复习：

1. 常见的数据库有哪些？

Mysql 甲骨文

Oracle 甲骨文

SqlServer 微软

DB2 IBM

Sybase Sybase

Access 微软

1. 生活中哪些地方使用到数据库？

Mantis ------储存缺陷

12306 ------- 存储车次

京东/美团 --------存储历史记录

1. 什么是数据库？

存储数据的仓库

一、数据库的发展史（了解）

1、程序管理阶段（20世纪50年代中期）

特点：数据不能长期保存

1. 文件系统阶段（20世纪50年代后期--60年代后期）

特点：数据缺乏独立性

1. 数据库系统阶段（60年代后期----？）

特点：数据共享、减少冗余

1. Mysql介绍

Mysql是一款关系型数据库管理系统，由瑞典公司研发，目前属于Oracle公司，

是最流行的数据库之一，主要分为两大版本，分别是社区版（免费版）和企业版（付费版），主要特点：体积小、速度快、可跨平台（Win/Linux/Mac）

使用场景：功能项目时，配合数据库进行查询，接口测试时，落库检查。

1. 数据库常用术语
2. 关系

一个关系就是一张2维表（表）

----Excal

1. 属性

2维表中的一列，称为属性，工作一般称为列或者字段

1. 元组

2维表中的一行，称为元组，工作一般称为行或者记录

1. Mysql访问和调式

1、准备工作：

1. 启动wampserver（绿色）
2. 访问Mysql
3. 通过DOS访问（临时）

步骤：

A：在DOS窗口中，输入：

Xxxx>mysql -uroot -p

Enter Password :在此输入密码

Mysql>

1. 通过工具（SQLyong）访问（工作）

常见工具（连接Mysql）介绍：

Sqlyog、Navicat、phpmyadmin、

Workbench、

步骤：

1. 启动SQLyog
2. 单机新建，输入连接名（自定义）
3. 单击测试连接，再单击连接按钮
4. SQL语句概述
5. SQl语句分类

数据定义语句

主要是对数据库中的表，创建、修改、删除

创建----create

修改----alter

删除----drop

数据操作语句

主要是对数据库表中的数据，插入、更新、删除

插入----insert

更新-----update

删除----delete

数据查询语句（重点）

主要是对数据库表中的数据，查询

查询----select

事务控制语句

数据控制语句

2、SQL语句编写规范

1）SQL语句不区分大小写

2）SQL语句建议关键词大写，其他部分小写，根据公司情况，具体要求。

3）一条SQL语句以分号结尾（;)

1. 数据库基础语句
2. 查看当前全部的数据库

语法格式：show databases;

1. 创建数据库

语法格式：create database 数据库名称

Charset 字符集

说明：

1. 常见的字符集有gbk、utf8
2. 数据库名称不能重复

案例：创建一个数据库，名称为：testing，字符集设置为utf8。

Create databases testing charset utf8;

练习：创建一个数据库，名称为tedu，字符集设置为gbk

Create database tedu1 charset gbk;

1. 查看创建好的数据库信息

语法格式：

Show create database 数据库名称；

案例：查看testing数据库信息

Show create database testing；

练习：查看mysql数据库信息

Show database mysql；

1. 选择数据库

语法格式：use 数据库名称；

案例：选择testing 数据库

Use testing；

练习：选择tedu1数据库

Use tedu1；

（5）查看数据库中全部的表

语法格式：Show tables；

案例：选择testing数据库，查看数据库中全部的表

Use testing；

Show tables；

练习：选择mysql数据库，查看数据库全部的表

1. 删除数据库

Drop database 数据库名；

案例：删除数据库testing，并验证

Drop database testing;

Show databases;

综合练习：

1. 创建一个数据库，名称为：test01，字符集设置为utf8

CREATE database test01 CHARSET utf8;

1. 查看test01数据库信息

show create database test01;

1. 选择test01数据库

use test01;

1. 查看test01数据库中全部的表

show tables;

1. 删除test01数据库，并验证

drop database test01;

show databases;

1. 数据类型

数据类型就是对数据（文字、字母、符号、音频、图片、视频）进行分类

1. 数值数据类型
   1. 整数类型

int或int（n） 表示整数类型，用来存储整数，n表示数值的宽度和大小无关

例如：int（5） 66666

b）小数类型

Double（n，m）表示小数类型，用来储存小数，n表示数值的共位数，m表示小数的位数

例如：double（7，2） 78954.17

1. 字符数据类型
   1. 固定长度字符类型

Char（n） 表示固定长度字符，n表示字符的长度

例如：char（5） abcde

* 1. 可变长度字符类型

Varchar（n） 表示可变长度字符，n表示字符的最大长度。

例如：varchar（10） abcdefg

1. 日期时间数据类型
   1. 日期类型

Date 表示日期类型，一般日期类型格式：YYYY-MM-DD

* 1. 日期时间类型

Datetime 表示日期时间类型，一般日期时间类型格式：

YYYY-MM-DD HH:MM:SS

1. 数据库操作语句
2. 创建表

语法格式：

create table 表名（

列名1 数据类型，

列名2 数据类型，

........,

列名n 数据类型

）;

说明：

表名不能重复

表中最后一列，不加逗号

案例：创建一张表，表名为test02，表中包含的字段有：id int，name varchar（30） ，age int

create table test02(

id INT,

name VARCHAR(30),

age int

);

1. 查看表结构

语法格式：desc 表名；

案例：查看test01表，表结构

Desc test01；

练习：查看test02表，表结构

Desc test02；

1. 插入语句
2. 向全部列插入数据

语法格式：

Insert into 表名 values(列值1，列值2，。。。。列值n);

··valuses中的列值必须和表结构中的列名是一一对应的。（数量、顺序、类型）

··在数据库中，数据类型数据，直接填写，字符和日期时间类型数据，需要添加单引号，对列值。

案例：根据语法格式，向test02插入数据

insert into test02 VALUES(101,'rose',20);

insert into test02 VALUES(102,'jack',25);

insert into test02 VALUES(103,'周杰伦',30);

insert into test02 VALUES(104,'mary',23);

练习：向test01表中，插入2条记录

1. 查看test01表结构

Desc test01；

1. 根据语法格式，插入数据

insert into test01 VALUES(1,'唐家三少',28,’男’,’杭州市’);

insert into test01 VALUES(1,'烽火戏诸侯',30,’男’,’上海市’);

1. 验证

Select \* from test01；

1. 向指定列插入数据

语法格式：

Insert into 表名 （列名1，列名2，。。。列名n）values(列值1，列值2，。。。。列值n);

说明：表名中的列名必须和values中的列值是一一对应的。

案例：向test01表中，插入数据

Id name age sex address

10 耳根 18

20 55 男 北京

21 天蚕土豆丝

我吃西红柿 男 广州

insert into test01(id,name,age)VALUES (10,'耳根',18);

INSERT INTO test01(id,age,sex,address) VALUES (20,55,'男','北京');

INSERT INTO test01 (id,name) VALUES (21,'天蚕土豆丝');

INSERT into test01 (name,sex,address) VALUES ('我吃西红柿','男','广州');

1. 更新数据

语法格式：

Update 表名 set 列名=该列新值，列名2=该列新值，.....列名n=该列新值 where 条件；

说明：如果没有where条件

更新某列的全部值

案例：跟新test01表中数据，将编号是20的，姓名更新为毕淑敏

UPDATE test01 SET name='毕淑敏' where id=20;

练习：更新test01表中数据，将姓名是我吃西红柿的，编号更新为22

UPDATE test01 set id=22 where name='我吃西红柿';

案例：更新test01表中数据，将标号是10号的，性别更新为男，地址更新为上海

update test01 set sex='男',address='上海' where id=10;

练习：更新test01表中的数据，将表中年龄全部更新为18

update test01 set age=18;

1. 删除表中数据

语法格式：

Delete from 表名 where 条件；

说明：如果没有where条件，删除表中所有的数据

案例：删除test01表，姓名是天蚕土豆丝的记录

DELETE from test01 where name='天蚕土豆丝';

练习：删除test01表中，全部数据

DELETE from test01;

1. 删除表

语法格式：

Drop table 表名；

案例：删除test01表，并验证

drop table test01;

练习：删除test02表，并验证

drop table test02;

1. 查询语句（学习重点）
2. 查询表中，全部列数据

语法格式：select \* from 表名；

说明：\* 表示全部的列

案例：查询student 表中，全部列数据

Select \* from student；

练习：查询emp表中，全部列数据

SELECT \* from emp;

2)查询表中，指定列数据

语法格式：

select 列名1，列名2，....列名n from 表名;

案例：查询student表中，学院编号（sid），姓名（sname），年龄（age），分数（score），地址（address）

Select sid,sname,score,address from student;

练习：查询dept表中，部门编号（deptno），部门名称（dname），部门地址（loc）

select deptno,dname,loc from dept;

1. 给列起别名

语法格式：

Select 列名1 as 别名1，列名2 as 别名2，....

列名n as别名n from 表名；

说明，还可以省略as，别名和别名之间使用空格分隔。

案例：查询emp表中，员工编号（empno），员工姓名（ename），入职时间（hiredate），工资（sal），并给每列起别名显示

select empno "员工编号",ename "员工姓名",hiredate "入职时间",sal "工资" from emp;

练习：查询cou01表中，课程编号（cno），课程名称（cname），授课老师（ctea），并给每列起别名显示

select cno "课程编号",cname "课程名称",ctea '授课老师' from cou01;

1. 去掉重复的列值（distinct）

语法格式：

Select distinct 列名 from 表名；

案例：查询emp表中，员工的部门编号（deptno），并去重显示

Select distinct deptno from emp;

练习：查询stedent表中，所属班级（sclass），并去重显示

select DISTINCT sclass '所属班级' from student;

1. 分页（限制）查询（limit）

语法格式：

Select \*/列名 from 表名 limit 初始位置，行数；

说明：

1. 初始位置 表示从那行开始查询，是一个可选值，如果每添加初始位置，默认值是0，表示从第一行开始查询。
2. 行数 表示查询的行数，

案例：查询student表中，前5条记录

SELECT \* from student LIMIT 5;

练习：查询student表中，从第8行开始查询，共查询5条记录

Select \* from student limit 7,5;

练习：查询emp表中，从第6行开始查询，到第10行结束

Select \* from student limit 5,5;

1. 排序（order by）

语法格式：

Select \*/列名 from 表名 order by 列名

Asc/desc ，列名2 asc/desc；

说明：asc表示升序

Desc 表示降序

多（两）列排序规则：现根据前面的列进行排序，如果前面的列有相同的列值，才会进行后面列的排序，否则，不排序

案例：查询emp表中，员工姓名（ename），职位（job），工资（sal），入职时间（hiredate），根据工资升序排列

Select ename ,job,hiredate from emp order by sal asc;

练习：查询emp表中，员工姓名（ename），职位（job），工资（sal），入职时间（hiredate），根据员工姓名降序排列

select job,sal,hiredate from emp order by ename desc;

案例：查询emp表中，员工姓名（ename），职位（job），工资（sal），部门标号（deptno），根据部门编号升序排列，再根据工资降序排列

Select ename,job,sal,deptno from emp order by deptno asc, sal desc;

练习：查询student表中学院编号（sid），姓名（sname），分数（score），班级（sclass），根据班级升序排列，再根据分数降序排列

SELECT sid,sname,score,sclass from student order by sclass ASC,score desc;

1. 条件查询（where）

语法格式：

Select \*/列名 from 表名

Where 条件

Order by 列名 asc/desc ，列名 asc/desc；

条件说明：

1. 关系运算符

>、<、=、>=、<=、<>、！=

1. 逻辑运算符

And（与）、or（或）、not（非）

1. 特殊情况

Is 、between....and、in、like

案例：查询student表中，分数（score）大于80分的，学员的信息

SELECT \* from student where score>80;

练习：查询emp表中，部门编号（deptno）是30号部门的，员工信息

练习：查询student表中，分数小于等于90的，学员姓名（sname），分数（score），地址address

SELECT sname,score,address from student where score<=90;

案例：查询emp表中，工资不等于1250的，员工编号（empno），姓名（ename），工资（sal），根据工资降序排列

SELECT empno,ename,sal from emp where sal!=1250 ORDER BY sal desc;

案例：查询student表中，分数在75--90分之间的，学员的编号（sid），姓名（sname），分数（score）

SELECT sid,sname,score FROM student where score>=75 and score <=90;

说明：and 表示与的意思，可以理解为并且（和），可以通过and连接多个条件。

例如：条件1and 条件2 and 条件3 ....

练习：查询emp表中，部门编号（deptno）是30号并且工资大于3000的，员工的姓名（ename），职位（job），工资（sal），部门编号（deptno）

SELECT ename,job,sal,deptno from emp where deptno=30 and sal>3000;

案例：查询student 表中，班级是1年1班或者分数大于90的，学员姓名（sname），分数（score），班级（sclass）

SELECT sname,score,sclass from student where sclass='1年1班' or score>90;

说明：or表示或的意思，可以理解为或者，可以通过or来连接多个条件

例如：条件1 or 条件2 or 条件3......

练习：查询emp表中，员工编号（empno）是7521、7369、7902的员工信息

SELECT \* from emp where empno=7521 or empno=7369 or empno=7902;

案例：查询emp表中，工资（sal）不等于1250的，员工的信息

SELECT \* from emp where sal!=1250;

SELECT \* from emp where NOT sal=1250;

说明：not表示非的意思，可以理解为取反，一般放在列名前

案例：查询emp表中，奖金（comm）是NULL的，员工的信息

SELECT \* from emp WHERE comm is NULL;

练习：查询emp表中，奖金（comm）不是NULL的，员工的信息

SELECT \* from emp where comm is not null;

SELECT \* from emp where not comm is null;

案例：查询emp表中，工资在1250---4450之间的，员工姓名（ename），

工资（sal），部门编号（deptno）

SELECT ename,sal,deptno from emp where sal BETWEEN 1250 and 4450;

说明：使用between...and 查询出来的结果包含最小值和最大值

SELECT ename,sal ,deptno from emp where sal>=1250 and sal<=4450;

练习：查询emp表中，工资不在1250---4450之间的，员工姓名（ename），

工资（sal），部门编号（deptno）

SELECT ename,sal,deptno from emp where sal not BETWEEN 1250 and 4450;

SELECT ename,sal ,deptno from emp where not sal>=1250 or not sal<=4450;

select ename,sal,deptno from emp where sal<1250 or sal>4450;

案例：查询student表中，班级是1年1班，1年2班，2年2班的，学员信息

SELECT \* from student where sclass="1年1班" or sclass="1年2班" or sclass="2年2班";

SELECT \* from student where sclass in ( "1年1班","1年2班","2年2班");

说明：in表示要查询的列值，是否保存在某列的列值中。

练习：查询student表中，地址是北京、上海、广州的学员信息

SELECT \* from student where address in ("北京","上海","广州");

练习：查询student表中，地址不是北京、上海、广州的学员信息

SELECT \* from student where not address in ("北京","上海","广州");

模糊查询（like）

语法格式：

Select \*/列名 from 表名 where 列名

Where 列名 like 条件；

条件说明：

% 表示0个 或者多个任意字符

\_ 表示任意1个字符

案例：查询emp表中，员工姓名首字母是M的，员工信息

select \* from emp where ename like "M%";

练习：查询emp表中，员工姓名尾字母是N的，员工信息

SELECT \* from emp where ename like "%N";

案例：查询emp表中，员工姓名中包含字母N的，员工的信息

SELECT \* from emp where ename like "%N%";

案例：查询emp表中，员工姓名中倒数第2个字母是N的，员工的信息

SELECT \* from emp WHERE ename like "%N\_";

练习：查询student 表中，学员姓名（sname），姓唐的，学员信息

SELECT \* from student where sname like "唐%";

练习：查询emp表中，员工姓名不包含字母O的，员工姓名（ename），职位（job），工资（sal）

SELECT ename,job,sal from emp where not ename like "%o%";

练习：查询emp表中，姓名不包含字母K并且奖金不为NULL的，员工的姓名（ename），职位（job），工资（sal），奖金（comm），根据工资升序排列

1. 聚合（分组）函数

Count、sum、avg、min、max

1. min（列名） 求最小值

案例：查询emp表中，员工的最低工资是多少

Select min(sal) from emp;

1. max（列名） 求最大值

案例：查询student 表中，学员的最高分数是多少？

SELECT max(score) '最高分数' from student;

1. sum（列名） 求和

案例：查询emp表中，员工的工资总和

select sum(sal) '工资总和'from emp;

1. avg （列名） 求平均工资

案例：查询emp表中的员工的平均工资

SELECT avg(sal) from emp;

练习：查询student表中；班级是1年1班的，学生的最高分数是多少？

select max(score) '最高分' from student WHERE sclass='1年1班';

1. count(列名/\*) 统计数量

列名：

表示统计某张表中，列值不为NULL的，总数量

\*：

表示统计某张表中，总数量

案例：查询emp表中，员工的数量

SELECT count(\*) from emp;

案例：查询emp表中，奖金（comm）不为NULL的，员工的总数量

SELECT count(comm) from emp ;

1. 分组查询（group by）

说明：根据表中的某一列，把相同的列值分成一组，然后对每一组数据，使用聚合函数。

语法格式：

Select 聚合函数/列名 from 表名

Where 条件

Group by 列名

Order by 聚合函数/列名 asc/desc ，聚合函数/列名 asc/desc；

案例：查询student表中，男生和女生的人数

SELECT sex, count(\*) from student group by sex;

案例：查询emp表中，每个部门的人数

SELECT deptno,count(\*) from emp GROUP BY deptno ;

案例：查询emp表中，工资大于2000的，每个部门的人数，部门的最高工资

SELECT deptno,count(\*),max(sal)

FROM emp

where sal>2000

group by deptno;

练习：查询student表中，分数（score）在70-90之间的，每个班级（sclass）学员人数，每个班级平均分数

select sclass ,count(\*),avg(score)

from student

where score BETWEEN 70 and 90

GROUP BY sclass;

练习：查询student表中，分数（score）在70-90之间的，每个班级（sclass）学员人数，每个班级平均分数，根据班级的平均分数升序排列

select sclass ,count(\*),avg(score)

from student

where score BETWEEN 70 and 90

GROUP BY sclass

ORDER BY avg(score) asc;

案例：查询emp表中，姓名不包含字母K的，每个职位（job）的人数，每个职位的最高工资

SELECT job, count(\*),max(sal)

from emp

where not ename like '%K%'

GROUP BY job;

10、having语句

说明：having 语句就是对分组后的数据，再次进行过滤，经常和分组查询一起使用。having语句不可以单独使用。

语法格式：

Select 聚合函数/列名 from 表名

Where 条件

Group by 列名

Having 条件

Order by 聚合函数/列名 asc/desc ，聚合函数/列名 asc/desc；

* where和having区别：
* where是对整张表的数据进行过滤，可以单独使用。
* having是对**分组后**的数据进行过滤，不可以单独使用，必须和group by一起使用

案例：查询emp表中，部门的平均工资大于2000的，每个部门的编号，部门的人数，部门的平均工资

select deptno,count(\*),avg(sal)

FROM emp

group by deptno

having avg(sal)>2000;

**练习：**查询emp表中，部门的人数大于等于3人的，每个部门的编号（deptno），部门的人数，部门的工资总和

SELECT deptno,count(\*),sum(sal)

FROM emp

GROUP BY deptno

HAVING count(\*)>=3;

案例：查询student表中，班级的人数大于等于2人，每个班级的名称（sclass），班级的人数，根据班级的名称降序排列

SELECT sclass,count(\*)

FROM student

GROUP BY sclass

HAVING count(\*)>=2

ORDER BY sclass desc;

笔试题中SQL语句编写思路：总结

1. 读题，找关键字（查询、排序、分页、分组、
2. 按照先后顺序，排好关键字
3. 再次读题，填写SQL语句其他部分

having条件判断：

1. 先确定条件
2. 判断条件中，描述的列，是否包含在，查询的表中
   1. 存在（真列）：使用where
   2. 不存在（假列）：使用having

案例：查询emp表中，工资在1000-5000之间的，每个部门的编号（deptno），部门的人数，部门的平均工资，要求：部门的平均工资大于2000，根据部门的编号降序排列

SELECT deptno,count(\*),avg(sal)

from emp

where sal BETWEEN 1000 and 5000

GROUP BY deptno

having avg(sal)>2000

ORDER BY deptno desc;

1. 子（嵌套）查询

说明：在一条SQL语句内部，又包含一条SQL语句

语句：

案例：查询emp表中，工资大于平均工资的，员工的姓名（ename），职位（job），工资（sal）

1. 先求平均工资

Select avg(sal) from emp;

1. 合成

select ename,job,sal

from emp

where sal>(Select avg(sal) from emp)；

案例：查询emp表中，工资大于30号部门最高工资的，员工的编号（empno），姓名（ename），工资（sal），部门编号（deptno）

SELECT empno,ename,sal,deptno

from emp

where sal > (select max(sal) from emp where deptno=30);

练习：查询student表中，分数（score）大于1年3班最高分数的，学员信息

SELECT \* from student

where score>(select max(score) from student where sclass='1年3班' );

练习：查询student表中，和姓名叫唐家三少是同一个班级（sclass）的，学员信息

SELECT \* from student

where sclass = (SELECT sclass from student where sname='唐家三少');

练习：查询student表中，和姓名叫唐家三少是同一个班级（sclass）的，其他学员信息

SELECT \* from student

where sclass=(

SELECT sclass from student where sname='唐家三少') and sname!='唐家三少';

1. 多表连接查询

说明：由于要查询的数据，分布在不同的表中，为了一次获取不同表中的数据，就需要用多表连接查询。

1）内连接查询

说明：查询只查询满足条件（有关系）的记录

语法格式：

Select 别名1. \*/列名， 别名2.\*/列名2

From 表1 别名1，表2 别名2，表3 别名3，.....

Where 关联条件；

案例：查询emp表中，员工姓名（ename），职位（job），工资（sal），以及dept表中，部门名称（dname），部门地址（loc）

SELECT e.ename,e.job,e.sal,d.dname,d.loc

from emp e,dept d

where e.deptno=d.deptno ;

说明：关联条件就是要查询的表中，相同的列，使其相等。

练习：查询stu01表中，学员编号（sid），姓名（sname），性别（sex），以及sco01表中，分数（score）

SELECT st.sno,st.sname,st.sex,sc.score

FROM stu01 st,sco01 sc

where st.sno=sc.sno;

案例：查询emp表中，工资（sal）在1000-5000之间的，员工姓名（ename），职位（job），工资（sal）以及dept表中，部门名称（dname），部门地址（loc）

SELECT e.ename,e.job,e.sal,d.dname,d.loc

FROM emp e,dept d

WHERE e.deptno=d.deptno and e.sal BETWEEN 1000 and 5000;

案例：查询sco01表中课程编号（cno）是1002的，学院的编号（sno），姓名（sname），年龄（age），地址（address）

SELECT st.sno,st.sname,st.age,st.address

FROM sco01 s,stu01 st

where s.sno=st.sno and s.cno=1002;

案例：查询emp表中，工资（sal）在1000-5000之间的，员工姓名（ename），职位（job），工资（sal），以及dept表中，部门名称（dname），部门地址（loc），根据工资升序排列

select e.ename,e.job,e.sal,d.dname,d.loc

FROM emp e,dept d

WHERE e.deptno=d.deptno and e.sal BETWEEN 1000 and 5000

ORDER BY e.sal asc;

案例：查询emp表中，员工姓名（ename），职位（job），工资（sal），以及salgrade中，工资等级（grade），该等级下最低工资（losal），该等级下最高工资（hisal）---特殊情况1

SELECT e.ename,e.job,e.sal,s.grade,s.losal,s.hisal

FROM emp e,salgrade s

where e.sal BETWEEN s.losal and s.hisal;

案例：查询emp表中，员工姓名（ename），员工职位（job），工资（sal），上级领导编号（mgr），上级领导姓名（ename）--特殊情况2

SELECT a.empno,a.ename,a.job,a.sal,a.mgr,b.ename

FROM emp a,emp b

where a.mgr=b.empno

;

练习：查询emp表中，姓名（ename）不包含字母k并且部门编号（deptno）是10好部门的，员工姓名（ename），工资（sal），入职时间（hiredate），以及dept表中全部数据，根据入职时间降序排列

select a.ename,a.deptno,a.sal,a.hiredate,b.\*

FROM emp a,dept b

where a.ename not like '%k%' and a.deptno=10 and a.deptno=b.deptno

ORDER BY a.hiredate desc;

1. 外连接查询

* 左外连接查询
* 右外连接查询
* ~~全外连接查询~~

说明：查询左边表的全部数据，以及与之有关系右边表的部分数据

语法格式：

Select 别名1.\*/列名，别名2.\*/列名

From 左边表 别名1 left join 右边表 别名2

on 关联条件；

案例：查询emp表中，所有员工的编号（empno），姓名（ename），职位（job），工资（sal），以及dept表中，部门名称（dname），部门地址（loc）--使用外连接查询

SELECT e.empno,e.ename,e.job,e.sal,d.dname,d.loc

FROM emp e left join dept d

ON e.deptno=d.deptno;

练习：查询dept表中，所有部门的编号（deptno），部门名称（dname），以及emp表中员工姓名（ename），员工职位（job），工资（sal），根据工资升序排列

SELECT d.deptno,d.dname,e.ename,e.job,e.sal

FROM dept d left join emp e

ON d.deptno=e.deptno

ORDER BY e.sal asc;

案例：查询emp表中，员工姓名不包含字母C的，所有员工的编号（empno），姓名（ename），职位（job），以及dept表中，部门名称（dname），部门地址（loc），根据员工姓名升序排列

1. 约束

说明：约束就是在创建表的时候，对表设置一些规则，只有满足这些规则，才可以向表中插入数据，我们把这些规则叫做约束。

**常见的约束有哪几种？**

* 主键约束（primary key）
* 唯一约束（unique）
* 默认值约束（default）
* 非空约束（not NULL）

1、主键约束

说明：主键约束就是用来标识表中的数据，避免出现重复的数据，被主键约束所修饰的列，列值是唯一且非空的，一张表中只能有一个主键约束。

（1）创建表时，添加主键约束

CREATE TABLE test\_01(

id int PRIMARY key,

name varchar(30),

age int

);

#验证ID 列非空特性

INSERT INTO test\_01 (name,age) VALUES ('肉丝',20); 报错

# 验证id 列唯一性

INSERT INTO test\_01 VALUES (101,'京酱肉丝',30);

1. 联合主键（重点）

说明：使用主键约束修饰2列或者多列的值

CREATE table test\_02(

id int,

name varchar(30),

score DOUBLE(4,1),

PRIMARY key(id,name)

);

1. 修改表时，添加主键约束

语法格式：

Alter table 表名 add primary key（列名1，列名2，....列名n）；

预置条件：

Create table test\_03(

Id int,

Name varchar(30),

Sex char(3)

);

案例：修改test\_03表，对name列添加主键约束，并验证

ALTER table test\_03 add PRIMARY key (name);

desc test\_03;

练习：创建一张表，表名为test\_04,表中包含的字段有：id int，name varchar（30），

Address varchar（50），修改表时，对name和address列添加主键约束，并验证

create table test\_04 (id int,name varchar(30),address varchar(50));

ALTER table test\_04 add PRIMARY key (name,address);

desc test\_04;

1. 删除主键约束

语法格式：

Alter table 表名 Drop primary key；

案例：删除test\_01表中，主键约束，并验证

alter table test\_01 drop PRIMARY key;

desc test\_01;

练习：删除test\_04表中，主键约束，并验证

alter table test\_04 drop PRIMARY key;

desc test\_04;

2、唯一约束（unique）

说明：唯一约束就是，指定一列或者多列的组合值，使其具有唯一性，防止用户输入重复的数据，被唯一约束所修饰的列，列值是唯一的，可以为NULL，一张表中可以有多个唯一约束。

1. 创建表时，添加唯一约束

CREATE table test\_05(

id int PRIMARY key,

name varchar(30) unique,

email varchar(50),

phone char(11),

sex char(3)

);

#被唯一约束所修饰的列，可以为NULL

1. 修改表时，添加唯一约束

语法格式：

Alter table 表名 add unique (列名1，列名2，...列名n）；

案例：修改test\_05表，对email添加唯一约束，并验证

alter table test\_05 add unique (email);

练习：修改test\_05表，对phone列添加唯一约束并验证

ALTER table test\_05 add unique (phone);

1. 删除唯一约束（重点）

语法格式1：

Alter table 表明 drop index key\_name 值；

语法格式2:

Show keys from 表名；

案例：删除test\_05表中，name列上的唯一约束，并验证

#1)先找到name列，的key\_name的值

show KEYS from test\_05 ;

#2） 删除name列唯一约束

ALTER table test\_05 drop index name;

1. 验证

Desc test\_05;

案例：删除test\_05表中，address列上的唯一约束，并验证

show keys from test\_05;

alter table test\_05 drop index address;

desc test\_05;

3、默认值约束（default）

说明：默认值约束就是用来，指定某列的默认值，当执行插入操作的时候，如果默认值约束所修饰的列，没有插入列值，系统会自动将默认值变成列值，每列只有一个默认值。

1）创建表时，添加默认值约束

CREATE table test\_06(

id int PRIMARY key,

name varchar(30) unique,

age int default 18,

address varchar(30),

email varchar(30),

etime date

);

desc test\_06;

#验证默认值特性

INSERT INTO test\_06 (id,name,address,email,etime) VALUES

(101,'jack','北京','jack@163.com','2022-04-28');

SELECT \* from test\_06;

1. 修改表时，添加默认值约束

语法格式：

Alter table 表名 modify 列名 数据类型 default 默认值；

案例：修改test\_06表，将age列默认值修改为20，并验证

alter table test\_06 MODIFY age int default 20;

desc test\_06;

案例：修改test\_06表，对address列添加默认值，默认值为北京并验证

alter table test\_06 MODIFY address varchar(30) default '北京';

desc test\_06;

1. 删除默认值约束

语法格式：

Alter table 表名 modify 列名 数据类型 default null；

或者

Alter table 表名 modify 列名 数据类型；

案例：删除test\_06表中，age列上的默认值约束

alter table test\_06 MODIFY age int;

desc test\_06;

练习：删除test\_06 表中，address列上的默认值约束

alter table test\_06 MODIFY address varchar(50) default null;

desc test\_06;

1. 非空约束（not null）

说明：非空约束就是，当执行插入操作的时候，被非空约束所修饰的列，列值不能为空。

1）创建表时，添加非空约束

create table test\_07(

id int PRIMARY key,

name varchar(30) not null,

sex char(3) not null,

age int default 18,

phone char(11) UNIQUE,

email varchar(30),

etime date

);

desc test\_07;

# 验证非空约束

INSERT into test\_07(id,sex,phone) VALUES (101,'男','15510593332');

1. 修改表时，添加非空约束

Alter table 表名 modify 列名 数据类型 not null；

案例：修改test\_07表，对email列添加非空约束，并验证

alter table test\_07 MODIFY email varchar(30) not null;

desc test\_07;

练习：修改test\_07表，对etime列添加非空约束，并验证

alter table test\_07 modify etime date not null;

desc test\_07;

1. 删除非空约束

语法格式：

Alter table 表名 modify 列名 数据类型；

案例：删除test\_07表中，name列非空约束

Alter table test\_07 modify name varchar（30）；

练习：删除test\_07 表中，email列非空约束

alter table test\_07 modify email varchar(30);

1. 索引（index）

说明：索引是建立在表中，列上的数据库对象，用于提高查询速度，索引是一种提高查询效率的机制。

1. 创建索引

语法格式：

Create index 索引名称 on 表名（列名）；

预置条件：

Create table test\_08(

Id int primary key,

Name varchar(30) unique,

Phone char(11),

Address varchar(30)

);

案例；创建一个索引，名称为index\_phone\_test08，对test\_08表中，phone列添加索引

create index index\_phone\_test08 on test\_08(phone);

1. 查看表中索引

语法格式：

Show index from 表名；

案例：查看test\_08表中索引

Show index from test\_08;

说明：

* 被主键约束所修饰的列，自带索引
* 被唯一约束所修饰的列，自带索引

Select id,name,phone from test\_08

Where phone=‘13074767410’；--10s

创建索引

Select id,name,phone from test\_08

Where phone=’13074767410’;-3s

1. 视图--连接

说明：视图就是一张虚拟表，可以通过视图，查询1张或者多张表中的数据。

1. 创建视图

语法格式：

Create view 视图名称

As

查询语句；

案例：创建一张视图，视图名称为

View\_emp\_dept ，查询emp表中员工编号（empno），姓名（ename），职位（job），工资（sal），以及dept表中，全部列数据，作为视图的结果

Create view view\_emp\_dept

As

Select e.empno,e.ename,e.job,e.sal,d.\*

From emp e,dept d

Where e.deptno=d.deptno;

案例：查询emp表中，员工姓名（ename），职位（job），工资（sal），以及dept表中，部门名称（dname），部门地址（loc），根据工资降序排列

# 多表连接

select e.ename,e.job,e.sal,d.dname,d.loc

from emp e,dept d

where e.deptno=d.deptno

ORDER BY e.sal desc;

# 查询视图

select ename,job,sal,dname,loc

from view\_emp\_dept

ORDER BY sal desc;

1. 函数
2. round（数值，位数） 四舍五入函数

说明：

当位数>0，表示几位小数

当位数=0，表示不保留小数

当位数<0,表示小数点前第几位进行四舍五入

Select round（45.737，2）； 45.74

Select round （45.737，0）； 46

Select round （45.737，-2）； 0

Select round （45.737，-1） 50

1. char\_length(列名）

获取字符的长度（个数）

案例：查询emp表中，员工姓名，以及姓名的长度

select ename,CHAR\_LENGTH(ename) from emp;