### JAVA第二阶段—DAY02-JAVA作业答案

1. 简答题：描述一下MySQL数据操作语言有几类，分别的语法是什么。

答案：

插入语句（insert）

INSERT INTO 表名 (字段名1, 字段名2, 字段名3…) VALUES (值1, 值2, 值3);

修改语句（update）

UPDATE 表名 SET 字段名=值 WHERE 字段名=值;

删除语句（delete）

DELETE FROM 表名 WHERE 字段名=值;

TRUNCATE TABLE 表名;

1. 简答题：描述一下MySQL有几类约束。

答案：

非空约束

唯一性约束

默认值约束

主键约束

外键约束

检查约束

1. 简答题：关联查询有几类，其语法分别是什么？

答案：

交叉关联

select ... from 表a cross join 表b;

内关联

select ... from 表a inner join 表b on 表a.字段名=表b.字段名;

左外关联

select ... from 表a left join 表b on 表a.字段名=表b.字段名;

右外关联

select ... from 表a right join 表b on 表a.字段名=表b.字段名;

自然关联

select ... from 表a natiral join 表b on 表a.字段名=表b.字段名;

select ... from 表a natiral left join 表b on 表a.字段名=表b.字段名;

select ... from 表a natiral right join 表b on 表a.字段名=表b.字段名;

1. 编程题：

执行以下SQL，建表插数：

#创建表及插入记录

CREATE TABLE class (

cid int(11) NOT NULL AUTO\_INCREMENT,

caption varchar(32) NOT NULL,

PRIMARY KEY (cid)

) ENGINE=InnoDB CHARSET=utf8;

INSERT INTO class VALUES

(1, '三年二班'),

(2, '三年三班'),

(3, '一年二班'),

(4, '二年九班');

CREATE TABLE course(

cid int(11) NOT NULL AUTO\_INCREMENT,

cname varchar(32) NOT NULL,

teacher\_id int(11) NOT NULL,

PRIMARY KEY (cid),

KEY fk\_course\_teacher (teacher\_id),

CONSTRAINT fk\_course\_teacher FOREIGN KEY (teacher\_id) REFERENCES teacher (tid)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO course VALUES

(1, '生物', 1),

(2, '物理', 2),

(3, '体育', 3),

(4, '美术', 2);

CREATE TABLE score (

sid int(11) NOT NULL AUTO\_INCREMENT,

student\_id int(11) NOT NULL,

course\_id int(11) NOT NULL,

num int(11) NOT NULL,

PRIMARY KEY (sid),

KEY fk\_score\_student (student\_id),

KEY fk\_score\_course (course\_id),

CONSTRAINT fk\_score\_course FOREIGN KEY (course\_id) REFERENCES course (cid),

CONSTRAINT fk\_score\_student FOREIGN KEY (student\_id) REFERENCES student(sid)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO score VALUES

(1, 1, 1, 10),

(2, 1, 2, 9),

(5, 1, 4, 66),

(6, 2, 1, 8),

(8, 2, 3, 68),

(9, 2, 4, 99),

(10, 3, 1, 77),

(11, 3, 2, 66),

(12, 3, 3, 87),

(13, 3, 4, 99),

(14, 4, 1, 79),

(15, 4, 2, 11),

(16, 4, 3, 67),

(17, 4, 4, 100),

(18, 5, 1, 79),

(19, 5, 2, 11),

(20, 5, 3, 67),

(21, 5, 4, 100),

(22, 6, 1, 9),

(23, 6, 2, 100),

(24, 6, 3, 67),

(25, 6, 4, 100),

(26, 7, 1, 9),

(27, 7, 2, 100),

(28, 7, 3, 67),

(29, 7, 4, 88),

(30, 8, 1, 9),

(31, 8, 2, 100),

(32, 8, 3, 67),

(33, 8, 4, 88),

(34, 9, 1, 91),

(35, 9, 2, 88),

(36, 9, 3, 67),

(37, 9, 4, 22),

(38, 10, 1, 90),

(39, 10, 2, 77),

(40, 10, 3, 43),

(41, 10, 4, 87),

(42, 11, 1, 90),

(43, 11, 2, 77),

(44, 11, 3, 43),

(45, 11, 4, 87),

(46, 12, 1, 90),

(47, 12, 2, 77),

(48, 12, 3, 43),

(49, 12, 4, 87),

(52, 13, 3, 87);

CREATE TABLE student(

sid int(11) NOT NULL AUTO\_INCREMENT,

gender char(1) NOT NULL,

class\_id int(11) NOT NULL,

sname varchar(32) NOT NULL,

PRIMARY KEY (sid),

KEY fk\_class (class\_id),

CONSTRAINT fk\_class FOREIGN KEY (class\_id) REFERENCES class (cid)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO student VALUES

(1, '男', 1, '理解'),

(2, '女', 1, '钢蛋'),

(3, '男', 1, '张三'),

(4, '男', 1, '张一'),

(5, '女', 1, '张二'),

(6, '男', 1, '张四'),

(7, '女', 2, '铁锤'),

(8, '男', 2, '李三'),

(9, '男', 2, '李一'),

(10, '女', 2, '李二'),

(11, '男', 2, '李四'),

(12, '女', 3, '如花'),

(13, '男', 3, '刘三'),

(14, '男', 3, '刘一'),

(15, '女', 3, '刘二'),

(16, '男', 3, '刘四');

CREATE TABLE teacher(

tid int(11) NOT NULL AUTO\_INCREMENT,

tname varchar(32) NOT NULL,

PRIMARY KEY (tid)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO teacher VALUES

(1, '张磊老师'),

(2, '李平老师'),

(3, '刘海燕老师'),

(4, '朱云海老师'),

(5, '李杰老师');

完成以下练习题

1、查询所有的课程的名称以及对应的任课老师姓名

SELECT course.cname,teacher.tname FROM course left JOIN teacher ON teacher\_id=tid;  
SELECT course.cname,teacher.tname FROM course inner JOIN teacher ON teacher\_id=tid;  
SELECT course.cname,teacher.tname FROM course JOIN teacher ON teacher\_id=tid;  
SELECT course.cname,teacher.tname FROM course, teacher where course.teacher\_id=teacher.tid;

2、查询学生表中男女生各有多少人

SELECT gender,COUNT(sid) FROM student GROUP BY gender;

3、查询物理成绩等于100的学生的姓名

SELECT a.cname, c.sname, b.num FROM course a,score b,student c  
WHERE a.cid=b.course\_id AND a.cname='物理' AND c.sid = b.student\_id  
AND b.num = 100;  
SELECT a.cname, c.sname, b.num FROM course a,score b,student c  
WHERE a.cid=b.course\_id AND a.cname='物理' AND c.sid = b.student\_id  
having num = 100;  
select b.cname, c.sname, a.num  
from score a  
 left join course b on a.course\_id=b.cid  
 left join student c on a.student\_id=c.sid  
where b.cname='物理' and a.num=100

4、查询平均成绩大于八十分的同学的姓名和平均成绩

select b.sname, avg(num) as anum  
from score a  
 left join student b on a.student\_id=b.sid  
group by b.sname  
having anum>80.0;  
select t1.sname, t2.avg\_num from student t1 join (select student\_id, avg(num) avg\_num from score group by student\_id having avg(num) > 80) t2 ON t1.sid = t2.student\_id;

5、查询所有学生的学号，姓名，选课数，总成绩

select ta.sid, ta.sname, count(tb.sid) as ccount, sum(num) as sumnum  
from student ta  
 left join score tb on ta.sid=tb.student\_id  
group by ta.sid, ta.sname;

6、 查询姓李老师的个数

select count(\*) from teacher where teacher\_name like ‘%李%’

或

select count(\*) from teacher where teacher\_name instr(‘李’);

7、 查询没有报李平老师课的学生姓名

select te.sname  
from student te where te.sid not in (  
select distinct ta.student\_id  
from score ta   
 inner join course tb on ta.course\_id = tb.cid  
 inner join teacher tc on tc.tid = tb.teacher\_id  
where tc.tname like '李平%'  
)

8、 查询物理课程比生物课程高的学生的学号

select a.sid, a.sname, b.num, c.num  
from student a,   
 (select \* from score where course\_id=(select cid from course where cname like '%物理%')) b,  
 (select \* from score where course\_id=(select cid from course where cname like '%生物%')) c  
where a.sid = b.student\_id and a.sid = c.student\_id  
  and b.num > c.num

9、 查询没有同时选修物理课程和体育课程的学生姓名

select d.sname from student d where d.sid not in (  
select a.sid  
from student a,   
 (select \* from score where course\_id=(select cid from course where cname like '%物理%')) b,  
 (select \* from score where course\_id=(select cid from course where cname like '%体育%')) c  
where a.sid = b.student\_id and a.sid = c.student\_id )

10、查询挂科超过两门(包括两门)的学生姓名和班级

select b.sname, c.caption  
from student b, class c  
where b.sid in (  
select a.student\_id  
from score a  
where num<60  
group by a.student\_id  
having count(num)>=2  
) and b.class\_id = c.cid

11、查询选修了所有课程的学生姓名

select sname  
from student  
where sid in (  
 select student\_id  
 from score group by student\_id  
 having count(sid) = (  
 select count(cid)  
 from course  
 )  
);

12、查询李平老师教的课程的所有成绩记录

select \*  
from score  
where course\_id in (  
 select cid  
 from course  
 inner join teacher on course.teacher\_id = teacher.tid  
 where teacher.tname = '李平老师'  
);

13、查询全部学生都选修了的课程号和课程名

select c.cid, c.cname  
from course c  
where c.cid in (  
select course\_id  
from score  
group by course\_id  
having count(num) = (select count(\*) from student) );

14、查询每门课程被选修的次数

SELECT course\_id, COUNT(student\_id)  
FROM score  
GROUP BY course\_id;

15、查询之选修了一门课程的学生姓名和学号

select sid, sname  
from student  
where sid in (  
 select student\_id  
 from score group by student\_id  
 having count(sid) = 1  
 );

16、查询所有学生考出的成绩并按从高到低排序（成绩去重）

select distinct num  
from score  
order by num desc;

解析要点：

1、分析数据来源

2、分析数据关联关系

3、提取数据字段

17、查询平均成绩大于85的学生姓名和平均成绩

数据来源：成绩来源于成绩表，包含学生ID，科目成绩；学生姓名来源于学生表，包含学生ID，学生姓名；

关联关系：学生表，1对多关联成绩表。

提取字段：学生姓名、平均成绩。

select tb.sname, avg(ta.num) as avg\_num  
from score ta, student tb  
where ta.student\_id = tb.sid  
group by tb.sname  
having avg\_num > 85;  
-- 可能存在同名学生，成绩也会被平均，要怎么改。  
select tb.sid, tb.sname, avg(ta.num) as avg\_num  
from score ta, student tb  
where ta.student\_id = tb.sid  
group by tb.sid, tb.sname  
having avg\_num > 85;  
select tb.sname, avg(ta.num) as avg\_num  
from score ta, student tb  
where ta.student\_id = tb.sid  
group by tb.sid  
having avg\_num > 85;  
​  
-- 1、查询成绩表，找出平均成绩大于85分的学生id  
-- 2、根据学生id，到学生表中，找出学生姓名  
select sname, avg\_num  
from student ta  
 inner join (  
 select student\_id, avg(num) avg\_num  
 from score   
 group by student\_id  
 having avg(num) > 85  
 ) tb on ta.sid = tb.student\_id

18、查询生物 成绩不及格 的学生姓名和对应生物分数

数据来源：

成绩来源于成绩表，包含学生ID，学科ID，科目成绩；学生姓名来源于学生表，包含学生ID，学生姓名；学科表，包含学科ID，学科名称。

关联关系：学生表，1对多关联成绩表；学科表，1对多关联成绩表

提取字段：学生姓名、生物成绩。

-- 1、根据学科名称，找到生物学科的学科id  
-- 2、根据学科id，找到不及格的学生Id和成绩  
-- 3、根据学生id，找到学生名称  
​  
select st.sname, sc.num  
from student st, (  
 select sc.student\_id, sc.num  
 from score sc  
 where sc.num < 60  
   and sc.course\_id = (  
   select cid from course  
 where cname like '%生物%'  
 )  
 ) sc  
where st.sid = sc.student\_id;

19、查询在所有选修了李平老师课程的学生中，这些课程(李平老师的课程，不是所有课程)平均成绩 最高的学生姓名

数据来源：

成绩来源于成绩表，包含学生ID，学科ID，科目成绩；

老师名称来源于老师表，包含老师ID，老师名称；

学科ID来源于学科表，包含学科ID，学科名称，授课老师id；

学生姓名来源于学生表，包含学生ID，学生姓名；

关联关系：学生表，1对多关联成绩表；学科表，1对多关联成绩表，老师表，1对多关联学科表

提取字段：学生姓名

select td.sname, td.sid, avg(tc.num) as avg\_num  
from teacher ta, course tb, score tc, student td  
where ta.tid = tb.teacher\_id  
  and tb.cid = tc.course\_id  
  and tc.student\_id = td.sid  
  and ta.tname like '李平%'  
group by td.sname, td.sid  
order by avg\_num desc  
limit 1;  
select td.sname  
from teacher ta, course tb, score tc, student td  
where ta.tid = tb.teacher\_id  
  and tb.cid = tc.course\_id  
 and tc.student\_id = td.sid  
  and ta.tname like '李平%'  
group by td.sname, td.sid  
order by avg(tc.num) desc  
limit 1;  
​  
-- 用子查询处理，拆开成几个步骤   
-- 1、根据李平老师名称查出来对应老师ID  
-- 2、根据老师ID查出来对应学科ID  
-- 3、根据学科ID查出来平均成绩和学生ID，并max，或按平均成绩降序排列，取第一条  
-- 4、根据学生ID查出来学生名称  
select ta.sname from student ta, (  
select student\_id, avg(num) as avg\_num from score where course\_id in (  
select cid from course where teacher\_id =  
 ( select tid from teacher where tname like '%李平%' )  
)  
group by student\_id  
order by avg\_num desc  
limit 1 ) tb  
where ta.sid = tb.student\_id;

20、查询每门课程 成绩 最好的前两名学生姓名

数据来源：

成绩来源于成绩表，包含学生ID，学科ID，科目成绩；

学科ID来源于学科表，包含学科ID，学科名称；

学生姓名来源于学生表，包含学生ID，学生姓名；

关联关系：学生表，1对多关联成绩表；学科表，1对多关联成绩表；

提取字段：学生姓名，学科名称

-- 1、从成绩中读取，每门课程最高、第二高的成绩  
-- 2、根据成绩，从成绩表中读取学生id，科目id  
-- 3、根据科目id，读取科目名称  
-- 4、根据学生id，读取学生名称  
select tc.cname, td.sname, te.num  
from course tc, student td, (  
   select ta.student\_id, ta.course\_id, ta.num from score ta, (   
 select sid, num,  
 (select num from score as s2 where s2.course\_id = s1.course\_id order by num desc limit 1) as first\_num,  
       (select num from score as s3 where s3.course\_id = s1.course\_id order by num desc limit 1,1) as second\_num  
     from score as s1  
 ) tb where ta.sid = tb.sid and (ta.num = tb.first\_num or ta.num = tb.second\_num)  
 ) as te  
where tc.cid = te.course\_id and td.sid = te.student\_id  
order by cname, num desc  
​  
-- 这种模式下，一个科目会出现多于2名最高分学生出现，应该怎么改  
select ta.sid, tb.sname, tc.course\_id, tc.num from   
( select score.sid, score.course\_id as course\_id, score.student\_id,  
     t.s1id, t.s2id  
   from score  
     left join(  
 select sid, course\_id,  
 (select student\_id from score as s2 where s2.course\_id = s1.course\_id order by num desc, student\_id limit 0,1) as s1id,  
       (select student\_id from score as s3 where s3.course\_id = s1.course\_id order by num desc, student\_id limit 1,1) as s2id  
       from score as s1  
     ) as t on score.sid = t.sid  
   where score.student\_id = t.s1id or score.student\_id = t.s2id  
) as ta, student tb, score tc  
where ta.student\_id = tb.sid  
 and ta.sid = tc.sid

21、查询不同课程但成绩相同的学号，课程号，成绩

select distinct s1.student\_id, s1.course\_id, s1.num, s2.num  
from score as s1, score as s2  
where s1.num = s2.num  
 and s1.course\_id != s2.course\_id;

22、查询没学过“李平”老师课程的学生姓名以及选修的课程名称；

-- 1、查出不是叶平老师教授的课程ID，课程名称  
-- 2、根据这些课程ID对应的学生Id  
-- 3、根据学生Id，找到学生名称  
select te.sname, tf.cname from student te, (  
select td.student\_id, tc.cname from score td  
 inner join (  
 select tb.cid, tb.cname from teacher ta, course tb  
 where ta.tname not like '%李平%' and tb.teacher\_id=ta.tid  
 ) as tc on td.course\_id =tc.cid  
) tf  
where te.sid = tf.student\_id

23、查询所有选修了学号为1的同学选修过的一门或者多门课程的同学学号和姓名；

-- 1、查询学号为1的学生修过的课程id  
-- 2、根据课程id，查询成绩表中学习这些学科的学生ID  
-- 3、根据学生id，查询这些学生ID的姓名  
select sid, sname from student where sid in (  
select student\_id from score where course\_id in   
 ( select course\_id from score where student\_id=1 )  
)

24、任课最多的老师中学生单科成绩最高

-- 1、查询任课最多的老师id

-- 2、根据老师id，查找对应的学科id

-- 3、根据学科Id，查询这些学科id成绩最高

-- 4、根据最高成绩，找到对应学科的学生id

-- 5、根据学生Id，查询对应这些学生id的学生姓名

select sname from student where sid in

(

select student\_id from score td, (

select course\_id, max(num) as max\_num from score tc, (

select cid from course ta,

( select teacher\_id, count(\*) from course group by teacher\_id order by count(\*) desc limit 1 ) tb

where ta.teacher\_id = tb.teacher\_id

) td

where tc.course\_id = td.cid

group by course\_id

) te

where td.course\_id = te.course\_id and td.num = te.max\_num

)