

Assignment 1
SQL Query Writing

Group 10

Kanchan Chowdhari

Payal Kaur

Purva Khandelwal

Samiksha Mhatre

1. Group 1 Queries

1.1 Query 1

Display each beer's name and style name. A beer should be displayed regardless of whether a style name exists or not.

SELECT

br.beer_name,
st.style_name

FROM

beerdb.beers br

LEFT OUTER JOIN beerdb.styles st
ON (br.style_id=st.style_id);

Output-

| | BEER_NAME | STYLE_NAME |
|----|--------------------------|-------------------------------|
| 1 | Kilt Lifter Scottish Ale | Scotch Ale |
| 2 | India Pale Ale | American-Style Pale Ale |
| 3 | Blonde Bock | Traditional German-Style Bock |
| 4 | Schwarzbier | German-Style Schwarzbier |
| 5 | Czech Lager | (null) |
| 6 | Yule Tide | Belgian-Style Tripel |
| 7 | Pilsner | (null) |
| 8 | Warka Strong | American-Style Lager |
| 9 | Porter | Porter |
| 10 | ESB | (null) |

1.2 Query 2

Display each beer's name, category name, color example, and style name, for all beers that have values for category name, color example, and style name.

SELECT

br.beer_name,
c.category_name,
co.examples,
st.style_name

FROM

beerdb.beers br

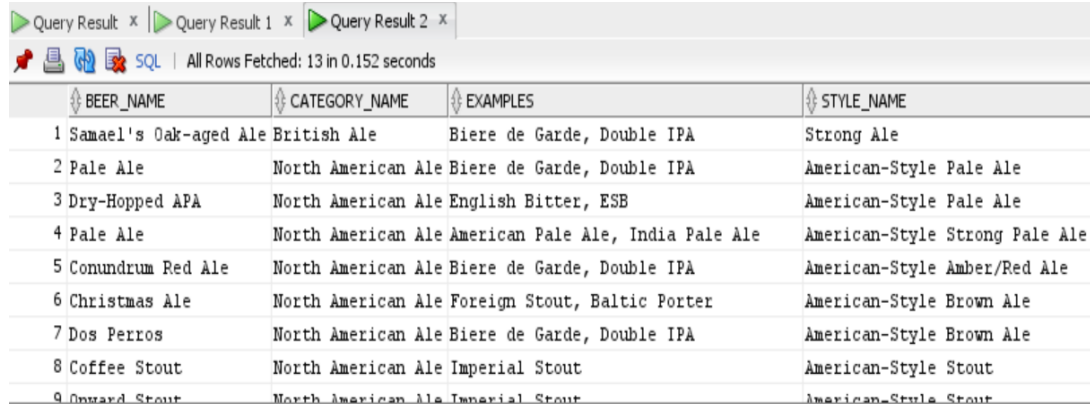
INNER JOIN beerdb.categories c
ON (br.cat_id= c.category_id)

```

INNER JOIN beerdB.colors co
  ON (br.srm= co.lovibond_srm)
INNER JOIN beerdB.styles st
  ON (br.style_id = st.style_id);

```

Output-



| | BEER_NAME | CATEGORY_NAME | EXAMPLES | STYLE_NAME |
|---|-----------------------|--------------------|-----------------------------------|--------------------------------|
| 1 | Samael's Oak-aged Ale | British Ale | Biere de Garde, Double IPA | Strong Ale |
| 2 | Pale Ale | North American Ale | Biere de Garde, Double IPA | American-Style Pale Ale |
| 3 | Dry-Hopped APA | North American Ale | English Bitter, ESB | American-Style Pale Ale |
| 4 | Pale Ale | North American Ale | American Pale Ale, India Pale Ale | American-Style Strong Pale Ale |
| 5 | Conundrum Red Ale | North American Ale | Biere de Garde, Double IPA | American-Style Amber/Red Ale |
| 6 | Christmas Ale | North American Ale | Foreign Stout, Baltic Porter | American-Style Brown Ale |
| 7 | Dos Perros | North American Ale | Biere de Garde, Double IPA | American-Style Brown Ale |
| 8 | Coffee Stout | North American Ale | Imperial Stout | American-Style Stout |
| 9 | Onward Stout | North American Ale | Imperial Stout | American-Style Stout |

1.3 Query 3

Display each brewer's name along with the minimum, maximum, and average alcohol by volume (ABV) of its beers. Exclude any beers with an ABV of zero. Show the brewers with the highest average ABV first.

```

SELECT
  br.brewery_id, bw.name,
  round(min(br.abv),2) as min_abv,
  round(max(br.abv),2) as max_abv,
  round(avg(br.abv),2) as avg_abv
FROM
  beerdB.breweries bw
  INNER JOIN beerdB.beers br
    ON (bw.brewery_id = br.brewery_id)
WHERE br.abv <> 0
GROUP BY br.brewery_id,bw.name
ORDER BY avg_abv desc;

```

Output-

| | BREWERY_ID | NAME | MIN_ABV | MAX_ABV | AVG_ABV |
|----|------------|--|---------|---------|---------|
| 1 | 1373 | Woodforde's Norfolk Ales | 99.99 | 99.99 | 99.99 |
| 2 | 1325 | Vettervö-ö-ö's Alt Heidelberger Brauhaus | 37 | 37 | 37 |
| 3 | 193 | Brasserie Grain D'Orge | 13 | 13 | 13 |
| 4 | 188 | Brasserie Dubuisson | 12 | 13 | 12.5 |
| 5 | 1115 | Schloss Eggenberg | 8.5 | 14 | 12.02 |
| 6 | 731 | Karl Strauss Brewery Gardens - Sorrento Mesa | 12 | 12 | 12 |
| 7 | 1192 | Stadsbrouwerij De Hemel | 12 | 12 | 12 |
| 8 | 276 | Brouwerij De Landtsheer | 10 | 12 | 11 |
| 9 | 1322 | Van Honsebrouch | 11 | 11 | 11 |
| 10 | 966 | Pacific Rim Brewing | 11 | 11 | 11 |
| 11 | 727 | JW Lees and Co (Brewers) Ltd. | 7.3 | 11.6 | 10.92 |

1.4 Query 4

Find which cities would be good for hosting microbrewery tours. A city must have at least 10 breweries to be considered. Display the city's name as well as how many breweries are in the city. Show cities with the most breweries first.

```

SELECT
    bw.city,
    count(bw.brewery_id) as no_of_breweries
FROM
    beerdb.breweries bw
GROUP BY bw.city
HAVING
    count(bw.brewery_id) >=10
    AND bw.city IS NOT NULL
ORDER BY no_of_breweries desc;

```

Output-

| | CITY | NO_OF_BREWERIES |
|---|---------------|-----------------|
| 1 | Portland | 17 |
| 2 | Denver | 13 |
| 3 | Seattle | 13 |
| 4 | San Francisco | 10 |
| 5 | Chicago | 10 |

1.5 Query 5

Display all beer names that (1) belong to a category with a name containing “Lager” somewhere in the name and (2) have an alcohol by volume (ABV) of eight or greater. Show the beer names in alphabetical order.

```

SELECT
    br.beer_name
FROM beerdb.beers br
    INNER JOIN beerdb.categories c
        ON (br.cat_id=c.category_id)
WHERE
    c.category_name like '%Lager%'
    AND br.abv >= 8
ORDER BY br.beer_name;

```

Output-

| | BEER_NAME |
|----|-----------------------------|
| 1 | Andygator |
| 2 | Aventinus Weizen-Eisbock |
| 3 | Baltika #9 |
| 4 | Broken Keg Ice Bock |
| 5 | Butthead Doppelbock |
| 6 | Climax Doppel Bock |
| 7 | Consecrator Doppelbock Beer |
| 8 | Curator Dunkler Doppelbock |
| 9 | Demolition |
| 10 | Detonator Doppelbock |
| 11 | Detonator Doppelbock |

1.6 Query 6

Display the name of all movies that have an IMDB rating of at least 8.0, with more than 100,000 IMDB votes, and were released from 2007 to 2013. Show the movies with the highest IMDB ratings first.

```

SELECT
    mv.film_title
FROM
    relmdb.movies mv
WHERE (mv.imdb_rating >=8
    and mv.imdb_votes > 100000
    and extract(year from mv.release_date) between 2007 and 2013)
ORDER BY mv.imdb_rating desc;

```

Output-

| | FILM_TITLE |
|---|-----------------------|
| 1 | The Dark Knight |
| 2 | Inception |
| 3 | Django Unchained |
| 4 | The Dark Knight Rises |

1.7 Query 7

Display each movie's title and total gross, where total gross is USA gross and worldwide gross combined. Exclude any movies that do not have values for either USA gross or worldwide gross. Show the highest grossing movies first.

```

SELECT
    film_title,
    worldwide_gross + usa_gross as total_gross
FROM
    (SELECT
        mv.film_title,
        mv.worldwide_gross,
        mv.usa_gross
    FROM
        relmdb.movies mv
    WHERE mv.worldwide_gross IS NOT NULL)
WHERE usa_gross IS NOT NULL
ORDER BY total_gross DESC;

```

Output-

| | FILM_TITLE | TOTAL_GROSS |
|---|---|-------------|
| 1 | The Dark Knight | 1539416888 |
| 2 | The Dark Knight Rises | 1529180386 |
| 3 | The Lord of the Rings: The Return of the King | 1497775426 |
| 4 | The Lord of the Rings: The Two Towers | 1268598476 |
| 5 | Star Wars | 1236333672 |
| 6 | The Lord of the Rings: The Fellowship of the Ring | 1187075074 |
| 7 | Inception | 1118108959 |
| 8 | Forrest Gump | 1007078912 |

1.8 Query 8

Display the titles of any movies where Tom Hanks or Tim Allen were cast members. Each movie title should be shown only once.

```
SELECT
  DISTINCT mv.film_title
FROM
  relmdb.movies mv
  INNER JOIN casts ct
    ON (mv.film_id = ct.film_id)
WHERE LOWER(ct.cast_member) IN ('tom hanks','tim allen')
```

| | FILM_TITLE |
|---|---------------------|
| 1 | Saving Private Ryan |
| 2 | Forrest Gump |
| 3 | Toy Story 3 |
| 4 | The Green Mile |

2. Group 2 Queries

2.1 Query 10

Label the strength of a beer based on its ABV. For each beer display the beer's name, ABV, and a textual label describing the strength of the beer. The label should be "Very High" for an ABV more than 10, "High" for an ABV of 6 to 10, "Average" for an ABV of 3 to 6, and "Low" for an ABV less than 3. Show the records by beer name.

```
SELECT
  br.beer_name,
  br.abv,
  (CASE
    WHEN br.abv >10 THEN 'Very High'
    WHEN br.abv BETWEEN 6 AND 10 THEN 'High'
    WHEN br.abv BETWEEN 3 AND 6 THEN 'Average'
    ELSE 'Low'
  END) "Label"
FROM
  beerdb.beers br
ORDER BY
  br.abv DESC,
  br.beer_name ASC
```

| | BEER_NAME | ABV | Label |
|----|-------------------------------------|-------|-----------|
| 1 | Norfolk Nog Old Dark Ale | 99.99 | Very High |
| 2 | Vetter 33 | 37 | Very High |
| 3 | Tactical Nuclear Penguin | 32 | Very High |
| 4 | Samuel Adams Utopias MMIV | 24 | Very High |
| 5 | Raison D'Extra | 20 | Very High |
| 6 | Tokyo* | 18.2 | Very High |
| 7 | 120 Minute IPA | 18 | Very High |
| 8 | Fort | 18 | Very High |
| 9 | World Wide Stout | 18 | Very High |
| 10 | Samuel Adams Triplebock 1994 | 17 | Very High |
| 11 | Mephistopheles Stout | 16.03 | Very High |
| 12 | Venus - Belgian-style Quadrupel | 16 | Very High |
| 13 | Olde School Barleywine | 15.04 | Very High |
| 14 | Black Magick | 15 | Very High |
| 15 | Broken Keg Ice Bock | 15 | Very High |
| 16 | Three Guy Off The Scale Barley Wine | 15 | Very High |
| 17 | The Beast Grand Cru | 14.9 | Very High |
| 18 | Samuel's Oak-aged Ale | 14.5 | Very High |
| 19 | Samuel's Oak-aged Ale | 14 | Very High |

2.2 Query 11

Find all breweries that specialize in a particular beer style. A brewer is considered specialized if they produce at least 10 beers from the same style. Show the brewer's name, style name, and how many beers the brewer makes of that style. Display the records by style name first and then by breweries with the most beers within that style.

```

SELECT
    st.style_name,
    bw.name,
    COUNT(br.beer_name)
FROM
    beerdb.beers br
    INNER JOIN beerdb.styles st
        ON (br.style_id = st.style_id)
    INNER JOIN beerdb.breweries bw
        ON (br.brewery_id = bw.brewery_id)
GROUP BY
    st.style_name,
    bw.name
HAVING COUNT(br.beer_name) >=10

```


| | STYLE_NAME | NAME | COUNT(BR.BEER_NAME) |
|---|-----------------------------------|--------------------------|---------------------|
| 1 | American-Style Lager | Anheuser-Busch | 10 |
| 2 | Imperial or Double India Pale Ale | Midnight Sun Brewing Co. | 11 |
| 3 | American-Style Light Lager | Miller Brewing | 10 |

2.3 Query 12

Display each brewer's name and how many beers they have associated with their brewery. Only include brewers that are located outside the United States and have more than the average number of beers from all breweries (excluding itself when calculating the average). Show the brewers with the most beers first. If there is a tie in number of beers, then sort by the brewers' names.

```

SELECT
    bw.name,
    COUNT(br.beer_id) No_of_beers
FROM
    beerdb.breweries bw
    INNER JOIN beerdb.beers br
        ON (bw.brewery_id = br.brewery_id)
WHERE LOWER(bw.COUNTRY) != LOWER('United States')
GROUP BY
    bw.name
HAVING COUNT(br.beer_id) > (SELECT
        AVG(COUNT(br.beer_id))
        FROM
            beerdb.breweries bw
            JOIN beerdb.beers br
                ON (bw.brewery_id = br.brewery_id)
        GROUP BY
            bw.brewery_id,
            br.beer_name
    )
ORDER BY
    COUNT(br.beer_id) DESC,
    bw.name ASC;

```

| | NAME | NO_OF_BEERS |
|----|---|-------------|
| 1 | De Proef Brouwerij | 17 |
| 2 | Unibroue | 16 |
| 3 | JW Lees and Co (Brewers) Ltd. | 14 |
| 4 | Nøgne Ø - Det Kompromissløse Bryggeri A/S | 14 |
| 5 | Steamworks | 14 |
| 6 | BrewDog Ltd | 12 |
| 7 | Brouwerij Van Steenberge | 12 |
| 8 | Kulmbacher Brauerei AG | 12 |
| 9 | Labatt Ontario Breweries | 12 |
| 10 | Paulaner | 12 |
| 11 | Brasserie De L'Abbaye Des Rocs | 10 |
| 12 | Brasserie-Brouwerij Cantillon | 10 |
| 13 | Niagara Falls Brewing | 10 |
| 14 | SabMiller India, | 10 |
| 15 | Brasserie de Bruneau | 9 |
| 16 | Brouwerij St. Bernardus | 9 |
| 17 | Brouwerij Sterkens | 9 |
| 18 | Coopers Brewery | 9 |
| 19 | Molson Breweries of Canada | 9 |
| 20 | Samuel Smith Old Brewery (Tadcaster) | 9 |
| 21 | Bierbrouwerij De Koningshoeven | 8 |
| 22 | Brouwerij Bavik - De Brabandere | 8 |
| 23 | Fuller, Smith & Turner PBC | 8 |
| 24 | Kiuchi Shuzou Goushi Kaisya | 8 |
| 25 | Ridgeway Brewing | 8 |
| 26 | Bamberger Mahr's-Brau | 7 |
| 27 | Big Ridge Brewing | 7 |
| 28 | Big River Brewing | 7 |
| 29 | Brauerei Beck | 7 |

2.4 Query 13

For each movie display its movie title, year, and how many cast members were a part of the movie. Exclude movies with five or fewer cast members. Display movies with the most cast members first, followed by movie year and title.

```

SELECT
    mv.film_title,
    mv.film_year,
    COUNT(ct.cast_member) AS no_of_cast
FROM
    relmdb.movies mv
JOIN relmdb.casts ct

```

```

    ON (mv.film_id = ct.film_id)
GROUP BY
    mv.film_title,
    mv.film_year
HAVING COUNT(ct.cast_member) >5
ORDER BY
    no_of_cast DESC,
    mv.film_year,
    mv.film_title

```

| | FILM_TITLE | FILM_YEAR | NO_OF_CAST |
|----|--|-----------|------------|
| 1 | Paths of Glory | 1957 | 18 |
| 2 | The Lord of the Rings: The Two Towers | 2002 | 18 |
| 3 | Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb | 1964 | 17 |
| 4 | The Great Dictator | 1940 | 15 |
| 5 | Sunset Blvd. | 1950 | 15 |
| 6 | Seven Samurai | 1954 | 15 |
| 7 | Lawrence of Arabia | 1962 | 15 |
| 8 | To Kill a Mockingbird | 1962 | 15 |
| 9 | A Clockwork Orange | 1971 | 15 |
| 10 | The Godfather | 1972 | 15 |
| 11 | The Godfather: Part II | 1974 | 15 |
| 12 | One Flew Over the Cuckoo's Nest | 1975 | 15 |
| 13 | Star Wars | 1977 | 15 |
| 14 | Apocalypse Now | 1979 | 15 |
| 15 | Star Wars: Episode V - The Empire Strikes Back | 1980 | 15 |
| 16 | Goodfellas | 1990 | 15 |
| 17 | Terminator 2: Judgment Day | 1991 | 15 |
| 18 | Forrest Gump | 1994 | 15 |
| 19 | Léon: The Professional | 1994 | 15 |
| 20 | Pulp Fiction | 1994 | 15 |
| 21 | The Shawshank Redemption | 1994 | 15 |
| 22 | Saving Private Ryan | 1998 | 15 |
| 23 | The Green Mile | 1999 | 15 |

2.5 Query 14

For each genre display the total number of films, average fan rating, and average USA gross. A genre should only be shown if it has at least five films. Any film without a USA gross should be excluded. A film should be included regardless of whether any fans have rated the film. Show the results by genre. (Hint: use the TRIM function to only show a single record from the same genre.)

```

SELECT
    TRIM(gn.genre) genre,
    COUNT(mv.film_title) as total_no_films,
    ROUND(AVG(fr.imdb_rating)) avg_rating,

```

```

ROUND(AVG(mv.usa_gross)) avg_usa_gross
FROM
    relmdb.movies mv
    INNER JOIN relmdb.genres gn
        ON (mv.film_id = gn.film_id)
    LEFT JOIN relmdb.fan_ratings fr
        ON (mv.Film_id = fr.film_id)
WHERE mv.usa_gross IS NOT NULL
GROUP by
    gn.genre
HAVING COUNT(mv.film_title) >=5
ORDER BY
    COUNT(mv.film_title) DESC

```

| | GENRE | TOTAL_NO_FILMS | AVG_RATING | AVG_USA_GROSS |
|----|-----------|----------------|------------|---------------|
| 1 | Drama | 1079 | 7 | 104099479 |
| 2 | Action | 703 | 7 | 232652545 |
| 3 | Thriller | 639 | 7 | 113425811 |
| 4 | Crime | 596 | 7 | 66922252 |
| 5 | Adventure | 512 | 7 | 280058724 |
| 6 | Sci-Fi | 511 | 7 | 239484118 |
| 7 | War | 311 | 7 | 92571740 |
| 8 | Comedy | 213 | 7 | 39647466 |
| 9 | Mystery | 207 | 7 | 160001608 |
| 10 | Romance | 205 | 7 | 220479240 |
| 11 | Fantasy | 193 | 7 | 328838162 |
| 12 | Biography | 130 | 7 | 46836394 |
| 13 | Drama | 130 | 7 | 46836394 |
| 14 | Crime | 130 | 7 | 46836394 |
| 15 | Horror | 101 | 7 | 5019181 |
| 16 | Action | 95 | 7 | 198116802 |
| 17 | Sci-Fi | 95 | 7 | 198116802 |
| 18 | Fantasy | 91 | 7 | 460935665 |
| 19 | Action | 91 | 7 | 460935665 |
| 20 | Adventure | 91 | 7 | 460935665 |

2.6 Query 15

Find the average budget for all films from a director with at least one movie in the top 25 IMDB ranked films. Show the director with the highest average budget first.

```

SELECT
    dr.director,
    ROUND(AVG(mv.budget)) AS Avergae_Budget
FROM
    relmdb.movies mv
    INNER JOIN relmdb.directors dr

```

```

    ON (mv.film_id = dr.film_id)
WHERE mv.imdb_rank <=25
GROUP BY
    dr.director
ORDER BY
    AVG(mv.budget) DESC

```

| | DIRECTOR | AVERAGE_BUDGET |
|----|----------------------|----------------|
| 1 | Christopher Nolan | 172500000 |
| 2 | Peter Jackson | 93666667 |
| 3 | Lana Wachowski | 63000000 |
| 4 | Andy Wachowski | 63000000 |
| 5 | Robert Zemeckis | 55000000 |
| 6 | Steven Spielberg | 25000000 |
| 7 | Frank Darabont | 25000000 |
| 8 | Irvin Kershner | 18000000 |
| 9 | George Lucas | 11000000 |
| 10 | Francis Ford Coppola | 9500000 |
| 11 | Quentin Tarantino | 8000000 |
| 12 | Martin Scorsese | 6368901 |
| 13 | Milos Forman | 4400000 |
| 14 | Akira Kurosawa | 2000000 |
| 15 | Sergio Leone | 1200000 |
| 16 | Sidney Lumet | 350000 |

2.7 Query 16

Find all duplicate fans. A fan is considered duplicate if they have the same first name, last name, city, state, zip, and birth date.

```

SELECT
    fn.fname,
    fn.lname,
    fn.city,
    fn.state,
    fn.zip,
    fn.birth_day,
    COUNT(*)
FROM
    relmdb.fans fn
GROUP BY
    fn.fname,
    fn.lname,
    fn.city,

```

```

fn.state,
fn.zip,
fn.birth_day
HAVING COUNT(*) > 1
ORDER BY
COUNT(*) DESC

```

| | FNAME | LNAME | CITY | STATE | ZIP | BIRTH_DAY | COUNT(*) |
|----|-----------|--------------|--------------|-------|-------|-----------|----------|
| 1 | THEODORE | BUCKLEY | APOLLO BEACH | FL | 33572 | 9 | 13 |
| 2 | NANCY | PEEPLES | DOVER | FL | 33527 | 16 | 5 |
| 3 | MARVIN | BODDEN | SEFFNER | FL | 33584 | 7 | 4 |
| 4 | ROSEMARIE | ADAM | VALRICO | FL | 33594 | 20 | 4 |
| 5 | DORIS | FREY | ZEPHYRHILLS | FL | 33543 | 12 | 3 |
| 6 | GEORGE | MOSHER | GIBSONTON | FL | 33534 | 7 | 2 |
| 7 | GARY | BRIMBLECOMBE | SEFFNER | FL | 33584 | 24 | 2 |
| 8 | ALAN | NOBLES | LITHIA | FL | 33547 | 24 | 2 |
| 9 | ELVIN | HANSEN | VALRICO | FL | 33594 | 22 | 2 |
| 10 | RUSSELL | MORMON | THONOTOSASSA | FL | 33592 | 18 | 2 |
| 11 | JAMES | MOSKE | BRANDON | FL | 33510 | 31 | 2 |
| 12 | DONALD | GODIN | SEFFNER | FL | 33584 | 19 | 2 |
| 13 | DAVID | COLLINS | RIVERVIEW | FL | 33569 | 9 | 2 |
| 14 | PATRICK | JEAN | BUSHNELL | FL | 33513 | 2 | 2 |
| 15 | GENE | NEWMAN | DOVER | FL | 33527 | 18 | 2 |
| 16 | ROBERT | GRIFFIN | DOVER | FL | 33527 | 30 | 2 |
| 17 | KATHLEEN | WEEKLY | RUSKIN | FL | 33570 | 2 | 2 |
| 18 | JOSE | RODRIGUEZ | RUSKIN | FL | 33570 | 12 | 2 |
| 19 | MYRLENE | MUSIC | GIBSONTON | FL | 33534 | 6 | 2 |
| 20 | RUDOLPH | SCHEIBNER | PLANT CITY | FL | 33567 | 22 | 2 |
| 21 | DONALD | LLOYD | PLANT CITY | FL | 33567 | 4 | 2 |
| 22 | FRANK | RAMDATH | DADE CITY | FL | 33525 | 14 | 2 |
| 23 | MICKEY | WILSON | BUSHNELL | FL | 33513 | 19 | 2 |
| 24 | MELVIN | POTTER | BRANDON | FL | 33511 | 12 | 2 |
| 25 | KENNETH | FRANKLIN | PLANT CITY | FL | 33565 | 27 | 2 |
| 26 | ROBERT | WILSON | DADE CITY | FL | 33525 | 3 | 2 |
| 27 | BOB | BONDS | GIBSONTON | FL | 33534 | 4 | 2 |

2.8 Query 17

We believe there may be erroneous data in the movie database. To help uncover unusual records for manual review, write a query that finds all actors/actresses with a career spanning 60 years or more. Display each actor's name, how many films they worked on, the year of the earliest and latest film they worked on, and the number of years the actor was active in the film industry (assume all years between the first and last film were active years). Display actors with the longest career first.

```

SELECT

```

```

ct.cast_member,
COUNT(mv.film_id),
MAX(mv.film_year),
MIN(film_year),
(MAX(mv.film_year) - MIN(film_year)+1) AS active_years
FROM
relmdb.casts ct
INNER JOIN movies mv
ON (ct.film_id = mv.film_id)
GROUP BY
ct.cast_member
ORDER BY
(MAX(mv.film_year) - MIN(film_year)+1) DESC

```

| CAST_MEMBER | COUNT(MV.FILM_ID) | MAX(MV.FILM_YEAR) | MIN(FILM_YEAR) | ACTIVE_YEARS |
|----------------------|-------------------|-------------------|----------------|--------------|
| 1 Sigourney Weaver | 3 | 2008 | 1931 | 78 |
| 2 Ian Holm | 2 | 2001 | 1931 | 71 |
| 3 Harry Dean Stanton | 2 | 1999 | 1931 | 69 |
| 4 Hank Mann | 2 | 1979 | 1940 | 40 |
| 5 Charles Chaplin | 2 | 1979 | 1940 | 40 |
| 6 Robert De Niro | 3 | 1990 | 1957 | 34 |
| 7 Alec Guinness | 3 | 1980 | 1962 | 19 |
| 8 Tom Hanks | 4 | 2010 | 1994 | 17 |
| 9 Frank Sivero | 2 | 1990 | 1974 | 17 |
| 10 Robert Duvall | 3 | 1974 | 1962 | 13 |
| 11 Clotilde Mollet | 2 | 2011 | 2001 | 11 |
| 12 Jeffrey DeMunn | 2 | 1999 | 1994 | 6 |
| 13 Anthony Daniels | 2 | 1980 | 1977 | 4 |
| 14 Mark Hamill | 2 | 1980 | 1977 | 4 |
| 15 David Prowse | 2 | 1980 | 1977 | 4 |
| 16 Peter Mayhew | 2 | 1980 | 1977 | 4 |
| 17 Jack Purvis | 2 | 1980 | 1977 | 4 |
| 18 Harrison Ford | 2 | 1980 | 1977 | 4 |
| 19 Carrie Fisher | 2 | 1980 | 1977 | 4 |
| 20 Kenny Baker | 2 | 1980 | 1977 | 4 |
| 21 Al Pacino | 2 | 1974 | 1972 | 3 |
| 22 John Cazale | 2 | 1974 | 1972 | 3 |
| 23 Diane Keaton | 2 | 1974 | 1972 | 3 |
| 24 John Ratzenberger | 2 | 2010 | 2008 | 3 |
| 25 Talia Shire | 2 | 1974 | 1972 | 3 |
| 26 Sean Astin | 2 | 2002 | 2001 | 2 |
| 27 Sally Field | 2 | 2002 | 2001 | 2 |

3. Group 3 Queries

3.1 Query 19

Assign breweries to groups based on the number of beers they brew. Display the brewery ID, name, number of beers they brew, and group number for each brewery. The group number should range from 1 to 4, with group 1 representing the top 25% of breweries (in terms of number of beers), group 2

representing the next 25% of breweries, group 3 the next 25%, and group 4 for the last 25%. Breweries with the most beers should be shown first. In the case of a tie, show breweries by brewery ID (lowest to highest).

```
SELECT
q1.name,
q1.brewery_id,
No_of_beers,
Rank_of_beers,
(CASE
  WHEN rank_of_beers BETWEEN 0 AND (0.25 * 854) THEN 'Group 1'
  WHEN rank_of_beers BETWEEN (0.25 * 854) AND (0.50 * 854) THEN 'Group 2'
  WHEN rank_of_beers BETWEEN (0.5 * 854) AND (0.75 * 854) THEN 'Group 3'
  ELSE 'Group 4'
END) "GROUP_NAME"
FROM
(
  SELECT
    bw.name,
    bw.brewery_id,
    COUNT(br.beer_id) No_of_beers,
    RANK() Over( ORDER BY COUNT(br.beer_id) DESC) Rank_of_beers
  FROM
    beerdb.breweries bw
    INNER JOIN beerdb.beers br
      ON (bw.brewery_id = br.brewery_id)
  GROUP BY
    bw.name,
    bw.brewery_id
  ORDER BY
    COUNT(br.beer_id) DESC,
    bw.name ASC
)q1;
```


| NAME | BREWERY_ID | NO_OF_BEERS | RANK_OF_BEERS | GROUP_NAME |
|--|------------|-------------|---------------|------------|
| 209 Rock Bottom Restaurant & Brewery - Minneapolis | 1066 | 8 | 181 | Group 1 |
| 210 Routh Street Brewery and Grille | 1079 | 8 | 181 | Group 1 |
| 211 Scuttlebutt Brewing | 1124 | 8 | 181 | Group 1 |
| 212 Seabright Brewery | 1126 | 8 | 181 | Group 1 |
| 213 Slab City Brewing | 1153 | 8 | 181 | Group 1 |
| 214 Smuttynose Brewing Co. | 1160 | 8 | 181 | Group 1 |
| 215 South Shore Brewery | 1169 | 8 | 181 | Group 1 |
| 216 Southend Brewery and Smokehouse - Charleston | 1171 | 8 | 181 | Group 1 |
| 217 The Bruery | 1246 | 8 | 181 | Group 1 |
| 218 The Church Brew Works | 1248 | 8 | 181 | Group 1 |
| 219 Twin Rivers Brewing | 1296 | 8 | 181 | Group 1 |
| 220 Two Brothers Brewing | 1298 | 8 | 181 | Group 1 |
| 221 Upstream Brewing Company at Legacy | 1315 | 8 | 181 | Group 1 |
| 222 Voodoo Brewing Co., LLC | 1330 | 8 | 181 | Group 1 |
| 223 Yuengling & Son Brewing | 1388 | 8 | 181 | Group 1 |
| 224 Allagash Brewing | 23 | 7 | 224 | Group 2 |
| 225 America's Brewing | 32 | 7 | 224 | Group 2 |
| 226 Back Road Brewery | 63 | 7 | 224 | Group 2 |
| 227 Bamberger Mahr's-Bru | 70 | 7 | 224 | Group 2 |

3.2 Query 20

Rank beers in descending order by their alcohol by volume (ABV) content.

Only consider beers with an ABV greater than zero. Display the rank number, beer name, and ABV for all beers ranked 1-10. Do not leave any gaps in the ranking sequence when there are ties (e.g., 1, 2, 2, 2, 3, 4, 4, 5). (Hint: derived tables may help with this query.)

```

SELECT
    beer_name,
    abv,
    Ranking
FROM
    (
        SELECT
            br.beer_name,
            br.abv,
            DENSE_RANK() OVER(ORDER BY br.abv DESC) Ranking
        FROM
            BEERDB.beers br
    ) BeerRank
WHERE abv>0
AND Ranking BETWEEN 1 AND 10;
```

|  All Rows Fetched: 12 in 0.141 seconds | | | |
|--|---------------------------------|-------|---------|
| | BEER_NAME | ABV | RANKING |
| 1 | Norfolk Nog Old Dark Ale | 99.99 | 1 |
| 2 | Vetter 33 | 37 | 2 |
| 3 | Tactical Nuclear Penguin | 32 | 3 |
| 4 | Samuel Adams Utopias MMIV | 24 | 4 |
| 5 | Raison D'Extra | 20 | 5 |
| 6 | Tokyo* | 18.2 | 6 |
| 7 | 120 Minute IPA | 18 | 7 |
| 8 | World Wide Stout | 18 | 7 |
| 9 | Fort | 18 | 7 |
| 10 | Samuel Adams Triplebock 1994 | 17 | 8 |
| 11 | Mephistopheles Stout | 16.03 | 9 |
| 12 | Venus - Belgian-style Quadrupel | 16 | 10 |

3.3 Query 21

Display the film title, film year and worldwide gross for all movies directed by Christopher Nolan that have a worldwide gross greater than zero. In addition, each row should contain the cumulative worldwide gross (current row's worldwide gross plus the sum of all previous rows' worldwide gross). Records should be sorted in ascending order by film year.

```

SELECT
    mv.film_title,
    mv.film_year,
    SUM(mv.worldwide_gross) OVER (ORDER BY mv.film_year ASC) AS worldwide_gross
FROM
    relmdb.movies mv
    JOIN relmdb.directors dr
        ON mv.film_id = dr.film_id
WHERE dr.director='Christopher Nolan'
AND mv.worldwide_gross>0;

```

| | FILM_TITLE | FILM_YEAR | WORLDWIDE_GROSS |
|---|--------------------|-----------|-----------------|
| 1 | American History X | 1998 | 23875127 |
| 2 | The Dark Knight | 2008 | 1028433571 |
| 3 | Inception | 2010 | 1853966335 |

4. Interesting Queries

4.1 Query 1

Display the count of fans' membership based on the gender using PIVOT.

```

SELECT
    *
FROM ( SELECT
        Gender,
        membership
      FROM relmdb.fans
    )
PIVOT
    ( COUNT(membership)
      FOR membership IN ('GOLD','SILVER','BRONZE','FREE')
    )
ORDER BY gender;

```

| | GENDER | 'GOLD' | 'SILVER' | 'BRONZE' | 'FREE' |
|---|--------|--------|----------|----------|--------|
| 1 | F | 497 | 499 | 450 | 464 |
| 2 | M | 1417 | 1415 | 1463 | 1450 |

4.2 Query 2

Display the count of movies having IMDB_votes in 3 ranges using UNION-

- 1) Votes between 20K and 50K
- 2) Votes between 50K and 80K
- 3) Votes above 80K

```

SELECT
    MAX(v20k) AS "Votes between 20k and 50k",

```

```

MAX(v50k) AS "Votes between 20k and 50k",
MAX(v80k) AS "Votes above 50k"
FROM( SELECT
    COUNT(1) v20k,
    0 v50k,
    0 v80k
FROM relmdb.movies mv
WHERE mv.imdb_votes > 20000
AND mv.imdb_votes < 50000
UNION
SELECT
    0 v20k,
    COUNT(1) v50k,
    0 v80k
FROM relmdb.movies mv
WHERE mv.imdb_votes >= 50000
AND mv.imdb_votes < 80000
UNION
SELECT
    0 v20k,
    0 v50k,
    COUNT(1) v80k
FROM relmdb.movies mv
WHERE mv.imdb_votes >= 80000
)

```

| | Votes above 20k | Votes between 20k AND 50k | Votes above 50k |
|---|-----------------|---------------------------|-----------------|
| 1 | 41 | 38 | 171 |

4.3 Query 3

Display the full name, age of fans, list of movies they have seen along with their respective MPAA_Ratings using LISTAGG.

```


SELECT
    f.fname||' '||f.lname as full_name,
    EXTRACT(YEAR from sysdate)- f.birth_year as AGE,
    LISTAGG(m.mpaa_rating,',') within GROUP (ORDER by m.mpaa_rating) "Mpaa_List",
    LISTAGG(m.Film_title,'| ') within GROUP (ORDER by m.Film_title) "Film_List"
FROM
    relmdb.fan_ratings fr
    INNER JOIN relmdb.fans f
        ON (fr.fan_id=f.fan_id)
    INNER JOIN relmdb.movies m

```

```

ON (fr.film_id=m.film_id)
WHERE m.mpaa_rating is not null
GROUP
    by f.fname||' '||f.lname,
    EXTRACT(YEAR from sysdate)- f.birth_year;

```

   | Fetched 50 rows in 0.649 seconds

| | Full_Name | Age | Mpaa_List | Film_ID_List |
|----|-----------|-----|-----------------------|--|
| 1 | D SOLE | 88 | PG,PG,PG-13,R | Back to the Future Star Wars The Dark Knight Rises The Matrix |
| 2 | M HAHN | 88 | Approved | North by Northwest |
| 3 | H NAVON | 88 | Approved,PG-13 | North by Northwest The Dark Knight Rises |
| 4 | J SMITH | 89 | PG-13,R | Goodfellas The Dark Knight Rises |
| 5 | JOE GAY | 81 | PG,R,R,R | City of God Léon: The Professional Rear Window The Pianist |
| 6 | L SMITH | 83 | PG-13,PG-13,PG-13,R,R | The Dark Knight Rises The Departed The Lord of the Rings: The Return of the King The Lord of |
| 7 | MOO KIM | 68 | R,R,R,R | City of God Goodfellas Pulp Fiction Schindler's List |
| 8 | SUE NIX | 86 | PG | Star Wars: Episode V - The Empire Strikes Back |
| 9 | WAYLAND | 84 | PG,PG-13,R,R | Alien Django Unchained Raiders of the Lost Ark The Dark Knight Rises |
| 10 | C BOWERS | 103 | PG-13 | The Lord of the Rings: The Fellowship of the Ring |
| 11 | D DUNGAN | 82 | PG-13,R | Schindler's List The Lord of the Rings: The Fellowship of the Ring |
| 12 | DAN LANE | 55 | PG,PG,R,R | Apocalypse Now Back to the Future Star Wars Terminator 2: Judgment Day |
| 13 | DON HULL | 63 | Approved | Sunset Blvd. |
| 14 | E STOKES | 77 | R | Alien |
| 15 | ERIC NEW | 89 | R,R,R | Fight Club Léon: The Professional The Silence of the Lambs |
| 16 | J POTTER | 97 | Approved | Psycho |