2. SELECT instructor.ID, `name`, COUNT(course\_id, sec\_id, semester, `year`)AS total FROM instructor LEFT OUTER JOIN teaches ON instructor.ID = teaches.ID GROUP BY instructor.ID;
3. SELECT ID, `name`, (SELECT COUNT(\*) FROM teaches WHERE teaches.ID = instructor.ID) AS num FROM instructor;
4. SELECT course\_id, sec\_id, CASE

WHEN `name` IS NULL THEN

"-"

ELSE

`name`

END AS `name`

FROM section NATURAL LEFT OUTER JOIN teaches NATURAL LEFT OUTER JOIN instructor WHERE semester = "Spring" && `year` = 2010;

1. SELECT dept\_name, COUNT(ID) AS num FROM department NATURAL LEFT OUTER JOIN instructor GROUP BY dept\_name;
3. (SELECT \* FROM student NATURAL JOIN takes) UNION (SELECT ID, `name`, dept\_name, tot\_cred, NULL, NULL, NULL, NULL, NULL FROM student WHERE ID NOT IN (SELECT student.ID FROM student JOIN takes WHERE student.ID = takes.ID));
4. (SELECT \* FROM student NATURAL JOIN takes) UNION (SELECT ID, `name`, dept\_name, tot\_cred, NULL, NULL, NULL, NULL, NULL FROM student WHERE ID NOT IN (SELECT student.ID FROM student JOIN takes WHERE student.ID = takes.ID)) UNION (SELECT ID, NULL, NULL, NULL, course\_id, sec\_id, semester,`year`, grade FROM takes WHERE ID NOT IN (SELECT student.ID FROM student JOIN takes WHERE student.ID = takes.ID));
6. (a,b) (b1,c)(b,d)
7. 不可能，因为c为null时只有当r中的b在s与t左外连接（此时c必不为null）中找不到对应的值，那么c，d就会同时为空。
8. 不做
9. 不做
10. 不做

CREATE TABLE employee

(

employee\_name VARCHAR(15) PRIMARY KEY,

street VARCHAR(30),

city VARCHAR(15)

);

CREATE TABLE company

(

company\_name VARCHAR(15) PRIMARY KEY,

city VARCHAR(15)

);

CREATE TABLE works

(

employee\_name VARCHAR(15) PRIMARY KEY,

company\_name VARCHAR(15),

salary DECIMAL(5),

FOREIGN KEY(employee\_name) REFERENCES employee(employee\_name),

FOREIGN KEY(company\_name) REFERENCES company(company\_name)

);

CREATE TABLE manages

(

employee\_name VARCHAR(15) PRIMARY KEY,

manager\_name VARCHAR(15),

FOREIGN KEY(employee\_name) REFERENCES employee(employee\_name)

);

2. WITH yi(i, ci, t, s, y, r) AS (SELECT ID, course\_id, time\_slot\_id, semester, `year`, room\_no FROM teaches NATURAL JOIN section) SELECT ID, course\_id, sec\_id, time\_slot\_id, semester, `year`, room\_no FROM teaches NATURAL JOIN section WHERE EXISTS (SELECT i FROM yi WHERE i = ID AND ci = course\_id AND t = time\_slot\_id AND s = semester AND y = `year` AND r != room\_no);
3. CREATE assertion a CHECK NOT EXISTS (WITH yi(i, ci, t, s, y, r) AS (SELECT ID, course\_id, time\_slot\_id, semester, `year`, room\_no FROM teaches NATURAL JOIN section) SELECT ID, course\_id, sec\_id, time\_slot\_id, semester, `year`, room\_no FROM teaches NATURAL JOIN section WHERE EXISTS (SELECT i FROM yi WHERE i = ID AND ci = course\_id AND t = time\_slot\_id AND s = semester AND y = `year` AND r != room\_no));
4. 不做。
6. 不做。
8. SELECT employee\_name FROM employee NATURAL LEFT OUTER JOIN manages WHERE manager\_name IS NULL;
9. WITH yi(e, m) AS (SELECT employee\_name, manager\_name FROM manages) SELECT employee\_name FROM employee WHERE employee\_name IN (SELECT e FROM yi WHERE e = employee\_name AND m IS NULL);

第一种情况是title本身的值为null，第二是一个学生没有参加任何课程（假设title本身不为null），因为takes表中cours\_id参照了course表，所以两表相连不会出现title为null的情况，只能是takes表中course\_id为null时才会发生title为空，但其为takes表的主键，那么只能是学生没参加任何课程才会发生这样的情况。



CREATE VIEW s\_cy AS SELECT `year`, SUM(credits) AS num\_credits FROM takes NATURAL JOIN course GROUP BY `year`;

1. 不做。
3. FOREIGN KEY(name) REFERENCES salaried\_worker(name) OR hourly\_worker(name)
4. 要强制执行此操作，那么每次插入数据之前都会进行数据的检查，如果在上述两表中未找到插入的name数据，就会插入失败。
5. 假设是由用户Satoshi完成授权，那么当该用户的权利被撤销时，其授予其它用户的相应权利也将被回收，而其它用户需要该权利的话又会再次的赋予，很麻烦，这种级联撤销还有可能影响到整个系统的发生，所以不能是用户Satoshi赋予权利。然而如果是由经理角色来授权，当Satoshi的权利被撤销时，其它用户由Satoshi赋予的相对应的权利不会受影响，就不会导致这种级联撤销，那么由于级联撤销所导致的问题也就不会出现。