A HISTORY AND EVALUATION OF SYSTEM R

Presented by : Aditi Miglani
Student Number : 301309908

OUTLINE

- Data Independence
- The Big Question
- Navigational Systems
- Relational Systems
- System R
- The Three Phases
- Conclusion

DATA INDEPENDENCE

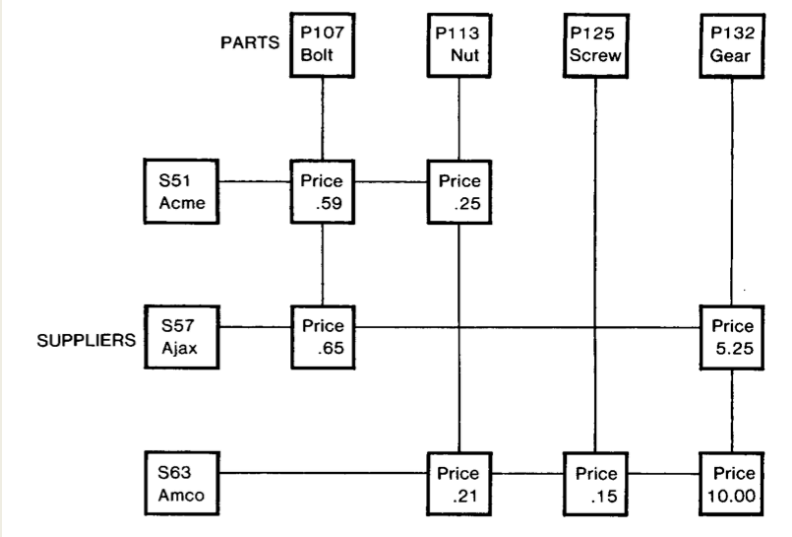
The immunity of applications to change in storage structure and access strategy

- End-users only deal with the information content of their data and not trivial things such as bits, pointers, arrays or lists associated with the data.
- They are unaware of any changes in the internal representation of data in the database.
- E.F. Codd introduced the Relational Data model as the next logical step towards data independence.

THE BIG QUESTION

Navigational Database vs Relational Database?

Navigational DB



A navigational database is one where records are found by following references or connections from other objects.

Relational DB

PARTS

PARTNO	NAME
P107	Bolt
P113	Nut
P125	Screw
P132	Gear

SUPPLIERS

SUPPNO	NAME
S51	Acme
S57	Ajax
S63	Amco

PRICES

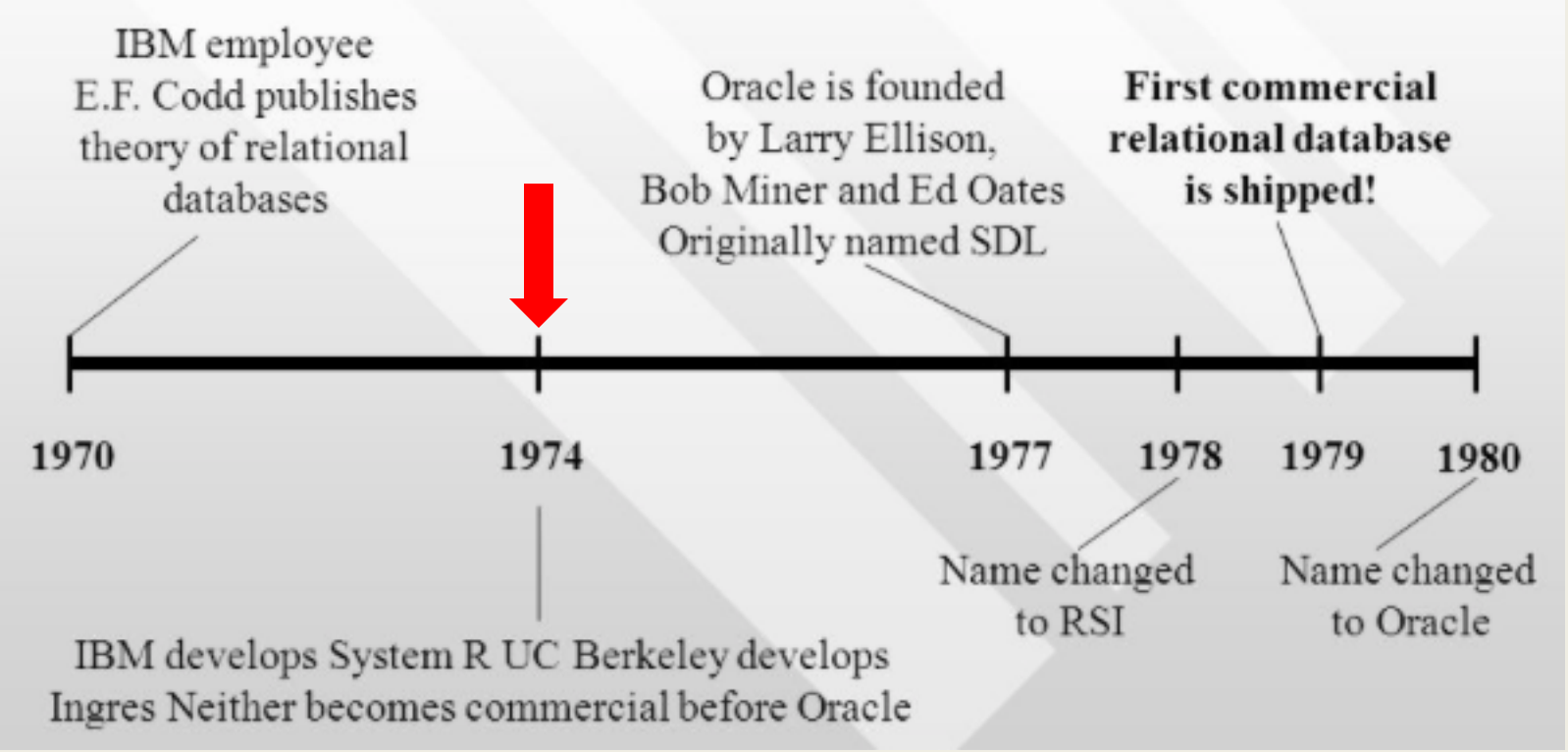
PARTNO	SUPPNO	PRICE
P107	S51	.59
P107	S57	.65
P113	S51	.25
P113	S63	.21
P125	S63	.15
P132	S57	5.25
P132	S63	10.00

A relational database stores information in the form of one or more tables of columns and rows, with a unique key identifying each row.

SYSTEM R

An experimental database system built as a research project at IBM San Jose Research Laboratory in the 1970's

- Introduced SQL to the world



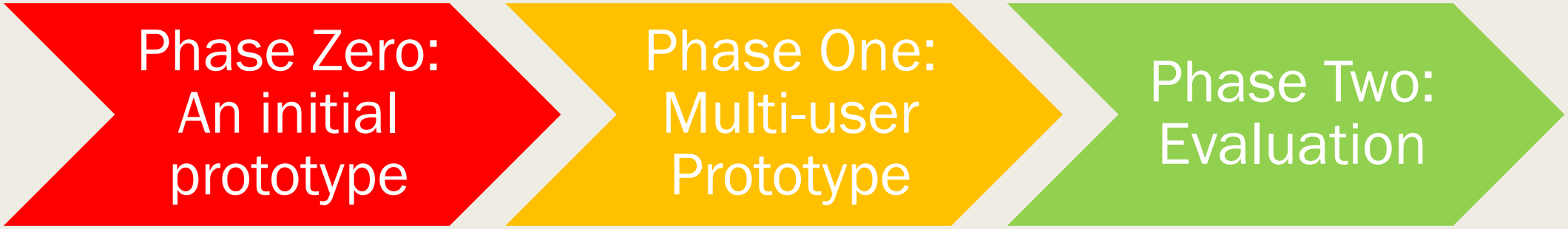
THE GOAL

To show that a RDBMS can incorporate the high performance and complete function required for everyday production use

OTHER KEY GOALS

- To provide a non-navigational user interface
- Support data independence for maximum user productivity
- To support a population of many concurrent users
- To provide a way to recover contents of database after a failure
- To allow storing different views of data and multi-user updates

THE THREE PHASES



Time : 1974 - most of 1975

Features :

- Development of a SQL user interface
- Quick implementation of a subset of functions for one user at a time.

Time : Most of 1976 - 1977

Features :

- Design and construction of a full-function, multi-user version
- Presenting an initial system architecture

Time : 1978 - 1980

Features :

- Involved experiments at the San Jose lab as well as several other user sites

PHASE ZERO : AN INITIAL PROTOTYPE

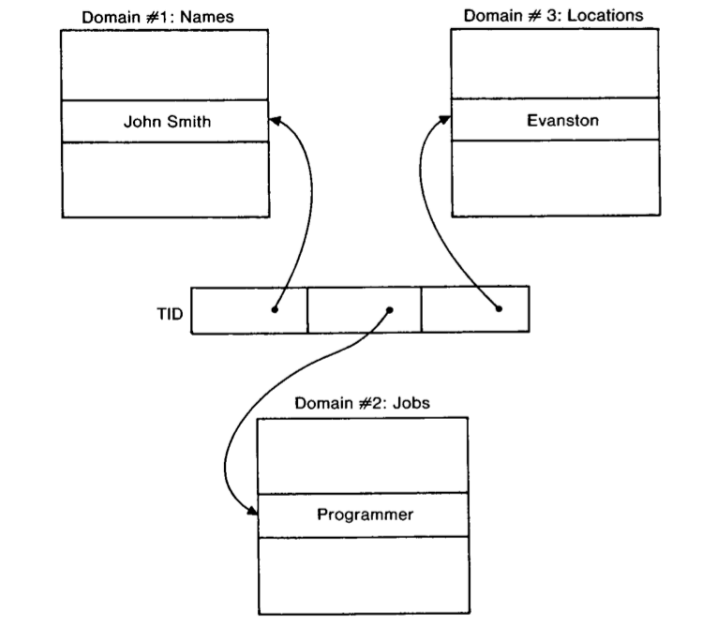
The intention was to learn from this initial prototype and then scrap the Phase zero code before construction of the more complete version of System R

Highlights

- The implemented subset of SQL language included query and updates of database and dynamic creation of new database relations
- Use of XRM, which is a single user access method and uses tuples to store relations
- Issues relating to concurrency or recovery were excluded from consideration in Phase Zero
- This phase had the ‘subquery’ construct implemented but not the ‘join’ construct
- A study was conducted during this phase to test the learnability and usability of SQL for humans

CHALLENGES and RESULTS

- The biggest challenge was the design of an optimizer algorithm for the efficient execution of SQL statements



The objective of the optimizer was to minimize the number of tuples fetched from the database in processing a query.

- It was felt that the phase zero optimizer was complex and was oriented towards complex queries. This fact was taken care of in further implementations

PHASE ONE : MULTI-USER PROTOTYPE

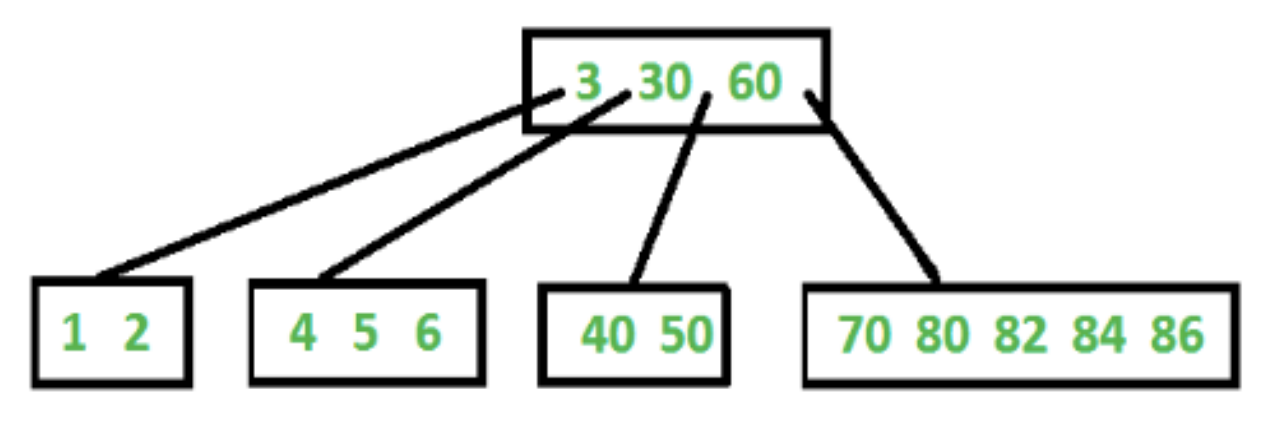
The goal of this phase was to construct a full-function multi-user version of System R

Highlights

- Involved the implementation of an access method called RSS and an optimizing SQL processor called RDS
- A locking subsystem was designed to prevent conflicts in case of multiple user updates
- A view and authorization subsystem was introduced to enhance the security of the system
- A recovery subsystem to ensure consistency in case of a hardware or software failure
- Support for both PL/I and COBOL application programs for users

CHALLENGES and RESULTS

- One of the biggest outcomes of this phase was that the makers decided to use RSS to store values in individual records of the database instead of in domains as in XRM for Phase Zero



RSS led to the use of indexes in the form of B-trees.
Indexes made it possible to scan a table in order or to directly access records that match a particular value.

- The locking subsystem adopted involved a hierarchy of locks with several different sizes of lockable units, from individual records to tables
- This involved the use of ‘intention’ and ‘exclusive’ locks by users

PHASE TWO : EVALUATION

This phase lasted almost 2.5 years and involved experiments performed on the system at the laboratory as well as actual use at a number of internal IBM sites and three selected customer sites

Highlights

General User Comments

- Performance characteristics and resource consumption was found to be satisfactory
- Interactive response slowed down during execution of very complex SQL statements

SQL Language

- Users praised uniformity of SQL syntax across different applications and ad hoc queries
- They made a number of suggestions for extensions and improvements to the language

Available access paths

- The use of indexes in B-trees was evaluated against other techniques like Hashing & Direct Links
- These were found useful only when a single record was to be fetched and not for multiple records

ADDITIONAL OBSERVATIONS

- Convoy Phenomenon

Experiments with the locking subsystem led to the discovery of a problem called the “Convoy Phenomenon”. This related to serializing all processes in the system, allowing them to acquire locks only once.

- Including a data communications front-end

For a canned transaction environment, it would be useful to include a data communications front-end to handle scheduling, logging and restart at the message level.

CONCLUSION

- System R has been able to demonstrate the feasibility of applying a relational database to a real production environment by taking care of :

- The evaluation of System R has led to a number of improvements, some of which have been already implemented and others are under study

THANK YOU

QUESTIONS?