1 Movies and Ratings Redux

Q1.1 SELECT cid, count(*)
    FROM ratings
    GROUP BY cid
    HAVING count(*) >= ALL(SELECT count(*) FROM ratings GROUP BY cid);

Q1.2 SELECT count(distinct cid)
    FROM ratings;

Q1.3 CREATE VIEW TopMovies
    AS
    SELECT mid, avg(rating) as arating
    FROM ratings
    GROUP BY mid
    ORDER BY avg(rating) DESC
    LIMIT 10;

    SELECT movies.mid, movies.title
    FROM movies, TopMovies
    WHERE movies.mid = TopMovies.mid
    ORDER BY movies.mid;

Q1.4 CREATE VIEW MoviesAvgRating
    AS
    SELECT mid, avg(rating) as arating
    FROM ratings
    GROUP BY mid
    ORDER BY avg(rating) DESC;

    SELECT movies.mid, MoviesAvgRating.arating
    FROM movies LEFT OUTER JOIN MoviesAvgRating
    ON movies.mid = MoviesAvgRating.mid
    ORDER BY movies.mid;

Q1.5 SELECT rating, count(*)
    FROM ratings
    GROUP BY rating
    ORDER BY rating ASC;

Q1.6 SELECT mid, avg(rating)
    FROM ratings
    WHERE timestamp >= '2009-03-01 00:00:00' AND
    timestamp < '2009-04-01 00:00:00'
GROUP BY mid
HAVING count(mid) >= 100
ORDER BY mid;

Q1.7 SELECT DISTINCT r2.cid
FROM ratings AS r1, ratings AS r2
WHERE r1.mid = r2.mid AND
  r1.rating = r2.rating AND
  r1.cid = 19002 AND
  r2.cid != 19002
ORDER BY cid;

Q1.8 CREATE VIEW BobPeers
AS
SELECT DISTINCT r2.cid
FROM ratings AS r1, ratings AS r2
WHERE r1.mid = r2.mid AND
  r1.rating = r2.rating AND
  r1.cid = 19002 AND
  r2.cid != 19002
ORDER BY cid;

CREATE VIEW BobPeerRatings
AS
SELECT mid, avg(rating) as arating
FROM BobPeers, ratings
WHERE BobPeers.cid = ratings.cid AND
  mid NOT IN (SELECT mid
               FROM ratings
               WHERE cid = 19002)
GROUP BY mid
ORDER BY avg(rating) DESC;

SELECT title, movies.mid, year
FROM BobPeerRatings, movies
WHERE BobPeerRatings.mid = movies.mid AND
  BobPeerRatings.arating IN (SELECT max(arating)
                             FROM BobPeerRatings);

Q1.9 CREATE VIEW Top5Movies
AS
SELECT mid
FROM ratings
GROUP BY mid
ORDER BY avg(rating) DESC
LIMIT 5;

CREATE VIEW AvgRatings
AS
SELECT cid, avg(rating) as arating
FROM ratings
WHERE mid IN (SELECT * FROM Top5Movies)
GROUP BY cid;

SELECT cid
FROM AvgRatings
WHERE arating IN (SELECT min(arating) FROM AvgRatings)
ORDER BY cid;

Q1.10 SELECT TTTT.cid
FROM (SELECT cid, year, avg(rating) AS ar
     FROM movies, ratings
     WHERE movies.mid = ratings.mid
     GROUP BY cid, year) TTTT,
     (SELECT cid, max(ar) AS mar
     FROM (SELECT cid, year, avg(rating) AS ar
                FROM movies, ratings
            WHERE movies.mid = ratings.mid
             GROUP BY cid, year) TT
     GROUP BY cid) TTT
WHERE TTT.cid = TTTT.cid AND
      TTT.mar = TTTT.ar AND
      TTTT.year = 2000
ORDER BY TTTT.cid;

Q1.11 We can do the query in many ways.

The following query uses the epoch function provided by Postgres and the fact that there are 86400 seconds in a day. Note that we have to use at time zone 'utc' otherwise Postgres will compute the day wrongly.

CREATE VIEW RatingsPerDay
AS
    SELECT cid, count(*) as activity,
           FLOOR(EXTRACT(epoch FROM timestamp at time zone 'utc') / 86400) AS day
    FROM ratings
    GROUP BY cid, day;

SELECT cid, activity
FROM RatingsPerDay
WHERE activity IN (SELECT max(activity) FROM RatingsPerDay);

Another way to do it is to use the EXTRACT(DAY .... ) function:

CREATE VIEW RatingsPerDay
AS
    SELECT cid, count(*) as activity, EXTRACT(DAY FROM timestamp) as day
    FROM ratings
    GROUP BY cid, day;

SELECT cid, activity
FROM RatingsPerDay
WHERE activity IN (SELECT max(activity) FROM RatingsPerDay)
ORDER BY cid;