07-08-2
2.

09-10-3
2.(1){c3,d3}
   (2){a1,b1,c1,d1},{a1,b1,c2,d2},{a2,b2,c1,d1},{a2,b2,c2,d2},
      {a3,b3,c1,d1},{a3,b3,c2,d2}
   (3)R ÷ S = {a1,b1},{a2,b2},{a3,b3}
      R ÷ S = {m|m ∈ {t[12]|t ∈ r} AND ¬ (m ∈ {t[12]|t ∈ { {k|k ∈ {h[1234]|h[12] ∈
{t[12]|t ∈ r} AND h[34] ∈ s}} AND ¬(k ∈ r)}})}

3.
(1) SELECT sname FROM Sailors, Boats, Reserves WHERE
      Sailors.sid = Reserves.sid AND Boats.bid = Reserves.bid
      AND Reserves.bid > 103 AND Boats.color = '蓝'
(2) SELECT bname FROM Boats WHERE color = '蓝' AND bid IN
      (SELECT bid FROM Reserves GROUP BY bid HAVING COUNT(DISTINCT sid) = 1)
(3)   SELECT sid FROM Reserves
      WHERE bid IN (SELECT bid FROM Boats WHERE color = '蓝')
      GROUP BY sid
      WHERE COUNT(DISTINCT bid) = (SELECT COUNT(*) FROM Boats WHERE color =
'蓝')
(4) SELECT Y.master FROM Sailor, Reserves
WHERE Sailor.sid = Reserves.sid
GROUP BY master
Having COUNT(DISTINCT bid) >= all(
SELECT count(DISTINCT bid) FROM Sailor, Reserves
WHERE Sailor.sid = Reserves.sid
GROUP BY master
)

7.
调度 1 17 错误调度 R2(A,x) 与 W1(A,x-1)为冲突操作，不能交换执行顺序
调度 2 19 错误调度 R2(A,x),W2(A,x-3) 与 W1(A,x-1)都为冲突操作，不能交换执行
顺序
调度 3 16 正确调度
只有调度为可串行化调度时才正确，即冲突操作的顺序不允许更改

8.
CREATE TRIGGER insert_on_sailors
BEFORE INSERT ON Sailors
REFERENCING NEW AS N
FOR EACH ROW
WHEN (EXISTS(SELECT * FROM Sailors GROUP BY master HAVING COUNT(*) >= 2))

ROLLBACK;

附加题：
(1) sid+bid
(2) 职工号+零件号，仓库号+零件号

10-11
4.
(1) SELECT 姓名，仓库位置 FROM 仓库管理员，仓库, 采购订单
WHERE 仓库管理员.库管员编号 = 采购订单. 库管员编号
      仓库.库管员编号 = 仓库管理员.库管员编号
AND 服装编号 = '0101'
(2) SELECT 服装编号 FROM 采购订单 GROUP BY 服装编号
HAVING COUNT(DISTINCT 库管员编号) = 1
(3) SELECT 姓名, 库管员编号 FROM 仓库管理员 WHERE 库管员编号 IN
(SELECT 库管员编号 FROM 采购订单 GROUP BY 库管员编号 HAVING
COUNT(DISTINCT 供应商编号) = SELECT COUNT(*) FROM 供应商)
  (4)
SELECT Y.库管员编号, Y. 订货日期, Y.应到货日期 FROM 采购订单 Y,
(SELECT 库管员编号,MAX(a) b, c FROM (SELECT 库管员编号, SUM(服装数量) a,
MAX(服装数量) c 采购订单 FROM GROUP BY 库管员编号)) T
WHERE Y.库管员编号 = T.库管员编号 AND Y.服装数量 = T.c

14-15-3
3.(1) 客人 关系模式中 主键为身份证号
     房间 关系模式中 主键为房间号
     住宿 关系模式中 主键为（身份证号，房间号，入住时间）
(2) A. SELECT 客人.身份证号 FROM 客人, 住宿
            WHERE 客人.身份证号 = 住宿. 身份证号
                  AND 籍贯 = '南京' AND 房间号 = '301'
   B. SELECT 身份证号 FROM 住宿 GROUP BY 身份证号
        HAVING COUNT(DISTINCT 房间号) = (SELECT COUNT(*) FROM 房间)
   C. SELECT 姓名, T.b FROM 客人 X,
    (SELECT 身份证号, COUNT(房间号) b FROM 住宿 WHERE YEAR(入住时间) =
'2014' GROUP BY 身份证号
        HAVING COUNT(房间号) >=
                 all(SELECT COUNT(*) FROM 住宿 WHERE YEAR(入住时间) = '2014'
GROUP BY 身份证号))T
     WHERE X.身份证号 = T.身份证号
4.

1.SELECT SNAME FROM STUDENT WHERE SNO NOT IN
    (SELECT SNO FROM SC WHERE CNO = 'CS110')

2.SELECT SNAME FROM STUDENT WHERE SEX = '女' AND SNO IN

 (SELECT SNO FROM SC WHERE CNO LIKE 'EE%')

3.SELECT SNO, COUNT(*), MAX(GRADE), MIN(GRADE), AVG(GRADE) FROM SC GROUP BY SNO

4.SELECT SNAME FROM STUDENT WHERE SNO IN

 (SELECT SNO FROM SC WHERE GRADE > 90 AND CNO IN

  （SELECT CNO FROM COURSE WHERE SEMESTER = '秋'）

 GROUP BY SNO HAVING COUNT(*) > 2)

5.SELECT SNAME,CNO FROM STUDENT,SC WHERE STUDENT.SNO = SC.SNO AND NOT EXISTS

(SELECT * FROM SC,(SELECT CNO, AVG(GRADE) AS AVGGRADE FROM SC GROUP BY CNO) AS X

   WHERE STUDENT.SNO = SC.SNO AND SC.CNO = X.CNO AND SC.GRADE < X.AVGGRADE)

15-16-3

2.course 表的主键 cno+dname, 外键

 Student 表的主键 sid,

 Enroll 表的主键 sid+cno+dname1, 外键 sid,cno+dname1

3.

(1) SELECT sid, sname FROM Student WHERE sid IN

(SELECT sid FROM Enroll WHERE grade > 3 AND dname1 = 'Computer Sci'

(2) SELECT cno,dname1 FROM Enroll GROUP BY cno, dname1 HAVING COUNT(sid) = 1

(3) SELECT cno, AVG(grade) FROM Enroll

WHERE dname1 = 'Compuer Sci' GROUP BY cno

(4) SELECT sname, sid FROM Student WHERE sid IN

(SELECT sid FROM Enroll WHERE dname1 = 'Computer Sci' GROUP BY sid

HAVING COUNT(*) =

(SELECT COUNT(*) FROM course WHERE dname = 'Computer Sci'))

(5)

SELECT a, sid FROM Enroll, (

SELECT cno, MAX(grade) a FROM Enroll WHERE dname1 = 'Computer Sci'

GROUP BY cno) T

WHERE Enroll.grade = a AND Enroll.cno = T.cno AND dname1 = 'Computer Sci'

8.Create trigger insert_on_enroll

 After Insert On enroll

 Referencing NEW As N

 For Each Statement

 Insert Into failedcourse

 Select sid, grade, cno, dname1, sectno, pname, dname2

 From N

 Where N.grade < 3.0