1. **Basic SQL (Section 6.3) — Review basic SQL as needed, then write sample queries that:**
   1. **Use one or more tuple variables (Section 6.3.2).**
   2. **Use one or more of the set operations, e.g., UNION, EXCEPT, INTERSECT (Section 6.3.4).**
2. **Advanced SQL (Sections 7.1.1–7.1.5) — This will include more new material; write sample queries that:**
   1. **Select based on a NULL field value (Section 7.1.1).**
   2. **Implement a nested sub-query, using [NOT] EXISTS, IN, ANY or ALL (Sections 7.1.2–7.1.4).**
   3. **Implement a correlated sub-query (Section 7.1.3).**

-- 1.1. Use one or more tuple variables (Section 6.3.2).

select C.status, P.firstName

from Casting C, Performer P

where C.performerId = 1;

-- 1.2. Use one or more of the set operations, e.g., UNION, EXCEPT, INTERSECT.

select C.movieId from Casting C

UNION ALL

select P.id from Performer P

UNION

select M.id from Movie M;

-- 2.1 Select based on a NULL field value

select \*

from Performer P

where P.lastName is NULL;

-- 2.2 Implement a nested sub-query, using [NOT] EXISTS, IN, ANY, or ALL

select id, firstName, lastName

from Performer P

where P.id IN (select C.PerformerId

from Casting C);

-- 2.3 Implement a correlated sub-query

select id, firstName, lastName,

(select count(\*)

from Casting C

where C.performerId = P.id) MOVIECOUNT

from Performer P;