Xi'an Jiaotong-Liverpool University
西交利物浦大學

SQL - Table and Data Part 2

Jianjun Chen

Contents

- Foreign key
  - Reference options
- Altering table definitions
- Dropping tables
- Tutorial on creating tables

Foreign Key

Reference options

Droping table with foreign keys

The Syntax of Foreign Key

- A foreign key consists of following parts:
  - A constraint name.
  - Columns of the referencing table.
  - Referenced table and referenced columns.
  - Reference options.

```
CONSTRAINT name
    FOREIGN KEY
        (col1, col2, ...)
    REFERENCES
    table-name
        (col1, col2, ...)
    [ON UPDATE ref_opt
    ON DELETE ref_opt]

    ref_opt: RESTRICT | CASCADE | SET NULL | SET DEFAULT
```

Columns of the referencing table

Referenced table name

Referenced columns

Reference options

FK Important Notice

1 Foreign key must reference a unique key or primary key

2 Data types of both columns must be compatible

```
CREATE TABLE branch (
    branchNo CHAR(4) PRIMARY KEY,
    street VARCHAR(50),
    ...
);

CREATE TABLE staff (
    staffNo CHAR(6) PRIMARY KEY,
    fName VARCHAR(20),
    branchNo CHAR(4),
    CONSTRAINT FK_staff_branchNo FOREIGN KEY
        (branchNo)
    REFERENCES
        branch (branchNo)
);
```

3 Column list must be enclosed with brackets

Foreign Key

it means the branchNo column of staff

- After applying FK, values of `staff`.`branchNo` will be checked by the database, the value must either be:
  - An existing value from `branch`.`branchNo`.
  - Or `null`.

- Given the `branch` table on the right, insert following tuples into `staff` (See example `TD-2.1`)
  - ('S1', 'staff1', 'b001'),
  - ('S2', 'staff2', 'B007'),
  - ('S3', 'staff3', 'B001'),
  - ('S4', 'staff4', 'B002'),
  - ('S1', 'staff5', 'B002');

Branch

branchNo	street	...
B001	...	...
B002	...	...
B005	...	...

- Which tuple will cause error?

MySQL: String Comparison

- In MySQL, string comparison is case-insensitive.

Branch

branchNo	street	...

```
INSERT INTO branch VALUES
('B001', 'city1', 'street1', 'postcode1'),
('b001', 'city1x', 'street1x', 'postcode1x');
```

- The `'b001'` above will violate the primary key constraint.
- Solution:

```
CREATE TABLE `branch` (
    `branchNo` CHAR(4) BINARY NOT NULL,
    PRIMARY KEY (`branchNo`),
    ...);
```

The Binary Keyword

- The `BINARY` keyword instructs MySQL to compare the characters in the string using their underlying ASCII values rather than just their letters.
  - Treating the storage as binary values.

```
CREATE TABLE `branch` (
    `branchNo` char(4) BINARY NOT NULL,
    PRIMARY KEY (`branchNo`),
    ...);
```

- In other databases, string comparison can be implemented differently!

Foreign Keys and Tuple Updates

- We just saw that attempts to add non-existing `branchNo` to `Staff` were rejected by DBMS.
- But what happens when we change/delete existing `branchNo` in Branch that are being referenced by `Staff`?
- What strategies can you think of if you were the designer?

Reference Options

- There are several options when this occurs:
  - **`RESTRICT`** – stop the user from doing it
    - The default option
  - **`CASCADE`** – let the changes flow on
  - **`SET NULL`** – make referencing values null
  - **`SET DEFAULT`** – make referencing values the default for their column

- These options can be applied to one or both kinds of the table updates:
  - **`ON DELETE`** – What will happen if referenced values are deleted.
  - **`ON UPDATE`** – What will happen if referenced values are updated.

On Update/Delete Set NULL

- Assume we delete in Branch table.
  - All ‘B005’ in the Staff table will be set to null.

- If we change ‘B005’ to ‘B006’.
  - All ‘B005’ will be set to null…Good decision?

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

NULL

On Update/Delete Cascade

- Assume we change ‘B005’ to ‘B006’ in Branch table
  - All ‘B005’ in Staff table will be changed to ‘B006’.
  - Seems reasonable
- Assume we delete ‘B005’ in Branch table.
  - All tuples with ‘B005’ in Staff table will also be deleted!
  - Good decision?

On Update/Delete Set Default

- Assume we delete or change ‘B005’ in Branch table.
- All ‘B005’ in Staff table will be changed to the default value of Staff (branchNo).
- This feature is not available in MySQL.

Final FK Definition

- The following FK definition is reasonable.

```
CONSTRAINT `FK_staff_branchNo`
    FOREIGN KEY (`branchNo`)
    REFERENCES `branch` (`branchNo`)
        ON DELETE SET NULL
        ON UPDATE CASCADE
```

- The default option (restrict) also works
  - How would you then update a branchNo?

```
CONSTRAINT staffFK
    FOREIGN KEY (branchNo)
    REFERENCES Branch (branchNo)
```

Example TD-2.2

- What will happen to the `staff` table if the following lines were executed?
  1. DELETE FROM branch WHERE branchNo = 'B001';
  2. UPDATE branch SET branchNo = 'B007' WHERE branchNo = 'B005';

```
CONSTRAINT `FK_staff_branchNo`
    FOREIGN KEY (`branchNo`)
    REFERENCES `branch` (`branchNo`)
        ON DELETE SET NULL
        ON UPDATE CASCADE
```

staffNo	fName	branchNo
S1	staff1	b001
S2	staff2	B001
S3	staff3	B002
S4	staff4	B002
S5	staff5	B005

Altering Tables

Alter column details

Alter table constraints

ALTER

- Add column:

```
ALTER TABLE table_name
    ADD column_name datatype [options like UNIQUE …];
```

- Drop column:

```
ALTER TABLE table_name DROP COLUMN column_name;
```

- Modify column name and definition:

```
ALTER TABLE table_name
    CHANGE COLUMN
    col_name new_col_name datatype [col_options];
```

- Modify column definition only:

```
ALTER TABLE table_name
    MODIFY COLUMN
    column_name datatype [col_options];
```

Example TD-2.3

```
ALTER TABLE staff ADD `lName` VARCHAR(20) NOT NULL;

ALTER TABLE staff DROP COLUMN `lName`;

ALTER TABLE staff CHANGE COLUMN `fName`
    `first_name` VARCHAR(20) NOT NULL;

ALTER TABLE staff MODIFY COLUMN
    `first_name` VARCHAR(40) NOT NULL;
```

Adding Constraints

- To add a constraint:

```
ALTER TABLE table-name
    ADD CONSTRAINT name definition;
```

- Examples `TD-2.3`:

```
ALTER TABLE branch
    ADD CONSTRAINT ck_branch UNIQUE (street);
```

```
ALTER TABLE staff ADD CONSTRAINT fk_staff_staff
    FOREIGN KEY (branchNo) REFERENCES branch (branchNo);
```

Removing Constraints

- To remove a constraint:

```
ALTER TABLE table-name
DROP INDEX name | DROP FOREIGN KEY name | DROP PRIMARY KEY
```

Can be used to drop unique keys

- Examples `TD-2.3`:

```
ALTER TABLE staff DROP PRIMARY KEY;
```

```
ALTER TABLE staff DROP FOREIGN KEY fk_staff_staff;
```

```
ALTER TABLE branch DROP INDEX ck_branch;
```

Deleting Tables

Drop

set foreign key check

Deleting Tables

- You can delete tables with the DROP keyword

```
DROP TABLE [IF EXISTS] table-name1, table-name2…;
```

- Tables will be dropped in that exact order
  - All tuples in the dropped tables will be deleted as well.
  - Undoing is sometimes **impossible**.

- Foreign Key constraints will prevent DROPS under the default RESTRICT option, to overcome this:
  - Remove the foreign key constraints first then drop the tables.
  - Drop the tables in the correct order (referencing table first).
  - Turn off foreign key check temporarily.

Example `TD-2.4`

- Please read and run the script on the LearningMall.

Tutorial on Defining Tables

Index

Choosing columns for PK

Choosing data type.

Surrogated primary keys

The “csse-mysql” Server

- During labs, you can use the phpMyAdmin installed on:

`csse-mysql.xjtlu.edu.cn`

- If your email account is Jianjun.Chen21@...
  - Your account will be JianjunChen21, and the password is your ID.
  - Remove the dot, preserve case and number, no email part.

- phpMyAdmin provides graphical user interfaces for:
  - Creating tables
  - Checking table constraints
  - Inserting tuples
  - Modifying tuples (limited rows displayed though)
  - Exporting database with data.

explained in the lecture

Tutorial Start

- This part of the lecture will partially cover lab 1.