作业

-- Table structure for t\_course

DROP TABLE IF EXISTS t\_course ;

CREATE TABLE t\_course (

cid int(11) NOT NULL AUTO\_INCREMENT,

cname varchar(40) DEFAULT NULL,

PRIMARY KEY ( cid )

) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8;

-- Records of t\_course

INSERT INTO t\_course VALUES ('1', 'Java');

INSERT INTO t\_course VALUES ('2', 'C++');

INSERT INTO t\_course VALUES ('3', '高数');

INSERT INTO t\_course VALUES ('4', 'C语言');

INSERT INTO t\_course VALUES ('5', 'Python');

-- Table structure for t\_student

DROP TABLE IF EXISTS t\_student ;

CREATE TABLE t\_student (

sid int(11) NOT NULL AUTO\_INCREMENT,

sname varchar(40) DEFAULT NULL,

age int(11) DEFAULT NULL,

PRIMARY KEY ( sid )

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8;

-- Records of t\_student

INSERT INTO t\_student VALUES ('1', '张三', '18');

INSERT INTO t\_student VALUES ('2', '李四', '22');

INSERT INTO t\_student VALUES ('3', '王五', '26');

-- Table structure for student\_course

DROP TABLE IF EXISTS student\_course ;

CREATE TABLE student\_course (

id int(11) NOT NULL AUTO\_INCREMENT,

sid int(11) DEFAULT NULL,

cid int(11) DEFAULT NULL,

selecttime datetime DEFAULT NULL,

PRIMARY KEY ( id ),

KEY sid ( sid ),

KEY cid ( cid ),

CONSTRAINT student\_course\_ibfk\_2 FOREIGN KEY ( cid ) REFERENCES t\_course ( cid ),

CONSTRAINT student\_course\_ibfk\_1 FOREIGN KEY ( sid ) REFERENCES t\_student ( sid )

) ENGINE=InnoDB AUTO\_INCREMENT=6 DEFAULT CHARSET=utf8;

-- Records of student\_course

INSERT INTO student\_course VALUES ('1', '1', '1', '2018-09-12 13:34:06');

INSERT INTO student\_course VALUES ('2', '1', '2', '2018-09-05 13:34:17');

INSERT INTO student\_course VALUES ('3', '2', '2', '2018-09-06 13:34:22');

INSERT INTO student\_course VALUES ('4', '2', '3', '2018-09-05 13:34:33');

INSERT INTO student\_course VALUES ('5', '2', '4', '2018-09-20 13:34:43');

select count(sid),avg(age),sum(age) from t\_student;

select \* from t\_student where age =(select max(age) from t\_student);

select sname,cname

from t\_student

left join student\_course on t\_student.sid = student\_course.sid

left join t\_course on student\_course.cid = t\_course.cid

WHERE sname = "张三";

select \* from t\_student left join student\_course on t\_student.sid = student\_course.sid

select cname,count(cname)

from t\_student

left join student\_course on t\_student.sid = student\_course.sid

inner join t\_course on student\_course.cid = t\_course.cid

GROUP BY cname

-- 部门表

CREATE TABLE DEPT(

DEPTNO INT PRIMARY KEY, -- 部门编号

DNAME VARCHAR(14) , -- 部门名称

LOC VARCHAR(13) ) ; -- 部门地址

INSERT INTO DEPT VALUES (10,'ACCOUNTING','NEW YORK');

INSERT INTO DEPT VALUES (20,'RESEARCH','DALLAS');

INSERT INTO DEPT VALUES (30,'SALES','CHICAGO');

INSERT INTO DEPT VALUES (40,'OPERATIONS','BOSTON');

-- 员工表

CREATE TABLE EMP(

EMPNO INT PRIMARY KEY, -- 员工编号

ENAME VARCHAR(10), -- 员工名称

JOB VARCHAR(9), -- 工作

MGR DOUBLE, -- 直属领导编号

HIREDATE DATE, -- 入职时间

SAL DOUBLE, -- 工资

COMM DOUBLE, -- 奖金

DEPTNO INT -- 部门号

);

INSERT INTO EMP VALUES

(7369,'SMITH','CLERK',7902,'1980-12-17',800,NULL,20);

INSERT INTO EMP VALUES

(7499,'ALLEN','SALESMAN',7698,'1981-02-20',1600,300,30);

INSERT INTO EMP VALUES

(7521,'WARD','SALESMAN',7698,'1981-02-22',1250,500,30);

INSERT INTO EMP VALUES

(7566,'JONES','MANAGER',7839,'1981-04-02',2975,NULL,20);

INSERT INTO EMP VALUES

(7654,'MARTIN','SALESMAN',7698,'1981-09-28',1250,1400,30);

INSERT INTO EMP VALUES

(7698,'BLAKE','MANAGER',7839,'1981-05-01',2850,NULL,30);

INSERT INTO EMP VALUES

(7782,'CLARK','MANAGER',7839,'1981-06-09',2450,NULL,10);

INSERT INTO EMP VALUES

(7788,'SCOTT','ANALYST',7566,'1987-07-13',3000,NULL,20);

INSERT INTO EMP VALUES

(7839,'KING','PRESIDENT',NULL,'1981-11-17',5000,NULL,10);

INSERT INTO EMP VALUES

(7844,'TURNER','SALESMAN',7698,'1981-09-08',1500,0,30);

INSERT INTO EMP VALUES

(7876,'ADAMS','CLERK',7788,'1987-07-13',1100,NULL,20);

INSERT INTO EMP VALUES

(7900,'JAMES','CLERK',7698,'1981-12-03',950,NULL,30);

INSERT INTO EMP VALUES

(7902,'FORD','ANALYST',7566,'1981-12-03',3000,NULL,20);

INSERT INTO EMP VALUES

(7934,'MILLER','CLERK',7782,'1982-01-23',1300,NULL,10);

--给员工表添加外键约束

ALTER TABLE emp add FOREIGN KEY(deptno) REFERENCES dept(deptno);

select dept.DNAME,dept.DEPTNO

from dept

inner join emp on dept.DEPTNO = emp.DEPTNO

GROUP BY dept.DNAME

select \* from dept

left join emp on dept.DEPTNO = emp.DEPTNO

WHERE sal > (select min(sal) from emp);

select \* from dept

inner join emp on dept.DEPTNO = emp.DEPTNO

WHERE sal > (select avg(sal) from emp);

select job,ename,dname from dept

inner join emp on dept.DEPTNO = emp.DEPTNO

where job = "clerk";

select dept.DEPTNO,min(sal) from dept

left join emp on dept.DEPTNO = emp.DEPTNO GROUP BY dept.DEPTNO

select dept.DEPTNO,dept.DNAME,ename from dept

left join emp on dept.DEPTNO = emp.DEPTNO

where dept.DNAME = "sales"

上课笔记

-- 部门表

create table dept(

dno int primary key ,

dname varchar(30),

address varchar(30)

);

insert into dept values(10,'研发部','北京');

insert into dept values(20,'市场部','上海');

insert into dept values(30,'财务部','武汉');

insert into dept values(40,'运营部','深圳');

-- 员工表

-- 员工表

CREATE TABLE emp (

id INT PRIMARY KEY, -- 员工id

ename VARCHAR(50), -- 员工姓名

mgr INT , -- 上级领导

joindate DATE, -- 入职日期

salary DECIMAL(7,2), -- 工资，

deptno int -- 部门id

);

INSERT INTO emp(id,ename,mgr,joindate,salary,deptno) VALUES

(1001,'孙悟空',1004,'2000-12-17','8000.00',10),

(1002,'卢俊义',1006,'2001-02-20','16000.00',10),

(1003,'林冲',1006,'2001-02-22','12500.00',10),

(1004,'唐僧',1009,'2001-04-02','29750.00',10),

(1005,'李逵',1006,'2001-09-28','12500.00',10),

(1006,'宋江',1009,'2001-05-01','28500.00',10),

(1007,'刘备',1009,'2001-09-01','24500.00',20),

(1008,'猪八戒',1004,'2007-04-19','30000.00',20),

(1009,'罗贯中',NULL,'2001-11-17','50000.00',20),

(1010,'吴用',1006,'2001-09-08','15000.00',20),

(1011,'沙僧',1004,'2007-05-23','11000.00',30),

(1012,'李逵',1006,'2001-12-03','9500.00',30),

(1013,'小白龙',1004,'2001-12-03','30000.00',30),

(1014,'关羽',1007,'2002-01-23','13000.00',null);

-- 创建学生表

CREATE TABLE student(

sid INT PRIMARY KEY AUTO\_INCREMENT,

sname VARCHAR(50)

);

-- 创建课程表

CREATE TABLE course(

cid INT PRIMARY KEY AUTO\_INCREMENT,

cname VARCHAR(20)

);

-- 创建中间表

CREATE TABLE s\_c\_table(

sno INT,

cno INT

);

-- 在sql语句里，''和""并没有区别，但是在代码里，字符串都是用双引号里括起来的

-- 外面双引号，里面单引号，外双内单

-- 多表关联查询（面试时必问）

select \* from student,course;-- 错误的写法

-- 关键字 sql join on

-- sql join 一共有7种，只学习5种

select \* from dept;

select \* from emp;

-- 内连接 取两个表的交集（两个表中都存在）

-- INNER JOIN 取两个表的交集

select \* from a表 INNER JOIN b表 on a.主键 = b.外键

select \* from dept INNER JOIN emp on dept.dno = emp.deptno

-- LEFT JOIN 以左边的表为主，只要左边的表存在的就都要

select \* from dept left JOIN emp on dept.dno = emp.deptno

-- RIGHT JOIN 以右边的表为主，只要右边的表存在的就都要

select \* from dept right JOIN emp on dept.dno = emp.deptno

-- 只要a表里面存在，他们的交集不要了（去除a表中公共部分后剩下的）

select \* from dept left JOIN emp on dept.dno = emp.deptno where emp.deptno is null

-- 只要b表里面存在，他们的交集不要了（去除a表中公共部分后剩下的）

select \* from dept right JOIN emp on dept.dno = emp.deptno where dept.dno is null

INNER JOIN

LEFT JOIN

RIGHT JOIN

LEFT JOIN b.外键 is null

RIGHT JOIN a.主键 is null

-- 三表关联

select \* from dept INNER JOIN emp on dept.dno = emp.deptno INNER JOIN c on dept.dno(主键) = c.aid（外键）

-- 子查询

WHERE where后面的sql语句查询语句只能是一个值

select max(salary) from emp -- 50000

select \*from emp where salary = (select max(salary) from emp)

-- from EXISTS 看笔记自学

-- 事务

-- ACID(具体内容看笔记)事务的四大特性

-- a 原子性

-- c 一致性

-- i 隔离性

-- d 持久性 数据库中数据是持久化的，即使服务器宕机了，数据库中的数据也不会受到影响

-- begin -- 开启事务

-- commit 提交事务 or rollback 回滚事务

-- 索引是什么？索引就像目录

-- 加上索引，就能加快查询速度，避免慢sql

-- 存在什么场景，需要一页一页的查找数据，直到找到这条数据

-- 主键索引

-- 不管是 update delete select 在 where 后面都最好是用id来操作呢？因为id就是主键索引，查询快

-- 索引分类：普通索引，主键索引：复合索引（也叫多列索引）

SHOW INDEX FROM emp;

-- 创建普通索引

create index index\_ename on emp(ename)

create index 索引名字 on 表名(字段)

-- 创建复合索引

create index index\_ename\_mgr on emp(ename,mgr)

-- 加入索引的场景

-- 1.where后面用的比较多的字段，都要加入索引

-- 2.sql join 的字段，要加索引，多表查询没有单表查询快，最好用主键和外键查询你，外键加个索引

-- 3.慢 sql(select \* from 表名 like '%xxx%')实际工作中写这种sql，会导致服务器中堆积大量sql，服务器来不及处理

-- 可以将select \* from 表名 like 'xxx%'

-- 索引失效的内容

-- 1.like 前面不要加百分号

-- 2.select 不要用\* 用id，name,age,class,sex用\*就会全表扫描

-- 3.select 避免使用字段计算 age-10禁止

-- 查询日期 或者时间类型，还有字符类型，必须加上''不加单引号会走mysql底层优化器，会影响性能