CS143
Relational Model

Professor Junghoo “John” Cho
Data Model

Q: What is a data model? Why do we need it?

A:

- Data model is a way we model/conceptualize/represent data
- We need a concrete representation to manage data in a computer
- Many different ways to model data
  - Graph model, tree model, object model, ...
Graph Model (= Network Model)

• Nodes, edges and labels
• Airline flights

SFO

LAX

UA35, 3PM

AA24, 10AM

DL09, 10PM

NYC
Tree model (= Hierarchical Model)

- Nodes, edges, and labels arranged in a tree
- Company hierarchy
Relational Model

• All data is represented as a set of “tables”
• The “most popular” data model used for database systems
• Example: School information
  • Student(sid, name, age, GPA, address, ...)
  • Class(dept, cnum, sec, title, instructor, ...)
  • Enroll(sid, dept, cnum, sec)

<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>
</tbody>
</table>
Example: JSON (JavaScript Object Notation)

- Syntax to represent objects in JavaScript
  - `[{ “x”: 3, “y”: “Good”}, { “x”: 4, “y”: “Bad” }]`

- One of the most popular data-exchange formats over Internet
  - As JavaScript gained popularity, JSON’s popularity grew
  - Simple and easy to learn
  - Others popular formats include XML, CSV, ...
Basic JSON Syntax

• Supports basic data types like numbers and strings, as well as arrays and “objects”

• Double quotes for string: “Best”, “UCLA”, “Worst”, “USC”

• Square brackets for array: [1, 2, 3, “four”, 5]

• Objects: (attribute, name) pairs. Use curly braces
  • { “sid”: 301, “name”: “James Dean” }

• Things can be nested
  • { “sid” : 301,
    “name”: { “first”: “James”, “last”: “Dean” },
    “classes”: [ “CS143”, “CS144” ] }
Data Model of JSON?

• Q: What is JSON’s data model? Tree (hierarchical)? Graph (network)? Relational?

{ "sid" : 301,
  "name": { "first": "James", "last": "Dean" },
  "classes": [ "CS143", "CS144" ]
}
History of Relational Model

• By far, the most significant invention in the history of DBMS
  • E.F. Codd, 1970
  • Completely revolutionized the field
  • Turing Award, 1981

• Extremely simple
  • Another evidence of the power of a simple yet widely-applicable idea in computer science

• Initial research prototypes
  • IBM System R → IBM DB2
  • Berkeley Postgress → PostgreSQL

• IBM and Oracle: first commercial RDBMS vendor. Still dominates the market together with Microsoft
Relational Model: Terminology

- 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
- Data representation is very similar to Excel spreadsheet

<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>
</tbody>
</table>
Terminology

• **Schema**: the structure of relations in database
  • Relation name, attribute name, domain (optional)
  • Example: Student(sid, name, addr, age, GPA)

• **Instance (= Data)**
  • Actual content of relation
    • Schema $\approx$ variable type, Instance $\approx$ value
Terminology

• Keys: A set of attributes that uniquely identifies a tuple in a relation
  • Student(sid, name, address, GPA, age)
  • Course(dept, cnum, sec, unit, instructor, title)
  • Underline key attributes in schema

• Multiple keys are possible
  • 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

• Name of a relation is unique across relations
• Name of an attribute is unique in a table
  • Same attribute name in different tables is OK
Set Semantics

• No duplicate tuples are allowed in relational model
  • Duplicates tuples are allowed in SQL for practical reasons. More on this later
  • Q: Can a relation with no duplicates have no keys?

• Tuple order does not matter
• Attribute order does not matter
  • In SQL, attribute order does matter, but not in pure relational model
Null Value

• Common interpretation
  • Do not know/Do not want to say/Not applicable

• Example
  • Student(id, major, name, GPA) – What GPA value before the first quarter?
Complications from Null

• Student (id, major, name, 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>21</td>
<td>2.1</td>
</tr>
<tr>
<td>303</td>
<td>James</td>
<td>301 Wilshire</td>
<td>NULL</td>
<td>3.9</td>
</tr>
<tr>
<td>401</td>
<td>Susan</td>
<td>183 Westwood</td>
<td>17</td>
<td>3.5</td>
</tr>
</tbody>
</table>

• Q1: students with age >= 20
  \{John, James\}

• Q2: students with age < 20
  \{Susan, James\}

• Q3: students with age >= 20 or age < 20
  \{John, Susan & James\}

• Due to NULL, DBMS may return “unexpected” answers
Null and SQL 3-valued logic

• Every condition is evaluated as True, False or Unknown
• Concrete rules to deal with Null and Unknown values
• Nulls and SQL 3-valued logic adds significant complexity to DBMS implementation and execution
Five Steps in Database Construction

1. Domain Analysis
   - Entity-Relationship (ER) model
2. Database Design
   - Relational design theory (= normalization theory)
3. Table Creation
   - Data Definition Language (DDL)
4. Load
   - Bulk-load command
5. Query and Update
   - Data Manipulation Language (DML)
Structured Query Language (SQL)

• The standard language for interacting with RDBMS

• Many versions of SQL standard exists
  • SQL89 (Ansi SQL): first standard
  • SQL92 (SQL2): the main standard, several hundred pages
  • SQL3 (SQL99): no vendors supports it all exactly! 1600 pages
  • SQL4(SQL03): bug-fix release
  • ...

• In our lectures, we mostly use SQL92 standard
  • Individual DBMS product may use a slightly different syntax, but will be mostly the same
SQL

• SQL has many components
  • Data Definition Language (DDL): schema definition, constraints, ...
  • Data Manipulation Language (DML): query, modification, ...
  • Transaction, Authorization, ...

• We cover schema definition part in the rest of the today’s lecture
Basic SQL Data 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 – 32 bit
  • Decimal(5,2) – 999.99
  • Real, Double -- 32bit, 64bit

• Datetime
  • Date – ‘2010-01-15’
  • Time – ‘13:50:00’
  • Timestamp – ‘2010-01-15 13:15:00’
    • On MySQL, Datetime is preferred to Timestamp
SQL Table Creation

- Course(dept, cnum, sec, unit, instructor, title)
SQL Table Creation

- CREATE TABLE statement
- One PRIMARY KEY per table
  - UNIQUE for other “keys”
  - PRIMARY KEY and UNIQUE are enforced through index (more on this later)
- No Null for PRIMARY KEY attributes. Null OK for UNIQUE (in SQL92)
  - MySQL automatically adds NOT NULL to PRIMARY KEY attributes
- DEFAULT to set the default value for an attribute
- DROP TABLE statement for “deleting” a table
Bulk Loading Data into Table

• No SQL standard for bulk data loading
• MySQL (and Oracle)
  • LOAD DATA LOCAL INFILE <data-file> INTO TABLE <table-name>

```
load data local infile 'movies.data' into table Movies;
```
What We Learned

- Data model
- Schema and data instance
- Relational model
  - Relation, attribute, tuple, domain
  - Key
  - Null value
  - Set semantic
- 5 steps for database construction
  - Domain analysis, database design, table creation, load, query & manipulation
- SQL table creation and bulk data loading