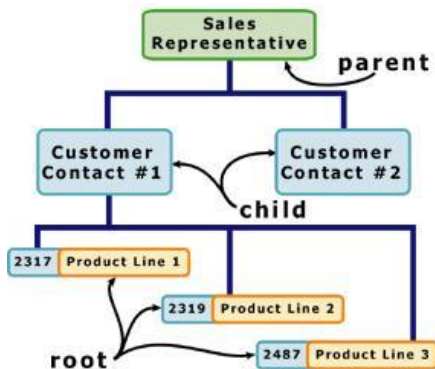


What is Database ?

A *database* is a structured collection of data. Here are some typical examples of databases:

- **An online store database** that stores products, customer details and orders
- **A database for a web forum** that stores members, forums, topics and posts
- **A database for a blog system**, such as WordPress, that stores users, blog posts, categories, tags, and comments

The software that manages databases is known as a **Database Management System**, or **DBMS**. There are different structural types of DBMS: hierarchical, network, relational, and object-oriented.



hierarchical DBMS



network DBMS

Diagram illustrating a relational database structure with three tables: Database 1, Database 2, and Database 3. Red arrows indicate relationships between specific records across the tables.

	First Name	Last Name	Social Security No.
1	John	Smith	010-22-9432
2	John	Smith	003-63-0037
3	John	Smith	000-45-0000
4	Sally	Smith	
5	Steve	Smith	
6			
7			

	Date of Birth	Social Security No.
1	6/12/82	010-22-9432
2	5/9/40	003-63-0037
3		020-45-9326
4		289-56-4321
5		170-54-233
6		
7		

	Address	Social Security No.
1	321 Byberry Road	010-22-9432
2	268 Monroe Avenue	003-63-0037
3	8120 Venshire Drive	020-45-9326
4	207 Congress Drive	289-56-4321
5	1519 Ashbury Lane	170-54-233
6		
7		

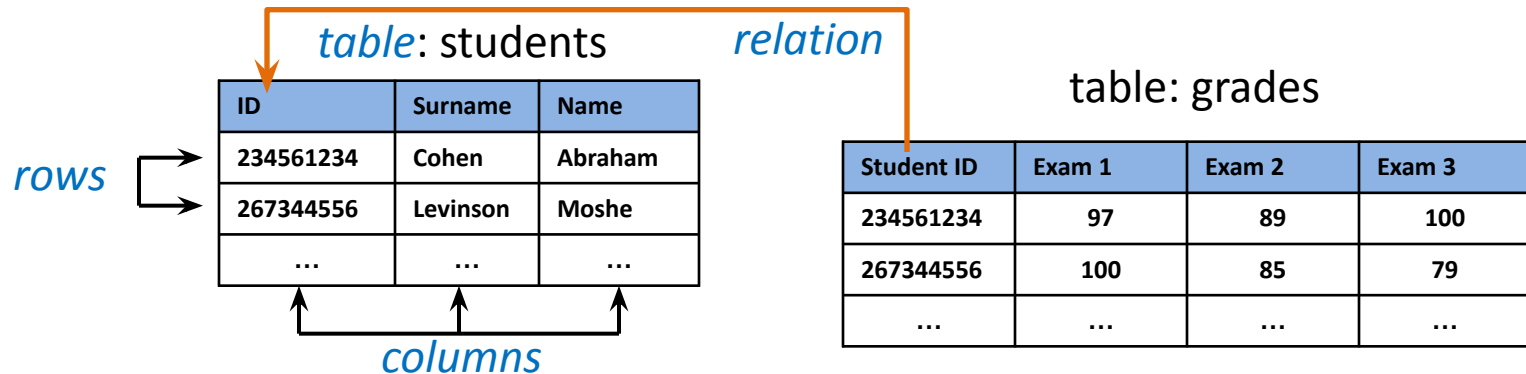
relational DBMS

What is Relational DB and SQL

Relational Databases

A **Relational Database Management System (RDBMS)** organizes data in **tables**.

A **table** has **rows** (or **records**) and **columns**. Tables can be **related** based on common columns to eliminate *data redundancy*.



Popular commercial RDBMS-s: Oracle, IBM DB2, MS SQL Server, MS Access, SAP SyBase and Teradata;
Popular free RDBMS-s: MySQL, Apache Derby (Java DB), mSQL (mini SQL), SQLite, PostgreSQL and Apache OpenOffice's Base.

Structure Query Language (SQL)

A high-level language designed for structuring relational databases and for creating, updating, reading and deleting (CURD) records. SQL defines a set of commands, such as SELECT, INSERT, UPDATE, DELETE, CREATE TABLE, DROP TABLE, and etc.

Edgar F. Codd of IBM proposed the Relational Database Model in 1970. SQL was developed by Donald D. Chamberlin and Raymond F. Boyce at IBM in the early 1970. ANSI (American National Standard Institute) established the first SQL standard in 1986. Most of the database vendors have their own dialects, e.g., PL/SQL (Oracle), Transact-SQL (Microsoft, SAP), PL/pgSQL (PostgreSQL).

MySQL

MySQL is the most used, and possibly the best *industrial-strength, open-source* and *free* Relational Database Management System (RDBMS). MySQL was developed by Michael "Monty" Widenius and David Axmark in 1995. It was owned by a Swedish company called MySQL AB, which was bought over by Sun Microsystems in 2008. Sun Microsystems was acquired by Oracle in 2010.

MySQL is successful, not only because it is free and open-source, but also for its speed, ease of use, reliability, performance, connectivity, portability, security, small size, and rich features. MySQL supports all features expected in a high-performance relational database, such as transactions, foreign key, replication, subqueries, stored procedures, views and triggers.

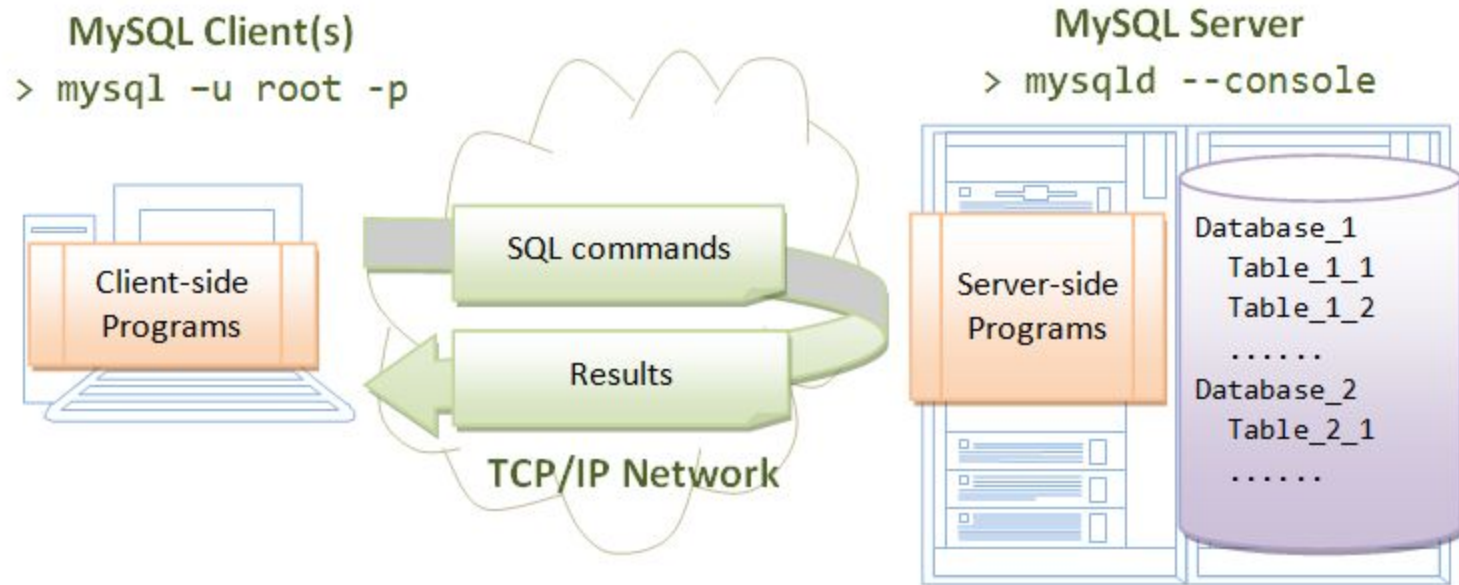
MySQL is often deployed in a AMP (Apache-MySQL-PHP) environment for Linux (LAMP), Windows (WAMP) and Mac (MAMP).

The mother site for MySQL is www.mysql.com.

The reference for MySQL is the "MySQL Reference Manual" at <http://dev.mysql.com/doc/> .

The MySQL tutorial is located at <http://www.mysqltutorial.org/> .

MySQL Server and Client



MySQL RDBMS operates as a client-server system over TCP/IP network. The server runs on a machine with an IP address, on a chosen TCP port number. The default TCP port number for MySQL is 3306.

Users access the server via a client program, connecting to the server at the given IP address and TCP port number.

A MySQL database server contains one or more **databases** (aka **schemas**). A database contains one or more **tables**. A table consists of **rows** (records) and **columns** (fields). The *server* program is called **mysqld** (suffix 'd' means “daemon” - non-interactive process running in the background). The *client* program is called **mysql**.

These programs are kept in the **bin** sub-directory of the MySQL installed directory.

MySQL Basic Statements

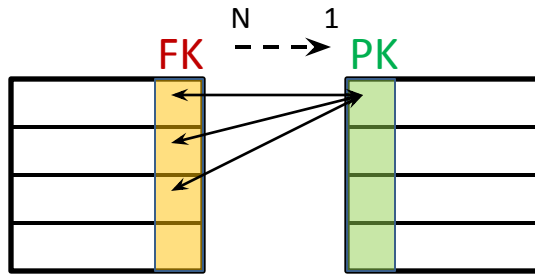
Statements Type	operation information	adding...	modifying...	removing...	displaying...
Data Definition & Administration Statements	data structure	CREATE	ALTER	DROP	SHOW DESCRIBE
Data Manipulation Statements	data content	INSERT LOAD DATA	UPDATE	DELETE	SELECT
Utility Statements	USE HELP EXPLAIN				

Data Loading / Unloading. Database Backup / Restore

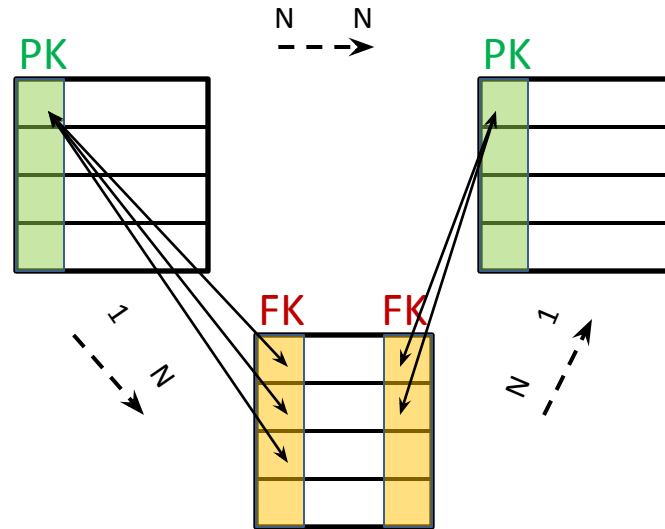
		Data for 1 table as delimiter-separated file (csv, tsv, e.t.c)	All DB as SQL script
From mysql window	Save	SELECT ... INTO OUTFILE	
	Restore	LOAD DATA LOCAL INFILE ...	SOURCE ... <i>file.sql</i>
From cmd window	Save		mysqldump ... > <i>file.sql</i>
	Restore	mysqlimport ... <i>file.tsv</i>	mysql ... < <i>file.sql</i>

Relations And Keys

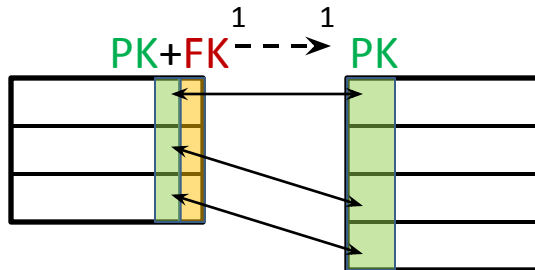
Many-to-One



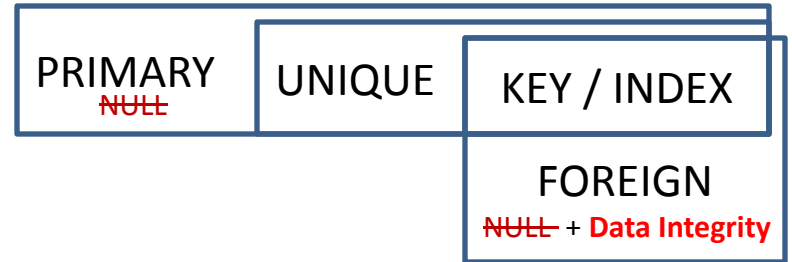
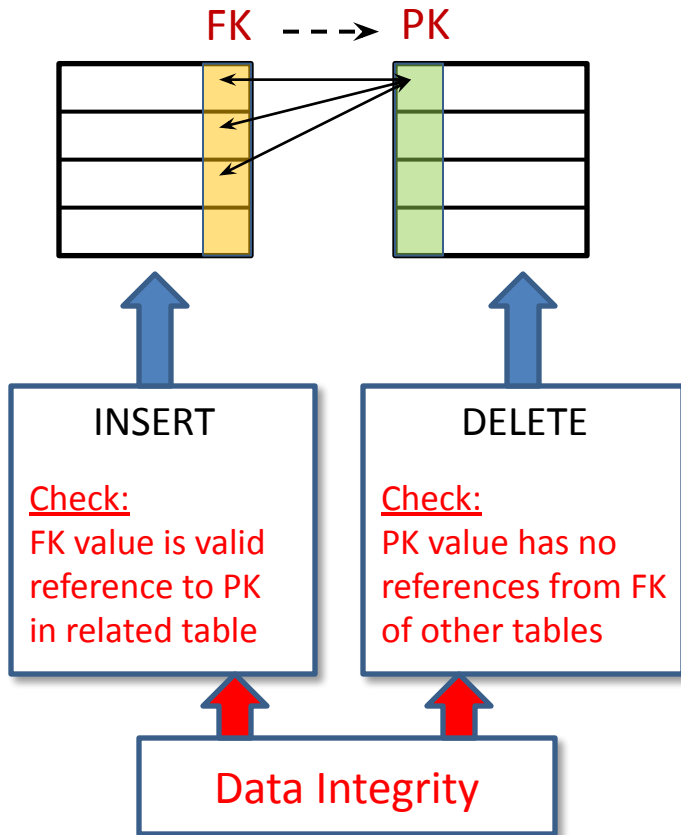
Many-to-Many



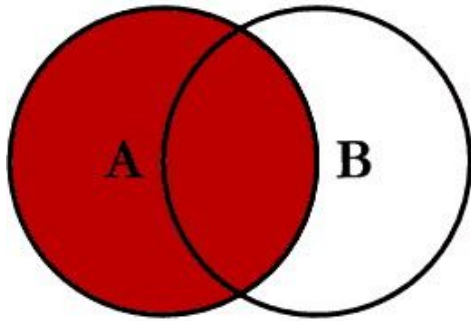
One-to-One



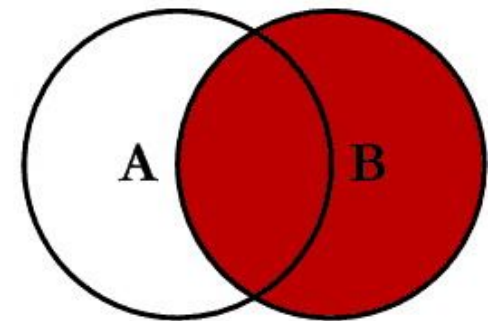
Keys And Data Integrity



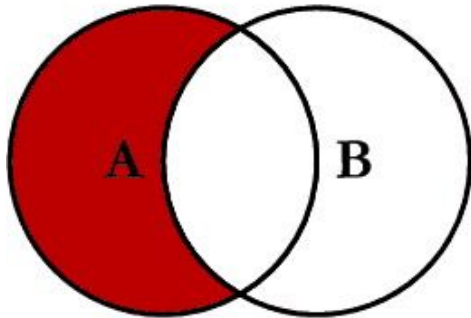
SQL JOINS



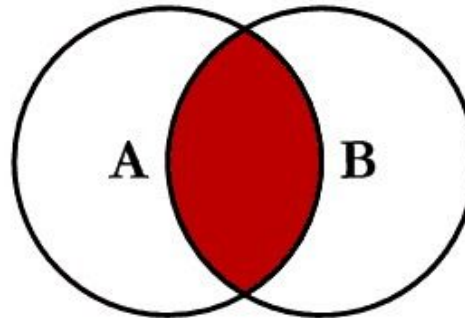
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



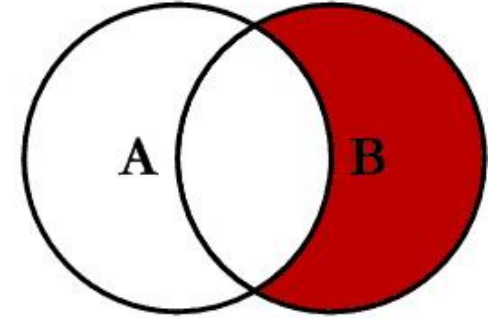
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



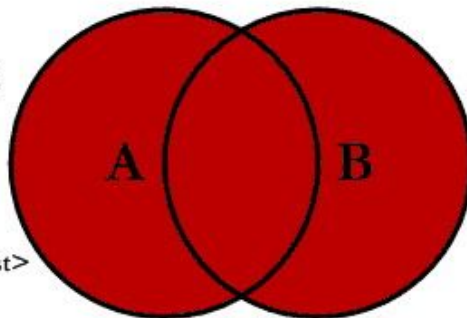
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



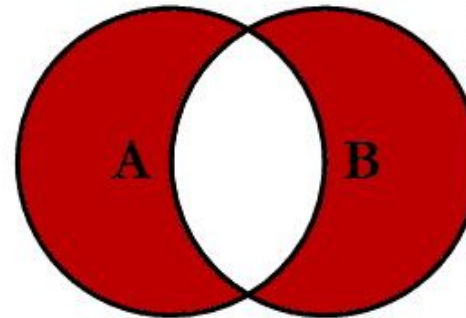
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```

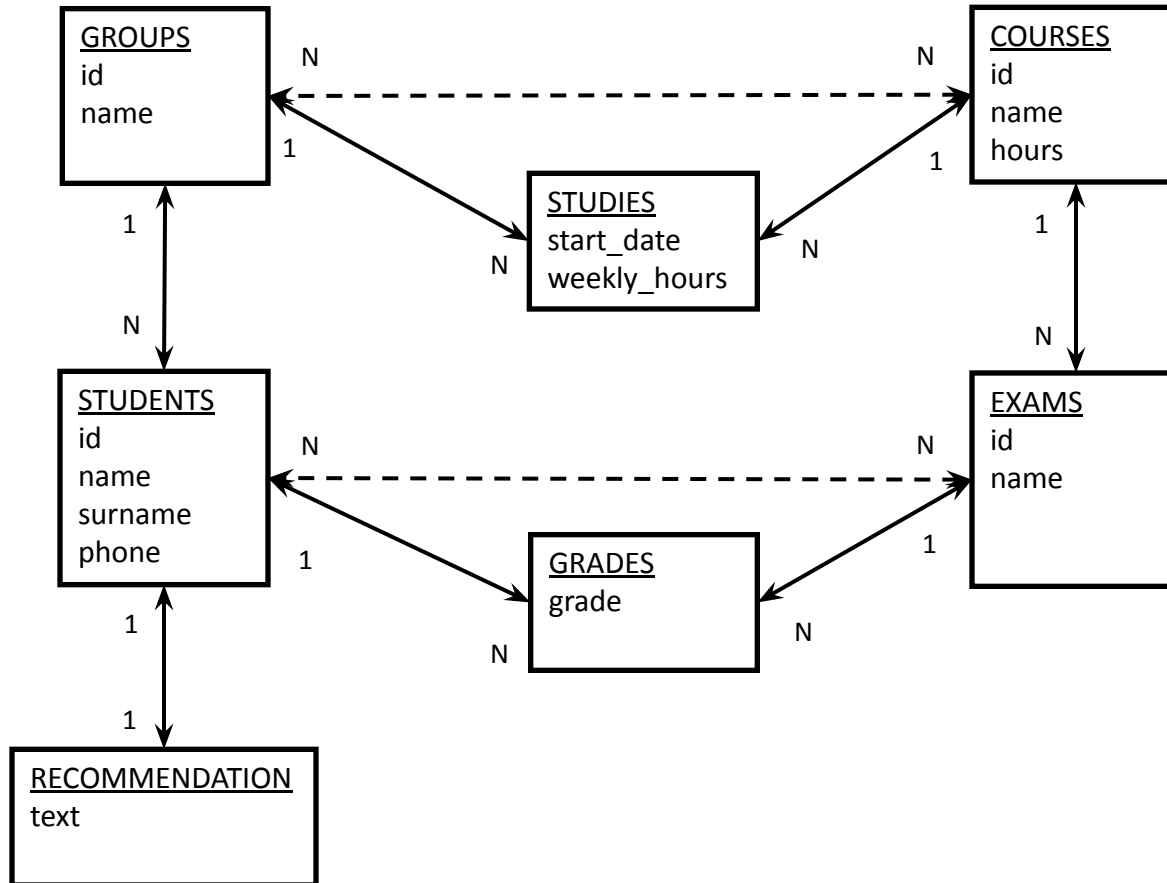


```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

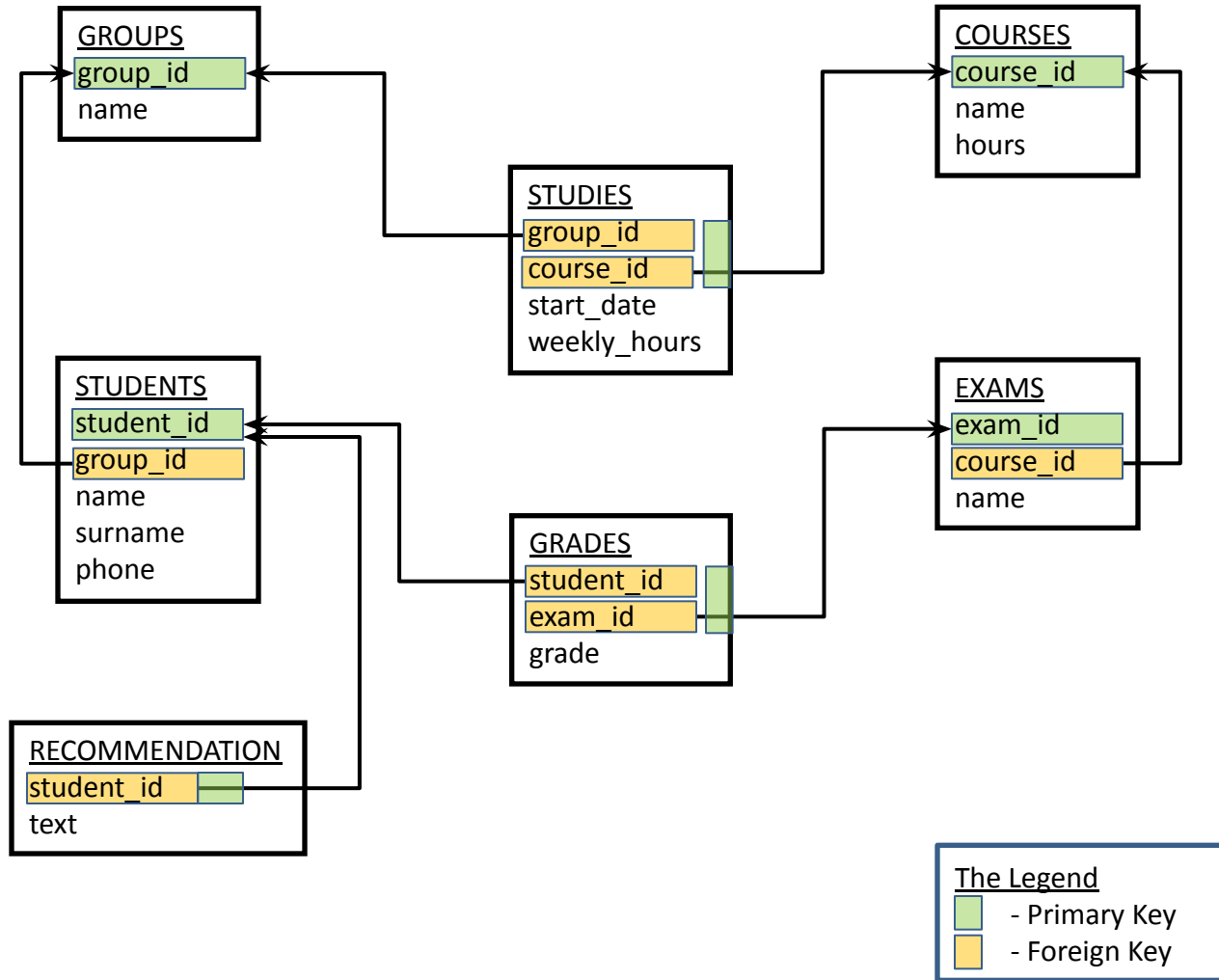
← not supported by MySQL →

© C.L. Moffatt, 2008

Example: “College”. Database Design



Example: “College”. Database Schema



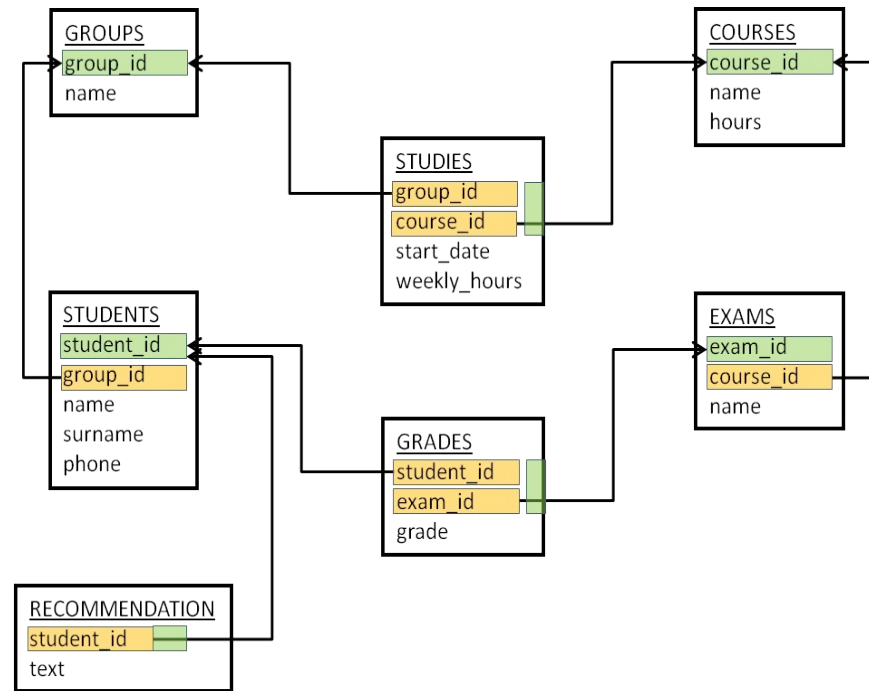
Example: “College”. Data Population

Data Population

To ensure data integrity during population of the database, tables that have **relation (Foreign Key)** would be populated after the tables on which they depend.

Below is example of data population sequence, preserving data integrity:

- Add **Course**
- Add Course-related **Exams**
- Add **Group**
- Add Group-related **Studies**
- Add **Students** belonging to Group
- Add Student's **Grade** for Exam
- Add Student's **Recommendation**



Example: “College”. Data Modification and Deletion

Data Modification

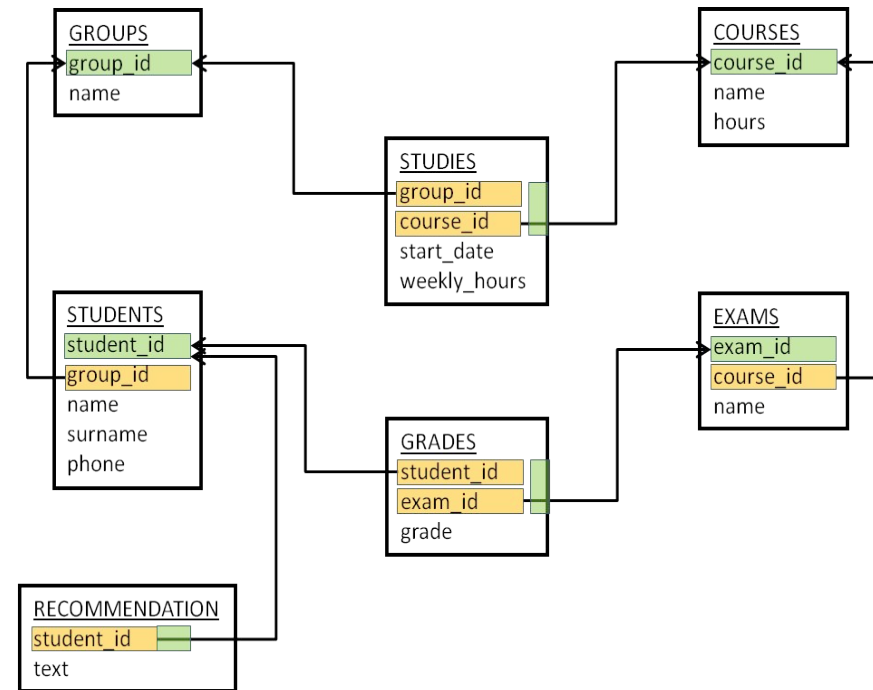
Any field, **excepting Primary Keys**, in any table could be modified without violation of data integrity.

Data Deletion

To preserve data integrity, any record from any table could be deleted:

- if it doesn't have **related** (by **Foreign Key**) records in other tables,
- or if it is deleted together with **related** records (Cascade Deletion).

Table	Deletion Dependencies
Courses	course <input type="checkbox"/> studies <input type="checkbox"/> exams <input type="checkbox"/> grades
Exams	exam <input type="checkbox"/> grades
Group	group <input type="checkbox"/> studies <input type="checkbox"/> students <input type="checkbox"/> grades <input type="checkbox"/> recommendations
Studies	study
Students	student <input type="checkbox"/> grades <input type="checkbox"/> recommendation
Grades	grade
Recommendation	recommendation



Example: “College”. Output Forms

Below is sample list of Output Forms which could be built using this database schema:

Per Group

- List of Students
- Learning Plan
- Tuition Fee Estimation
- Specific Exam Results
- Average Course Results
- Rate Students due to accepted grades

Per Student

- Personal Information
- Learning Plan
- Tuition Fee Estimation
- Exam Results: Specific Exam, Per Course, Overall Results
- Diploma (courses, hours, grades, dates, recommendation)

Per Course

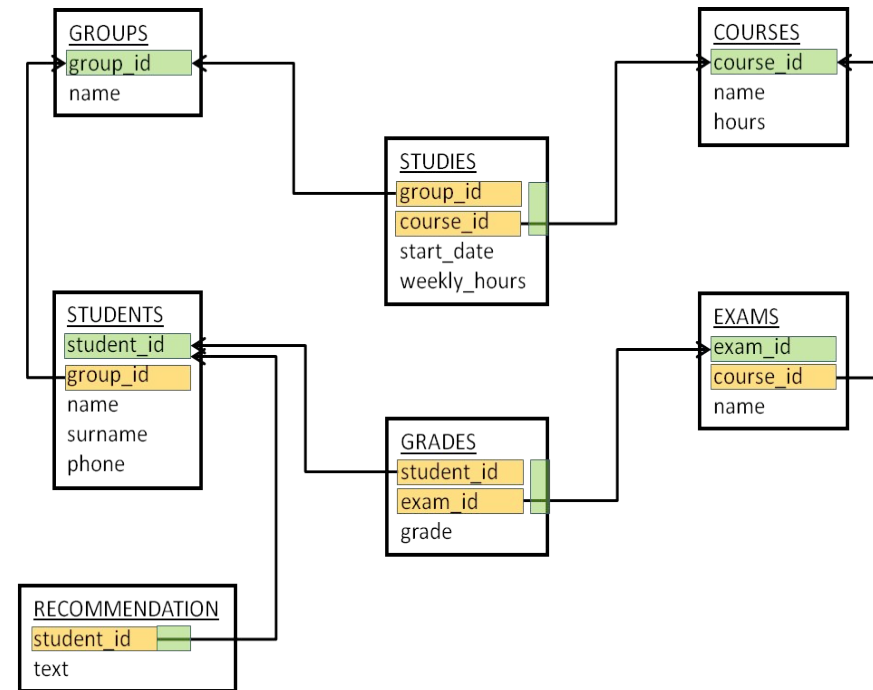
- List of Exams
- Most “Easy” and “Difficult” Exam due to accepted grades
- Rate Students due to accepted grades
- Rate Groups due to accepted grades

Per Exam

- List results (per student, per group)
- Rate Students due to accepted grades
- Rate Groups due to accepted grades

Statistics

- Most “Easy” and “Difficult” Course due to accepted grades
- Rate Students due to accepted grades
- Rate Groups due to accepted grades



MySQL Entities :

Persistent (Server side) and Temporary (Session runtime)

Persistent
Databases
Tables, Views
KEY/INDEX) Constraints (PK,FK,
Procedures & Functions
Triggers

Temporary
Variables
Temporary Tables
Prepared Statements
Transactions
Locks