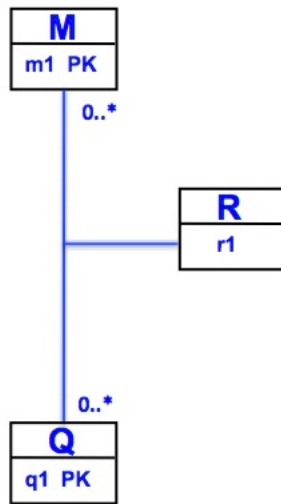


## Q9 UML Key

### QUESTION 1

Consider the UML snippet, in which a class M is associated with a class Q, through an association R (also represented here by an association class of the same name). m1 is the primary key of M, and q1 is the primary key of Q. Identify each option that is an equivalent SQL translation of the UML snippet. The attribute types are not relevant to this exercise and are not included (do not consider this omission an error). Note that there are multiple declarations per line, so look carefully.



Correct answer (option 5):

```
CREATE TABLE M (  
    m1, PRIMARY KEY (m1)  
)
```

```
CREATE TABLE Q (  
    q1, PRIMARY KEY (q1)  
)
```

```
CREATE TABLE R (  
    r1, m1, q1, PRIMARY KEY (m1, q1),  
    FOREIGN KEY (m1) REFERENCES M,  
    FOREIGN KEY (q1) REFERENCES Q  
)
```

Wrong answers

1) This choice includes an assertion that enforces a 1..\* cardinality (aka multiplicity) constraint.

```
CREATE TABLE M (  
  m1, PRIMARY KEY (m1)  
)
```

```
CREATE TABLE Q (  
  q1, PRIMARY KEY (q1)  
)
```

```
CREATE TABLE R (  
  r1, m1, q1, PRIMARY KEY (m1, q1),  
  FOREIGN KEY (m1) REFERENCES M,  
  FOREIGN KEY (q1) REFERENCES Q  
)
```

```
CREATE ASSERTION MparticipatesQ  
CHECK (NOT EXISTS  
      (SELECT *  
       FROM M  
       WHERE M.m1 NOT IN (SELECT m1 FROM R)))
```

2) This choice doesn't include foreign key constraints in R

```
CREATE TABLE M (  
  m1, r1, PRIMARY KEY (m1)  
)
```

```
CREATE TABLE Q (  
  q1, r1, PRIMARY KEY (q1)  
)
```

```
CREATE TABLE R (  
  r1, m1, q1, PRIMARY KEY (m1, q1)  
)
```

3) R doesn't even include the key attributes of M and Q

```
CREATE TABLE M (  
  m1, r1, PRIMARY KEY (m1)  
)
```

```
CREATE TABLE Q (  
  q1, r1, PRIMARY KEY (q1)  
)
```

```
CREATE TABLE R (  
  r1, PRIMARY KEY (r1)  
)
```

4) The in-table CHECK statements enforce 1..\* cardinality constraints, albeit imperfectly

```
CREATE TABLE M (  
  m1, PRIMARY KEY (m1),  
  CHECK (EXISTS (SELECT R.m1 FROM R WHERE m1 = R.m1))  
)
```

```
CREATE TABLE Q (  
  q1, PRIMARY KEY (q1),  
  CHECK (EXISTS (SELECT R.q1 FROM R WHERE q1 = R.q1))  
)
```

```
CREATE TABLE R (  
  r1, m1, q1, PRIMARY KEY (m1, q1),  
  FOREIGN KEY (m1) REFERENCES M,  
  FOREIGN KEY (q1) REFERENCES Q  
)
```