UCS310
Database Management System

Introduction to Relational Data Model

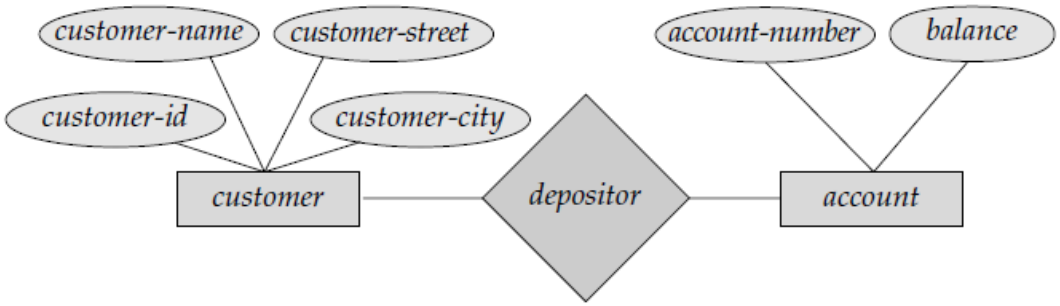
Lecture-04

Date: 20 Jan 2023

Dr. Sumit Sharma
Assistant Professor
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala

Recap

- DBMS Components
- Advantages and Disadvantages of DBMS
- Components of the database engine
  - Storage manager
  - Query processor
  - Transaction management
- Different DBMS architectures
- Introduction about two data models
  - Entity-Relationship (ER) model
  - Relational model



customer-id	customer-name	customer-street	customer-city
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

account-number	balance
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

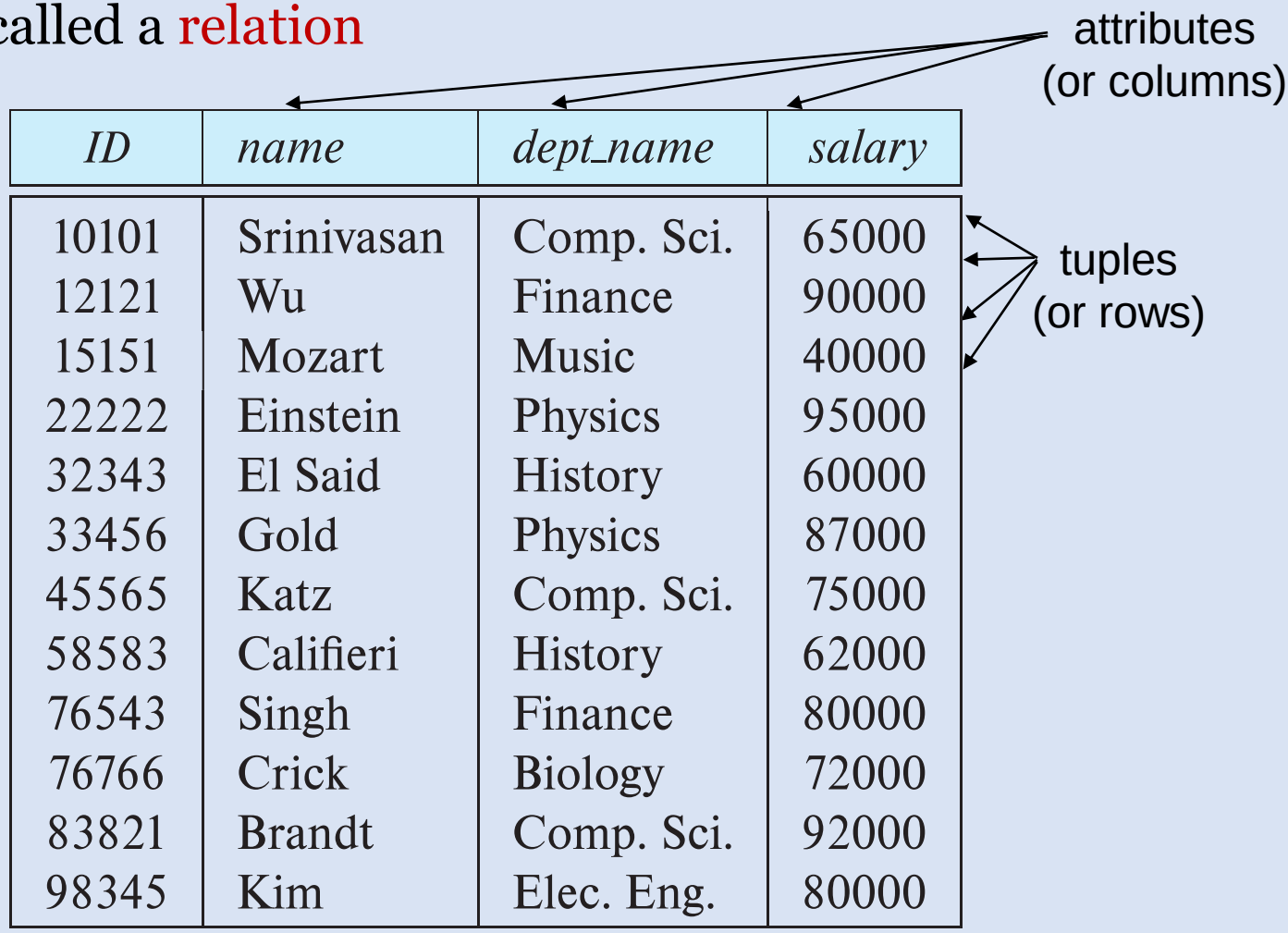
(b) The *account* table

customer-id	account-number
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

Example of a Instructor Relation

- Relational Model represents data as a collection of tables
- A table is also called a relation



ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Relation Schema and Instance

- $A_1, A_2, \ldots, A_n$ are attributes
- $R\ (A_1, A_2, \ldots, A_n\ )$ is a relation schema

  Example:

  instructor (ID, name, dept_name, salary)

  instructor(ID: integer, name: string,
  dept_name: string, salary: float)

- A relation instance ***r*** defined over schema *R* is denoted by *r (R)*.
- The current values of a relation are specified by a table
- An element ***t*** of relation ***r*** is called a tuple and is represented by a row in a table

Attributes

- The set of allowed values for each attribute is called the **domain** of the attribute
- Attribute values are (normally) required to be **atomic**; that is, indivisible

  Example:

  Employees ages: Possible ages of employees of a company (values between 20 & 70 years old)

  Instructor phone number

- The special value **null** is a member of every domain. Indicated that the value is "unknown"
- The null value causes complications in the definition of many operations

Degree of a Relation

- Number of attributes in a relation schema
- Example:

  instructor = (ID, name, dept_name, salary)

Cardinality

- Number of tuples present in a relation schema
- Example:

  instructor = (ID, name, dept_name, salary)

instructor	ID	NAME	DEPT_NAME	SALARY
	1001	Preeti	CSE	80000
	1002	Ishani	ECE	70000
	1003	Sonal	CSE	100000

Cardinality =3

Characteristics of Relation

- **Order of tuples** is irrelevant (tuples may be stored in arbitrary order)
- Example: instructor relation with unordered tuples

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

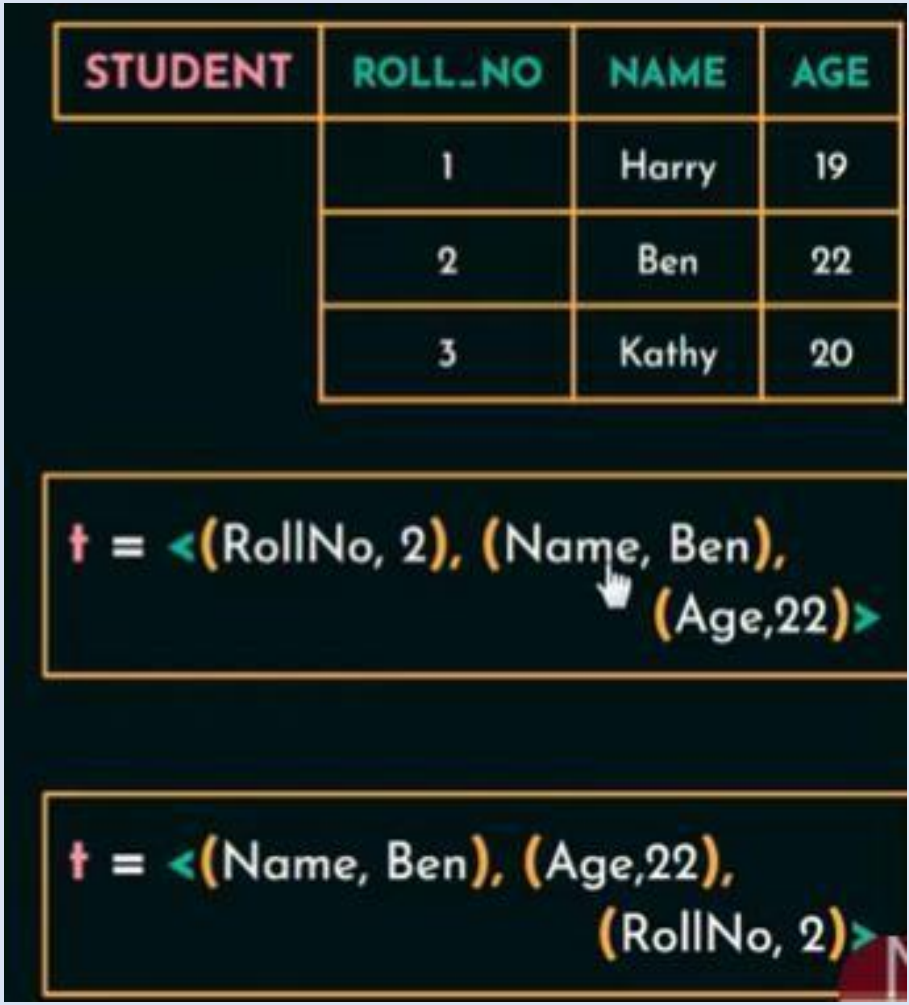
Characteristics of Relation

- **Ordering of values within a tuple**:
- An n-tuple -> ordered list of n values, so the ordering of values in a tuple is important.
- With an alternative definition of relation, ordering of values in a tuple is unnecessary

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

Characteristics of Relation

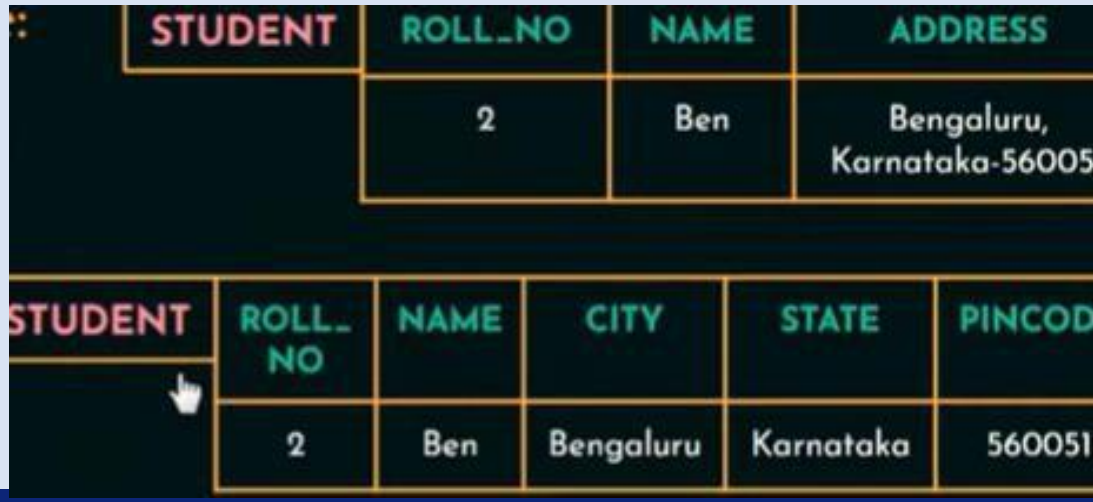
- **Ordering of values within a tuple**:
- A tuple is a set of (<attribute>, <value>) pair, then ordering of attributes is not important



Characteristics of Relation

Values & Nulls in a Tuple:

- Each value in a tuple is an atomic value, i.e., it does not have composite values
- Nulls: unknown or not applicable



Characteristics of Relation

Interpretation of a relation:

- The relation schema can be represented as a declaration or assertion
- Each tuple can be interpreted as a fact

Database Schema

- Database schema -- is the logical structure of the database.
- Database instance -- is a snapshot of the data in the database at a given instant in time.
- Example:
  - schema: instructor (ID, name, dept_name, salary)
  - Instance:

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Keys of Relation

- Keys in DBMS is an attribute or set of attributes that help to identify a row (tuple) in a relation (table)
- Allow to find the relation between two tables
- Help to enforce identity and integrity in the relationship

- Primary Key
- Candidate Key
- Alternate Key
- Composite Key
- Foreign Key
- Super Key

Keys

- Let, $K \subseteq R$
- K is a **superkey** of R if values for K are *sufficient to identify a unique tuple* of each possible relation $r(R)$
  - Example: {ID} and {ID,name} are both superkeys of instructor.
- Superkey K is a **candidate key** if K is minimal
  - Example: {ID} is a candidate key for the Instructor
- One of the candidate keys is selected to be the **primary key**.
  - Which one?
- **Foreign key** constraint: Value in one relation must appear in another
  - **Referencing** relation
  - **Referenced** relation
  - Example: dept_name in instructor is a foreign key from the instructor referencing department

Keys

- **Super Key** – a group of single or multiple keys which identifies rows in a table
- **Primary Key** – is a column or group of columns in a table that uniquely identify every row in that table
- **Candidate Key** – a set of attributes that uniquely identify tuples in a table. Candidate Key is a super key with no repeated attributes
- **Foreign Key** – a column that creates a relationship between two tables. The purpose of foreign keys is to maintain data integrity and allow navigation between two different instances of an entity

Relational Model Constraints

- **Constraints on databases:**
  - Inherent Model - Based: Inherent in the model (already existing)
  - Example: duplicate records are not allowed
- **Schema based:**
  - Defined directly in the schemas of the data model
  - Example: Age of an employee should be between 25-65
- **Application based:**
  - Must be expressed and enforced by the application programs

Constraints

- The constraints can be placed at the column level or table level.

- **Column level constraint**:
  - These constraints are defined along with the column definition
  - Can be applied to any one column at a time
  - If the constraints span across multiple columns, then the table-level constraints are used.

- **Table level constraint:**
  - If the data constraint attached to a specific cell in a table references the content of another cell in the table then the table-level constraint is used.

Relational Model Constraints: Schema Based

- Domain Constraints
- Key Constraints
- Constraints on NULL
- Entity Integrity Constraint
- Referential Integrity Constraint

Schema Based Constraints

Domain Constraints

- It specifies that within each tuple or within each row the value of an attribute has to be atomic or individual
- Performs the datatype check of each attribute

STUDENT	ROLL_No	NAME	AGE
	1	Preeti	34
	2	Ishani	14
	3	Sonal	A

Violates Domain Constraint

Schema Based Constraints

Key Constraints

- An attribute that can uniquely identify each tuple in a relation is called a Key

STUDENT	ROLL_No	NAME	AGE
	1	Preeti	24
	2	Ishani	18
	3	Sonal	20

Key Constraints

There are a number of key constraints in SQL that ensure that an entity or record is uniquely or differently identified in the database

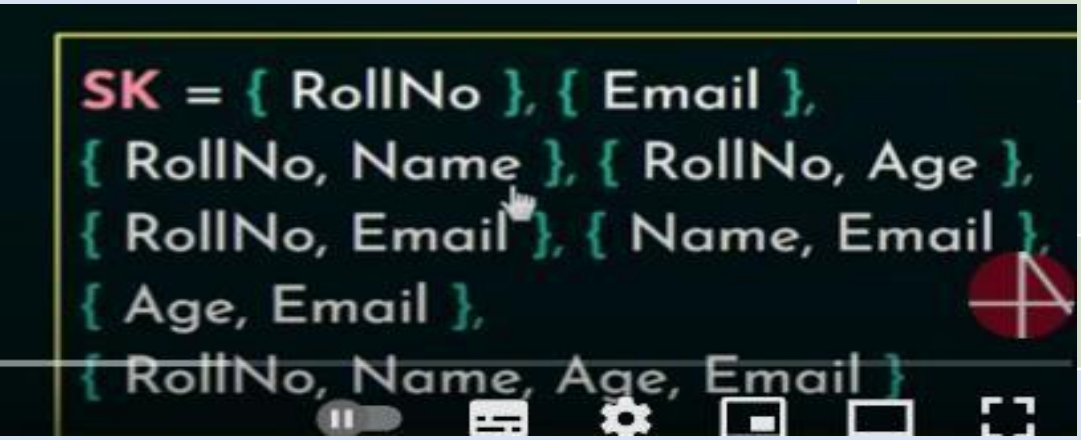
There can be more than one key in the table but it can have only one primary key

- Some of the key constraints in SQL are :
  - Primary key constraint
  - Foreign key constraint

Schema Based Constraints

Super Key Constraints

- A super key specifies that no two tuples can have the same value
- Every relation has at least one superkey – a set of all attributes



STUDENT	ROLLNo	NAME	AGE	Email
	1	Preeti	24	preeti@gmail.com
	2	Ishani	18	ishani@gmail.com
	3	Sonal	20	sonal_cs.thapar.edu

Schema Based Constraints

Key Constraints

- Two tuples cannot have identical values for all the attributes in the key
- It is a minimal superkey

Schema Based Constraints

Candidate Keys:

- Set of attributes that uniquely identify the tuples in a relation

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

Schema Based Constraints

Constraints on NULL values:

- Specifies whether null values are permitted or not (NOT NULL)

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

Schema Based Constraints

Entity Integrity Constraints

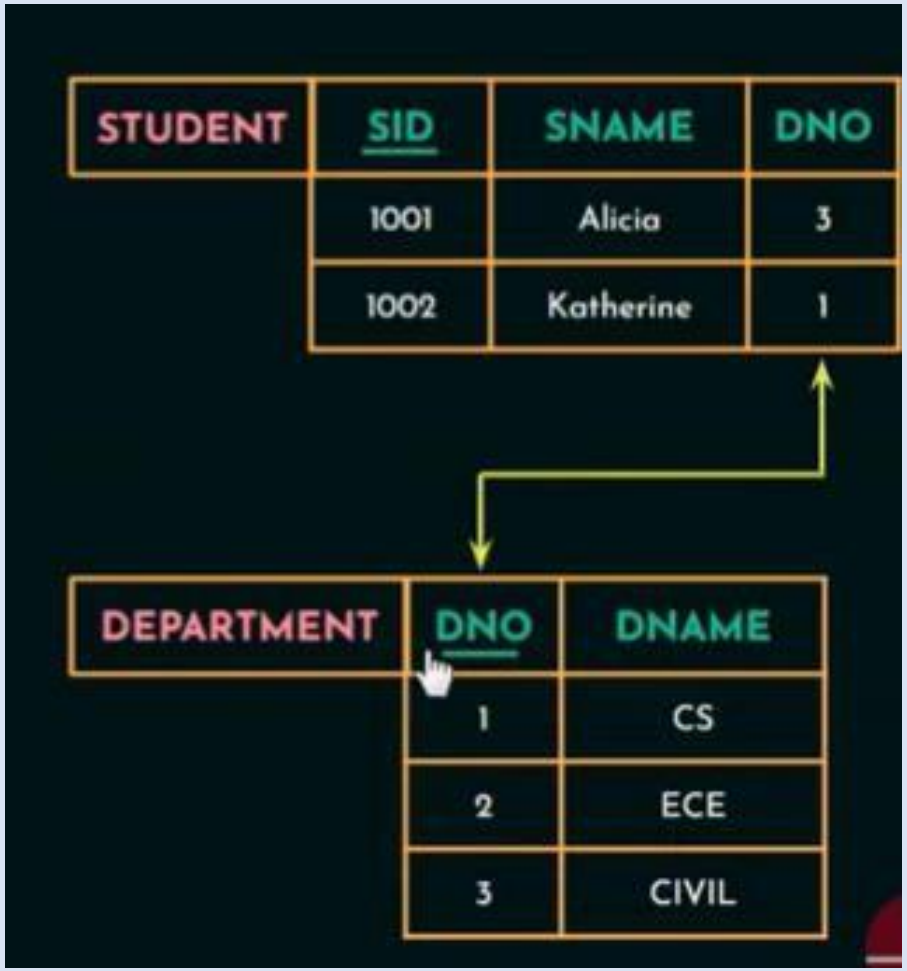
- States that no primary key value can be null

STUDENT	RollNo	Name	Age	Grade
	1	Jeremy	14	A
	2	Charles	14	A
	null	Charles	13	B

Schema Based Constraints

Referential Integrity Constraints

- Specified between two relations
- States that a tuple in one relation that refers to another relation must refer to an existing tuple in that relation

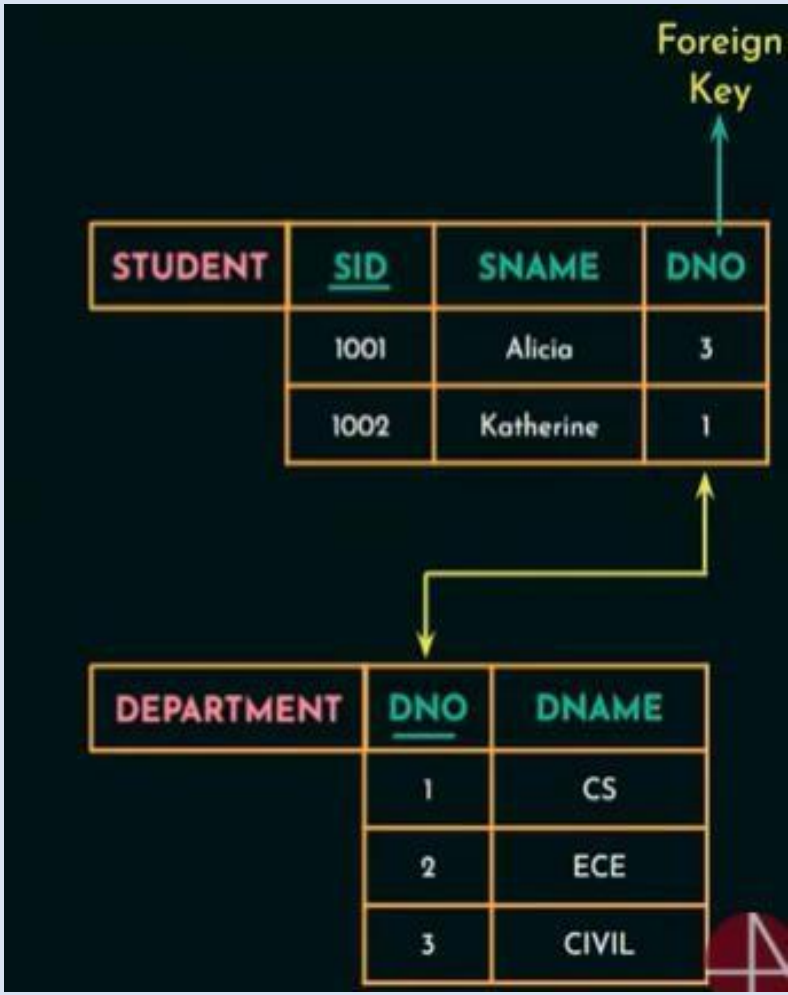


Schema Based Constraints

Foreign Key (FK) must satisfy

- Same domain

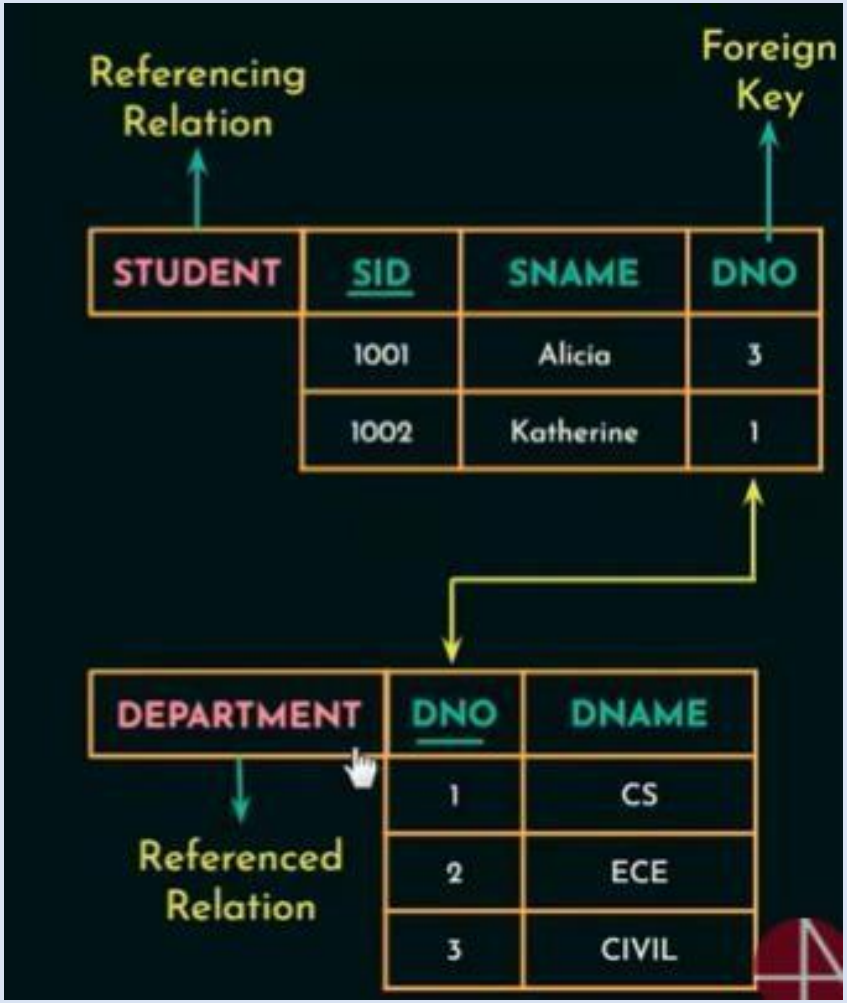
- Value of FK in a tuple either occurs as a value of PK, i.e., t1[FK]=t2[PK]



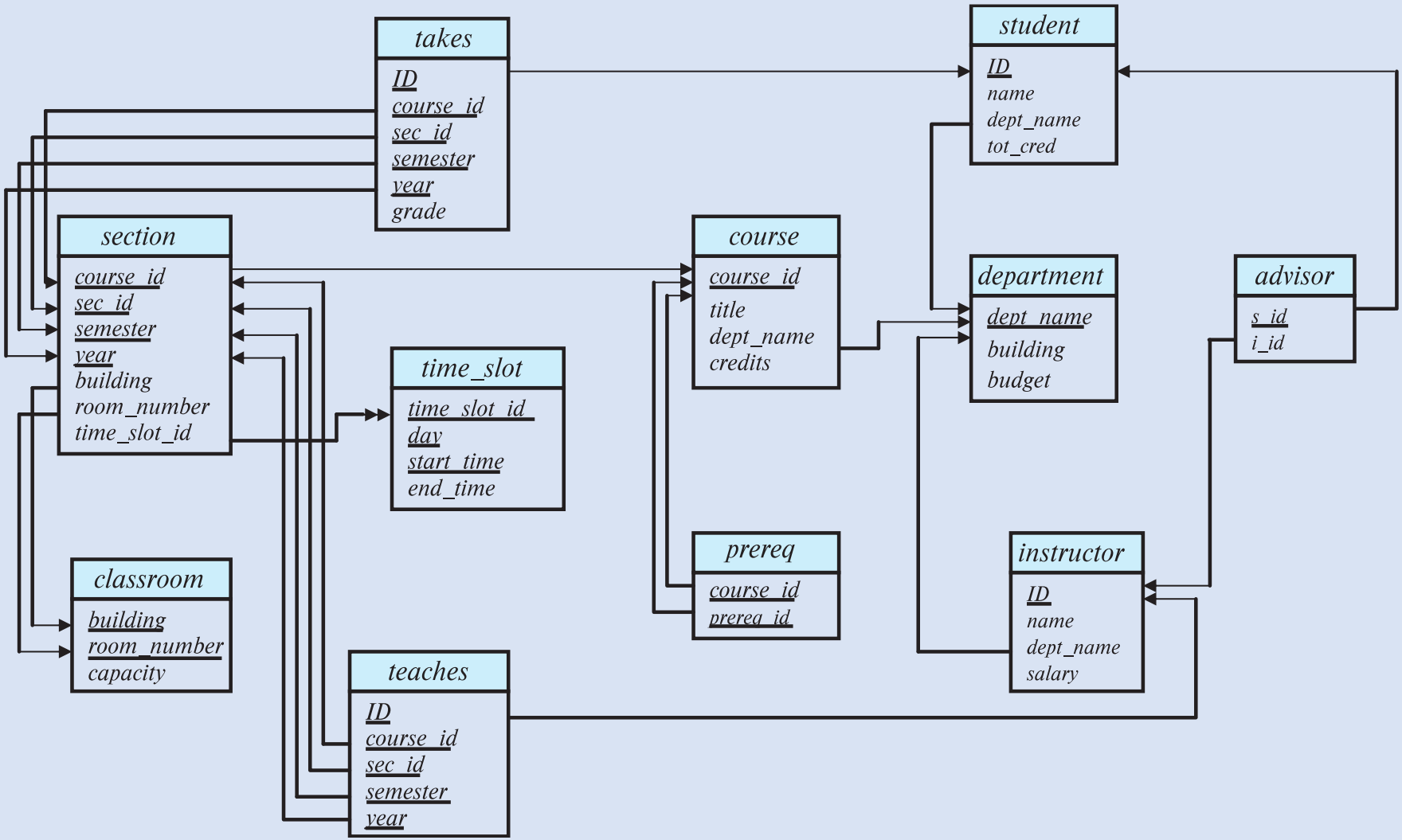
Schema Based Constraints

Foreign Key (FK) must satisfy

- Same domain
- Value of FK in a tuple either occurs as a value of PK, i.e., t1[FK]=t2[PK]



Schema Diagram for University Database



Thanks!