```sql
-- Computation queries should return empty set

-- (1)
-- Focal Length Min = Focal Length Max for Prime Lens

ALTER TABLE camera_lens
ADD CONSTRAINT focal_len_eq CHECK(
  EXISTS(
    SELECT * FROM lens AS l
    WHERE l.focal_len_min = l.focal_len_max AND pid = l.pid
  )
);
-- OR
SELECT * FROM lens AS l
WHERE l.focal_len_min != l.focal_len_max
  AND l.pid IN (SELECT pl.pid FROM prime_lens AS pl);

-- (2)
-- 1 <= Score <= 5 for Evaluation

ALTER TABLE evaluation
ADD CONSTRAINT score_range CHECK (score >= 1 AND score <= 5);
-- OR
SELECT * FROM evaluation AS e WHERE e.score < 1 OR e.score > 5;

-- (3)
-- Only one of {3, 4, 5} for a camera

ALTER TABLE finder_ov
ADD CONSTRAINT ov_xor CHECK (
  NOT EXISTS (SELECT f.fid FROM finder_tv WHERE f.fid = fid)
  AND NOT EXISTS (SELECT f.fid FROM finder_or WHERE f.fid = fid)
);
ALTER TABLE finder_tv
ADD CONSTRAINT tv_xor CHECK (
  NOT EXISTS (SELECT f.fid FROM finder_or WHERE f.fid = fid)
  AND NOT EXISTS (SELECT f.fid FROM finder_ov WHERE f.fid = fid)
);
ALTER TABLE finder_or
ADD CONSTRAINT or_xor CHECK (
  NOT EXISTS (SELECT f.fid FROM finder_ov WHERE f.fid = fid)
  AND NOT EXISTS (SELECT f.fid FROM finder_tv WHERE f.fid = fid)
);

-- (4)
-- At least one evaluation for a product
-- Can not be a constraint as you need pid to insert an evaluation

SELECT p.pid FROM product AS p
WHERE p.pid NOT IN (SELECT e.pid FROM evaluation AS e);

-- (5)
-- At least two camera lenses for a replacable lens camera
-- Can not be a constraint as you need pid to insert lenses

SELECT r1.pid FROM replacable_lens_cam AS r1
WHERE r1.pid NOT IN (
  SELECT r2.pid AS pid FROM camera_lens AS cl1, camera_lens AS cl2, replacable_lens_cam AS
  WHERE cl1.camera_id = cl2.camera_id
  AND cl1.camera_id = r2.pid AND cl1.lens_id != cl2.lens_id
);
```