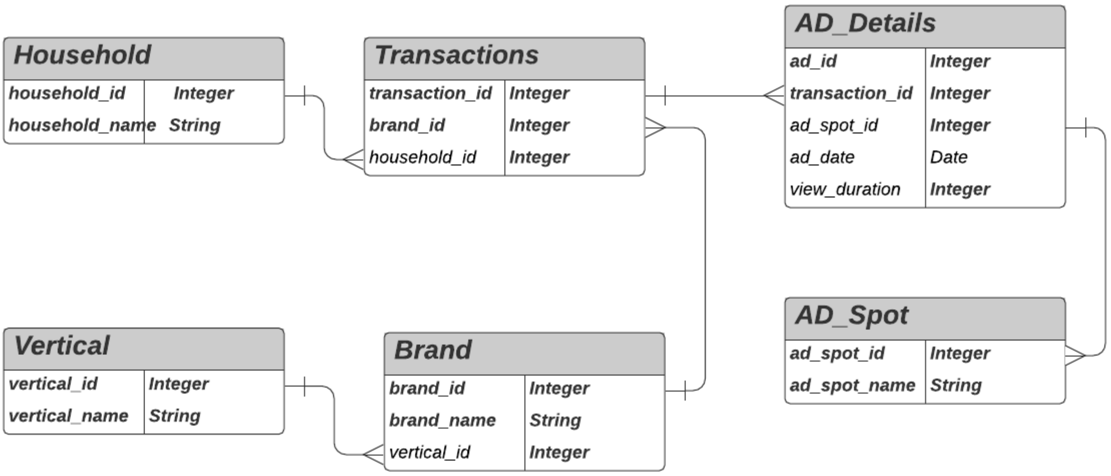
Database schema



Query for Creating tables:

```
create table Household (
household_id  int PRIMARY KEY,
household_name varchar(255)
);

create table Transactions (
transaction_id  int PRIMARY KEY AUTO_INCREMENT,
brand_id int FOREIGN KEY REFERENCES Brand(brand_id),
household id int
);

create table AD_Details (
ad_id int PRIMARY KEY AUTO_INCREMENT,
transaction_id int FOREIGN KEY REFERENCES Transactions(transaction_id),
ad_spot_id int FOREIGN KEY REFERENCES AD Spot(ad_spot_id),
ad_date date,
view_duration int

);

create table Vertical (
vertical_id int PRIMARY KEY,
vertical_name varchar(255)

);

create table Brand (
brand_id int PRIMARY KEY AUTO_INCREMENT,
brand_name varchar(255),
vertical_id int FOREIGN KEY REFERENCES Vertical(vertical_id),
);
```

```
create table AD_Spot (
ad_spot_id int PRIMARY KEY,
ad_spot_name varchar(255)
);
```

Process to insert the data

Initially, load the huge file into chunks of data (parallel processing) using the Python libraries. The records are inserted into StagingTable with following columns.

HouseholdId
Brand
Vertical
AD_SpotID
AD_Date
View_Duration
BrandID
TransactionID

1. Update the brand table if brand is not present:

```
Create procedure updateBrandDetails()
Begin
        INSERT INTO Brand (name,vertical_id)
        SELECT s.brand FROM staging_table s
        Inner join vertical v
        On s.vertical = v.vertical_name
        WHERE NOT EXISTS (SELECT * FROM Brand b
        WHERE b.name = s.brand)

End
```

2. First find the brandID using the below procedure.

```
Create procedure updateBrandID()
Begin
        Update staging_table s
                SET s.BrandID = (select b.brand_id
                From Brand b where b.name = s.Brand)
End
```

3. Now insert the transaction data into transaction table and update transaction_id in staging table:

```
Create procedure insertTrans()
Begin
        Insert into Transaction(brand_id, household_id)
        OUTPUT Inserted.TransactionID INTO @TransIds
        Select s.BrandID, s.HouseholdId from staging_table s
```

```
    Update staging_table s
        SET s.transactionID = (select TransactionID from TransIds)
End
```

4. Insert the AD details data into AD_Details table:

```
Create procedure insertAD()
Begin
    Insert into AD_Details (transaction_id, ad_spot_id,ad_date,view_duration)
    Select s.TransactionID, s. AD_SpotID, s. AD_Date, s.View_Duration from
    staging_table s
End
```

Retrieve the data

1. Number of Ads viewed

```
CREATE PROCEDURE NoOfAds()
BEGIN
    Select count (*) from AD_Details;
END
```

2. Number of Distinct Ads

```
CREATE PROCEDURE NoOfAds()
BEGIN
    Select count (DISTINCT ad_spot_id) FROM AD_Details;
END
```

3. Total duration of the Ads viewed (distributed by households_id)

```
CREATE PROCEDURE NoOfAds()
BEGIN
    Select t.households_id, sum (a.view_duration) from transactions as t
    Inner join AD_Details as a
    On t.transactionID = a.transactionID
    Group by t.household_id
END
```