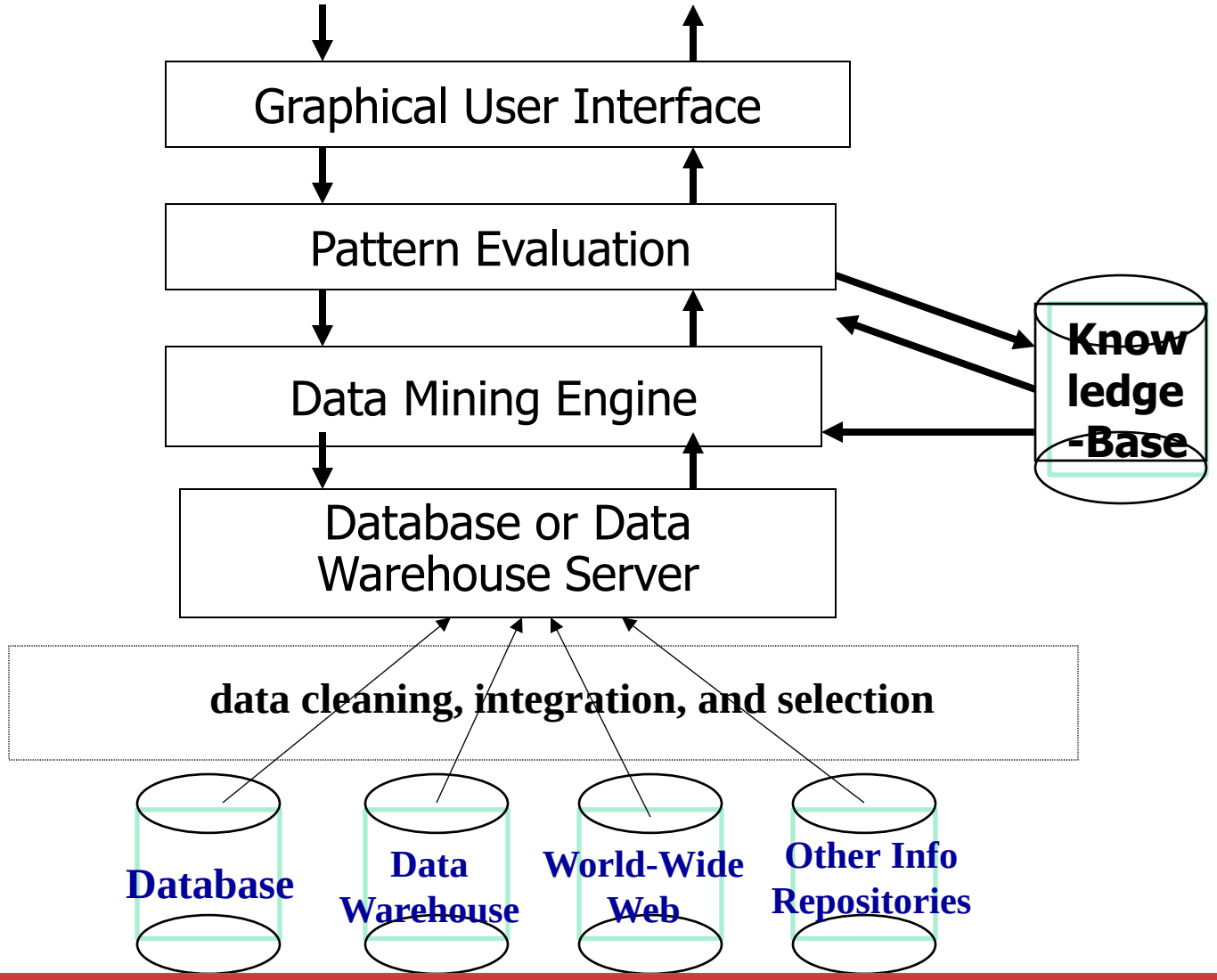
Integration of Data Mining and Data Warehousing

- A good system architecture will facilitate the data mining system
- to make best use of the software environment,
- accomplish data mining tasks in an efficient and timely manner,
- interoperate and exchange information with other information systems,
- be adaptable to users' diverse requirements, and evolve with time.

Architecture: Typical Data Mining System



Integration of Data Mining and Data Warehousing

- **Data mining systems, DBMS, Data warehouse systems coupling**
  - No coupling, loose-coupling, semi-tight-coupling, tight-coupling
- **On-line analytical mining data**
  - integration of mining and OLAP technologies
- **Interactive mining multi-level knowledge**
  - Necessity of mining knowledge and patterns at different levels of abstraction by drilling/rolling, pivoting, slicing/dicing, etc.
- **Integration of multiple mining functions**
  - Characterized classification, first clustering and then association

Coupling Data Mining with DB/DW Systems

- No coupling—flat file processing, not recommended
- Loose coupling
  - Fetching data from DB/DW
- Semi-tight coupling—enhanced DM performance
  - Provide efficient implement a few data mining primitives in a DB/DW system, e.g., sorting, indexing, aggregation, histogram analysis, multiway join, precomputation of some stat functions
- Tight coupling—A uniform information processing environment
  - DM is smoothly integrated into a DB/DW system, mining query is optimized based on mining query, indexing, query processing methods, etc.

Coupling Data Mining with DB/DW Systems

No coupling—flat file processing, not recommended

- No coupling means that a DM system will not utilize any function of a DB or DW system.
- It may fetch data from a particular source (such as a file system)
- Process data using some data mining algorithms, and then store the mining results in another file.

Coupling Data Mining with DB/DW Systems

Drawbacks of no coupling

- First, a DB system provides a great deal of flexibility and efficiency at storing, organizing, accessing, and processing data.
- Without using a DB/DW system, a DM system may spend a substantial amount of time finding, collecting, cleaning, and transforming data.
- In DB and/or DW systems, data tend to be well organized, indexed, cleaned, integrated, or consolidated, so that finding the task-relevant, high-quality data becomes an easy task.

Coupling Data Mining with DB/DW Systems

Drawbacks of no coupling

- Second, there are many tested, scalable algorithms and data structures implemented in DB and DW systems.
- It is feasible to realize efficient, scalable implementations using such systems.
- Moreover, most data have been or will be stored in DB/DW systems.
- Without any coupling of such systems, a DM system will need to use other tools to extract data, making it difficult to integrate such a system into an information processing environment.
- Thus, no coupling represents a poor design

Coupling Data Mining with DB/DW Systems

Loose coupling - Fetching data from DB/DW

- Loose coupling means that a DM system will use some facilities of a DB or DW system. Fetching data from a data repository managed by these systems, performing data mining, and then storing the mining results either in a file or in a designated place in a database or data warehouse.
- Loose coupling is better than no coupling because it can fetch any portion of data stored in databases or data warehouses by using query processing, indexing, and other system facilities

Coupling Data Mining with DB/DW Systems

Loose coupling – Advantage

- It incurs some advantages of the flexibility, efficiency, and other features provided by such systems.

Loose coupling – Disadvantage

- Many loosely coupled mining systems are main memory-based. Because mining does not explore data structures and query optimization methods provided by DB or DW systems, it is difficult for loose coupling to achieve high scalability and good performance with large data sets.

Coupling Data Mining with DB/DW Systems

Semi-tight coupling — enhanced DM performance

- Provide efficient implement a few essential data mining primitives (identified by the analysis of frequently encountered data mining functions) in a DB/DW system, e.g., sorting, indexing, aggregation, histogram analysis, multi-way join, pre-computation of some essential statistical measures, such as sum, count, max, min, standard deviation, and so on.

Coupling Data Mining with DB/DW Systems

Semi-tight coupling — Advantage

- Some frequently used intermediate mining results can be pre-computed and stored in the DB/DW system.
- Because these intermediate mining results are either pre-computed or can be computed efficiently, this design will enhance the performance of a DM system.

Coupling Data Mining with DB/DW Systems

Tight coupling—A uniform information processing environment

- DM system is smoothly integrated into a DB/DW system.
- The data mining subsystem is treated as one functional component of an information system
- Data mining queries and functions are optimized based on mining query analysis, data structures, indexing schemes, and query processing methods of a DB or DW system.

Coupling Data Mining with DB/DW Systems

Tight coupling — Advantages

- With further technology advances, DM, DB, and DW systems will evolve and integrate together as one information system with multiple functionalities. This will provide a uniform information processing environment.
- This approach is highly desirable because it facilitates efficient implementations of data mining functions, high system performance, and an integrated information processing environment.

Coupling Data Mining with DB/DW Systems

- Loose coupling, though not efficient, is better than no coupling because it uses both data and system facilities of a DB/DW system.
- Tight coupling is highly desirable, but its implementation is nontrivial and more research is needed in this area.
- Semi-tight coupling is a compromise between loose and tight coupling. It is important to identify commonly used data mining primitives and provide efficient implementations of such primitives in DB or DW systems.