KPI 1

```sql
SELECT * FROM bankanalysis.combined_data;
WITH loan_stats AS (
    SELECT
        YEAR(STR_TO_DATE(issiue_d, '%d/%m/%Y')) AS `year`,
        loan_amnt
    FROM combined_data
    WHERE issiue_d IS NOT NULL AND loan_amnt IS NOT NULL
),
ranked_loans AS (
    SELECT
        year,
        loan_amnt,
        ROW_NUMBER() OVER (PARTITION BY year ORDER BY loan_amnt) AS rn,
        COUNT(*) OVER (PARTITION BY year) AS cnt
    FROM loan_stats
),
median_loan AS (
    SELECT year, loan_amnt
    FROM ranked_loans
    WHERE rn = FLOOR(cnt / 2) + 1
)
SELECT
    rl.year,
    COALESCE(CONCAT('$',
        CASE
            WHEN SUM(rl.loan_amnt) >= 1000000 THEN ROUND(SUM(rl.loan_amnt) / 1000000)
            WHEN SUM(rl.loan_amnt) >= 1000 THEN ROUND(SUM(rl.loan_amnt) / 1000)
            ELSE ROUND(SUM(rl.loan_amnt))
        END,
        CASE
            WHEN SUM(rl.loan_amnt) >= 1000000 THEN 'M'
            WHEN SUM(rl.loan_amnt) >= 1000 THEN 'K'
            ELSE ''
        END), '$0') AS `Total Loan Amount`,
    COALESCE(CONCAT('$',
        CASE
            WHEN AVG(rl.loan_amnt) >= 1000000 THEN ROUND(AVG(rl.loan_amnt) / 1000000)
            WHEN AVG(rl.loan_amnt) >= 1000 THEN ROUND(AVG(rl.loan_amnt) / 1000)
            ELSE ROUND(AVG(rl.loan_amnt))
        END,
        CASE
            WHEN AVG(rl.loan_amnt) >= 1000000 THEN 'M'
            WHEN AVG(rl.loan_amnt) >= 1000 THEN 'K'
            ELSE ''
        END), '$0') AS `Average Loan Amount`,
    COALESCE((
        SELECT CONCAT('$',
            CASE
                WHEN ml.loan_amnt >= 1000000 THEN ROUND(ml.loan_amnt / 1000000)
                WHEN ml.loan_amnt >= 1000 THEN ROUND(ml.loan_amnt / 1000)
                ELSE ROUND(ml.loan_amnt)
            END,
```

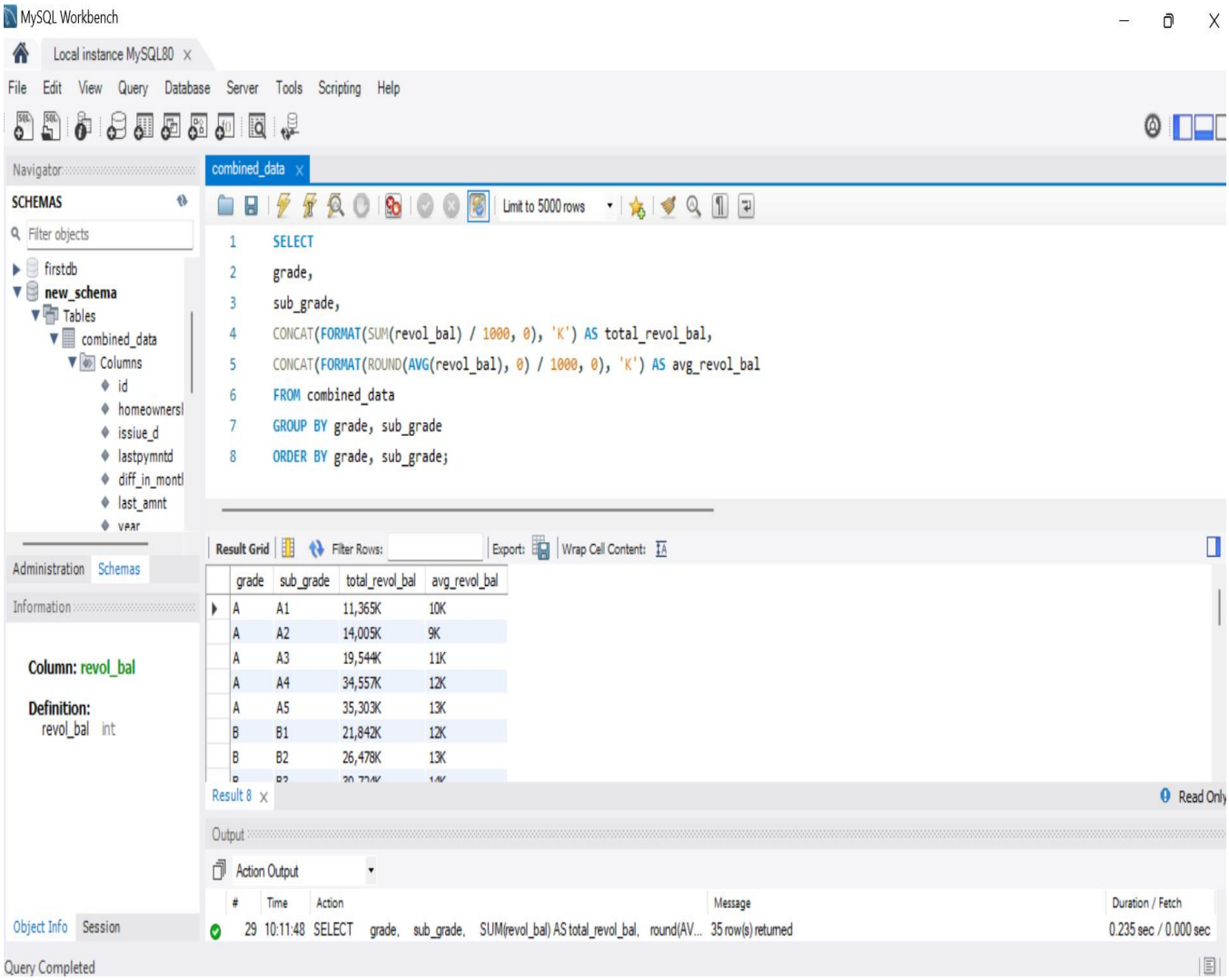
```
            CASE
                WHEN ml.loan_amnt >= 1000000 THEN 'M'
                WHEN ml.loan_amnt >= 1000 THEN 'K'
                ELSE ''
            END
        )
        FROM median_loan ml WHERE ml.year = rl.year), '$0') AS `Median Loan Amount`,
    COALESCE(CONCAT('$',
        CASE
            WHEN MAX(rl.loan_amnt) >= 1000000 THEN ROUND(MAX(rl.loan_amnt) / 1000000)
            WHEN MAX(rl.loan_amnt) >= 1000 THEN ROUND(MAX(rl.loan_amnt) / 1000)
            ELSE ROUND(MAX(rl.loan_amnt))
        END,
        CASE
            WHEN MAX(rl.loan_amnt) >= 1000000 THEN 'M'
            WHEN MAX(rl.loan_amnt) >= 1000 THEN 'K'
            ELSE ''
        END), '$0') AS `Max Loan Amount`,
    COALESCE(CONCAT('$',
        CASE
            WHEN MIN(rl.loan_amnt) >= 1000000 THEN ROUND(MIN(rl.loan_amnt) / 1000000)
            WHEN MIN(rl.loan_amnt) >= 1000 THEN ROUND(MIN(rl.loan_amnt) / 1000)
            ELSE ROUND(MIN(rl.loan_amnt))
        END,
        CASE
            WHEN MIN(rl.loan_amnt) >= 1000000 THEN 'M'
            WHEN MIN(rl.loan_amnt) >= 1000 THEN 'K'
            ELSE ''
        END), '$0') AS `Min Loan Amount`

FROM ranked_loans rl
GROUP BY rl.year
HAVING rl.year IS NOT NULL
ORDER BY rl.year DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

year	Total Loan Amount	Average Loan Amount	Median Loan Amount	Max Loan Amount	Min Loan Amount
2011	$261M	$12K	$10K	$35K	$1K
2010	$122M	$11K	$10K	$25K	$1K
2009	$46M	$10K	$9K	$25K	$1K
2008	$14M	$9K	$8K	$25K	$500
2007	$2M	$9K	$7K	$25K	$500

Kpi 2


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
firstdb
new_schema
Tables
combined_data
Columns
id
homeownersl
issiue_d
lastpymntd
diff_in_montl
last_amnt
year
Administration Schemas
Information
Column: revol_bal
Definition:
revol_bal int
Object Info Session
combined_data
Limit to 5000 rows
1 SELECT
2 grade,
3 sub_grade,
4 CONCAT(FORMAT(SUM(revol_bal) / 1000, 0), 'K') AS total_revol_bal,
5 CONCAT(FORMAT(ROUND(AVG(revol_bal), 0) / 1000, 0), 'K') AS avg_revol_bal
6 FROM combined_data
7 GROUP BY grade, sub_grade
8 ORDER BY grade, sub_grade;
Result Grid Filter Rows: Export: Wrap Cell Content:
grade sub_grade total_revol_bal avg_revol_bal
A A1 11,365K 10K
A A2 14,005K 9K
A A3 19,544K 11K
A A4 34,557K 12K
A A5 35,303K 13K
B B1 21,842K 12K
B B2 26,478K 13K
Result 8
Read Only
Output
Action Output
Time Action Message Duration / Fetch
29 10:11:48 SELECT grade, sub_grade, SUM(revol_bal) AS total_revol_bal, round(AV... 35 row(s) returned 0.235 sec / 0.000 sec
Query Completed

KPI 3

```sql
CREATE DATABASE IF NOT EXISTS my_database;
USE my_database;
SELECT DATABASE();
CREATE TABLE IF NOT EXISTS payments (
    id INT PRIMARY KEY,
    total_pymnt DECIMAL(15,6),
    verification_status VARCHAR(50)
);
SHOW TABLES;
DESCRIBE payments;
LOAD DATA INFILE 'C:/Users/shantanu/Desktop/data.csv'
INTO TABLE payments
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(id, total_pymnt, verification_status);
SELECT * FROM payments LIMIT 10;
INSERT INTO payments (id, total_pymnt, verification_status)
VALUES
    (54734, 29330.356700, 'Verified'),
    (55742, 8215.537060, 'Not Verified'),
    (57245, 1457.819531, 'Not Verified');
```

```sql
SELECT * FROM payments;
SELECT
verification_status,
SUM(total_pymnt) AS total_payment
FROM payments
GROUP BY verification_status;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

verification_status	total_payment
Verified	29330.356700
Not Verified	8215.537060

KPI 4

```sql
SELECT * FROM bankanalysis.combined_data;
SELECT
    addr_state AS `State`,
    COALESCE(DATE_FORMAT(STR_TO_DATE(issiue_d, '%d/%m/%Y'), '%M'), 'Unknown') AS `Month`,
    COALESCE(loan_status, 'Unknown') AS `Loan Status`,
    COUNT(*) AS `Loan Count`
FROM combined_data
WHERE issiue_d IS NOT NULL
    AND addr_state IS NOT NULL
    AND loan_status IS NOT NULL
    AND STR_TO_DATE(issiue_d, '%d/%m/%Y')
 IS NOT NULL  -- Ensure date conversion works
GROUP BY addr_state, Month, loan_status
ORDER BY addr_state, Month;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

State	Month	Loan Status	Loan Count
CA	December	Fully Paid	661
CA	November	Fully Paid	578
CA	October	Fully Paid	550
CA	August	Fully Paid	532
CA	September	Fully Paid	527
CA	May	Fully Paid	515
CA	June	Fully Paid	456
CA	July	Fully Paid	451
CA	April	Fully Paid	423
CA	March	Fully Paid	406

Result 12

KPI 5

```sql
SELECT * FROM bankanalysis.combined_data;
SELECT
    homeownership,
    ROUND(AVG(diff_in_months), 0) AS `Avg. Diff In Months`,
    CONCAT('$',
        CASE
            WHEN AVG(last_amnt) >= 1000000 THEN ROUND(AVG(last_amnt) / 1000000, 2)
            WHEN AVG(last_amnt) >= 1000 THEN ROUND(AVG(last_amnt) / 1000, 2)
            ELSE ROUND(AVG(last_amnt), 2)
        END,
        CASE
            WHEN AVG(last_amnt) >= 1000000 THEN 'M'
            WHEN AVG(last_amnt) >= 1000 THEN 'K'
            ELSE ''
        END
    ) AS `Avg. Last Pymnt Amnt`,
    CONCAT('$',
        CASE
            WHEN SUM(loan_amnt) >= 1000000 THEN ROUND(SUM(loan_amnt) / 1000000, 2)
            WHEN SUM(loan_amnt) >= 1000 THEN ROUND(SUM(loan_amnt) / 1000, 2)
            ELSE ROUND(SUM(loan_amnt), 2)
        END,
        CASE
            WHEN SUM(loan_amnt) >= 1000000 THEN 'M'
            WHEN SUM(loan_amnt) >= 1000 THEN 'K'
            ELSE ''
        END
    ) AS `Last Pymnt Amnt`
FROM
    combined_data
WHERE
    diff_in_months >= 0
GROUP BY
    homeownership
ORDER BY
    `Avg. Diff In Months` DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

homeownership	Avg. Diff In Months	Avg. Last Pymnt Amnt	Last Pymnt Amnt
MORTGAGE	36	$3.19K	$223.92M
OWN	35	$2.70K	$31.34M
RENT	35	$2.21K	$188.69M
NONE	42	$178.00	$16.80K
OTHER	32	$1.77K	$1.04M

Result 4