Database Management Systems

Reagan Shirk

September 21, 2020

Relational Models

- I didn't watch the game, but just read that the Chargers punted on 4^{th}\$ & 1 on their own 34 in OT? Who made that decision?? What a terrible call
- Anyways... back to class stuff
- Relational models have attributes (columns) and tuples (rows)
  - Basically a fucking spreadsheet my dudes
- Lots of DBMS use relational models/concepts of relational models
- Creating a database with a relationship model requires telling the system what attributes you have, which means we need to study...

Attribute Types

- Your attributes have types (int, float, string, etc)
- The **domain** of the attribute is a set of **allowed values** for the attribute (Based on the application)
  - As a database designer, we don't have the authority to make this decision but we have to focus on the application
  - *null* is a member of every every domain and it causes complications all the god damn always
- Attribute values are normally **atomic**, meaning that we can't divide the attribute into a given number of things
  - ER model allows for composite and multivalue attributes, this model **does not**
- Examples:

Attribute	Domain
us_phone_number	Set of valued 10 digit phone numbers in USA
gpa	Possible values of computed grade point average, a value between 0 and 4
employee_age	Possible ages of employees in a company, e.g. a value between 16 and 70

Relation Schema and Instance

- You can convert the ER diagram into a *collection* of relation schema
  - Relation schema is just the list of column headers (attributes)
  - Need a schema before you can insert the data
- What does a relation schema look like?
  - *instructor(<u>ID</u>, name, dept_name, salary)
  - The underline is representing the primary key
- When you're given sets $D_1, D_2, \cdots, D_n$, a **relation** r is a subset of $D_1 \times D_2 \times \cdots \times D_n$
- **Relations are unordered** which means the order of the tuples doesn't matter, it be arbitrary

Keys

- Let $K \subseteq R$
- K is a **super key** of R if values for K are able to identify a unique tuple of each possible relation $r(R)$
  - An example of this is that $\{ID\}$ and $\{name, dept_name\}$ are both super keys of instructor because they both identify unique tuples
  - $\{dept_name\}$ is not a super key because that will return multiple tuples
- A **candidate key** is a super key that is minimal, i.e. $\{ID\}$ is a candidate key but $\{ID, name\}$ is not because you don't *need* name to get the unique tuple
  - Is it possible to have more than one candidate key for the same relation schema? Yes
  - In our example, $\{ID\}$ is a candidate key and so is $\{name, dept_name\}$ because both are in their most minimal form
  - Basically, a candidate key is a super key that doesn't have any redundancy
- One of the candidate keys is selected to be the primary key, but how do you determine which one?
  - Whichever is a unique identifier by itself
  - The order is

$$\text{super key} \in \text{candidate key} \in \text{primary key}$$

  - The primary key is a selected candidate key
  - The candidate key is a super key that has no subset which is also a super key
  - The super key is a unique tuple

Converting Entity Sets with Simple Attributes

- A strong entity set reduces to a schema with the same attributes
- A weak entity set becomes a relation schema that includes the primary key of the identifying strong entity set
- What are the relation schemas for the strong entity *course* and weak entity set *section*
  - course($\underline{course-id}$, title, credits)
  - section($\underline{course-id}, \underline{sec-id}, \underline{semester}, \underline{year}$)