Relational Database Model

and Introduction to SQL

© Department of Computer Science
Northern Illinois University
September 2014
Basic Structure

• Relation or Table
  – Named
  – NO repeating fields (no occurs clause in COBOL terminology)
  – Shown as Relation-Name (A1, A2, ….., An)

<table>
<thead>
<tr>
<th>Relation Name</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>A2</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
<td>An</td>
</tr>
</tbody>
</table>
Basic Structure

• Columns
  – named attributes
  – must be atomic values
  – values valid within a domain

Relation Name

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>..</th>
<th>..</th>
<th>..</th>
<th>..</th>
<th>..</th>
<th>..</th>
<th>An</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Basic Structure

- **Rows**
  - also called tuples
  - similar to record
  - must have primary key

<table>
<thead>
<tr>
<th>Relation Name</th>
<th>A1</th>
<th>A2</th>
<th>..</th>
<th>..</th>
<th>..</th>
<th>..</th>
<th>..</th>
<th>An</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Basic Structure

• Keys
  – Super Key:
    • an attribute or set of attributes that uniquely identify a tuple
    • every relation has at least one superkey, the set of all attributes
    • a relation can have more than one superkey
Basic Structure

• Keys
  – Candidate Key:
    • a minimum set of attributes that uniquely identify a tuple
    • a minimal super key
    • a relation may have more than one candidate key
  – Primary Key:
    • one and only one per relation.
    • a chosen candidate key
Basic Structure

• Keys Example
  – Employee (Emp-ID, Emp-Name, Emp-Birthdate, Emp-Address, Emp-Salary)
    • Super key:
      – Emp-ID, Emp-Address
      – Emp-Name, Emp-Birthdate, Emp-Salary
    • Candidate key:
      – Emp-ID
      – Emp-Name, Emp-Birthdate
    • Primary key:
      – Emp-ID
Basic Structure

• Keys Example
  – Employee-Project( Emp-ID, Project-ID, Emp-Title-Proj, Hours-Worked )
    • Super key:
      – Emp-ID, Project-ID, Emp-Title-Proj
      – Emp-ID, Project-ID, Hours-Worked
    • Candidate key:
      – Emp-ID, Project-ID
      – Project-ID, Emp-Title-Proj (assuming each employee has a different title within a project)
    • Primary key:
      – Emp-ID, Project-ID
Basic Structure

- **Foreign Key**
  - used to reference another relation
  - attributes of FK have same domain as the primary key of the home relation
  - example

<table>
<thead>
<tr>
<th>Section</th>
<th>Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sect-ID</td>
<td>Crse-ID</td>
</tr>
<tr>
<td>Sect-Time</td>
<td>Crse-Title</td>
</tr>
<tr>
<td>PK</td>
<td>PK</td>
</tr>
<tr>
<td>FK</td>
<td>FK</td>
</tr>
</tbody>
</table>

- PK
- FK
Basic Structure

- **Foreign Key**

<table>
<thead>
<tr>
<th>Course</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Crse-ID</td>
<td>Crse-Title</td>
</tr>
<tr>
<td>PK</td>
<td>PK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Student</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Student-SSN</td>
<td></td>
</tr>
<tr>
<td>PK</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>completed</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Crse-ID</td>
<td>Student-SSN</td>
</tr>
<tr>
<td>PK</td>
<td>PK</td>
</tr>
<tr>
<td>FK</td>
<td>FK</td>
</tr>
</tbody>
</table>
**Terminology**

- **Domain**
  - set of atomic valid values of one or more attribute
  - may be specified as a data type
- **Atomic values**
  - indivisible data values
- **Null value**
  - designates a MISSING attribute value
  - may or may not be allowed for an attribute
Null Values

• Does the value exist? (Y/N)
• Is the value known? (Y/N)
• Data – exists and known
• Missing – exists, but not known
• N/A – doesn't exist, but known not to exist
• Unknown – existence? value?
Terminology

• **Degree**
  - number of attributes (columns) in a relation
  - does not change dynamically

• **Cardinality**
  - number of tuples (rows) in a relation
  - changes dynamically with additions and deletions of tuples using DML
**Terminology**

- **Intention**
  - a named relation and its attribute names
  - also called schema of a relation
  - the DDL is used to modify the intention

- **Extension**
  - the data (tuples) in a relation
  - the state of a relation
  - the DML is used to modify the extension
Characteristics of a Relation

• Order Independence

• Two kinds
  – (1) ordering of tuples within a relation
    • do not have any particular order
    • considered an unordered set
Characteristics of a Relation

• Order Independence
  – (2) ordering of attributes within a relation
    • do not have any particular order as long as correspondence between the attribute and its values is maintained
  • Example
    
    Student(Stud-ID, Stud-Name, Stud-ID)

    Student(Stud-Address, Stud-Name, Stud-ID)
Relational Constraints

• Domain / Integrity Constraints
  – specify the valid values of each attribute
  – editing criteria
    • salary not > 100k
    • height < 8 feet
Relational Constraints

• Entity Integrity Constraint
  – states that no attribute of a primary key can contain a null value
    • Game (Date, Location, Time)
      – here neither Date nor Location nor both can ever contain a null value in this relation
Relational Constraints

- Referential Integrity Constraint
  - a foreign key can
    - EITHER contain an existing value of the PK in the home relation
    - OR contain a NULL value

<table>
<thead>
<tr>
<th>Sect-ID</th>
<th>Sect-Time</th>
<th>Crse-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>10:00am</td>
<td>C2</td>
</tr>
<tr>
<td>S2</td>
<td>2:00pm</td>
<td>C2</td>
</tr>
<tr>
<td>S3</td>
<td>3:00pm</td>
<td>C3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Crse-ID</th>
<th>Crse-Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Course 1</td>
</tr>
<tr>
<td>C2</td>
<td>Course 2</td>
</tr>
<tr>
<td>C3</td>
<td>Course 3</td>
</tr>
</tbody>
</table>
Relational Constraints

- Referential Integrity Constraint:
  - a foreign key can
    - EITHER contain a valid value of the PK in the home relation
    - OR contain a NULL value

<table>
<thead>
<tr>
<th>Sect-ID</th>
<th>Sect-Time</th>
<th>Crse-ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>10:00am</td>
<td>C2</td>
</tr>
<tr>
<td>S2</td>
<td>2:00pm</td>
<td>C2</td>
</tr>
<tr>
<td>S3</td>
<td>3:00pm</td>
<td>C3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Crse-ID</th>
<th>Crse-Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Course 1</td>
</tr>
<tr>
<td>C2</td>
<td>Course 2</td>
</tr>
<tr>
<td>C3</td>
<td>Course 3</td>
</tr>
</tbody>
</table>
Relational Operators

• Update operators
  – Insert
  – Delete
  – Modify

• Retrieval operators
  – Relational Algebra
  – Relational Calculus
  – SQL
• Name derived from Structured Query Language
• Comprehensive database language
  – DDL
  – DML
  – view definition
  – transaction control
• Can be embedded in a programming language
SQL

Data Definition Language (DDL)
CREATE
DROP
ALTER

Data Manipulation Language (DML)
SELECT
INSERT
UPDATE
DELETE

Data Control Language (DCL)
GRANT
REVOKE
### DDL in DB2

#### CREATE statement:
- CREATE STOGROUP
- CREATE DATABASE
- CREATE TABLESPACE
- CREATE TABLE
- CREATE INDEX
- CREATE VIEW
- CREATE SYNONYM

#### DROP statement:
- DROP STOGROUP
- DROP DATABASE
- DROP TABLESPACE
- DROP TABLE
- DROP INDEX
- DROP VIEW
- DROP SYNONYM

#### ALTER statement:
- ALTER STOGROUP
- ALTER TABLESPACE,  ALTER TABLE,  ALTER INDEX