CS 327E Class 2

Sept 04, 2020

Relational Data Model

- Database = Collection of relations
- Relation = A table with columns (attributes) and rows (tuples)
- Column properties: named, domain, unordered
- Row properties: single-valued attributes, unique, unordered

  How do we enforce a unique row constraint?

- Referential integrity: Every non-null foreign key must match an existing primary key values.

Notation: `customer(id, first_name, last_name, …)`

SQL Queries: Basic Form

```
SELECT {c1}, {c2}, {c3}, ... {cn}
FROM {table}
WHERE {c1} > {c2}
ORDER BY {c1}, {c2}
```

Why MySQL?

- It’s been around for a long time
- Relational model
- Structured data
- Feature-rich SQL support
- Open-source
- Simple and easy-to-use
- Supports many languages
- Small to medium size data (< TB storage)
- Low to moderate QPS of reads and writes (10K)

Set up MySQL and Jupyter Notebooks on GCP

MySQL Guide:
https://github.com/cs327e-fall2020/snippets/wiki/MySQL-Setup-Guide

Jupyter Guide:

https://github.com/cs327e-fall2020/snippets/wiki/Jupyter-Setup-Guide

Exercise: MySQL database creation

Staging Schema

Current_Student(sid, fname, lname, dob, cno, cname, credits, grade)

New_Student(sid, fname, lname, dob)

Class(tid, instructor, dept, cno, cname, credits)

Relational Data Model Terminology

- Entity: An object or a thing
- Usually a noun
- Common examples: Person, Team, Product, Order, Shipment

Analogies with OOP:

- Entity: analogous to class
- Record: analogous to objects
- Attribute: analogous to members of an object

Questions:

- How do we represent relationships between entities?
- Can entities have methods in addition to members?

Relational Data Model Design Principles

P1. A table models a single entity and an entity is modeled by a single table.

P2. The collection of fields of an entity represent the attributes of that entity.

P3. Each field is given a primitive type that best fits its domain of values.

P4. Each table has a primary key (PK) which is made up of one or more fields that uniquely represent each record.

P5. A child table has a foreign key (FK) which references its parent's PK.

P6. A *m:n* relationship is modeled as a junction table.

Design Principles Applied to College Database:

How many violations can you find?

Classes		
	tid	VARCHAR
	instructor	VARCHAR
	dept	VARCHAR
	cno	VARCHAR
	cname	VARCHAR
	credits	INT

Current_Students		
	sid	VARCHAR
	fname	VARCHAR
	lname	VARCHAR
	dob	VARCHAR
	cno	VARCHAR
	cname	VARCHAR
	credits	INT
	grade	VARCHAR

New_Students		
PK	sid	VARCHAR
	fname	VARCHAR
	lname	VARCHAR
	dob	DATE

Design Principles Applied to College Database:

What can go wrong: data anomalies

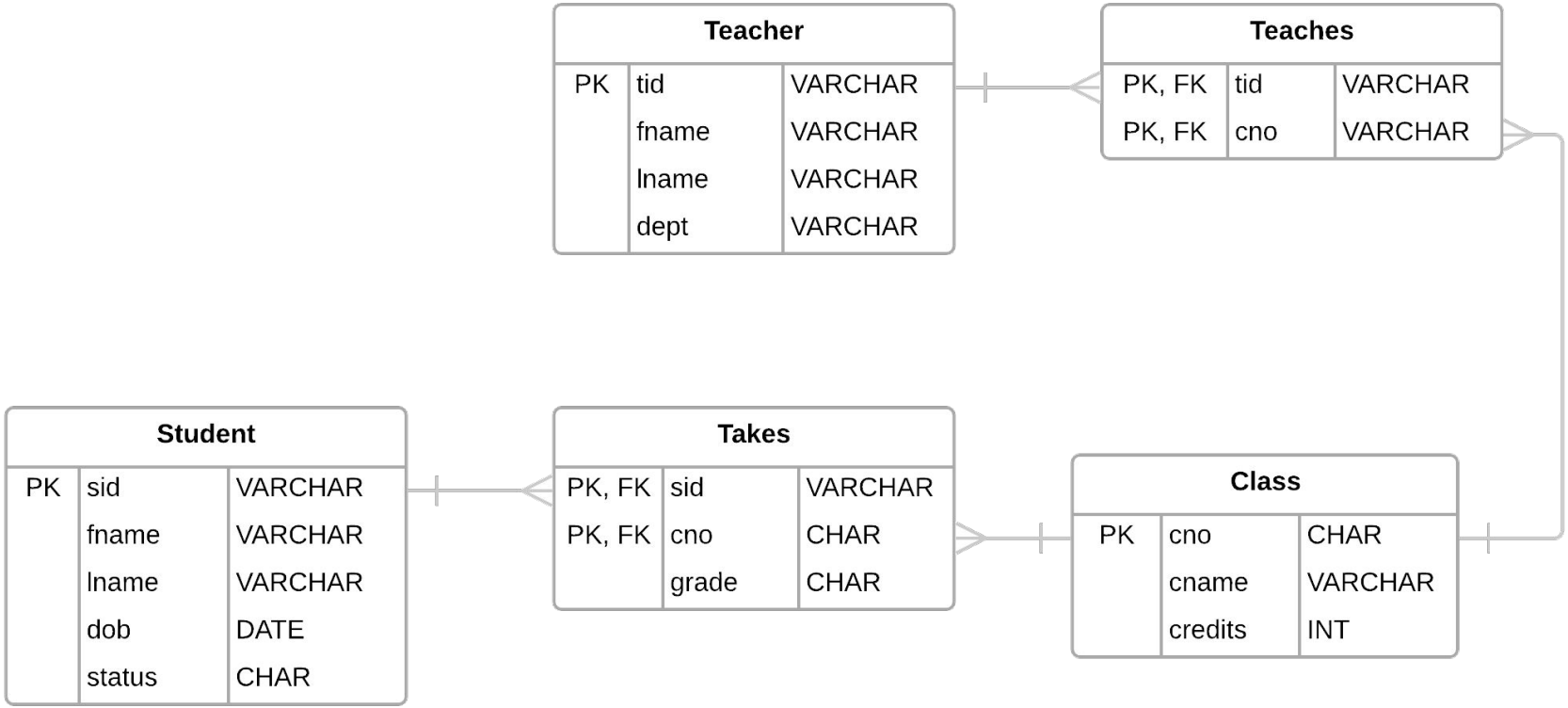
Classes		
	tid	VARCHAR
	instructor	VARCHAR
	dept	VARCHAR
	cno	VARCHAR
	cname	VARCHAR
	credits	INT

Current_Students		
	sid	VARCHAR
	fname	VARCHAR
	lname	VARCHAR
	dob	VARCHAR
	cno	VARCHAR
	cname	VARCHAR
	credits	INT
	grade	VARCHAR

New_Students		
PK	sid	VARCHAR
	fname	VARCHAR
	lname	VARCHAR
	dob	DATE

- Insert Anomaly
- Update Anomaly
- Delete Anomaly

Data Modeling Exercise



Common Transforms

- `CREATE TABLE T2 AS SELECT a, b, c FROM T1`

- ```
  SELECT a, b, c FROM T1
  UNION [DISTINCT]
  SELECT x AS a, y AS b, z AS c FROM T2
  ```

- ```
  SELECT a, b, c, 'some string' AS s FROM T1
  UNION ALL
  SELECT d, e, f, 'some string' AS s FROM T2
  ```

“CRUD” Operations

```
INSERT INTO {table} [({c1}, {c2}, {c3}, {cn})]
VALUES ({v1}, {v2}, {v3}, {vn});

UPDATE {table} SET {c1} = {v1}, {c2} = {v2}, …
  {cn} = {vn}
[WHERE {c1} = {v1}];

DELETE FROM {table} [WHERE {cn} = {vn}];
```

Practice Problems

Who takes CS327E *or* CS329E?

Who takes CS327E *and* CS329E?

Student(<u>sid</u>, fname, lname, dob, status)

Class(<u>cno</u>, cname, credits)

Teacher(<u>tid</u>, instructor, dept)

Takes(<u>sid</u>, <u>cno</u>, grade)

Teaches(<u>tid</u>, <u>cno</u>)

Second Question

Who takes CS327E *and* CS329E?

Student(sid, fname, lname, dob, status)
Class(cno, cname, credits)
Teacher(tid, instructor, dept)
Takes(sid, cno, grade)
Teaches(tid, cno)

Is this query a correct implementation?

```
SELECT sid
FROM Current_Student
WHERE cno = 'CS327E'
  AND cno = 'CS329E'
```

Project 1

http://www.cs.utexas.edu/~scohen/projects/Project1.pdf