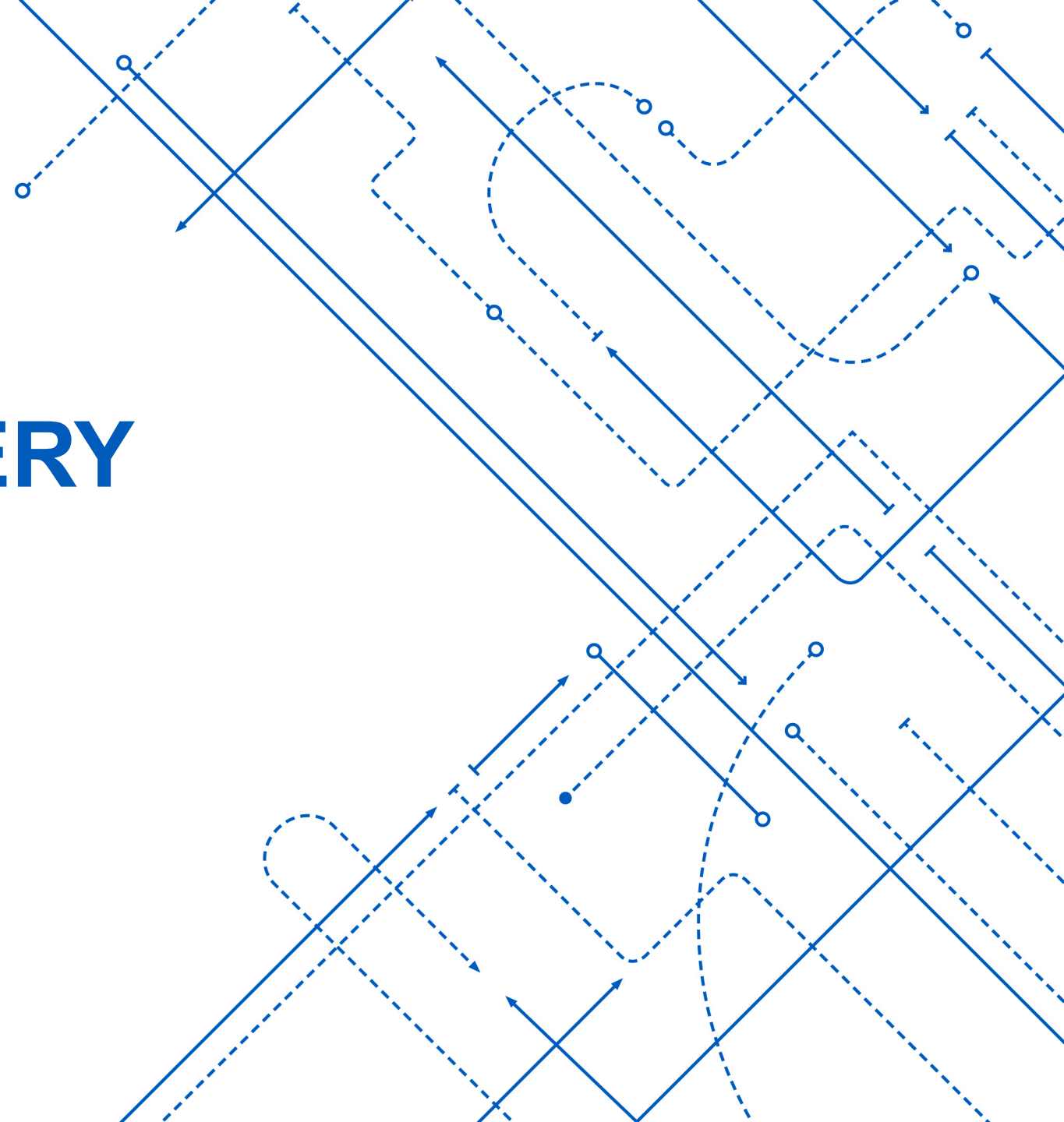


# CSE460/560 DATA MODELS AND QUERY LANGUAGES

## Relational Data Model

Cheng-En Chuang
(Slides Adopted from Jan Chomicki and Ning Deng)

University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

# Outline

1. Relational Data Model Basics
2. Constraints

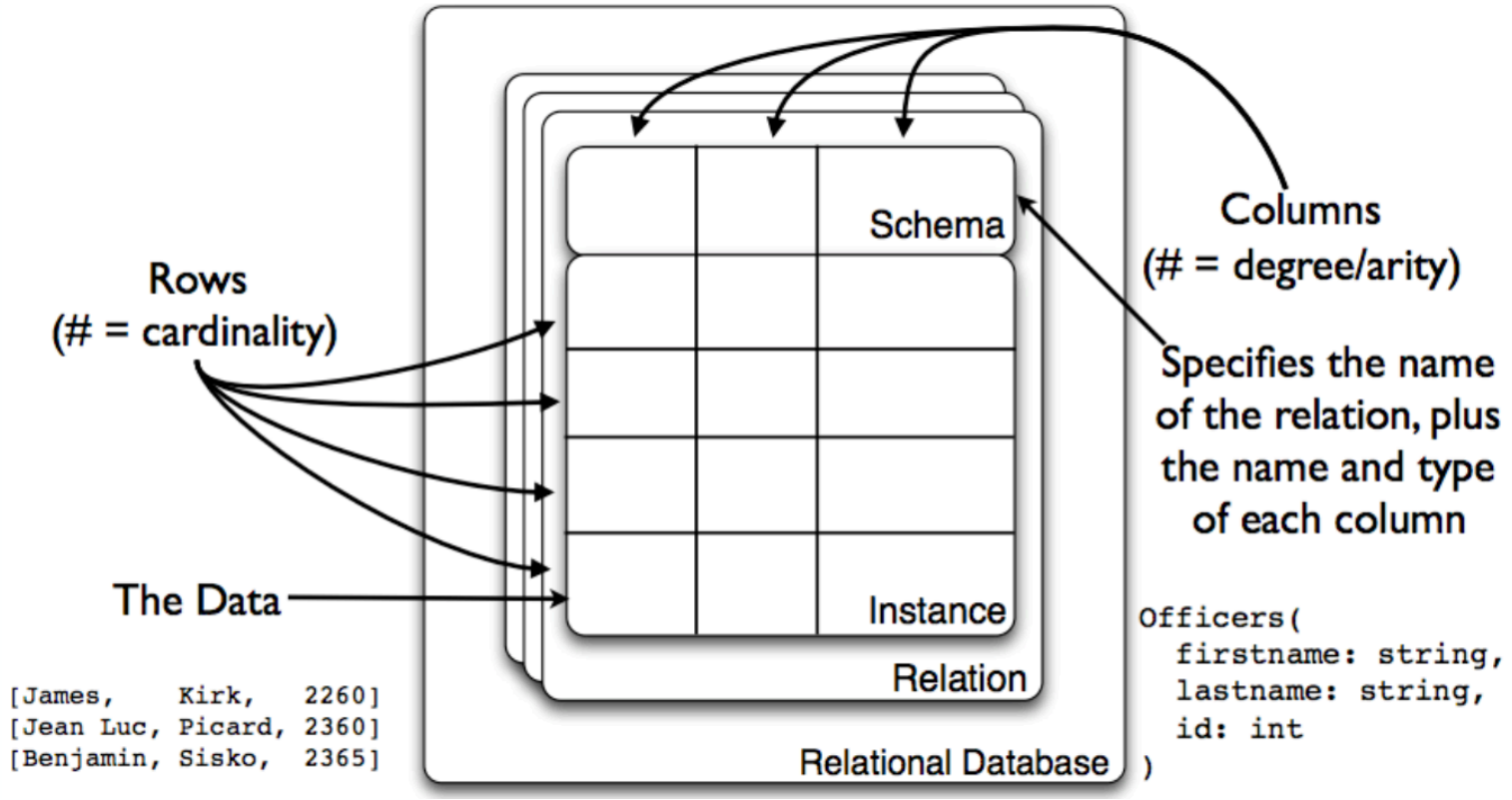
# Outline

1. **Relational Data Model Basics**
2. Constraints

# Relational Data Model in Brief

- Single way to represent data: a two-dimensional table, **relation**
  - Domain: predefined set of atomic values, such as integers, strings, etc.
  - Attributes: the columns of a relation are named by attributes
    - Each attribute has a domain.
    - Usually an attribute describes the meaning of entries in the column.
  - Schema
    - The name of a relation and the finite set of attributes of the relation
  - Tuple
    - The rows of a relation, in essence a sequence of values and nulls.
    - A tuple has one value for each attribute of the relation.
- A relation is **a set of tuples** (rows) with the same schema

# Relational Data Model




[Graph: Dr. Oliver Kennedy CSE 562 slides]

# Collection

Different types of collection:

| Uniqueness | | OrderMatters |
|---|---|---|
| <John, 1701> | <John, 1701> | <Marry, 1000> |
| <Marry, 1000> | <Marry, 1000> | <Marry, 1000> |
| <Jane, 1720> | <Jane, 1720> | <John, 1701> |
| | <Marry, 1000> | <Jane, 1720> |
| **Set** | **Bag** | **List** |

# Relation Instances

| *year* | *genre* | *title* | *length* |
|---|---|---|---|
| 1977 | sciFi | Star Wars | 124 |
| 1992 | comedy | Wayne's World | 95 |
| 1939 | drama | Gone With the Wind | 231 |

# Outline

1. Relational Data Model Basics
2. **Constraints**

# Integrity Constraints

- Logical conditions that have to be satisfied in **every** database instance
- Roles of constraints
  - **Guarding** against incorrect data into a database (data quality)
  - Providing **object identity** (key and foreign key constraints)
  - Representing **relationships and associations** between relations

# Domain Constraints

- Stronger restrictions on the contents of an attribute that provided by the attribute's type
- Example
  - 0 <= GPA <= 4.0
- Special domain constraints: NOT NULL
  - Attribute now allowed to contain NULL values

# Key Constraints

- How do you define a key? What is the properties of a key?
- A **key constraint** of a relation schema *R* is a set of attributes *S* (called a key) of *R*
- An instance *I* of *R* satisfies a key constraint *C* with key *S*
  - If *R* does NOT contain a pair tuples that agree on *S* but disagree on some other attributes
- Formal definition of key constraint
  - For each two tuples $t_1 \in I$, $t_2 \in I$, if $t_1[S] = t_2[S]$, then $t_1[A] = t_2[A]$ for every attribute $A$ in $R$

# Key Constraints

Find out the tuples that violates key constraints

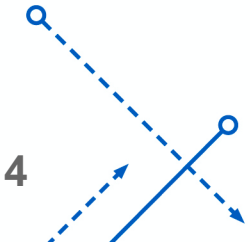
| Student ID | Name | DoB | Dept |
|---|---|---|---|
| 1111 | John | 1980/01/05 | CS |
| 1112 | Mary | 1987/10/12 | EE |
| 1121 | Jane | 1990/11/21 | CS |
| 1242 | Tom | 1994/05/24 | EE |
| 1111 | Bob | 1980/01/05 | CS |

# Properties of Keys

- Adequacy
  - Uniqueness of key values should be guaranteed by the properties of application domain
  - It is an error to have different tuples (in the same relation) with the same key values
- Minimality
  - A key should be as small as possible (good database design)
  - No subset of a key can also be designated as a key
- Key in conceptual database design
  - Keys of entity
  - Natural vs. artificial keys

# Primary Key

- There can be more than one candidate keys in a relation shema
  - One is selected as the primary key
    - Can’t be null
    - Typically used in indexing



# Relational Model is Value-based

- No duplicates
  - set of tuples
- No pointers
  - the only way referring to a tuple is by providing its key
- No notion of location
  - It is not possible to refer to the location of a tuple

# Foreign Keys

- Let $R_1, R_2$ be two relation schemas:
  - A foreign key is a pair of sets of attributes $(S_1, S_2)$ such that
    - $S_1 \subseteq R_1, S_2 \subseteq R_2$
    - $S_2$ is a key of $R_2$
    - The number of attributes and their respective domains in $S_1$ and $S_2$ are the same
    - The attribute values of $S_1$ also appear in the set of attribute values $S_2$
- A pair of instances $(I_1, I_2)$ satisfies a foreign key constraint $(S_1, S_2)$ if
  - For every tuple $t_1 \in I_1, t_1[S_1] = t_2[S_s]$ for some $t_2 \in I_2$
  - or $t_1[S_1]$ is null

# Foreign Keys

- What might be an issue if we try to update?

| StudentID | Name | BirthDate | Dept |
|---|---|---|---|
| 1111 | John | 1980/01/05 | CS |
| 1112 | Mary | 1987/10/12 | EE |
| 1121 | Jane | 1990/11/21 | CS |
| 1242 | Tom | 1994/05/24 | EE |

# Foreign Keys

| StudentID | Name | BirthDate | DeptID |
|---|---|---|---|
| 1111 | John | 1980/01/05 | D1 |
| 1112 | Mary | 1987/10/12 | D2 |
| 1121 | Jane | 1990/11/21 | D1 |
| 1242 | Tom | 1994/05/24 | D2 |

| DeptID | DName |
|---|---|
| D1 | Computer Science |
| D2 | Electric Engineering |
| D3 | Linguistics |

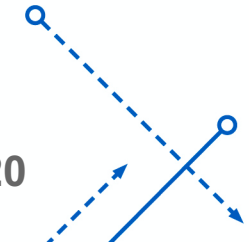
# Foreign Keys

| StudentID | Name | BirthDate | DeptID |
|---|---|---|---|
| 1111 | John | 1980/01/05 | D1 |
| 1112 | Mary | 1987/10/12 | D2 |
| 1121 | Jane | 1990/11/21 | D1 |
| 1234 | Joy | 1995/06/01 | |
| 1242 | Tom | 1994/05/24 | CHE |

| DeptID | DName |
|---|---|
| D1 | Computer Science |
| D2 | Electric Engineering |
| D3 | Linguistics |

# Summary

- Domain constraints: Limitations on valid values of a field
- Key constraints: A field(s) that must be unique for each row
- Foreign key constraints: A set of attributes referencing a key of another relation



# Constraints in DBMS

- DBMS support for constraints
  - All declared constraints are check after every transaction
  - If any constraint is violated, the transaction is backed out
  - Typically DBMSes support limited kinds of constraints
    - Key
    - Foreign key

# Enforcing Constraints

- Basic enforcement
  - Reject inserts/deletes/updates that violate constraints
- Insertion: Domain, Key, FK constraint
- Update: Domain, Key, FK constraint
- Delete: Only FK constraint
  - Do we need to care other constraints on delete? Why?

# Referntial Integrity Enforcement

- Foreign key constraints are complex
  - What happen when an insertion which references a non-exist foreign key?
    - Rejected
  - What happen when a referenced tuple being deleted?
    - Delete all referencing tuples
    - Disallow until there are no referencing tuples
    - Replace the referencing foreign key by some default value (or NULL)
  - What happen when a referenced tuple being updated?
    - Same as deletion

# Relational Schema in SQL

```
CREATE TABLE relation-name(
        Attr1 Type1 LC1,
        Attr2 Type2 LC2,
        ...
        Attrn Typen LCn,
       GC1, GC2, ... GCk );
```

where: LC1...LCn are local (tuple) constraints, and GC1,...GCk are global (table) constraints.

```
    CREATE TABLE Student(
        SID VARCHAR(10) NOT NULL,
        Name VARCHAR(20) NOT NULL,
        DOB DATE NOT NULL,
        Gender CHAR(1),
        PRIMARY KEY(SID)
);
```

# Integrity Enforcement

```
CREATE TABLE MajorsIn(
        SID VARCHAR(10) REFERENCES Student(SID)
        ON DELETE CASCADE
        ON UPDATE NO ACTION,
        MID INTERGER REFERENCES Major(MID)
        ON DELETE CASCADE
        ON UPDATE SET NULL,
        When DATE,
        PRIMARY KEY (SID,MID));
```

- CASCADE: Delete or update reference
- NO ACTION: Reject deletion or update
- SET DEFAULT v: Replace reference with v (typically a bad idea)
- SET NULL: replace reference with NULL

# Recommended Reading

Database Systems: The Complete Book
Chapter 2.1 – 2.3, 2.5