# Project Overview:

This project is a SQL-based analysis of the credit card spends. The project aims to provide further insights from the dataset.

The project consists of Three main parts:

1. Data Extraction/ Importation to MySQL Workbench
2. Exploratory Data Analysis (EDA)
3. Spending Insights

### About the dataset :

This dataset offers a comprehensive overview of credit card transactions in India. The information spans various aspects, including gender, card types, city-wise expenditures, and the types of expenses. The dataset provides a valuable opportunity to uncover deeper trends in customer spending and explore correlations between different data points, offering invaluable business intelligence. Analyzing this diverse set of variables can paint a detailed picture of how money is being spent in India today using credit cards.

### Columns and their description:

| Column name | Description |
|---|---|
| **City** | The city in which the transaction took place. (Text) |
| **Date** | The date of the transaction. (Text) |
| **Time** | The time of the transaction corresponding to the date(text) |
| **Card Type** | The type of credit card used for the transaction. (Text) |
| **Exp Type** | The type of expense associated with the transaction. (Text) |
| **Gender** | The gender of the cardholder. (Text) |
| **Amount** | The amount of the transaction. (Number) |

# Data Extraction:

I started the project by obtaining the ‘Credit Card Spending Habits in India’ dataset in a compressed ZIP file, which I subsequently extracted and converted into the CSV (Comma Separated Values) format. Following this, I created a MySQL database and imported the dataset into it.

# Exploratory Data Analysis :

Once the dataset had been successfully imported into MySQL workbench, I proceeded to write a series of queries aimed at data preparation methods such as renaming the column names, modifying the data types of columns, dropping unnecessary columns and handling *null* values.

credit_card_script | SQL File 33*

```sql
-- loaded the dataset into database named credit_card
-- checking the number of rows
select count(*) FROM credit_card.credit_card_transactions;
select * FROM credit_card_transactions;

-- checking the data types
describe credit_card_transactions;

-- To modify the datatypes of columns , we need to set the sql safe update to false
SET SQL_SAFE_UPDATES = 0;

-- change the data type of date column to date from string
update credit_card_transactions
set date = str_to_date(date, "%d-%m-%Y");
```

credit_card_script | SQL File 33*

```sql
-- modifying the datatypes
alter table credit_card_transactions
modify column  date date;

alter table credit_card_transactions
modify column  time TIME;

alter table credit_card_transactions
modify column  Gender Varchar(100);

alter table credit_card_transactions
modify column  City varchar(100);

-- changing the column names
alter table credit_card_transactions
change column `card type` card_type varchar(100);

alter table credit_card_transactions
change column `Exp Type` exp type varchar(100);
```

credit_card_script* | SQL File 33*

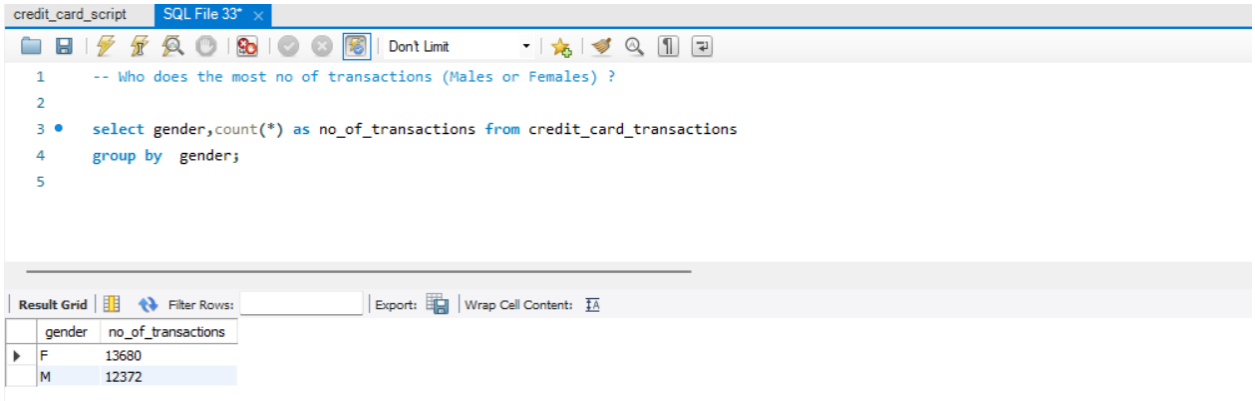
```sql
alter table credit_card_transactions
change column date transaction_date date;

-- dropping unnecessary columns
alter table credit_card_transactions
drop column `index`;

select * from credit_card_transactions;
```

# Insights into Spending Patterns :

Subsequently, I wrote queries aimed at extracting KPIs and some valuable insights from the data. The full data cleaning steps and queries are documented on my GitHub repository.

Here are some of the questions that I have answered .

## 1. Who does the most no of transactions (Males or Females) ?




It can be observed that women have done more transactions as compared to men.

## 2. Which card type is used most and least number of times ?

```sql
-- Which card type is highly and least used ?
with cte as (
select *,
dense_rank() over(order by no_of_times_used desc ) as rn from (
select card_type, count(*) as no_of_times_used from credit_card_transactions
group by card_type ) x )

select card_type , no_of_times_used,
case when rn=1 then "Most no of times used" else "Least no of times used" end as `Frequency type` from cte
where rn =1
or rn = (select max(rn) from cte)
;
```

| card_type | no_of_times_used | Frequency type |
|---|---|---|
| Silver | 6840 | Most no of times used |
| Gold | 6367 | Least no of times used |

It can be observed that Silver card type is most times used and Gold card type is least times used in credit card transactions.

## 3. Find the 3 cities having the lowest number of transactions ?

```sql
-- 3 City having lowest number of transactions ?
with cte as (
select *, dense_rank() over(order by no_of_transactions ) as rn  from (
select city, count(transaction_date) as no_of_transactions from credit_card_transactions
group by city ) x )

select city, no_of_transactions from cte
where rn in (1,2,3)
order by rn ;
```

| city | no_of_transactions |
|---|---|
| Rayagada, India | 1 |
| Alirajpur, India | 1 |
| Varanasi, India | 1 |
| Hugli-Chinsurah, India | 1 |
| Thodupuzha, India | 1 |
| Mahbubnagar, India | 1 |
| Godhra, India | 1 |
| Vellore, India | 1 |
| Bagaha, India | 1 |
| Changanassery, India | 1 |
| Tirur, India | 1 |
| Fazilka, India | 1 |
| Taki, India | 2 |
| Vijayawada, India | 2 |
| Repalle, India | 2 |

Note : Output contains records having a number of transactions from 1 to 3. In the picture, there are only a few records shown for reference.

## 4. Top 3 cities having the most number of transactions ?

```sql
-- Top 3 cities having the most number of transactions ?
with cte as (
select *, dense_rank() over(order by no_of_transactions desc ) as rn  from (
select city, count(transaction_date) as no_of_transactions from credit_card_transactions
group by city ) x )

select city, no_of_transactions from cte
where rn in (1,2,3)
order by rn ;
```

| city | no_of_transactions |
|---|---|
| Bengaluru, India | 3552 |
| Greater Mumbai, India | 3493 |
| Ahmedabad, India | 3491 |

It can be seen that Bengaluru tops the list in terms of total number of transactions followed by Greater Mumbai and Ahmedabad.

## 5. What is the average transaction value during weekends and weekdays in each card type?

It can be observed that the silver card type has the highest average transaction value and the

```sql
-- What is the average transaction value during weekends and weekdays in each card type?
select * from credit_card_transactions;
with cte as (
select card_type, round(avg(amount), 2) as avg_weekend_trans_value
 from (select * from credit_card_transactions where  date_format(transaction_date, "%a") in ("sat", "sun")  ) x
group by card_type )
, cte2 as (
select card_type,  round(avg(amount), 2) as avg_weekday_trans_value
 from (select * from credit_card_transactions where  date_format(transaction_date, "%a") not in ("sat", "sun") ) y
group by card_type)

select cte.card_type, cte.avg_weekend_trans_value, cte2.avg_weekday_trans_value
from cte join cte2 on cte.card_type=cte2.card_type;
```

| card_type | avg_weekend_trans_value | avg_weekday_trans_value |
|---|---|---|
| Signature | 154557.61 | 158192.58 |
| Platinum | 157342.29 | 157552.29 |
| Silver | 159836.24 | 154910.92 |
| Gold | 155692.91 | 154205.85 |

Signature card type has the least average transaction value during the weekends.
Similarly,  the Signature card type has the highest average transaction value and the Gold card type has the least average transaction value during the weekends.

## 6. In which expense category does most and least number of transactions happen ?

```sql
-- In which expense category does the most and least number of transactions happen ?
with cte as (
select exp_type, count(transaction_date) as no_of_transactions from credit_card_transactions
group by exp_type ),

cte2 as (
select *,
DENSE_RANK() over(order by no_of_transactions asc) as rn_asc,
DENSE_RANK() over(order by no_of_transactions desc) as rn_desc from cte )

, cte3 as (
select exp_type, no_of_transactions, rn_asc, rn_desc from cte2
where rn_asc=1 or rn_desc=1 )

select exp_type, no_of_transactions,
case when rn_desc=1 then "Most Number of Transactions"
when rn_asc=1 then "Least Number of Transactions" end as Frequency
from cte3;
```

| exp_type | no_of_transactions | Frequency |
|---|---|---|
| Food | 5463 | Most Number of Transactions |
| Travel | 738 | Least Number of Transactions |

'Food' expense type has the most number of transactions while the 'Travel' expense type has the least number of transactions.

## 7. Which expense type dominates most of the subcategories of Card Type ?

```sql
-- Which exp type dominates most of the subcategories of Card Type ?
with cte as (
select * from (
select card_type,exp_type, count(transaction_date) over(partition by card_type,exp_type ) as cnt
from credit_card_transactions ) x
group by  card_type,exp_type , cnt
order by card_type,exp_type )
, cte2 as (
select *,
dense_rank() over(partition by card_type order by cnt desc) as rn_desc,
dense_rank() over(partition by card_type order by cnt ) as rn_asc
from cte )

select a.card_type, a.exp_type as most_no_of_transactions_expensive_category,
b.exp_type as least_no_of_transactions_expensive_category from
(select card_type, exp_type , cnt from cte2 where rn_desc=1 ) a join
(select card_type, exp_type, cnt from cte2 where rn_asc=1) b
on a.card_type=b.card_type;
```

| card_type | most_no_of_transactions_expensive_category | least_no_of_transactions_expensive_category |
|---|---|---|
| Gold | Food | Travel |
| Platinum | Food | Travel |
| Signature | Food | Travel |
| Silver | Food | Travel |

Food is the most frequent category in all card types & travel being the least.

## 8. Which card type has the highest & lowest contribution to the total amount ?

```sql
-- Which card type has the highest & least contribution to total amount ?

select * from credit_card_transactions;
select card_type, round((amount*1.0/total_amount)*100, 3) as contribution_percent from (
select *
from (select card_type, sum(amount) as amount from credit_card_transactions group by card_type) a ,
(select sum(amount) as total_amount from credit_card_transactions) b ) x
order by round((amount*1.0/total_amount)*100, 3) desc;
```

| card_type | contribution_percent |
|---|---|
| Silver | 26.249 |
| Signature | 24.861 |
| Platinum | 24.728 |
| Gold | 24.161 |

Silver has the highest overall contribution while gold has the least contribution to the total amount.

## 9. Which Expense Type has the highest & lowest contribution to the total amount ?

```sql
-- Which Expense Type has the highest & lowest contribution to the total amount ?
select exp_type, round((amount*1.0/total_amount)*100, 3) as contribution_percent from (
select *
from (select exp_type, sum(amount) as amount from credit_card_transactions group by exp_type) a ,
(select sum(amount) as total_amount from credit_card_transactions) b ) x
order by round((amount*1.0/total_amount)*100, 3) desc;
```

| exp_type | contribution_percent |
|---|---|
| Bills | 22.260 |
| Food | 20.239 |
| Fuel | 19.366 |
| Entertainment | 17.827 |
| Grocery | 17.625 |
| Travel | 2.681 |

Bills have the highest overall contribution percentage while travel is least to the total amount.

## 10. Which gender type has the highest & lowest contribution to the total amount ?

```sql
-- Which Gender Type has the highest & lowest contribution to the total amount ?
select gender, round((amount*1.0/total_amount)*100, 3) as contribution_percent from
(
select * from
(select gender, sum(amount) as amount from credit_card_transactions group by gender) a ,
(select sum(amount) as total_amount from credit_card_transactions) b
) x
order by round((amount*1.0/total_amount)*100, 3) desc;
```

| gender | contribution_percent |
|---|---|
| F | 54.120 |
| M | 45.880 |

Females have a larger contribution percentage to the total amount as compared to men.

## 11. Compute the contribution percentage number of males and females in each card type with respect to the total number of transactions.

```sql
-- contribution percentage number of males and females in each card type with respect to total number of transactions
with cte as (
select * from (
select card_type, gender, count(transaction_date) as count_of_trans
from credit_card_transactions
group by card_type, gender
order by card_type, gender) a , (select count(*) as total_count_of_trans from credit_card_transactions ) b )

select card_type, gender, round((count_of_trans*1.0/total_count_of_trans) *100, 2) as contribution_percentage
from cte ;
```

| card_type | gender | contribution_percentage |
|---|---|---|
| Gold | F | 12.62 |
| Gold | M | 11.82 |
| Platinum | F | 12.48 |
| Platinum | M | 12.08 |
| Signature | F | 12.93 |
| Signature | M | 11.82 |
| Silver | F | 14.48 |
| Silver | M | 11.77 |

The output table shows the contributions percentage to the total amount by Male and Female to each of the card types.

## 12. During weekends which city has the highest total spend to total no of transactions ratio ?

```sql
-- during weekends which city has highest total spend to total no of transcations ratio
select * from credit_card_transactions;
select city, round((sum(amount)*1.0/count(*))*100) as ratio
from credit_card_transactions
where date_format(transaction_date, "%W") in ( "Sunday", "Saturday")
group by city
order by round((sum(amount)*1.0/count(*))*100, 2) desc
limit 1;
```

| city | ratio |
|---|---|
| Sonepur, India | 29990500 |

During the weekends, city Sonepur has the highest total spend to total no of transactions ratio.

## 13. Write a query to print 3 columns: city, highest_expense_type , lowest_expense_type.

```sql
-- write a query to print 3 columns: city, highest_expense_type , lowest_expense_type
with cte as (
select city, exp_type, sum(amount) as amount  from credit_card_transactions
group by city, exp_type order by city, exp_type )
, cte2 as (
select *,
dense_rank() over(partition by city order by amount desc) as rn_desc,
dense_rank() over(partition by city order by amount ) as rn_asc
from cte )

select a.city, a.exp_type as highest_expense_type, b.exp_type as lowest_expense_type from
(select city, exp_type from cte2 where rn_desc=1 ) a join
(select city, exp_type from cte2 where rn_asc=1) b
on a.city=b.city;
```

Result Grid

| city | highest_expense_type | lowest_expense_type |
|---|---|---|
| Achalpur, India | Grocery | Entertainment |
| Adilabad, India | Bills | Food |
| Adityapur, India | Food | Grocery |
| Adoni, India | Bills | Entertainment |
| Adoor, India | Fuel | Bills |

The output shows the highest and lowest expense type for each of the cities.

## 14. Write a query to find the city which had the lowest percentage spent for gold card type.

```sql
-- Write a query to find city which had the lowest percentage spent for gold card type

select * from credit_card_transactions;
select city from (
select city, sum(amount) as amount
from credit_card_transactions
where card_type = "Gold"
group by city) a , (select sum(amount) as total_amount from credit_card_transactions) b
order by (amount*1.0/total_amount)*100
limit 1 ;
```

| city |
|---|
| Dhamtari, India |

**Dhamtari** city has the lowest percentage spent for gold card type.

## 15. Which card and expense type combination saw the highest month over month growth in Jan-2014 ?

```sql
-- Which card and expense type combination saw the highest month over month growth in Jan-2014 ?
with cte as (
select * from credit_card_transactions
where transaction_date between "2013-12-01" and "2014-01-31"
order by transaction_date )
, cte2 as (
select * from (
select *, lead(amount, 1) over(partition by card_type, exp_type order by name_of_month desc ) as jan_amount from (
select card_type, exp_type, date_format(transaction_date, "%m") as name_of_month , sum(amount) as amount
from cte
group by card_type, exp_type, date_format(transaction_date, "%m") ) x ) y
where jan_amount is not null )

select card_type, exp_type, ((jan_amount*1.0/amount) - 1 ) *100 as MoM_Growth_Rate from cte2
order by ((jan_amount*1.0/amount) - 1 ) desc
limit 1;
```

| card_type | exp_type | MoM_Growth_Rate |
|---|---|---|
| Gold | Travel | 87.92008 |

'Gold' card type and 'Travel' expense type combination recorded the highest month over month growth(MoM) in Jan-2014.

## 16. Write a query to print the highest spend month and amount spent in that month for each card type.

```sql
-- Write a query to print the highest spend month and amount spent in that month for each card type.

select * from credit_card_transactions;
with cte as (
select *,
max(amount)  over(partition by card_type ) as max_amount from (
select card_type, date_format(transaction_date, "%Y-%m") as `year_month`,
sum(amount) as amount
from credit_card_transactions
group by card_type, date_format(transaction_date, "%Y-%m")  ) x )

select card_type,`year_month`, max_amount from cte
where max_amount=amount;
```

| card_type | year_month | max_amount |
|---|---|---|
| Gold | 2015-01 | 55455064 |
| Platinum | 2014-08 | 57936507 |
| Signature | 2013-12 | 58799522 |
| Silver | 2015-03 | 59723549 |

It can be observed in the output the maximum amount spent in each card type and month of this maximum amount recorded.

17. **Write a query to print top 5 cities with highest spends and their percentage contribution to total credit card spends and further within each city, find the percentage contribution of different card types.**

credit_card_script* | SQL File 33*

```sql
/* Write a query to print top 5 cities with highest spends and their percentage contribution to total credit card
spends and further within each city, find the percentage contribution of different card types. */
with cte1 as (
with cte as ( select city ,amount, round((amount*1.0/ total_amount) *100, 8) as contribution_percentage from
(select city, sum(amount) as amount from credit_card_transactions
group by city) a, (select sum(amount) as total_amount
from credit_card_transactions ) b )
select city,amount, contribution_percentage from (
select city,amount, contribution_percentage,
dense_rank() over(order by contribution_percentage desc) as rn from cte ) x where rn<=5 )

,cte2 as (select city, card_type, round(
((sum(Amount) over(partition by city, card_type))  *1.0/
(sum(amount) over(partition by city)))*100, 8) as contribution_percentage_of_card from (
select city, card_type, sum(Amount) as amount from credit_card_transactions group by city, card_type ) x )

select cte1.city,cte1.amount, cte1.contribution_percentage,
cte2.card_type, cte2.contribution_percentage_of_card
from cte1 join cte2 on cte1.city=cte2.city
```

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

| city | amount | contribution_percentage | card_type | contribution_percentage_of_card |
|---|---|---|---|---|
| Greater Mumbai, India | 576751476 | 14.15399 | Gold | 23.92417 |
| Greater Mumbai, India | 576751476 | 14.15399 | Platinum | 25.10140 |
| Greater Mumbai, India | 576751476 | 14.15399 | Signature | 23.90177 |
| Greater Mumbai, India | 576751476 | 14.15399 | Silver | 27.07266 |
| Bengaluru, India | 572326739 | 14.04540 | Gold | 23.56548 |

Result 112

## 18. Which city took the least number of days to reach its 500th transaction after the first transaction in that city ?

```sql
-- Which city took the least number of days to reach its 500th transaction after the first transaction in that city ?
with cte as (
select city, count(*) as no_of_transactions, min(transaction_date) as min_transaction_date,
max(transaction_date) as max_transaction_date
from credit_card_transactions group by city having  count(*)>=500 )

, cte2 as (
select *, count(*) over(partition by city order by transaction_date, time) as no_of_trans_in_city
from credit_card_transactions)

, cte3 as (
select cte.city,cte.no_of_transactions,cte.min_transaction_date, cte.max_transaction_date ,cte2.transaction_date,
cte2.time ,cte2.no_of_trans_in_city
from cte join cte2 on cte.city=cte2.city )
```

```sql

, cte4 as (
select city,min_transaction_date,max_transaction_date ,transaction_date,
min(days_diff) as min_days_taken_to_500th_transaction from
(select city,min_transaction_date,max_transaction_date,transaction_date,
timestampdiff(day, min_transaction_date, transaction_date) as days_diff, no_of_trans_in_city from
(select * from cte3 where no_of_trans_in_city=500 ) x ) y
group by city,min_transaction_date,max_transaction_date,transaction_date )

select city, min_transaction_date,transaction_date as 500th_transaction_date,
min_days_taken_to_500th_transaction as days_taken_to_500th_transaction from
(select *,dense_rank() over(order by min_days_taken_to_500th_transaction ) as rn from cte4 ) z
where rn=1 ;
```

| city | min_transaction_date | 500th_transaction_date | days_taken_to_500th_transaction |
|---|---|---|---|
| Bengaluru, India | 2013-10-04 | 2013-12-24 | 81 |

It can be observed that Bengaluru has taken the least number of days(81 days) out of all cities to record 500 th transaction from its very first transaction.

## 19. Write a query to print the transaction details for each card type when it reaches a cumulative of 10,00,000 total spends.

```sql
/* Write a query to print the transaction details for each card type when it reaches a
cumulative of 10,00,000 total spends */
select * from credit_card_transactions;
select city, transaction_date, time, transactionid,card_type, exp_type,gender, amount, cumulative_total from
(select *, first_value(cumulative_total)
over(partition by card_type  order by transaction_date, time rows between unbounded preceding and unbounded following)
as transaction_reaching_cum_1000000 from
(select *, sum(amount) over(partition by card_type order by transaction_date, time) as cumulative_total
from credit_card_transactions )  x
where cumulative_total>=1000000 ) y
where transaction_reaching_cum_1000000=cumulative_total;
```

| city | transaction_date | time | transactionid | card_type | exp_type | gender | amount | cumulative_total |
|---|---|---|---|---|---|---|---|---|
| Sultanganj, India | 2013-10-04 | 13:02:57 | b671cf91-49de-471a-bf33-7b39a2e32a3d | Gold | Entertainment | M | 131374 | 1076577 |
| Bengaluru, India | 2013-10-05 | 15:05:19 | 0e2a5429-6345-4ebf-9415-2fe1a3fa493b | Platinum | Food | M | 183626 | 1117014 |
| Delhi, India | 2013-10-04 | 15:17:57 | 4dcd7178-fc99-46d6-8406-c255572ca709 | Signature | Bills | F | 550782 | 1513551 |
| Bahadurgarh, India | 2013-10-04 | 17:15:58 | 27214f85-73f1-4417-acb6-5ae933a419b5 | Silver | Food | F | 91707 | 1031394 |

The output table shows transaction details(like date of transaction , city of the transaction and some other details for that transaction) for each card type when it recorded its cumulative transaction value of INR 10,00,000 .

For example, Gold card type has registered a cumulative total of more than 10,00,000 for the first time on 2013-10-04 at 13:02:57 in Sultanganj city and the type of expense of that transaction is Entertainment and value of that transaction is INR 131374.

# Conclusions :

In conclusion, the analysis of credit card spending in India has provided valuable insights into consumer behavior and financial trends across the nation. The key findings can be summarized as follows:

**Summary of Key Findings :**

Our examination of the dataset revealed compelling patterns and trends within credit card transactions. Notable figures and statistics have been identified, forming the basis for a comprehensive understanding of spending habits in India. City-wise expenditures, gender-specific spending, and prevalent card types were among the significant variables analyzed. The data unveiled distinct patterns, showcasing how individuals across different demographics engage in financial transactions.

**Business Implications :**

The implications of our findings extend to practical applications for businesses operating in the Indian market. Tailoring marketing strategies, product offerings, and financial services based on demographic spending patterns can lead to more impactful and customer-centric approaches.

**Limitations and Recommendations :**

While our analysis provides valuable insights, it is important to acknowledge certain limitations. Factors such as data constraints or assumptions made during the analysis should be considered. Future research could focus on expanding data sources and refining methodologies to enhance the depth and accuracy of future analyses.

**Overall Significance :**

I have answered some of the key business driven questions in credit card spendings in India by writing  SQL queries. The insights gained from answering these questions serves as a foundation for strategic decision-making and business intelligence.