**DASC 5333**

8/27/2024

**2.2 Simple schema (structures of the tables) and keys**

**Relations/tables (primary key underscored):**

Grade(grade, gradePoint)
School(schoolCode, schoolName)
Department(deptCode, deptName, schoolCode, numStaff)
Faculty(facId, fname, lname, deptCode, rank)
Course(courseId, rubric, number, title, credits)
Class(classId, courseId, semester, year, facId, room)
Student(StuId, fname, lname, major, minor, ach, advisor): stuId is the PK of Student
Enroll(stuId, classId, grade, n\_alerts): stuId in not the PK of Enroll

The primary key of a table uniquely identifies a row in the table.

CREATE TABLE IF NOT EXISTS Student (

 stuId INT NOT NULL,

 fname VARCHAR(30) NOT NULL,

 lname VARCHAR(30) NOT NULL,

 major CHAR(4) NULL,

 minor CHAR(4) NULL,

 -- ach: accumulated credit hours, including transferred credits.

 ach INTEGER(3) UNSIGNED NULL DEFAULT 0,

 advisor INT NULL,

 CONSTRAINT Student\_stuId\_pk PRIMARY KEY(stuId),

 -- an artificial example of a CHECK constraint.

 CONSTRAINT Student\_ach\_cc CHECK ((ach>=0) AND (ach < 250)),

 CONSTRAINT Student\_major\_fk FOREIGN KEY (major)

 REFERENCES Department(deptCode) ON DELETE CASCADE,

 CONSTRAINT Student\_minor\_fk FOREIGN KEY (minor)

 REFERENCES Department(deptCode ) ON DELETE CASCADE,

 CONSTRAINT Student\_advisor\_fk FOREIGN KEY (advisor)

 REFERENCES Faculty(facId)

);

A foreign key of a table (child table) is a column that references a primary key in another table (parent/host/referenced table)

MySQL enforce referential integrity: A FK value must exists as a PK value in the parent table.

INSERT INTO Student(stuId, fname, lname, major, minor, ach, advisor) VALUES

 (100000,'Bun','Yue','CSCI','CITEC',22,1011);

ERROR 1062 (23000): Duplicate entry '100000' for key 'PRIMARY'

MySQL enforce entity integrity: no two rows should have the same PK values.

PK for Enroll: {classId, stuId}; composite PK.

CREATE TABLE IF NOT EXISTS Enroll(

 stuId INT NOT NULL,

 classId INT NOT NULL,

 grade VARCHAR(2) NULL,

 n\_alerts INT NULL,

 CONSTRAINT Enroll\_classId\_stuId\_pk PRIMARY KEY (classId, stuId),

 CONSTRAINT Enroll\_classNumber\_fk FOREIGN KEY (classId)

 REFERENCES Class(classId) ON DELETE CASCADE,

 CONSTRAINT Enroll\_stuId\_fk FOREIGN KEY (stuId)

 REFERENCES Student (stuId) ON DELETE CASCADE,

 CONSTRAINT Enroll\_grade\_fk FOREIGN KEY (grade)

 REFERENCES Grade (grade) ON DELETE CASCADE

);

**Foreign keys:**

1. Student(advisor) references Faculty(facId)
2. Student(major) references Department(deptCode)
3. Student(minor) references Department(deptCode)
4. Faculty(deptCode) references Department(deptCode)
5. Department(schoolCode) references School(schoolCode)
6. Enroll(stuId) references Student(stuId)
7. Enroll(classId) references Class(classId)
8. Enroll(grade) references Grade(grade)
9. Class(courseId) references Course(courseId)
10. Class(facId) references Faculty(facId)
11. Course(Rubric) references Department(deptCode)

[toyu\_relations.pdf](https://dcm.uhcl.edu/yue/courses/joinDB/Fall2024/notes/toyu/toyu_relations.pdf):

* Contains the relation schema and instance of toyu.
* Will be provided in the mid-term and final examination.

[toyu\_schema.pdf](https://dcm.uhcl.edu/yue/courses/joinDB/Fall2024/notes/toyu/toyu_schema.pdf):

* Contains a more detailed description of the relation schema of toyu, including candidate keys, foreign keys, nullable and non-nullable attributes/columns.
* Includes functional dependencies and the highest normal form of every table.

**3.2 MySQL Clients Setup:**

It is common to use multiple clients to connect to a backend database server. In this course, we will use three clients in our classes. You may use your own favorite clients (e.g., MySQL Workbench). However, I may not be as helpful in these clients.



[1] MySQL Command-Line Prompt: will be used in this class.

1. Come with (1) XAMPP/MariaDB or (2) MySQL 8.x. (Note that the two versions of mysql prompt are somewhat different.)
	1. MariaDB mysql: <https://mariadb.com/kb/en/mysql-command-line-client/>
	2. MySQL 8.x mysql: <https://dev.mysql.com/doc/refman/8.0/en/mysql.html>
2. A command line text-based MySQL-specific client.
3. You may set the PATH variable so you can call mysql prompt anywhere, such as by adding "c:\xampp\mysql\bin" in the PATH system environment variable.

***Example:***

**mysql –h *host* -u *user* -p**

or

**mysql –h *host* -u *user* -p -P port\_number**

[2] PhPMyAdmin

1. A Web-based GUI client focused on DB administration.
2. After starting both MySQL and Apache in XAMPP, go to localhost in your browser.
3. MySQL specific.

[3] HeidiSQL: will be used in this class

1. A general Windows GUI SQL client

[4] MySQL Workbench:

1. A GUI MySQL client that comes with MySQL 8.x (but not XAMPP)

8/20/2024

Self annotation

**Introduction to the course
DASC 5333/CSCI 4333**

by K. Yue

**1. Promotion**

* This course is (hopefully) one of the more useful CS/DS courses for students.
* World data is estimated to double every two years.

**2. How to be successful in the course**

General Course Suggestions:

1. Course expectation is demanding.
2. Please consider forming the habit of listening carefully and asking a lot of questions.

General Professionalism:

1. Attitude
2. Be considerate.
3. Be helpful and useful to others.
4. Be a good listener.
5. Be responsive.
6. Hardworking
7. Attention to details.
8. Focus: uni-tasking

Some general tips:

1. Engagement: Participate. Ask questions, a lot of them. Help others. Plan ahead.
2. Preparation: start as early as possible and do not fall behind.
3. Don’t copy and paste. Instead, copy, integrate, and apply.
4. [SEE-I](https://en.everybodywiki.com/SEE-I): State, Elaborate, Exemplify and illustrate.
5. Form good habits.

Some good traits of Computer and Data Scientists:

1. Habits of trying to make sense of stuff.
2. Intellectual curiosity.
3. Tinkering and experimentation.
4. Open-minded, not dogmatic.
5. A large tool set.

**3. Resources**

* Companion materials of our textbook: please consult the course page in Canvas for additional resources related to the textbook.
* Contents of the course will be based mostly on
	1. Lecture notes posted in the course website: <http://dcm.uhcl.edu/yue/courses/joindb/current/index.html>.
	2. Classroom demonstrations.
	3. Assignments.
* Please read the appropriate pages in the textbook and lecture notes in this site *before* coming to the class.
* Document your learning. Bring a notebook to the class. Otherwise, it may be a good idea to print out the notes and bring them to the class so you can make notes during the class.

**4. Introduction**

* *Persistent* data is the backbone of many applications.
* Three main choices of storing persistent data:
	1. Files
	2. Databases: focus of this course.
	3. Cloud-based storage and database.
* Some advantages of DBMS (according to Ricardo, the optional textbook of this class):
	1. Sharing of data
	2. Control of redundancy
	3. Data consistency
	4. Improved data standards
	5. Better data security
	6. Improved data integrity
	7. Balance of conflicting requirements
	8. Faster development of new applications
	9. Better data accessibility
	10. Economy of scale
	11. More control over concurrency
	12. Better backup and recovery procedures
* How do we *make sense* of these 12 different advantages?
	1. Different textbooks may have different collections of the advantages of DBMS because of different classifications.
	2. No need to memorize them.
	3. Better to assimilate them and construct your own list.
	4. Make your own notes. Use [SEE-I](https://en.everybodywiki.com/SEE-I) (In your own words, state, elaborate, and exemplify with examples, and illustrate the concept.)
		+ However, do not overuse metaphors.
	5. Learning through documentation, communications, and teaching.
* What are some disadvantages of DBMS?
	1. Complexity
	2. Cost
	3. Learning curve
	4. Possible single points of failure and bottleneck

**5. A Simple Introduction to the Relational Model**

* Relational databases are the most popular databases: <https://db-engines.com/en/ranking>. It is based on the relational model.
* There are many other data models.
* In layman's term: A *table* (relation) is the basic unit of a relational database.
* A table is composed of many *rows* (tuples).
* Each row has many *column* (attribute) values.
* A primary key is roughly a *minimal* set of columns in a table that*uniquely identify* a row.
* Two tables can be related to each other by *foreign keys*. A foreign key is roughly a column in a table in which its value must be equal to the referenced value of the primary key in another table (called the paren or referenced table).
* Relational DBMS is the most popular DBMS. Examples:
	+ DB-engine ranking: <https://db-engines.com/en/ranking>
	+ Top 10 DBMS in Data Science: <https://towardsdatascience.com/top-10-databases-to-use-in-2021-d7e6a85402ba>
* SQL is the 'glue' in many DB systems.

***Classroom discussion***

Please ask questions about the toy University DB ([toyu](https://dcm.uhcl.edu/yue/courses/joinDB/Fall2024/notes/toyu/toyu.html))

**3. MySQL**

* The standard query language for RDBMS is Structured Query Language (SQL).
* We use MySQL (or MariaDB) in this class.

DBMS mostly uses a client-server architecture.



XAMPP MariaDB/MhySQL is the DB server.

**3.1 MySQL Server Setup**

We will use MariaDB that is a part of XAMPP. Do not recommend installing standalone MySQL.

[1] Install XAMPP, which contains many server software configured to work together for development purposes. For XAMPP, we will use MySQL/Maria DB and Apache (Web server).

1. Recommended to install XAMPP in the*top* level: c:\xampp (likely the default).



Started MySQL server above (in green).

1. Set up development accounts immediately using *phpMyAdmin* after installation.
2. Change the root password (optional but recommended): a secure step that requires tinkering.
3. To ensure that PHPMyAdmin will work on a new admin account (optional):
	1. Use PHPMyAdmin to create a new admin account, e.g., "frog\_ad", with the password "a\_new\_prince" for both hostname '%' and 'localhost'
	2. PhpMyAdmin uses the default root account (with no initial password) via localhost.
	3. Thus, you will need to supply the new username and password to start up PhpMyAdmin by editing the file c:\xampp\phpMyAdmin\config.inc.php, search change the line to, for example:
		1. $cfg['Servers'][$i]['user'] = 'frog\_ad';
		2. $cfg['Servers'][$i]['password'] = 'a\_new\_prince';

**3.2 MySQL Clients Setup:**

It is common to use multiple clients to connect to a backend database server. In this course, we will use three clients in our classes. You may use your own favorite clients (e.g., MySQL Workbench). However, I may not be as helpful in these clients.



Serverl XAMPP/MariaDB

[1] MySQL Command-Line Prompt: will be used in this class.

Working directory:

1. Come with (1) XAMPP/MariaDB or (2) MySQL 8.x. (Note that the two versions of mysql prompt are somewhat different.)
	1. MariaDB mysql: <https://mariadb.com/kb/en/mysql-command-line-client/>
	2. MySQL 8.x mysql: <https://dev.mysql.com/doc/refman/8.0/en/mysql.html>
2. A command line text-based MySQL-specific client.
3. You may set the PATH variable so you can call mysql prompt anywhere, such as by adding "c:\xampp\mysql\bin" in the PATH system environment variable.

***Example:***

**mysql –h *host* -u *user* -p**

or

**mysql –h *host* -u *user* -p -P port\_number**

[2] PhPMyAdmin

1. A Web-based GUI client focused on DB administration.
2. After starting both MySQL and Apache in XAMPP, go to localhost in your browser.
3. MySQL specific.



Create d5333@%:

CREATE [USER](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/information-functions.html%23function_user) 'd5333'@'%' IDENTIFIED VIA mysql\_native\_password USING '\*\*\*';GRANT ALL PRIVILEGES ON \*.\* TO 'd5333'@'%' REQUIRE NONE WITH GRANT OPTION MAX\_QUERIES\_PER\_HOUR 0 MAX\_CONNECTIONS\_PER\_HOUR 0 MAX\_UPDATES\_PER\_HOUR 0 MAX\_USER\_CONNECTIONS 0;

[3] HeidiSQL: will be used in this class

1. A general Windows GUI SQL client



User: root
password:

[4] MySQL Workbench:

1. A GUI MySQL client that comes with MySQL 8.x (but not XAMPP)

**Creating and populating toyu:**

MySQL creation script for creating toyu: [Createtoyu.sql.txt](https://dcm.uhcl.edu/yue/courses/joinDB/Fall2024/notes/toyu/Createtoyu.sql.txt) (remove .txt while saving). You can execute it with the source command in MySQL command line prompt:

1. Save the script file as createtoyu.sql in your working directory (e.g. …/demo/d5333) for this class.
2. Open a command line terminal in your working directory.
3. Start mysql command prompt: "mysql -u your\_local\_mysql\_account -p"
4. Run "source createtoyu.sql" within mysql command prompt.