Project – 3 : Exercise and Solution:

1. **Create a EMR cluster with Hadoop + Hive.**

Sol)

We will create the above in AWS with utilizing the services of EC2, EMR

First we will create a EC2 Instance and Key-pair to run EMR on EC2 instances. Once the key pair is generated we will download the .pem file this is used for SSH connection with the EC2 instance from the local system . The (.pem file) contains the public and primary key

- Public key (will be in EC2 instance) the Private key (Local Takes this key) thus SSh connection is established.
- Eg) the below connection is used for SSH connection with EMR in EC2 from local Terminal
- ssh -i ~/ProjectPro-EMR-Pyspark-Project.pem hadoop@ec2-35-183-45-198.ca-central-1.compute.amazonaws.com

2. **Create a S3 bucket, and a folder called sacrament_real_estate under it. Put the csv data file in this location.**

Sol)

Now we have spinned the EMR cluster and now have to load the data to **Amazon S3** Bucket.

- So for this we are creating a **S3** bucket first with default permissions and inside the bucket we are creating a **folder** following the naming conventions and then uploading the data into the folder in the S3 bucket.
- S3://Bucket Name/Folder Name/
- The below is the location in the S3 bucket.
- s3://p3-projectpro/input-data/

3. **ssh into EMR master node, and get into hive terminal. Under hive, in the default schema, create an external table "sacramento_real_estate_external" containing all the columns from the csv data, and associate String data type with them.**

The columns in the csv data file are:
street, city, zip, state, beds, baths, sq__ft, type, sale_date, price, latitude, longitude

Sol)

- ssh -i ~/ProjectPro-EMR-Pyspark-Project.pem hadoop@ec2-35-183-45-198.ca-central-1.compute.amazonaws.com

```
CREATE EXTERNAL TABLE sacramento_real_estate_external (
  street STRING,
  city STRING,
  zip STRING,
  state STRING,
  beds STRING,
  baths STRING,
  sq__ft STRING,
  type STRING,
  sale_date STRING,
  price STRING,
  latitude STRING,
  longitude STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION '/data/input;
```

4. **Now create another table "sacramento_real_estate_final" with the following columns and their corresponding data types partitioned by sale_date in dd-MM-yyyy format:**

Column	**Data Type**
street	string
city	string
zip	int
state	string

beds	int
baths	int
sq__ft	int
type	string
sale_date	string (“dd-MM-yyyy”)
price	int
latitude	float
longitude	float

Sol)

```
CREATE TABLE sacramento_real_estate_final (
  street STRING,
  city STRING,
  zip INT,
  state STRING,
  beds INT,
  baths INT,
  sq__ft INT,
  type STRING,
  price INT,
  latitude FLOAT,
  longitude FLOAT
)
PARTITIONED BY (sale_date STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS PARQUET;
```

Now from here when we are inserting the data we can modify the date format

```
INSERT OVERWRITE TABLE sacramento_real_estate_final PARTITION (sale_date)
SELECT
  street,
  city,
  CAST(zip AS INT),
  state,
  CAST(beds AS INT),
  CAST(baths AS INT),
  CAST(sq__ft AS INT),
  type,
  CAST(price AS INT),
  CAST(latitude AS FLOAT),
  CAST(longitude AS FLOAT),
  from_unixtime(unix_timestamp(sale_date, 'MM/dd/yyyy'), 'dd-MM-yyyy') as sale_date
FROM sacramento_real_estate_external;
```

5. **Insert the data from “sacramento_real_estate_external” table into “sacramento_real_estate_final” table.**

Note: You can choose to insert lesser rows in case you get errors while inserting the complete data set.

Sol)

```
INSERT INTO sacramento_real_estate_final PARTITION (sale_date)
SELECT
  street,
  city,
  CAST(zip AS INT),
  state,
  CAST(beds AS INT),
  CAST(baths AS INT),
  CAST(sq__ft AS INT),
  type,
  CAST(price AS INT),
  CAST(latitude AS FLOAT),
  CAST(longitude AS FLOAT),
  from_unixtime(unix_timestamp(sale_date, 'MM/dd/yyyy'), 'dd-MM-yyyy') as sale_date
FROM sacramento_real_estate_external
LIMIT 100;
```

Or instead of Limit we can use any Filter conditions based on the business Requirement