

```
-- =====
-- TFI DASHBOARD VALIDATION SCRIPT (MSSQL VERSION)
-- =====

-----

-- 0. DATA INTEGRITY & NULL CHECKS (Perform these first)
-----

use MyDatabase

SELECT

    -- Logical Check: Day 1 should not exceed Total Collection

    SUM(CASE WHEN day1_WW_Gross_cr > total_WW_cls THEN 1 ELSE 0 END) AS
logic_error_day1_gt_total,

    -- Null Checks

    SUM(CASE WHEN movie_name IS NULL THEN 1 ELSE 0 END) AS null_movies,

    SUM(CASE WHEN hero_name IS NULL THEN 1 ELSE 0 END) AS null_heroes,

    SUM(CASE WHEN total_WW_cls IS NULL THEN 1 ELSE 0 END) AS
null_total_collections,

    SUM(CASE WHEN day1_WW_Gross_cr IS NULL THEN 1 ELSE 0 END) AS
null_day1_collections,

    -- Range Check: Year of release (Ensuring no future dates or invalid years)

    SUM(CASE WHEN year_of_release > YEAR(GETDATE()) OR year_of_release < 1990
THEN 1 ELSE 0 END) AS invalid_years

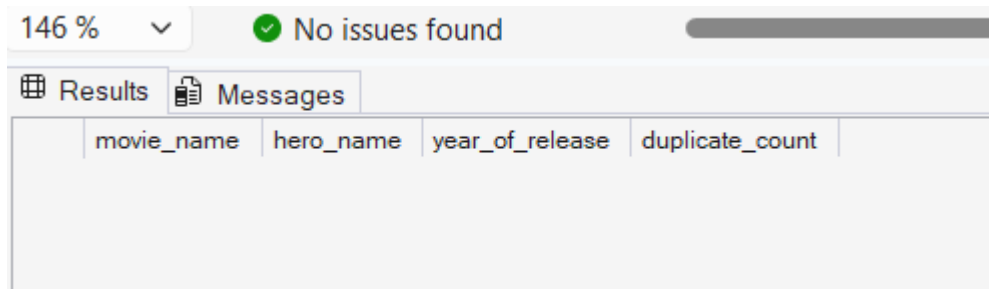
FROM data;
```

The screenshot shows a SQL query results window with a status bar at the top indicating '146 %' zoom and 'No issues found'. Below the status bar are tabs for 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with 7 columns: 'logic_error_day1_gt_total', 'null_movies', 'null_heroes', 'null_total_collections', 'null_day1_collections', and 'invalid_years'. There is one row of data where all values are 0.

	logic_error_day1_gt_total	null_movies	null_heroes	null_total_collections	null_day1_collections	invalid_years
1	0	0	0	0	0	0

-- Checking for Duplicate Movie Entries

```
SELECT movie_name, hero_name, year_of_release, COUNT(*) as duplicate_count
FROM data
GROUP BY movie_name, hero_name, year_of_release
HAVING COUNT(*) > 1;
```



movie_name	hero_name	year_of_release	duplicate_count
------------	-----------	-----------------	-----------------

-- 1. MAIN KPI CARDS (FILTERED FOR PRABHAS)

-- Success Rate (70.59%), Peak (1,810 Cr), Highest Opening (215 Cr)

```
SELECT
    'Prabhas' as Hero,
    COUNT(*) as total_movies,
    ROUND(100.0 * SUM(CASE
        WHEN verdict IN ('Hit', 'Blockbuster', 'All Time Blockbuster', 'Super Hit') THEN 1
        ELSE 0
    END) / COUNT(*), 2) as success_rate_percent,
    MAX(total_WW_cls) as peak_lifetime_cr,
    MAX(day1_WW_Gross_cr) as highest_openingDay_cr
FROM data
WHERE hero_name = 'Prabhas';
```

146 % ✓ No issues found

Results Messages

	Hero	total_movies	success_rate_percent	peak_lifetime_cr	highest_openingDay_cr
1	Prabhas	17	70.59000000000000	1810	214.5

-- 2. HIGHEST TOTAL WW GROSS (TOP 6 - PRABHAS)

-- Funnel chart showing Baahubali 2, Kalki, etc.

SELECT TOP 6

movie_name,

total_WW_cls as collection_cr

FROM data

WHERE hero_name = 'Prabhas'

ORDER BY total_WW_cls DESC;

146 % ✓ No issues found

Results Messages

	movie_name	collection_cr
1	Baahubali 2: The Conclusion	1810
2	Kalki 2898 AD	1100
3	Baahubali: The Beginning	650
4	Salaar	650
5	Saaho	435
6	Adipurush	393

-- 3. GLOBAL TREEMAP VALIDATION (UNFILTERED)

-- "Highest Day1 WW Gross of TFI" (Includes Pushpa 2, RRR, etc.)

SELECT TOP 6

movie_name,

hero_name,

day1_WW_Gross_cr

FROM data

ORDER BY day1_WW_Gross_cr DESC;

	movie_name	hero_name	day1_WW_Gross_cr
1	Pushpa 2: The Rule	Allu Arjun	270
2	RRR	Ram Charan	223
3	RRR	Jr NTR	223
4	Baahubali 2: The Conclusion	Prabhas	214.5
5	Game Changer	Ram Charan	186
6	Kalki 2898 AD	Prabhas	182.600006103516

-- 4. TOP 6 MOVIES BY OPENING DAY (FILTERED FOR PRABHAS)

--The pink bar chart (Baahubali 2: 215, Kalki: 183, Salaar: 165, etc.)

SELECT TOP 6

movie_name,

day1_WW_Gross_cr

FROM data

WHERE hero_name = 'Prabhas'

ORDER BY day1_WW_Gross_cr DESC;

	movie_name	day1_WW_Gross_cr
1	Baahubali 2: The Conclusion	214.5
2	Kalki 2898 AD	182.600006103516
3	Salaar	165.300003051758
4	Adipurush	95
5	Saaho	93
6	Baahubali: The Beginning	72.5

-- 5. VERDICT DISTRIBUTION (DONUT CHART - PRABHAS)

-- Hit(6), Blockbuster(4), ATB(2), Flop(4), Disaster(1)

SELECT

verdict,

COUNT(*) as movie_count

FROM data

WHERE hero_name = 'Prabhas'

GROUP BY verdict

ORDER BY movie_count DESC;

	verdict	movie_count
1	Hit	6
2	Blockbuster	4
3	Flop	4
4	All Time Blockbuster	2
5	Disaster	1

-- 6. ANNUAL TRENDS (COMBO CHART - PRABHAS)

-- Blue bars (Collections) and Yellow line (Success Rate)

SELECT

year_of_release,

```

SUM(total_WW_cls) as yearly_total,
ROUND(100.0 * SUM(CASE
    WHEN verdict IN ('Hit', 'Blockbuster', 'All Time Blockbuster', 'Super Hit') THEN 1
    ELSE 0
END) / COUNT(*), 2) as yearly_success_rate
FROM data
WHERE hero_name = 'Prabhas'
GROUP BY year_of_release
ORDER BY year_of_release ASC;

```

	year_of_release	yearly_total	yearly_success_rate
1	2002	5	0.000000000000
2	2004	79.7999992370605	66.670000000000
3	2005	55	100.000000000000
4	2008	35	100.000000000000
5	2009	45	0.000000000000
6	2010	38	100.000000000000
7	2011	25	100.000000000000
8	2013	86	100.000000000000
9	2015	650	100.000000000000
10	2017	1810	100.000000000000
11	2019	435	100.000000000000
12	2022	200	0.000000000000
13	2023	1043	50.000000000000