Table of Contents:

Task Decomposition with Abstract Code

Main Menu

Enter Household Info

Add Appliance

View/Add/Delete Appliances

Add/Skip Power Generation

View/Add/Delete Power Generation

Wrapping Up

View Report/Query Data

View Top 25 popular manufactures

Manufacturer/model search

Heating/cooling method detail

Water heater statistics by state

Off-the-grid household dashboard

Household averages by radius

Appendix

Main Menu

Abstract Code

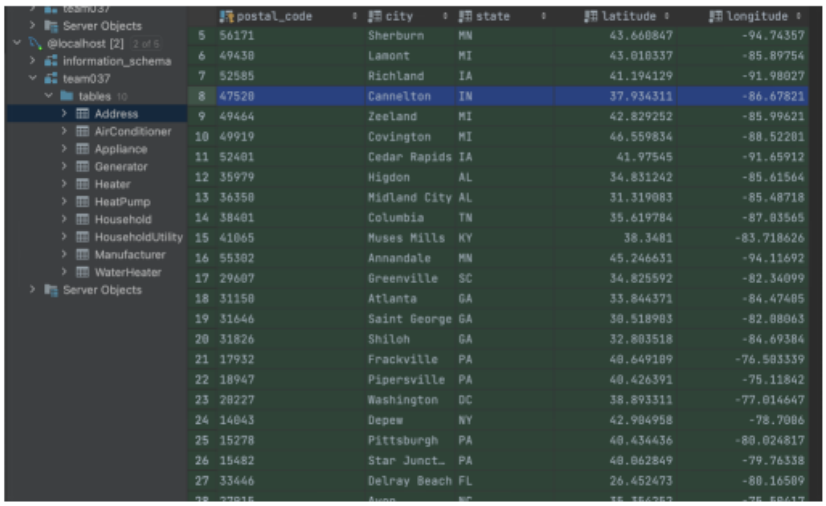
- Show "***Enter My House Info***" and "***View reports/query data***" link.
- Two options:
  - Click on "***Enter My House Info"*** link. Jump to the **Enter Household Info** task.
  - Click on the "***View reports/query data"*** link. Jump to the **View Reports/Query Data** task.

Enter Household Info

Abstract Code

- Each householder enters *email*($Email), *postal code*($PostalCode), *square footage* ($SquareFootage), *thermo heating*($Heating), and *thermo cooling*($Cooling) (in case we uncheck the box *no heating* and *no cooling*) input fields. Each one has to select *home type* ($HouseholdType) and check/uncheck *public utilities* ($Utilities).

#At the very first step, import postal_codes.csv given from class into Address table in the software DataGrid. Part of table results is shown as below,



	postal_code	city	state	latitude	longitude
5	56171	Sherburn	MN	43.660847	-94.74357
6	49438	Lamont	MI	43.018337	-85.89754
7	52585	Richland	IA	41.194129	-91.98027
8	47520	Cannelton	IN	37.934311	-86.67821
9	49464	Zeeland	MI	42.829252	-85.99621
10	49919	Covington	MI	46.559834	-88.52201
11	52401	Cedar Rapids	IA	41.97545	-91.65912
12	35979	Higdon	AL	34.831242	-85.61564
13	36350	Midland City	AL	31.319083	-85.48718
14	38401	Columbia	TN	35.619784	-87.03565
15	41065	Muses Mills	KY	38.3481	-83.718626
16	55302	Annandale	MN	45.246631	-94.11692
17	29607	Greenville	SC	34.825592	-82.34099
18	31150	Atlanta	GA	33.844371	-84.47485
19	31646	Saint George	GA	30.518983	-82.08063
20	31826	Shiloh	GA	32.803518	-84.69384
21	17932	Frackville	PA	40.649189	-76.503339
22	18947	Pipersville	PA	40.426391	-75.11842
23	20227	Washington	DC	38.893311	-77.014647
24	14043	Depew	NY	42.904958	-78.7086
25	15278	Pittsburgh	PA	40.434436	-80.024817
26	15482	Star Junct…	PA	40.062849	-79.76338
27	33446	Delray Beach	FL	26.452473	-80.16589

- After we fill out all required inputs, click on ***Next*** button:

Table of Contents

```
SELECT COUNT (email) AS email_count
FROM Household
WHERE email = $Email;
SELECT COUNT(postal_code) AS postal_count
FROM Address
WHERE postal_code = $PostalCode;
INSERT INTO Household (email, square_footage, household_type, postal_code, heating, cooling) VALUES ($Email, $SquareFootage, $HouseholdType, $PostalCode, $Heating, $Cooling);

#node.js (Javascript) loop to insert different utilities for this household.
FOR EACH Utilities
INSERT INTO HouseholdUtility (email, utilities) VALUES ($Email , $Utilities);
END FOR
```

- If data validation (email_count = 0 and postal_count = 1 from the database and no empty string for thermostat setting meanwhile not indicating no heating and no cooling) is successful for both *email*($Email), *postal code*($PostalCode), then:
  - Add Household Information into the database.
  - Jump to **Add Appliance** form.
- Else:
  - An appropriate error message should be displayed.

Add Appliance

Abstract Code

- If the user has no appliance to add, *appliance type*($ApplianceType) is equal to Null, then click ***Add.***
  - Jump to the **View Appliances** task.
- Else if *appliance type*($ApplianceType) is equal to 'Water Heater', then Execute **WaterHeater** task:
  - Jump to **WaterHeater** form.
  - Query the **Manufacturer** information for **Appliance.**

```
SELECT DISTINCT manufacturer_name FROM Manufacturer;
```

Table of Contents

- The user must select *manufacturer* ($Manufacturer), edit *model name*($ModelName) (optionally), *energy source* ($EnergySourceWH), and enter the information for *capacity* ($Capacity), *temperature setting* ($TemperatureSetting) and *BTU Rating* ($BTURating).
- When ***Add*** button is clicked:
  - If all input fields are valid:
    - That input information will be saved to the database for current **Appliance** for current **Household.**
    - Jump to the **View Appliances** task.

#Note: For appliance order, we will declare a global hashmap in node.js to track the household's email (as a key) along with maximum appliance order as well as current maximum generator order (as an arraylist with two values) for selected appliance/power generator. The initial maximum appliance order and generator order is set as 0 with a given key (household email). We will keep updating this hashmap if we need to add any appliance. The general data structure of this global hashmap can be described as

< $Email, <$CurrentMaxApplianceOrder,$CurrentMaxGeneratorOrder> >

```
# if user choose water heater
# firstly, update the appliance table:
INSERT INTO Appliance
(email,appliance_order,manufacturer_id,model_name,btu_rating)
(SELECT $Email, $ApplianceOrder, m.manufacturer_id , $ModelName, $BTURating
FROM Manufacturer
WHERE manufacturer_name = $SelectedManufactuerName);

#then update water heater table:
INSERT INTO WaterHeater
(email,appliance_order,capacity,temperature_setting,energy_source_wh)
VALUES ($Email, $ApplianceOrder , $Capacity, $TemperatureSetting,
$EnergySourceWH);
```

  - Else:
    - Display appropriate error messages.

- Else if appliance *type*($ApplianceType) is equal to 'Air Handler' then Execute **Air Handler** task:
  - Firstly, jump to **AirHandler** form. In this form, the following public attributes and checking box will be displayed:*manufacturer* ($Manufacturer), *model name*($ModelName) and *BTU Ratings* ($BTURating).three checkbox options for different *air handler types*($AirHandlerType).
    - Query the **Manufacturer** information for **Appliance.** Then select

manufacturer ($Manufacturer) and edit *model name*($ModelName) and enter the information for *BTU Ratings*($BTURating).

```
SELECT DISTINCT manufacturer_name FROM Manufacturer;
```

- Select *air handler type*($AirHandlerType) in the check box:
  - If *the air handler type*($AirHandlerType) check box *Air Condition* is selected, then jump to **AC** task:
    - Display the *Energy Efficiency Ratio* for the text input and then enter the information for *Energy Efficiency Ratio* ($EER).
  - If *the air handler type*($AirHandlerType) check box *Heater* is selected, then jump to the **Heater** task:
    - Display *energy source* ($EnergySourceH) box and then select one type of energy source.
  - If *air handler type*($AirHandlerType) check box *Heater Pump* is selected, then jump to the **HeaterPump** task:
    - Display *SEER* ($SEER) text box as well as *HSPF* ($HSPF), enter the information for both of them.
- When ***Add*** button is clicked:
  - If all input fields are valid:
    - Those input information will be saved to the database for current **Appliance** for current **Household.**

#Note: For appliance order, we will declare a global hashmap in node.js to track the household's email (as a key) along with maximum appliance order as well as current maximum generator order (as an arraylist with two values) for selected appliance/power generator. The initial maximum appliance order and generator order is set as 0 with a given key (household email). We will keep updating this hashmap if we need to add any appliance. The general data structure of this global hashmap can be described as

< $Email, <$CurrentMaxApplianceOrder,$CurrentMaxGeneratorOrder> >

```
#AC/Heater/HeatPump:

# firstly, update the appliance table:
INSERT INTO Appliance
(email,appliance_order,manufacturer_id,model_name,btu_rating)
(SELECT $Email, $ApplianceOrder, m.manufacturer_id , $ModelName, $BTURating
FROM Manufacturer
WHERE manufacturer_name = $SelectedManufactuerName);

# then update AC/Heater/HeatPump
```

Table of Contents

```
#If AC is selected:
INSERT INTO AirConditioner (email,appliance_order,eer)
VALUES ($Email, $ApplianceOrder ,$EER);

#If heater is selected:
INSERT INTO Heater (email,appliance_order,energy_source_h)
VALUES ($Email, $ApplianceOrder , $EnergySourceH);

#If heat pump is selected:
INSERT INTO HeatPump (email,appliance_order,hspf,seer)
VALUES ( $Email,  $ApplianceOrder , $HSPF, $SEER);
```

        - ➢ Jump to the **View Appliances** task.
    - Else:
        - ➢ Display appropriate error messages (such as 'Invalid data type' or 'Invalid range for SEER').

View/Add/Delete Appliances

Abstract Code

- After we finished **Add Appliance** from the last part, firstly run **View Appliances** task:
    - It will query and summarize the basic information (eg. Appliance order#, Type, Manufacturer, Model, etc.) into a table with appropriate layout of ***Delete*** and ***Add Another Appliance*** button.

Table of Contents

```
#View appliance:

SELECT ap.appliance_order,
(CASE WHEN w.appliance_order IS NOT NULL AND w.email IS NOT NULL THEN 'WaterHeater'
WHEN (ac.appliance_order IS NOT NULL AND ac.email IS NOT NULL) OR
(ht.appliance_order IS NOT NULL AND ht.email IS NOT NULL) OR
(hp.appliance_order IS NOT NULL AND hp.email IS NOT NULL) THEN 'AirHandler' END) as Type,
m.manufacturer_name,
ap.model_name
FROM Appliance ap
LEFT JOIN Manufacturer m
ON ap.manufacturer_id = m.manufacturer_id
LEFT JOIN  WaterHeater w
ON  ap.email=w.email AND ap.appliance_order = w.appliance_order
LEFT JOIN  AirConditioner ac
ON  ap.email=ac.email AND ap.appliance_order = ac.appliance_order
LEFT JOIN  Heater ht
ON ap.email=ht.email AND ap.appliance_order = ht.appliance_order
LEFT JOIN  HeatPump hp
ON  ap.email=hp.email AND ap.appliance_order = hp.appliance_order
WHERE ap.email = $Email;
```

- Secondly, it will enable us to edit the current appliances. We will have two cases below:
  - If we click on ***Delete*** button, it will run **Delete Appliances** task:
    - Delete and remove the current selected **Appliance** with its order number and related information for this **Household.**

Table of Contents

```
# if user chooses to delete water heater
 DELETE FROM WaterHeater
 WHERE email = $Email and appliance_order= $ApplianceOrder;

 # if user chooses to delete the AirHandler (AC/Heater/HeatPump) :
 DELETE FROM AirConditioner
 WHERE email = $Email and appliance_order= $ApplianceOrder;

 DELETE FROM Heater
 WHERE email = $Email and appliance_order= $ApplianceOrder;

 DELETE FROM HeatPump
 WHERE email = $Email and appliance_order= $ApplianceOrder;

 DELETE FROM Appliance
 WHERE appliance_order= $ApplianceOrder and email= $Email;
```

- Execute **View Appliances** task.

Table of Contents

```
SELECT ap.appliance_order,
(CASE WHEN w.appliance_order IS NOT NULL AND w.email IS NOT NULL THEN 'WaterHeater'
WHEN (ac.appliance_order IS NOT NULL AND ac.email IS NOT NULL) OR
(ht.appliance_order IS NOT NULL AND ht.email IS NOT NULL) OR
(hp.appliance_order IS NOT NULL AND hp.email IS NOT NULL) THEN 'AirHandler' END) as Type,
m.manufacturer_name,
ap.model_name
FROM Appliance ap
LEFT JOIN Manufacturer m
ON ap.manufacturer_id = m.manufacturer_id
LEFT JOIN  WaterHeater w
ON  ap.email=w.email AND ap.appliance_order = w.appliance_order
LEFT JOIN  AirConditioner ac
ON  ap.email=ac.email AND ap.appliance_order = ac.appliance_order
LEFT JOIN  Heater ht
ON ap.email=ht.email AND ap.appliance_order = ht.appliance_order
LEFT JOIN  HeatPump hp
ON  ap.email=hp.email AND ap.appliance_order = hp.appliance_order
WHERE ap.email = $Email;
```

        - If all appliances are deleted, jump back to the **Add Appliance** form.
    - If we click on ***Add Another Appliance*** button, it will run **Add Appliance** task:
        - Jump to the **Add Appliance** form.
- When ready, the user can click ***Next*** button to jump to **Power Generation Edit** form.

Add/Skip Power Generation

Abstract Code

- After clicking ***Next*** button from Appliance list, jump to **Power Generation Edit** form.
- If the house is “on-the-grid”, then ***Skip*** button is enabled. There are two options for us to choose:

```
# judge whether the household is “off-the- grid” or “on-the-grid”
SELECT COUNT(email) FROM HouseholdUtility
WHERE utilities IS NULL and email = $Email;

# If 1 is returned, the household is “off-the-grid” and the Skip button is disabled;
# If 0 is returned, the household is “on-the-grid” and the Skip button is enabled.
```

Table of Contents

- If we click on ***Skip*** button, run **Skip Power Generation** task:
  - No generator for this **Household** will be saved to the database.
  - Execute **View Power Generation** task.
- Else:
  - Go to step **Add power Generation Info** task to enter power generation Information.

- Else, the house is "off-the-grid", then ***Skip*** button is disabled, and go to step **Add power Generation Info** task to enter power generation Information.

- For **Add power Generation Info** task:
  - Select one *generation type*($GenerationType) from the dropdown box.
  - Enter the information for *Monthly kwh*($AverageMonthlyKiliwattHours) and *Storage kwh*($BatteryStorageCapacity)
  - Finally, after making sure everything is correct. Click on ***Add*** button:
    - If all input fields are valid:
      - The information will be passed and saved in the database.

#Note: For appliance order, we will declare a global hashmap in node.js to track the household's email (as a key) along with maximum appliance order as well as current maximum generator order (as an arraylist with two values) for selected appliance/power generator. The initial maximum appliance order and generator order is set as 0 with a given key (household email). We will keep updating this hashmap if we need to add any power generator. The general data structure of this global hashmap can be described as

< $Email, <$CurrentMaxApplianceOrder,$CurrentMaxGeneratorOrder> >

INSERT INTO Generator (email, generator_order, generation_type, battery_storage_capacity, average_monthly_kilowatt_hours)
VALUES($Email, $GeneratorOrder, $GenerationType, $BatteryStorageCapacity, $AverageMonthlyKilowattHours);

      - Jump to **View Power Generation** task.
    - Else:
      - Display appropriate error messages.

View/Add/Delete Power Generation

Abstract Code

- After finishing **Power Generation Edit** form, **View Power Generation** task will be firstly executed:
  - It will query and summarize the basic information (e.g. Generator order#, Type, Monthly kWh, Battery kWh, etc.) into a table with appropriate layout of ***Delete***

Table of Contents

and ***Add More Generator*** button.

```
#View power generator:

SELECT generator_order AS Num, generation_type AS Type,battery_storage_capacity AS BatterykWh,average_monthly_kilowatt_hours AS MonthlykWh
FROM Generator
WHERE  email=$Email AND  generator_order= $GeneratorOrder;
```

- Secondly, it will enable us to edit the current generators. We will have two scenarios below:
  - If we click on ***Delete*** button, it will run **Delete Power Generations** task:
    - Delete and remove the current selected **Generator** with its order number and related information for this **Household.**

```
DELETE FROM Generator
WHERE email= $Email and generator_order= $GeneratorOrder;
```

    - Execute **View Power Generations** task.

```
SELECT generator_order AS Num, generation_type AS Type,battery_storage_capacity AS BatterykWh,average_monthly_kilowatt_hours AS MonthlykWh
FROM Generator
WHERE  email=$Email AND  generator_order= $GeneratorOrder;
```

    - If all power generators are deleted, jump to the **Power Generation Edit** form.
  - If we click on ***Add More Generator*** button, it will run **Add Power Generation** task:
    - Jump to the **Power Generation Edit** form.
- When ready, the user can click ***Finish*** button to jump to **Wrapping up** form.

Wrapping Up

Abstract Code

- Show "***Return to The Main Menu***" link.
- One option:
  - Click on "***Return to The Main Menu"*** link. Jump to the **Main Menu** form.

Table of Contents

View Report/Query Data

Abstract Code

- After clicking "***View reports/query data"*** link from **Main Menu** form, it will jump to and then run the **View Reports/Query Data** task:
  - Display Report Links List with "***List Top 25 Popular Manufacturers"*** link, "***Search Manufacturers/Model"*** link, "***Report Heating/Cooling Method"*** link, "***Report Water Heater Statistics by State"*** link, "***Create off-the-grid Household Dashboard"*** link and "***Report Household averages by Radius"*** link***.***
  - Then we have many options to choose:
    - Click the "***List Top 25 Popular Manufacturers"*** link. Jump to the **List Top 25 Popular Manufacturers** task.
    - Click the "***Search Manufacturers/Model"*** link. Jump to the **Search Manufacturers/Model** task.
    - Click the "***Report Heating/Cooling Method"*** link. Jump to the **Report Heating/Cooling Method** task.
    - Click the "***Report Water Heater Statistics by State"*** link. Jump to the **Report Water Heater Statistics by State** task.
    - Click the "***Create off-the-grid Household Dashboard"*** link. Jump to the **Create off-the-grid Household Dashboard** task.
    - Click the "***Report Household averages by Radius"*** link. Jump to the **Report Household averages by Radius** task.

View Top 25 popular manufacturers

Abstract code

- Run the view **List Top 25 popular manufacturers** task: Query for information about the household information and appliances listing from the HTTP Session/Cookie.
- If there is no information returned:
  - Display a message with 'Nothing is found'.
- Else:
  - Display one table listing the top twenty-five manufacturers with the most appliances in the database,
    - One column for each manufacturer along with a ***manufacturer drilldown*** button; another column for the raw count of appliances for that manufacturer; Table ordered by count descending.

```
SELECT m. manufacturer_name, COUNT(appliance_order) AS ApplianceCount
FROM Manufacturer AS m, Appliance AS ap
WHERE m. manufacturer_id = ap. manufacturer_id
GROUP BY m. manufacturer_id
ORDER BY ApplianceCount DESC
```

Table of Contents

```
LIMIT 25;
```

- If clicking on ***manufacturer drilldown*** button:
  - List the manufacturer name at the top, then a table is displayed:
    - One column for each appliance type; another column with the raw count of appliances of that type for that manufacturer (integer); Ordered by count ascending.

Table of Contents

```
WITH tmp1 AS(
SELECT ap.manufacturer_id,
        m.manufacturer_name,
        (CASE WHEN ac.email IS NOT NULL THEN 'Air_Conditioner'
        WHEN w.email IS NOT NULL THEN 'WaterHeater'
        WHEN ht.email IS NOT NULL THEN 'Heater'
        WHEN hp.email IS NOT NULL THEN 'HeatPump' end )AS appliance_type
FROM Appliance AS ap
INNER JOIN Manufacturer m ON ap.manufacturer_id = m.manufacturer_id
LEFT JOIN AirConditioner AS ac ON ac.email = ap.email AND ac.appliance_order =
ap.appliance_order
LEFT JOIN WaterHeater AS w ON w.email = ap.email AND w.appliance_order =
ap.appliance_order
LEFT JOIN heater AS ht ON ht.email = ap.email AND ht.appliance_order =
ap.appliance_order
LEFT JOIN HeatPump AS hp ON hp.email = ap.email AND hp.appliance_order =
ap.appliance_order
WHERE m.manufacturer_name = '$input_manufacturer_name' ),

tmp2 AS(
SELECT  manufacturer_id,manufacturer_name,
(CASE WHEN appliance_type IN ('Air_Conditioner','Heater','HeatPump') THEN 'AirHandler'
        ELSE appliance_type END )AS appliance_type
FROM tmp1
WHERE appliance_type IN ('Air_Conditioner','Heater','HeatPump'))

SELECT appliance_type, count(manufacturer_id) AS Appliance_Count
FROM tmp1
GROUP BY appliance_type
UNION
SELECT appliance_type, count(manufacturer_id) AS Appliance_Count
FROM tmp2
GROUP BY appliance_type;
```

Table of Contents

Manufacturer/model search

Abstract code

- Run **Search Manufacturers/Model** task:
  - Display a *search row* ($ManuModel) with ***Search*** button next to it. Enter the information for the search row and click on ***Search*** button:
  - If the information is not found:
    - Display a message with 'Nothing is found'.
  - Else:
    - Return a list of distinct results where entered string matches any part of a manufacturer name or model name.
- The table content is shown below:
  - Columns for the manufacturer name and model name; ordered by manufacturer name ascending and model name ascending.
  - The cell with the manufacturer name and the model name (or both) that matched the search string must be highlighted with a light green background.

```
SELECT DISTINCT m.manufacturer_name, ap.model_name
FROM Appliance AS ap INNER JOIN Manufacturer AS m ON
( (ap.manufacturer_id = m.manufacturer_id AND m.manufacturer_name LIKE
'%input_string%') OR (ap.manufacturer_id = m.manufacturer_id AND ap.model_name LIKE
'%input_string%') )
ORDER BY m.manufacturer_name ASC,ap.model_name ASC;
```

Heating/cooling method detail

Abstract code

- Run **Report Heating/Cooling Method** task:
  - Display one table grouped by household types, the table content is described below:
    - The following statistics will be displayed, with a column for each household type: the count of air conditioners, average air conditioner BTUs, average EER, the count of heaters, average heater BTUs, the most common energy source, the count of heater pumps, average heat pump BTUs, average SEER and the average HSPF.

Table of Contents

```
WITH ac_output AS (
SELECT h.household_type, COUNT(h.household_type) AS ac_count,
ROUND(AVG(btu_rating),0) AS avg_btu_rating, ROUND(AVG(eer),1) AS avg_eer
FROM Household AS h INNER JOIN (
   SELECT ac.email, ap.btu_rating, ac.eer
   FROM AirConditioner AS ac INNER JOIN Appliance AS ap
   ON ac.email = ap.email AND ac.appliance_order = ap.appliance_order
) AS ac2
ON h.email = ac2.email
GROUP BY h.household_type
),

heater_output1 AS (
SELECT h.household_type, ht2.btu_rating, ht2.energy_source_h
FROM Household AS h INNER JOIN (
   SELECT ht.email, ap.btu_rating, ht.energy_source_h
   FROM Heater AS ht INNER JOIN Appliance AS ap
   ON ht.email = ap.email AND ht.appliance_order = ap.appliance_order
) AS ht2
ON h.email = ht2.email
),

heater_output2 AS (
   SELECT DISTINCT hto1.household_type, (
      SELECT energy_source_h
      FROM heater_output1 AS hto2
      WHERE hto2.household_type = hto1.household_type
      GROUP BY hto2.energy_source_h
      ORDER BY COUNT(hto2.energy_source_h) DESC
      LIMIT 1
      ) AS energy_source_h
   FROM heater_output1 AS hto1
),

heater_output AS (
SELECT hto1.household_type, COUNT(hto1.household_type) AS
heater_count,ROUND(AVG(btu_rating),0) AS avg_btu_rating, hto2.energy_source_h AS
most_common_energy_source
FROM heater_output1 AS hto1 INNER JOIN heater_output2 AS hto2
ON hto1.household_type = hto2.household_type
GROUP BY hto1.household_type
),
```

Table of Contents

```
heatpump_output AS (
SELECT h.household_type, COUNT(h.household_type) AS
heatpump_count,ROUND(AVG(btu_rating),0) AS avg_btu_rating,ROUND(AVG(hspf),1)
AS avg_hspf, ROUND(AVG(seer),1) AS avg_seer
FROM Household AS h INNER JOIN (
   SELECT hp.email, ap.btu_rating, hp.hspf, hp.seer
   FROM HeatPump AS hp INNER JOIN Appliance AS ap
   ON hp.email = ap.email AND hp.appliance_order = ap.appliance_order
) AS hp2
ON h.email = hp2.email
GROUP BY h.household_type
)

SELECT
      h.household_type,coalesce(aco.ac_count,'') AS
      ac_count,coalesce(aco.avg_btu_rating,'') AS
      ac_avg_btu_rating,coalesce(aco.avg_eer,'') AS
      ac_avg_eer,coalesce(ho.heater_count,'') AS
      heater_count,coalesce(ho.avg_btu_rating,'') AS
      heater_avg_btu_rating,coalesce(ho.most_common_energy_source, '') AS
      most_common_energy_source, coalesce(hpo.heatpump_count, '') AS
      heatpump_count,coalesce(hpo.avg_btu_rating, '') AS
      heatpump_avg_btu_rating,coalesce(hpo.avg_seer, '') AS
      avg_seer,coalesce(hpo.avg_hspf, '') AS avg_hspf
FROM (SELECT DISTINCT household_type FROM Household) AS h
   LEFT JOIN ac_output AS aco ON aco.household_type = h.household_type
   LEFT JOIN heater_output AS ho ON ho.household_type = h.household_type
   LEFT JOIN heatpump_output AS hpo ON hpo.household_type = h.household_type
```

Table of Contents

Water heater statistics by state

Abstract code

- Run **Report Water Heater Statistics by State** task:
  - First, display a table with water heater statistics for each state. The table content is shown below:

```
SELECT ad.state, COALESCE(ROUND(AVG(w.capacity)),0) AS avg_wh_capacity,
    COALESCE(ROUND(AVG(ap.btu_rating)),0) AS avg_wh_btus,
    COALESCE(ROUND(AVG(w.temperature_setting)),0) AS avg_wh_temp,
    COALESCE(COUNT(IF(w.temperature_setting IS NOT NULL, 1, NULL)), 0) AS count_with_temp_setting,
    COALESCE(COUNT(IF(w.temperature_setting IS NULL, 1, NULL)), 0) AS count_without_temp_setting
FROM Address ad
LEFT JOIN Household h ON ad.postal_code = h.postal_code
LEFT JOIN Appliance ap ON h.email = ap.email
LEFT JOIN WaterHeater w ON ap.email = w.email AND ap.appliance_order = w.appliance_order
GROUP BY ad.state
ORDER BY ad.state ASC;
```

    - Each column for the state's abbreviation along with ***state abbreviation*** button, the average water heater capacity, the average water heater BTUs, the average water heater temperature setting, the count of water heaters where a temperature setting has been provided, the count of water heaters where no temperature setting has been provided, sorted by state abbreviation ascending.
    - If there are no water heaters and/or households for a state, display zeroes for all statistical columns.
  - After clicking each ***state abbreviation*** button, jump to the state water heater reports table:

Table of Contents

```
SELECT w.energy_source_wh AS energy_source,
    COALESCE(ROUND(MIN( w.capacity)),'') AS min_state_wh_capacity,
    COALESCE(ROUND(AVG(w.capacity)),'') AS avg_state_wh_capacity,
    COALESCE(ROUND(MAX( w.capacity)),'') AS max_state_wh_capacity,
    COALESCE(ROUND(MIN( w.temperature_setting),1),'') AS min_state_wh_temp,
    COALESCE(ROUND(AVG(w.temperature_setting),1),'') AS avg_state_wh_temp,
    COALESCE(ROUND(MAX( w.temperature_setting),1),'') AS max_statewh_temp
FROM Address ad
LEFT JOIN Household h ON ad. postal_code = h. postal_code
LEFT JOIN Appliance ap ON h.email = ap.email
LEFT JOIN WaterHeater w ON ap.email = w.email AND ap. appliance_order = w. appliance_order
WHERE ad.state = '$SelectState'
GROUP BY w.energy_source_wh
ORDER BY w.energy_source_wh ASC;
```

- Display the state as a header or report title.
- For the table content, display a column for each energy source, grouped by energy source; Display a column of the minimum water heater capacity; Display a column of the average water heater capacity; Display a column of the maximum water heater capacity; Display a column of the minimum temperature setting; Display a column of the average temperature setting; Display a column of the max temperature setting; Display the row ordered by energy source in ascending order.

Off-the-grid household dashboard

Abstract code

- Run **Create off-the-grid Household Dashboard** task, display the following tables:
  - First table: one column listed the state with the most “off-the-grid” households, one column displayed the count of its households.

Table of Contents

```
SELECT ad.state, COUNT(h.email) AS household_count
FROM Household h
LEFT JOIN Address ad ON h.postal_code = ad.postal_code
WHERE email IN (SELECT email FROM HouseholdUtility WHERE utilities IS NULL)
GROUP BY ad.state
HAVING COUNT(h.email) IN (
 SELECT MAX(count)
 FROM (
        SELECT COUNT(h.email) AS count
        FROM Household h
        LEFT JOIN Address ad ON h.postal_code = ad.postal_code
        WHERE email IN (SELECT email FROM HouseholdUtility WHERE utilities IS NULL)
        GROUP BY ad.state
 ) max
);
```

- Display the average battery storage capacity.

```
SELECT COALESCE(ROUND(AVG(battery_storage_capacity)),'') AS
avg_battery_storage_capacity

FROM   Generator g

WHERE   g.email IN (SELECT email FROM HouseholdUtility WHERE utilities IS NULL);
```

- Second table: two columns for the percentages for each power generation type (solar-electric, wind, or mixed).

Table of Contents

```
WITH GeneratorUser AS (
SELECT DISTINCT email, generation_type
FROM Generator
)
SELECT 'solar_electric' AS power_generation_type,
   CONCAT(COALESCE(ROUND(100 * SUM(CASE WHEN generation_type = 'solar_electric'
   AND email NOT IN (SELECT email FROM GeneratorUser WHERE generation_type = 'wind')
THEN 1 ELSE 0 END) / COUNT(DISTINCT email), 1),0), '%') AS percentage
FROM   GeneratorUser
WHERE   email IN (SELECT email FROM HouseholdUtility WHERE utilities IS NULL)
UNION
SELECT 'wind' AS power_generation_type,
   CONCAT(COALESCE(ROUND(100 * SUM(CASE WHEN generation_type = 'wind'
   AND email NOT IN (SELECT email FROM GeneratorUser WHERE generation_type =
'solar_electric') THEN 1 ELSE 0 END) / COUNT( DISTINCT email), 1),0), '%') AS percentage
FROM   GeneratorUser
WHERE   email IN (SELECT email FROM HouseholdUtility WHERE utilities IS NULL)
UNION
SELECT 'wind+solar' AS power_generation_type,
   CONCAT(COALESCE(ROUND(100 *(SUM(CASE WHEN generation_type = 'wind'
  AND email IN (SELECT email FROM GeneratorUser WHERE generation_type =
'solar_electric') THEN 1 ELSE 0 END) / COUNT( DISTINCT email)) , 1),0), '%') AS percentage
FROM   GeneratorUser
WHERE   email IN (SELECT email FROM HouseholdUtility WHERE utilities IS NULL);
```

- Third table: one column for the “off-the-grid”/ “on-the-grid” household category, one column for the average water heater gallon capacity.

Table of Contents

```
SELECT 'off_the_grid' AS household_category,

COALESCE(ROUND(AVG(capacity),1),'') AS avg_wh_capacity

FROM WaterHeater w

WHERE email IN (SELECT email FROM HouseholdUtility WHERE utilities IS NULL)

UNION

SELECT 'on_the_grid' AS household_category,

COALESCE(ROUND(AVG(capacity),1),'') AS avg_wh_capacity

FROM WaterHeater w

WHERE email IN (SELECT email FROM HouseholdUtility WHERE utilities IS NOT NULL);
```

- Fourth table: columns for the minimum, average, and maximum BTUs for all “off-the-grid” households’ appliances will be displayed, grouped by appliance type.

Table of Contents

```
SELECT 'air_handler' AS appliance_type, COALESCE(MIN(btu_rating),'') AS
min_btus,COALESCE(ROUND(AVG(btu_rating)),'') AS avg_btus,COALESCE(MAX(btu_rating),'')
AS max_btus

FROM Appliance

WHERE email IN (SELECT email FROM HouseholdUtility WHERE utilities IS NULL)

AND email IN (SELECT email FROM AirConditioner UNION SELECT email FROM Heater UNION
SELECT email FROM HeatPump)

AND appliance_order IN (

SELECT appliance_order FROM AirConditioner WHERE email IN (SELECT email FROM
HouseholdUtility WHERE utilities IS NULL)

UNION

SELECT appliance_order FROM Heater WHERE email IN (SELECT email FROM
HouseholdUtility WHERE utilities IS NULL)

UNION

SELECT appliance_order FROM HeatPump WHERE email IN (SELECT email FROM
HouseholdUtility WHERE utilities IS NULL))

UNION

SELECT 'air_conditioner' AS appliance_type, COALESCE(MIN(btu_rating),'') AS
min_btus,COALESCE(ROUND(AVG(btu_rating)),'') AS avg_btus,COALESCE(MAX(btu_rating),'')
AS max_btus

FROM AirConditioner ac

LEFT JOIN Appliance ap ON ac.email = ap.email AND ac.appliance_order =
ap.appliance_order

WHERE ac.email IN (SELECT email FROM HouseholdUtility WHERE utilities IS NULL)

UNION

SELECT 'heater' AS appliance_type, COALESCE(MIN(btu_rating),'') AS
```

Table of Contents

```
min_btus,COALESCE(ROUND(AVG(btu_rating)),'') AS avg_btus,COALESCE(MAX(btu_rating),'')
AS max_btus

FROM Heater ht

LEFT JOIN Appliance ap ON ht.email = ap.email AND ht.appliance_order =
ap.appliance_order

WHERE ht.email IN (SELECT email FROM HouseholdUtility WHERE utilities IS NULL)

UNION

SELECT 'heat_pump' AS appliance_type, COALESCE(MIN(btu_rating),'') AS
min_btus,COALESCE(ROUND(AVG(btu_rating)),'') AS avg_btus,COALESCE(MAX(btu_rating),'')
AS max_btus

FROM HeatPump hp

LEFT JOIN Appliance ap ON hp.email = ap.email AND hp.appliance_order =
ap.appliance_order

WHERE hp.email IN (SELECT email FROM HouseholdUtility WHERE utilities IS NULL)

UNION

SELECT 'water_heater' AS appliance_type, COALESCE(MIN(btu_rating),'') AS
min_btus,COALESCE(ROUND(AVG(btu_rating)),'') AS avg_btus,COALESCE(MAX(btu_rating),'')
AS max_btus

FROM WaterHeater w

LEFT JOIN Appliance ap ON w.email = ap.email and w.appliance_order = ap.appliance_order

WHERE w.email IN (SELECT email FROM HouseholdUtility WHERE utilities IS NULL);
```

Table of Contents

Household averages by radius

Abstract code

- Run **Report Household averages by Radius** task:
  - Display two input fields: postal code to center the search on, and the search radius with ***Search*** button on the side.
  - Enter two input fields.
  - After clicking on the ***Search*** button:

```
SELECT COUNT(postal_code) FROM Address WHERE postal_code='$PostalCode';
```

If 0 is returned, the '$PostalCode' input by user is not valid and an error message will be displayed. If 1 is returned, the '$PostalCode' input by user is valid.
If the search radius falls in the following category (0, 5, 10, 25, 50, 100, and 250), we consider the '$Radius' input by user is valid. Otherwise, the '$Radius' input by user is not valid and an error message will be displayed.
If both input fields are valid, users can go ahead and generate the report.

Table of Contents

```
WITH Input AS (
SELECT latitude AS lat, longitude AS lon
FROM Address
WHERE postal_code='$PostalCode'),

AddressInput AS (
SELECT ad.postal_code, ad.latitude AS lat1, ad.longitude AS lon1, i.lat AS lat2, i.lon AS lon2
FROM   Address ad, Input i  ),

Distance AS (
SELECT ad.postal_code, ad.lat1, ad.lon1, ad.lat2, ad.lon2,
 2 * 3958.75  *ATAN2(SQRT(
    POWER(SIN((RADIANS(lat2) - RADIANS(lat1)) / 2) ,2)+
            COS(RADIANS(lat1)) *
            COS(RADIANS(lat2)) *
    POWER(SIN((RADIANS(lon2) - RADIANS(lon1)) / 2), 2)),
            SQRT(1 - (POWER(SIN((RADIANS(lat2) - RADIANS(lat1)) / 2) ,2)+
            COS(RADIANS(lat1)) *
            COS(RADIANS(lat2)) *
    POWER(SIN((RADIANS(lon2) - RADIANS(lon1)) / 2), 2)) )
) AS distance

FROM    AddressInput ad ),

PostalCode AS (SELECT d.postal_code
 FROM Distance d
 WHERE d.distance <= '$Radius'),

AllHousehold AS (
SELECT h.email, h.postal_code, h.square_footage, h.household_type, h.heating, h.cooling
FROM PostalCode pc
```

Table of Contents

```
LEFT JOIN Household h ON pc.postal_code = h.postal_code),
AllHouseholdInfo AS (
SELECT COUNT(DISTINCT a.email) AS households_count,
 SUM(IF(a.household_type ='House', 1, NULL)) AS house_count,
 SUM(IF(a.household_type ='Apartment', 1, NULL)) AS apartment_count,
 SUM(IF(a.household_type ='Townhome', 1, NULL)) AS townhome_count,
 SUM(IF(a.household_type ='Condominium', 1, NULL)) AS condominium_count,
 SUM(IF(a.household_type ='Mobile Home', 1, NULL)) AS mobile_home_count,
 ROUND(AVG(a.square_footage),1) AS avg_square_footage,
 ROUND(AVG(a.heating),1) AS avg_heating_temp,
 ROUND(AVG(a.cooling),1) AS avg_cooling_temp
FROM AllHousehold a),

UtilityInfo AS (
SELECT  a.email, GROUP_CONCAT(DISTINCT hu.utilities SEPARATOR ',') AS public_utilities_used,
        SUM(IF(hu.utilities IS NULL, 1, NULL)) AS off_the_grid_count
FROM AllHousehold a
LEFT JOIN HouseholdUtility hu ON a.email = hu.email
GROUP BY a.email),

HouseholdWithGeneration AS (
SELECT a.email, g.generation_type, g.battery_storage_capacity, g.average_monthly_kilowatt_hours
FROM AllHousehold a
LEFT JOIN Generator g ON a.email = g.email),

GenerationInfo1 AS (
SELECT COUNT(DISTINCT h.email) AS homes_with_power_generation_count,
  ROUND(AVG(h.average_monthly_kilowatt_hours)) AS average_monthly_kilowatt_hours
FROM HouseholdWithGeneration h),
```

Table of Contents

```
GenerationInfo2 AS (
SELECT COUNT(DISTINCT h.email) AS homes_with_battery_storage_count
FROM HouseholdWithGeneration h
WHERE h.battery_storage_capacity IS NOT NULL),

GenerationInfo3 AS (
SELECT h.generation_type AS most_common_generation_method, COUNT(h.email) as count
FROM HouseholdWithGeneration h
GROUP BY h.generation_type
ORDER BY count DESC
 LIMIT 1)
 SELECT COALESCE(a.households_count,0) AS households_count, COALESCE(a.house_count,0) AS
house_count,

    COALESCE(a.apartment_count,0) AS apartment_count, COALESCE(a.townhome_count,0) AS
townhome_count,

    COALESCE(a.condominium_count,0) AS condominium_count,  COALESCE(a.mobile_home_count,0)
AS mobile_home_count,

    COALESCE(a.avg_square_footage, 0) AS avg_square_footage,  COALESCE(a.avg_heating_temp, 0) AS
avg_heating_temp,

    COALESCE(a.avg_cooling_temp, 0) AS avg_cooling_temp,
    COALESCE(u.public_utilities_used, '') AS public_utilities_used,
    COALESCE(u.off_the_grid_count, 0) AS off_the_grid_count,
    COALESCE(g1.homes_with_power_generation_count, 0) AS homes_with_power_generation_count,
    COALESCE(g1.average_monthly_kilowatt_hours, 0) AS average_monthly_kilowatt_hours,
    COALESCE(g2.homes_with_battery_storage_count, 0) AS homes_with_battery_storage_count,
    COALESCE(g3.most_common_generation_method, '') AS most_common_generation_method
  FROM AllHouseholdInfo a, UtilityInfo u, GenerationInfo1 g1,GenerationInfo2 g2, GenerationInfo3 g3
```

Table of Contents

- If the input postal code and radius are valid:
  - Display the postal code, the search radius, and the count of households.
  - Display a table: a column for the household types, and another column for the count of households for each household type.
  - Display average square footage, average heating temperature, average cooling temperature, which public utilities are used (displayed in a single cell, separated by commas), the count of "off-the-grid" homes, the count of homes with power generation, the most common generation method, average monthly power generation and the count of the households with battery storage.
- Else:
  - Display appropriate error messages.

Table of Contents

Appendix

Markups	Examples
Bold Underline: **Form**	**Water Heater** form
Bold Italics: Buttons	***Enter My House Info*** button
Bold*: Task*	**Enter Household Info** task
Italics: Form input fields (checkbox or free-form text)	*Energy Source, Energy Efficiency Ratio, etc.*
"Bold Italics with quotation marks": Form link	"***View reports/query data***" link
'Single Quote': Message	'Nothing is found'
Bold highlighted in Blue: Entity in ER	**Household**
Highlighted in Orange: Table in Database	HouseholdUtility
$String Highlighted in Green: input string in UI	$input_manufacturer_name
$Double/Int Highlighted in Red: input number in UI	$ApplianceOrder
Highlighted in Purple: CTE Table name	ac_output

Table of Contents