Databases

Lecture 2

Relation Terminology

- Relation == 2D table
  - **Attribute** == column name
  - **Tuple/Record** == row (not the header row)
- Database == collection of relations
- Every relation has two parts:
  - **Schema** defines names of columns and their data types.
  - **Instance** defines the data rows/tuples/records.

Schema

- A schema is written by the name of the relation followed by a parenthesized list of attributes.
  - `Grades(First, Last, Course, Grade)`

`Grades`

First	Last	Course	Grade
Hermione	Granger	Potions	A
Draco	Malfoy	Potions	B
Harry	Potter	Potions	A
Ronald	Weasley	Potions	C

Domains

- Every attribute in a relation has a specific elementary data type, or **domain**.
  - string, int, float, date, time (no complicated objects!)

```
Grades(First:string, Last:string,
Course:string, Grade:char)
```

Degree and cardinality

First	Last	Course	Grade
Hermione	Granger	Potions	A
Draco	Malfoy	Potions	B
Harry	Potter	Potions	A
Ronald	Weasley	Potions	C

- **Degree/arity** of a relation is the number of attributes in a relation.
- **Cardinality** is the number of tuples in a relation.

Keys to a good relation(ship)

Keys of a relation

- Keys are a kind of **integrity constraint**.
- A set of attributes K forms a key for a relation R if:
  - we forbid two tuples in an instance of R to have the same values for all attributes of K.

First	Last	Course	Grade
Hermione	Granger	Potions	A
Draco	Malfoy	Potions	B
Harry	Potter	Potions	A
Ronald	Weasley	Potions	C

```
Grades(First, Last, Course, Grade)
```

Keys of a relation

- Keys help associate tuples in different relations.

Students

SID	First	Last
123	Hermione	Granger
111	Draco	Malfoy
234	Harry	Potter
345	Ronald	Weasley

Grades

SID	CRN	Grade
123	777	A
111	777	B
234	777	A
345	777	C

Classes

CRN	Title	Semester	Year
777	Potions	Fall	1997
888	Potions	Spring	1997
999	Transfiguration	Fall	1996
789	Transfiguration	Spring	1996

Example

- Let's expand these relations to handle the kinds of things you'd like to see in BannerWeb.
- Keep track of students, professors, courses, who teaches what, enrollments, pre-requisites, grades, departments & their chairs.
  - Only one chair per department.
  - Student cannot enroll in multiple copies of the same course in one semester.
  - Other constraints that are logical.

Relational algebra

- Language for querying a relational database.
  - Basis for SQL.
- Why?
  - Less powerful than C++/Python is a good thing!
  - Easy to learn
  - Easy for DBMS to optimize

Review of relations

- Relation is a subset of a cross-product.

Example:

S = {Harry, Ron, Draco}

C = {Potions, Transfiguration, Charms, Herbology}

Define a relation R:

$$R = \{(Harry, Potions), (Hermione, Potions), (Hermione, Charms)\}$$

Notice that $R \subseteq S \times C$

Students

SID	First	Last
100	Hermione	Granger
101	Draco	Malfoy
102	Harry	Potter
103	Ronald	Weasley

Grades

SID	CRN	Grade
100	900	A
101	900	B
102	900	A
103	900	C
100	901	A
100	902	B
101	903	F

Classes

CRN	Title	Teacher	Semester	Time
900	Potions	Snape	F97	9am
901	Transfiguration	McGonagall	F97	10am
902	Divination	Firenze	F97	3pm
903	Divination	Trelawney	F97	9am
904	Def Dark Arts	Snape	F99	9am
905	Def Dark Arts	Quirrell	F97	3pm