CS275 – Intro to Databases

Basic SQL - *Chap. 4*

Data Creation

CREATE SCHEMA COMPANY authorization ‘jsmith’;

```
CREATE TABLE Students (sid      CHAR(20),
                       name     CHAR(30),
                       login    CHAR(20),
                       age      INTEGER,
                       gpa      REAL
                      );
```

Table Update

```
CREATE TABLE Students (sid     CHAR(20),
                       name    CHAR(30),
                       login   CHAR(20),
                       age     INTEGER,
                       gpa     REAL
                      );
```

```
INSERT
INTO Students (sid, name, login, age, gpa)
VALUES(53688, ‘Smith’, ‘smith@ee’, 18, 3.2);
```

```
DELETE
FROM `Students`
WHERE `Students`.`name` = ‘Smith’;
```

```
UPDATE `Students`
SET  `Students`.age = `Students`.`age`+1, `Students`.gpa = `Students`.`gpa` -1
WHERE `Students`.`sid` = 53688;
```

Table Update

```
UPDATE Students S
SET  S.gpa = S.gpa -1
WHERE S.gpa >= 3.3
```

Integrity Constraints

- An IC
  - is a condition specified on a database schema
  - restricts the data that can be stored in an instance
  - is enforced by DBMS

Integrity Constraints

- Candidate key

```
CREATE TABLE Students (sid          CHAR(20),
                       name         CHAR(30),
                       login  CHAR(20),
                       age    INTEGER,
                       gpa    REAL,
                       UNIQUE (name, age),
                       CONSTRAINTS StudentKey PRIMARY KEY (sid)
                      );
```

Integrity Constraints

- Primary key

```
CREATE TABLE Students (sid          CHAR(20),
                       name         CHAR(30),
                       login  CHAR(20),
                       age    INTEGER,
                       gpa    REAL
                       UNIQUE (name, age),
                       CONSTRAINTS StudentKey PRIMARY KEY (sid)
                       );
```

Integrity Constraints

- Foreign key

Student(sid: **string**, name: **string**, login: **string**, age: **integer**, gpa: **real**)

Enrolled(studentid: **string**, cid: **string**, grade: **string**)

- How do we ensure only certain students can enroll?

Integrity Constraints

- Foreign key on a table can refer to itself.
- Example?
  - Employee and Manager

Integrity Constraints

- Not NULL constraint
- When do we need this?

Integrity Constraints

```
CREATE TABLE Enrolled (studentid   CHAR(20),
                       cid      CHAR(20),
                       grade   CHAR(10)  NOT NULL,
                       PRIMARY KEY (studentid, cid),
                       FOREIGN KEY (studentid) REFERENCES Students
                      );
```

General Constraints

- Business rules
  - Only students with GPA > 2.5 are allowed to enroll
- Other examples
- Why are they not IC's?

Enforcing Integrity Constraints

INSERT
INTO Students (sid, name, login, age, gpa)
VALUES(null, ‘Smith’, ‘smith@ee’, 18, 3.2)

What other constraints can be violated by insertion?

- domain
- unique
- primary key
- foreign key

Enforcing Integrity Constraints

DELETE
FROM Students S
WHERE S.name='Smith'

What other constraints can be violated by deletion?

foreign key

Enforcing Integrity Constraints

UPDATE Students S

SET S.gpa = S.gpa – 0.1

WHERE S.gpa >= 3.3

What other constraints can be violated by update?

Enforcing Foreign Key Constraints

- Insertion
  - What happens if the studentid is invalid when inserting a new enrollment record?

Enforcing Foreign Key Constraints

- Deletion
  - What happens when deleting a student record?

Enforcing Foreign Key Constraints

- Deletion
  - What happens when deleting a student record?
    - Do not delete.

```
CREATE TABLE Enrolled (studentid   CHAR(20),
                       cid     CHAR(20),
                       grade           CHAR(10),
                       PRIMARY KEY (studentid, cid),
                       FOREIGN KEY (studentid) REFERENCES students
                             ON DELETE NO ACTION
                       );
```

Enforcing Foreign Key Constraints

- Deletion
  - What happens when deleting a student record?
    - Remove the record in the Student table
    - Remove all records in the Enrollment table with the given StudentID.

```
CREATE TABLE Enrolled (studentid      CHAR(20),
                       cid      CHAR(20),
                       grade    CHAR(10),
                       PRIMARY KEY (studentid, cid),
                       FOREIGN KEY (studentid) REFERENCES students
                               ON DELETE CASCADE
                       );
```

Enforcing Foreign Key Constraints

- Deletion
  - What happens when deleting a student record?
    - Remove the record in the Student table
    - Set the StudentID to NULL for all records in Enrollment with the given StudentID.

```
CREATE TABLE Enrolled (studentid      CHAR(20),
                       cid      CHAR(20),
                       grade    CHAR(10),
                       PRIMARY KEY (studentid, cid),
                       FOREIGN KEY (studentid) REFERENCES students
                            ON DELETE SET NULL
                       );
```

Enforcing Foreign Key Constraints

- Deletion
  - What happens when deleting a student record?
    - Remove the record in the Student table
    - Set the StudentID to the default value for all records in Enrollment with the given StudentID.

```
CREATE TABLE Enrolled (studentid      CHAR(20) DEFAULT '53666',
                       cid      CHAR(20),
                       grade    CHAR(10),
                       PRIMARY KEY (studentid, cid),
                       FOREIGN KEY (studentid) REFERENCES students
                            ON DELETE SET DEFAULT
                       );
```

Enforcing Foreign Key Constraints

- Update
  - What happens when updating the ID of a student?

Enforcing Foreign Key Constraints

- Update
  - What happens when updating the ID of a student?
    - ON UPDATE NO ACTION
    - ON UPDATE CASCADE
    - ON UPDATE SET DEFAULT
    - ON UPDATE SET NULL

Enforcing Foreign Key Constraints

- Update
  - What happens when updating the ID of a student?

```
CREATE TABLE Enrolled (studentid   CHAR(20),
                       cid     CHAR(20),
                       grade           CHAR(10),
                       PRIMARY KEY (studentid, cid),
                       FOREIGN KEY (studentid) REFERENCES students
                             ON DELETE CASCADE
                             ON UPDATE NO ACTION
                       );
```

Data Retrival

- SELECT *

  FROM Students S

  WHERE S.age < 18

Data Retrival

- SELECT S.name, S.login
  FROM Students S
  WHERE S.age < 18