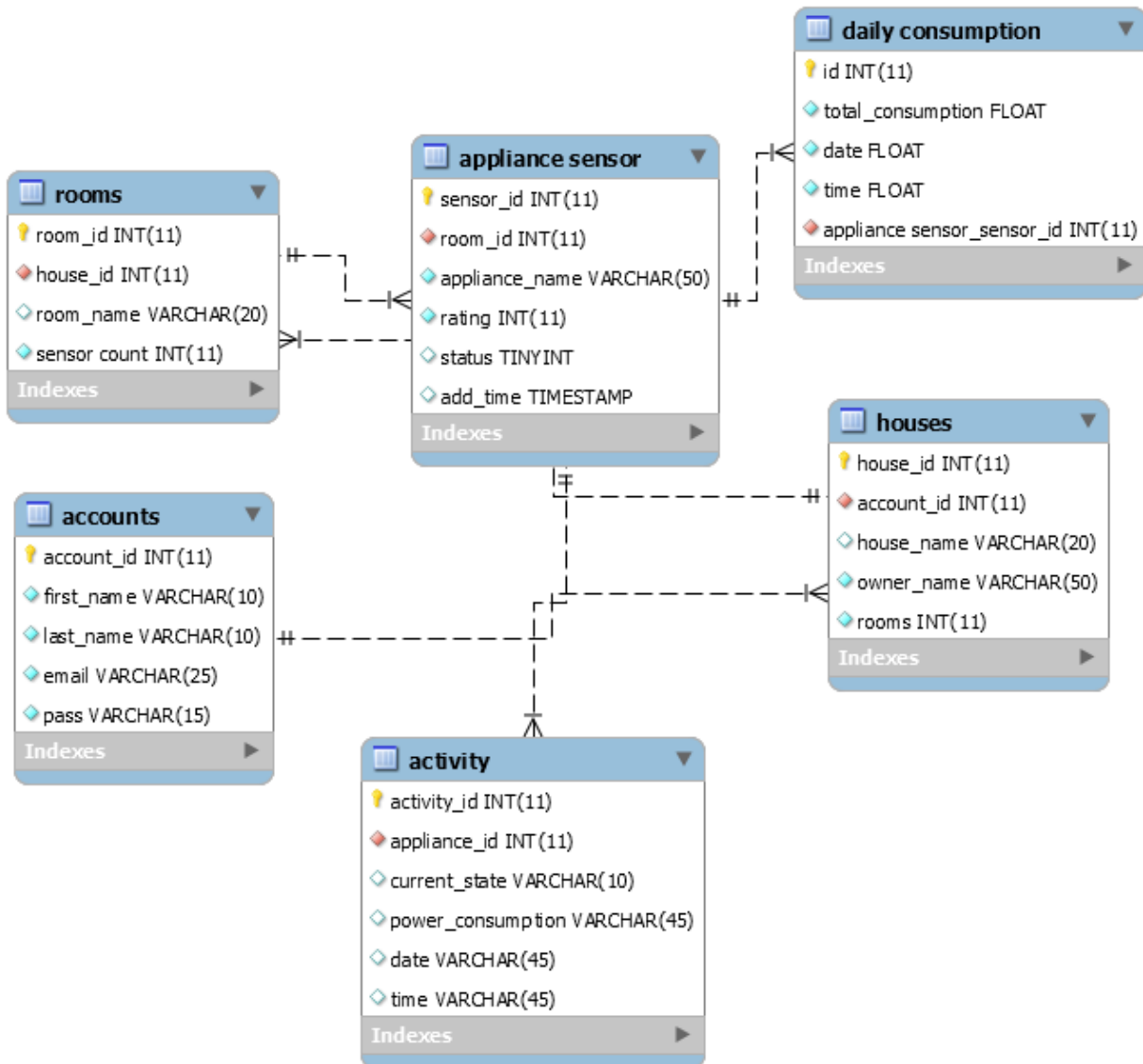


Smart Home Analytics



Smart Home Analytics ERD

Data Collection:

For the project we found a data set which we used to extract data according to our need.

<https://www.kaggle.com/taranvee/smart-home-dataset-with-weather-information>

date	time	timestamp	use [kW]	gen [kW]	House overall	Dishwasher [k]	Furnace 1 [k]	Furnace 2 [k]	Home office [k]	Fridge [kW]	Wine cellar [k]	Garage do	Kitchen 1	Kitchen 14	Kitchen 3	Barn [kW]	Well
Friday, Jan	5:00:00 AM	1.45E+09	0.93283333	0.003483333	0.932833333	3.33E-05	0.0207	0.0619167	0.442633333	0.12415	0.006983333	0.013083	0.000417	0.00015	0	0.03135	0.0
Friday, Jan	5:00:01 AM	1.45E+09	0.93433333	0.003466667	0.934333333	0	0.02071667	0.0638167	0.444066667	0.124	0.006983333	0.013117	0.000417	0.00015	0	0.03135	0.0
Friday, Jan	5:00:02 AM	1.45E+09	0.93181667	0.003466667	0.931816667	1.67E-05	0.0207	0.0623167	0.446066667	0.123533333	0.006983333	0.013083	0.000433	0.000167	1.67E-05	0.031517	0.0
Friday, Jan	5:00:03 AM	1.45E+09	1.02205	0.003483333	1.02205	1.67E-05	0.1069	0.0685167	0.446583333	0.123133333	0.006983333	0.013	0.000433	0.000217	0	0.0315	0.0
Friday, Jan	5:00:04 AM	1.45E+09	1.1394	0.003466667	1.1394	0.000133333	0.23693333	0.0639833	0.446533333	0.12285	0.00685	0.012783	0.00045	0.000333	0	0.0315	0.0
Friday, Jan	5:00:05 AM	1.45E+09	1.39186667	0.003433333	1.391866667	0.000283333	0.50325	0.0636667	0.447033333	0.1223	0.006716667	0.012433	0.000483	0.000567	0	0.03145	0.0
Friday, Jan	5:00:06 AM	1.45E+09	1.36621667	0.00345	1.366216667	0.000283333	0.4994	0.0637167	0.443266667	0.12205	0.006733333	0.012417	0.000517	0.00055	0	0.03155	0.0
Friday, Jan	5:00:07 AM	1.45E+09	1.4319	0.003416667	1.4319	0.00025	0.47786667	0.1786333	0.444283333	0.1218	0.006783333	0.01255	0.000483	0.00045	0	0.031733	0.0
Friday, Jan	5:00:08 AM	1.45E+09	1.6273	0.003416667	1.6273	0.000183333	0.44765	0.3657	0.441466667	0.121616667	0.00695	0.012717	0.000467	0.0003	1.67E-05	0.031767	0.0
Friday, Jan	5:00:09 AM	1.45E+09	1.73538333	0.003416667	1.735383333	1.67E-05	0.17155	0.6825	0.438733333	0.121633333	0.007233333	0.01335	0.000367	5.00E-05	0	0.031667	0.0
Friday, Jan	5:00:10 AM	1.45E+09	1.58508333	0.003416667	1.585083333	5.00E-05	0.0221	0.6787333	0.4402	0.12145	0.007433333	0.013583	0.00035	0.000117	3.33E-05	0.031667	0.0
Friday, Jan	5:00:11 AM	1.45E+09	1.51031667	0.003433333	1.510316667	3.33E-05	0.02196667	0.6206667	0.43695	0.12125	0.007316667	0.013533	0.000333	0.0001	0	0.03175	0.0
Friday, Jan	5:00:12 AM	1.45E+09	1.45986667	0.00345	1.459866667	5.00E-05	0.02188333	0.5774667	0.43995	0.121033333	0.007233333	0.013517	0.000367	8.33E-05	1.67E-05	0.031783	0.0
Friday, Jan	5:00:13 AM	1.45E+09	0.84058333	0.003433333	0.840583333	0	0.02095	0.1448	0.444783333	0.035016667	0.007033333	0.013183	0.00065	0.000183	1.67E-05	0.031783	0.0
Friday, Jan	5:00:14 AM	1.45E+09	0.7032	0.003433333	0.7032	1.67E-05	0.02073333	0.0619667	0.443833333	0.004783333	0.006966667	0.013117	0.000733	0.000233	0	0.03175	0.0
Friday, Jan	5:00:15 AM	1.45E+09	0.57188333	0.00345	0.571883333	0	0.02065	0.06365	0.307783333	0.004916667	0.00705	0.0131	0.000733	0.00015	0	0.031733	0.0
Friday, Jan	5:00:16 AM	1.45E+09	0.48573333	0.00345	0.485733333	1.67E-05	0.02061667	0.0634333	0.22045	0.004983333	0.007033333	0.013117	0.00075	8.33E-05	0	0.031833	0.0
Friday, Jan	5:00:17 AM	1.45E+09	0.52316667	0.003433333	0.523166667	0	0.02063333	0.0621167	0.26005	0.00495	0.007	0.013083	0.000733	0.0001	1.67E-05	0.03185	0.0
Friday, Jan	5:00:18 AM	1.45E+09	0.5362	0.00345	0.5362	0	0.02068333	0.0629167	0.272066667	0.00495	0.007033333	0.01315	0.000733	0.000117	0	0.031867	0.0
Friday, Jan	5:00:19 AM	1.45E+09	0.53415	0.00345	0.53415	1.67E-05	0.02066667	0.06265	0.270066667	0.00495	0.0071	0.01315	0.000733	0.0001	0	0.0319	0.0
Friday, Jan	5:00:20 AM	1.45E+09	0.53381667	0.00345	0.533816667	0	0.02063333	0.0629667	0.270033333	0.00495	0.007033333	0.013117	0.00075	0.0001	0	0.031817	0.0
Friday, Jan	5:00:21 AM	1.45E+09	0.52363333	0.00345	0.523633333	1.67E-05	0.02055	0.0632833	0.259816667	0.00495	0.007033333	0.013083	0.000733	0.000117	0	0.031733	0.0
Friday, Jan	5:00:22 AM	1.45E+09	0.57725	0.003416667	0.57725	0	0.02068333	0.10975	0.257	0.004983333	0.007066667	0.013133	0.000733	8.33E-05	0	0.031683	0.0
Friday, Jan	5:00:23 AM	1.45E+09	0.67956667	0.003433333	0.679566667	1.67E-05	0.02086667	0.1940833	0.2571	0.005016667	0.007116667	0.0132	0.0007	6.67E-05	1.67E-05	0.031717	0.0
Friday, Jan	5:00:24 AM	1.45E+09	1.29316667	0.0034	1.293166667	1.67E-05	0.10743333	0.6224667	0.2541	0.0052	0.00725	0.01335	0.00065	0.0001	3.33E-05	0.031567	0.0
Friday, Jan	5:00:25 AM	1.45E+09	1.54666667	0.003433333	1.546666667	1.67E-05	0.22056667	0.68005	0.253933333	0.00555	0.007266667	0.013183	0.0007	6.67E-05	1.67E-05	0.031067	0.0

The raw data is labelled as Home.csv in our submission file

One row of the raw data represented one second instance of the electricity consumption of a home. The raw data had 0.5 Million rows of data for a course of 7 days. Every device had a different column like dish washer and fridge had separates columns for their consumption.

To extract the data according to our need, we took 5 appliances data and distributed them to 3 different rooms, like fridge, microwave and dishwasher were allocated to Kitchen and home office and living room had only one appliance data in it.

For generation of our data according to our ERD, we used python to extract the data and compiled them accordingly. Each room was given a room id and each room had appliances and were given a appliance id.

```
#assign sensor ID
dishwasher_id = 0
fridge_id = 1
microwave_id = 2
livingroom_id = 3
home_office_id = 4
```

The appliance columns were loaded into python list structure:

```
#Kitchen appliances Sensor Data  
dishwasher = df['Dishwasher [kW]']  
fridge = df['Fridge [kW]']  
microwave = df['Microwave [kW]']  
livingroom = df['Living room [kW]']  
home_office = df['Home office [kW]']  
date_load = df['date']  
time_load = df['time']
```

A final list was created which had all the power consumption of all the 5 appliances using the following python script:

```
power_consumption = []  
appliance_id = []  
activity_ids = []  
date = []  
time = []  
startingid= 0  
for x in range(len(dishwasher)):  
    power_consumption.append(dishwasher[x])  
    appliance_id.append(dishwasher_id)  
    activity_ids.append(startingid)  
    date.append(date_load[x])  
    time.append(time_load[x])  
    startingid+=1  
    power_consumption.append(fridge[x])  
    appliance_id.append(fridge_id)  
    activity_ids.append(startingid)  
    date.append(date_load[x])  
    time.append(time_load[x])  
    startingid+=1  
    power_consumption.append(microwave[x])  
    appliance_id.append(microwave_id)  
    activity_ids.append(startingid)  
    date.append(date_load[x])  
    time.append(time_load[x])  
    startingid+=1  
    power_consumption.append(livingroom[x])  
    appliance_id.append(livingroom_id)  
    activity_ids.append(startingid)  
    date.append(date_load[x])  
    time.append(time_load[x])  
    startingid+=1  
    power_consumption.append(home_office[x])  
    appliance_id.append(home_office_id)  
    activity_ids.append(startingid)  
    date.append(date_load[x])  
    time.append(time_load[x])  
    startingid+=1
```

The final activity table data looked like this:

	activity_id	appliance_id	current_state	power_consumption	date	time
0	0	0	NaN	0.000033	Friday, January 1, 2016	5:00:00 AM
1	1	1	NaN	0.124150	Friday, January 1, 2016	5:00:00 AM
2	2	2	NaN	0.004067	Friday, January 1, 2016	5:00:00 AM
3	3	3	NaN	0.001517	Friday, January 1, 2016	5:00:00 AM
4	4	4	NaN	0.442633	Friday, January 1, 2016	5:00:00 AM

We also created a daily power consumption table using the following python code:

```
start = 1
daily_consumption = []
daily_id = []
sensor_id=[]
id= 0
for x in range(6):

    daily_consumption.append(np.sum(dishwasher[86400*x:86400*(x+1)]))
    sensor_id.append(0)
    daily_id.append(id)
    id+=1
    daily_consumption.append(np.sum(fridge[86400*x:86400*(x+1)]))
    sensor_id.append(1)
    daily_id.append(id)
    id+=1
    daily_consumption.append( np.sum(microwave[86400*x:86400*(x+1)]))
    sensor_id.append(2)
    daily_id.append(id)
    id+=1
    daily_consumption.append(np.sum(livingroom[86400*x:86400*(x+1)]))
    sensor_id.append(3)
    daily_id.append(id)
    id+=1
    daily_consumption.append(np.sum(home_office[86400*x:86400*(x+1)]))
    sensor_id.append(4)
    daily_id.append(id)
    id+=1

print(len(daily_consumption))
print(len(sensor_id))
print(sensor_id)
```

Apart from activity and daily consumption table, accounts, house, appliance and rooms csv was created manually. All the final csv files can be found in the folder of **Project_Dataset**.