

# **The Relational Model (Part II)**

# What We have Learned

- **Basic concepts of relational models**
  - Schemas
  - Tables
- **Use SQL to create schema/tables**
- **Integrity constraints**
  - Keys
  - Domain constraints

# Integrity Constraints

- **Integrity constraints (ICs):** conditions specified on a database schema
- **Legal instances:** instances that satisfy ICs
- **Types of ICs**
  - Domain constraints: (e.g., age of students must be at least 18)
  - Keys
  - Foreign keys

# Primary Keys

- **A set of fields is a superkey if:**
  - No two distinct tuples can have same values in all key fields
- **A set of fields is a key for a relation if :**
  - It is a superkey
  - No subset of the fields is a superkey
  - i.e., a minimal superkey
- **>1 key for a relation?**
  - One of the keys is chosen (by DBA) to be the *primary key*.
  - The primary key of a relation cannot contain a NULL value as a value for their components

# Exercise

- Based on following instance, give some examples of (sets of) attributes that are *NOT keys for sure*.

SID	Name	Login	Age	GPA
53666	Jones	Jones@cs	18	3.4
53668	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	18	3.2

# Primary Keys in SQL

When  $>1$  key for a relation, one of the keys is chosen (by DBA) to be the *primary key*.

Syntax for Primary Key definition:

```
CREATE TABLE <name> (  
    <field1> <domain>,  
    <field2> <domain>,  
    ...  
    PRIMARY KEY (keyfield1, keyfield2,...)  
)
```



# Primary Key Example

- The schema of the Enrolled Relation is  
*Enrolled (sid: string, cid: string, grade: real)*
- Fact about data:
  - For a given student and a specific course, there is a single grade.
- Questions:
  - (1) What's the primary key?
  - (2) Write SQL to create the table with the key constraint.

# Is Primary Key Necessary?

- The primary key serves three purposes:
  - indicates that the column(s) should be unique
  - indicates that the column(s) should be non-null
  - document the intent that this is the unique identifier of the row
- When the primary keys are necessary?
  - Join the table with other tables
  - Construct index on the table



# Add/Drop Primary Key

## **Syntax for adding primary Key:**

```
ALTER TABLE <Table_Name> ADD PRIMARY KEY  
(keyfield1, keyfield2,...);
```

## **Syntax for Dropping Primary Key:**

```
ALTER TABLE <Table_Name> DROP PRIMARY KEY;
```

## **Syntax for showing the primary key:**

```
SHOW COLUMNS FROM Table_Name;
```

OR

```
SHOW KEYS FROM Table_Name:
```

# Candidate Keys in SQL

- When **>1** key for a relation, all keys are called **candidate keys** (one of which is chosen as the **primary key**).

```
CREATE TABLE <name> (  
    <field1> <domain>,  
    <field2> <domain>,  
  
    ...  
    PRIMARY KEY (field1, field2,...),  
    UNIQUE (Candidate_Key_1_field1,  
Candidate_Key_1_field2, ...),  
    UNIQUE (Candidate_Key_2_field1,  
Candidate_Key_2_field2, ...)  
);
```



# Exercise

- The schema of the Enrolled Relation is  
*Enrolled (sid: string, cid: string, grade: real)*
- Facts:
  - Students are of unique IDs.
  - Each student can take only one course, and receive a single grade for that course.
  - No two students in the same course receive the same grade.
- Question: what are the candidate keys?

```
CREATE TABLE Enrolled
(sid CHAR(20),
 cid CHAR(20),
 grade CHAR(2),
 PRIMARY KEY(sid),
 UNIQUE(cid, grade));
```

# Add/Drop Candidate Key (in MySQL)

## **Syntax for Adding Candidate Key:**

```
ALTER TABLE <Table_Name> ADD UNIQUE  
(keyfield1, keyfield2,...);
```

## **Syntax for dropping candidate key:**

**(note:** Unique key is dropped as INDEX)

- Step 1: SHOW INDEX FROM Table\_Name; // find the index name of the candidate key;
- Step 2: ALTER TABLE <Table\_Name> DROP INDEX index\_name; //index\_name = name of the candidate key

# Foreign Keys

- A foreign key of relation R is a set of attributes that are the keys of relation S.

Enrolled

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B

Key: (sid, cid)

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

Key: (sid)

- sid is not the key of Enrolled table, but the key of Students table.
- Thus sid is the foreign key of Enrolled table.

# Foreign Keys in SQL

Syntax for foreign Key definition:

```
CREATE TABLE <name> (  
    <field1> <domain>,  
    <field2> <domain>,  
    ...  
    PRIMARY KEY (field1, field2,...)  
    FOREIGN KEY (foreign_key_field1,...)  
    REFERENCES Table1*  
)
```

Table1\*: the table whose primary key is the specified foreign key.

# Foreign Keys in SQL

- **In the Enrolled Relation, only students listed in the Students relation should be allowed to enroll for courses.**

– Enrolled.sid is a foreign key!

```
CREATE TABLE Enrolled  
(sid CHAR(20), cid CHAR(20), grade float,  
PRIMARY KEY (sid,cid),  
FOREIGN KEY (sid) REFERENCES Students);
```

Enrolled

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

# Add/Drop Foreign Keys (in MySQL)

## Syntax for adding foreign keys:

```
ALTER TABLE <Table_Name> ADD FOREIGN KEY  
(keyfield1, keyfield2,...) REFERENCES table1  
[ON DELETE reference_option]  
[ON UPDATE reference_option];
```

## Syntax for dropping candidate key:

```
ALTER TABLE <Table_Name> DROP FOREIGN KEY  
key_name;
```



# Referential Integrity

- E.g. *sid* is a foreign key referring to **Students**:
  - Enrolled(*sid*: string, *cid*: string, *grade*: string)
  - If all foreign key constraints are enforced, referential integrity is achieved (i.e., no dangling references.)

# Enforcing Referential Integrity

- Remember Students and Enrolled; *sid* in Enrolled is a foreign key that references Students.
  - i.e., any enrolled student must be a student first.
- What should be done if an Enrolled tuple with a non-existent student id is inserted?
  - Reject it!*

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

## Enforcing Referential Integrity (cont.)

- **What should be done if a Students tuple is deleted?**
  - Option 1: Also delete all Enrolled tuples that refer to it.
  - Option 2: Disallow deletion of a Students tuple that is referred to.
  - Option 3: Set sid in Enrolled tuples that refer to it to a *default sid*.
  - Option 4: Set sid in Enrolled tuples that refer to it to a special value *null*, denoting '*unknown*' or '*inapplicable*'.
- **Similar if primary key of Students tuple is updated.**

Enrolled							
sid	cid	grade					
53666	Carnatic101	C					
53666	Reggae203	B					
53650	Topology112	A					
53666	History105	B					

Students				
sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8