The Relational Model

Textbook 6.1-6.4

History

- Earlier models: Hierarchical, Network (ch. 10 & 11)
  - complex to implement and use; widely used
- Introduced by Codd & Date (IBM), 1970
  - Semantically poor; tractable to analyze
- Of primarily research interest until 1980's
  - by now is overwhelmingly the leading model

The Big Idea

- "Tables" ("relations") represent entities
  - made up of "rows" ("tuples") of fixed-sized attribute values, one row for each entity instance
- A "column" of a table holds all the values of a particular attribute
- A database typically contains a number of tables

That's about it!

A few more facts

- Tables are sets (mathematically)
  - no duplicate rows (implies each table has a key)
  - no ordering implied
  - RDBMS may allow ordering, duplicates
- "Null" attribute values allowed
- Caution: "relation" is not the same as E/R "relationship"

Simplicity of the Model

- Even compared to E/R...
  - no explicit relationships between entities
  - no compound attributes, no multivalued attributes ("first normal form")
  - no weak entities
  - no cardinality constraints
- Up to programmers to realize such semantics

Constraints in the Model

- Domain constraints:
  - that values of a column come from a defined domain
- Key constraints:
  - that there be a key
  - that primary key value not be null
- "Referential integrity" (in the case where R1 contains a "foreign key" of R2)
  - that a foreign key value in R1 always refer to some row in R2 with that key value

"Relating" Relations

- As noted: there is no direct analog of the E/R "relationship" (diamond shape)
- Possible solution: foreign key in one of the relations
  - awkward if not 1-1
- Common solution: A "relationship relation"
  - attributes are the keys of the two relations
  - tuples stand for pairs of related entities
  - As in E/R, the two entity sets may be the same

Relational DB Operations

- Keep in mind: relations are sets
- As long as two relations have the same columns:
  - set union, intersection, difference
  - result has same columns as inputs
- Cartesian product
  - has how many columns?
  - has how many rows?

Select and Project

- "Select"
  - take a subset of the rows based on some condition
- "Project"
  - take a subset of the columns
- We'll see notation later

"Join"

- Perhaps the most characteristic operation of the relational model
- Used constantly
- Challenging to implement efficiently
- The idea: a Cartesian product on two relations with *common attribute domains*, followed by a select based on those attributes.
  - especially common: equality match ("natural join")

Aggregate Functions

- Not set-theoretic
- COUNT, AVERAGE, MAX, MIN, etc.
- Actual query languages have many of these
- Rows can be "grouped" by some attribute and the function applied to each group (rows with common values of the grouping attribute)

Update Operations

- "Insert"
  - Add a new tuple to a table
- "Modify"
  - Change an attribute value in an existing tuple of an existing table
- Update operations compared to retrieval queries:
  - simple
  - in many applications, relatively infrequent

DDL Operations

- Used only the the DBA
- Create table
  - define attribute domains and names
  - declare constraints
- Declare schema
  - group relations together into a database
- Modify relation or schema
  - add/delete attributes, etc.

Relational Query Languages

- Operations are specified in a particular "query language"
- Relational Algebra: whole-table operations
- Relational Calculus: set construction
- SQL: set operations and procedural features, English-like syntax
- QBE: 2-dimensional set construction