

1.登录数据库:	<pre>sudo mysql -u root -p 回车输入密码: lj19906788 sudo mysql -u admin -p 回车输入密码: lm19936688</pre>
2.查看数据库	<pre>show databases;</pre>
3.选择数据库	<pre>use mysql; use env_monitor;</pre>
4.查看数据库的使用者	
5.env_monitor远程访问权限	<pre>update env_monitor set host='%' where user='root';</pre>
6.创建数据库	<pre>CREATE DATABASE dbname; create database device_info; create database users;</pre>
7.授权localhost主机上admin管理env_monitro数据库的所有权限。	<pre>grant all on env_monitor.* to admin@localhost;</pre>
8.修改登录密码	<pre>CREATE USER 'root'@'%' IDENTIFIED BY 'root'; ALTER USER 'root'@'%' IDENTIFIED WITH mysql_native_password BY 'root的密码'; alter user 'root'@'%' identified with mysql native password by '123456';</pre>
9.修改权限	<pre>GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' WITH GRANT OPTION; all表示所有权限</pre>
10.刷新权限	<pre>FLUSH PRIVILEGES;</pre>
11.设置root管理员在数据库中的专有密码:	<pre>sudo mysql_secure_installation</pre>
12.设置端口号:	<pre>sudo vim /etc/mysql/mysql.conf.d/mysqld.cnf 在文件中添加port参数, port=13306样式来进行修改端口</pre>
13.通过本地python远程链接数据库	<pre>pip3 install python-mysql</pre>
14.重新启动mysql:	<pre>sudo service mysql restart</pre>

-----	-----
15.查看数据库	show databases;
16.使用数据库：	use 数据库名;
17.查看当前使用的数据库	select database();
18.创建数据库	create database 数据库名 charset=utf8;
19.删除数据库	drop database 数据库名;
-----	-----
20.查看当前数据库中所有的表	show tables;
21.查看表结构	desc 表名;
22.创建表	<pre>create table table_name(id int unsigned not null auto_increment primary key, name varchar(30) not null column1 datatype contrai, column2 datatype, column3 datatype, columnN datatype, PRIMARY KEY(one or more columns));</pre> <p>相关的关键字说明 create table table_name();</p> <p>auto_increment 表示自动增长 not null 表示不能为空 primary key 表示主键,主键只有有一个 default 表示默认 create table 数据表名 (字段 类型 约束) 最后一个字段后面不能加逗号</p>
23.在表中插入数据 23.在表中添加数据 往下看第33条、第34条	<pre>insert into 表名 values (id号, “字段的值”) insert into classes values (0, “python1 班”) insert into device_info values (SN号, type) insert into users (name, created_at) VALUES ('John Doe', NOW());</pre>

	<p>注意填入数据时，严格按照每种数据的格式进行填写</p> <p>varchar 类型的数据使用单引号</p> <p>int 类型的数据使用整型数字，不加单引号</p> <p>插入时间使用now()实现</p>
24.查看表中的所有数据：	<pre>select * from 表名; selec * from device_info;</pre>
25.修改表-添加字段	<pre>alter table 表名 add 列名 类型;</pre> <p>例：</p> <pre>alter table students add birthday datetime; alter table device_info add temprature int;</pre> <p>添加字段是指表中的一个列，比如添加时间、数据种类等列名</p> <p>同时需要制定这一列的数据类型</p>
26.修改表-修改字段：重命名 可以修改列名的名字，也就是使用change 改字段的名字	<pre>alter table 表名 change 原名 新名 类型及约束;</pre> <p>例：</p> <pre>alter table students change birth birthday datetime not null default "2023-10-10";</pre> <p>这里修改的是一个列名，default是指默认值</p>
27.修改表-修改字段：不重命名 <ul style="list-style-type: none"> • modify只能修改类型，不能改列名的名字，也就是不能改字段的名字 	<pre>alter table 表名 modify 列名 类型及约束;</pre> <p>例：</p> <pre>alter table students modify birth date not null;</pre>
28.修改表-删除字段 字段就是一列，删除某一个特征	<pre>alter table 表名 drop 列名;</pre> <p>例：</p> <pre>alter table students drop birthday;</pre>
29.删除整个表	<pre>drop table 表名;</pre> <p>例：</p> <pre>drop table students; drop table device;</pre>
-----	-----
30.查询数据：查看表中的所有字段：	<pre>select * from 表名;</pre> <p>例：</p> <pre>select * from device_info;</pre>
31.查询表中指定字段	<pre>select 列1,列2,... from 表名;</pre> <p>例：</p>

	select sn,type from device_info;
32.按条件查询	select 列1,列2,... from 表名 where 条件; 例: select id,name,gender from students where id>=2 and id<=6; select sn,type from device_info where type = 1;
33.全列插入: <ul style="list-style-type: none"> 说明: 主键列是自动增长, 但是在全列插入时需要占位, 通常使用0或者 default 或者 null 来占位, 插入成功后以实际数据为准 全列插入: 值的顺序与表中字段的顺序对应 	insert into 表名 values(...) 例: insert into students values(0,'郭靖',1,'蒙古','2016-1-2');
34.部分列插入:	insert into 表名(列1,...) values(值1,...) 例: insert into students(name,hometown,birthday) values('黄蓉','桃花岛','2016-3-2');
35.全列多行插入: <ul style="list-style-type: none"> 值的顺序与给出的列顺序对应 	insert into 表名 values(...),(...)...; 例: insert into classes values(0,'python1'), (0,'python2');
36.删除数据	delete from 表名 where 条件 例: delete from students where id=5; delete from device_info where sn = XXXXXX;
37.修改数据	update 表名 set 列1=值1,列2=值2... where 条件; 例: update students set gender=0,hometown='北京' where id=5; update students set gender=0,hometown='北京' where name="诸葛亮";
-----	-----
38.条件查询:	select * from 表名 where 条件; 例: select * from students where id=1;

```
select * from device where sn = xxxx;
```

使用where子句对表中的数据筛选，
结果为true的行会出现在查询结果集中

where后面支持多种运算符，进行条件的处理

- 比较运算符
- 逻辑运算符
- 模糊查询
- 范围查询
- 空判断

比较运算符

- 等于: =
- 大于: >
- 大于等于: >=
- 小于: <
- 小于等于: <=
- 不等于: != 或 <>

例1：查询编号大于3的学生

```
select * from students where id > 3;
```

例2：查询编号不大于4的学生

```
select * from students where id <= 4;
```

例3：查询姓名不是“黄蓉”的学生

```
select * from students where name != '黄蓉';
```

例4：查询没被删除的学生

```
select * from students where is_delete=0;
```

逻辑运算符

- and
- or
- not

例5：查询编号大于3的女同学

```
select * from students where id > 3 and  
gender=0;
```

例6：查询编号小于4或没被删除的学生

```
select * from students where id < 4 or  
is_delete=0;
```

模糊查询

- like
- %表示任意多个任意字符
- _表示一个任意字符
-

例7：查询姓黄的学生

```
select * from students where name like  
'黄%';
```

例8：查询姓黄并且“名”是一个字的学生

```
select * from students where name like '黄  
_';
```

例9：查询姓黄或叫靖的学生

```
select * from students where name like  
'黄%' or name like '%靖';
```

范围查询

- in表示在一个非连续的范围
- between ... and ...表示在一个连续的范围

例10：查询编号是1或3或8的学生

```
select * from students where id in(1,3,8);
```

例11：查询编号为3至8的学生

```
select * from students where id between 3  
and 8;
```

例12：查询编号是3至8的男生

```
select * from students where (id between 3  
and 8) and gender=1;
```

	<p>空判断</p> <ul style="list-style-type: none"> • 注意：null与"是不同的 • 判空is null <p>例13：查询没有填写身高的学生 select * from students where height is null;</p> <p>例14：查询填写了身高的学生 select * from students where height is not null;</p> <p>例15：查询填写了身高的男生 select * from students where height is not null and gender=1;</p> <p>优先级</p> <ul style="list-style-type: none"> • 优先级由高到低的顺序为：小括号，not，比较运算符，逻辑运算符 • and比or先运算，如果同时出现并希望先算or，需要结合()使用
-----	----- -----

最终数据库密码：znailn0822554x&05060019

22.创建表

```
1 create table table_name(
2     id int unsigned not null auto_increment primary key,
3     name varchar(30) not null
4     column1 datatype contrai,
5     column2 datatype,
6     column3 datatype,
7     .....
8     columnN datatype,
9     PRIMARY KEY(one or more columns)
10 );
```

最后一条语句后面没有逗号

```
1 create table device(  
2     sn varchar(60) not null primary key,  
3     type int unsigned not null  
4 );
```

例子:

```
1 create table table_name(  
2     id int unsigned not null auto_increment primary key,  
3     name varchar(30) not null,  
4     age tinyint unsigned default 0,  
5     hight decimal(5,2),  
6     gender enum("男", "女") default "男",  
7     class_id int unsigned,  
8  
9     .....  
10    columnN datatype,  
11    PRIMARY KEY(one or more columns)  
12 );
```