

Relations

CSC 173

What is a Relation?

- A set (collection of unique) of n-ary tuples
 - Attributes – entries (fields in a tuple) drawn from a domain
 - Scheme of a relation – list of domain/attribute names
 - Database – collection of relations (scheme of which is the collection of schemes for the relations in the database)
- A relation on the set of domains D_1, D_2, \dots, D_n is a set of n-tuples, each of which is an element of the Cartesian product $D_1 \times D_2 \times \dots \times D_n$

Basic Operations

- $\text{Insert}(t, R)$ – insert tuple t in the relation R if not already present
- $\text{Delete}(P, R)$ – delete every tuple from relation R that satisfies the predicate P
- $\text{Lookup}(P, R)$ – return a relation consisting of the tuples in R that satisfy the predicate P

Implementing Relations

- Using
 - Arrays
 - Linked lists
 - Tree
 - Hash table
- How to sort or hash the relation? Depends on
 - Size of the relation
 - Keys
 - Sorts of lookup operations (queries)
 - Expected frequency of each sort of query

Database Design

- Principles behind scheme design
 - Two pieces of information that are independent should reside in two different relations
 - Two pieces of information that are related should reside in the same relation
- Key (index) selection is important
 - Dictates the types of queries that will be fast

How Can Relations and Relational Algebra Help?

- Conceptual framework for organizing information
 - helps you decide what information to keep together, based on expected relationships and queries
- Abstract language for expressing queries against which implementation can be compared, e.g., for completeness
- Algebra for expression simplification, to optimize queries for performance

Relational Algebra

- An algebra is a formal structure consisting of sets and operations on those sets
- Relational algebra is a formal system for manipulating relations
 - Operands are relations
 - Operations include usual set operation – union, intersection, and difference – as well as
 - Selection
 - Projection
 - Join

Algebraic Laws

- Join
 - Commutative
 - Non-associative except in special cases
- Selection
 - Selection pushing
 - Selection splitting
 - Commutative
- Projection
 - Projection pushing

Structured Query Language (SQL)

Language used by many modern databases for expressing queries

```
SELECT StudentId
FROM CSG
WHERE Course = "CS101"
```

List all student ids taking the course CS101

```
SELECT Day, Hour
FROM CR, CDH
WHERE CR.Course = CDH.Course AND Room = "Turing Aud."
```

The days and times that the Turing Auditorium is in use

- We have a formal mathematical framework that characterizes the transformations that are “semantically correct”
- We have a set of techniques, most often heuristic in nature, that drive decisions as to what is “computationally beneficial”

Example Types for Storing a Relation (using C)

```
typedef struct TUPLE *TUPLELIST;
struct TUPLE {
    int StudentId;
    char Name[30];
    char Address[50];
    char Phone[8];
    TUPLELIST next;
};
typedef TUPLELIST HASHTABLE[1000];

typedef struct NODE *TREE;
struct NODE {
    char Name[30];
    TUPLELIST toTuple;
    TREE lc;
    TREE rc;
};
```

Example: Bank Database

Relation: attribute1, attribute2, ...

- Deposit: branch_name, account_number, customer_name, balance
- Customer: customer_name, street, customer_city
- Borrow: branch_name, loan_number, customer_name, amount
- Branch: branch_name, assets, branch_city
- Client: customer_name, banker_name

Example: Course Database

- CSG: Course, StudentID, Grade
- SNAP: StudentID, Name, Address, Phone
- CDH: Course, Day, Hour
- CR: Course, Room

Example: Machine Database

- Workstations: Name, Room, Mem, OS, Model, Speed
- Printers: Name, Room, Status, Model
- Cycle Servers: Name, Status, FreeMem, #Users, OS
- File Servers: Name, Status, FreeDisk, OS
- Users: Id, Name, Status, FileServer, Console, Idle

Workstations

Name	Room	Mem	OS	Model	Speed
Crabgrass	727	128MB	Solaris	U10	333MHz
Fusion	727	176MB	MacOS	G3	500MHz
Welling.	717	512MB	Linux	P4	2GHz
chemical	735	96MB	MacOS	G3	300MHz
Saturn	727	64MB	Win2000	P2	265MHz

Printers

Name	Room	Status	Model
Chaucer	737	Up	LJ4
Shake	602	Up	LJ4
Keats	706	Up	LJ2
Poe	707	Down	LJ2
Picasso	737	Up	P560

Cycle Servers

Name	Status	Free Mem	#Users	OS
Heart	Up	95MB	23	Solaris
Brain	Up	1.2GB	4	Solaris
Aorta	Down	3.2GB	0	Solaris
Ganga	Down	2.0GB	0	Tru64
Tapti	Up	850MB	2	Tru64

File Servers

Name	Status	Free disk	OS
Nuclear	Up	5.8GB	MacOS
Steward	Up	8.5GB	Solaris
Slate	Up	452MB	Solaris
Butler	Down	12.6GB	Solaris

Users

Id	Name	Status	File Server	Console	Idle
scott	M. Scott	Faculty	Nuclear	Fusion	:15
sandhya	S. Dwarkadas	Faculty	butler	Welling.	0
stewart	C. Stewart	Grad	steward	zagreb	1:13
mchhabra	M. Chhabra	grad	butler	manila	:07
marty	M. Guenther	staff	nuclear	chemical	12:06:14
kparkins	K. Parkins	staff	butler	saturn	0
pderosa	P. Derosa	UG	steward	heart	:14