Domáca úloha — Riešenia.

Databáza je o amerických filmoch:

```
Filmy(názov, režisér, rok)
Obsadenia(názov, herec, plat)
Recenzie(názov, kritik, hodnotenie)
```

Atribúty sú reťazce okrem atribútov rok, plat, a hodnotenie, ktoré sú celé čísla. Kvôli stručnosti a prehľadnosti používajte skrátenú schému:

```
F(n, r, d),
O(n, h, p),
R(n, k, v).
```

Dotazy napíšte tak, aby výsledky neobsahovali duplikáty riadkov, ale nepoužívajte DISTINCT tam, kde je to zbytočné.
Napíšte nasledujúce dotazy:
(a) Nájdite režisérov, ktorí aj hrali (pomocou poddotazu vo WHERE).

```
SELECT DISTINCT r
FROM F
WHERE r IN (SELECT h FROM O);
```

Kalkul: $\exists(u,x,y,z)F(u,r,x) \wedge O(y,h,z)$
Algebra: $\Pi_r F \cap \Pi_h O$

(b) Nájdite hercov, korí nikdy nerežírovali (bez použitia poddotazu vo WHERE).

```
(SELECT h FROM O)
EXCEPT
(SELECT r FROM F);
```

Úlohu možno splniť aj keď používaná implementácia SQL nepodporuje rozdiel.

```
SELECT h
FROM O LEFT OUTER JOIN F ON h = r
WHERE r IS NULL;
```

Kalkul: $\exists(u,x,y,z)O(y,h,z) \wedge \neg F(u,r,x)$
Algebra: $\Pi_h O - \Pi_r F$

(c) Nájdite názvy všetkých filmov zo 60-tých rokov, kde hrali spolu Dustin Hoffman a Anne Bancroft.

```
SELECT F.n
FROM F, O O1, O O2
WHERE F.n = O1.n
      AND F.n = O2.n
      AND d >= 1960 AND d < 1970 /* d BETWEEN 1960 AND 1969 */
      AND O1.h = "Dustin Hoffman"
      AND O2.h = "Anne Bancroft";
```

Kalkul: $\exists(r,d,p1,p2)F(n,r,d) \wedge O(n,"DH",p1) \wedge O(n,"AB",p2) \wedge d \geq 1960 \wedge d < 1970$
Algebra: $\Pi_n \sigma_{d \geq 1960 \wedge d<1970 \wedge h1="DH" \wedge h2="AB"} (F(n,r,d) \bowtie O(n,h1,p1) \bowtie O(n,h2,p2))$

(d) Bez použitia rozdielu (EXCEPT, -), nájdite všetkých hercov, ktorí hrali vo všetkých filmoch režírovaných Kurosawom. (Herec hral vo všetkých filmoch režírovaných Kurosawom, ak neexistuje film režírovaný Kurosawom, ktorý sa nenachádza medzi filmami v ktorých daný herec hral.)

```
SELECT DISTINCT O1.h
FROM O O1
WHERE  NOT EXISTS
       (SELECT *
        FROM   F
        WHERE  r = "Kurosawa"
              AND  F.n NOT IN
                               (SELECT O2.n
                                FROM O O2
                                WHERE  O2.h = O1.h));
```

Použitie počítacieho argumentu – counting argument (Herec hral vo všetkých filmoch režírovaných "Kurosawom", ak hral v toľkých filmoch režírovaných Kurosawom, koľko ich Kurosawa režíroval.) je možné, má svoje uskalia a sťaží aj logické a algebraické vyjadrenie dotazu.

```
SELECT h
FROM F, O
WHERE  F.r = "Kurosawa" AND O.n = F.n
GROUP BY h
HAVING COUNT(n) = (SELECT COUNT(*) FROM F WHERE F.r = "Kurosawa");
     /* Čo ak dvojroľa v Kurosawovom filme ?*/
```

Kalkul: $\forall(n,d)F(n,"Ku",d) \Rightarrow \exists(p)O(n,h,p)$
Algebra: $\Pi_h(O(n,h,p) - \Pi_h((\Pi_n(\sigma_{r="Ku"}F(n,r,d)) \times \Pi_h O(n,h,p)) - \Pi_{n,h}O(n,h,p))$

(e) Nájdite názvy všetký filmov, ktoré recenzovali aspoň dvaja kritici (s použitím agregačnej funkcie).

```
SELECT n
FROM R
GROUP BY n
HAVING COUNT(*) > 1;
```

(f) To isté ako (e), ale s použitím poddotazu namiesto agregačnej funkcie.

```
SELECT DISTINCT n
FROM R R1
WHERE EXISTS /* R1.n IN */ (SELECT R2.n
                            FROM R R2
                            WHERE R1.k <> R2.k);
```

(g) To isté ako (e), ale bez poddotazu aj bez agregačnej funkcie.

```
SELECT DISTINCT R1.n
FROM R R1, R R2
WHERE R1.n = R2.n
      AND R1.k <> R2.k;
```

Kalkul: $\exists(k1, k2, v1, v2) R(n, k1, v1) \wedge R(n, k2, v2) \wedge k1 \neq k2$
Algebra: $\Pi_n \sigma_{k1 \neq k2} R(n, k1, v1) \bowtie R(n, k2, v2)$

(h) Pomocou agregačnej funkcie zistite názvy všetkých filmov, ktoré majú práve jednu recenziu.

```
SELECT n
FROM R
GROUP BY n
HAVING COUNT(*) = 1;
```

(i) To isté ako (h), ale nepoužite agregačné funkcie.

```
SELECT n
FROM R R1
WHERE NOT EXISTS (SELECT *
                  FROM R R2
                  WHERE R1.k <> R2.k);
```

Iná alternativa

```
SELECT n
FROM R R1
WHERE R1.n NOT IN (SELECT R2.n
                   FROM R R2
                   WHERE R1.k <> R2.k);
```

Kalkul: $\forall(k1, v1, k2, v2) R(n, k1, v1) \wedge (R(n, k2, v2) \Rightarrow k1 = k2$
Algebra: $\Pi_n R - \Pi_n \sigma_{k1 \neq k2}(R(n, k1, v1) \bowtie R(n, k2, v2))$

(j) Pre všetkých kritikov vypočítajte priemerné hodnotenie, ktoré udelili. Vypíšte len tých, ktorých priemerné hodnotenie je väčšie ako 1 a usporiadajte ich zostupne podľa hodnotenia.

```
SELECT k, AVG(v)
FROM R
GROUP BY k
HAVING AVG(v) >= 1;
```

(k) Definujeme hodnotu filmu ako priemerné hodnotenie udelené mu kritikmi. Vypíšte: názov, hodnotenie, a rok pre najvyššie hodnotené filmy každého roku. Filmy majú byť usporiadane podľa hodnotenia zostupne. Pri rovnakom hodntení majú byť novšie filmy uvedné skôr.

```
SELECT   n, d, AVG(v),
FROM     F, R
WHERE    F.n = R.n
GROUP BY n, d
ORDER BY AVG(v) DESC, d DESC;
```

Niekto, mohol pochopiť, že treba vytlačiť iba názvy filmov s maximálnym priemerom za každý rok.

```
SELECT   R1.n, d, AVG(R1.v)
FROM     F F1, R R1
WHERE    F1.n = R1.n
GROUP BY R1.n, d
HAVING AVG(R1.v) >= ALL (SELECT AVG(R2.v)
                         FROM R R2, F F2
                         WHERE R2.n = F2.n AND F2.d = F1.d
                         GROUP BY R2.n , F2.d )
ORDER BY d DESC;
```

(l) Definujte hodnotenie režiséra ako priemerné hodnotenie jeho filmov. Nájdite mená hodnotenia všetkých režisérov, ktorí režírovali filmy s Al Pacinom.

```
SELECT   r, AVG(fv)
FROM     F, (SELECT n, AVG(v) AS fv FROM R GROUP BY n) FA
WHERE    F.n = FA.n
GROUP BY r
HAVING   r IN
             (SELECT r
              FROM   F, O
              WHERE  F.n = O.n
                     AND h = "Al Pacino");
```

Nie každý SQL podporuje poddotaz za FROM v takom prípade je potrebné tento poddotaz naharadiť príkazom:

```
CREATE VIEW FA AS (SELECT n, AVG(v) AS fv FROM R GROUP BY n);
/* Vyžaduje právo modifikovať schéma a zapisovať do databázy. */
SELECT   r, AVG(fv)
FROM     F, FA
WHERE    F.n = FA.n
GROUP BY r
HAVING   r IN
             (SELECT r
              FROM   F, O
              WHERE  F.n = O.n
                     AND h = "Al Pacino");
DROP VIEW FA; /* Inak by view zostalo v databáze. */
```

Poddotaz vo FROM vytvára dočasný pohľad v pracovnom priestora nevyžaduje právo zápisu do databázy.

(m) Je uvedený dotaz spráne riešenie (l)? Zdôvodnite svoju odpoveď.

```
SELECT r, AVG(v)
FROM Filmy F, Recenzie R
WHERE
      F.n IN (SELECT n FROM Obsadenie WHERE h = "Al Pacino")
      AND F.n = R.n;
```

Nie. Počíta priemer filmov a nie režisérov (priemer priemerov).Navyše obsahuje aj syntaktickú chybu. Chýba GROUP BY.

(n) Pre každý rok nájdite herca, ktorý v danom roku najviac zarobil (súčet platov za všetky filmy v ktorých učinkoval).

```
SELECT F1.d, O1.h, SUM(O1.p)
FROM F F1, O O1
WHERE F1.n = O1.n
GROUP BY F1.d, O1.h
HAVING SUM(P) >= ALL (SELECT SUM(O2.p)
                      FROM F F2, O O2
                      WHERE F2.n = O2.n AND F2.d =  F1.d
                      GROUP BY h);
```

Ani vložené GROUP BY nemusí podporovať každá implemetácia SQL. Na druhej strane v ORACLE možno namiesto ALL použiť aj MAX(SUM(O2.p)).