

```
-- (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;
```

```
-----

-- (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;
```

```
-----

-- (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;

-----

-- (4) -- Director Impact 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 =
public_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;

-----

-- (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;

-----

-- (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;

```