

Inclusion of New Types in Relational Data Base Systems

Michael Stonebraker
Presenter: Jun Ma

11/15/09

EECS 584, Fall 2009

1

Outline

- Motivation
- Abstract data types
- New Access Methods
- Query Processing and Access Path Selection
- Conclusions

11/15/09

EECS 584, Fall 2009

2

Motivation

Traditional database system:

- Built-in data types: Integer, floating point number, character string, etc.
- Built-in operators: +, -, *, /
- Access method: hashing, B-tree

Geographic application:

- Data types: lines, polygons
- Operators: intersection, distance, containment
- Access method: R-trees, Grid files

Another example:

"Apr 15" - "March 15" = 31 days

11/15/09

EECS 584, Fall 2009

3

Motivation

A complete extended type system should allow:

1. The definition of **user-defined data types**.
2. The definition of **new operators** for these data types.
3. The implementation of **new access methods** for data types.
4. **Optimized query processing** for commands containing new data types and operators.

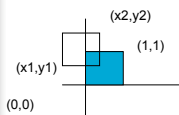
11/15/09

EECS 584, Fall 2009

4

A motivating example

Create box(id=i4, x1=f8, x2=f8, y1=f8, y2=f8)



Select *
From Box b
Where b.x2>0 and b.y2>0
and b.x1<1 and b.y1<1
and b.x1<b.x2 and b.y1<b.y2

Find all the boxes that overlap with unit square:
Retrieve (box.all) where not
(box.x2<=0 or box.x1>=1 or box.y2<=0 or box.y1>=1)

11/15/09

EECS 584, Fall 2009

5

Box data type

Create box(id=i4, desc=box)

a) Find all boxes overlap with unit square

Retrieve (box.all) where box.desc != "0,1,0,1"
!! -> Overlap operator
Access method: R-tree

Select *
From Box b
Where b.desc != "0,1,0,1"

b) Find all boxes with the area equals to 25

Retrieve (box.all) where box.desc AE "0,5,0,5"
AE -> area equals
Access method: B-tree (on area)

Select *
From Box b
Where b.desc AE "0,5,0,5"

c) Find all pairs of boxes which overlaps:

range of b1 is box
range of b2 is box
retrieve (b1.all,b2.all) where b1.desc != b2.desc
need query optimizer to construct an access plan

Select *
From Box b1,Box b2
Where b1.desc != b2.desc

11/15/09

EECS 584, Fall 2009

6

Operators for Boxes

Binary operator	symbol	left operand	right operand	result
overlaps	!!	box	box	boolean
contained in	<<	box	box	boolean
is to the left of	<L	box	box	boolean
is to the right of	>R	box	box	boolean
intersection	??	box	box	box
distance	*	box	box	float
area less than	AL	box	box	boolean
area equals	AE	box	box	boolean
area greater	AG	box	box	boolean

Unary operator	symbol	operand	result
area	AA	box	float
length	LL	box	float
height	HH	box	float
diagonal	DD	box	line

11/15/09

EECS 584, Fall 2009

7

- Define a new type
define type-name length=value,
input=file-name,
output=file-name

- Define the new operator
define operator token=value,
left-operand=type-name,
right-operand=type-name,
result=type-name,
precedence-level like operator-2,
file=file-name

an example: define operator token=!!
left-operand=box,
right-operand=box,
result=boolean,
precedence like *,
file=/usr/foobar

11/15/09

EECS 584, Fall 2009

8

Comment on prototype

- Dynamic linking (more flexibility but slow) is not preferable to static linking (less flexibility but fast).
- ADT routines provides a serious safety loophole (error or malicious), which can be solved by running routines in a separate address space (performance penalty is severe).
- Two environments for ADT procedure.
A protected environment for debugging purpose.
Unprotected environment

11/15/09

EECS 584, Fall 2009

9

Using existing access methods to implement new data types

B-tree

- P1) key-1 < key-2 and key-2 < key-3 then key-1 < key-3
- P2) key-1 < key-2 implies not key-2 < key-1
- P3) key-1 < key-2 or key-2 < key-1 or key-1 = key-2
- P4) key-1 <= key-2 if key-1 < key-2 or key-1 = key-2
- P5) key-1 = key-2 implies key-2 = key-1
- P6) key-1 > key-2 if key-2 < key-1
- P7) key-1 >= key-2 if key-2 <= key-1

TEMPLATE-1	AM-name	condition
	B-tree	P1
	B-tree	P2
	B-tree	P3
	B-tree	P4
	B-tree	P5
	B-tree	P6
	B-tree	P7

Not used internally
in the system

TEMPLATE-2	AM-name	opr-name	opr	left	right	result
	B-tree	=	opt	fixed	type1	boolean
	B-tree	<	opt	fixed	type1	boolean
	B-tree	<=	opt	fixed	type1	boolean
	B-tree	>	opt	fixed	type1	boolean
	B-tree	>=	opt	fixed	type1	boolean

opt/req
fixed/variable
type1/type2

Templates for Access Methods

11/15/09

EECS 584, Fall 2009

10

AM relation B-tree

AM	class	AM-name	opr	generic name	opr-id opr	Ntups	Npages
int-ops	B-tree	=	=	id1	N / Ituples	2	
int-ops	B-tree	<	<	id2	F1 * N	F1 * NUMpages	
int-ops	B-tree	<=	<=	id3	F1 * N	F1 * NUMpages	
int-ops	B-tree	>	>	id4	F2 * N	F2 * NUMpages	
int-ops	B-tree	>=	>=	id5	F2 * N	F2 * NUMpages	
area-op	B-tree	AE	=	id6	N / Ituples	3	
area-op	B-tree	AL	<	id7	F1 * N	F1 * NUMpages	
area-op	B-tree	AG	>	id8	F1 * N	F1 * NUMpages	

The AM Relation

- Ntups: number of tuples satisfy the query
- Npages: number of pages touched
- N: number of tuples in a relation
- NUMpages: number of pages of storage used by relation
- Ituples: number of index keys in an index
- F1=(value - low-key)/(high-key - low-key)
- F2=(high-key - value)/(high-key - low-key)
- Specify class to indicate what operator class to use (index on area-op)

11/15/09

EECS 584, Fall 2009

11

Using New Access Methods to support New Data Types

TEMPLATE-1	AM-name	condition
	hash	Key-1 = Key-2 implies Hkey1 = Hkey-2
	R-tree	Key-1 <= Key-2 and Key-2 <= Key-2 implies Key-1 <= Key-3
	R-tree	Key-1 <= Key-2 implies not Key-2 <= Key-1
	R-tree	Key-1 <= Key-2 implies Key-1 <= Key-2 or Key-1 = Key-2
	R-tree	Key-1 <= Key-1 UU Key-2
	R-tree	Key-2 <= Key-1 UU Key-2

Contained-in <<
Equals ==
Contained-in-or-equals <=
UU union of two boxes

TEMPLATE-2	AM-name	opr-name	opr	left	right	result
	hash	H	opt	fixed	int	int
	R-tree	<=	req	fixed	type1	boolean
	R-tree	=	opt	fixed	type1	boolean
	R-tree	<=	opt	fixed	type1	boolean
	R-tree	UU	req	fixed	type1	boolean

Templates for Access Methods

AM	class	AM-name	opr	generic name	opr-id	Ntups	Npages
box-ops	R-tree	=	=	id10			
box-ops	R-tree	<<	<<	id11			
box-ops	R-tree	<=	<=	id12			
box-ops	R-tree	UU	UU	id13			
hash-op	hash	H	H	id14			
hash-op	hash	H	H	id15			

The AM Relation

11/15/09

EECS 584, Fall 2009

12

Implementing New Access Methods

- Open(relation-name)
- Close(descriptor)
- Get-first(descriptor, OPR, value)
- Get-next(descriptor, OPR, value, tuple-id)
- Get-unique(descriptor, tuple-id)
- Insert(descriptor, tuple)
- Delete(descriptor, tuple-id)
- Replace(descriptor, tuple-id, new-tuple)
- Build(descriptor, keyname, OPR)

11/15/09 EECS 584, Fall 2009 13

Interface for transaction management and crash recovery

Logical logging of events
VS
physical logging of the changes of bits

1. Changes to the schema often require additional work besides changes to the system catalogs.
2. Some database updates are extremely inefficient when physically logged.

Log should be a collection of events.
For each event type T, REDO(T) UNDO(T)
LOG(event-type, event-data)

- replace a tuple
- insert a tuple at a specific tuple identifier address
- delete a tuple
- change the storage structure of a relation
- create a relation
- destroy a relation

11/15/09 EECS 584, Fall 2009 14

Concurrency control system

- Standard scheduler model
 - begin, abort, commit, savepoint: made by higher level software.
 - read, write: made by access method
- Special case parallelism on index or system catalog records
 - lock(object, mode)
 - unlock(object)
- Buffer management
 - get, fix, unfix, put, order

11/15/09 EECS 584, Fall 2009 15

Query Processing And Access Path Selection

- Stup: estimate the expected number of records satisfying:
 - where relname.field-name OPR value
 - selectivity=(1/ltuples) ELSE 1/10
- S: the expected number of records satisfying:
 - where relname-1.field-1 OPR relname-2.field-2
- Whether merge-sort is feasible for the operator ("=" and "<" are required)
- Whether hash-join is feasible for the operator (hash condition must be true with "=")

11/15/09 EECS 584, Fall 2009 16

Complete AE operator

Define operator token=AE.

- left-operand=box,
- right-operand=box,
- result=boolean,
- precedence like *,
- file=/usr/foobar,
- Stups=1,
- S=min(N1,N2),
- merge-sort with AL,
- hash-join

11/15/09 EECS 584, Fall 2009 17

Query processing plan


- relname.field-name OPR value
- Find boxes with area larger than 25

```

Select *
From Box b
Where b.desc AL "0,5,0,5"
  
```

If b.desc is the key and AL has been used to build the index B-tree, then we can use index to do the query. The expected number of index page and tuple accesses is given by the appropriate row in AM.

11/15/09 EECS 584, Fall 2009 18



Query processing plan

- rename-1.field-1 OPR rename-2.field-2
- Find boxes with equal area

```
Select *  
From Box b1,Box b2  
Where b1.desc AE b2.desc
```

Left-operand and Right-operand are all Box data type, AE has the merge-sort property with AL. Hash-join is also feasible joining strategy for this operator. The expect size of the result is S.

11/15/09

EECS 584, Fall 2009

19