Washington University in St. Louis

Engineering

Database Management Systems

- Relations and Constraints

Doug Shook

Relations

- A relation can be thought of as a set of tuples
  - Much like a table
  - Tuples contain attributes
  - Each tuple has an attribute that uniquely identifies it

- We use relations as a mathematical model of a database
  - All operations are performed on relations

Schema

- Schemas come in the format R(A1, A2, A3, …, An)
- Example:

  Student(id, fname, lastname, birthdate)

Tuples

- Does the order of a tuple matter?
- Relations are sets of tuples
  - Any problems with this?

Domain

- Every attribute has a domain associated with it
  - Data type
  - Set of possible values

- What are the domains for:

  - A student ID?
  - First and last names?
  - Phone numbers?

State

- The state of a relation is some subset of all the possible combinations of domains
- Sometimes called a “value” or “population”

Exercise

- Given the following schema:
  - R(A1, A2)
  - dom(A1) = (1, 2)
  - dom(A2) = (a, b, c)
- What is the domain of the entire relation?
- How many possible states are there?

Constraints

- Three main types:
  - Implicit
  - Explicit
  - Application based

Schema Integrity

- When using constraints for integrity, we have three types:
  - Key constraint
  - Entity integrity constraints
  - Referential integrity constraints

- Domain constraints are considered separately

Key constraints

- Super Key
  - One or more attribute that guarantees uniqueness for a set of tuples
- Key
  - "minimal" super key
- If a tuple has n attributes, what is the maximum number of possible superkeys?

Primary Keys

- From the list of candidate keys, we choose a primary key
  - How should we narrow down the candidates?
  - What will the primary key be used for?
- Entity integrity
  - The primary key attributes of each relation cannot be null

Schemas

- A schema consists of:
  - A set of relations
  - A set of integrity constraints
- Example:

S = {Student, Course, Department}

Database State

- The database state consists of a set of relation states
  - These relation states must satisfy integrity constraints
- Sometimes referred to as a snapshot, or instance
- How do we modify this state?

Referential Integrity

- Involves two relations
  - Primary key → foreign key
- The foreign key MUST exist in the primary key table

Exercise

- For each of the modification operations (INSERT, UPDATE, DELETE), determine whether it is possible for them to violate:
  - Domain Constraints
  - Key Constraints
  - Referential Integrity
  - Entity integrity

- If it is possible, explain how

Handling Violations

- What should we do if a violation occurs?

Constraints in SQL

- Domain constraints require triggers:

```
DELIMITER $$
CREATE TRIGGER pos_id BEFORE INSERT ON
 Students
FOR EACH ROW BEGIN
    IF NEW.sid < 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "sid must be positive";
    END IF;
END$$
DELIMITER ;
```

Constraints in SQL

- Foreign keys can be created before or after table creation:

```
CREATE TABLE Student (sid int,
    courseid int,
    FOREIGN KEY (courseid)
    REFERENCES Course(courseid));


ALTER TABLE Student
ADD FOREIGN KEY (courseid)
REFERENCES Course(courseid);
```

Exercise

- Add foreign keys to your course tracking database based on the design you came up with.

- Hint: If you need to add a column (such as a foreign key column) to a table, you can do so as follows:

```
ALTER TABLE Student
ADD COLUMN courseid int;
```

Exercise

- Go through your course tracking database and come up with some domain constraints. Add those constraints using triggers, then test them.