezava-se pelo encapsulamento completo, onde cada objeto deveria ter todas as informações e comportamentos necessários para o seu funcionamento, ou seja, não deveria depender de informações externas ou de outros programas/objetos que não fossem o próprio objeto. Essa abordagem era ineficiente quando se tratava de relacionamentos entre objetos. Diferente disso, os BDR são totalmente eficazes quando o assunto é relacionamento.
- falta de padronização. Diferente dos BDO, os BDR possuem uma padronização quanto ao tipo, estrutura de armazenamento, linguagem de manipulação de dados, etc.
- embora os BDOR sejam mais ricos em alguns aspectos, é essa mesma riqueza que pode acabar gerando mais complexidade dentro do processo de design de banco de dados, além de dificultar na manutenção do banco de dados. Os BDR possuem uma estrutura madura o suficiente que acaba tendo aspectos muito bem definidos.
- dificuldade na modelagem de dados objeto-relacional, quando combinado os recursos de BDR e BDO. Como os BDR possuem estrutura e paradigmas únicos, o processo de manipulação e modelagem se torna mais simples
- adoção escassa, já que o tempo e custo envolvido em uma possível migração de um BDR para um BDO ou BDOR faz com que empresas hesitem em utilizar esse novo paradigma. Como o mercado é dominado quando se fala na utilização de BDR, os BDR não têm uma preocupação iminente quando a sua escassez de uso e/ou migração
- processos que são comuns nos BDR como otimização, migração e restauração de dados ainda carecem de amadurecimento nos BDOO, já que os BDOO são relativamente novos em comparação aos BDR

Exemplos de uso

A Oracle Database é uma solução de banco de dados muito popular e que possui suporte a recursos orientado a objetos, como tipos de dados definidos pelos usuários (*UDTs*), herança e polimorfismo, sendo assim considerado um BDOR. O Oracle Database foi usado em parceria com a NASA. Um exemplo é o projeto *Mars Rover*, que utilizou a tecnologia Oracle Database no armazenamento e gerenciamento de informações durante a missão (WAGNER, 2022). Ainda segundo Wagner (op. cit.), foi apenas um sucesso diante de um longo relacionamento entre a NASA e a própria Oracle, “começando quando a NASA padronizou os servidores da Sun Microsystems na década de 1980, no Oracle Database em 2006, continuando com a aquisição da Sun pela Oracle em 2010 e até hoje”.

Um outro exemplo de BDOO é o db4o. Foi desenvolvido pela Versant Corporation e foi projetado para fornecer alta performance em locais que necessitasse de um gerenciamento de grande volume de dados. Ele fornecia suporte para as linguagens JAVA e .Net e usava a linguagem de consulta *Object Query Language (OQL)*, além de possuir licença GPL (Licença Pública Geral) e comercial (GALANTE et. al, 2009). Apesar de estar presente e obter sucesso em projetos de algumas empresas, como a Boeing para o projeto do P-8A Multi-Mission Maritime Aircraft (ROSENBERGER, 2007), o db4o foi descontinuado. Embora isso tenha ocorrido, o db4o foi muito importante para projeto da Boeing, já que segundo Rosenberger (op. cit.), a escolha do banco de dados foi baseada em um banco que tivesse suporte nativo para objetos, o que resultaria em uma menor administração do banco de dados e uma maior produtividade para os desenvolvedores. Outras empresas renomadas também adotaram o db4o como banco de dados para o seus projetos. Ainda segundo o autor, algumas empresas e relatos de funcionários de como o db4o auxiliava no processo de desenvolvimento de cada uma delas:

- RICOH: db4o forneceu uma solução de persistência para os desafios da empresa e pelos rigorosos padrões de qualidade. Após um longo período avaliativo da empresa, foi constatado que o db4o tinha uma

flexibilidade que atendia a arquitetura da empresa, visando produtividade no desenvolvimento de software orientado a objetos.

- Intel: o db4o tornou o desenvolvimento de aplicativos muito mais intuitivo para o grupo. Outras alternativas convencionais de banco de dados não atendiam as necessidades da empresa, e foi com o db4o que a arquitetura existente passou a funcionar adequadamente.

As demandas e necessidades de empresas como Seagate, BOSCH Indra e outras também foram atendidas com base no banco db4o.

Referências Bibliográficas

ALZAHRANI, H. **Evolution of Object-Oriented Database Systems**. 2016. Disponível em: [link para o site](#). Acesso em: 14 de abr. de 2023.

BOSCARIOLI, C.; BEZERRA, A.; BENEDICTO, M.; DELMIRO, G. **Uma reflexão sobre Banco de Dados Orientados a Objetos**. IV CONGED, 2006.

DANCUK, M. **Object-Oriented Database**. 2021. Disponível em: [link para o site](#). Acesso em: 14 de abr. de 2023.

DATE, C.J. **Introdução a Sistemas de Bancos de Dados**. Grupo GEN, 2004. *E-book*. Disponível em: [link do site](#). Acesso em: 16 abr. 2023.

DEVARAKONDA, R. S. **Object-relational database systems - the road ahead**. Crossroads, v. 7, n. 3, 15 mar. 2001.

ELMASRI, R., & NAVATHE, S. B. **Sistemas de Banco de Dados**. 6. ed. São Paulo: Pearson, 2011.

GALANTE, A. C.; MOREIRA, E. L. R.; BRANDÃO, F. C. **Banco de Dados Orientado a Objetos: Uma Realidade**, Revista de Sistemas de Informação da FSMA n. 3, 2009 p. 55-69

RAMAKRISHNAN R.; GEHRKE J. **Sistemas de Gerenciamento de Banco de Dados**. Editorial: São Paulo: Mcgraw-Hill Medical, 2008.

RICARTE, I. **Sistemas de Bancos de Dados Orientados a Objetos**. Depto. Engenharia de Computação e Automação Industrial. Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, 1998.

ROSENBERGER, C., WITTIG, C. **The Open Source Object Database**. 16 de mai. de 2007. Apresentação do Power Point. Disponível em: [link da apresentação](#). Acesso em: 15 de abr. de 2023.

STONEBRAKER, M.; BROWN, P.; MOORE, D. **Object-relational DBMSs**. [s.l.] Morgan Kaufmann, 1999.

TELES, J.; CÉSAR, P.; PRADO, P.; ARAÚJO, R. **Banco de Dados Orientado a Objetos**. Ciência da Computação, Universidade Federal da Bahia, Salvador, 2011.

WAGNER, M. NASA's Deep Space Network communicates across the Solar System - and beyond. **Oracle Connect**, 27 de jun. de 2022. Disponível em: [link do site](#). Acesso em: 15 de abr. de 2023