SQL Analytic queries

Borja Ruiz Amantegui, Miguel Lobo Martín y Francesca Sallicati.

1. Good comrades: stars in top ten of number of different co-stars they have worked with.

```
SELECT count('X'), act1.actor FROM casts act1 INNER JOIN casts
act2 ON act2.TITLE = act1.TITLE WHERE act2.actor != act1.actor
GROUP BY(act1.actor) ORDER BY count('X') DESC;
```

2. Targeted countries: in order to plan marketing actions, we want to retrieve countries in the highest quartile of avg expenses, though in the lower quartile in number of clients.

```
SELECT * FROM
(WITH countryincome AS (SELECT contracts.country, sum(amount)
expenses, ROW_NUMBER() OVER (ORDER BY sum(amount)DESC) AS
Rownumber FROM INVOICES, CONTRACTS WHERE invoices.contractid =
contracts.contractid GROUP BY(CONTRACTS.COUNTRY) ORDER BY
sum(amount) DESC)
SELECT country, expenses FROM countryincome WHERE expenses <=
(SELECT PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY expenses)
FROM countryincome)) a
INNER JOIN
(WITH countries AS (SELECT count('X') contracts, country,
ROW_NUMBER() OVER (ORDER BY count('X') DESC) AS Rownumber FROM
CONTRACTS GROUP BY(COUNTRY))
SELECT contracts, country FROM countries WHERE Rownumber >=
(SELECT PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY
contracts) FROM countries)) b ON a.country=b.country;
```

3. Total taps for each episode, for each season, and for each tv series in the base, restricted to the month of May/16.

```
SELECT title, season, episode, count('X') "Taps", FROM
TAPS_SERIES WHERE VIEW_DATETIME '%/05/16' GROUP BY ROLLUP
(TITLE, SEASON, EPISODE);
```

4. Sales Boost analysis: 5 dates with greater difference between number of licenses sold and the amount of the inmediate previous day.

```
SELECT * FROM (SELECT rownum r, datetime, counter, counter-
LAG(counter,1)  OVER (PARTITION BY NULL ORDER BY NULL)
difference FROM (SELECT datetime, count(title) counter FROM
((SELECT title, datetime FROM LIC_MOVIES) UNION ALL (SELECT
title, datetime FROM LIC_SERIES)) GROUP BY datetime ORDER BY
datetime)
ORDER BY difference DESC)
WHERE r>1 AND rownum<=5;
```

5. For each type of contract during 2016, accumulated monthly income.

```
SELECT sum(amount), contract_type FROM contracts JOIN invoices
ON (contracts.contractid = invoices.contractid) WHERE year
LIKE '2016' GROUP BY contract_type;
```

6. For each month, movies in the highest decile of views, in the highest decile of facebook likes, and also the highest decile of gross (if sorted in ascendant order, the tenth decile).

```
SELECT month, title, views, gross, movie_facebook_likes FROM (
   SELECT title, month, views
    FROM (SELECT title, month, views, ntile(10) OVER (ORDER BY
views ASC) decile
      FROM (SELECT title, EXTRACT(month FROM view_datetime)
month ,count(*) views
          FROM TAPS_MOVIES
               GROUP BY GROUPING SETS ((title, EXTRACT(month
FROM view_datetime)))))
WHERE decile = 10)

INNER JOIN

(SELECT movie_title, gross
     FROM(SELECT movie_title, gross,ntile(10) OVER (ORDER BY
gross ASC) decile
          FROM MOVIES
     WHERE gross IS NOT NULL)
WHERE decile = 10) gross_table

ON title = movie_title

     INNER JOIN

(SELECT movie_title, movie_facebook_likes
  FROM(SELECT movie_title, movie_facebook_likes,ntile(10)
OVER(ORDER BY movie_facebook_likes ASC) decile
    FROM MOVIES
    WHERE movie_facebook_likes IS NOT NULL)
  WHERE decile = 10) facebook_table
ON gross_table.movie_title = facebook_table.movie_title ORDER
BY month;
```

7. Traffic Peak Week: 7-day period of higher traffic (minutes viewed)

```
SELECT CONCAT(day-6,CONCAT('-', day)) interval, total_traffic
FROM
  (SELECT day, traffic, sum(traffic) OVER (ORDER BY day ROWS 6
PRECEDING) total_traffic FROM
    (SELECT movies_table.day day, movies_table.traffic +
series_table.traffic traffic FROM
       (SELECT EXTRACT(day FROM view_datetime) day,
sum(duration*PCT) traffic
         FROM MOVIES
         INNER JOIN
         TAPS_MOVIES ON title=movie_title GROUP BY EXTRACT(day
FROM view_datetime)) movies_table

    INNER JOIN

    (SELECT EXTRACT(day FROM view_datetime) day,
sum(avgduration*PCT) traffic FROM SEASONS
    INNER JOIN
     TAPS_SERIES ON SEASONS.title=TAPS_SERIES.title GROUP BY
EXTRACT(day FROM view_datetime)) series_table

ON movies_table.day = series_table.day ORDER BY
movies_table.day)
ORDER BY traffic DESC)
WHERE rownum<=1;
```