

**ITEC 3335 Database Development**  
**Fall 2017**  
**Suggested Solution to Final Examination**

(1)

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

(2) (a)

SupplierId -> SupplierName  
ProductId -> ProductDesc  
SupplierId, ProductId, ArrivalTime -> Quantity

(b) {SupplierId, ProductId, ArrivalTime}

(c) SupplierName, ProductDesc, Quantity

(d) 1NF as SupplierId -> SupplierName since SupplierId is a part of a candidate key and Quantity is non-prime.

(e)

Supplier(SupplierId, SupplierName) {SupplierId -> SupplierName}  
Product(ProductId, ProductDesc) {ProductId -> ProductDesc}  
Supply(SupplierId, ProductId, ArrivalTime, Quantity) {SupplierId, ProductId, ArrivalTime -> Quantity}

(3)

```
-- (a) List the names of faculty members and the numbers of classes
--      they teach (not including those not teaching a class).
--
```

```
select f.lname, f.fname, count(c.classId) as numClasses
from faculty f, class c
where f.facId = c.facId
group by f.facId, f.lname, f.fname
order by numClasses desc;
```

```
-- a version that includes faculty not teaching any class.
```

```
select f.lname, f.fname,
       sum(if(c.classId is null, 0, 1)) as numClasses
from faculty f left join class c on(f.facId = c.facId)
group by f.facId, f.lname, f.fname
order by numClasses desc;
```

```
--
-- (b) List the names of faculty members and the numbers of advisees.
--      Include only assistant professors with more than one advisees
--      in the result.
--
```

```

select f.lname, f.fname, count(stuId) as numAdvisees
from faculty f, student s
where f.facId = s.advisor
and f.rank = 'Assistant Professor'
group by f.facId, f.lname, f.fname
having numAdvisees > 1
order by numAdvisees desc;

-- (c) List the faculty names who teache at least one class
-- but do not advise any students.
--
select distinct concat(f.fname, ' ', lname) as faculty,
               count(c.classId) as numClasses
from faculty f join class c on (f.facId = c.facId)
where f.facId not in
      (select distinct advisor from student)
group by faculty
order by numClasses desc;

-- (d) List the names of students who have enrolled in at least one
-- class taught by faculty with id 1011 and also at least one class
-- taught by faculty with id 1012.
select distinct s.fname, s.lname
from student s, enroll e1, enroll e2, class c1, class c2
where s.stuId = e1.stuId
and s.stuId = e2.stuId
and e1.classId = c1.classId
and e2.classId = c2.classId
and c1.facId = 1011
and c2.facId = 1012;

```

(4) For example:

```

#      Get command line argument of product line id.
if len(sys.argv) > 1:
    major = sys.argv[1]
else:
    major = 'CSCI'

cursor = cnx.cursor()
query = """
select s.stuId, concat(s.fname, ' ', s.lname) as student,
       concat(f.fname, ' ', f.lname) as advisor
from student s, faculty f
where s.advisor = f.facId
and s.major = %s
order by s.lname;
"""

cursor.execute(query,(major,))

#      Print product report.
print('Student in the major ' + major + ':')
print('-----')

```

```
for (sid, student, advisor) in cursor:  
    print(student + ' [id #' + str(sid) + ']: advisor: '  
          + advisor + '.')
```

(5)

```
Child(ChildId, ChildFName, ChildLName, JoinSince):  
    {ChildId-> ChildFName, ChildLName, JoinSince}  
    BCNF
```

```
Activity(ActivityId, Name, Date):  
    {ActivityId->Name, Date}  
    BCNF
```

```
Score(ChildId, ActivityId, Score):  
    {ChildId, ActivityId -> Score}  
    Score
```