

## Table of Contents:

[Main menu](#)

[Enter Household information](#)

[Add Appliance](#)

[Appliance List](#)

[Add power generation](#)

[Power generation List](#)

[Wrapping up](#)

[View report/query data](#)

[View Top 25 popular manufacturers](#)

[Manufacturer drilldown report request](#)

[Manufacturer/model search](#)

[Heating/cooling method details](#)

[Water heater statistics by state](#)

[State drilldown report Request](#)

[Off-the-grid household dashboard](#)

[Household averages by radius](#)

## Main menu

### Abstract Code

- Showing “**Enter household info**” and “**View report/query data**” links.
- Upon:
  - Click **Enter household info** link- Jump to the **Household information** task.
  - Click **View report/query data** link- Jump to the **View report/query data** task.

## Enter household info

### Abstract Code

- Populate dropdowns for home type and utilities

```
SELECT home_type FROM 'Household_type';
```

```
SELECT utility_type FROM 'UtilityType';
```

- User enters *email* (**\$Email**), *postal code* (**\$postal code number**), *square footage* input fields, either enter *thermostat setting for heating and cooling* input fields or select *no heat* or *no cooling*, select *home type*, *public utilities*.
- Enter *email*:

```
SELECT email FROM 'Household' WHERE email='$Email';
```

- If Household record is found, exist email message is shown.
- Else if or invalid email format, error message is shown.
- Else, continue entering other information.
- Enter *postal code*:

```
SELECT 'Zip Code' number FROM 'postal code' WHERE  
Zip code = '$postal code number';
```

## Phase 2 Abstract Code w/SQL | CS 6400 - Spring 2023 | Team 098

- If postal code number is not found or invalid format, error message is shown.
- Else, continue entering other information.
- Enter Click **Next** button, the household information is saved in database, jump to the **Add Appliances** task.

```
INSERT INTO Household (email, sq_ft, type, cool_setting, heat_setting,  
zip_code) VALUES ($email, $sq_ft, $type, $cool_setting, $heat_setting,  
$zip_code);
```

### Add Appliance

#### Abstract Code

- Populate dropdown for appliance type and manufacturer

```
SELECT App_type FROM 'Appliance_type';
```

- Select *manufacturer*.

```
SELECT Name FROM 'manufacturer';
```

- User selects *appliance type*, *manufacturer*, enters *Model name*.
- Select Appliance type:
- if *Air handler* is selected the *Air conditioner*, *Heater*, and *Heat pump* check boxes are displayed :
  - Heating/cooling method radios are shown:
    - *Air conditioner* is checked: user enters *BTUs*, *Energy efficiency ratio* (EER).
    - *Heater* is checked: user enters *BTUs*, select *Energy source*.
    - *Heat pump* is checked: user enters *BTUs*, *seasonal performance factor* (HSPF).

## Phase 2 Abstract Code w/SQL | CS 6400 - Spring 2023 | Team 098

- User finishes input and clicks **Add**

```
INSERT INTO Appliance (email, app_order, app_type, model_name, btu_rating,
manufacturer) VALUES ($email, $app_order, $app_type, $model_name,
$btu_rating, $manufacturer);
```

- Depending on users options the type of appliance is added to it's corresponding relation

```
INSERT INTO AirConditioner(email, app_order, energy_efficiency_ratio)
VALUES ($email, $app_order, $ energy_efficiency_ratio);
```

```
INSERT INTO Heater (email, app_order, energy_source)
VALUES ($email, $app_order, $ energy_source);
```

```
INSERT INTO HeaterPump(email, app_order, seasonal_efficiency_ratio,
heat_performance_factor) VALUES ($email, $app_order,
$ seasonal_efficiency_ratio, $heat_performance_factor);
```

- if *water heater* is selected:
  - User selects *Energy source*, enters *capacity(gallons)*, *BTU rating* and *temperatures*.

```
INSERT INTO WaterHeater(email, app_order, capacity, temp_setting,
energy_source) VALUES ($email, $app_order, $ capacity, $temp_setting,
$energy_source);
```

- Enter *Model name* is optional and the text box will be populated with grey "optional" input that the user can write over if they choose.
- Click **Add** button: save data into database and jump to **view/delete appliance list** task.

## Appliance Listing

### Abstract Code

- display list of added appliances:

## Phase 2 Abstract Code w/SQL | CS 6400 - Spring 2023 | Team 098

```
SELECT Appl_order# AS 'Appliance number', Appliance Type AS Type, Manufacturer, Model_Name AS Model FROM 'Appliance' WHERE email='$Email';
```

- Click **delete** button:

```
DELETE FROM 'Appliance' WHERE email='$Email' AND Appliance #='$ Appl_order #';
```

- When Click **+Add another appliance** link:
  - Go back to **Add Appliance** task.
- Click **Next** button:
  - Jump to the **Add Power generation** task.

### Add power generation

#### Abstract Code

- Populate power generation type

```
SELECT Power_gen_type FROM PowerGenerationType;
```

- If user selects **public utilities** in **Enter Household Info** form, Skip button is shown:
  - When click **Skip** button- finish submitting data and jump to **Wrapping up** form.

```
SELECT email FROM 'Utility' WHERE email = $email;
```

- Else, the **Skip** button is hide for “off-grid household” and power generation information is required.
- User selects *power generation type*, enters *monthly kwh* and *storage kwh* (optional).
- When click **Add** button:
- save data into database and jump to **power generation** task.

## Phase 2 Abstract Code w/SQL | CS 6400 - Spring 2023 | Team 098

```
INSERT INTO PowerGenerator(email, pg_order, generation_type, battery_storage,
average_kwh) VALUES ($email, $ pg_order, $ generation_type, $ battery_storage,
$ average_kwh);
```

### Power Generation List

#### Abstract Code

- Display list of added power generation:

```
SELECT pg_order AS Num, 'Generation type' AS Type, 'Average kwh' AS 'Monthly
kWh', 'Battery Storage' AS 'Battery kWh' FROM 'Power Generator' WHERE
email='$Email';
```

- When click **delete** button:

```
DELETE FROM 'Power Generator' WHERE email='$Email' AND Num='$pg_order#';
```

- When click **+Add more power** link:
  - Go back to the **Add power generation** task.
- When click **Finish** button:
  - finish submitting data and jump to **Wrapping Up** form.

### Wrapping Up

#### Abstract Code

- Display "Submission Complete!" message
- When click **Return to the main menu** button:
- Go back to **Main menu** form.

## View reports/query data

### Abstract Code

- Display Lists of reports links including “**Top 25 manufacturers**”, “**Manufacture/model search**”, “**Heating/cooling method detail**”, “**Water heater statistics by state**”, “**off-the-grid household dashboard**” and “**Household average by radius**”:
  - Click **Top 25 manufacturers** link- jump to **Top 25 manufacturers** form.
  - Click **Manufacture/model search** link- jump to **Manufacture/model search input** form.
  - Click **Heating/cooling method detail** link- jump to **Heating/cooling method detail** form.
  - Click **Water heater statistics by state** link- jump to **Water heater statistics by state** form.
  - Click **off-the-grid household dashboard** link- jump to **off-the-grid household dashboard** form.
  - Click **Household average by radius** link- jump to **Household average by radius input** form.

## Top 25 manufacturers

### Abstract Code

- display list of top 25 manufacturers with the most appliances:

```
SELECT 'Manufacturer name', COUNT ('Manufacturer name') FROM 'Appliance'  
GROUP BY 'Manufacturer name' ORDER BY COUNT('Manufacturer name') DESC  
LIMIT 25 ;
```

- User clicks **manufacturer name** link from the list-Jump to **Manufacturer drilldown report request** task.

## Manufacturer drilldown report request

### Abstract Code

- display table about count of appliance type for this manufacturer name:

```
SELECT 'Manufacturer name' AS title, 'Appliance Type', COUNT ('Appliance Type')  
FROM 'Appliance' WHERE 'Manufacturer name'='$Manufacturer name' GROUP BY  
'Appliance Type';
```

## Manufacturer/model search

### Abstract Code

- User enters any string in input fields.
- If data validation is successful for matching any part of a manufacturer name or model name, then:
  - User clicked on **submit** button, display table with manufacturers' name and its models' name; matched string must highlight with light green background:

```
SELECT 'Manufacturer name', 'Model name' FROM 'Appliance' WHERE LOWER ('Manufacturer name') LIKE '%value%' OR LOWER ('Model name') LIKE '%value%' ORDER BY 'Manufacturer name', 'Model Name' ASC;
```

- If user inputs invalid string, error message will be returned.

## Heating/cooling method detail

### Abstract Code

- Run **report of heating/cooling method detail** task:
  - Display table of count of air conditioners, with average BTUs and average EER:

```
SELECT Type, COUNT ('Energy Efficiency Ratio') AS 'Count of Air Conditioners', ROUND (AVERAGE (CAST (BTUs AS FLOAT)), 0) AS 'Average Air Conditioner BTUs', ROUND (AVERAGE (CAST ('Energy Efficiency Ration' AS FLOAT)), 1) AS 'Average EER' FROM ('Air Conditioner' NATURAL JOIN 'Appliance') NATURAL JOIN 'Household' GROUP BY Type;
```

- Display table list of count of heater, with average BTUs and most common energy source:

```
(SELECT Type, COUNT ('Energy Source') AS 'Counter of heaters', ROUND (AVERAGE (CAST (BTUs AS FLOAT)), 0) AS 'Average Heater BTUs' FROM ('Heater' NATURAL JOIN 'Appliance') NATURAL JOIN 'Household' GROUP BY Type) NATURAL JOIN (SELECT Type, MAX (COUNT ('Energy Source')) AS 'Most Common Energy Source' FROM ('Heater' NATURAL JOIN 'Appliance') NATURAL JOIN 'Household' GROUP BY Type, Energy Source);
```

- Display table of count of heat pumps, with average BTUs and average SEER and average HSPF:



## Phase 2 Abstract Code w/SQL | CS 6400 - Spring 2023 | Team 098

```
SELECT Type, COUNT (Email) AS 'Count of Heat Pumps', ROUND (AVERAGE  
(CAST (BTUs AS FLOAT)), 0) AS 'Average Heat Pumps BTUs', ROUND (AVERAGE  
(CAST ('Seasonal Energy Efficiency Ratio' AS FLOAT)), 1) AS 'Average SEER',  
ROUND (AVERAGE (CAST ('Heating Seasonal Performance Factor' AS FLOAT)), 1)  
AS 'Average HSPF' FROM ('Heat Pump' NATURAL JOIN 'Appliance') NATURAL JOIN  
'Household' GROUP BY Type;
```

### Water heater statistics by states

#### Abstract Code

- Run **water heater stats request** task:
  - Display table with water heater statistics for each state, including average water heater capacity, average BTUs, count of water heater w/o temperature setting:

```
SELECT State, IFNULL(ROUND (AVERAGE (CAST (Capacity AS FLOAT)), 0), 0) AS  
'Average Water heater Capacity', IFNULL( ROUND (AVERAGE (CAST (BTUs AS  
FLOAT)), 0), 0) AS 'Average Water Heater BTUs', IFNULL(ROUND (AVERAGE (CAST  
( 'Temperature Setting' AS FLOAT)), 1), 0) AS 'Average Water Heater Temperature  
Setting', IFNULL(COUNT ('Temperature Setting'), 0) AS 'Count of Water Heater with  
Temperature Setting', IFNULL( (COUNT(email) - COUNT ('Temperature Setting')), 0)  
AS 'Count of Water Heater w/o Temperature Setting' FROM ('Water Heater'  
NATURAL JOIN 'Household') NATURAL JOIN 'Postal Code' GROUP BY State  
ORDER BY State ASC;
```

- User clicks **state name** link-Jump to **state drilldown report request** task.

### State drilldown report Request

#### Abstract Code

- User clicks **state name** link, display table about minimum, average, and maximum of water heater capacity and temperature setting for each energy source in this state which will be table title:

```
SELECT state AS title, IFNULL(ROUND (MIN (CAST (Capacity AS FLOAT)), 0), 0) AS  
'Minimal Water heater Capacity', IFNULL(ROUND (AVERAGE (CAST (Capacity AS  
FLOAT)), 0), 0) AS 'Average Water Heater Capacity', IFNULL(ROUND (MAX (CAST  
(Capacity AS FLOAT)), 0), 0) AS 'Maximum Water Heater Capacity', IFNULL(ROUND  
(MIN (CAST ('Temp Setting' AS FLOAT)), 0), 0) AS 'Minimal Water heater Temperature  
Setting', IFNULL(ROUND (AVERAGE (CAST ('Temp Setting' AS FLOAT)), 0), 0) AS  
'Average Water Heater Temperature Setting', IFNULL(ROUND (MAX (CAST ('Temp  
Setting' AS FLOAT)), 0), 0) AS 'Maximum Water Heater Temperature Setting' FROM  
( 'Water Heater' NATURAL JOIN 'Household') NATURAL JOIN 'Postal Code' WHERE  
state = '$State' GROUP BY 'energy source' ASC;
```

## Off-the-grid household dashboard

### Abstract Code

- Display state with most off-the-grid household and count of its household:

```
SELECT State, COUNT (DISTINCT(email)) AS 'Count of household' FROM ('Power Generator' NATURAL JOIN 'Household') NATURAL JOIN 'Postal Code' GROUP BY State ORDER BY 'Count of household' DESC LIMIT 1;
```

- Display table of average battery storage capacity:

```
SELECT ROUND (AVERAGE (CAST ('Battery Storage' AS FLOAT)), 0) AS 'Average of battery storage capacity' FROM 'Power Generator';
```

- the percentage for each power generation types:

```
SELECT 'Generation Type', ROUND (SUM ('Battery Storage')/SUM (SUM ('Battery Storage')), 0) AS 'Percentage of battery storage capacity' FROM 'Power Generator' GROUP BY Generation Type ORDER BY 'Count of household' DESC;
```

- Display table of average water heater gallon capacity of all off-the-grid household and all on-the-grid household:

```
SELECT ROUND (AVERAGE (CAST (on-capacity AS FLOAT)), 1) AS Average Water Heater Gallon capacity of all on-the-grid, ROUND (AVERAGE (CAST (off-capacity AS FLOAT)), 1) AS 'Average Water Heater Gallon capacity of all off-the-grid' FROM (SELECT email, Appl_order#, Capacity AS on-capacity FROM 'Water Heater' AS WH INNER JOIN 'Utilities' AS U on WH.email = U.email) OUTER JOIN (SELECT email, Appl_order#, Capacity AS off-capacity FROM 'Water Heater' WHERE email NOT IN (SELECT email FROM 'Utilities'));
```

- Display table of Minimal, average and maximum BTUs for all off-the-grid households' appliances, grouped by appliance type:

```
SELECT ROUND (MIN (CAST (BTUs AS FLOAT)), 0) AS 'Minimum BTUs', ROUND (AVERAGE (CAST (BTUs AS FLOAT)), 0) AS 'Average BTUs', ROUND (MAX (CAST (BTUs AS FLOAT)), 0) AS 'Maximum BTUs' FROM (SELECT email, Appl_order#, 'Appliance Type', BTUs FROM 'Appliance' WHERE email NOT IN (SELECT email FROM 'Utilities')) GROUP BY 'Appliance Type';
```

## Household averages by radius

### Abstract Code

- User enters valid *postal code* (\$postal code number):

```
SELECT 'Zip Code' FROM 'postal code' WHERE 'Zip Code' = '$postal code number';
```

- If postal code number is not found or invalid format, error message is shown.
  - Else, select *radius* (\$radius).
- User clicked on **submit** button- Run **reports by searching postal code and radius** task:

```
WITH(
SELECT "zip_code", $radius, COUNT("zip_code") as hh_count, "home_type",
"square_footage", "heating_thermostat_setting", "cooling_thermostat_setting",
"utilities", "generation_type", "average_kwh", "battery_storage", "utility_list",
COUNT(DISTINCT("household.email")) - COUNT(DISTINCT Utility.email) AS otg_hh,
COUNT(DISTINCT power_generator.email) AS pg_hh
FROM (
(household JOIN power_generator ON household.email = power_generator.email )
JOIN(
SELECT "zip_code" FROM 'postal code', (
SELECT 3958.75*C AS distance FROM (
SELECT 2*ATAN2(SQRT(A), SQRT(1-A)) AS C
FROM(
SELECT (SIN(LAT/2)*SIN(LAT/2)) + (COS(Latitude)* COS(Latitude)) *
(SIN(LON/2)*SIN(LON/2)) AS A
FROM(
SELECT Latitude AS lat2, Longitude AS lon2, lat1, lon1
FROM 'postal code', (
SELECT Latitude AS lat1, Longitude AS lon1 FROM 'postal
code' WHERE 'zip_code' = ' $postal code number' )
)
)
)
) WHERE distance ≤ $radius ) )
ON zip_code = household.zip_code)
JOIN (SELECT email, string_agg('utility_type', ' , ') FROM 'Utility') AS utility_list
ON household.email = Utility.email
)
```

## Phase 2 Abstract Code w/SQL | CS 6400 - Spring 2023 | Team 098

```
) AS base_table,  
SELECT "zip code", $radius, hh_count, IFNULL(COUNT("zip_code"), 0),  
ROUND(AVG(CAST( "square_footage" AS FLOAT)), 0),  
ROUND(AVG(CAST( "heating_thermostat_setting" AS FLOAT)),1),  
ROUND(AVG( CAST("cooling_thermostat_setting" AS FLOAT)),1), "utility_list",  
ROUND(AVG(CAST("average_kwh" AS FLOAT)), 0),  
MAX(COUNT("generation_type")), "otg_hh" FROM base_table  
GROUP BY "home_type", SELECT count(1) FROM 'base_table';
```