|  |  |
| --- | --- |
| 1.登录数据库： | sudo mysql -u root -p  回车输入密码：lj19906788  sudo mysql -u admin -p  回车输入密码：lm19936688 |
| 2.查看数据库 | show databases; |
| 3.选择数据库 | use mysql;  use env\_monitor; |
| 4.查看数据库的使用者 |  |
| 5.env\_monitor远程访问权限 | update env\_monitor set host='%' where user='root'; |
| 6.创建数据库 | CREATE DATABASE dbname;  create database device\_info;  create database users; |
| 7.授权localhost主机上admin管理env\_monitro数据库的所有权限。 | grant all on env\_monitor.\* to admin@localhost; |
| 8.修改登录密码 | 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'; |
| 9.修改权限 | GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' WITH GRANT OPTION;  all表示所有权限 |
| 10。刷新权限 | FLUSH PRIVILEGES; |
| 11.设置root管理员在数据库中的专有密码： | sudo  mysql\_secure\_installation |
| 12.设置端口号： | 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 |
| 13.**通过本地python远程链接数据库** | **pip3 install python-mysql** |
| 14.**重新启动**mysql**：** | sudo service mysql restart |
| ----------------------------- | -------------------------------------------------------- |
| 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 | 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)  );      相关的关键字说明  create table table\_name();    auto\_increment 表示自动增长  not null 表示不能为空  primary key 表示主建,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 |
| 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 | 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());    注意填入数据时，严格按照每种数据的格式进行填写  varchar 类型的数据使用单引号  int 类型的数据使用整型数字，不加单引号  插入时间使用now()实现 |
| 24.查看表中的所有数据： | select \* from 表名;  selec \* from device\_info; |
| 25.修改表-添加字段 | alter table 表名 add 列名 类型;  例：  alter table students add birthday datetime;  alter table device\_info add temprature int；    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  同时需要制定这一列的数据类型 |
| 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 | alter table 表名 change 原名 新名 类型及约束;  例：  alter table students change birth birthday datetime not null default "2023-10-10";    这里修改的是一个列名，default是指默认值 |
| 27.修改表-修改字段：不重命名   1. 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 | alter table 表名 modify 列名 类型及约束;  例：  alter table students modify birth date not null; |
| 28.修改表-删除字段  字段就是一列，删除某一个特征 | alter table 表名 drop 列名;  例：  alter table students drop birthday; |
| 29.删除整个表 | drop table 表名;  例：  drop table students;  drop table device； |
| ----------------------------- | --------------------------------------------------------------- |
| 30.**查询数据：查看表中的所有字段：** | select \* from 表名;  例：  select \* from device\_info; |
| 31.**查询表中指定字段** | select 列1,列2,... from 表名;  例：  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;  selet sn,type from device\_info where type = 1; |
| 33.**全列插入：**   1. 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 2. 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：**   1. 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   * 1. 比较运算符   2. 逻辑运算符   3. 模糊查询   4. 范围查询   5. 空判断   [比较运算符](https://doc.itprojects.cn/0001.zhishi/python.0020.mysql/index.html#/20?id=%e6%af%94%e8%be%83%e8%bf%90%e7%ae%97%e7%ac%a6)   1. 等于: = 2. 大于: > 3. 大于等于: >= 4. 小于: < 5. 小于等于: <= 6. 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](https://doc.itprojects.cn/0001.zhishi/python.0020.mysql/index.html#/20?id=%e9%80%bb%e8%be%91%e8%bf%90%e7%ae%97%e7%ac%a6)   1. and 2. or 3. 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](https://doc.itprojects.cn/0001.zhishi/python.0020.mysql/index.html#/20?id=%e6%a8%a1%e7%b3%8a%e6%9f%a5%e8%af%a2)   1. like 2. %表示任意多个任意字符 3. \_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](https://doc.itprojects.cn/0001.zhishi/python.0020.mysql/index.html#/20?id=%e8%8c%83%e5%9b%b4%e6%9f%a5%e8%af%a2)   1. in表示在一个非连续的范围内 2. between ... and ...表示在一个连续的范围内     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;      [空判断](https://doc.itprojects.cn/0001.zhishi/python.0020.mysql/index.html#/20?id=%e7%a9%ba%e5%88%a4%e6%96%ad)   1. 注意：null与''是不同的 2. 判空is null       例13：查询没有填写身高的学生  select \* from students where height is null;  例14：查询填写了身高的学生  select \* from students where height is not null;  例15：查询填写了身高的男生  select \* from students where height is not null and gender=1;  [优先级](https://doc.itprojects.cn/0001.zhishi/python.0020.mysql/index.html#/20?id=%e4%bc%98%e5%85%88%e7%ba%a7)   1. 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 2. and比or先运算，如果同时出现并希望先算or，需要结合()使用 |
| ----------------------------- | -------------------------------------------------------------- |
|  |  |
|  |  |

22.创建表

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)

);

最后一条语句后面没有逗号

create table device(

sn varchar(60) not null primary key,

type int unsigned not null

);

例子：

create table table\_name(

id int unsigned not null auto\_increment primary key,

name varchar(30) not null,

age tinyint unsigned default 0,

hight decimal(5,2),

gender enum("男"，"女") default "男"，

class\_id int unsigned,

.....

columnN datatype,

PRIMARY KEY(one or more columns)

);