(1)   (a)   2NF   (b)   BCNF   (c)   1NF   (d)   3NF

(2) For example:

```
#      Get command line arguments
dept_code = 'ITEC'
if len(sys.argv) > 1:
    dept_code = sys.argv[1]

query = '''
SELECT DISTINCT s.stuId,
    concat(s.fname, ' ', s.lname) AS student,
    count(e.classId) AS n_classes
FROM student s LEFT JOIN enroll e ON (s.stuId = e.stuId)
WHERE s.major = %s
GROUP BY s.stuId, student
'''

cursor.execute(query,(dept_code))

result = ''
for (sid, student, n_classes) in cursor:
    result = result + ('[sid: ' + str(sid) + '] ' + student
        + ': enrolled in ' + str(n_classes) + ' classes.\n')

#      Print course result.
print('Major students of the department ' + dept_code  + ':')
print('----------------------------------------------------------------')
print(result)
```

(3) For example:

```
-- 1. List information (classId, course name, semester, year,
-- instructor faculty id, and faculty's department code) taught
-- by a 'Professor' in the following manner.
--
SELECT DISTINCT c.classId,
    co.name AS course,
    c.semester,
    c.year,
    c.facId AS `Instructor facId`,
    f.deptCode AS department
FROM class c INNER JOIN course co ON (c.courseId = co.courseId)
    INNER JOIN faculty f ON (c.facId = f.facId)
WHERE f.`rank` = 'Professor';

-- 2. Course names that 'Tony Hawk' enrolled in.
SELECT DISTINCT co.name AS course
FROM student s INNER JOIN enroll e ON (s.stuId = e.stuId)
    INNER JOIN class c ON (e.classId = c.classId)
    INNER JOIN course co ON (c.courseId = co.courseId)
WHERE s.fname = 'Tony' AND s.lname = 'Hawk';
```

```
-- 3. List the number of majors per department and school
-- in the following manner.
SELECT d.deptName AS department,
    sc.schoolName AS school,
    count(stuId) AS `number of majors`
FROM student s INNER JOIN department d ON (s.major = d.deptCode)
    INNER JOIN school sc ON (d.schoolCode = sc.schoolCode)
GROUP BY department, school;

-- 4. List the CSE's department names with their number of
-- faculty members in descending order of this number.
-- Only list the CSE's department with 2 or more faculty members.

SELECT d.deptName AS department,
    count(f.facId) AS n_faculty
FROM faculty f INNER JOIN department d ON (f.deptCode = d.deptCode)
WHERE d.schoolCode = 'CSE'
GROUP BY department
ORDER BY n_faculty DESC;

-- 5. List the classId of every class with both CSCI and ITEC (major)
-- students enrolled in.

SELECT DISTINCT e1.classId
FROM student s1 INNER JOIN enroll e1 ON (s1.stuId = e1.stuId)
    INNER JOIN enroll e2 ON (e1.classId = e2.classId)
    INNER JOIN student s2 ON (e2.stuId = s2.stuId)
WHERE s1.major = 'CSCI'
AND s2.major = 'ITEC';
```

(4)

(a)	T	(b)	T	(c)	F	(d)	T	(e)	T
(f)	F	(g)	T	(h)	F	(i)	F	(j)	T

(5)   R(A,B,C,D) with F = {C->D, CB->A, D->B}

F is equivalent to F' = {C->AD, D->B}

(a)   A+ = A, B+=B, C+=ABCD, D+=BD
(b)   CK: C
(c)   2NF, as D->B violates 3NF.
(d)   R1(A,C,D) and R2(B,D)

(6)   (a)

ISBN -> title, publisherId, pubName
authorId -> authLname, authFname
publisherId -> pubName
pubName -> publisherId

(b)   1 CK: ISBN, authorId

(c)   1NF since AuthorId -> authFname violates 2NF (AuthorId is a part of a CK and authFname is non-prime attribute.)