

Introduction to the Relational Model

The Relational Model

- Most used model for databases
- Very simple model
- Query with high-level languages: simple, yet expressive
- Efficient implementations

Relations

- Database = set of named relations (or tables)
- Each relation has a set of named **attributes** (or columns)
- The number of attributes is the arity of the relation

Student Relation

ID	Name	Gpa	photo	
-----	-----	-----	-----	

College Relation

State	Name	enroll	
-----	-----	-----	

Tuple

- Each **tuple** (or **row**) has a value for each attribute
 - No specific order between them
- The number of tuples is the cardinality of the relation
- Each attribute has a **type** (or a **domain**)
 - Set of possible values. Examples: integer, text, ...

Schema vs Instance

Schema

- structural description of relations in the database
- name, attributes and types of those attributes
- typically set up in advance

Instance

- actual contents at given point in time

- change over time

NULL value

- Special value for "unknown" or "undefined"
- Useful but one has to be careful when querying relations with NULL value

Key

- Set of one or more attributes whose combined values are unique within a relation
- Often denoted by underlying the set of key attributes
- Importance:
 - Identify specific tuples; efficiency; refer to tuples of another relation

Foreign Key

- An attribute (or set of attributes) that always matches a key attribute in another relation
- Often denoted by an arrow pointing to the name of the relation being referenced

Relational Notation

- Student(ID, name, GPA, country->Country)
- Classroom(building, number, capacity)
- Country(ID, name)
- In foreign keys, the name of the attribute can be different from the referenced attribute/relation

Composite Key

- A composite key is a multi-column primary-key or foreign-key
 - Classroom(building, number, capacity)
 - Professor(ID, name, building->Classroom.building, number->Classroom.number)