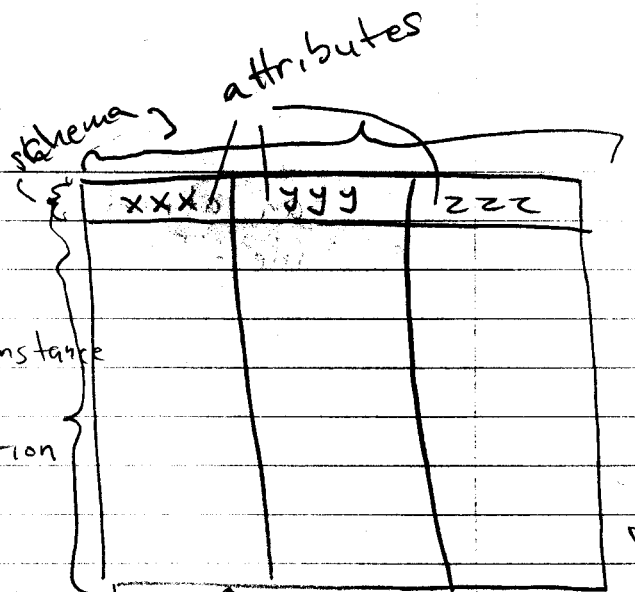


09/01/09 Devin Clark
Relational Databases
Relation means Table



1. Relational Schema vs rel. instance

1) Rel. Schema

design/Blue print of relation

3 Components

- Table Name = Account

- Attributes = 3 in the one here

- Domain: says each attribute could be of a kind of data, how the data is defined, and how it should be set, how high or low the info can go. Ex month = 11, there can be only 1-12, month \neq 13

Account(acc_#: int, branch = int, _____)

2) Relational Instance (i.e., the real table)

Instance: a collection of tables, each of which is built following the schema

2. Properties of Relations

We have concerns

1) Relations are unordered

2) Relations should have no duplicated tuples

fastest way to find them is $O(\log n)$, not so good

In practice, they can be kept.

Hashing will be used usually to help find duplicates

3) Each value in a tuple/row is ATOMIC

4) NULL Value, which captures unknown data

NULL \neq 3 is not correct, NULL can't be compared and nothing can be asked about it.

3. Constraints in a relation

Conditions that must be followed by All relational instances of a specific schema

1) Key Constraint

- a key is an i.d. of a row of tuples

- an attribute (or a set of attributes) whose value identifies a unique row
- no two rows in the table should have the same key value.

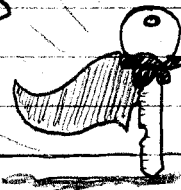
Ex: Employee (SSN, DOB, age, salary, dept, years)
 if there are two people with the same SSN, then all other values are the same, since the key says that no² SSN can be the same.

same { 101-23-4567 " " " " " "
 { 101-23-4567 " " " " " "

(SSN, age)
 (DOB, Name)

superkey = set of attributes that's an ID,
 some part of it can also be an ID

**SUPER
KEY**



Primary Key:
 picked from
 candidate keys