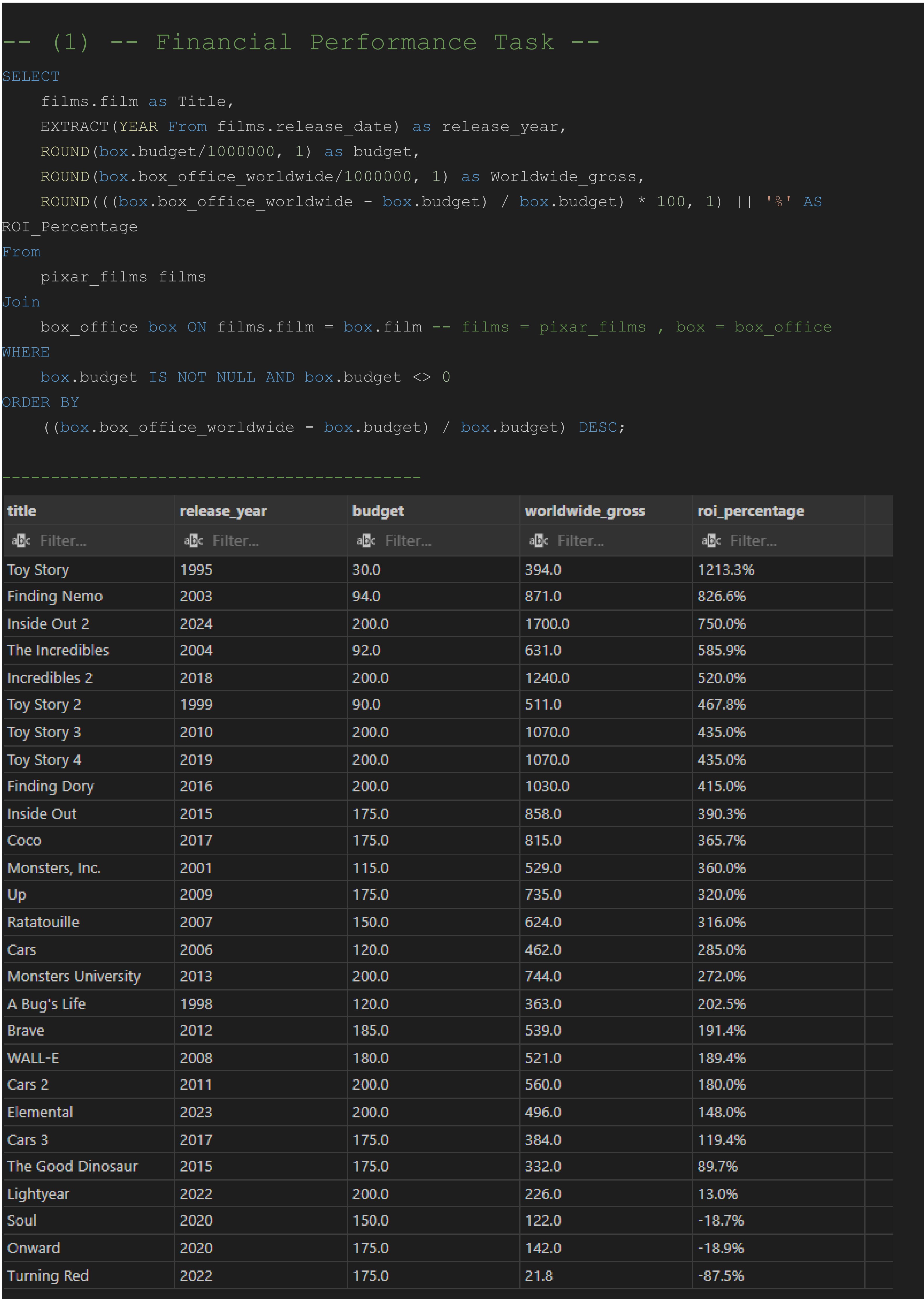
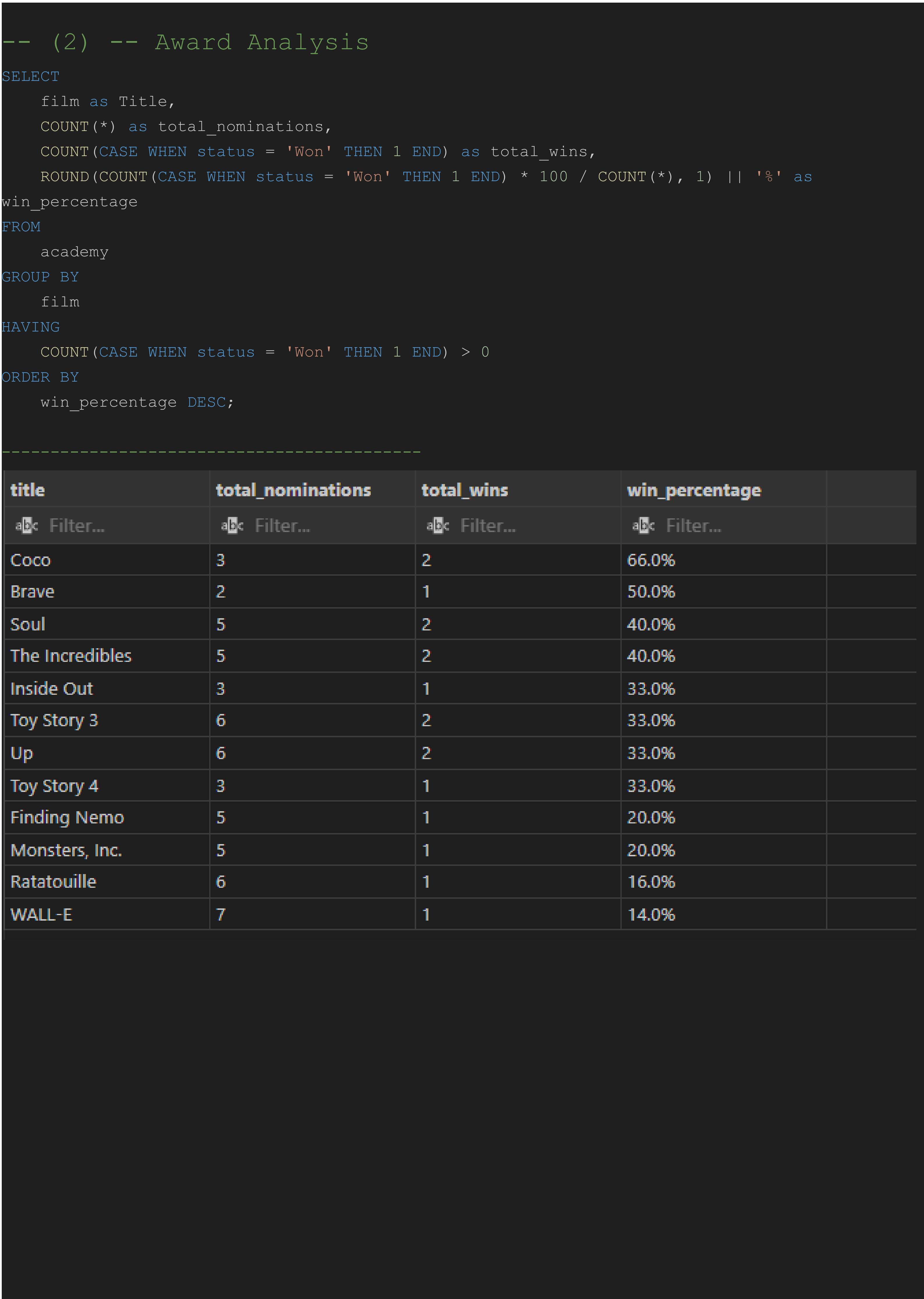


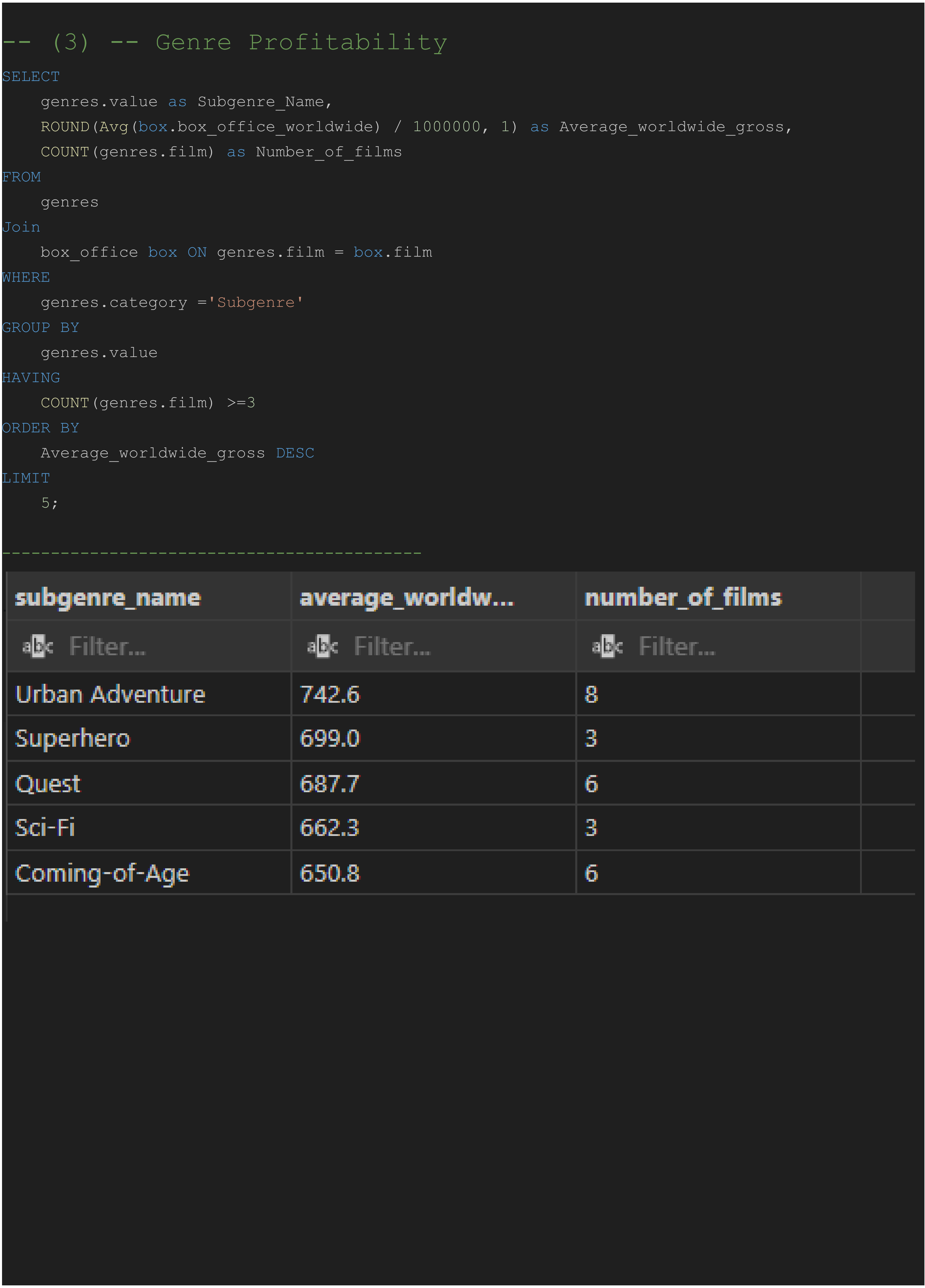
```
-- (1) -- Financial Performance Task --
SELECT
    films.film as Title,
    EXTRACT(YEAR From films.release_date) as release_year,
    ROUND(box.budget/1000000, 1) as budget,
    ROUND(box.box_office_worldwide/1000000, 1) as Worldwide_gross,
    ROUND(((box.box_office_worldwide - box.budget) / box.budget) * 100, 1) || '%' AS
ROI_Percentage
From
    pixar_films films
Join
    box_office box ON films.film = box.film -- films = pixar_films , box = box_office
WHERE
    box.budget IS NOT NULL AND box.budget <> 0
ORDER BY
    ((box.box_office_worldwide - box.budget) / box.budget) DESC;

-------------------------------------------
```

| title | release_year | budget | worldwide_gross | roi_percentage |
|---|---|---|---|---|
| Filter... | Filter... | Filter... | Filter... | Filter... |
| Toy Story | 1995 | 30.0 | 394.0 | 1213.3% |
| Finding Nemo | 2003 | 94.0 | 871.0 | 826.6% |
| Inside Out 2 | 2024 | 200.0 | 1700.0 | 750.0% |
| The Incredibles | 2004 | 92.0 | 631.0 | 585.9% |
| Incredibles 2 | 2018 | 200.0 | 1240.0 | 520.0% |
| Toy Story 2 | 1999 | 90.0 | 511.0 | 467.8% |
| Toy Story 3 | 2010 | 200.0 | 1070.0 | 435.0% |
| Toy Story 4 | 2019 | 200.0 | 1070.0 | 435.0% |
| Finding Dory | 2016 | 200.0 | 1030.0 | 415.0% |
| Inside Out | 2015 | 175.0 | 858.0 | 390.3% |
| Coco | 2017 | 175.0 | 815.0 | 365.7% |
| Monsters, Inc. | 2001 | 115.0 | 529.0 | 360.0% |
| Up | 2009 | 175.0 | 735.0 | 320.0% |
| Ratatouille | 2007 | 150.0 | 624.0 | 316.0% |
| Cars | 2006 | 120.0 | 462.0 | 285.0% |
| Monsters University | 2013 | 200.0 | 744.0 | 272.0% |
| A Bug's Life | 1998 | 120.0 | 363.0 | 202.5% |
| Brave | 2012 | 185.0 | 539.0 | 191.4% |
| WALL-E | 2008 | 180.0 | 521.0 | 189.4% |
| Cars 2 | 2011 | 200.0 | 560.0 | 180.0% |
| Elemental | 2023 | 200.0 | 496.0 | 148.0% |
| Cars 3 | 2017 | 175.0 | 384.0 | 119.4% |
| The Good Dinosaur | 2015 | 175.0 | 332.0 | 89.7% |
| Lightyear | 2022 | 200.0 | 226.0 | 13.0% |
| Soul | 2020 | 150.0 | 122.0 | -18.7% |
| Onward | 2020 | 175.0 | 142.0 | -18.9% |
| Turning Red | 2022 | 175.0 | 21.8 | -87.5% |



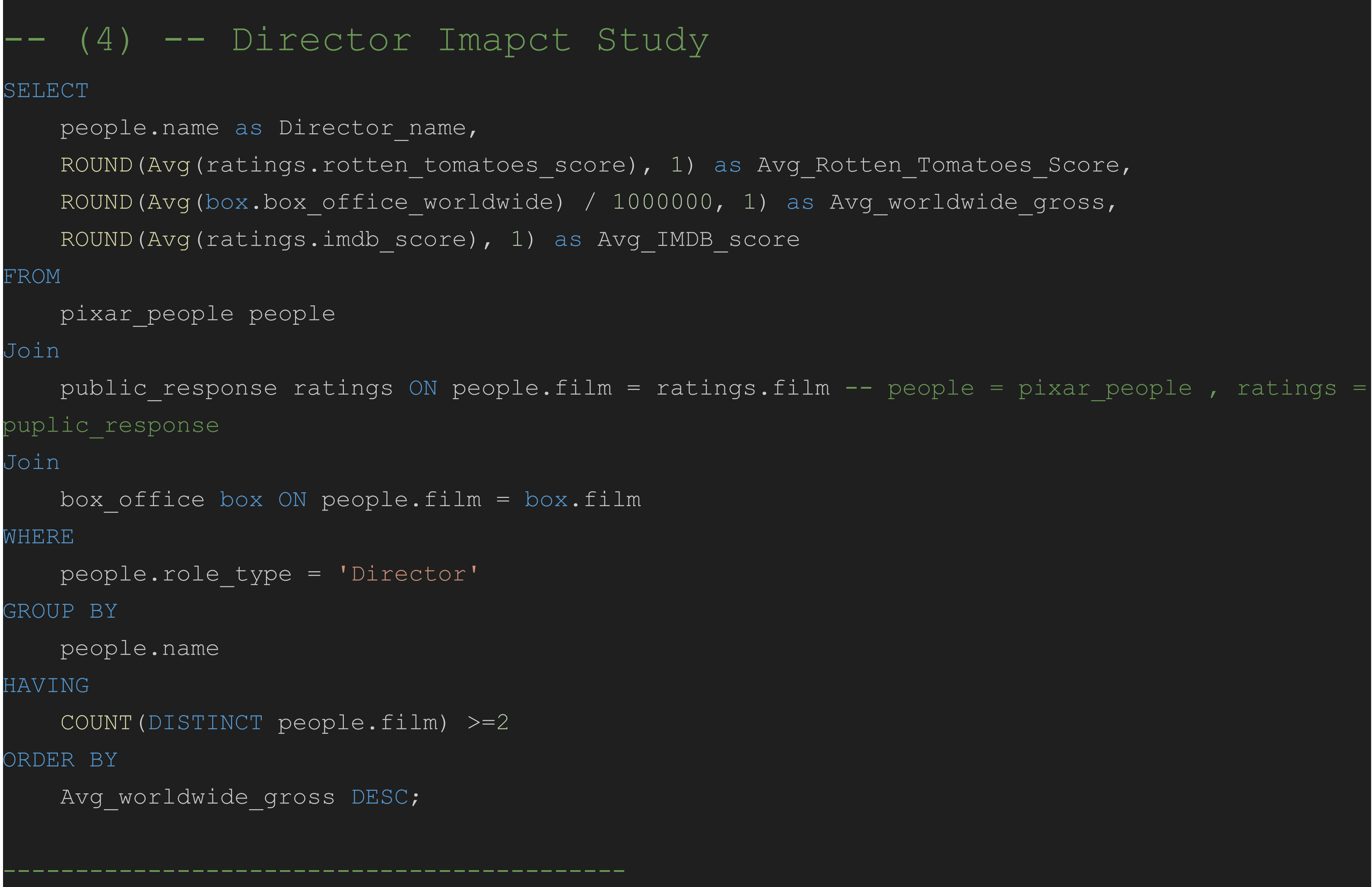
```
-- (2) -- Award Analysis
SELECT
    film as Title,
    COUNT(*) as total_nominations,
    COUNT(CASE WHEN status = 'Won' THEN 1 END) as total_wins,
    ROUND(COUNT(CASE WHEN status = 'Won' THEN 1 END) * 100 / COUNT(*), 1) || '%' as
win_percentage
FROM
    academy
GROUP BY
    film
HAVING
    COUNT(CASE WHEN status = 'Won' THEN 1 END) > 0
ORDER BY
    win_percentage DESC;

-------------------------------------------
```

| title | total_nominations | total_wins | win_percentage |
|---|---|---|---|
| Coco | 3 | 2 | 66.0% |
| Brave | 2 | 1 | 50.0% |
| Soul | 5 | 2 | 40.0% |
| The Incredibles | 5 | 2 | 40.0% |
| Inside Out | 3 | 1 | 33.0% |
| Toy Story 3 | 6 | 2 | 33.0% |
| Up | 6 | 2 | 33.0% |
| Toy Story 4 | 3 | 1 | 33.0% |
| Finding Nemo | 5 | 1 | 20.0% |
| Monsters, Inc. | 5 | 1 | 20.0% |
| Ratatouille | 6 | 1 | 16.0% |
| WALL-E | 7 | 1 | 14.0% |

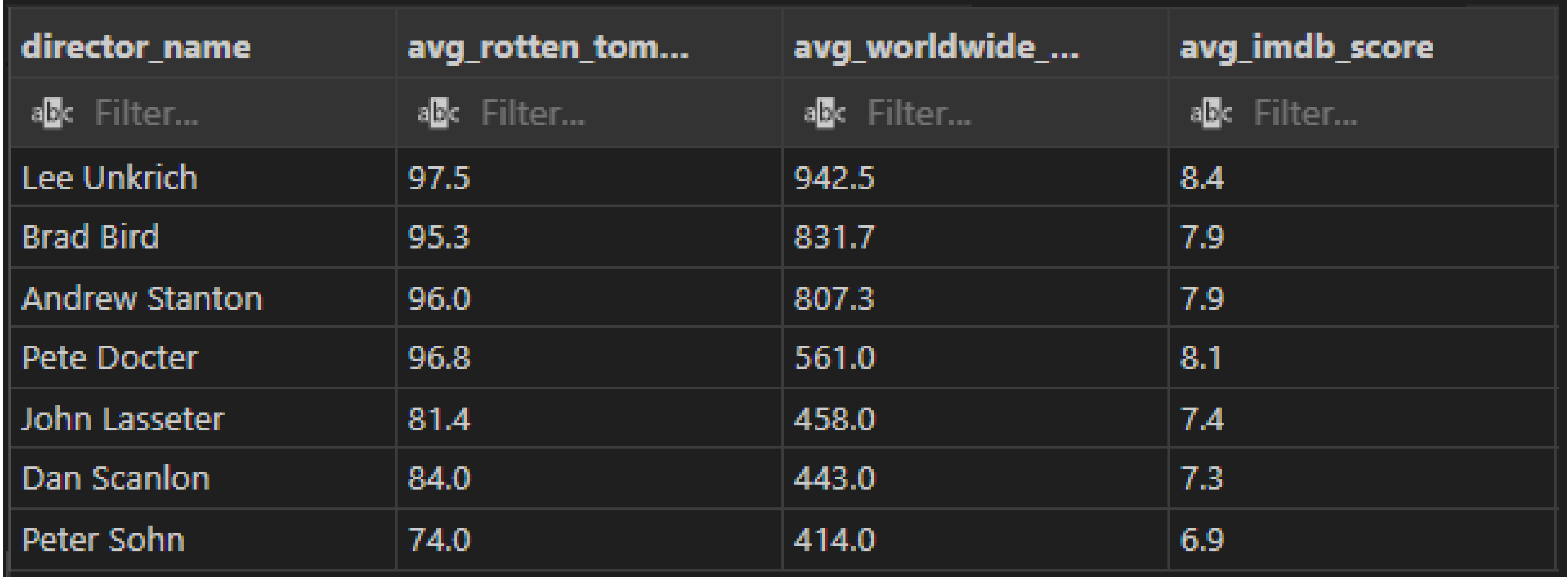


```sql
-- (3) -- Genre Profitability
SELECT
    genres.value as Subgenre_Name,
    ROUND(Avg(box.box_office_worldwide) / 1000000, 1) as Average_worldwide_gross,
    COUNT(genres.film) as Number_of_films
FROM
    genres
Join
    box_office box ON genres.film = box.film
WHERE
    genres.category ='Subgenre'
GROUP BY
    genres.value
HAVING
    COUNT(genres.film) >=3
ORDER BY
    Average_worldwide_gross DESC
LIMIT
    5;

----------------------------------------
```

| subgenre_name | average_worldw... | number_of_films |
|---|---|---|
| Filter... | Filter... | Filter... |
| Urban Adventure | 742.6 | 8 |
| Superhero | 699.0 | 3 |
| Quest | 687.7 | 6 |
| Sci-Fi | 662.3 | 3 |
| Coming-of-Age | 650.8 | 6 |

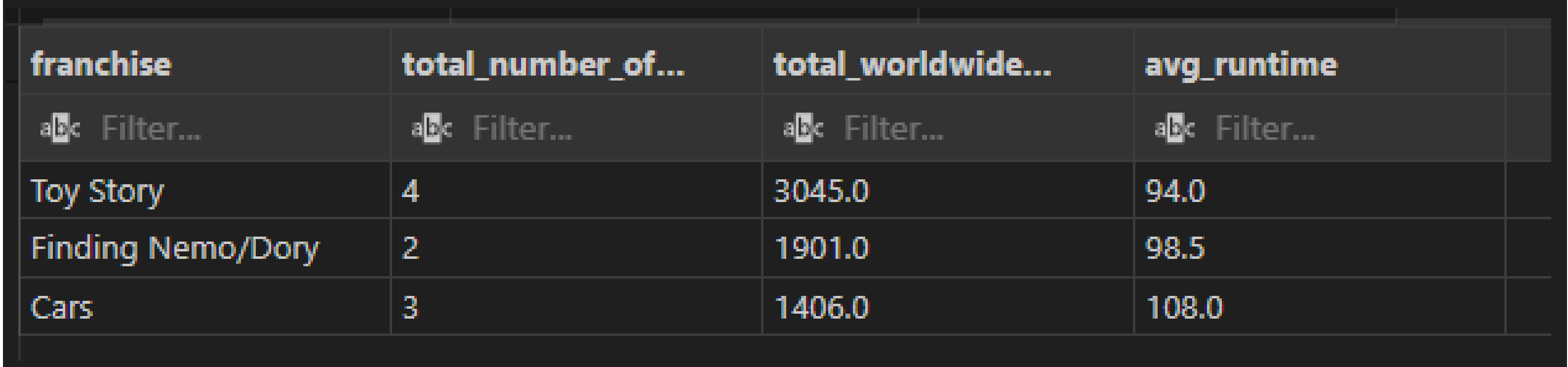


```sql
-- (4) -- Director Imapct Study
SELECT
    people.name as Director_name,
    ROUND(Avg(ratings.rotten_tomatoes_score), 1) as Avg_Rotten_Tomatoes_Score,
    ROUND(Avg(box.box_office_worldwide) / 1000000, 1) as Avg_worldwide_gross,
    ROUND(Avg(ratings.imdb_score), 1) as Avg_IMDB_score
FROM
    pixar_people people
Join
    public_response ratings ON people.film = ratings.film -- people = pixar_people , ratings =
puplic_response
Join
    box_office box ON people.film = box.film
WHERE
    people.role_type = 'Director'
GROUP BY
    people.name
HAVING
    COUNT(DISTINCT people.film) >=2
ORDER BY
    Avg_worldwide_gross DESC;

-------------------------------------------
```

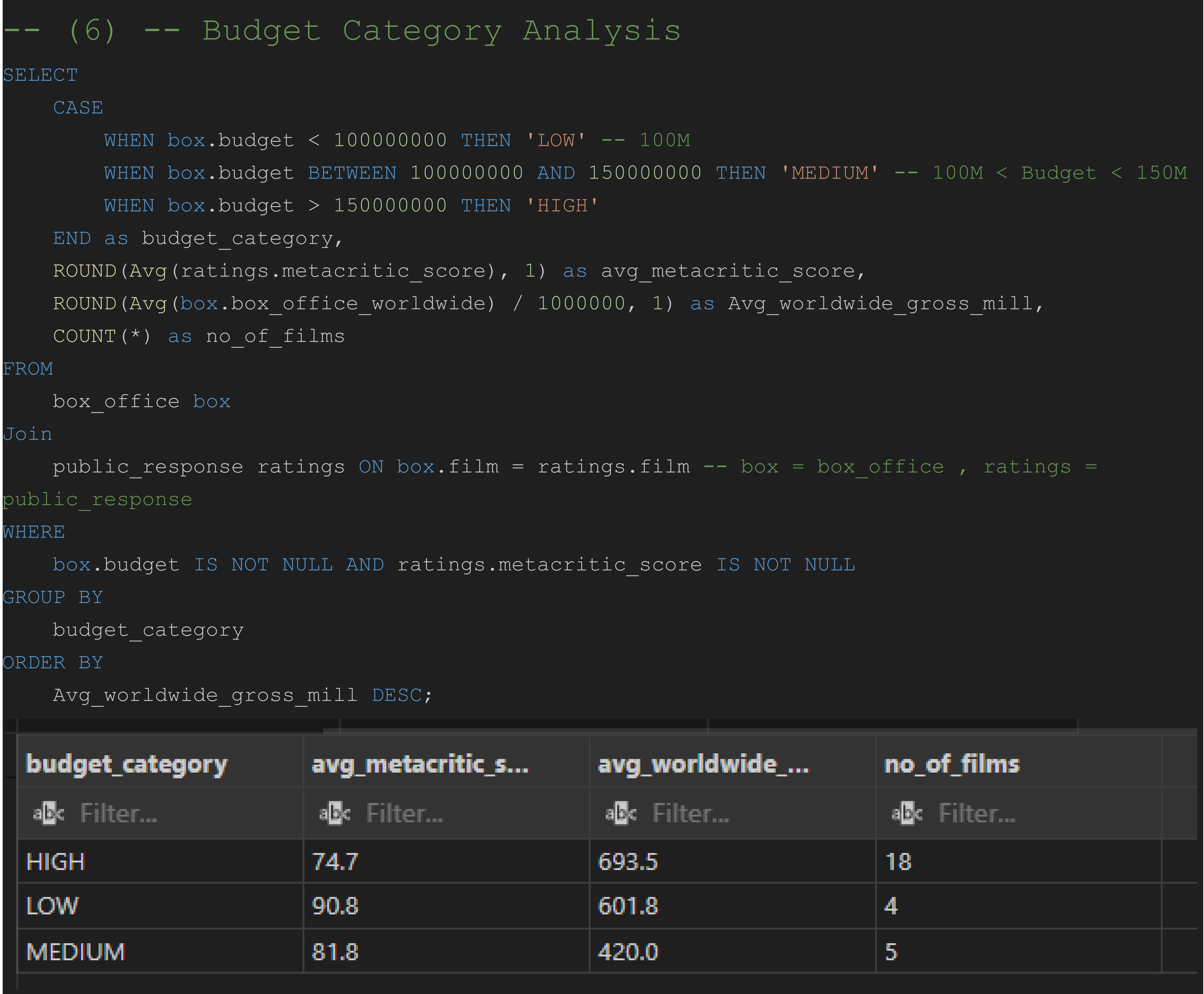


| director_name | avg_rotten_tom... | avg_worldwide_... | avg_imdb_score |
|---|---|---|---|
| Filter... | Filter... | Filter... | Filter... |
| Lee Unkrich | 97.5 | 942.5 | 8.4 |
| Brad Bird | 95.3 | 831.7 | 7.9 |
| Andrew Stanton | 96.0 | 807.3 | 7.9 |
| Pete Docter | 96.8 | 561.0 | 8.1 |
| John Lasseter | 81.4 | 458.0 | 7.4 |
| Dan Scanlon | 84.0 | 443.0 | 7.3 |
| Peter Sohn | 74.0 | 414.0 | 6.9 |

```sql
-- (5) -- Franchise Comparison
SELECT
    CASE
        WHEN films.film IN ('Finding Nemo', 'Finding Dory') THEN 'Finding Nemo/Dory'
        ELSE REGEXP_REPLACE(films.film, '\s[0-9]+$', '')
    END AS franchise,
    COUNT(*) total_number_of_films,
    ROUND(SUM(box.box_office_worldwide) / 1000000, 1) as Total_worldwide_gross,
    ROUND(Avg(films.run_time), 1) as Avg_Runtime
FROM
    pixar_films films
Join
    box_office box ON films.film = box.film -- films = pixar_films , box = box_office
WHERE
    REGEXP_REPLACE(films.film, '\s[0-9]+$', '') IN ('Toy Story', 'Cars')
        OR films.film IN ('Finding Nemo', 'Finding Dory')
GROUP BY
    franchise
ORDER BY
    Total_worldwide_gross DESC;

-------------------------------------------
```



| franchise | total_number_of... | total_worldwide... | avg_runtime |
|---|---|---|---|
| Toy Story | 4 | 3045.0 | 94.0 |
| Finding Nemo/Dory | 2 | 1901.0 | 98.5 |
| Cars | 3 | 1406.0 | 108.0 |



```
-- (6) -- Budget Category Analysis
SELECT
    CASE
        WHEN box.budget < 100000000 THEN 'LOW' -- 100M
        WHEN box.budget BETWEEN 100000000 AND 150000000 THEN 'MEDIUM' -- 100M < Budget < 150M
        WHEN box.budget > 150000000 THEN 'HIGH'
    END as budget_category,
    ROUND(Avg(ratings.metacritic_score), 1) as avg_metacritic_score,
    ROUND(Avg(box.box_office_worldwide) / 1000000, 1) as Avg_worldwide_gross_mill,
    COUNT(*) as no_of_films
FROM
    box_office box
Join
    public_response ratings ON box.film = ratings.film -- box = box_office , ratings =
public_response
WHERE
    box.budget IS NOT NULL AND ratings.metacritic_score IS NOT NULL
GROUP BY
    budget_category
ORDER BY
    Avg_worldwide_gross_mill DESC;
```

| budget_category | avg_metacritic_s... | avg_worldwide_... | no_of_films |
|---|---|---|---|
| Filter... | Filter... | Filter... | Filter... |
| HIGH | 74.7 | 693.5 | 18 |
| LOW | 90.8 | 601.8 | 4 |
| MEDIUM | 81.8 | 420.0 | 5 |