1. 数据库基础（重点）

1.1 什么是数据库

存储数据用文件就可以了，为什么还要弄个数据库?

文件保存数据有以下几个缺点:

- 文件的安全性问题
- 文件不利于数据查询和管理
- 文件不利于存储海量数据
- 文件在程序中控制不方便

数据库存储介质:

- 磁盘
- 内存

为了解决上述问题，专家们设计出更加利于管理数据的东西——数据库，它能更有效的管理数据。**数据库的水平是衡量一个程序员水平的重要指标。**

1.2 主流数据库

- SQL Sever： 微软的产品，.Net程序员的最爱，中大型项目。
- Oracle： 甲骨文产品，适合大型项目，复杂的业务逻辑，并发一般来说不如MySQL。
- MySQL：世界上最受欢迎的数据库，属于甲骨文，并发性好，不适合做复杂的业务。主要用在电商，SNS，论坛。对简单的SQL处理效果好。
- PostgreSQL :加州大学伯克利分校计算机系开发的关系型数据库，不管是私用，商用，还是学术研究使用，可以免费使用，修改和分发。
- SQLite： 是一款轻型的数据库，是遵守ACID的关系型数据库管理系统，它包含在一个相对小的C库中。它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百K的内存就够了。
- H2： 是一个用Java开发的嵌入式数据库，它本身只是一个类库，可以直接嵌入到应用项目中。

1.3 基本使用

1.3.1 MySQL安装

- Centos 6.5下编译安装MySQL 5.6.14
- CentOS 7 通过 yum 安装 MariaDB
- Windows下安装MySQL5.7

1.3.2 连接服务器

输入:

```
mysql -h 127.0.0.1 -P 3306 -u root -p
```

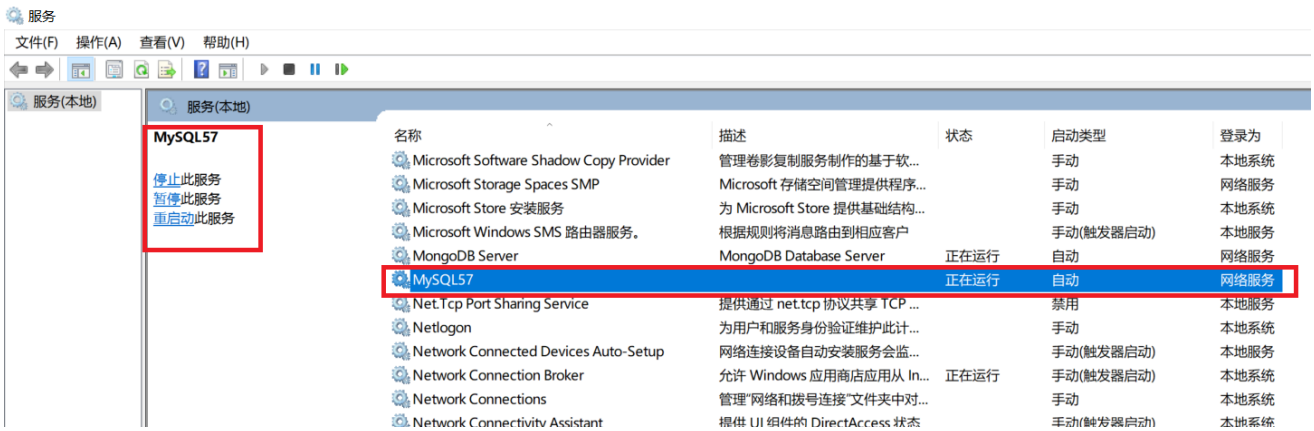
输出:

```
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.21-log MySQL Community Server (GPL)


Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

> 注意：
>
> 如果没有写`-h 127.0.0.1`默认是连接本地
>
> 如果没有写`-P 3306`默认是连接3306端口号

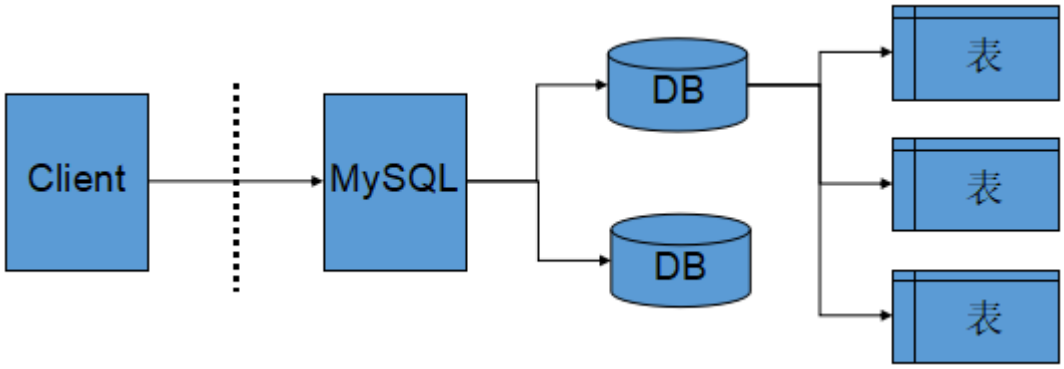
1.3.3 服务器管理

- 执行`win+r`输入`services.msc`打开服务管理器
- 通过下图左侧停止，暂停，重启动按钮进行服务管理



1.3.4 服务器，数据库，表关系

- 所谓安装数据库服务器，只是在机器上安装了一个数据库管理系统程序，这个管理程序可以管理多个数据库，一般开发人员会针对每一个应用创建一个数据库。
- 为保存应用中实体的数据，一般会在数据库中创建多个表，以保存程序中实体的数据。
- 数据库服务器、数据库和表的关系如下：



1.3.5 使用案例

- 创建数据库

```
create  database helloworld;
```

- 使用数据库

```
use helloworld;
```

- 创建数据库表

```
create table student(
    id int,
    name varchar(32),
    gender varchar(2)
);
```

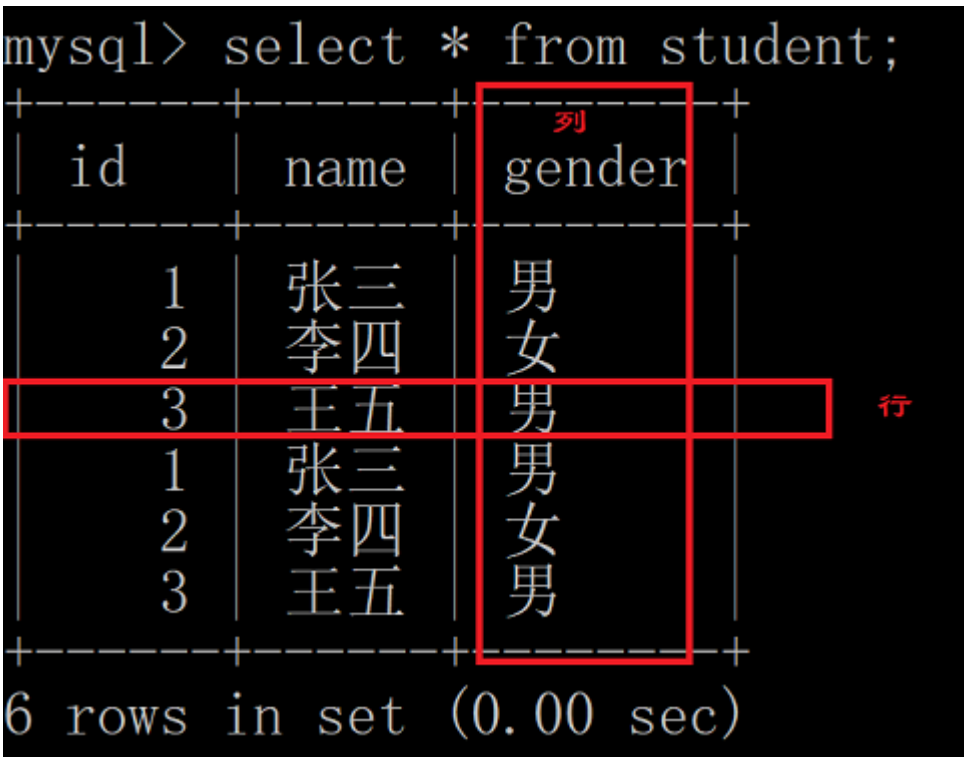
- 表中插入数据

```
insert into student (id, name, gender) values (1, '张三', '男');
insert into student (id, name, gender) values (2, '李四', '女');
insert into student (id, name, gender) values (3, '王五', '男');
```

- 查询表中的数据

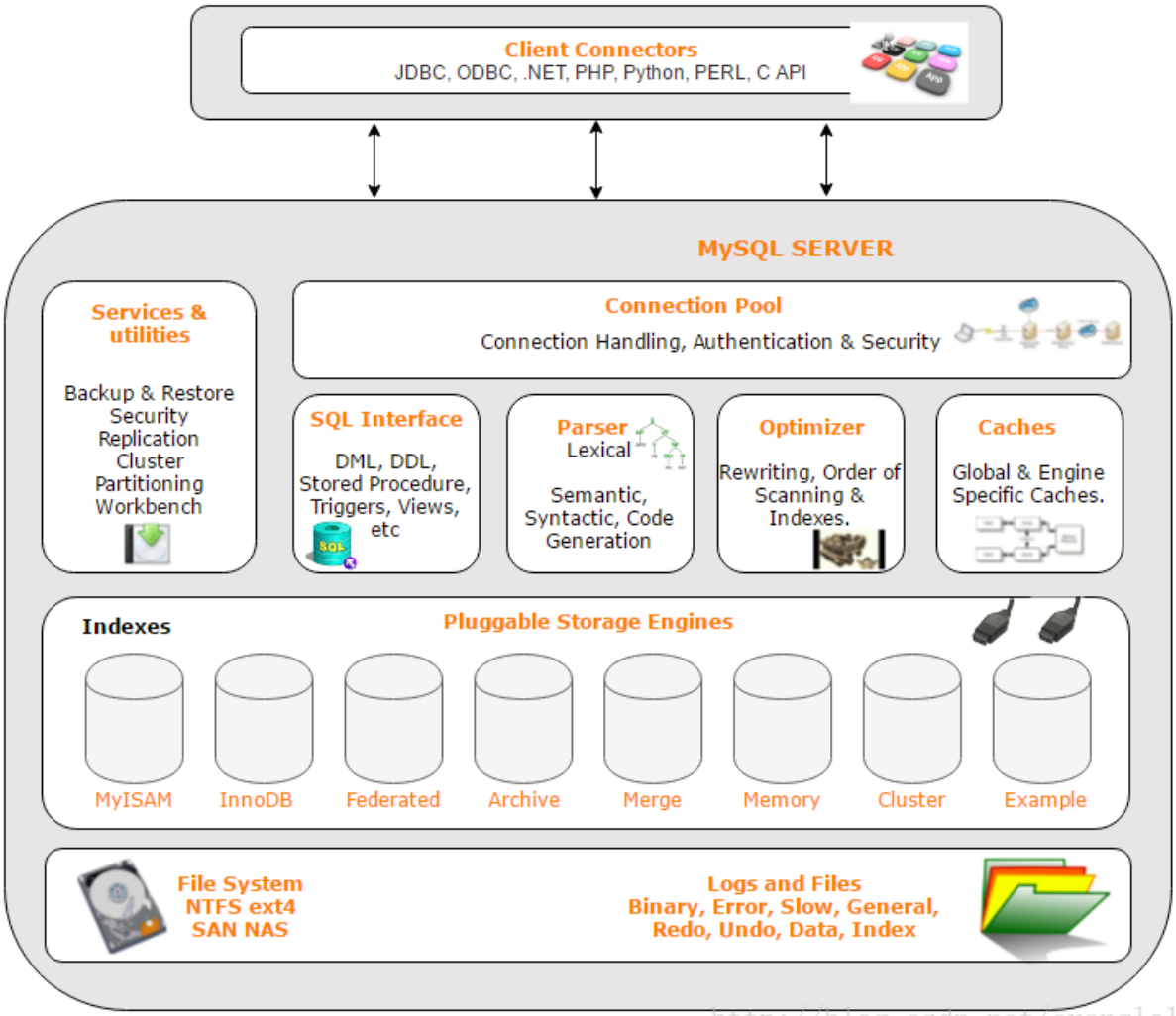
```
select * from student;
```

1.3.6 数据逻辑存储



1.4 MySQL架构

MySQL 是一个可移植的数据库，几乎能在当前所有的操作系统上运行，如 Unix/Linux、Windows、Mac 和 Solaris。各种系统在底层实现方面各有不同，但是 MySQL 基本上能保证在各个平台上的物理体系结构的一致性。



1.5 SQL分类

- DDL数据定义语言，用来维护存储数据的**结构**

  代表指令：`create, drop, alter`

- DML数据操纵语言，用来对**数据**进行操作

  代表指令：`insert, delete, update`

  - DML中又单独分了一个DQL，数据查询语言，代表指令：`select`
- DCL数据控制语言，主要负责权限管理和事务

  代表指令：`grant, revoke, commit`

1.6 存储引擎

1.6.1 存储引擎

存储引擎是：数据库管理系统如何存储数据、如何为存储的数据建立索引和如何更新、查询数据等技术的实现方法。

MySQL的核心就是插件式存储引擎，支持多种存储引擎。

1.6.2 查看存储引擎

```
show engines;
```

```
MySQL Storage Engine

+--------------------+---------+----------------------------------------------------------------+--------------+------+------------+
| Engine             | Support | Comment                                                        | Transactions | XA   | Savepoints |
+--------------------+---------+----------------------------------------------------------------+--------------+------+------------+
| InnoDB             | DEFAULT | Supports transactions, row-level locking, and foreign keys     | YES          | YES  | YES        |
| MRG_MYISAM         | YES     | Collection of identical MyISAM tables                          | NO           | NO   | NO         |
| MEMORY             | YES     | Hash based, stored in memory, useful for temporary tables      | NO           | NO   | NO         |
| BLACKHOLE          | YES     | /dev/null storage engine (anything you write to it disappears) | NO           | NO   | NO         |
| MyISAM             | YES     | MyISAM storage engine                                          | NO           | NO   | NO         |
| CSV                | YES     | CSV storage engine                                             | NO           | NO   | NO         |
| ARCHIVE            | YES     | Archive storage engine                                         | NO           | NO   | NO         |
| PERFORMANCE_SCHEMA | YES     | Performance Schema                                             | NO           | NO   | NO         |
| FEDERATED          | NO      | Federated MySQL storage engine                                 | NULL         | NULL | NULL       |
+--------------------+---------+----------------------------------------------------------------+--------------+------+------------+
```

1.6.3 存储引擎对比

Feature	MyISAM	BDB	Memory	InnoDB	Archive	NDB
Storage Limits	No	No	Yes	64TB	No	Yes
Transactions (commit, rollback, etc.)		✔		✔		
Locking granularity	Table	Page	Table	Row	Row	Row
MVCC/Snapshot Read				✔	✔	✔
Geospatial support	✔					
B-Tree indexes	✔	✔	✔	✔		✔
Hash indexes			✔	✔		✔
Full text search index	✔					
Clustered index				✔		
Data Caches			✔	✔		✔
Index Caches	✔		✔	✔		✔
Compressed data	✔				✔	
Encrypted data (via function)	✔	✔	✔	✔	✔	✔
Storage cost (space used)	Low	Low	N/A	High	Very Low	Low
Memory cost	Low	Low	Medium	High	Low	High
Bulk Insert Speed	High	High	High	Low	Very High	High
Cluster database support						✔
Replication support	✔	✔	✔	✔	✔	✔
Foreign key support				✔		
Backup/Point-in-time recovery	✔	✔	✔	✔	✔	✔
Query cache support	✔	✔	✔	✔	✔	✔
Update Statistics for Data Dictionary	✔	✔	✔	✔	✔	✔