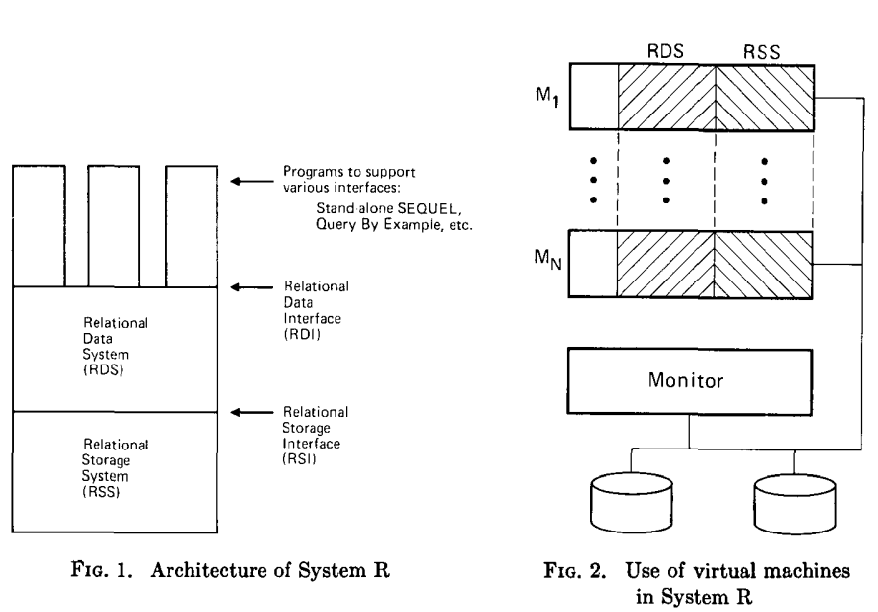
Architecture of System R



First, we will describe the system as seen by a single transaction.

Second, we will investigate its multiuser dimensions.

Figure 1 gives a functional view of the system including its major interfaces and components.

The Relational Storage Interface (RSI) is an internal interface which handles access to single tuples of base relations.

This interface and its supporting system, the Relational Storage System (RSS), is actually a complete storage subsystem in that it manages devices, space allocation, storage buffers, transaction consistency and locking, deadlock detection, backout, transaction recovery, and system recovery. Furthermore, it maintains indexes on selected fields of base relations, and pointer chains across relations.

The Relational Data Interface (RDI) is the external interface which can be called directly from a programming language, or used to support various emulators and other interfaces. It provides high level, data independent facilities for data retrieval, manipulation, definition, and control.

The Relational Data System (RDS), which supports the RDI, provides authorization, integrity enforcement, and support for alternative views of data.

The RDS maintains the catalogs of external names, since the RSS uses only system generated internal names. The RDS contains an optimizer which chooses an appropriate access path for any given request from among the paths supported by the RSS.

The current operating system environment for this experimental system is VM/370.

illustrates the use of many virtual machines to support concurrent transactions on shared data. For each logged-on user there is a dedicated database machine. Each of these database machines contains all code and tables needed to execute all data management functions; that is, services are not reserved to a centralized machine.

The RDI interfaces SEQUEL to a host programming language by means of a concept called a cursor, A cursor is a name which is used at the RDI to identify a set of tuples called its active set (e.g. the result of a query) and furthermore to maintain a position on one tuple of the set. The cursor is associated with a set of tuples by means of the RDI operator SEQUEL; the tuples may then be retrieved, one at a time, by the RDI operator FETCH.

use of many virtual machines to support concurrent transactions on shared data. For each logged-on user there is a dedicated database machine. Each of these database machines contains all code and tables needed to execute all data management functions; that is, services are not reserved to a centralized machine.

The provision for many database machines, each executing shared, reentrant code and sharing control information, means that the database system need not provide its own multitasking to handle concurrent transactions.

Monitor Machine, which contains many system administrator facilities. For example, the Monitor Machine controls logon authorization and initializes the database machine for each user. The Monitor also schedules periodic checkpoints and maintains usage and performance statistics for reorganization and accounting purposes

1. **Catalog:**
   * The catalog in System R refers to the data dictionary or metadata repository. It stores information about the structure of the database, including details about tables, views, indexes, constraints, and other database objects.
2. **Tuple Identifier:**
   * A tuple identifier is a unique identifier assigned to each row or tuple in a relation (table) of the database. It is used to distinguish one tuple from another and is often implemented as a combination of the page number and the position of the tuple within that page. Tuple identifiers are crucial for efficient retrieval and manipulation of data.
3. **Image:**
   * In the context of System R, an image refers to a representation of a relation or a set of related tuples. It is essentially the physical storage structure of the data. The image includes the actual data pages, indexes, and other structures used to store and access the tuples in the database.
4. **Clustering Image:**
   * A clustering image in System R is a storage organization technique where related tuples are physically stored together on the same data page or pages. This organization aims to improve the efficiency of retrieval operations for queries that involve multiple related tuples, as they can be accessed in a more sequential manner.
   * A clustered index is a type of database index that determines the physical order of data rows in a table based on the values of one or more columns. Unlike a non-clustered index, which stores a separate data structure with a mapping between index keys and the corresponding row identifiers, a clustered index reorganizes the actual data pages of the table to match the order of the index.
5. **View:**
   * In the context of a relational database, a view is a virtual table that is based on the result of a SELECT query. It does not store the data itself but provides a way to present a subset of the data from one or more tables in a relational database. Views can be used to simplify complex queries, encapsulate complex logic, and provide a security mechanism by restricting access to specific columns or rows.
6. **Cost-Based Query Optimization:**
   * Cost-based query optimization is a technique used by the database management system to evaluate and choose the most efficient execution plan for a given SQL query. The optimization process involves estimating the cost of different possible execution plans based on factors such as access methods, join algorithms, and index usage. The goal is to select the plan with the lowest estimated cost, which should result in the fastest query execution time.
7. **Access Path:**
   * An access path in the context of a relational database refers to the method or route used to retrieve data from a table. It includes considerations such as the use of indexes, scan methods, and join strategies. The selection of an appropriate access path is crucial for optimizing query performance, and it is typically determined by the query optimizer during the query planning phase.

These concepts were fundamental in the development of System R and have influenced the design and implementation of many subsequent relational database management systems.

A cursor in the context of databases is a programming construct that allows for the traversal and manipulation of records in a result set. It acts as a pointer to a specific row within a set of rows retrieved by a query. Cursors provide a way to iterate over the rows of a result set, enabling operations such as fetching, updating, and deleting individual records.

There are two main types of cursors:

1. **Implicit (or Forward-Only) Cursors:**
   * These cursors are implicitly created and used by default when a query is executed. They can only **move forward through the result set**, and they are often **read-only**. Once a row is fetched, it cannot be revisited. Implicit cursors are straightforward to use but may have limitations in terms of functionality.
2. **Explicit (or Scrollable) Cursors:**
   * Explicit cursors are explicitly declared by the programmer, providing **more control over** the traversal of the result set. They **can move both forward and backward th**rough the result set, allowing for greater flexibility. However, explicit cursors require **more complex syntax** and involve additional management by the programmer.