Object-Relational Database Management System (ORDBMS)

➤ *ORDBMS* stands for Object-Relational Database Management System.

➤ An ORDBMS is a database management system that is a hybrid between the object-oriented model (OODBMS) and the relational model (RDBMS).

➤ Each of those two models has their strengths and weaknesses. By combining the two models, a DBMS can take advantage of various strengths from each model.

An Object-Relational Database Management System (ORDBMS) is a type of database management system (DBMS) that combines the features of both relational databases and object-oriented databases. The primary goal of an ORDBMS is to provide a bridge between the relational model and the object-oriented model, allowing developers to use both approaches in a seamless manner.

1. **Data Modeling:**

   Relational Model: Like traditional relational databases, ORDBMS uses tables, rows, and columns to represent data. It supports SQL for defining and querying data.

2. **Object-Oriented Model**:

   Extends the relational model to include object-oriented concepts such as classes, objects, inheritance, and methods. Tables can have attributes that are complex data types, and relationships can be more closely aligned with object-oriented principles.

3. **Complex Data Types**:

   ORDBMS supports complex data types such as arrays, structs, and user-defined data types. These data types are more flexible than the simple scalar types in traditional relational databases.

4. **Inheritance**:

Object-oriented databases often include the concept of inheritance, allowing one object or class to inherit properties and behaviors from another. ORDBMS may support inheritance through the use of tables and relationships.

5. **Encapsulation:**

Encapsulation is a fundamental concept in object-oriented programming, and ORDBMS may provide mechanisms to encapsulate data and behavior within objects, allowing for better modularity and code organization.

6. **Methods (Functions and Procedures):**

ORDBMS allows developers to associate methods (functions and procedures) with objects. These methods can be invoked to perform operations on the data associated with the object.

7. **Query Language**:

SQL (Structured Query Language) remains a key component in ORDBMS, providing a powerful and standardized way to query and manipulate data.

8. **Standardization:**

ORDBMS solutions may adhere to industry standards for object-relational database technology. For example, SQL:1999 and SQL:2003 are standards that include features specific to object-relational databases.

9. **Popular ORDBMS include**:

PostgreSQL: An open-source relational database management system with support for object-relational features.

Oracle Database: A commercial relational database management system that includes extensive support for object-relational features.

IBM Db2: Another commercial database management system with support for object-relational capabilities.

While ORDBMS combines the best of both worlds, it's important to note that not all relational databases are object-relational. Many relational databases adhere strictly to the traditional relational model. The choice between a traditional

relational database and an ORDBMS depends on the specific needs and preferences of the application and its developers.

ORDBMS Examples

Examples of ORDBMSs include:

➢ PostgreSQL. Open source ORDBMS developed by the PostgreSQL Global Development Group.
➢ Oracle Database by Oracle Corporation.
➢ Informix by IBM
➢ SQL Server by Microsoft
➢ Greenplum Database by Pivotal Software

Comparison of Object Oriented Database Management System and Relational Object Oriented Database Management System:

OODBMS	ORDBMS
It stands for Object Oriented Database Management System.	It stands for Object Relational Database Management System.
Object-oriented databases, like Object Oriented Programming, represent data in the form of objects and classes.	An object-relational database is one that is based on both the relational and object-oriented database models.
OODBMSs support ODL/OQL.	ORDBMS adds object-oriented functionalities to SQL.
Every object-oriented system has a different set of constraints that it can accommodate.	Keys, entity integrity, and referential integrity are constraints of an object-oriented database.
The efficiency of query processing is low.	Processing of queries is quite effective.

****