Lecture 2 Relational Model

CS211 - Introduction to Database

Outline

- Review: terms in Lecture 1
- Conceptual foundation of relational model
- Some characteristics of a relation
- Keys, Foreign keys, and integrity constraints

Review: Terms in Lecture 1

- Database System
  - Database
  - Database Application
  - Database Administrator
  - Database Management System
  - Computer System (computer, network, os, i/o device etc.)

- Relation model, Schema, Table/Relation
  - Relational model is a data model
  - Data model is the structure of schema
  - Schema is the structure of table/relation
  - Table or relation contain data instance

Review: Terms in Lecture 1

- SQL
  - Data Manipulate Language
  - Data Definition Language
  - Data Control Language
- No-SQL
  - Data format is not table
- 2-tier vs. 3-tier database architecture

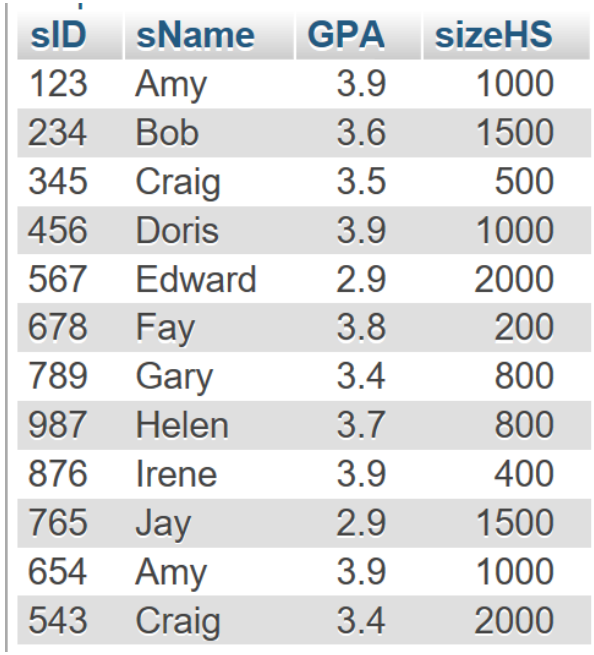
- 3 tiers, 2 mappings, 2 independence
  - External
    - Views for users (subset of tables)
  - Conceptual / Global
    - Tables of entire database
  - Internal
    - Storage of data
  - E-C Mapping
    - Logic data independence
  - C-I Mapping
    - Physical data independence

Conceptual foundation of relational model

CS211 - Introduction to Database

Relational Model

- Proposed by E.F.Codd in 1970
- Relational model based on
    - Table used to implement relation
    - Set Theory used to define data manipulation
    - First order predicate logic used to define integrity constraints



sID	sName	GPA	sizeHS
123	Amy	3.9	1000
234	Bob	3.6	1500
345	Craig	3.5	500
456	Doris	3.9	1000
567	Edward	2.9	2000
678	Fay	3.8	200
789	Gary	3.4	800
987	Helen	3.7	800
876	Irene	3.9	400
765	Jay	2.9	1500
654	Amy	3.9	1000
543	Craig	3.4	2000

- Set theory: relational algebra
    - set a union set b
- Integrity Constraints
    - GPA should less than or equal to 4.0
    - student.GPA <= 4.0

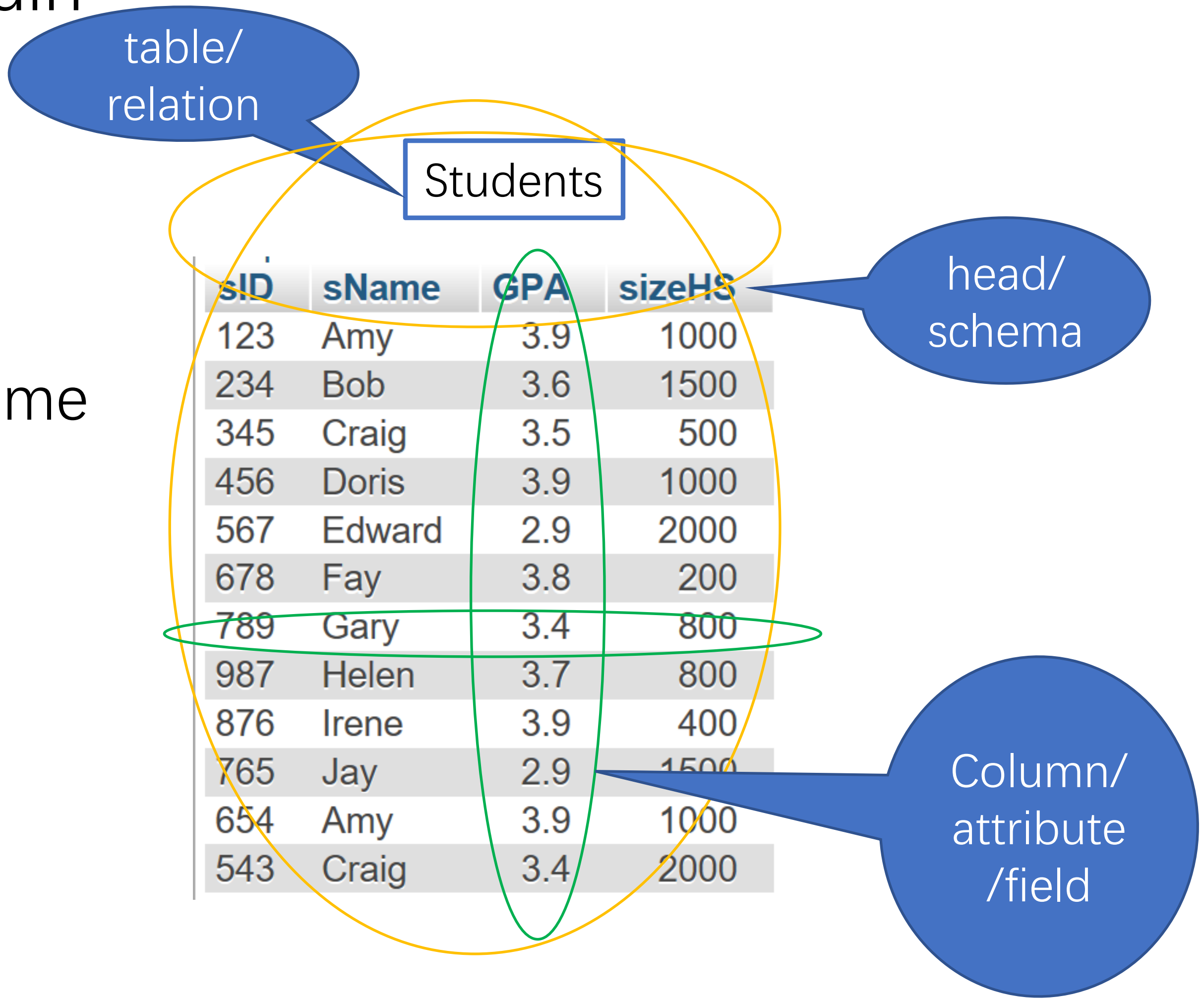
Entity

- Something needs to be represented in database
  - instructor, student, emplyee, project …
- An entity represents one theme, topic or business concept

- Entity is similar to a Class
- Entity Student, Amy Choo is an instance of student

Relation

- A relation is a table that has certain characteristics
  - row - instance of entity
  - column - attribut of entity
  - cell - hold a single value
  - all values in a column are of the same data type
  - each column has a unique name
  - no two rows can be identical



Students

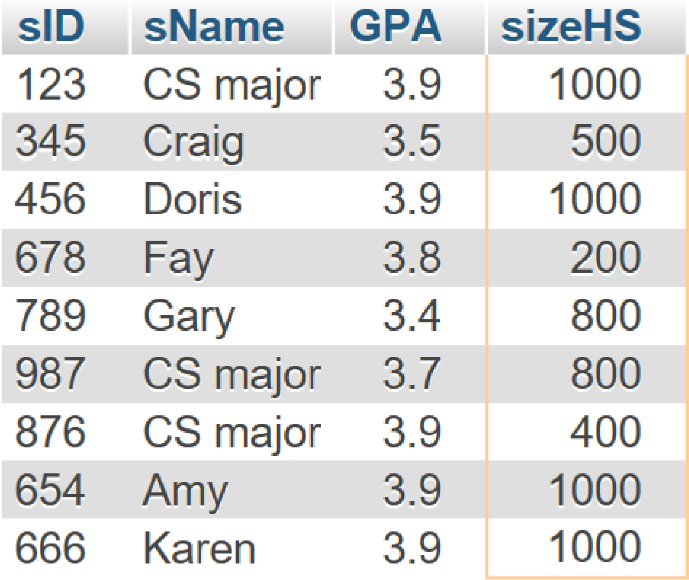
sID	sName	GPA	sizeHS
123	Amy	3.9	1000
234	Bob	3.6	1500
345	Craig	3.5	500
456	Doris	3.9	1000
567	Edward	2.9	2000
678	Fay	3.8	200
789	Gary	3.4	800
987	Helen	3.7	800
876	Irene	3.9	400
765	Jay	2.9	1500
654	Amy	3.9	1000
543	Craig	3.4	2000

Relational schema

- **Relational schema/schema** is the structure of **Relation**

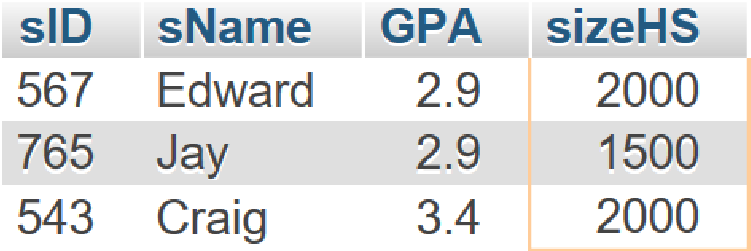
student(sID int, sName char(10), GPA double, sizeHs int)

Schema



sID	sName	GPA	sizeHS
123	CS major	3.9	1000
345	Craig	3.5	500
456	Doris	3.9	1000
678	Fay	3.8	200
789	Gary	3.4	800
987	CS major	3.7	800
876	CS major	3.9	400
654	Amy	3.9	1000
666	Karen	3.9	1000

Relation1



sID	sName	GPA	sizeHS
567	Edward	2.9	2000
765	Jay	2.9	1500
543	Craig	3.4	2000

Relation2

Relational schema - definition

- Relational schema: R(A1:D1, A2:D2, ···, An:Dn)
  - Ai: name of the i-th attribute
  - Di: Ai’s data domain (domain: a set of values with same type)
  - Value of Di must be in the domain Di

  - Given a schema R(A1:D1, A2:D2, A3:D3)
    - Attribute A1's domain is D1 = {high mid low}
    - Attribute A2's domain is D2 = {1 2 3}
    - Attribute A3's domain is D3 = {red blue}

Characteristics of relation

CS211 - Introduction to Database

Cartesian Product

- Basing on definitions of schema, we can infer

- All possible tuples in schema R(A1:D1, A2:D2, A3:D3) is the Cartesian Product of D1 D2 and D3, i.e.

$$D1 \times D2 \times D3 = \{(d1, d2, d3) \mid di \in Di, i = 1..3\}$$

  - The totoal number of all possilbe tuples in R is

$$|D1| \times |D2| \times |D3|$$

$|Di|$ is the cardinality of set Di

Meaningful of a relation

- Howevre, tuple in Cartesian Product is not always meaningful

Male	Female	Child
John	Jenny	Jeremy
John	Jenny	Amy
John	Alice	Jeremy
John	Alice	Amy
Alan	Jenny	Jeremy
Alan	Jenny	Amy
Alan	Alice	Jeremy
Alan	Alice	Amy

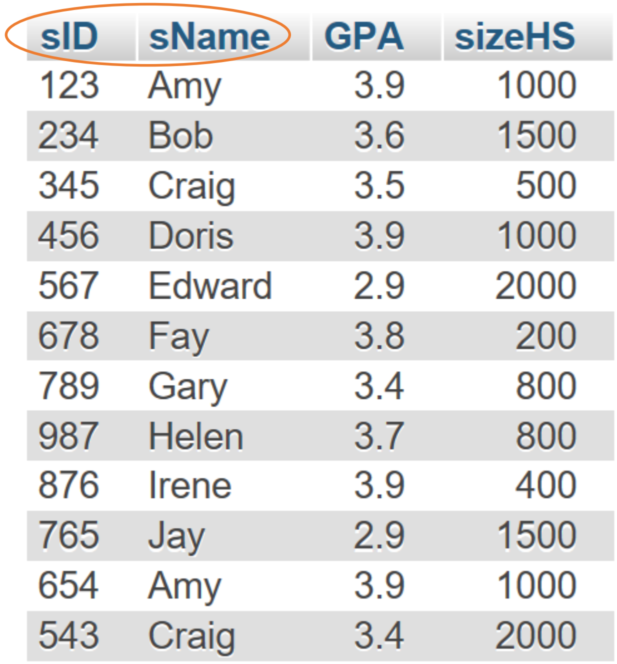
family

Husband	Wife	Kid
John	Jenny	Amy
Alan	Alice	Jeremy

Relation is a subset of Cartesian Product, each tuple is meaningful.

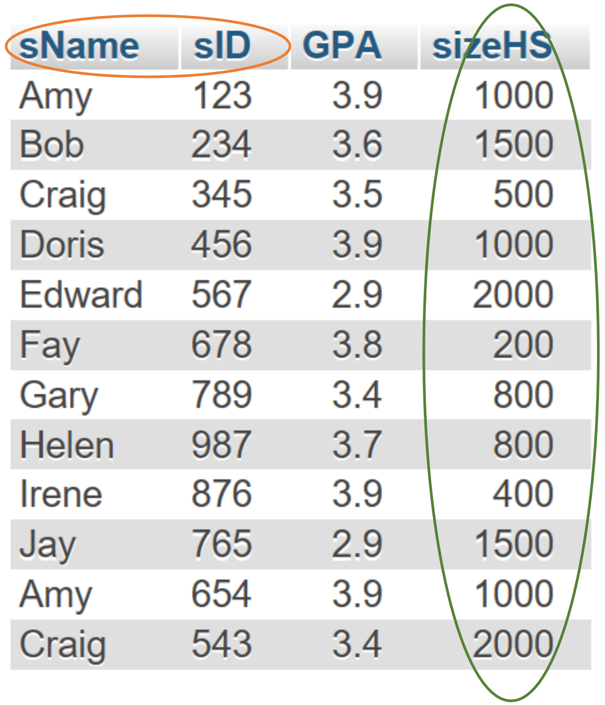
$$\{\text{John}, \text{Alan}\}\times\{\text{Jenny}, \text{Alice}\}\times\{\text{Jeremy}, \text{Amy}\}$$

Equivalence of a relation



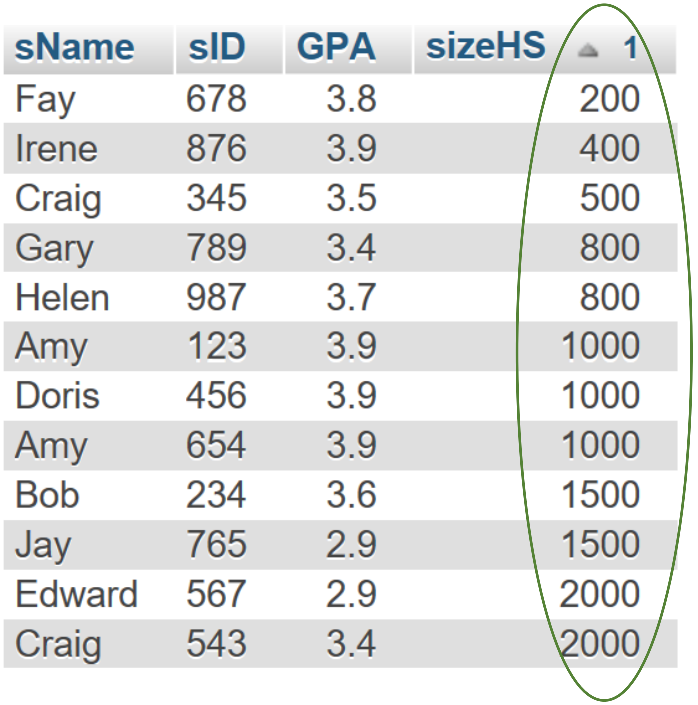
sID	sName	GPA	sizeHS
123	Amy	3.9	1000
234	Bob	3.6	1500
345	Craig	3.5	500
456	Doris	3.9	1000
567	Edward	2.9	2000
678	Fay	3.8	200
789	Gary	3.4	800
987	Helen	3.7	800
876	Irene	3.9	400
765	Jay	2.9	1500
654	Amy	3.9	1000
543	Craig	3.4	2000

=



sName	sID	GPA	sizeHS
Amy	123	3.9	1000
Bob	234	3.6	1500
Craig	345	3.5	500
Doris	456	3.9	1000
Edward	567	2.9	2000
Fay	678	3.8	200
Gary	789	3.4	800
Helen	987	3.7	800
Irene	876	3.9	400
Jay	765	2.9	1500
Amy	654	3.9	1000
Craig	543	3.4	2000

=

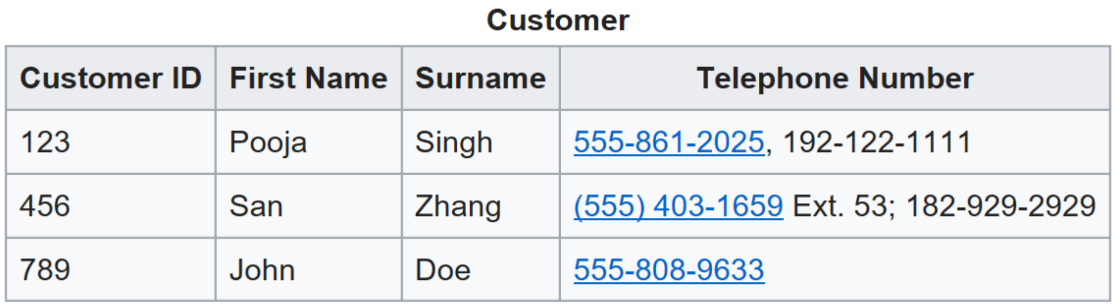


sName	sID	GPA	sizeHS ▲ 1
Fay	678	3.8	200
Irene	876	3.9	400
Craig	345	3.5	500
Gary	789	3.4	800
Helen	987	3.7	800
Amy	123	3.9	1000
Doris	456	3.9	1000
Amy	654	3.9	1000
Bob	234	3.6	1500
Jay	765	2.9	1500
Edward	567	2.9	2000
Craig	543	3.4	2000

the order of atrributes, the order of tuples is unimportant

1st NF of a relation

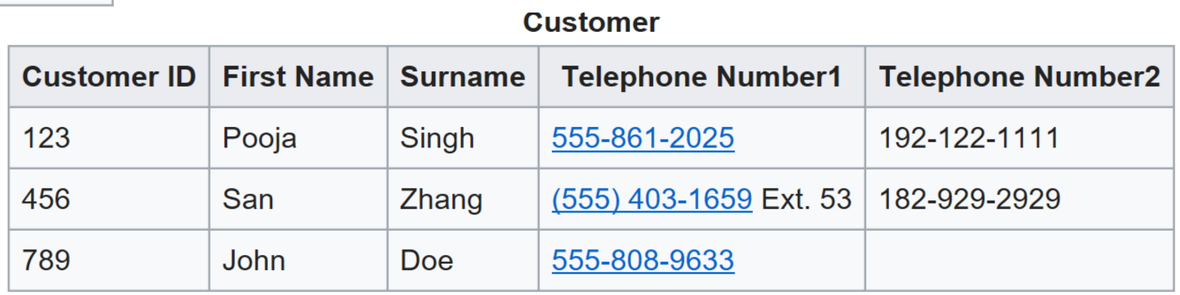
- A relation must be in first normal form (1st NF) when none of its domains have any sets as elements.



Customer

Customer ID	First Name	Surname	Telephone Number
123	Pooja	Singh	555-861-2025, 192-122-1111
456	San	Zhang	(555) 403-1659 Ext. 53; 182-929-2929
789	John	Doe	555-808-9633

Non-relation



Customer

Customer ID	First Name	Surname	Telephone Number1	Telephone Number2
123	Pooja	Singh	555-861-2025	192-122-1111
456	San	Zhang	(555) 403-1659 Ext. 53	182-929-2929
789	John	Doe	555-808-9633	

relation

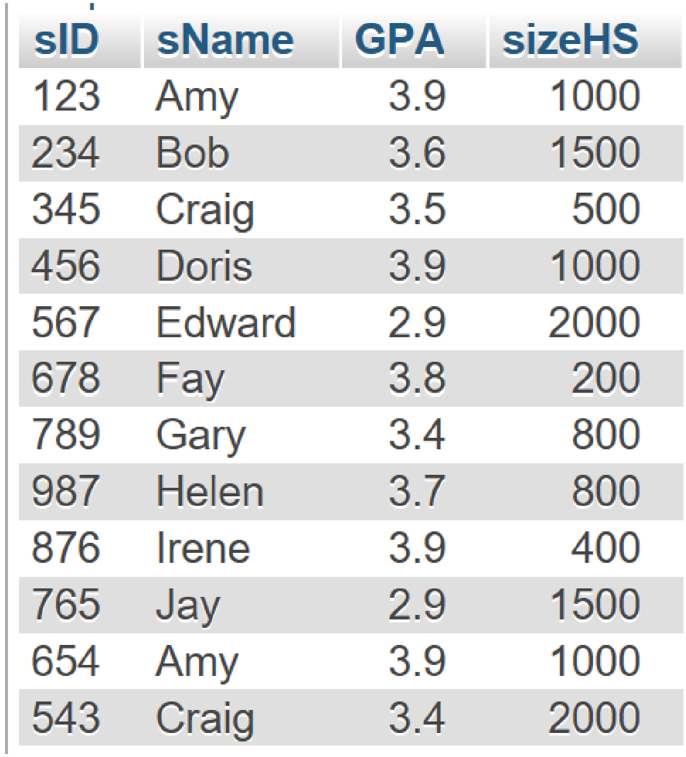
Keys, Foreign keys, and integrity constraints

CS211 - Introduction to Database

Key, Candidate key

- A **key/super key** is a set of one or more attributes that allow us to identify uniquely a tuple in the relation.
  - sID
  - sID, sName
  - sID, sName, GPA
  - …
- Minimal superkeys are called **candidate key**
  - sID

Student

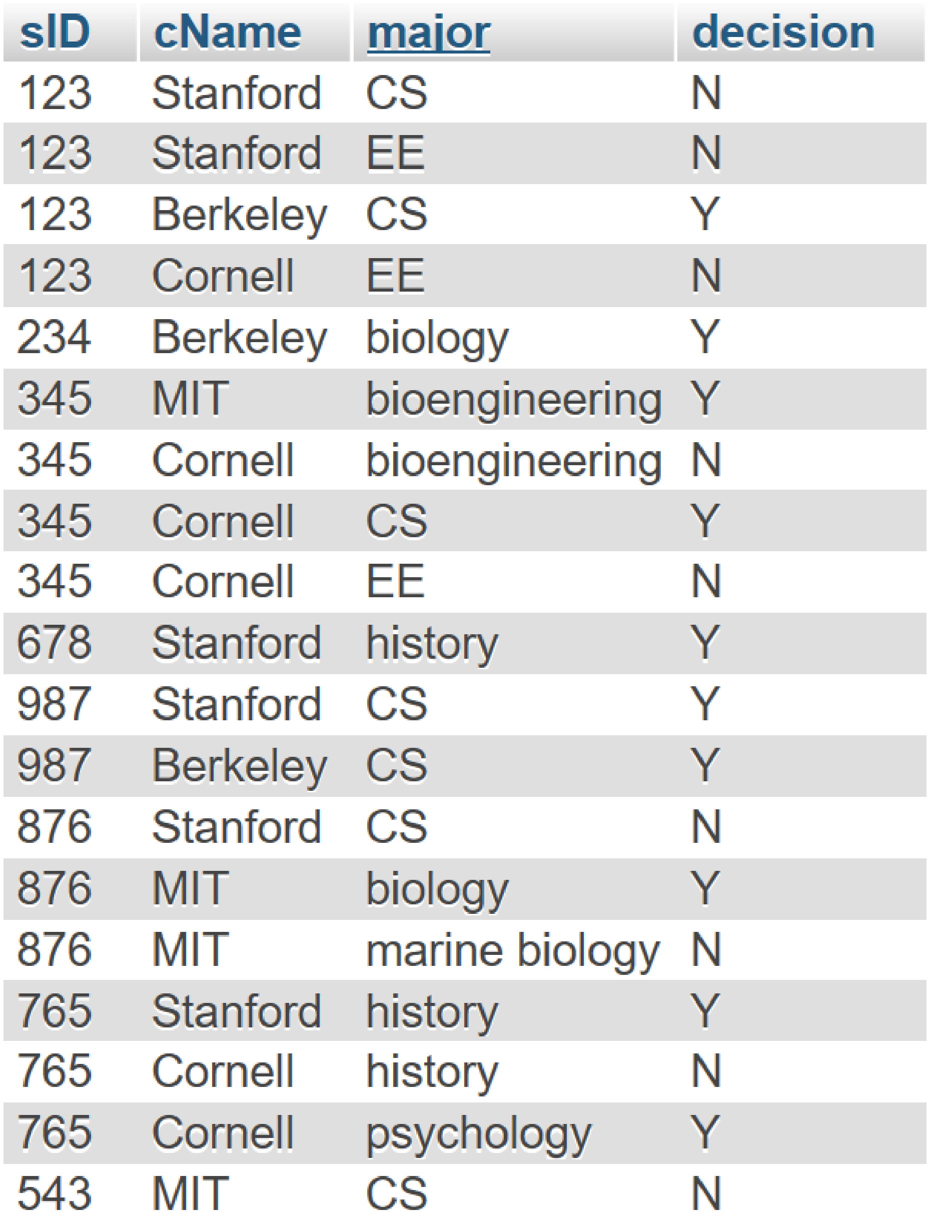


sID	sName	GPA	sizeHS
123	Amy	3.9	1000
234	Bob	3.6	1500
345	Craig	3.5	500
456	Doris	3.9	1000
567	Edward	2.9	2000
678	Fay	3.8	200
789	Gary	3.4	800
987	Helen	3.7	800
876	Irene	3.9	400
765	Jay	2.9	1500
654	Amy	3.9	1000
543	Craig	3.4	2000

Candidate Key

- **Apply** contain data of application
  - students apply universities
  - allow to apply multiple university
  - allow to apply multiple majors of a university
- (sID, cName, major)

Apply



sID	cName	major	decision
123	Stanford	CS	N
123	Stanford	EE	N
123	Berkeley	CS	Y
123	Cornell	EE	N
234	Berkeley	biology	Y
345	MIT	bioengineering	Y
345	Cornell	bioengineering	N
345	Cornell	CS	Y
345	Cornell	EE	N
678	Stanford	history	Y
987	Stanford	CS	Y
987	Berkeley	CS	Y
876	Stanford	CS	N
876	MIT	biology	Y
876	MIT	marine biology	N
765	Stanford	history	Y
765	Cornell	history	N
765	Cornell	psychology	Y
543	MIT	CS	N

Primary key & Candidate Key

- Employee(empID, empName, Mobile)

- Two candidate keys
  - empID
  - Mobile

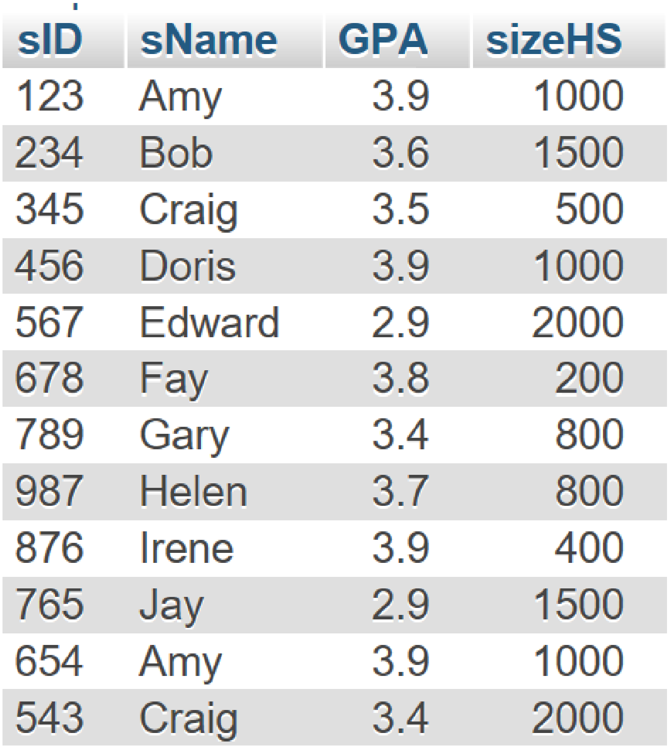
- **primary key** denotes a candidate key that is chosen by the database designer as the principal means of identifying tuples within a relation.
- DBMS use the primary key manage tuples within a relation.

Foreign key

sID is a foreign key from apply referencing Student.sID

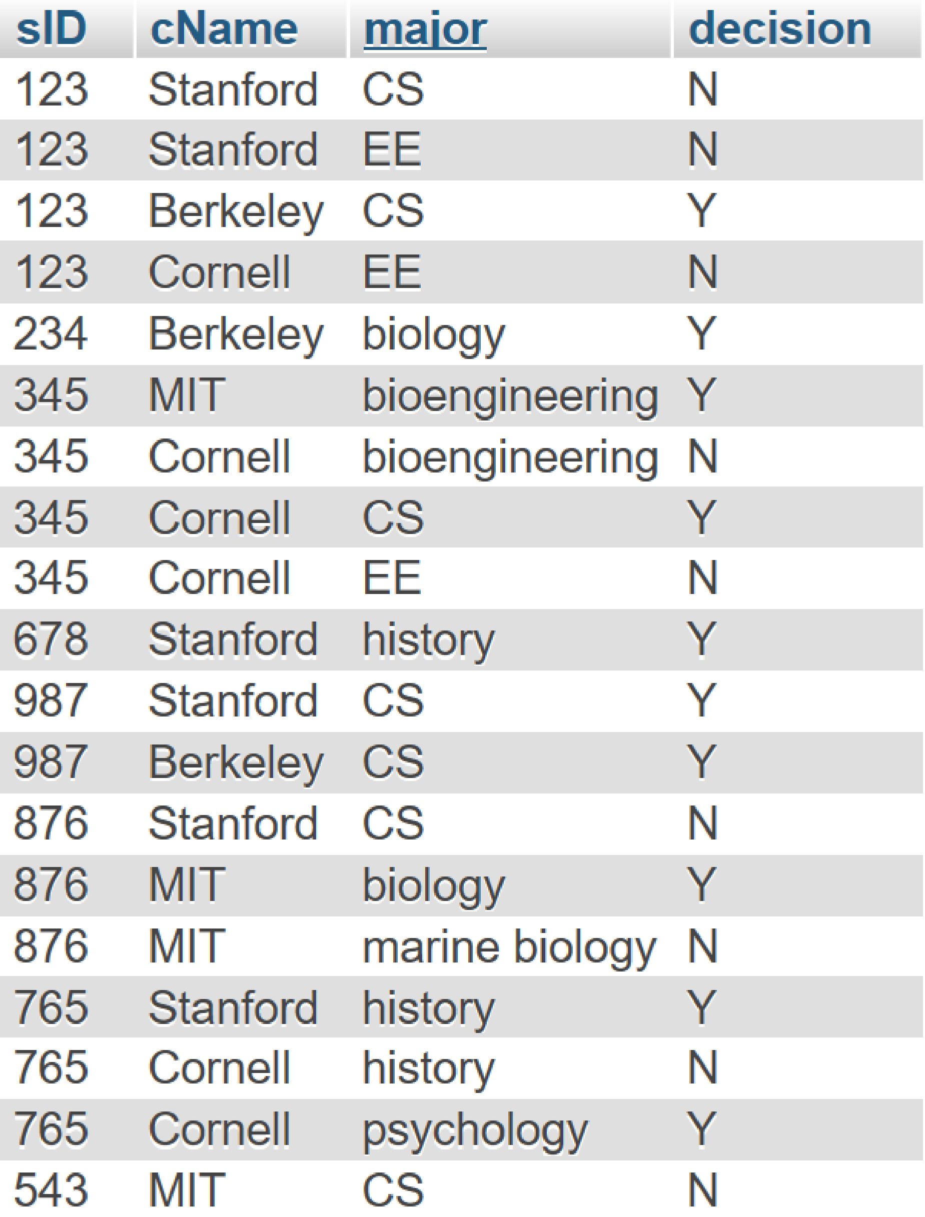
A relation, say *r*1, may include among its attributes the primary key of another relation, say *r*2. This attribute is called a **foreign key** from *r*1, referencing *r*2.

Student



sID	sName	GPA	sizeHS
123	Amy	3.9	1000
234	Bob	3.6	1500
345	Craig	3.5	500
456	Doris	3.9	1000
567	Edward	2.9	2000
678	Fay	3.8	200
789	Gary	3.4	800
987	Helen	3.7	800
876	Irene	3.9	400
765	Jay	2.9	1500
654	Amy	3.9	1000
543	Craig	3.4	2000

Apply



sID	cName	major	decision
123	Stanford	CS	N
123	Stanford	EE	N
123	Berkeley	CS	Y
123	Cornell	EE	N
234	Berkeley	biology	Y
345	MIT	bioengineering	Y
345	Cornell	bioengineering	N
345	Cornell	CS	Y
345	Cornell	EE	N
678	Stanford	history	Y
987	Stanford	CS	Y
987	Berkeley	CS	Y
876	Stanford	CS	N
876	MIT	biology	Y
876	MIT	marine biology	N
765	Stanford	history	Y
765	Cornell	history	N
765	Cornell	psychology	Y
543	MIT	CS	N

Data Integrity

- Data integrity refers to the correctness and consistency of data in stored in a database
- Types of data integrity
  - Entity integrity
  - Referential integrity
  - User defined integrity

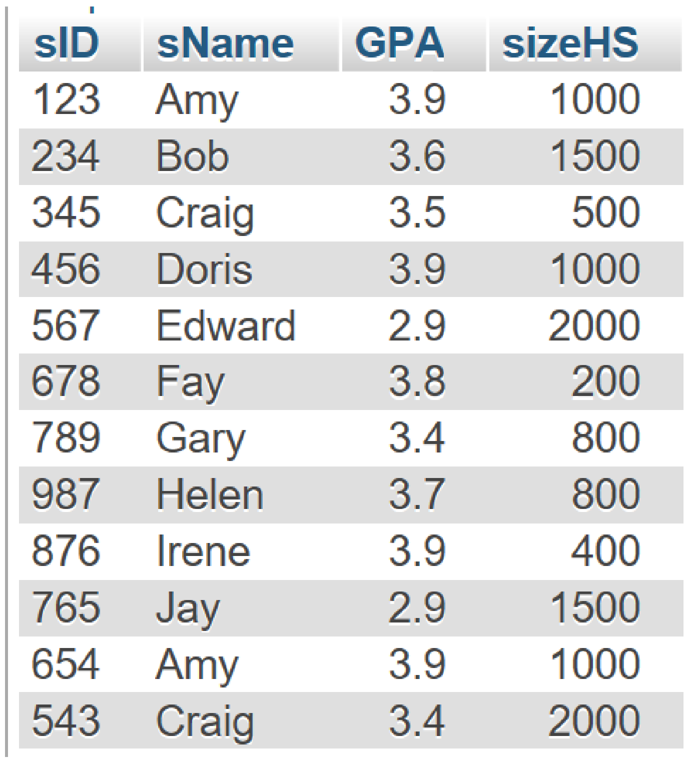
Entity Integrity

- Value of Primary key can not be NULL
- You cannot have two tuples with the same Primary Key in the same table
  - violation

(NULL, John, 3.0, 1000)

(123, John, 3.0, 1000)

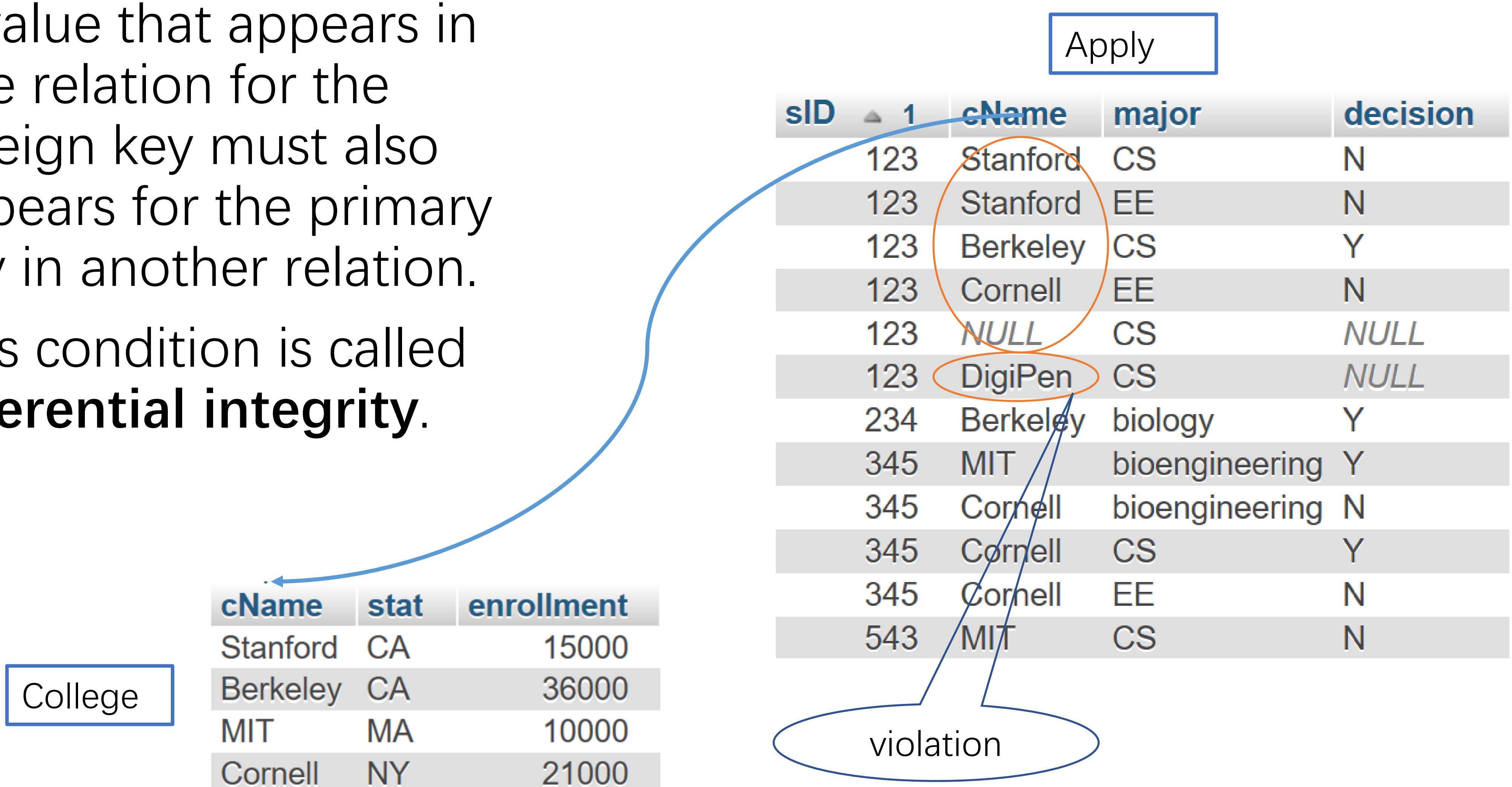
Student



sID	sName	GPA	sizeHS
123	Amy	3.9	1000
234	Bob	3.6	1500
345	Craig	3.5	500
456	Doris	3.9	1000
567	Edward	2.9	2000
678	Fay	3.8	200
789	Gary	3.4	800
987	Helen	3.7	800
876	Irene	3.9	400
765	Jay	2.9	1500
654	Amy	3.9	1000
543	Craig	3.4	2000

Referential Integrity

- A value that appears in one relation for the foreign key must also appears for the primary key in another relation.
- This condition is called **referential integrity**.



Apply

sID ▲ 1	cName	major	decision
123	Stanford	CS	N
123	Stanford	EE	N
123	Berkeley	CS	Y
123	Cornell	EE	N
123	*NULL*	CS	*NULL*
123	DigiPen	CS	*NULL*
234	Berkeley	biology	Y
345	MIT	bioengineering	Y
345	Cornell	bioengineering	N
345	Cornell	CS	Y
345	Cornell	EE	N
543	MIT	CS	N

College

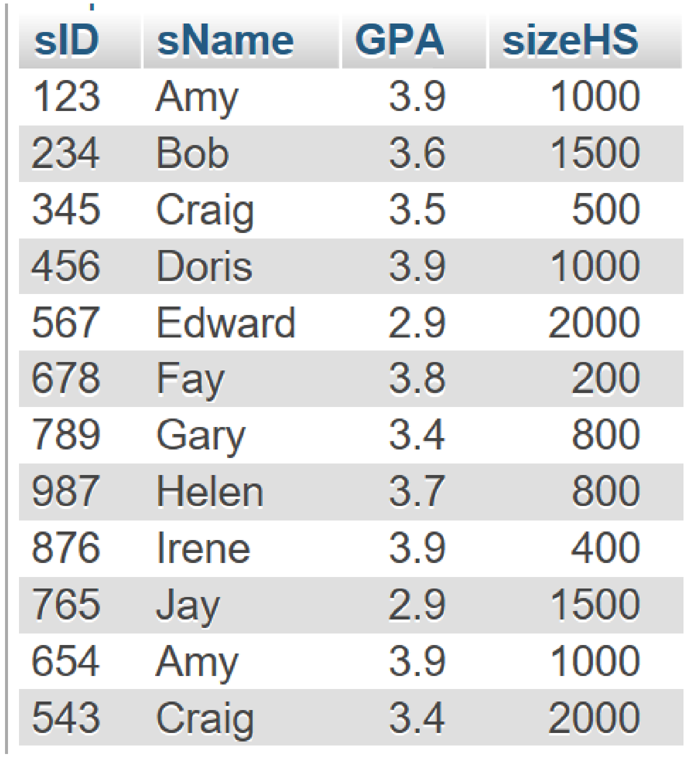
cName	stat	enrollment
Stanford	CA	15000
Berkeley	CA	36000
MIT	MA	10000
Cornell	NY	21000

User defined Integrity

- Range of GPA must in [0..4]
  - violation

(123, John, 4.5, 1000)

Student



sID	sName	GPA	sizeHS
123	Amy	3.9	1000
234	Bob	3.6	1500
345	Craig	3.5	500
456	Doris	3.9	1000
567	Edward	2.9	2000
678	Fay	3.8	200
789	Gary	3.4	800
987	Helen	3.7	800
876	Irene	3.9	400
765	Jay	2.9	1500
654	Amy	3.9	1000
543	Craig	3.4	2000