A HISTORY AND EVALUATION OF SYSTEM R

JINGZHI LIU
Outlines

• Background & Introduction
• Key Goals Established for System R
• Phase Zero : An initial Prototype
• Phase One : Construction of a Multiuser Prototype
• Phase Two : Evaluation
• Conclusion
Background & Introduction

• 1970: E.F.Codd (IBM)
  First time mentioned about the relational model concept in *A Relational Model of Data for Large Shared Data Banks*.

• 1974: Don Chamberlin, Ray Boyce (IBM)
  Through the practical System R project, published *SEQUEL: A Structured English Query Language*.

• Then the SQL was developed based on that paper.

• 1976: Don & Ray first published version *SQUEL/2* (SQL)

• 1980: Changed the name SQUEL to SQL.
Background & Introduction

- Key Words Collection from the papers: *Relational Model* and *SEQUEL*
Background & Introduction

- Key Words Collection : The History and Evaluation of System R
Background & Introduction

• Timeline of the System R Development

E. F. Codd: Relational Model 1970
Don Chamberlin, Ray Boyce: SEQUEL 1975
System R: Phase Zero 1974
Phase One 1975
System R: Phase Two 1976
Changed name to SQL 1978
Today’s SQL 1980

Time
Key Goals Established for System R:

• To provide a high-level, nonnavigational user interface for maximum user productivity and data independence.

• To support different types of database use including programmed transactions ad hoc queries, and report generation.

• To support a rapidly changing database environment could easily be added to and removed from the database without stopping the system.

• To support a population of many concurrent users, with mechanisms to protect the integrity of the database in a concurrent-update environment.
Key Goals Established for System R:

• To provide a means of recovering the contents of the database to a consistent state after a failure of hardware or software.

• To provide a flexible mechanism whereby different views of stored data can be defined and various users can be authorized to query and update these views.

• To support all of the above functions with a level of performance comparable to existing lower-function database systems.
Phase Zero: An Initial Prototype

- Phase Zero: For Single user.
- Involved the development of the SQL user interface and a quick implementation of a subset of system functions.
- XRM (Any Relationship Management), which is a single user method, was used as relational access method.
- An interpreter program was written in PL/I to execute statements in the high-level SQL (formerly SEQUEL) language on top of XRM.
Phase Zero:

- The Phase Zero implementation was primarily intended for use as a standalone query interface by end users at interactive terminals.
- XRM Storage Structure:
- XRM stores relations in the form of “tuples”, each of which has a unique 32-bit “tuple identifier” (TID).
Phase Zero: Valuable lessons

- Lesson 1: A better measure of cost would have been “number of I/Os”.
- Lesson 2: The optimizer was quite complex and was oriented toward complex queries.
- Lesson 3: Optimizer cost measure should be a weighted sum of CPU time and I/O count.
Phase One: Construction of a Multiuser Prototype

- Basic introduction
- The Compilation Approach
- RSS Access Paths
- The Optimizer
- Views and Authorization: major objectives
- The Recovery Subsystem
- The Locking Subsystem
Phase One:

- Phase One was constructed for multiuser.
- Phase One has the Locking Subsystem and Recovery Subsystem etc. compare to the Phase Zero.
- RSS (the Research Storage System), as an access method, was originally designed to support multiple concurrent users.
- RDS (the Research Data System) was an optimizing SQL processor, which runs on top of the RSS.
Phase One:

- The Compilation Approach

a) The process of creating and invoking an access module is illustrated in following figures.
Phase One:

- The Compilation Approach:

  System R records with each access module a list, named as “dependencies”, on database objects. And it is stored in the form of a regular relation in the system catalog.

- RSS Access Paths
  a) Access paths: index scans, relation scans, link scans.
  b) Disadvantages: Variable in length; longer than equivalent XRM records. The used values need to be represented many times.
  c) Advantages: All the data values of a record could be fetched by a single I/O.
Phase One:

• The Optimizer

1. To **minimize** the weighted sum of the predicted number of I/Os and RSS calls in processing an SQL statement.

2. The optimizer chooses to **scan** each table in the SQL query **two methods**:
   a) Join Method 1: Scan over the qualifying rows of table.
   b) Join Method 2: Often used when no suitable index exists.

• Views and Authorization:

  1. **major objectives**: Power, Flexibility
  2. Allow SQL query to be used as the definition of a **view** (SQL parse tree).
  3. Authorization subsystem is based on privileges, which is controlled by SQL **GRANT** and **REVOKE** statements.
Phase One:

- The Recovery Subsystem
  
  1. The key objective is provision of a means whereby the database may be recovered to a consistent state in the even of a failure.
  
  2. Three types of failure:
     
     a) The disk media fail
     b) The system fail: Recovery work Require that System R be reinitialized
     c) An individual transaction fail: Recovery takes place on-line

- The Locking Subsystem

Goal: To prevent interference among concurrent users.
Phase Two: Evaluation

- Basic introduction
- SQL: The suggestions for extensions and improvements RSS Access Paths
- The Compilation Approach compare to the others
- The Optimizer
- The Recovery Subsystem
- The Locking Subsystem
Phase Two: Basic introduction

- Phase Two was designed close to today’s SQL.
- SQL: The suggestions for extensions and improvements from the users comments.
  
a) An easy-to-use syntax when testing for the existence or nonexistence of a data item was required.

b) A means of searching for character strings who contents are only partially known was required.

c) Codd called an “outer join” as the need of some user applications

(ref: E.F.Codd. Extending the database relational model to capture more meaning)
Phase Two:

• The Compilation Approach compare to the others
  
a) Smaller cost for power, flexibility, and data independence of the SQL language
  
b) Simplifying effect on the system architecture.

• Available Access Paths
  
a) The principal access path used for retrieving data associatively by its value is the B-tree index.
  
b) Hashing and direct links (physical pointers from one record to another) were not utilized for user data.
Phase Two:

Fig: A B-Tree Index
Phase Two:

- The Optimizer

  a) The optimizer searches through a tree of path choices, computing estimated costs and pruning the tree until it arrives at a single preferred access path.

  b) Two basic assumptions of the optimizer

     1. The values in each column are uniformly distributed from some minimum to some maximum value.

     2. The distribution of values of the various columns are independent of each other.

- The Recovery Subsystem: Similar to Phase One

  The combined "shadow page" and log mechanism used in System R to against media, system, and transaction failures.
Phase Two:

- The Locking Subsystem:
  a) a choice of three levels of isolation from other users.
     1. Level 1: A transaction may read uncommitted data.
     2. Level 2: A transaction is protected against reading uncommitted data.
     3. Level 3 (system default): A transaction is guaranteed that successive reads of the same record will yield the same value.
  b) Most System R users ran their queries and application programs at Level 3.
Conclusions

• In particular, System R has illustrated the feasibility of compiling a very high-level data sublanguage, SQL, into machine-level code.
• Continuing research programs for this paper:
  1. Adaption of System R to a distributed database environment
  2. Extension of the optimizer algorithms to encompass a broader set of access paths.
Q&A
Thanks!