Lecture 13: Building database with SQL

BADM/ACCY 352

Spring 2017

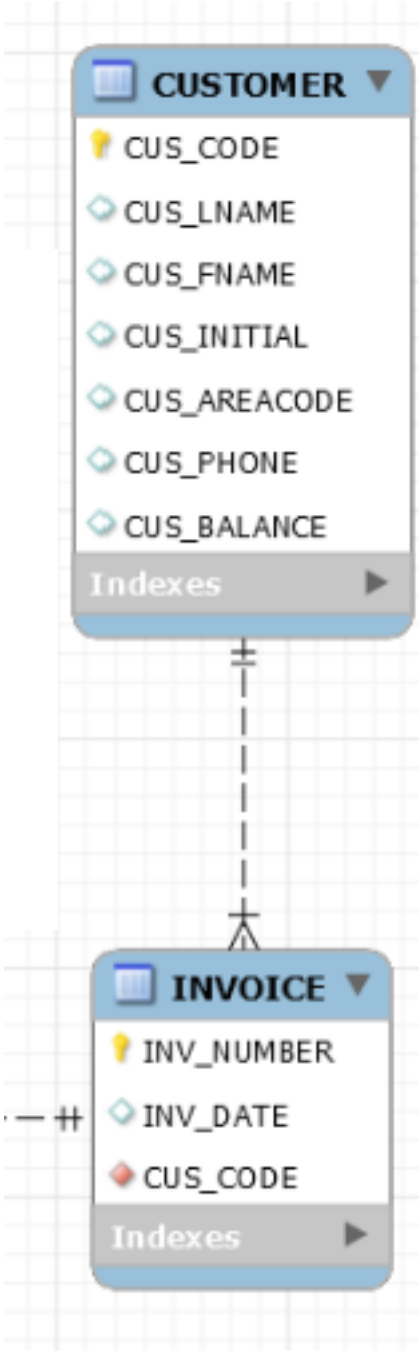
Instructor: Yi Yang, PhD

Previous Lectures

- We have learned how to retrieve data from relational database.
  - Basic data retrieval
  - Group by
  - Join

```
SELECT columns
FROM table
[WHERE condition]
[GROUP BY column]
[HAVING condition]
[ORDER BY column [ASC | DESC]]
[LIMIT count]
```

```sql
SELECT C.CUS_CODE, COUNT(*) AS NUM_INV
FROM CUSTOMER AS C INNER JOIN INVOICE AS I
ON C.CUS_CODE = I.CUS_CODE
WHERE CUS_AREACODE=615
GROUP BY C.CUS_CODE
HAVING NUM_INV > 1
ORDER BY C.CUS_CODE;
```



SQL Command

- **SQL commands fit into two broad categories:**
  - Data definition language (DDL)
    - **Create** database objects, such as tables, indexes, and define access rights.
  - Data manipulation language (DML)
    - **Retrieve** data from database tables
    - **Update** and **delete** data

- CRUD: Create, Read, Update, Delete
- Or simply: Read and Write

Database operation

We are using databases every day.

- Grocery store: you purchase products.
- Bank: you transfer money to your friend.
- Facebook: you click ‘like’ button of a post.
- Airline: you book ticket.

Step to build a database

0. Database design (data modeling).
1. Create database
2. Given the data model, create tables, define relationships between tables
3. Insert/edit/delete/update data
4. Query database
5. (Optional) delete table/database

Create/Use/Drop database

Create a database

```
CREATE DATABASE [IF NOT EXISTS] db_name
```

After you create a database, there is no tables in the database.

Use a database

```
USE db_name;
```

Drop a database

```
DROP DATABASE [IF EXISTS] db_name
```

Database privilege

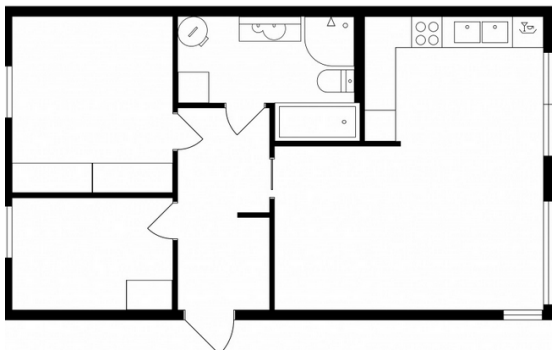
Privileges are authority levels used to access the database itself, access objects within the database, manipulate data in the database, and perform various administrative functions within the database.

☑ ALL PRIVILEGES	
☑ ALTER	☑ CREATE
☑ CREATE ROUTINE	☑ CREATE TEMPORARY TABLES
☑ CREATE VIEW	☑ DELETE
☑ DROP	☑ EXECUTE
☑ INDEX	☑ INSERT
☑ LOCK TABLES	☑ REFERENCES
☑ SELECT	☑ SHOW VIEW
☑ TRIGGER	☑ UPDATE

Database Schema

- Database schema = **Data model + data type**
- A *database schema* is the collection of table schema for the whole database.
- A *table schema* is the logical definition of a table – it defines what the name of the table is, and what the name and type of each column is.

floor plan/blueprint



schema

house

database

room

table

Database Schema

- In *relational databases*, schema is defined before the database is implemented. It's not recommended to change the schema frequently.
- NoSQL databases are schema-free.
- One of the most significant differences between relational databases and NoSQL databases.

Create Table (simple version)

```
CREATE TABLE table_name(
    column1 datatype,
    column2 datatype,
    ...
);
```

- For example, to create a PRODUCT table.

```
CREATE TABLE PRODUCT(
P_CODE varchar(10),
P_DESCRIPT varchar(35),
P_INDATE datetime,
P_PRICE float(8),
P_DISCOUNT float(8),
V_CODE int
);
```

Create Table (simple version)

- However, the tables created are ‘isolated’.
- We can not guarantee data integrity.
- Thus, we need to apply constraints to the tables.

Constraints

- NOT NULL: ensure that a column does not accept nulls
- UNIQUE: ensure all values in a column are unique
- PRIMARY KEY
- AUTO_INCREMENT: automatically increment primary key value by 1
- FOREIGN KEY: ensure referential integrity
  - the value of V_CODE in PRODUCT must refer to an existing value in VENDOR.
  - you can't delete/update a value in VENDOR if that value is referred in PRODUCT.
- DEFAULT: assign default value to an attribute

Create Table

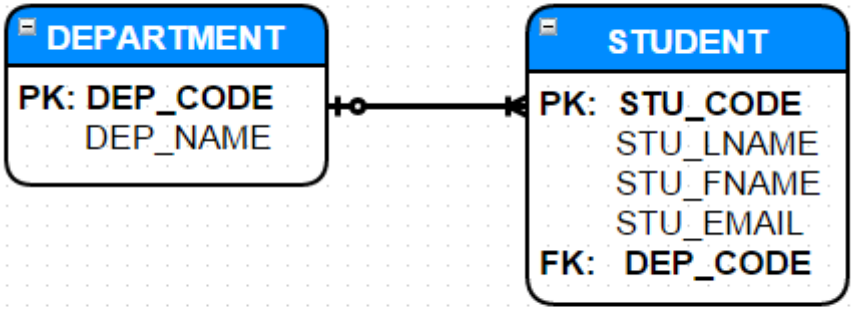
```
CREATE TABLE table_name(
    column1 datatype [constraint],
    column2 datatype [constraint],
    .....
    PRIMARY KEY (column1, ...),
    FOREIGN KEY (column1, ...) REFERENCES table1_name (column1)
);
```

- For example, to create a PRODUCT table that links to VENDOR table via V_CODE.

```
CREATE TABLE PRODUCT1(
    P_CODE varchar(10) NOT NULL UNIQUE,
    P_DESCRIPT varchar(35) NOT NULL,
    P_INDATE datetime NOT NULL,
    P_PRICE float(8) NOT NULL,
    P_DISCOUNT float(8) DEFAULT 0.0,
    V_CODE int,
    PRIMARY KEY(P_CODE),
    FOREIGN KEY(V_CODE) REFERENCES VENDOR(V_CODE)
);
```

Practice

Give then following relational model, create two tables: Department and Student.



Practice

```
CREATE TABLE DEPARTMENT(
    DEP_CODE int NOT NULL UNIQUE,
    DEP_NAME varchar(30) NOT NULL,
    PRIMARY KEY (DEP_CODE)
);

CREATE TABLE STUDENT(
    STU_CODE int NOT NULL UNIQUE,
    STU_LNAME varchar(20) NOT NULL,
    STU_FNAME varchar(20) NOT NULL,
    STU_EMAIL varchar(20),
    DEP_CODE int,
    PRIMARY KEY (STU_CODE),
    FOREIGN KEY (DEP_CODE) REFERENCES DEPARTMENT (DEP_CODE)
);
```

Drop Table

```
DROP TABLE IF EXISTS table_name;
```

If you drop a table whose PK is used in other tables as FK, you will get an error.

- "cannot delete or update a parent row: a foreign key constraint fails"

Populate table with data

Syntax

```
INSERT INTO table_name VALUES(value1, value2,.....);
```

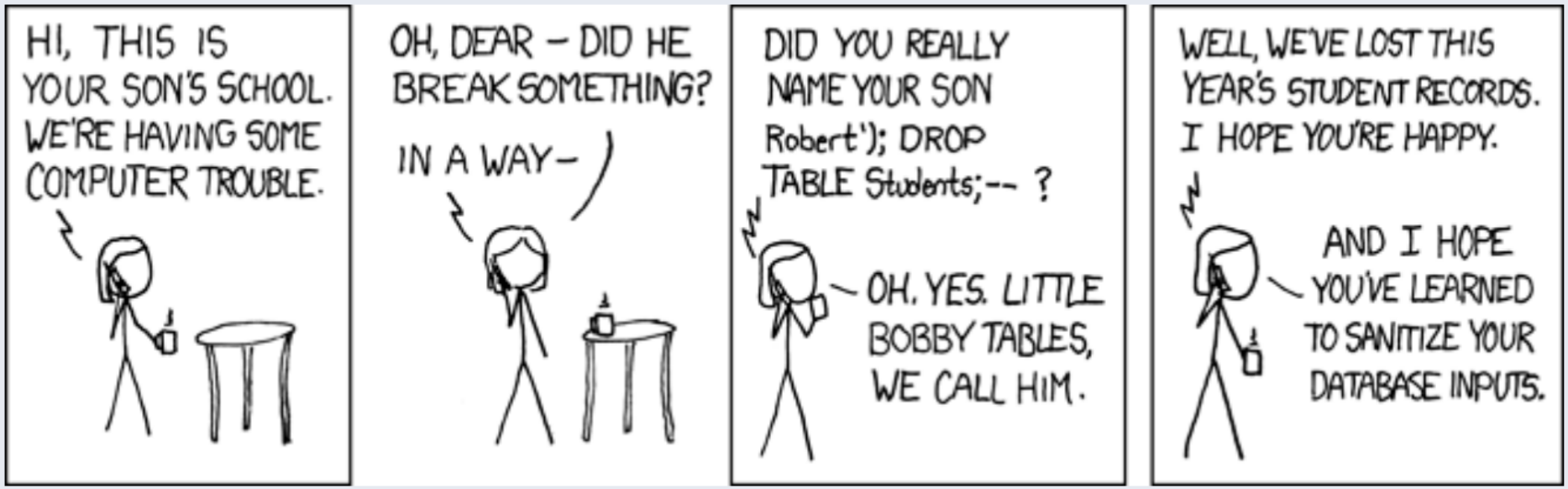
Example

```
INSERT INTO DEPARTMENT VALUES(0001, 'Business Administration');
```

- The row contents are entered between parentheses.
- Character and date values must be entered between quotes.
- Numerical entries are not enclosed in quotes.
- Attributes are separated by commas.
- You can insert NULL value for attributes that don't have NOT NULL constraints.

Database security: SQL injection attack

Security - http://xkcd.com/327/



A normal insert query:

INSERT INTO Students (firstname) VALUES ('Robert');

This malicious query:

INSERT INTO Students (firstname) VALUES ('Robert');DROP TABLE Students;-- ');

Updating Table Rows

- Sometimes you need to modify data in a table.
- Syntax

```
UPDATE table_name
SET column1 = value1,
    column2 = value2,
    ...
 WHERE condition;
```

- E.g. use the primary key to locate the correct row

```
UPDATE STUDENT SET STU_GPA = 3.3
WHERE STU_CODE = 115;
```

Updating Table Rows (cont.)

- If more than one attribute is to be updated.

```
UPDATE STUDENT
SET STU_GPA = 3.3, STUDENT_LNAME='Smith'
WHERE STU_CODE = 115;
```

- If you do not specify WHERE condition, the UPDATE command will apply changes to ***all*** rows in the table.

Deleting Table Rows

- Syntax

```
DELETE FROM table_name
[WHERE condition];
```

- You can use primary key or any attributes to find the exact record.

```
DELETE FROM STUDENT WHERE STU_CODE=115;

DELETE FROM STUDENT WHERE STU_LNAME='Smith';
```

- What happened if no WHERE condition is specified? **All rows from the table will be deleted.**

Changing column with ALTER

- Relational database is very efficient in row-based operation, e.g. adding a new row, choosing a row. (Why?)
- Sometimes, we may need to change table structures by changing column name/datatype and by adding a new columns

Q: I periodically need to make changes to tables in mysql database, mostly adding columns, any suggestions?

A: Don't. No really. Just don't. It should be a very rare occasion when this is ever necessary.

ALTER TABLE

- Add a column

```
ALTER TABLE PRODUCT ADD P_SALE CHAR(1);
```

- Modify a column datatype

```
ALTER TABLE PRODUCT MODIFY P_SALE INTEGER;
```

- Drop a column

```
ALTER TABLE PRODUCT DROP COLUMN P_SALE;
```

ALTER TABLE

- Table alternation performs bad with large dataset.

- Why?
  - Creating an empty table with the desired new structure
  - inserting all the data from the old table into the new one
  - deleting the old table.
- Again, you design schema carefully before implementing the database.

How FKs help guarantee referential integrity

For INSERT or UPDATE: MySQL rejects any insert or update operation that attempts to create a foreign key value in a child table if there is no a matching candidate key value in the parent table.

- It's not possible to insert a student whose department code does not exist.

For DELETE: What will MySQL do when a delete operation affects a key value in the parent table that has matching rows in the child table?

- If you delete your FB account, do your comments/likes still exist?

How FKs help guarantee referential integrity

MySQL supports several actions:

- CASCADE: Delete or update the row from the parent table, and automatically delete or update the matching rows in the child table
- Set NULL: Delete or update the row from the parent table, and set the foreign key column or columns in the child table to NULL.
- Restrict: Rejects the delete or update operation for the parent table.

What happens to your comments/likes, if you delete your Facebook account

USER

FB_ID	FB_PROFILE
~~Alice.001~~	…
Bob.001	…
Charlie.001	…

Comments

FB_ID	Comm_ID	Comment
~~Alice.001~~	~~C001~~	~~I like it~~
~~Alice.001~~	~~C002~~	~~LOL..~~
Charlie.001	C003	Love it!

Summary

- CREATE a database
- INSERT/UPDATE/DELETE data

- Wednesday: Lab session on building database