Tejas Babu

Extract, transform, load (ETL) is the method of copying data from one or more sources into a destination system

ETL Project

Contents

[**Introduction of ETL:** 1](#_Toc194141)

[**Technology Used:** 2](#_Toc194142)

[**Sql to create MySQL database and table:** 2](#_Toc194143)

[**Jupyter notebook for projects** 6](#_Toc194144)

[**Sample Graph using Matplotlib with loaded data:** 6](#_Toc194145)

[**A.** **Investment from another City which in Top 21 list** 7](#_Toc194146)

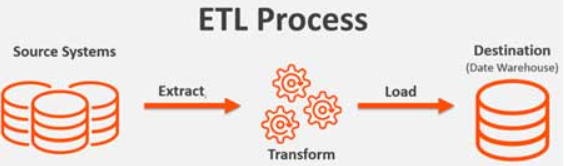
[**B. List of Zip code with highest Assessed Value in Houston (in 100000000)** 7](#_Toc194147)

[**C. List of City with Highest Appraised Value in HCAD except Houston (in 100000)** 8](#_Toc194148)

# **Introduction of ETL:**

Extract, transform, load (ETL) is the general procedure of copying data from one or more sources into a destination system which represents the data differently from the source. The term comes from the three basic steps needed: extracting (selecting and exporting) data from the source, transforming the way the data is represented to the form expected by the destination, and loading (reading or importing) the transformed data into the destination system.

Data is key for any business to succeed. Business can use ETL process to gather data from various sources, clean data with data scrubbing process, validate, transform in necessary format using Cleaning, Formatting, validation checks, Restructuring, Deduplication, Filtering, Merging, Splitting, Derivation, Summarization, Integration, Aggregation, Complex data validation and finally load transformed data into repository.



1. Scope of current ETL Projects:

For this project we are using subset of public data provided by hcad .

* Real Property

1. Real\_acct.csv
2. Real\_neighbourhood\_code.csv
3. Owners.csv
4. Deeds.csv

* Personal Property

1. T\_business\_acct.csv
2. T\_business\_detail.csv
3. T\_jur\_exempt.csv

For this project we will only read, csv file using Jupyter notebook, keep only necessary column, clean it if required and load it into MySQL database. And generate report from MySQL database using loaded data. We also plan to generate some Graph using MetPlotlib.

# **Technology Used:**

MySQL,

Pandas,

Python,

SqlAlchemy,

Jupyter Notebook,

Matplotlib

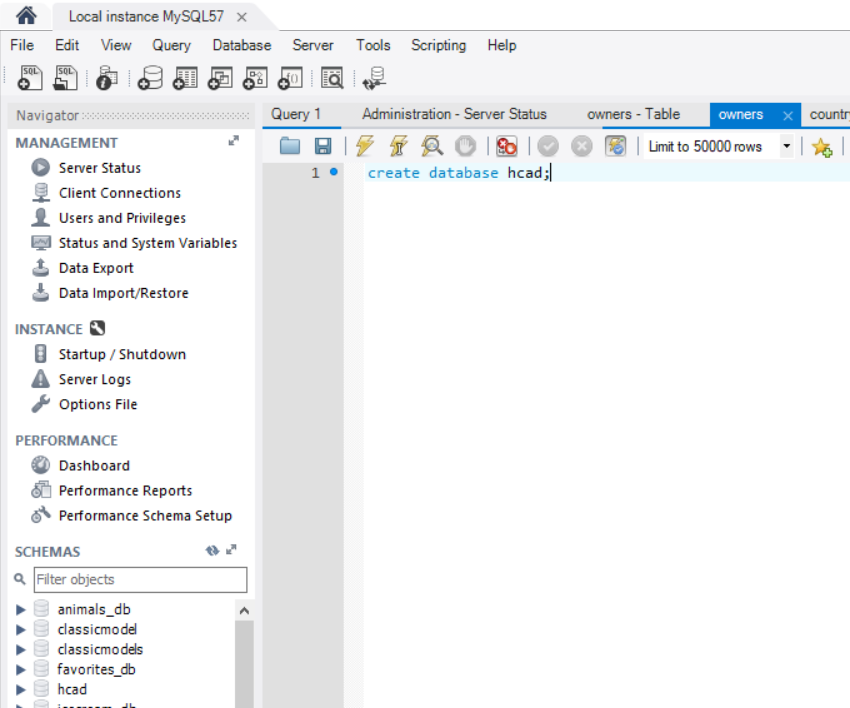
Excel

# **Sql to create MySQL database and table:**

Creating HCAD database:

1. Connect to MySQL using MySQL Work Bench
2. Run following Sql as show in image

create database hcad;



### **Tables:**

CREATE TABLE `hcad`.`real\_acct` (

`Account` INT NOT NULL,

`TaxYear` INT NULL,

`Mailto` VARCHAR(450) NULL,

`MailAddr` VARCHAR(450) NULL,

`City` VARCHAR(100) NULL,

`State` VARCHAR(100) NULL,

`zipcode` VARCHAR(45) NULL,

`Neighbourhood` VARCHAR(45) NULL,

`Total\_land` INT NULL,

`LandValue` FLOAT NULL,

`Assessed\_value` FLOAT NULL,

PRIMARY KEY (`Account`));

use hcad;

ALTER TABLE real\_Acct MODIFY COLUMN Account BIGINT ;

CREATE TABLE `hcad`.`owner` (

`Account` INT NOT NULL,

`Name` VARCHAR(450) NULL,

PRIMARY KEY (`Account`));

use hcad;

ALTER TABLE owner MODIFY COLUMN Account BIGINT ;

CREATE TABLE `hcad`.`deeds` (

`Account` INT NOT NULL,

`DateOfSale` VARCHAR(15) NULL,

`Clerk\_id` VARCHAR(45) NULL,

`deed\_id` VARCHAR(2) NULL,

PRIMARY KEY (`Account`));

use hcad;

ALTER TABLE deeds MODIFY COLUMN Account BIGINT ;

CREATE TABLE `hcad`.`neighbourhood` (

`neighbourhood\_cd` DECIMAL(8,2) NOT NULL,

`group\_cd` INT(11) NULL,

`description` VARCHAR(300) NULL,

PRIMARY KEY (`neighbourhood\_cd`));

Once you load data using Jupyter notebook then run below sql statement.

CREATE TABLE `hcad`.`business\_acct` (

`ACCOUNT` BIGINT(32) NOT NULL,

`TAX\_YEAR` INT NULL,

`O\_NAME` VARCHAR(400) NULL,

`OWNER` VARCHAR(400) NULL,

`SITE\_ADDRESS` VARCHAR(400) NULL,

`SITE\_CITY` VARCHAR(400) NULL,

`SITE\_STATE` VARCHAR(400) NULL,

`APPRAISED\_VALUE` VARCHAR(100) NULL,

`PRIOR\_APPRAISED\_VALUE` VARCHAR(100) NULL,

PRIMARY KEY (`ACCOUNT`));

Once you load data clean up APPRAISED\_VALUE using below statement

SET SQL\_SAFE\_UPDATES = 0;

Use hcad;

delete FROM hcad.business\_acct where Appraised\_value like 'Y%';

delete FROM hcad.business\_acct where appraised\_value = ' ';

delete FROM hcad.business\_acct where appraised\_value like 'L%';

delete FROM hcad.business\_acct where appraised\_value like 'I%';

delete from hcad.real\_acct where zipcode like '-%';

delete from hcad.real\_acct where zipcode = ' ';

delete from neighbourhood where description = ' ';

commit;

ALTER TABLE `hcad`.`business\_acct`

CHANGE COLUMN `APPRAISED\_VALUE` `APPRAISED\_VALUE` INT(11) NULL DEFAULT NULL ;

ALTER TABLE `hcad`.`business\_acct`

CHANGE COLUMN `PRIOR\_APPRAISED\_VALUE` `PRIOR\_APPRAISED\_VALUE` INT(11) NULL DEFAULT NULL ;

CREATE TABLE `hcad`

.`business\_detail` (

`Account` INT(11) NOT NULL,

`Dept\_code` VARCHAR(15) NULL,

`Dept\_code\_de` VARCHAR(150) NULL,

PRIMARY KEY (`Account`));

CREATE TABLE `hcad`.`exempt\_data` (

`Account` INT(11) NOT NULL,

`Tax\_dist` INT(11) NULL,

`Tax\_dist\_name` VARCHAR(400) NULL,

`Exempt\_cat` VARCHAR(45) NULL,

`Exempt\_dsc` VARCHAR(45) NULL,

PRIMARY KEY (`Account`));

### **Views:**

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `hcad`.`investment` AS

SELECT

(SUM(`hcad`.`real\_acct`.`LandValue`) / 100000) AS `Value`,

`hcad`.`real\_acct`.`City` AS `city`

FROM

`hcad`.`real\_acct`

WHERE

(`hcad`.`real\_acct`.`City` <> 'HOUSTON')

GROUP BY `hcad`.`real\_acct`.`City`

HAVING ((SUM(`hcad`.`real\_acct`.`LandValue`) / 100000) > 100);

USE `hcad`;

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `hcad`.`assesssedvalue` AS

SELECT

(SUM(`hcad`.`real\_acct`.`Assessed\_value`) / 100000) AS `sum(Assessed\_value)/100000`,

`hcad`.`real\_acct`.`zipcode` AS `zipcode`

FROM

`hcad`.`real\_acct`

GROUP BY `hcad`.`real\_acct`.`zipcode`

HAVING ((SUM(`hcad`.`real\_acct`.`Assessed\_value`) / 100000) > 1000);

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `hcad`.`appraisedvalue` AS

SELECT

(SUM(`hcad`.`business\_acct`.`APPRAISED\_VALUE`) / 100000) AS `sum(APPRAISED\_VALUE)`,

`hcad`.`business\_acct`.`SITE\_CITY` AS `city`

FROM

`hcad`.`business\_acct`

WHERE

((`hcad`.`business\_acct`.`SITE\_CITY` <> ' ')

AND (`hcad`.`business\_acct`.`SITE\_CITY` <> 'HOUSTON'))

GROUP BY `hcad`.`business\_acct`.`SITE\_CITY`

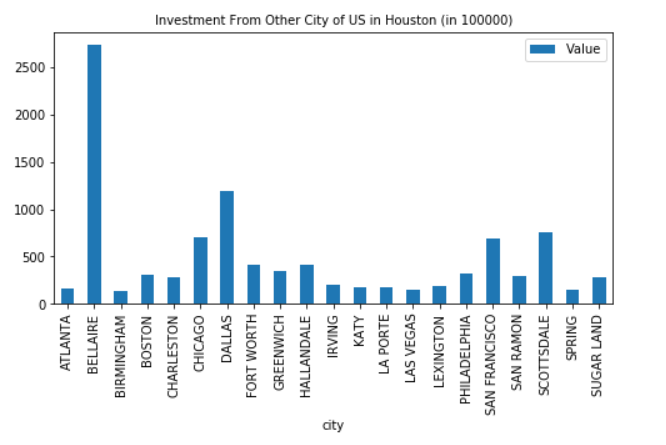
HAVING ((SUM(`hcad`.`business\_acct`.`APPRAISED\_VALUE`) / 100000) > 1000);

# **Jupyter notebook for projects**

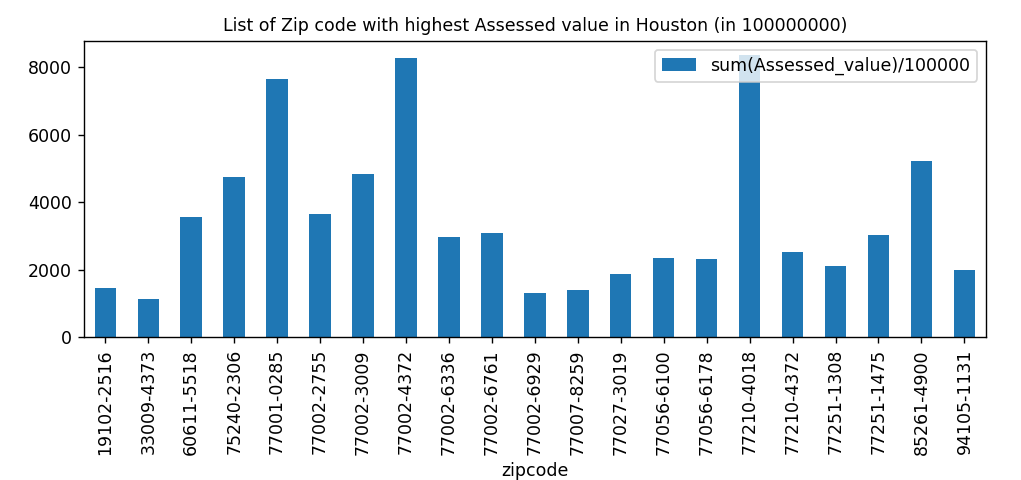
Refer GIT Hub repository****

# **Sample Graph using Matplotlib with loaded data:**

## **Investment from another City which in Top 21 list**



## **B. List of Zip code with highest Assessed Value in Houston (in 100000000)**



## **C. List of City with Highest Appraised Value in HCAD except Houston (in 100000)**

