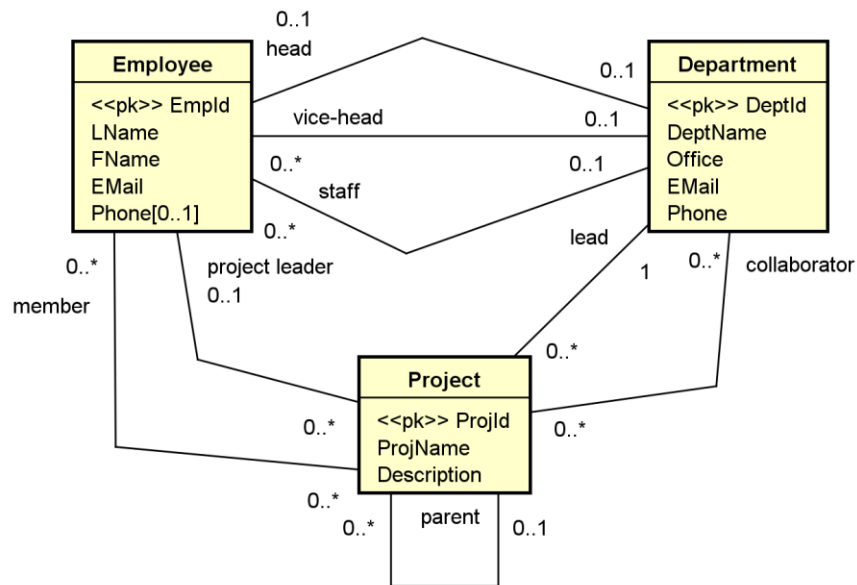


**DASC 5333 Database Systems for Data Science**  
**CSCI 4333 Design of Database Systems**  
**Spring 2023**  
**Suggested Solution to Section 1 Mid-Term Examination**

(1) For example:



(2) For example:

<b>Relation</b>	P( <u>A</u> , B, C, RId)	<b>Relation</b>	Q( <u>QId</u> , RId)
[CK] (1) A, (2) C [FK] (1) RId references R(RId)  [Nullable] B [Non-nullable] A, C, RId [Note]		[CK] (1) QId [FK] (1) RId references R(RId)  [Nullable] [Non-nullable] QId, RId [Note] (1) QId is created as the surrogate primary key.	
<b>Relation</b>	R(RId, D, E, Y_RId, Z_RId)	<b>Relation</b>	U( <u>UId</u> , A, QId)
[CK] (1) RId [FK] (1) Y_RId references R(RId), (2) Z_RId references R(RId)  [Nullable] Y_RId, Z_RId [Non-nullable] RId, D, E [Note] (1) RId is created as the surrogate primary key.		[CK] (1) UId, (2) {A, QId} [FK] (1) A references P(A), (2) QId references Q(QId)  [Nullable] [Non-nullable] UId, A, QId [Note] (1) UId is created as the surrogate primary key.	
<b>Relation</b>	QF( <u>QF_Id</u> , QId, F)	<b>Relation</b>	
[CK] (1) QF_Id, (2) {QId, F}		[CK]	

[FK] (1) QId references Q(QId)  [Nullable] [Non-nullable] QF_Id, QId, F [Note] (1) QId is created as the surrogate primary key.	[FK]  [Nullable] [Non-nullable] [Note]
---	--

(3)

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

(4)

- (a) Minimum: 1 (e.g. ABC); maximum: 3 (e.g. A, B, and C).
- (b) Minimum: 3 (the two CKs each have (n-1) attributes)  
Maximum:  $3 * 2^{n-2}$  (the two CKs are both simple, i.e., each with one attribute)

(6)

(a)

```
SELECT DISTINCT CONCAT(co.rubric, ' ', co.number) AS course,
    c.classId,
    CONCAT(s.fname, ' ', s.lname) AS student,
    e.grade
FROM course AS co INNER JOIN class AS c USING (courseId)
    INNER JOIN enroll AS e USING (classId)
    INNER JOIN student AS s USING (stuld)
WHERE co.rubric = 'CSCI';
```

(b)

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

(c)

```
SELECT DISTINCT s.stuld,
    CONCAT(s.fname, ' ', s.lname) AS student
```

```
FROM student AS s
WHERE stuld NOT IN
    (SELECT e.stuld
     FROM enroll AS e INNER JOIN class AS c ON (e.classId = c.classId)
      INNER JOIN course AS co ON (c.courseId = co.courseId)
      WHERE co.rubric = 'CSCI');
```

(d)

```
SELECT DISTINCT CONCAT(co.rubric, ' ', co.number) AS course,
    co.title,
    IFNULL(COUNT(e.stuld), 0) AS `accumulative number of students`
FROM course AS co LEFT JOIN class AS c USING (courseId)
    LEFT JOIN enroll AS e USING (classId)
GROUP BY course, title;
```