MySQL （一）

```
#今日必备单词：
# ------插入------
insert
/ɪn'sɜ:rt/

# ------查询-------
select
/sɪ'lekt/

# ------修改------
update
/ˌʌp'deɪt/

# ------删除-------
delete
/dɪ'li:t/
```

简介

一、 为什么要学习数据库?

程序的组成：前端【负责展示数据】

后端【提供和处理数据】-保存数据【数据库当中】

所有程序的核心都是数据，而数据存储于数据库当中，因此我们要学数据库。

二、列举常见的 关系型数据库 和 非关系型 都有那些?

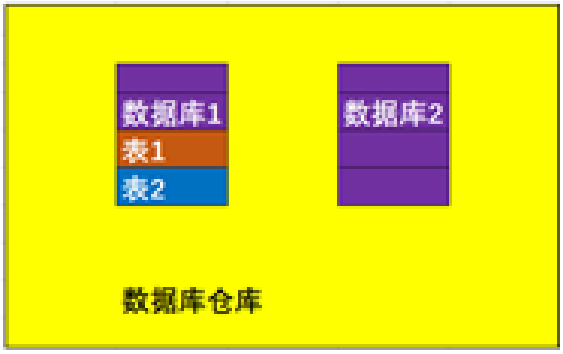
```
关系型数据库【所有数据以表格的形式进行存储，并且表格和表格之间会有关联】：
Oracle、DB2、Microsoft SQL Server、Microsoft Access、MySQL

非关系型数据库【指的是以非表格形式进行存储的数据库，比如以键值对存储的redis,比如说jison形式存储的，以列表形式存储的。。。。】：
NMongoDb、redis、HBase
```

三、关系型数据库结构

```
mysql---库---表---数据

表：存储具体的数据
数据库 ： 数据仓库 ，这个数据仓库中有很多的表格
数据库管理工具 ： mysql就是这样一个工具，使用sql语言去对数据库进行管理。
```



MySQL账户操作

`mysql` 超级管理员账号: `root` 密码: `root`

```
# 如何进入mysql
mysql -uroot -proot
# -u 是登录用户账号的意思，后面接账号
#  -p  是登录用户密码

# 退出
\q    exit;

# 查看数据库
show databases;

#注意！！！！python交互的时候  因为root 用户限制了远程登录，  所以需要创建新的有远程登录的账号进行使用

# 1、创建账户
create user '用户名'@'%' identified by '密码';
#   create 创建
#   user 用户
#   identified 设置密码
% --- 通配符 指 用户可以从任何地方进行登录


# 2、给权限：
grant all on *.* to '用户名'@'%';
all --- 所有的增删改查
*.* ---- 所有的库.所有的表

# 3、刷新：
flush privileges;

# 查看当前账号
select user();

# 删除用户
drop user '用户名'@'%';
```

MySQL库操作

```
show databases;  # 查看有哪些数据库
# mysql中有四个默认存在的数据库，不要随便动他们。

create database 数据库名;  # 创建数据库

# show create database 数据库名;       # 查看数据库结构（指令）

drop database 数据库名;   # 删库

use 数据库名; # 进入数据库（使用该数据库，对库中的表格进行增删改查操作）

select database(); # 查看当前在哪个数据库
```

MySQL表操作

```
# 创建表格
# 语法:
create table 表名(字段1 数据类型, 字段2 数据类型...);
# 实例:创建 名为hk的表格   表格包含 id  name  class 三个字段
create table hk(student_ID int(11),student_Name varchar(20),student_Class
varchar(20));
#INT(11)表示允许的整数范围为-2147483648到2147483647。
#VARCHAR(20)表示该字段最多可以存储20个字符。

show tables;  # 查看当前库所有的表

desc 表名;   # 查看表结构

# show create table 表名;    # 查看表结构

# 删除表（删完后可以再次查询检查一下）
drop table 表名;
```

`MySQL`是一种流行的关系型数据库管理系统（`RDBMS`），它支持多种数据类型，用于存储各种类型的数据。以下是`MySQL`中常用的数据类型：

数值类型：

`int`：整数类型，常用于存储整数值。
`float`：单精度浮点数类型，用于存储近似值。
`double`：双精度浮点数类型，用于存储较高精度的近似值。
`decimal`：(需要指定大小) 高精度十进制数类型，用于存储精确的小数值。

字符串类型(需要指定大小)：

`char`：定长字符串类型，适用于存储固定长度的字符串。
`varchar`：变长字符串类型，适用于存储可变长度的字符串。
`text`：较长的文本字符串类型，用于存储大段文本数据。

日期与时间类型：

`date`：日期类型，存储日期，格式为`YYYY-MM-DD`。
`time`：时间类型，存储时间，格式为`HH:MM:SS`。
`datetime`：日期时间类型，存储日期和时间，格式为`YYYY-MM-DD HH:MM:SS`。
`timestamp`：类似于`DATETIME`，但是会自动记录数据插入或修改的时间。

布尔类型：

`bool` / `boolean`：布尔类型，用于存储True或False。

枚举类型（`ENUM`）：

`ENUM`：枚举类型，用于存储预定义的字符串值列表。

二进制类型(需要指定大小)：

`blob`：用于存储二进制数据，例如图片、音频等。
`binary` / `varbinary`：用于存储二进制数据，可以指定长度。

其他类型：

`json` 用于存储`JSON`格式的数据。
`set`：集合类型，用于存储预定义的字符串值集合。
不同的数据类型在存储空间、精度和适用场景等方面有所区别，选择合适的数据类型可以优化数据库性能和节省存储空间。

MySQL表数据操作

```
# ------插入------
# 语法:
insert into 表名(字段1, 字段2…) values(值1, 值2...)
insert 表名 set 字段1 = 值1, 字段2 = 值2, ...
# 实例:
insert into hk(student_ID,student_Name, student_Class)
values(1,'buyu','classA');
insert hk set student_ID=2,student_Name='tongyao',student_Class='classB';

# ------查询-------
# 语法:
select 字段 from 表名;
select 字段 from 表名 where 查询条件;
# 实例:
select * from hk;   # * 代表所有的字段
select student_Name from hk;
select * from hk where student_ID=1;

# ------修改------
# 语法:
update 表格名称 set 字段名=值 where 条件;
# 实例:
update hk set student_Name='linli' where student_ID=2;
update hk set student_Name='hualong';# 如果不指定条件就会全部进行修改

# ------删除-------
# 语法:
delete from 表名 where 条件;
# 实例:
```

```
delete from hk where student_ID=2;    # 如果不指定条件就会全部删除
```

mysql表的修改(拓展)

```
# 改表名
alter table 旧表名 rename to 新表名;
# 改字段
alter table 表名 change 旧字段名 新字段名 int(11);
# 改字段类型
alter table 表名 modify 字段名 varchar(20);
# 新增一个字段？？？ （去查资料拓展一下）

/*数据库注释*/
```

作业：

1、登录数据库 用root

2、创建属于自己的账户名，记得分权限和刷新，!!! 记得给权限和刷新

3、登录新账号

4、创建新库，名字自己取

5、进入库并抄写 sql 指令，每抄一条指令就执行一次，一定要执行成功再进入下一条，报错就解决，解决优先翻译，然后百度。

提交作业：

请提交students，subjects，grades 这三张表格的查询结果，截图。

以下为作业sql指令：

```
# 创建一张学生表.
CREATE TABLE students (
  number CHAR(9),
  name VARCHAR(20),
  age INT,
  birth DATE
);
INSERT INTO students VALUES ('201804001', 'Qiye', 16, '2002-01-01');
INSERT INTO students VALUES ('201804002', 'JackLee', 17, '2001-01-02');
INSERT INTO students VALUES ('201804003', 'Julia', 18, '2000-01-03');
INSERT INTO students VALUES ('201804004', 'Stefer', 19, '2001-01-04');
INSERT INTO students VALUES ('201804005', 'Steven', 20, '2000-01-05');
INSERT INTO students VALUES ('201804006', 'Mark', 21, '1999-01-06');
INSERT INTO students VALUES ('201804007', 'Stark', 22, '1999-01-07');
INSERT INTO students VALUES ('201804008', 'Tonny', 23, '1999-01-08');
```

```
INSERT INTO students VALUES ('201804009', 'Jarvis', 24, '1999-01-09');
INSERT INTO students VALUES ('201804010', 'ZhangSan', 25, '1999-01-10');
INSERT INTO students VALUES ('201804011', 'lisi', 23, '1990-04-10');
INSERT INTO students VALUES ('201804012', 'wanger', 20, '1991-01-10');
INSERT INTO students VALUES ('201804013', 'mazi', 21, '1993-01-13');
INSERT INTO students VALUES ('201804014', 'xiaoxing', 80, '1995-01-22');
INSERT INTO students VALUES ('201804015', 'hundan', 36, '1998-05-10');
INSERT INTO students VALUES ('201804016', 'xiaowang', 30, '1999-07-10');
INSERT INTO students VALUES ('201804017', 'laowang', 18, '1997-08-10');

SELECT * FROM students;


# 创建一张学科表.
CREATE TABLE subjects (
  number CHAR(4),
  title VARCHAR(30),
  duration INT
);
INSERT INTO subjects VALUES ('0001', 'python', 32);
INSERT INTO subjects VALUES ('0002', 'java', 16);
INSERT INTO subjects VALUES ('0003', 'web', 16);
INSERT INTO subjects VALUES ('0004', 'go', 32);
INSERT INTO subjects VALUES ('0005', 'c++', 32);

SELECT * FROM subjects;


# 创建一张成绩表.
CREATE TABLE grades (
  student_number CHAR(9),
  subject_number CHAR(4),
  grade INT
);
INSERT INTO grades VALUES ('201804001', '0001', 90);
INSERT INTO grades VALUES ('201804002', '0001', 89);
INSERT INTO grades VALUES ('201804003', '0001', 88);
INSERT INTO grades VALUES ('201804004', '0001', 87);
INSERT INTO grades VALUES ('201804005', '0001', 86);
INSERT INTO grades VALUES ('201804006', '0001', 85);
INSERT INTO grades VALUES ('201804007', '0001', 84);
INSERT INTO grades VALUES ('201804008', '0001', 83);
INSERT INTO grades VALUES ('201804009', '0001', 82);
INSERT INTO grades VALUES ('201804017', '0001', 91);
INSERT INTO grades VALUES ('201804016', '0001', 91);
INSERT INTO grades VALUES ('201804011', '0001', 91);
INSERT INTO grades VALUES ('201804010', '0001', 81);
INSERT INTO grades VALUES ('201804001', '0002', 80);
INSERT INTO grades VALUES ('201804002', '0002', 79);
INSERT INTO grades VALUES ('201804003', '0002', 78);
INSERT INTO grades VALUES ('201804004', '0002', 77);
INSERT INTO grades VALUES ('201804005', '0002', 76);
INSERT INTO grades VALUES ('201804006', '0002', 75);
INSERT INTO grades VALUES ('201804007', '0002', 74);
INSERT INTO grades VALUES ('201804008', '0002', 73);
```

```sql
INSERT INTO grades VALUES ('201804009', '0002', 72);
INSERT INTO grades VALUES ('201804010', '0002', 71);
INSERT INTO grades VALUES ('201804012', '0001', 91);
INSERT INTO grades VALUES ('201804013', '0001', 91);
INSERT INTO grades VALUES ('201804014', '0001', 91);
INSERT INTO grades VALUES ('201804015', '0001', 91);
INSERT INTO grades VALUES ('201804016', '0001', 91);
SELECT * FROM grades;
```