The Relational Model

Database Systems Lecture 3
Natasha Alechina

In This Lecture

• Relational data integrity
• For more information
  • Connolly and Begg chapter 3
  • E.F. Codd’s paper
    ‘A Relational Model of Data for Large Shared Data Banks’ – now a link from the module web page, ~nza/G51DBS.

The Relational Model

• Introduced in E.F. Codd’s 1970 paper
  “A Relational Model of Data for Large Shared Databanks”
• The foundation for most (but not all) current database systems

• Concerned with 3 main things
  • Data structure (how data is represented)
  • Data integrity (what data is allowed)
  • Data manipulation (what you can do with the data)

The Relational Model

Relational Data Structure

• Data is stored in relations (tables)
• Each relation has a scheme (heading)
• The scheme defines the relation’s attributes (columns)
• Data takes the form of tuples (rows)

The Relational Model

New thing: scheme (and attributes)

Before... After

1  2
John Mary
Mark Jane

Just numbers of columns
Tuples

Attributes
Name Age

1  2
John Mary
Mark Jane

Just numbers of columns
Tuples

Attributes
Name Age

Unnamed and named tuples

A tuple is <John, 23>

A tuple is {(Name,John), (Age,23)}

The Relational Model
Not a big difference!

- There is no fundamental difference between named and unnamed perspectives on relations.
- We could have written tuples \(<a,b,c>\) as sets of pairs \(\{(1,a), (2,b),(3,c)\}\); only we know anyway in which order 1,2,3 go, so we can skip the numbers.
- Written as sets of pairs (partial functions), tuples can be written in any order, e.g. \(\{(3,c),(2,b),(1,a)\}\).

Relational Data Structure

- More formally -
  - A scheme is a set of attributes.
  - A tuple assigns a value to each attribute in its scheme.
  - A relation is a set of tuples with the same scheme.

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>23</td>
</tr>
<tr>
<td>Mary</td>
<td>20</td>
</tr>
<tr>
<td>Mark</td>
<td>18</td>
</tr>
<tr>
<td>Jane</td>
<td>21</td>
</tr>
</tbody>
</table>

Tuples, e.g. \(\{(Name, John), (Age, 23)\}, \{(Name, Mary), (Age, 20)\}, \{(Name, Mark), (Age, 18)\}, \{(Name, Jane), (Age, 21)\}\)

Relations

- Scheme is (ID, Name, Salary, Department)
- Attributes are ID, Name, Salary, and Department
- Degree is 4
- Tuple, e.g. (ID, A368), (Name, Jane Brown), (Salary, 22,000), (Department, Accounts)

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Salary</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>M139</td>
<td>John Smith</td>
<td>18,000</td>
<td>Marketing</td>
</tr>
<tr>
<td>M140</td>
<td>Mary Jones</td>
<td>22,000</td>
<td>Marketing</td>
</tr>
<tr>
<td>A368</td>
<td>Jane Brown</td>
<td>22,000</td>
<td>Accounts</td>
</tr>
<tr>
<td>P222</td>
<td>Mark Brown</td>
<td>24,000</td>
<td>Personnel</td>
</tr>
<tr>
<td>A367</td>
<td>David Jones</td>
<td>20,000</td>
<td>Accounts</td>
</tr>
</tbody>
</table>

Cardinality is 5

Relational Data Integrity

- Data integrity controls what data can be in a relation.
- **Domains** restrict the possible values a tuple can assign to each attribute.
- **Candidate and Primary Keys** identify tuples within a relation.
- **Foreign Keys** link relations to each other.

Attributes and Domains

- A **domain** is given for each attribute.
- The domain lists the possible values for that attribute.
- Each tuple assigns a value to each attribute from its domain.

Examples

- An ‘age’ might have to come from the set of integers between 0 and 150.
- A ‘department’ might come from a given list of strings.
- A ‘notes’ field might allow any string at all.

Candidate Keys

- A set of attributes in a relation is called a candidate key if, and only if,
  - Every tuple has a unique value for the set of attributes (uniqueness).
  - No proper subset of the set has the uniqueness property (minimality).

<table>
<thead>
<tr>
<th>ID</th>
<th>First</th>
<th>Last</th>
</tr>
</thead>
<tbody>
<tr>
<td>S139</td>
<td>John</td>
<td>Smith</td>
</tr>
<tr>
<td>S140</td>
<td>Mary</td>
<td>Jones</td>
</tr>
<tr>
<td>S141</td>
<td>John</td>
<td>Brown</td>
</tr>
<tr>
<td>S142</td>
<td>Jane</td>
<td>Smith</td>
</tr>
</tbody>
</table>

Candidate key: (ID, (First,Last)) looks plausible but we may get people with the same name. (ID, First), (ID, Last), and (ID, First, Last) satisfy uniqueness, but are not minimal. (First) and (Last) do not give a unique identifier for each row.
Choosing Candidate Keys

• Important: don’t look just on the data in the table to determine what is a candidate key
• The table may contain just one tuple, so anything would do!
• Use knowledge of the real world – what is going to stay unique!

Primary Keys

• One Candidate Key is usually chosen to be used to identify tuples in a relation
• This is called the Primary Key
• Often a special ID attribute is used as the Primary Key

<table>
<thead>
<tr>
<th>ID</th>
<th>First</th>
<th>Last</th>
</tr>
</thead>
<tbody>
<tr>
<td>S139</td>
<td>John</td>
<td>Smith</td>
</tr>
<tr>
<td>S140</td>
<td>Mary</td>
<td>Jones</td>
</tr>
<tr>
<td>S141</td>
<td>John</td>
<td>Brown</td>
</tr>
<tr>
<td>S142</td>
<td>Jane</td>
<td>Smith</td>
</tr>
</tbody>
</table>

We could choose either (ID) or (First, Last) as the Primary Key. ID is more convenient as it is a single column and we know it will always be unique (what happens if another John Smith is added?)

NULLs and Primary Keys

• Missing information can be represented using NULLs
• A NULL indicates a missing or unknown value
• More on this later...

Entity Integrity: Primary Keys cannot contain NULL values

Foreign Keys

• Foreign Keys are used to link data in two relations. A set of attributes in the first (referencing) relation is a Foreign Key if its value always either
  • Matches a Candidate Key value in the second (referenced) relation, or
  • Is wholly NULL
• This is called Referential Integrity

Foreign Keys - Example

<table>
<thead>
<tr>
<th>DID</th>
<th>DName</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>Marketing</td>
</tr>
<tr>
<td>14</td>
<td>Accounts</td>
</tr>
<tr>
<td>15</td>
<td>Personnel</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EID</th>
<th>EName</th>
<th>DID</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>John Smith</td>
<td>13</td>
</tr>
<tr>
<td>16</td>
<td>Mary Brown</td>
<td>14</td>
</tr>
<tr>
<td>17</td>
<td>Mark Jones</td>
<td>13</td>
</tr>
<tr>
<td>18</td>
<td>Jane Smith</td>
<td>NULL</td>
</tr>
</tbody>
</table>

(DID) is a Candidate Key for Department - Each entry has a unique value for DID

Employee (DID) is a Foreign Key in Employee - each Employee’s DID value is either NULL, or matches an entry in the Department relation. This links each Employee to (at most) one Department

Employee (ID) is a Candidate Key for Employee, and (Manager) is a Foreign Key, which refers to the same relation - every tuple’s Manager value is either NULL or matches an ID value
### Referential Integrity

- When relations are updated, referential integrity can be violated.
- This usually occurs when a referenced tuple is updated or deleted.
- There are a number of options:
  - **RESTRICT** - stop the user from doing it
  - **CASCADE** - let the changes flow on
  - **NULLIFY** - make values NULL

#### Referential Integrity - Example

**Department**

<table>
<thead>
<tr>
<th>DID</th>
<th>DName</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>Marketing</td>
</tr>
<tr>
<td>14</td>
<td>Accounts</td>
</tr>
<tr>
<td>15</td>
<td>Personnel</td>
</tr>
</tbody>
</table>

**Employee**

<table>
<thead>
<tr>
<th>EID</th>
<th>EName</th>
<th>DID</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>John Smith</td>
<td>13</td>
</tr>
<tr>
<td>16</td>
<td>Mary Brown</td>
<td>14</td>
</tr>
<tr>
<td>17</td>
<td>Mark Jones</td>
<td>13</td>
</tr>
<tr>
<td>18</td>
<td>Jane Smith</td>
<td>NULL</td>
</tr>
</tbody>
</table>

**Department**

<table>
<thead>
<tr>
<th>DID</th>
<th>DName</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Marketing</td>
</tr>
</tbody>
</table>

**Employee**

<table>
<thead>
<tr>
<th>EID</th>
<th>EName</th>
<th>DID</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>John Smith</td>
<td>16</td>
</tr>
<tr>
<td>16</td>
<td>Mary Brown</td>
<td>16</td>
</tr>
<tr>
<td>17</td>
<td>Mark Jones</td>
<td>16</td>
</tr>
<tr>
<td>18</td>
<td>Jane Smith</td>
<td>NULL</td>
</tr>
</tbody>
</table>

#### RESTRICT

- **RESTRICT** stops any action that violates integrity.
  - You cannot update or delete Marketing or Accounts.
  - You can change Personnel as it is not referenced.

#### CASCADE

- **CASCADE** allows the changes made to flow through.
  - If Marketing’s DID is changed to 16 in Department, then the DIDs for John Smith and Mark Jones also change.
  - If Accounts is deleted then so is Mary Brown.

#### NULLIFY

- **NULLIFY** sets problem values to NULL.
  - If Marketing’s DID changes then John Smith’s and Mark Jones’ DIDs are set to NULL.
  - If Accounts is deleted, Mary Brown’s DID becomes NULL.

#### Naming Conventions

- **Naming conventions**
  - A consistent naming convention can help to remind you of the structure.
    - Assign each table a unique prefix, so a student name may be stuName, and a module name modName.
- **Naming keys**
  - Having a unique number as the primary key can be useful.
    - If the table prefix is abc, call this abcID.
    - A foreign key to this table is then also called abcID.
Example

These entries are clearly related to the Student table

These entries are clearly related to the Module table

These entries are clearly related to tables other than Enrolment

The Relational Model

This Lecture in Exams

<table>
<thead>
<tr>
<th>DVD</th>
<th>Stars</th>
<th>Actors</th>
</tr>
</thead>
<tbody>
<tr>
<td>dvdID</td>
<td>dvdTitle</td>
<td>dvdPrice</td>
</tr>
<tr>
<td>1</td>
<td>The Matrix</td>
<td>14.99</td>
</tr>
<tr>
<td>2</td>
<td>Dracula</td>
<td>14.99</td>
</tr>
<tr>
<td>3</td>
<td>Amelie</td>
<td>17.99</td>
</tr>
</tbody>
</table>

The Relational Model

This Lecture in Exams

Give the two conditions that must be satisfied for a set of attributes to be a candidate key of a relation (2 marks)

Given the data shown above, what are the candidate key(s) of the tables DVD, Stars, and Actor? (3 marks)

I would prefer to ask not given the data, but given what you know about DVDs, Actors and Stars.

Explain, with reference to the relations given, the term entity integrity. (2 marks)

Explain, with reference to the relations given, the term referential integrity. (2 marks)

The Relational Model

Next Lecture

- Entity/Relationship models
- Entities and Attributes
- Relationships and Cardinality Ratios
- E/R Diagrams
- For more information
  - Connolly and Begg chapter 11

The Relational Model