CS354: Database

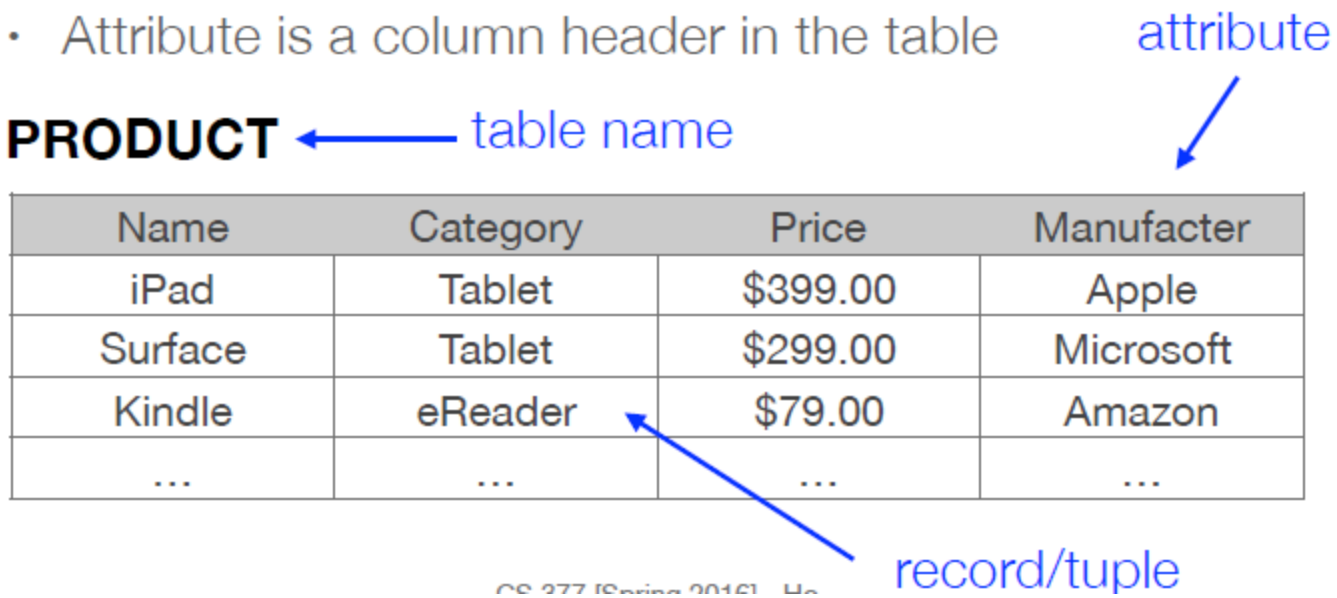
Relational Model

- First formal database model
- Introduced by Ted Codd in 1970
- Conceptual basis of relational databases
  - Simple and based on the mathematical relations
  - Declarative method for specifying data and queries
- Previous models include hierarchical and network models

Relation

Data is stored in **tables** (**relations**)

- Tuple is a row in the table
- Attribute is a column header in the table



Relation Definitions

- **Domain**: set of atomic values that are assigned to an attribute (e.g., name: string, category: string, price: real)
- **Relation Schema** $R(A_1, A_2, \ldots, A_n)$: made of of a relation name R and a set of attributes $A_1, A_2, \ldots, A_n$ (e.g., Product(name, category, price, manufacturer)
  - In practice, the domain is added for each attribute
- **Degree of a relation**: number of attributes in the relation schema
  - this is different than the degree in ER model!

Schema and Instances

- **Database schema**: a collection of relation schemas
- **Instance of a relation**: set of tuples or records
- **Instance of a database**: a collection of relation instances
- Can view schemas as types while instances as values in a programming language
- Schemas are stable over long periods of time while instance changes constantly with data inserts, updates, and deletions

STUDENT

Name	StudentNumber	Class	Major

COURSE

CourseName	CourseNumber	CreditHours	Department

PREREQUISITE

CourseNumber	PrerequisiteNumber

SECTION

SectionIdentifier	CourseNumber	Semester	Year	Instructor

GRADE_REPORT

StudentNumber	SectionIdentifier	Grade

Schema diagram

STUDENT	Name	StudentNumber	Class	Major
	Smith	17	1	CS
	Brown	8	2	CS

COURSE	CourseName	CourseNumber	CreditHours	Department
	Intro to Computer Science	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Database	CS3380	3	CS

SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
	85	MATH2410	Fall	98	King
	92	CS1310	Fall	98	Anderson
	102	CS3320	Spring	99	Knuth
	112	MATH2410	Fall	99	Chang
	119	CS1310	Fall	99	Anderson
	135	CS3380	Fall	99	Stone

GRADE_REPORT	StudentNumber	SectionIdentifier	Grade
	17	112	B
	17	119	C
	8	85	A
	8	92	A
	8	102	B
	8	135	A

PREREQUISITE	CourseNumber	PrerequisiteNumber
	CS3380	CS3320
	CS3380	MATH2410
	CS3320	CS1310

A database that stores student and course information.

Relational Model Notation

Notation	Description
$R(A_1, A_2, \ldots, A_n)$	Relation schema R of degree n
Q, R, S	Relation names
q, r, s	Relations
t, u, v	Tuples
$t(a_1, a_2, \ldots, a_n)$	tuple t of a relation
$t[A_i]$	the value of the attribute A_i in the tuple t
$t[A_i, A_j, A_k]$	value of the attributes A_i, A_j ,A_k in the tuple t

Relational Model Constraints

- Restrictions on actual values in a database
- **Inherent model-based constraints** or **implicit constraints**: inherent in the data model (e.g., no duplicate tuples)
- **Schema-based constraints** or **explicit constraints**: can be directly expressed in schemas of the data model
- **Application-based / semantic constraints**, or **business rules**: cannot be directly expressed in schemas and can only be enforced and expressed in the application program

Schema-based Constraints: Domain Constraints

- Value of each attribute A must be an atomic value from the domain of A
- Typical data types associated with domains
  - Numeric data types for integers and real numbers
  - Characters, fixed-length or variable-length strings
  - Booleans
  - Date, Time, Timestamp
  - …

Schema-based Constraints: Key Constraints

- No two tuples can have the same combination of values for all their attributes
- **Superkey**: set of attributes in a relation R such that no 2 different tuples will have the same values for that set of attributes

$$\forall t_1, t_2 \in R : t_1[SK] \neq t_2[SK]$$

- **Key**: minimal set of attributes in relation R such that no 2 tuples have the same values (i.e., key is a minimal superkey)
- **Candidate key**: any key

Schema-based Constraints: Key Constraints (2)

- **Primary key**: key chosen to be used to identify tuples in a relation
  - Once chosen, you must use that primary key throughout the database
  - Other candidate keys are unique keys
  - Every relation schema must have a primary key
- **Foreign key**: set of attributes inside some relation *R1* that is a primary key of another relation *R2*

Example: Keys

primary key candidate key

PERSON

PID	SSN	Name	Address
52032	111-12-2345	John Doe	123 My Street
12345	444-23-1234	Jane Smith	555 South Street
79823	555-67-8910	Tom Thumb	224 First Street
…	…	…	…

primary key foreign key

PURCHASE

TID	PID	Product	Price
123456778	52032	iPad Air 2	$399.00
123470901	52032	Kindle	$79.00
234096701	79823	Surface	$499.00
…	…	…	…

Schema-based Constraints: Entity Integrity

- The attribute values of the primary key cannot have NULL values
- Primary key is used to identify a tuple
- NULL value means not applicable or not available which hinders the ability to identify a tuple

PERSON

PID	SSN	Name	Address
52032	111-12-2345	John Doe	123 My Street
NULL	444-23-1234	Jane Smith	555 South Street
…	…	…	…

Not allowed!

Schema-based Constraints: Referential Integrity

- A tuple in one relation (t_1 in R_1) that refers to another relation (t_2 in R_2) must refer to an existing tuple in that relation (t_2 must exist)

$$t_1[FK] = t_2[PK]$$

- R_1 is the referencing relation and R_2 is the referenced relation

tuple must exist in PERSON table

PURCHASE

TID	PID	Product	Price
123456778	52032	iPad Air 2	$399.00
123470901	52032	Kindle	$79.00
234096701	79823	Surface	$499.00
...	...	...	...

ER Model vs Relational Model

ER model (conceptual model)

- Several concepts: entities, relationships, attributes
- Well-suited for capturing application requirements
- Not well-suited for computer implementation

Relational model (implementation model)

- Single concept: relation (not same as mathematical concept!)
- Everything is represented with a collection of tables
- Well-suited for efficient manipulations on computers