# SQL常见面试题

1，基础

create table Employee (name char(20) not null, age int, income float)

insert into Employee (name, age, income values ("jack","28","8000"))

insert into Employee (firstname,lastname) select firstname,lastname from Employee\_2（双表）

select \* from Employee where income > 1000

SELECT e.AGE FROM EMPLOYEE e (别名)

Update EMPLOYEE set name =’jack’ where income like ‘80%’

update EMPLOYEE set name = (select name from EMPLOYEE\_2 where income = 1999) where income = 2000 （双表）

delele from Employee where lastname = "user\_1" and firstname like "ja%"

alter table EMPLOYEE add address char(60)

Alter table EMPLOYEE drop address

select top 25 percent \* from EMPLOYEE (前25%的表)

select name from EMPLOYEE order by income desc （收入降序， order by \*\* desc）

select age from EMPLOYEE group by age

--------------------------------------------------------------------------

2.用一条SQL 语句 查询出每门课都大于80 分的学生姓名(注意name里面有重复) 【min，group by】

name kecheng fenshu

张三 语文 81

张三 数学 75

李四 语文 76

李四 数学 90

王五 语文 81

王五 数学 100

王五 英语 90

select name from table group by name having min(fenshu)>80

3. 删除除了自动编号不同, 其他都相同的学生冗余信息【delete ，where ，not in， min，group by】

自动编号 学号 姓名 课程编号 课程名称 分数

1 2005001 张三 0001 数学 69

2 2005002 李四 0001 数学 89

3 2005001 张三 0001 数学 69

A: delete tablename where 自动编号 not in (select min(自动编号) from tablename group by学号, 姓名, 课程编号, 课程名称, 分数)

--------------------------------------------------------------------------

4.查询平均成绩大于60分的同学的学号和平均成绩；

select stuId,avg(score) from Scores group by stuId having avg(score) >60;

# 数据库基础

1. 主键、外键的作用

主键：是表中的唯一标示键。

外键：是主键的从属，表示了两个表之间的联系。

1. 索引是什么？优点与不足?

加快查询的技术，2分法什么的。 如果不使用索引，mysql会从第一条数据开始，读完整个表，直到找到相关 的数据，表越大，花费的时间越多。

优点： 1、保证表中数据的唯一性; 2、加速数据的检索速度（二分法）;

缺点： 要占用额外存储空间。

索引 Index（独立的数据结构）

MySQL只对<，<=，=，>，>=，BETWEEN，IN使用索引

在LIKE以通配符%和\_开头作查询时，MySQL不会使用索引。

1. 什么是触发器？

比如你有两个表 A 和 B， A表有ID 和 NAME两列，B表有ID,PLAY,NAMEID三列，A与B没有关联

如果你想删除B表的内容，但是又同时想删除A表中A.ID=B.NAMEID，那么你就要写两条语句

如果使用触发器，就只用写删除B表的内容语句，一旦删除内容，则触发器设定的表规则被触发，那么A表的 相关内容也一起删除了

# mysql常用函数：

0.面试常用

1）top 25 percent ： select top 25 percent \* from EMPLOYEE

2）order by ： select name from EMPLOYEE order by income desc

3）group by ： select age from EMPLOYEE group by age

4）having min() ： select name from table group by name having min(fenshu)>80

5) avg(score) ： select stuId,avg(score) from Scores group by stuId having avg(score) >60;

1.数学函数

1）ABS(X) :返回X的绝对值

2）15 % 7 :余数

3）ROUND(1.58) :四舍五入

2.字符串函数

1）CONCAT('My', 'S', 'QL'); :连结的字符串

2）LENGTH('text') :字符串长度

3）LOCATE('bar', 'foobarbar'); :字符串位置

4）REPLACE('www.mysql.com', 'w', 'Ww');

5）REVERSE('abc');

3.日期和时间函数

1）NOW() :返回当前日期时间

2）CURDATE() :返回当前日期

3）CURTIME() :返回当前时间

4）REPLACE('www.mysql.com', 'w', 'Ww');

5）REVERSE('abc');

4.流程控制类函数

1）IF(v,t,f) 如果v为真返回t，否则返回f

例如：SELECT IF(1!=1,'1','2') FROM DUAL;

输出：2

2）CASE WHEN v then v1 end 判断如果v为真返回v1

例如：SELECT CASE WHEN 1=1 THEN 'true' ELSE 'false' END FROM DUAL;

输出：true

查询姓“李”的老师的个数；

select count(distinct(Tname)) from Teacher where Tname like '李%';

查询每门课程被选修的学生数

select c#,count(S#) from sc group by C#;

查询课程名称为“数据库”，且分数低于60的学生姓名和分数

Select Sname,isnull(score,0) from Student,SC,Course

where SC.S#=Student.S# and SC.C#=Course.C# and Course.Cname='数据库'and score <60;

查询不及格的课程，并按课程号从大到小排列

select c# from sc where scor e <60 order by C# desc ;

求选了课程的学生人数

select count(\*) from sc;

查询各个课程及相应的选修人数

select count(\*) from sc group by C#;

检索至少选修两门课程的学生学号

select S# from sc group by s# having count(\*) > = 2

检索“004”课程分数小于60，按分数降序排列的同学学号

select S# from SC where C#='004'and score <60 order by score desc;