

**TABLE 4.2** TRC summary of fundamental EMPLOYEE TRAINING queries.

Query	✓TRC
$Q_{\sigma}$	{ E   employee(E) and E.eSalary > 100000 };
$Q_{\pi}$	{ E.eLast, E.eFirst, E.eTitle   employee(E) };
$Q_{\cup}$	managers := { E.eID   employee(E) and E.eTitle='Manager' }; coaches := { E.eID   employee(E) and E.eTitle='Coach' }; { T   managers(T) or coaches(T) };
$Q_{-}$	managers := { E.eID   employee(E) and E.eTitle='Manager' }; takenCourse := { T.eID   takes(T) }; { T   managers(T) and not takenCourse(T) };
$Q_{\times}$	{ E.eID, C.cID   employee(E) and trainingCourse(C) };

**TABLE 4.3** TRC summary of additional EMPLOYEE TRAINING queries.

Query	✓TRC
$Q_{\cap}$	managers := { E.eID   employee(E) and E.eTitle='Manager' }; takenCourse := { T.eID   takes(T) }; { T   managers(T) and takenCourse(T) };
$Q_{\rightarrow}$	{ E, A   employee(E) and technologyArea(A) and E.eID=A.aLeadID };
$Q_{\bowtie}$	{ C.cTitle, T.tYear, T.tMonth, T.tDay   trainingCourse(C) and takes(T) and C.cID=T.cID };

**% Abstract Division Example**

```

%
% abTable(a, b)
%   primary key (a, b)
% bTable(b)
%   primary key (b)
forallExistsEquivalence :=
  { T.a | abTable(T) and not (exists B)( not (not bTable(B) or
    (exists AB) (abTable(AB) and AB.a=T.a and AB.b=B.b) ) ) };

```

**Find Max Value**

%merge more info

```

current_member_temp1:= {T.groupCode, T.artistID, A.lastName, A.yearBorn
|current_member(T) and artist(A) and T.artistID = A.artistID};

```

%Find oldest artist

```

oldest := { T | current_member_temp1(T) and not (exists E) (current_member_temp1(E) and
E.yearBorn < T.yearBorn) };

```