CS143: Relational Model

Book Chapters

(5th) Chapters 1.3-7, 2.1, 3.1-2, 4.1
(6th) Chapters 1.3-6, 2.1-4, 3.1-2, 4.5
(7th) Chapters 2.1-4, 3.1-2

Things to Learn

- Data model
- Relational model
- Database construction steps

DataBase Management System (DBMS)

- Q: What is DBMS?
  - A system that manages data and provides six properties
    - Massive
    - Convenient
    - Efficient
    - Safe
    - Persistent
    - Multi-user

- High-level architecture of DBMS:
Data Model

- The way we model/conceptualize/visualize/represent data
- Need some representation to manage data in a computer
- Many different ways to model data
  - Example (Airline flight): Graph model
    * Node: city
    * Edge: flight between cities
    * Label on edge: flight time, etc.
    * Example standard: RDF (Resource Description Framework)
  - Example (Company hierarchy): Tree model
    * CEO → Presidents → Vice presidents → Department heads . . .
    * Example standard: XML (eXtensible Markup Language), JSON (JavaScript Object Notation)
  - Models to learn in the class: Relational and E/R model

Example to Use in the Class

- School information
  - Student(sid, name, age, GPA, address, . . .)
  - Class(dept, cnum, sec, title, instructor, . . .)
  - Enroll(sid, dept, cnum, sec)
  - . . .

Relational Model

- Example: Student(sid, name, address, age, GPA)

<table>
<thead>
<tr>
<th>sid</th>
<th>name</th>
<th>addr</th>
<th>age</th>
<th>GPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>301</td>
<td>John</td>
<td>183 Westwood</td>
<td>19</td>
<td>2.1</td>
</tr>
<tr>
<td>303</td>
<td>Elaine</td>
<td>301 Wilshire</td>
<td>17</td>
<td>3.9</td>
</tr>
<tr>
<td>401</td>
<td>James</td>
<td>183 Westwood</td>
<td>17</td>
<td>3.5</td>
</tr>
<tr>
<td>208</td>
<td>Esther</td>
<td>421 Wilshire</td>
<td>20</td>
<td>3.1</td>
</tr>
</tbody>
</table>

- All data is represented as relations (= tables)
- Each relation has a set of attributes (= columns)
- Each relation contains a set of tuples (= rows)
- Each attribute has a domain (= type)
  - Only atomic types
- Similar to Excel spreadsheet
History of Relational Model

- By far, the most significant invention in the history of DBMS
  - E.F. Codd, 1970
  - Completely revolutionized the field
  - Before it, network and hierarchical model: difficult to use and pose queries
  - Turing Award, 1981

- Extremely simple and strong mathematical foundation
- Supported by most DBMS systems
- An argument for simplicity

Concepts and Terminology

Schema
The structure of relations in database: relation name, attribute name, domain (optional).

- Example:
  - Student(sid, name, address, GPA, age)
  - Course(dept: char(2), cnum: int, sec: int, unit: int, title: char(100))
    char(2): string of length 2

Instance (= Data)
Actual contents (tuples) of relation (explain using the table example)

- Schema ≈ Type, Instance ≈ Value
- Schema ≈ Class, Instance ≈ Instance

Keys
- A set of attributes that are known to be unique in the relation
  - Student(sid, name, address, GPA, age)
  - Course(dept, cnum, sec, unit, instructor, title)
- Multiple keys possible
  - Course(dept, cnum, sec, unit, instructor, title)
  - Course(dept, cnum, sec, unit, instructor, title)
  - Course(dept, cnum, sec, unit, instructor, title)

- Q: When do we need keys? How can they be used?
Name Scope

- Names of relation: Unique across relations
- Names of attributes: Unique in a table, same name in different tables OK

Set semantics

- No duplicate tuples (different in SQL. More discussion later)
  - Q: Can a relation with no duplicates have no keys?
- Tuple order does not matter
  - Authors of a paper: Need explicit ordering
- Orders of attributes do not matter

Null value

- Common interpretation
  - Do not know
  - Do not want to say
  - Not applicable
  - Example: Student(id, dept, name, GPA) – before first quarter?
- Complications from Null
  - Example: Student(id, dept, name, age, GPA)
  - $Q_1$: Find students whose age $\geq$ 20. Susan’s age is Null. Susan in the result?
  - $Q_2$: Find students whose age < 20. Susan in the result?
  - $Q_3$: Find students whose age $\geq$ 20 or age < 20 and. Susan in the result?
    * But $Q_3 = Q_1 \cup Q_2$? Something is wrong.
- Relational algebra, SQL: 3-valued logic
  - Every condition is evaluated as True, False or Unknown
  - Various (arbitrary) rules to deal with anomalous situation
  - More discussion later
- Unfortunately, Nulls are very common in DB.
- Common sources of error in data analysis: “Pay attention to Nulls!”
Steps in Database Construction

Flow chart diagram

1. **Domain Analysis**: Understand application-domain semantics being captured
   - E/R diagram
   - discussed later

2. **Database design**: Design tables to capture the information
   - Relational design theory (functional dependency, normal form, etc)
   - discussed later if time and interest permit

3. **Table creation**: using Database Definition Language
   - DDL: A language to define relations and their characteristics:
     - Schema, integrity constraints, indexes, . . .

4. **Load**: typically bulk-load. insert tuple possible

5. **Query and update**: using Data Manipulation Language
   - DML: A language to query and update relations

SQL and DDL, Load, DML

What is SQL?

- Structured Query Language
- The standard language for interacting with all commercial RDBMS
- The history of SQL standard
  - SQL89: first standard
  - SQL92: the main and most widely-supported standard. several hundred pages
    - We will mainly use the standard SQL92 in class. Individual product uses slight variations of the standard. Some class query may not run on them.

- SQL has many components
  - DDL: Schema definition, constraints, indexes, . . .
  - DML: data retrieval, modification, . . .
  - Transactions, Authorization, . . .
- We learn schema definition part today.
Basic SQL Types

- Basic SQL types (commonly used subset)
  - String
    * `Char(n)` – padded fixed length
      • Padding character is system dependent (space for Oracle, auto-removed for MySQL)
    * `Varchar(n)` – variable length
  - Number
    * `Integer` – 32bit
    * `Decimal(5,2)` – 999.99
    * `Real, Double` – 32bit, 64bit
  - Datetime
    * `Date` – ‘2002-01-15’
    * `Time` – ‘13:50:00’
    * `Timestamp` – ‘2002-01-15 13:50:00’ (On MySQL, Datetime is preferred)

- Schema definition (table creation)
  - `Course(dept, cnum, sec, unit, instructor, title)`
    * `CREATE TABLE Course (`
      `  dept CHAR(2) NOT NULL,`
      `  cnum INTEGER NOT NULL,`
      `  sec INTEGER NOT NULL,`
      `  unit INTEGER,`
      `  instructor VARCHAR(30),`
      `  title VARCHAR(30),`
      `  PRIMARY KEY(dept, cnum, sec) )`
    * No Null in primary key

- `Course(dept, cnum, sec, unit, instructor, title)`
  `Course(dept, cnum, sec, unit, instructor, title)`
  `Course(dept, cnum, sec, unit, instructor, title)`
  * `CREATE TABLE Course (`
    `  dept CHAR(2) NOT NULL DEFAULT 'CS',`
    `  cnum INTEGER NOT NULL,`
    `  sec INTEGER NOT NULL,`
    `  unit INTEGER,`
    `  instructor VARCHAR(30),`
    `  title VARCHAR(30) DEFAULT,`
    `  PRIMARY KEY(dept, cnum, sec),`
    `  UNIQUE(dept, cnum, instructor),`
    `  UNIQUE(dept, sec, title) )`
  * One primary key per table
* Unique for other keys
* Primary key, unique are enforced through index (more discussion later)
* SQL92: No Null in primary key. Null OK for unique
  - MySQL: automatically add not null for primary key attributes
* DEFAULT for default values

- SQL for dropping a table
  - DROP TABLE Course

**Loading data**

- Vendor specific
- MySQL
  - LOAD DATA LOCAL INFILE <datafile> INTO TABLE Course
- Microsoft SQL Server
  - BULK INSERT Course FROM <datafile>

**Things to remember**

- Data model
- Schema
- Instance
- Relational model
  - relation, attribute, tuple, domain
  - key
  - null value
  - set semantics
- Database construction steps
  1. Domain analysis: E/R model, UML
  2. Database design: Database design theory
  3. Table creation: DDL
  4. Load
  5. Query and update: DML