

SQL面试题讲解-（不含解答）

讲师：伊川

用一条SQL语句查询出每门课都大于80分的学生姓名

name	course	score
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

```
-- 建表语句
CREATE TABLE `mst_stu` (
  `name` varchar(255) DEFAULT NULL,
  `course` varchar(255) DEFAULT NULL,
  `score` int(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- 数据
INSERT INTO `mst_stu` VALUES ('张三', '语文', 81);
INSERT INTO `mst_stu` VALUES ('张三', '数学', 75);
INSERT INTO `mst_stu` VALUES ('李四', '语文', 76);
INSERT INTO `mst_stu` VALUES ('李四', '数学', 90);
INSERT INTO `mst_stu` VALUES ('王五', '语文', 81);
INSERT INTO `mst_stu` VALUES ('王五', '数学', 100);
INSERT INTO `mst_stu` VALUES ('王五', '英语', 90);
```

查询后一天 temperature 比前一天高的date

查找与之前（昨天的）日期相比温度更高的所有日期的 Id。

id	date	temperature
1	2022-04-01	20
2	2022-04-02	25
3	2022-04-03	21
4	2022-04-04	24

```
-- mst_weather
CREATE TABLE `mst_weather` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `date` date DEFAULT NULL,
  `temperature` int(255) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4;
INSERT INTO `mst_weather` VALUES (1, '2022-04-01', 20);
INSERT INTO `mst_weather` VALUES (2, '2022-04-02', 25);
INSERT INTO `mst_weather` VALUES (3, '2022-04-03', 21);
INSERT INTO `mst_weather` VALUES (4, '2022-04-04', 24);
```

查询每个主播的最大level以及对应的最小gap(注意:不是每个主播的最大level和最小gap)

zhuobo_id	level	gap
123	8	20
123	9	40
123	9	30
246	6	30
246	6	20

```
CREATE TABLE `mst_zhubo` (
  `zhubo_id` int(11) NOT NULL,
  `level` int(255) DEFAULT NULL,
  `gap` int(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
INSERT INTO `mst_zhubo` VALUES (123, 8, 20);
INSERT INTO `mst_zhubo` VALUES (123, 9, 40);
INSERT INTO `mst_zhubo` VALUES (123, 9, 30);
INSERT INTO `mst_zhubo` VALUES (246, 6, 30);
INSERT INTO `mst_zhubo` VALUES (246, 6, 20);
```

下表是每个课程class_id对应的年级(共有primary、middle、high三个),以及某种比率rate

class_id	grade	rate
abc123	primary	70%
abc123	middle	65%
abc123	high	72%
hjjk86	primary	69%
hjjk86	middle	63%
hjjk86	high	74%

请写出SQL查询出如下形式的表:

class_id	primary	middle	high
abc123	70%	65%	72%
hjjk86	69%	63%	74%

```
CREATE TABLE `mst_class` (
  `class_id` varchar(255) NOT NULL,
  `grade` varchar(255) DEFAULT NULL,
  `rate` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
INSERT INTO `mst_class` VALUES ('abc123', 'primary', '70%');
INSERT INTO `mst_class` VALUES ('abc123', 'middle', '65%');
INSERT INTO `mst_class` VALUES ('abc123', 'high', '72%');
INSERT INTO `mst_class` VALUES ('hjkk86', 'primary', '69%');
INSERT INTO `mst_class` VALUES ('hjkk86', 'middle', '63%');
INSERT INTO `mst_class` VALUES ('hjkk86', 'high', '74%');
```

怎么把这样一个表

year	month	amount
1991	1	1
1991	2	2
1991	3	3
1991	4	4
1992	1	1
1992	2	2
1992	3	3
1992	4	4

查成这样一个结果

year	m1	m2	m3	m4
1991	1	2	3	4
1992	1	2	3	4

```
CREATE TABLE `mst_year` (
  `year` int, `month` int, `amount` int
```

```

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
INSERT INTO `mst_year` VALUES (1991, 1, 1);
INSERT INTO `mst_year` VALUES (1991, 2, 2);
INSERT INTO `mst_year` VALUES (1991, 3, 3);
INSERT INTO `mst_year` VALUES (1991, 4, 4);
INSERT INTO `mst_year` VALUES (1992, 1, 1);
INSERT INTO `mst_year` VALUES (1992, 2, 2);
INSERT INTO `mst_year` VALUES (1992, 3, 3);
INSERT INTO `mst_year` VALUES (1992, 4, 4);

```

有两个表A和B，均有key和value两个字段，如果B的key在A中也有，就把B的value换为A中对应的value

这道题的SQL语句怎么写？

设计表，关系如下：教师、班级、学生、科室、科室与教师为一对多关系，教师与班级为多对多关系，班级与学生为一对多关系，科室中需体现层级关系。

1. 写出各张表的逻辑字段

教师 mst_Teacher

Tid	Tname	Kid
1	王老师	1
2	张老师	2
3	孙老师	3
4	李老师	3
5	伊老师	4

```

CREATE TABLE `mst_teacher` (
  `Tid` int PRIMARY KEY AUTO_INCREMENT,
  `Tname` varchar(10),
  `Kid` int
);
insert into mst_teacher VALUES(1,'王老师',1),(2,'张老师',2),(3,'孙老师',3),
(4,'李老师',3),(5,'伊老师',4);

```

班级 mst_cla

Cid	Cname
-----	-------

1	1班	
2	2班	
3	3班	

+-----+-----+

```
CREATE TABLE `mst_cla` (
  `Cid` int PRIMARY KEY AUTO_INCREMENT,
  `Cname` varchar(10)
);
insert into mst_cla VALUES(1,'1班'),(2,'2班'),(3,'3班');
```

教师&班级 mst_tc

+-----+-----+-----+

id	Tid	Cid
1	3	1
2	3	2
3	3	3
4	4	1
5	4	2
6	4	3

+-----+-----+

```
CREATE TABLE `mst_tc` (
  `id` int PRIMARY KEY AUTO_INCREMENT,
  `Tid` int, `Cid` int
);
insert into mst_tc VALUES(1,3,1),(2,3,2),(3,3,3),(4,4,1),(5,4,2),(6,4,3);
```

学生 mst_St

+-----+-----+-----+

SIId	Sname	Cid
1	赵雷	1
2	钱电	1
3	孙风	1
4	李云	2
5	周梅	2
6	吴兰	3
7	郑竹	3

+-----+-----+

```
CREATE TABLE `mst_St` (
  `SIId` int PRIMARY KEY AUTO_INCREMENT,
  `Sname` varchar(20), `Cid` int
);
insert into mst_St VALUES(1,'赵雷',1),(2,'钱电',1),(3,'孙风',1),(4,'李云',2),
(5,'周梅',2),(6,'吴兰',3),(7,'郑竹',3);
```

科室 mst_ks

+-----+-----+-----+

Kid	Kname	Pid
1	校长室	0

2	教学处	1
3	ui办公室	2
4	h5办公室	2

```

+-----+-----+-----+
CREATE TABLE `mst_ks` (
  `Kid` int PRIMARY KEY AUTO_INCREMENT,
  `Kname` varchar(20), `Pid` int
);
insert into mst_ks VALUES(1, '校长室', 0), (2, '教学处', 1), (3, 'ui办公室', 2), (4, 'h5办公室', 2);

```

根据上述表关系

查询教师id=4的学生数

```

-- 已知教师Id就可以先到教师与班级的关系表中获取当前老师所带的班级
select cid from mst_tc where Tid = 4;
-- 获取所带班级后,到学生表中获取所带班级中的学员数即可
select count(*) from mst_St where Cid in (select cid from mst_tc where Tid = 4);

```

查询科室id=2的下级部门数

```

-- 科室id=3的下级部门的pid也就等于科室id,因此查询当前科室的父级id = 2的即可
select count(*) from mst_ks where pid = 2;

```

查询所带学生最多的教师id

```

-- 先查询出每个老师带的学生数
select t.tid,t.tname,count(stu.sid)
from mst_teacher as t join mst_tc on t.tid = mst_tc.tid
join mst_st as stu on stu.cid = mst_tc.cid
group by t.tid,t.tname;

```

tid	tname	count(stu.sid)
3	孙老师	7
4	李老师	5

```

-- 在以上结果的基础上,按照学生数排序,取一个
select t.tid,t.tname,count(stu.sid) as num
from mst_teacher as t join mst_tc on t.tid = mst_tc.tid
join mst_st as stu on stu.cid = mst_tc.cid
group by t.tid,t.tname order by num desc limit 1;

```

(1)某奶粉品牌有以下销售数据(订单表Orderinfo), 请计算每个人得消费金额、消费频次、购买产品数量、第一次购买时间和最后一次购买时间。

CustomerID	OrderID	Sales	Quantity	OrderDate
A	01	100	1	2017-03-01
A	02	420	3	2017-03-15
B	03	300	4	2017-03-02
B	04	1000	1	2017-04-01
C	05	500	3	2017-05-03
C	06	200	1	2017-05-04
.....				

```
create table `mst_Orderinfo`(`CustomerID` char(5),`OrderID` int PRIMARY KEY
AUTO_INCREMENT,`Sales` int,`Quantity` int,`OrderDate` date);
insert INTO mst_Orderinfo VALUES
  ('A',01,100,1,'2017-03-01'),
  ('A',02,420,3,'2017-03-15'),
  ('B',03,300,4,'2017-03-02'),
  ('B',04,1000,1,'2017-04-01'),
  ('C',05,500,3,'2017-05-03'),
  ('C',06,200,1,'2017-05-04');
```

(2)该奶粉品牌还有一张订单明细表(OrderDetail),请结合上题得订单表, 计算出每个SKU被多少客户购买了。

OrderDetailID	OrderID	SKU	Qutity
01	01	SKU1	1
02	02	SKU1	2
03	02	SKU2	1
04	03	SKU2	2
05	03	SKU3	2
06	04	SKU6	1
07	05	SKU4	2
.....			


```
create table `mst_OrderDetail`(`OrderDetailID` int PRIMARY KEY
AUTO_INCREMENT,`OrderID` int,`SKU` char(5),`Qutity` int);
insert into mst_OrderDetail values(1,1,'SKU1',1),(2,2,'SKU1',2),
(3,2,'SKU2',1),(4,3,'SKU2',2),(5,3,'SKU3',2),(6,4,'SKU6',1),(7,5,'SKU4',2);
```

(3)请结合Orderinfo表与OrderDetail表，找出购买了SKU1又购买SKU2产品的人。

现有一张房源表。

字段有房源基础信息以及小区id，小区name字段。

求sql查出top10的房源量的小区，按房源量降序排序。

```
select 小区id,小区name,count(*) as 房源量 from 房源表 group by 小区id,小区name
order by 房源量 desc limit 10;
```

说一下你熟悉的数据库，这个数据库有什么特点？

mysql 关系型数据库 开源中最常用的数据库,有多种表引擎,支持事务,支持表关系 redis 非关系型数据库,键值对的存储方式,并且数据存储在内存中,有多种不同的数据类型 mongodb 非关系型数据库,基于分布式文件存储的数据库,是nosql中最像关系型数据库的

请用sql描述，如何给表中插入数据？如何更新数据？ insert into 表名(字段列表) values(参数列表)
update 表名 set 字段 = 新值,... where

你常用的mysql引擎有哪些？各引擎间有什么区别？ myisam 存储由三个文件组成,一个存储表结构,一个存储数据,一个存储索引 不支持事务,非聚簇索引 innodb 存储由两个文件组成,一个存储表结构,一个存储数据和索引 支持事务,主索引是聚簇索引,并且表中始终存在主键(即便在建表是没有创建主键)

MySQL数据库中现有如下表，名为tbl:

id	name	age	sex
1	zhangsan	25	1
2	lisi	22	0
3	wangwu	35	0
4	zhaoliu	24	1
5	tianqi	32	1

按照要求写出SQL语句:

(1)查询前3条纪录。

limit 3

(2)查询所有纪录，并按照字段age降序排列。

order by age desc

请描述MySQL中left join和inner join的区别。join 和 inner join 都是一样的,会连接两个表中存在关系字段的数据 和 where的关联查询一样 left join 和 right join一样 都是以左表(或右表)数据为基准,去查询另外一个表的数据, 如果没有对应字段的数据,则补充为null

请写出你知道MySQL储存引擎和区别。

请列出你最常用的mysql版本, mysql默认端口号是多少? 请写出你最常用的mysql数据库备份和恢复命令。 5.7, 端口 3306 127.0.0.1 localhost 本机的IP

-- 数据导入 mysql -uroot -p < F:\mysql-5.7.27-win64\bf\ops.sql

-- 数据备份 导出 mysqldump -uroot -p -B -F -R -x --master-data=2 ops >F:\mysql-5.7.27-win64\bf\ops.sql

参数说明: -B: 指定数据库 -F: 刷新日志 -R: 备份存储过程等 -x: 锁表 --master-data: 在备份语句里添加CHANGE MASTER语句以及binlog文件及位置点信息

数据库A:dataahn, 中具有表c、表d。数据库B:datapubhn, 现想要在数据库B中对数据库A中表c与表d中的字段进行增删改查。如何授权? grant select,insert,update,delete on B.*,A.c,A.d to 'zhansgan'@'%' identified by '123456'

对于处理高复杂sql, 数据库查询特别慢, 你有什么高招? 复杂查询简单化 不用嵌套,不用join,多表查询拆分 针对语句的查询字段 设计索引 给经常作为where条件的字段添加索引 必要时添加 联合索引

以下哪条语句在数据库sql书写时最为高效() B

A.select * from emp where dep>3

B.select * from emp where dep>=4

C.select * from emp where dep≥4

D.select * from emp where dep>=3 and dep≠4

数据库中执行删除操作, 若要删除表中的所有行, 建议使用() B

A.delete 语句

B.truncate 语句

C.drop 语句

D.commit 语句

2在查询语句的select子句中尽量避免使用__来表示全部列名。 *

SQL中增加、删除、修改对应的命令是什么？

查找条件为：姓名不是NULL的纪录() C

A.WHERE NAME ! NULL

B.WHERE NAME NOT NULL

C.WHERE NAME IS NOT NULL

D.WHERE NAME!=NULL

在SQL语言中，子查询是() D

A.选取单表中字段子集的查询语句

B.选取多表中字段子集的查询语句

C.返回单表中数据子集的查询语言

D.嵌入到另一个查询语句之中的查询语句

以下能够删除一列的是() B

A.alter table emp remove addcolumn

B.alter table emp drop column addcolumn

C.alter table emp delete column addcolumn

D.alter table emp delete addcolumn

学生关系模型S(S#,Sname,Sex,Age),S的属性分别表示学生的学号、姓名、性别、年龄。要在表S中删除一个属性"年龄",可选用的SQL语句是() D

A.UPDATE S Age

B.DELETE Age from S

C.ALTER TABLE S 'Age'

D.ALTER TABLE S DROP Age

补全语句:select vend_id,count(*) as num_prods from products group by____;

SELECT distinct 部门名称, count(sid), sum(个人工资) FROM 工资表 GROUP BY 部门名称 having count (sid) > 10;

查询结果是__。

用SELECT进行模糊查询时，可以使用匹配符，但要在条件值中使用__或%等通配符来配合查询。

MySQL是一种(多用户、单用户)的数据库管理系统。

对一个超过200个汉字的内容，应用一个型的字段来存放。

char 定长 长度范围 0-255 超出指定长度无法存储,

varchar 变长 长度范围 0-65535 超出指定长度无法存储

text 文本 长度范围 0-65535 不需要指定长度 text和以上两个字段的最大区别就是 text只能使用前缀索引

```
add index index_name(text(10))
```

查看当前数据库中表名语句是_____;

Mysql数据库GD2312、utf8\utf8mb4字符集的区别__。GD2312 收录了6763个汉字,其它文字符号682个,兼容ASCII字符 占用空间: 如果是ASCII字符,则占一个字节编码 如果是否其它则采用2个字节编码

UTF8 收录地球上能想到的所有字符, 而且还在不断扩充,同时兼容ASCII字符 存储空间为 1-4个字节

- utf8mb3: 阉割过的 utf8` 字符集, 只使用1~3个字节表示字符。
- utf8mb4: 正宗的 utf8 字符集, 使用1~4个字节表示字符。