Project Report

IFT 458 - PD 2

Spring 2018

Ashley Mendoza

Vincent Li

Dr. Kuitche

Feb 5, 2018

Problem Definition

PV Module

A PV module is identified by a unique model number. It also has the following specifications: module length, width, weight, cell area, cell technology, total number of cells, number of cells in series, number of series in parallel, series fuse rating, interconnect material, interconnect supplier, superstrate type, superstrate manufacturer, substrate type, substrate manufacturer, frame material, frame adhesive, encapsulant type, encapsulant manufacturer, junction box type, junction box manufacturer, junction box adhesive, cable type, connector type, and maximum system voltage.

Each module has a number of bypass diodes. Each bypass diode has a model number, a rating, and a max junction temperature.

A PV module is produced by a PV module manufacturer. We keep track of the module manufacturing date, the manufacturing locations (i.e. a module may be manufactured at several locations), and the module manufacturing performance rating. The performance rating has the following components: nominal operating cell temperature (NOCT), short-circuit current (Isc), open-circuit voltage (Voc), current at maximum power (Imp), voltage at maximum power (Vmp), fill factor (FF), and maximum power output (Pmp).

For a manufacturer, we record the manufacturer name, a contact person name, address, office phone and/or cell phone, and email address.

A PV module is tested and certified by an accredited testing laboratory. We keep track of the testing lab name, address, and contact person’s name, phone number, and email address. We also keep track of the tests the PV module was subjected to and the relevant test standard.

For each test, we record the following data: name of the test engineer, test date, nominal operating cell temperature (NOCT), short-circuit current (Isc), open-circuit voltage (Voc), current at maximum power (Imp), voltage at maximum power (Vmp), fill factor (FF), and maximum power output (Pmp), and the pass/fail verdict.

Introduction

The purpose of the following Project Deliverable was to design the backend functionality for the solar PV company website using the given problem definition. This report will outline the process taken by providing a conceptual diagram, an ER diagram, a complete UML class relational diagram, and the SQL code to implement the structure. Next, it will provide a detailed description of how a non-technical user can take the zip file and get it up and running. Lastly, it will indicate what was achieved, learned, the challenges faced and overcome, and ways to improve the structure.

Method

The following outlines the four steps taken to form the conceptual diagram for the back-end design as discussed in lecture.

Step 1: Identify entities and attributes

A PV module is identified by a unique model number. It also has the following specifications:

- module length,
- width,
- weight,
- cell area,
- cell technology,
- total number of cells,
- number of cells in series,
- number of series in parallel,
- series fuse rating,
- interconnect material,
- interconnect supplier,
- superstrate type,
- superstrate manufacturer,
- substrate type,
- substrate manufacturer,
- frame material,
- frame adhesive,
- encapsulant type,
- encapsulant manufacturer,
- junction box type,
- junction box manufacturer,
- junction box adhesive,
- cable type,
- connector type,
- maximum system voltage.

Each module has a number of bypass diodes. Each bypass diode has a

- model number,
- a rating

- max junction temperature.

A PV module is produced by a PV module manufacturer. We keep track of the module

- manufacturing date, the
- manufacturing locations (i.e. a module may be manufactured at several locations), and the
- module manufacturing performance rating.

The performance rating has the following components:

- nominal operating cell temperature (NOCT),
- short-circuit current (ISC),
- open-circuit voltage (Voc),
- current at maximum power (Imp),
- voltage at maximum power (Vmp),
- fill factor (FF), and
- maximum power output (Pmp).

For a manufacturer, we record the

- manufacturer name,
- a contact person name,
- address,
- office phone and/or cell phone,
- and email address.

A PV module is tested and certified by an accredited testing laboratory. We keep track of the

- testing lab name,
- address, and
- contact person's name,
- phone number, and
- email address.

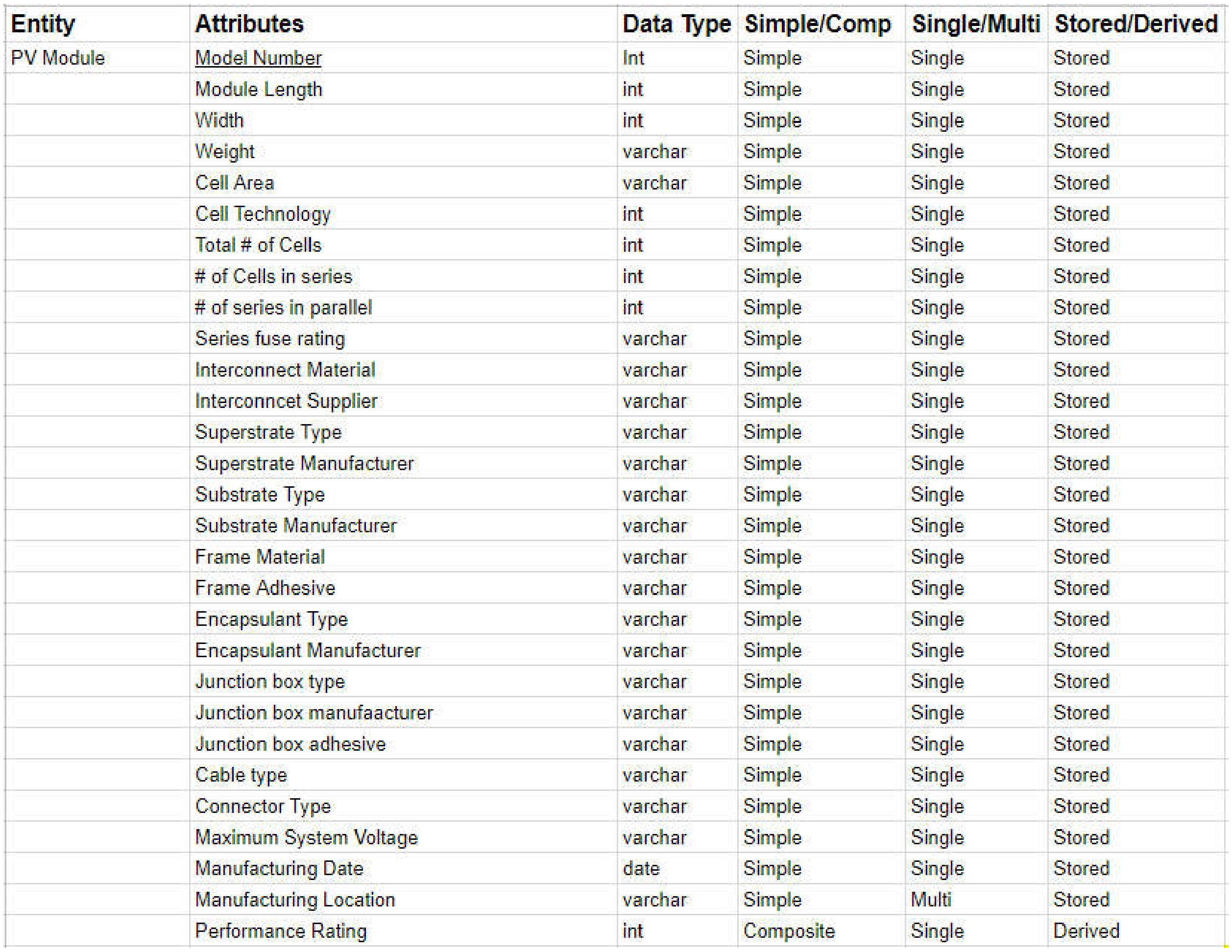
We also keep track of the tests the PV module was subjected to and the relevant test standard.

For each test, we record the following data:

- name of the test engineer,

- test date,
- nominal operating cell temperature (NOCT),
- short-circuit current (Isc),
- open-circuit voltage (Voc),
- current at maximum power (Imp),
- voltage at maximum power (Vmp),
- fill factor (FF),
- and maximum power output (Pmp), and
- the pass/fail verdict.

Step 2: Entities - Attribute Types



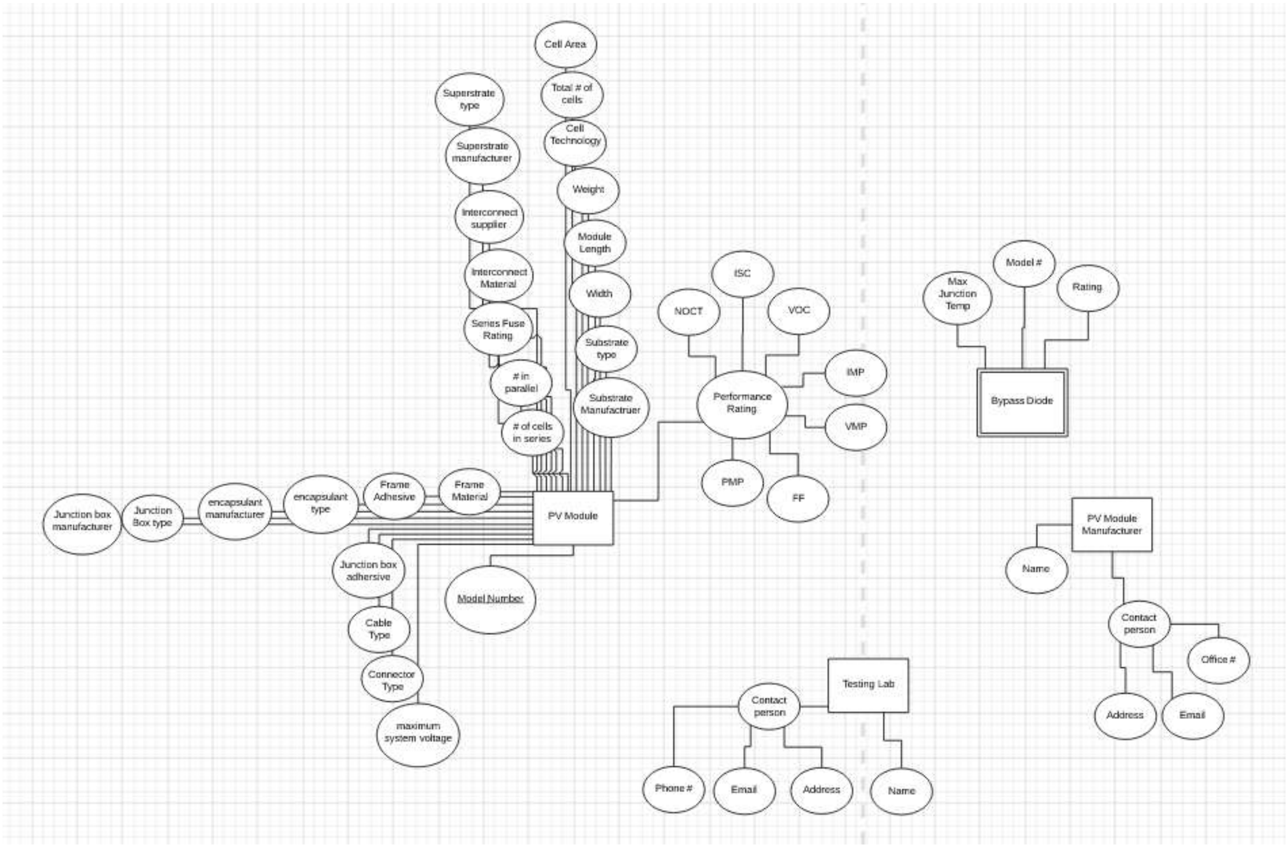
Entity	Attributes	Data Type	Simple/Comp	Single/Multi	Stored/Derived
PV Module	Model Number	Int	Simple	Single	Stored
	Module Length	int	Simple	Single	Stored
	Width	int	Simple	Single	Stored
	Weight	varchar	Simple	Single	Stored
	Cell Area	varchar	Simple	Single	Stored
	Cell Technology	int	Simple	Single	Stored
	Total # of Cells	int	Simple	Single	Stored
	# of Cells in series	int	Simple	Single	Stored
	# of series in parallel	int	Simple	Single	Stored
	Series fuse rating	varchar	Simple	Single	Stored
	Interconnect Material	varchar	Simple	Single	Stored
	Interconncet Supplier	varchar	Simple	Single	Stored
	Superstrate Type	varchar	Simple	Single	Stored
	Superstrate Manufacturer	varchar	Simple	Single	Stored
	Substrate Type	varchar	Simple	Single	Stored
	Substrate Manufacturer	varchar	Simple	Single	Stored
	Frame Material	varchar	Simple	Single	Stored
	Frame Adhesive	varchar	Simple	Single	Stored
	Encapsulant Type	varchar	Simple	Single	Stored
	Encapsulant Manufacturer	varchar	Simple	Single	Stored
	Junction box type	varchar	Simple	Single	Stored
	Junction box manufaacturer	varchar	Simple	Single	Stored
	Junction box adhesive	varchar	Simple	Single	Stored
	Cable type	varchar	Simple	Single	Stored
	Connector Type	varchar	Simple	Single	Stored
	Maximum System Voltage	varchar	Simple	Single	Stored
	Manufacturing Date	date	Simple	Single	Stored
	Manufacturing Location	varchar	Simple	Multi	Stored
	Performance Rating	int	Composite	Single	Derived

Entity	Attributes	Data Type	Simple/Comp	Single/Multi	Stored/Derived
Bypass Diode	Model Number	Varchar	Simple	Single	Stored
	Maximum Junction Temp	varchar	Simple	Single	Stored
	Rating	int	Simple	Single	Stored

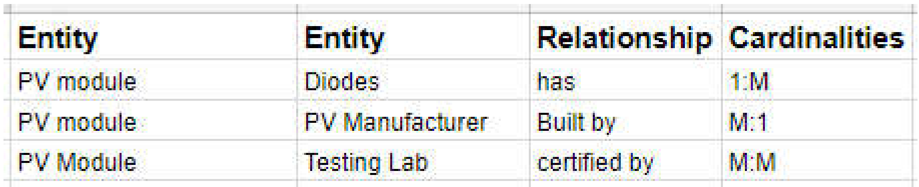
Entity	Attributes	Data Type	Simple/Comp	Single/Multi	Stored/Derived
PV Manufacturer	Name	varchar	Simple	Single	Stored
	Contact Person	varchar	Composite	Single	Stored

Entity	Attributes	Data Type	Simple/Comp	Single/Multi	Stored/Derived
Testing Lab	Lab Name	varchar	Simple	Single	Stored
	Lab Address	varchar	Simple	Single	Stored
	Contact Person	varchar	Simple	Single	Stored

Step 3: Preliminary Design

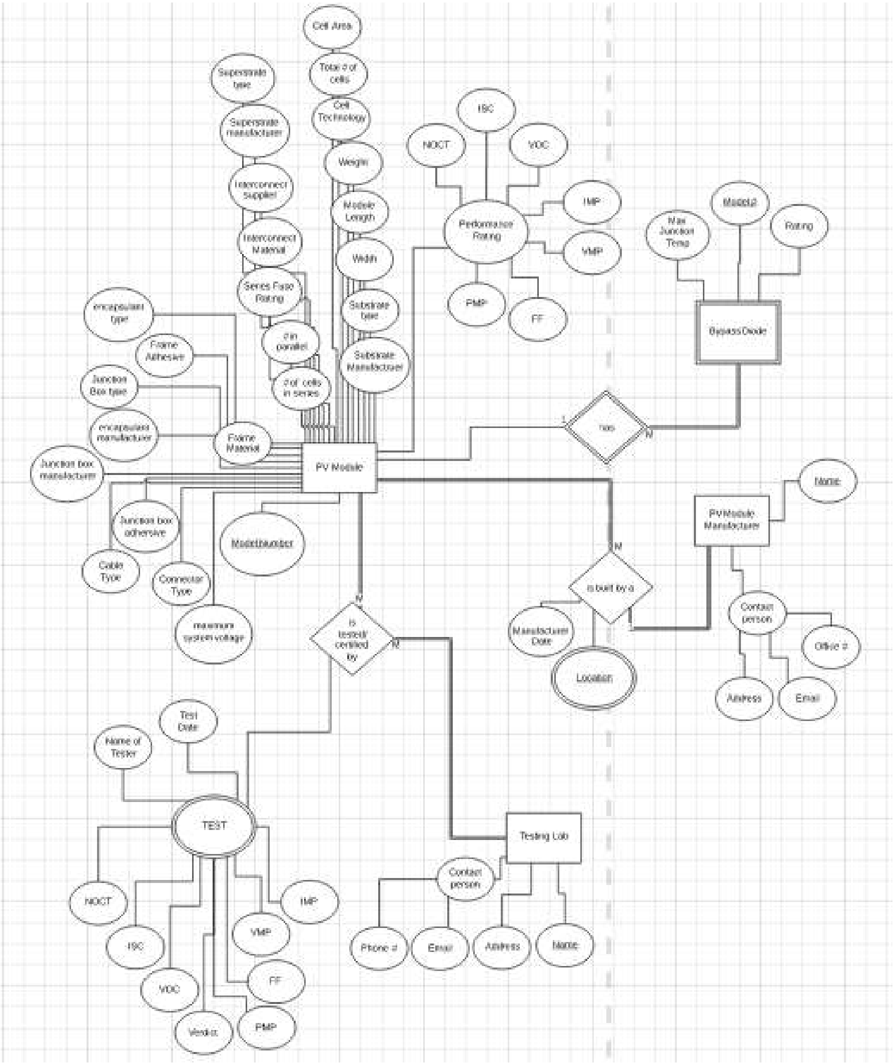


Step 4: Entities Relationships and Cardinalities

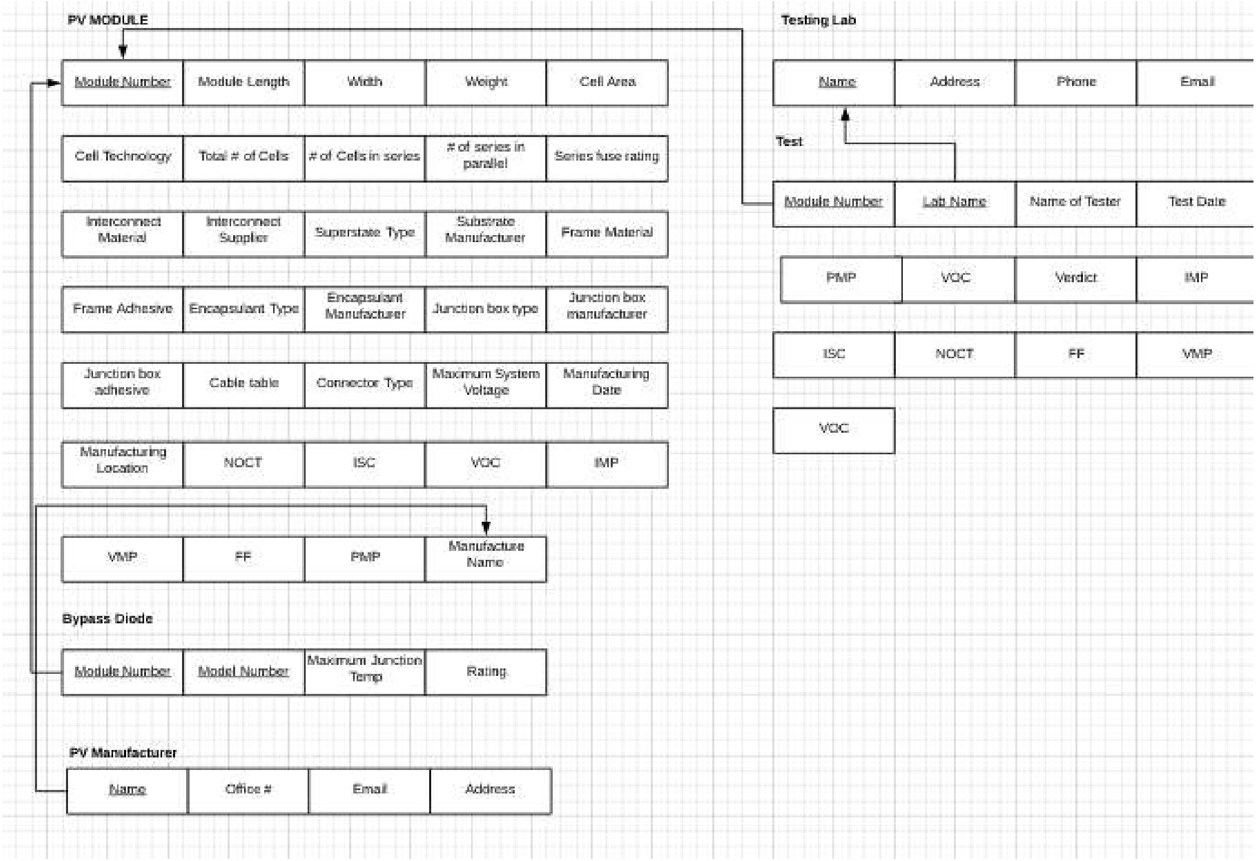


Entity	Entity	Relationship	Cardinalities
PV module	Diodes	has	1:M
PV module	PV Manufacturer	Built by	M:1
PV Module	Testing Lab	certified by	M:M

The following diagram is an extension of the conceptual diagram showing all the steps in the algorithm called an ER to Relational mapping.



Below is the completed Relational diagram with all the constraints, including a table showing the data types, such as string, number, etc. Here, Bypass Diode is a weak entity and is completely dependant of PV Module, thus the Bypass Diode will have a Foreign Key (Module Number) to reference PV Module. Next, PV Module has a many to one relationship with PV Manufacturer, thus PV Module will get the Foreign Key (Manufacturer Name) and the attributes of the relationship (Manufacturing Location and Manufacturing Date). Lastly, PV Module and Testing Lab have a many to many relationship, so there will be an intermediate table (Test) that will inherit all the attributes of the relationship and have a composite Foreign Key of Testing Lab Name and Module Number in order to reference both tables.



PV MODULE

Module Number	Module Length	Width	Weight	Cell Area
Cell Technology	Total # of Cells	# of Cells in series	# of series in parallel	Series fuse rating
Interconnect Material	Interconnect Supplier	Superstate Type	Substrate Manufacturer	Frame Material
Frame Adhesive	Encapsulant Type	Encapsulant Manufacturer	Junction box type	Junction box manufacturer
Junction box adhesive	Cable table	Connector Type	Maximum System Voltage	Manufacturing Date
Manufacturing Location	NOCT	ISC	VOC	IMP
VMP	FF	PMP	Manufacture Name	

Bypass Diode

Module Number	Model Number	Maximum Junction Temp	Rating

PV Manufacturer

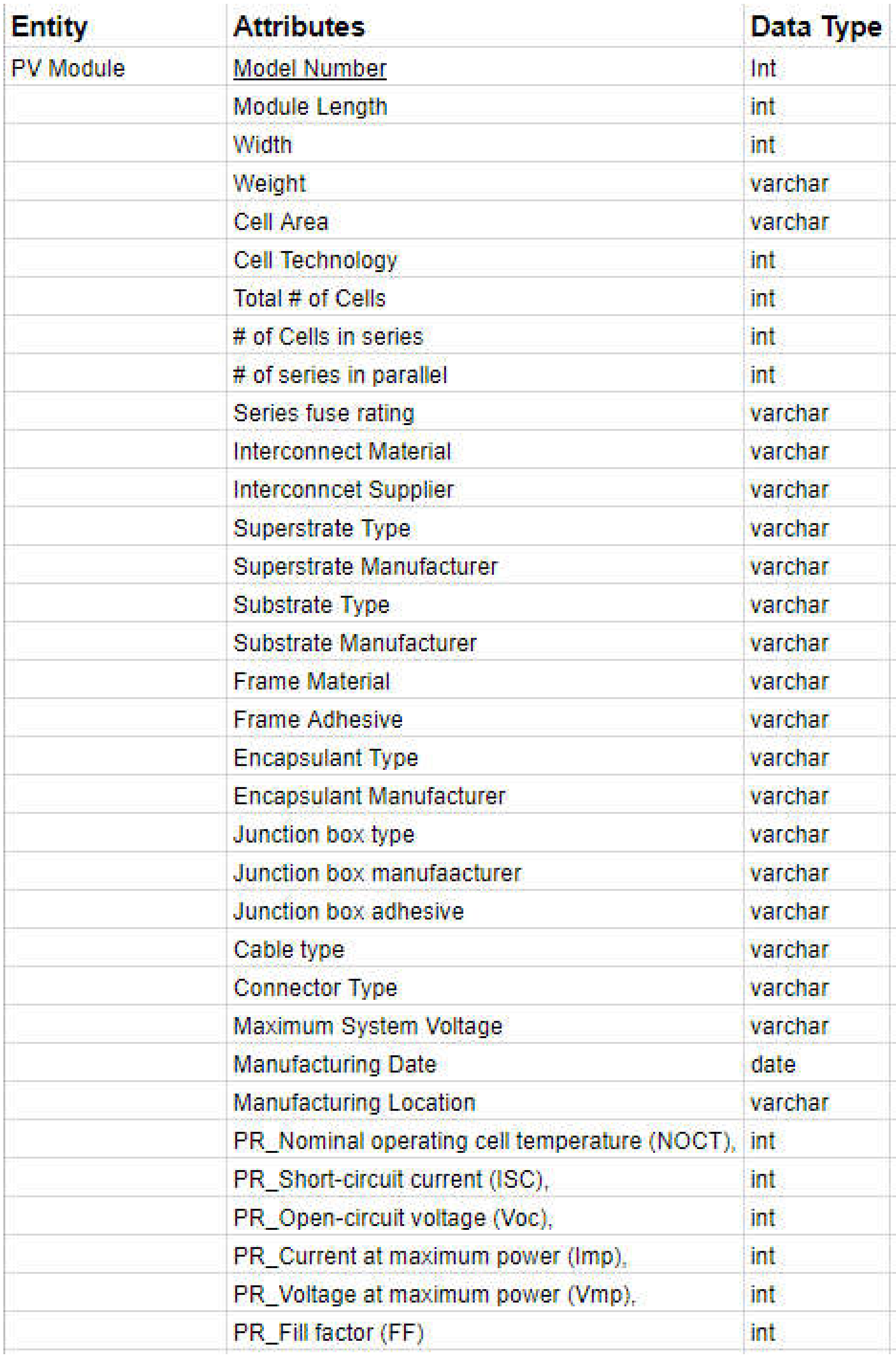
Name	Office #	Email	Address

Testing Lab

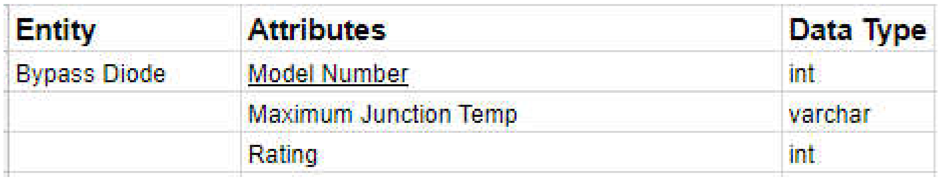
Name	Address	Phone	Email

Test

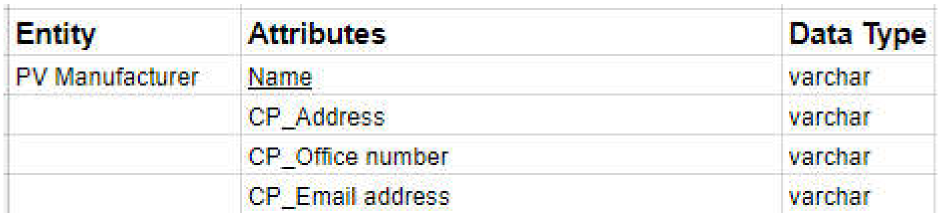
Module Number	Lab Name	Name of Tester	Test Date
PMP	VOC	Verdict	IMP
ISC	NOCT	FF	VMP
VOC			



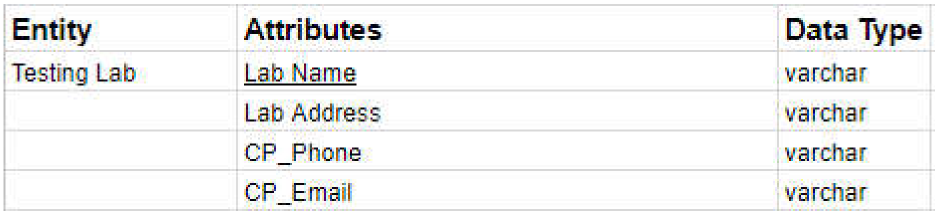
Entity	Attributes	Data Type
PV Module	Model Number	Int
	Module Length	int
	Width	int
	Weight	varchar
	Cell Area	varchar
	Cell Technology	int
	Total # of Cells	int
	# of Cells in series	int
	# of series in parallel	int
	Series fuse rating	varchar
	Interconnect Material	varchar
	Interconncet Supplier	varchar
	Superstrate Type	varchar
	Superstrate Manufacturer	varchar
	Substrate Type	varchar
	Substrate Manufacturer	varchar
	Frame Material	varchar
	Frame Adhesive	varchar
	Encapsulant Type	varchar
	Encapsulant Manufacturer	varchar
	Junction box type	varchar
	Junction box manufaacturer	varchar
	Junction box adhesive	varchar
	Cable type	varchar
	Connector Type	varchar
	Maximum System Voltage	varchar
	Manufacturing Date	date
	Manufacturing Location	varchar
	PR_Nominal operating cell temperature (NOCT),	int
	PR_Short-circuit current (ISC),	int
	PR_Open-circuit voltage (Voc),	int
	PR_Current at maximum power (Imp),	int
	PR_Voltage at maximum power (Vmp),	int
	PR_Fill factor (FF)	int



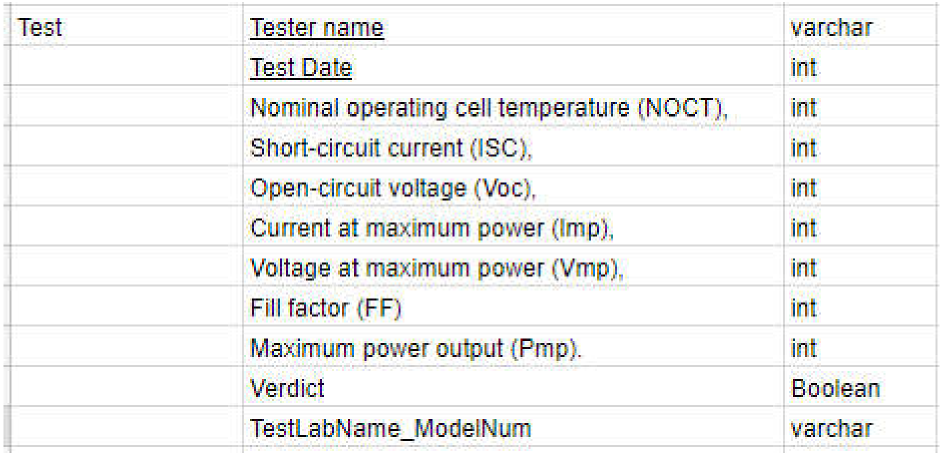
Entity	Attributes	Data Type
Bypass Diode	Model Number	int
	Maximum Junction Temp	varchar
	Rating	int



Entity	Attributes	Data Type
PV Manufacturer	Name	varchar
	CP_Address	varchar
	CP_Office number	varchar
	CP_Email address	varchar

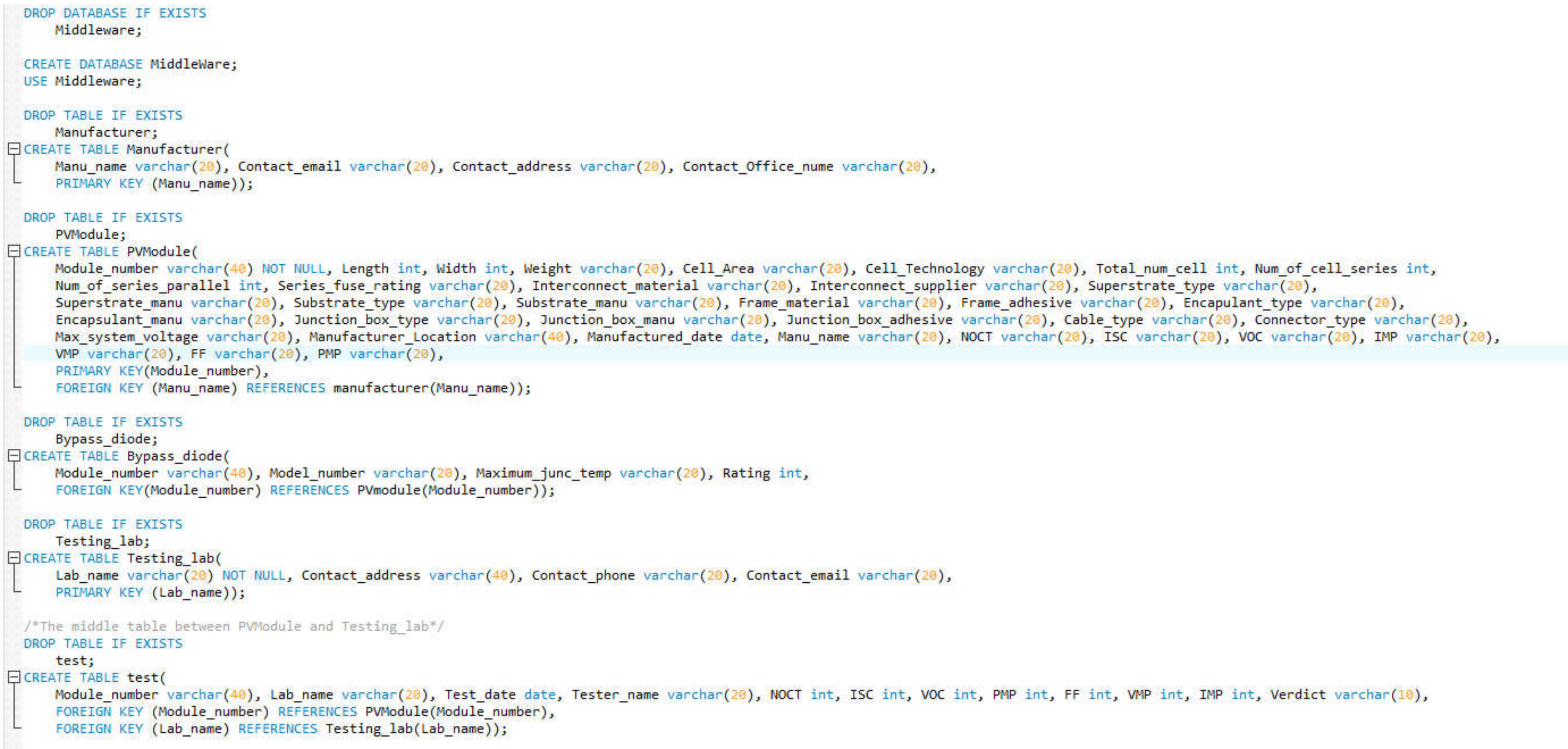


Entity	Attributes	Data Type
Testing Lab	Lab Name	varchar
	Lab Address	varchar
	CP_Phone	varchar
	CP_Email	varchar

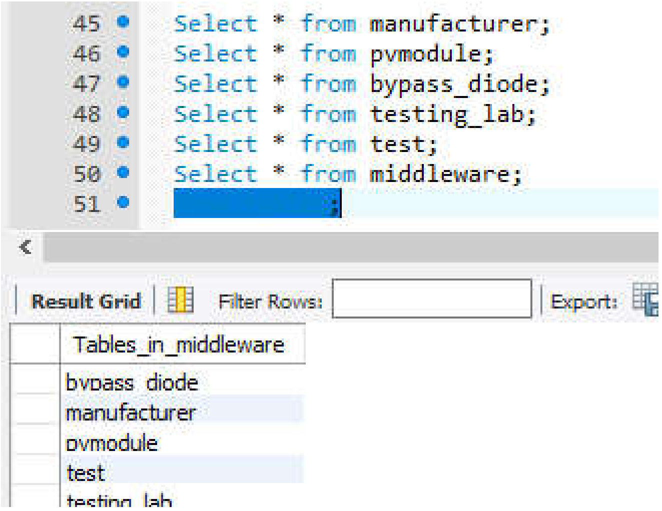


Test	Tester name	varchar
	Test Date	int
	Nominal operating cell temperature (NOCT),	int
	Short-circuit current (ISC),	int
	Open-circuit voltage (Voc),	int
	Current at maximum power (Imp),	int
	Voltage at maximum power (Vmp),	int
	Fill factor (FF)	int
	Maximum power output (Pmp).	int
	Verdict	Boolean
	TestLabName_ModelNum	varchar

The final step demonstrates the SQL queries in the Cygwin terminal, step by step, used to implement the Relation diagram design using MySQL database. Figure shows the statements used demonstrate that mySQL is installed, then the process of logging into mySQL (MariaDB) and finally the successful statement of creating a database.



```
DROP DATABASE IF EXISTS
    Middleware;

CREATE DATABASE MiddleWare;
USE Middleware;

DROP TABLE IF EXISTS
    Manufacturer;
CREATE TABLE Manufacturer(
    Manu_name varchar(20), Contact_email varchar(20), Contact_address varchar(20), Contact_Office_nume varchar(20),
    PRIMARY KEY (Manu_name));

DROP TABLE IF EXISTS
    PVModule;
CREATE TABLE PVModule(
    Module_number varchar(40) NOT NULL, Length int, Width int, Weight varchar(20), Cell_Area varchar(20), Cell_Technology varchar(20), Total_num_cell int, Num_of_cell_series int,
    Num_of_series_parallel int, Series_fuse_rating varchar(20), Interconnect_material varchar(20), Interconnect_supplier varchar(20), Superstrate_type varchar(20),
    Superstrate_manu varchar(20), Substrate_type varchar(20), Substrate_manu varchar(20), Frame_material varchar(20), Frame_adhesive varchar(20), Encapulant_type varchar(20),
    Encapsulant_manu varchar(20), Junction_box_type varchar(20), Junction_box_manu varchar(20), Junction_box_adhesive varchar(20), Cable_type varchar(20), Connector_type varchar(20),
    Max_system_voltage varchar(20), Manufacturer_Location varchar(40), Manufactured_date date, Manu_name varchar(20), NOCT varchar(20), ISC varchar(20), VOC varchar(20), IMP varchar(20),
    VMP varchar(20), FF varchar(20), PMP varchar(20),
    PRIMARY KEY(Module_number),
    FOREIGN KEY (Manu_name) REFERENCES manufacturer(Manu_name));

DROP TABLE IF EXISTS
    Bypass_diode;
CREATE TABLE Bypass_diode(
    Module_number varchar(40), Model_number varchar(20), Maximum_junc_temp varchar(20), Rating int,
    FOREIGN KEY(Module_number) REFERENCES PVmodule(Module_number));

DROP TABLE IF EXISTS
    Testing_lab;
CREATE TABLE Testing_lab(
    Lab_name varchar(20) NOT NULL, Contact_address varchar(40), Contact_phone varchar(20), Contact_email varchar(20),
    PRIMARY KEY (Lab_name));

/*The middle table between PVModule and Testing_lab*/
DROP TABLE IF EXISTS
    test;
CREATE TABLE test(
    Module_number varchar(40), Lab_name varchar(20), Test_date date, Tester_name varchar(20), NOCT int, ISC int, VOC int, PMP int, FF int, VMP int, IMP int, Verdict varchar(10),
    FOREIGN KEY (Module_number) REFERENCES PVModule(Module_number),
    FOREIGN KEY (Lab_name) REFERENCES Testing_lab(Lab_name));
```



```
Select * from manufacturer;
Select * from pvmodule;
Select * from bypass_diode;
Select * from testing_lab;
Select * from test;
Select * from middleware;
;
```

Tables_in_middleware
bypass_diode
manufacturer
pvmodule
test
testing_lab

User manual

The following databases can be constructed using the SQL file with the .sql extension. To execute this code, first open the .sql file in the MS MySQL environment. This will open the file in a query window. Next, highlight the code and click the "Execute"

Conclusion

In conclusion, completing this portion of the project allowed us to review the concepts learned in IFT 433 and served mostly as a refresher. We did, however, learn how to use mySQL vice MS SQL and the minor differences in syntax. Additionally, we learned that relationships can also have attributes themselves. The most challenging task we faced was being able to determine the proper relationships between tables. At first, we had improper many to many relationship and one to many relationships. We did overcome this hurdle by reading the PDF provided for this topic, listening to the online lectures, discussing with each other our thoughts, and finally from revisions from Dr. Kuitche. The only way that this backend design can be improved is if we add default values to the attributes and add constraints to verify user inputs.