

DBMS

ASSIGNMENT

# OBJECT ORIENTED DATABASES

- An integration of database capabilities with an object oriented programming language.
- object oriented databases are designed to work well with object oriented programming.
- ODBMSs provide the lowest cost for development and performance.
- Relational database technology has failed to handle the needs of complex information systems.
- OOP views programs as sets of data structures that have both data elements and program instructions.
- Traditional programming is organized around logic first and data second, whereas OOP is organized around data first and logic second.
- when developing an object oriented program.
  - Identify the objects involved
  - Design those objects as data elements and programs
  - Finally, a flowchart or pseudo code would be created.
- An OODB combines object oriented programming principles with database management principles.
- Object oriented programming concepts such as encapsulation, polymorphism and inheritance are enforced as well as database management concepts such as ACID properties (Atomicity, consistency, isolation and durability) which lead to system integrity.

and secondary storage management systems which allow for managing very large amounts of data.

### Advantages of OODBs

- integrated with programming language
- Automatic method storage
- user-defined types.

### Disadvantages of OODBs

- ODBMS require object-oriented programming
- converting data to an ODBMS is very expensive.
- ODBMS does not provide effective query and reporting tools

### OBJECT QUERY LANGUAGE

- Declarative query language
- Not computationally complete
- Syntax based on SQL
- Additional Flexibility.

### Example of OQL query

The following is a sample query.

"What are the names of the black products?"

select distinct p.name

from products p

where p.colour = "black"

⇒ valid in both SQL and OQL, but results are different.

## Original table

product no	Name	colour.
P1	Ford Mustang	Black
P2	Toyota celica	green
P3	Mercedu SLK	Black

## Result in SQL

Name
Ford Mustang
Mercedu SLK

## Result in OQL

string	string
Ford Mustang	Mercedu SLK.

## persistant storage

OODBS are designed to provide persistent storage for objects, meaning that the ~~open~~ objects can be stored in the database and retrieved later as needed.

This allows objects to retain their state across multiple program invocations, making OODBS suitable for long term data storage.

## Transaction management

OODBS support transaction management to ensure data integrity and consistency. Transactions are units of work that include a set of database operations.



and the ACID properties are maintained to ensure reliability and recoverability

### Concurrency control

OODBs handle concurrent access to the database by multiple users or processes. They employ concurrency control to maintain data consistency and prevent conflicts.

### Application Areas

OODBs are particularly suitable for domains that involve complex data structures and rich object relationships. They are commonly used in areas such as computer-aided design (CAD), scientific simulations, geometric information systems (GIS), multimedia applications and other domains where the data model aligns closely with the object oriented paradigm.

## ACTIVE DATABASES

Active database are database systems that supports mechanisms that enable them to respond automatically to events that are taking place either inside or outside the database system itself by supporting the specification and implementation of reactive behaviours.

The reactive behaviours reside on rules which integrates cause with an expected effect. This functionality is defined in terms of event-condition-action rules (ECA-rules). These rules (mainly active rules) allow the system to monitor and react to specific events.

### Active rules/production rules

These are stored programs, which are automatically executed or fired when some event occurs. Triggers can be written to respond to Data manipulation language (DML), DDL and Database operation.

### Architecture of an Active Database

The architecture depends on the knowledge model and the execution model of the system which are two components required for providing reactive capabilities in an active database.

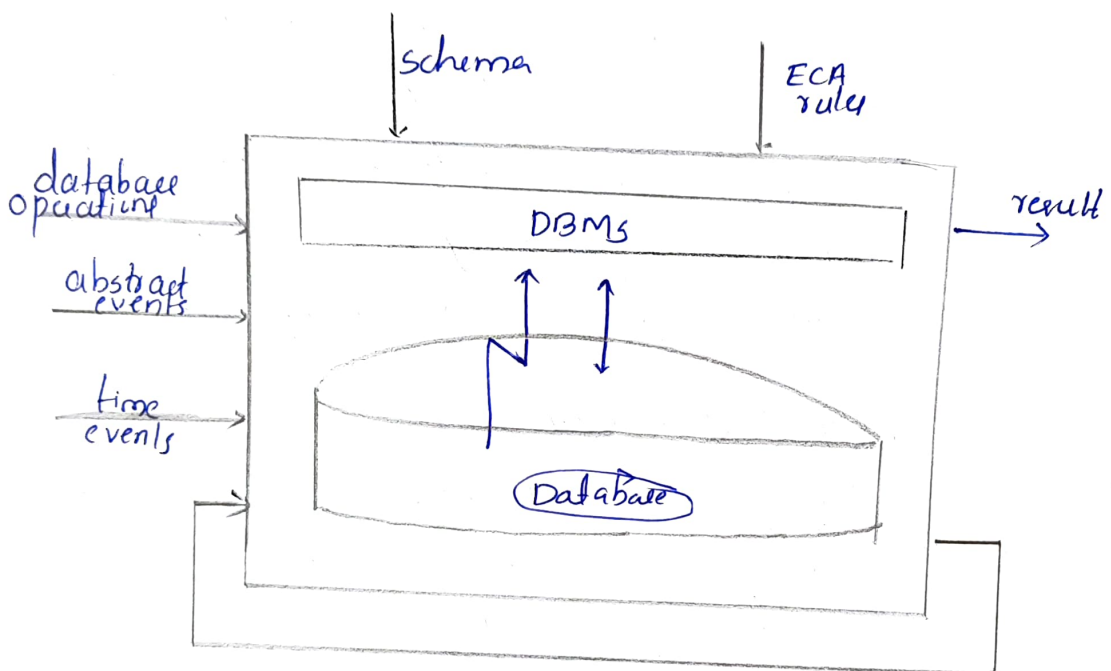
The two approaches to designing the architecture of an Active database are;

1. Built in architecture: Active database components become a part of the database.

\* implementation from scratch

\* integrated architecture - This involves modifying and extending an existing passive DBMS internally

2. Layered Architecture: Active database components are built on top of an existing passive database system.



### Feature of an active database

- It possesses all the components of a conventional database - data modelling facilities, query language, multi-user access, recovery etc.
- It supports all the functions of a traditional database including data definition, data manipulation, storage management, transaction management, concurrency control and ~~and~~ crash recovery.
- An active database supports definition and management of ECA-rules.



4. An active database must detect event occurrences.
5. An active database must be able to evaluate conditions and to execute actions.

### Applications of Active databases

1. Applications which depend on data monitoring activities such as CRM, Telecommunications Network management, program trading, Medical and Financial Decision Support Systems can greatly benefit from active database.
2. Production control - eg: power plants.
3. Maintenance tasks, eg: inventory control.
4. Financial applications.
5. Air traffic control.
6. Statistics gathering and authorization tools.

### STRENGTHS/BENEFITS OF ACTIVE DATABASES

1. Active database systems enhance traditional database functionalities with powerful rule processing capabilities.
2. Triggers in active databases enable a uniform and centralized description of the business rules relevant to the information systems.
3. The layered approach is beneficial for active object oriented database if the base systems in the tier implemented in an object-oriented way such that functionality to be rewritten can be easily modified or wrapped.



## Weaknesses of active databases

1. Insufficient methodological support in design and analysis
2. Lack of standardization.
3. Missing development and administration tools for triggers.
4. Weak performance
5. Lack of support for application development in many active ~~but~~ database management system prototypes.
6. Distribution and parallelism has not been widely treated as active databases have been considered primarily in centralized ~~but~~ database environments.