

## ***Loan Approval Intelligence Suite***

### **SQL Query Results**

#### **Dataset:**

	Column Name	Data Type
🔑	loan_id	smallint
	no_of_dependents	float
	education	nvarchar(50)
	self_employed	bit
	income_annum	float
	loan_amount	float
	loan_term	time(7)
	cibil_score	float
	residential_assets_value	float
	commercial_assets_value	float
	luxury_assets_value	float
	bank_asset_value	float
	loan_status	nvarchar(50)
	Income_to_Loan	float
	Total_Income	float
	loan_term_months	

#### **Query with their respective result:**

##### **-- 1. Write queries to count approved vs. rejected loans**

```
SELECT
    Loan_Status,
    COUNT(*) AS Total_Count
FROM dbo.cleaned_loan_approval_dataset
GROUP BY Loan_Status;
```

#### **Result:**

Loan_Status	Total_Count
Approved	2656
Rejected	1613

## -- 2. Analyze approval rates by Education

```
SELECT
    Education,
    COUNT(*) AS Total_Applications,
    SUM(CASE WHEN loan_status = 'Approved' THEN 1 ELSE 0 END) AS Approved,
    ROUND(
        (SUM(CASE WHEN Loan_Status = 'Approved' THEN 1 ELSE 0 END) * 100.0) / COUNT(*),2
    ) AS Approval_Rate_Percent
FROM dbo.cleaned_loan_approval_dataset
GROUP BY Education
ORDER BY Approval_Rate_Percent DESC;
```

### Result:

Education	Total_Applications	Approved	Approval_Rate_Percent
Graduate	2144	1339	62.4500000000000
Not Graduate	2125	1317	61.9800000000000

## -- 3. Use CASE statements to classify risk levels

```
SELECT
    loan_id,
    income_annum,
    loan_amount,
    CASE
        WHEN loan_amount > (income_annum * 0.8) THEN 'High Risk'
        WHEN loan_amount BETWEEN (income_annum * 0.5) AND (income_annum * 0.8)
        THEN 'Medium Risk'
        WHEN loan_amount < (income_annum * 0.5) THEN 'Low Risk'
        ELSE 'Unknown'
    END AS Risk_Level
FROM dbo.cleaned_loan_approval_dataset;
```

### Result:

loan_id	income_annum	loan_amount	Risk_Level
1	9600000	29900000	High Risk
2	4100000	12200000	High Risk
3	9100000	29700000	High Risk
4	8200000	30700000	High Risk .....

**-- 4. Create views for Overall Approval Summary**

```
GO
CREATE VIEW dbo.Approval_Summary AS
SELECT
    COUNT(*) AS Total_Applications,
    SUM(CASE WHEN Loan_Status = 'Approved' THEN 1 ELSE 0 END) AS Total_Approved,
    SUM(CASE WHEN Loan_Status = 'Rejected' THEN 1 ELSE 0 END) AS Total_Rejected,
    ROUND(
        (SUM(CASE WHEN Loan_Status = 'Approved' THEN 1 ELSE 0 END) * 100.0) / COUNT(*),
        2
    ) AS Approval_Rate_Percent
FROM dbo.cleaned_loan_approval_dataset;
GO
```

```
SELECT * FROM dbo.Approval_Summary;
```

**Result:**

Total_Applications	Total_Approved	Total_Rejected	Approval_Rate_Percent
4269	2656	1613	62.2200000000000

**-- 5. Write subqueries to find top 5 income earners**

```
SELECT TOP 5 *
FROM (
    SELECT
        loan_id,
        income_annum,
        loan_amount
    FROM dbo.cleaned_loan_approval_dataset
) AS income_table
ORDER BY income_annum DESC;
```

**Result:**

loan_id	income_annum	loan_amount
83	9900000	21200000
158	9900000	33300000
301	9900000	38200000
361	9900000	38400000
510	9900000	39500000

## -- 6. Use window functions to rank applicants by income

```
SELECT
    loan_id,
    income_annum,
    loan_amount,
    RANK() OVER (ORDER BY income_annum DESC) AS income_rank
FROM dbo.cleaned_loan_approval_dataset
ORDER BY income_rank;
```

### Result:

loan_id	income_annum	loan_amount	income_rank
83	9900000	21200000	1
158	9900000	33300000	1
301	9900000	38200000	1
361	9900000	38400000	1
510	9900000	39500000	1
531	9900000	31800000	1
715	9900000	22700000	1
782	9900000	20400000	1
.....			

## -- 7. Filter Applicants with Missing or Invalid CIBIL Scores

```
SELECT *
FROM dbo.cleaned_loan_approval_dataset
WHERE cibil_score IS NULL
    OR cibil_score = 0
    OR cibil_score < 300
    OR cibil_score > 900;
```

### Result:

loan_id	loan_term	Income_to_Loan
no_of_dependents	cibil_score	Total_Income
education	residential_assets_value	loan_term_months
self_employed	commercial_assets_value	bank_asset_value
income_annum	luxury_assets_value	
loan_amount	loan_status	

**-- 8. Find applicants whose total asset value is greater than their loan amount**

```
SELECT
    loan_id,
    income_annum,
    loan_amount,
    (residential_assets_value + commercial_assets_value + luxury_assets_value +
    bank_asset_value) AS total_assets,
    loan_status
FROM dbo.cleaned_loan_approval_dataset
WHERE (residential_assets_value + commercial_assets_value + luxury_assets_value +
bank_asset_value) > loan_amount
ORDER BY total_assets DESC;
```

**Result:**

loan_id	income_annum	loan_amount	total_assets	loan_status
368	9400000	29800000	85200000	Rejected
2456	8400000	19700000	85100000	Rejected
99	9400000	29400000	84600000	Approved
1420	9100000	21700000	83400000	Rejected
896	9400000	27900000	83300000	Rejected
3785	9800000	21200000	82500000	Approved
1769	9900000	29100000	82000000	Approved
1273	9900000	35800000	81700000	Rejected
.....				

**-- 9. Average CIBIL score and income by employment type**

```
SELECT
    self_employed,
    ROUND(AVG(cibil_score), 2) AS avg_cibil_score,
    ROUND(AVG(income_annum), 2) AS avg_income
FROM dbo.cleaned_loan_approval_dataset
GROUP BY self_employed
ORDER BY avg_cibil_score DESC;
```

**Result:**

self_employed	avg_cibil_score	avg_income
0	600.78	5052430.39
1	599.1	5065720.93

**-- 10. Average Loan Amount and Income by Number of Dependents**

```
SELECT
    no_of_dependents,
    ROUND(AVG(income_annum), 2) AS avg_income,
    ROUND(AVG(loan_amount), 2) AS avg_loan_amount,
    COUNT(*) AS total_applicants
FROM dbo.cleaned_loan_approval_dataset
GROUP BY no_of_dependents
ORDER BY no_of_dependents;
```

**Result:**

no_of_dependents	avg_income	avg_loan_amount	total_applicants
0	5001544.94	14979073.03	712
1	5040459.11	15212195.12	697
2	5041949.15	15201977.4	708
3	5196561.21	15590784.04	727
4	4987367.02	14787500	752
5	5089153.05	15035661.22	673

----- End -----