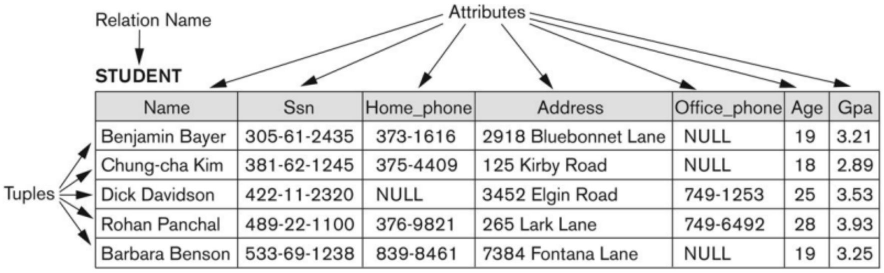
Relational Data Model

Notes by Mannan Ul Haq (BDS-4A)

Relational model represents data as a collection of tables. A table is also called a relation.

A relation typically contains a set of rows or tuples. The data elements in each row represent certain facts that correspond to a real-world entity or relationship.

Each column has a column header that gives an indication of the meaning of the data items in that column. The column headers are also called attributes.


Relation Name
Attributes
Tuples

STUDENT

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25

Domain:

- A set of atomic values allowed for an attribute.
- **Example: 1)** Name: string of characters that represent name of persons. **2)** Employee_ages: Possible ages of employees of a company (values between 20 & 70 years old).

Relational Schema:

The Schema (or description) of a Relation:

- Denoted by R(A1, A2, ...An)
- R is the name of the relation
- The attributes of the relation are A1, A2, ..., An

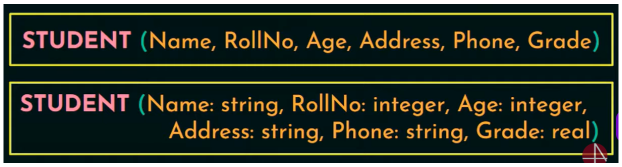
Example:

- CUSTOMER (Cust-id, Cust-name, Address, Phone#)
- CUSTOMER is the relation name
- Defined over the four attributes: Cust-id, Cust-name, Address, Phone#

Each attribute has a domain or a set of valid values. For example, the domain of Cust-id is 6 digit numbers.

Degree of a Relation:

Number of attributes in a relation.

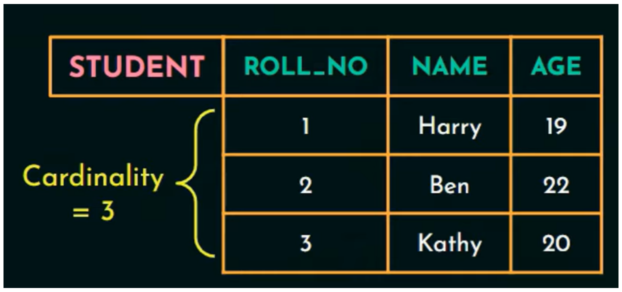


STUDENT (Name, RollNo, Age, Address, Phone, Grade)

STUDENT (Name: string, RollNo: integer, Age: integer, Address: string, Phone: string, Grade: real)

Cardinality:

Total number of tuples present in a relation.



STUDENT	ROLL_NO	NAME	AGE
Cardinality = 3	1	Harry	19
	2	Ben	22
	3	Kathy	20

Relational Database schema:

Is a set of relational schema and a set of integrity constraints.

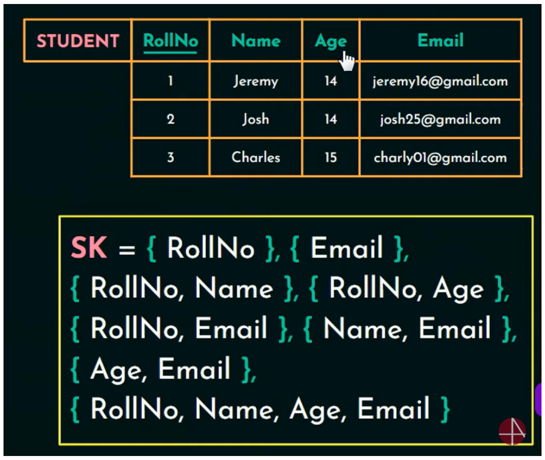
Relation State (or Relation Instance):

Set of tuples at a given time.

Key of a Relation:

It refers to a column or set of columns in a table that uniquely identify each row or record in that table:

1. **Super Key:**
    - A super key is a set of one or more attributes (columns) that, taken together, uniquely identify each record in a table.
    - Every relation has at least one super key → set of all attributes.
    - Example:



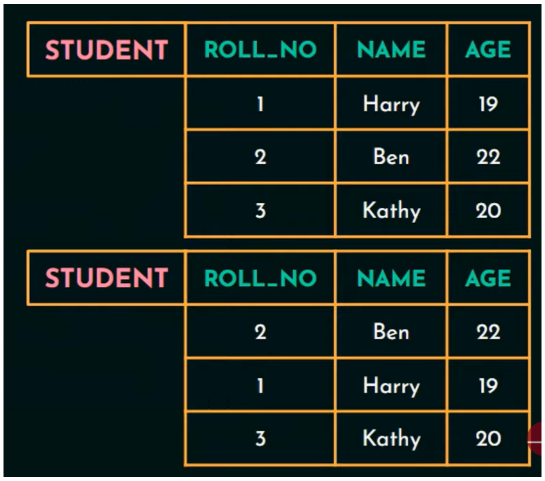
STUDENT	RollNo	Name	Age	Email
	1	Jeremy	14	jeremy16@gmail.com
	2	Josh	14	josh25@gmail.com
	3	Charles	15	charly01@gmail.com

SK = { RollNo }, { Email },
{ RollNo, Name }, { RollNo, Age },
{ RollNo, Email }, { Name, Email },
{ Age, Email },
{ RollNo, Name, Age, Email }

2. **Minimal Key:**
    - It is the smallest set of attributes required to ensure uniqueness.
    - Example: In a table of Student, if {RollNo, Email} is a super key, but removing Email still ensures uniqueness, {RollNo} is the minimal key.
3. **Candidate Key:**
    - A candidate key is a minimal super key, meaning it is a set of attributes that uniquely identify each record, and removing any attribute from the set would cause it to lose its uniqueness.
    - Each table can have multiple candidate keys.
    - Example: In a table of Student, both RollNo and Email could be candidate keys because they uniquely identify each student.
4. **Primary Key:**
    - The primary key is a candidate key chosen by the database designer to uniquely identify records in a table.
    - It is the key used for establishing relationships between tables and enforcing data integrity constraints.
    - Primary keys must be unique and non-null for each record.
    - Example: In a table of Student, the RollNo might be chosen as the primary key.
5. **Composite Key:**
    - A composite key is a primary key composed of multiple attributes.
    - It is used when a single attribute does not provide enough uniqueness.

Characteristics of Relations

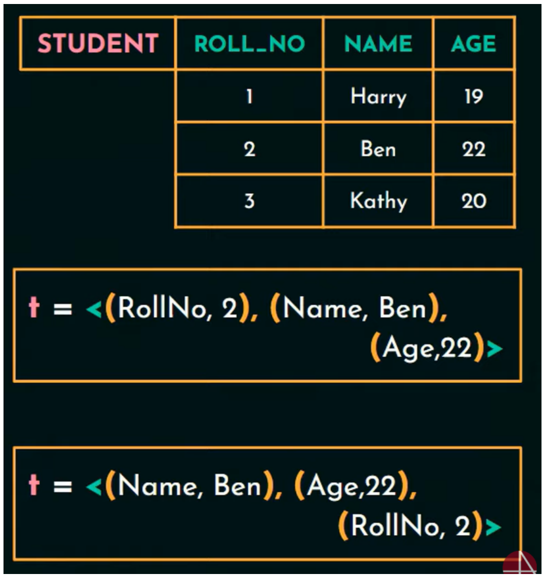
- Duplicate rows are not allowed.
- Tuples in a relation need not have any particular order.



STUDENT	ROLL_NO	NAME	AGE
	1	Harry	19
	2	Ben	22
	3	Kathy	20

STUDENT	ROLL_NO	NAME	AGE
	2	Ben	22
	1	Harry	19
	3	Kathy	20

- Ordering of values within a tuple is important.

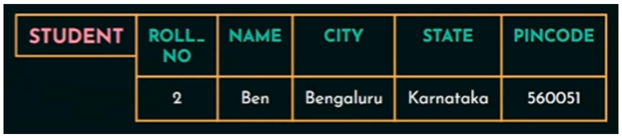


STUDENT	ROLL_NO	NAME	AGE
	1	Harry	19
	2	Ben	22
	3	Kathy	20

t = <(RollNo, 2), (Name, Ben), (Age,22)>

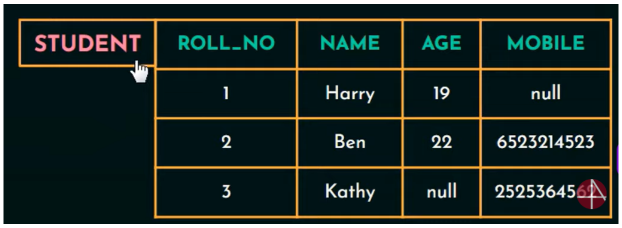
t = <(Name, Ben), (Age,22), (RollNo, 2)>

- Only atomic values are allowed.



STUDENT	ROLL_NO	NAME	CITY	STATE	PINCODE
	2	Ben	Bengaluru	Karnataka	560051

- Nulls → unknown or not applicable.



STUDENT	ROLL_NO	NAME	AGE	MOBILE
	1	Harry	19	null
	2	Ben	22	6523214523
	3	Kathy	null	2525364562

Relational Model Constraints

Constraints determine which values are permissible and which are not in the database.

They are of three main types:

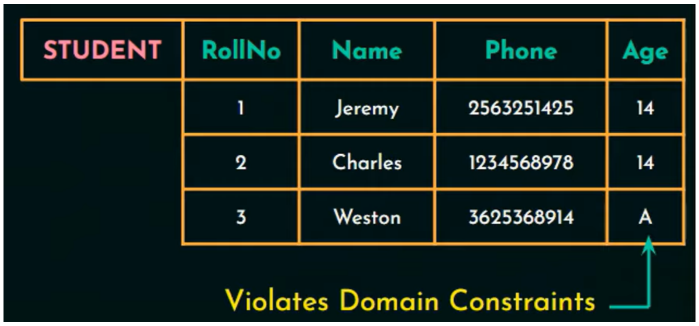
1. Inherent or Implicit Constraints: These are based on the data model itself. (E.g., relational model does not allow a list as a value for any attribute)

2. Schema-based or Explicit Constraints: They are expressed in the schema by using the facilities provided by the model.

3. Application based or semantic constraints: These are beyond the expressive power of the model and must be specified and enforced by the application programs.

Schema-based Constraints:

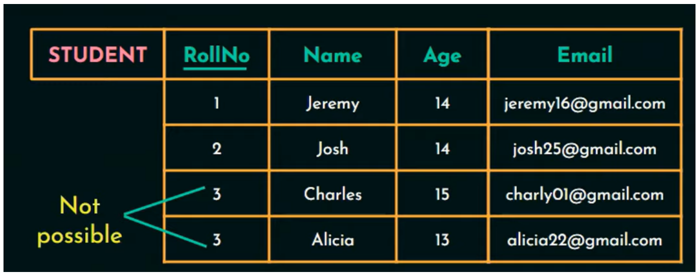
1. **Domain Constraints:** Must be an atomic value. Performs data type check.



STUDENT	RollNo	Name	Phone	Age
	1	Jeremy	2563251425	14
	2	Charles	1234568978	14
	3	Weston	3625368914	A

Violates Domain Constraints

2. **Key Constraints:** Two tuples cannot have identical values for all the attributes in the key. And it is a minimal super key.

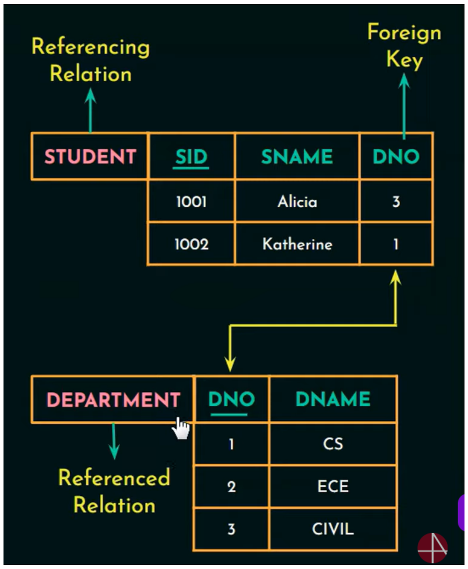


STUDENT	RollNo	Name	Age	Email
	1	Jeremy	14	jeremy16@gmail.com
	2	Josh	14	josh25@gmail.com
	3	Charles	15	charly01@gmail.com
	3	Alicia	13	alicia22@gmail.com

Not possible

3. **Entity Integrity Constraint:** States that no primary key value can be null.

4. **Referential Integrity Constraint:** It is between two relations. States that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation. Tuples in the referencing relation R1 have attributes FK (called foreign key attributes) that reference the primary key attributes PK of the referenced relation R2.



Referencing Relation

Foreign Key

STUDENT	SID	SNAME	DNO
	1001	Alicia	3
	1002	Katherine	1

DEPARTMENT	DNO	DNAME
	1	CS
	2	ECE
	3	CIVIL

Referenced Relation

Update Operations on Relations

Inserting a Tuple:

- **Description:** Adding a new record to a relation.
- **Possible Violations:**
  - Domain Constraint: Inserted attribute values must adhere to specified domains.
  - Key Constraint: The value of key attributes in the new tuple must not already exist in the relation.
  - Referential Integrity: Foreign key values in the new tuple must reference existing primary key values.
  - Entity Integrity: Primary key value cannot be null in the new tuple.

Deleting a Tuple:

- **Description:** Removing an existing record from a relation.
- **Possible Violations:**
  - Referential Integrity: Deleting a tuple may violate referential integrity if its primary key is referenced from other tuples.

Modifying a Tuple:

- **Description:** Updating attribute values of an existing tuple.
- **Possible Violations:**
  - Domain Constraint: Modified attribute values must remain within specified domains.
  - NOT NULL Constraint: Attributes being modified cannot be set to null.
  - Other Constraints: Depending on the attribute being updated, key constraints or referential integrity may also be violated.
  - Updating Primary Key (PK): Similar to a delete followed by an insert, necessitating similar options.
  - Updating Foreign Key (FK): May violate referential integrity.
  - Updating Ordinary Attribute: Can only violate domain constraints.

Handling Violations:

- Cancel the operation (RESTRICT or REJECT).
- Inform the user of the violation.
- Trigger additional updates to correct the violation (CASCADE or SET NULL):
  - CASCADE option: propagate the new primary key value into the foreign keys of the referencing tuples
  - SET NULL option: set the foreign keys of the referencing tuples to NULL
- Execute a user-specified error-correction routine.