

Banco de Dados Orientado a Objetos e Objeto Relacional

Alexandre Specian Cardoso
Hemerson Leandro Siqueira Carlin
Jefferson Adriano do Rosario



Banco de Dados

- Armazenar dados específicos
 - Mais utilizados são os relacionais
 - Necessidade de se trabalhar com aplicações mais complexas, levou a evolução dos BDOO's
- 

BDOO

- Origem na combinação de ideias dos modelos de dados tradicionais e de linguagens de programação orientada a objetos
 - Possuem maior naturalidade conceitual
 - Estão em harmonia com fortes tendências em linguagens de programação e engenharia de software
- 

BDOO - Utilização

- Objetos Complexos: formados por construtores (conjuntos, listas, tuplas, registros, coleções, arrays)
 - Objetos Simples (inteiros, booleanos, strings)
 - SGBDOO Consultas complexas
- 

Exemplos de Aplicações Complexas

- Projetos de engenharia e arquitetura.
 - Experiências científicas.
 - Telecomunicações.
 - Sistemas de informações geográficas.
 - Multimídia.
- 

Surgimento dos BDOO's

- Cada objeto possui um identificador: OID
 - Não utiliza a linguagem SQL
 - Não há queries
 - Busca de objetos: metodologias pré-definidas
- 

Banco OO x Banco Relacional

Modelo Relacional	Modelo OO
Tabelas (entidades)	Objetos
Linhas (registros)	Tuplas
Query's (consultas,etc)	Native Query's
Sql Ansci	Métodos, construtores

Exemplos de SGBDOO's

- DB4O
 - Caché
 - ZODB
 - Oracle 10g
- 

DB4O

- Banco de dados para manipulação de objetos
 - Fácil armazenamento e manipulação
 - Alta performance
 - Baixo custo (praticamente zero)
- 

Caché

- Provê a união dos paradigmas relacional e orientação a objetos
 - Aplicações relacionais podem coexistir com negócios construídos com a tecnologia orientada a objetos
 - Pós-relacional
- 

ZODB

- Banco de dados orientado a objetos para Python
 - Alto grau de transparência
 - Objetos obtidos e atualizados através de interações normais de objetos
 - Alto desempenho e uso eficiente da memória
- 

Oracle 10g

- Combina um banco de dados relacional com conceitos de OO.
 - Possui o Object Type, que equivale a uma classe
 - Captura tanto a estrutura, quanto o comportamento de um objeto
- 

Vantagens

- Capacidade de armazenamento
 - Poder de processamento de requisições
 - Não possuem chaves primárias e estrangeiras
 - Comunicação por mensagens
- 

Desvantagens

- Falta de padronização
 - Alto custo da aquisição de tecnologias
 - Aprendizagem e adaptação ao novo ambiente
- 

Banco de dados Objeto Relacional

- Armazena objeto de forma relacional
 - Utiliza classes e atributos para formar e preencher tabelas
 - Reduz a impedância da programação OO
 - Necessita de uma ferramenta para fazer a tradução Objeto – Tabela
 - Java ou Net
- 

Ferramentas

- Hibernate
 - XML
 - Doctrine
 - ORMer
 - Propel
- 

Conclusão

Hoje em dia os SGBDR's estão fortemente consolidados no mercado de trabalho. Porém com o desenvolvimento das LPOO's, é cada vez mais necessário a utilização de um SGBDOO, que por sua vez ainda precisam amadurecer.

Enquanto isso, a utilização de um SGBDOR vem se tornando uma alternativa de grande relevancia para quem quer o melhor dos dois paradigmas.