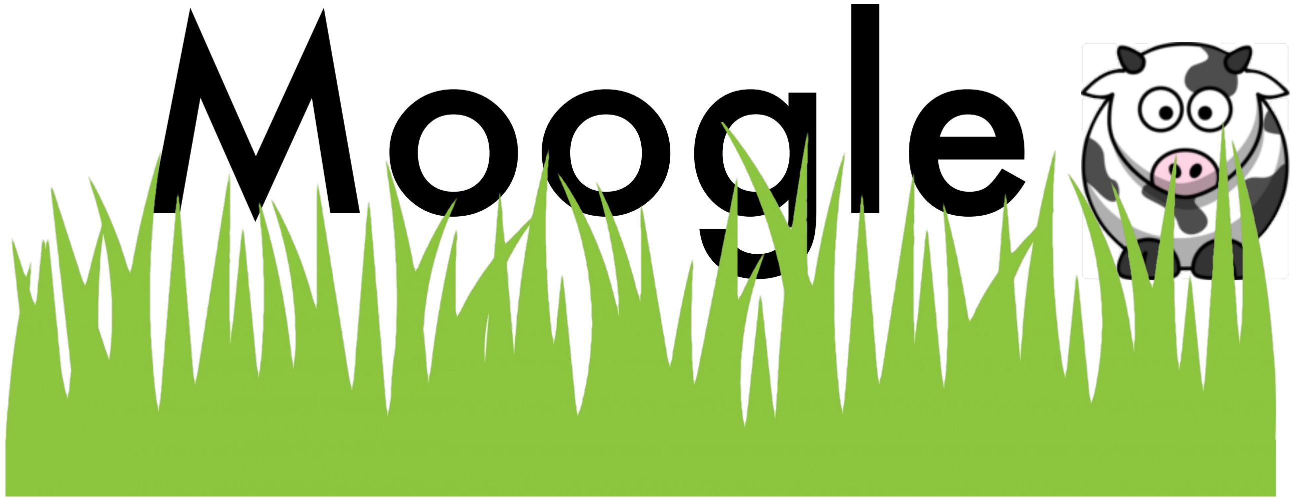
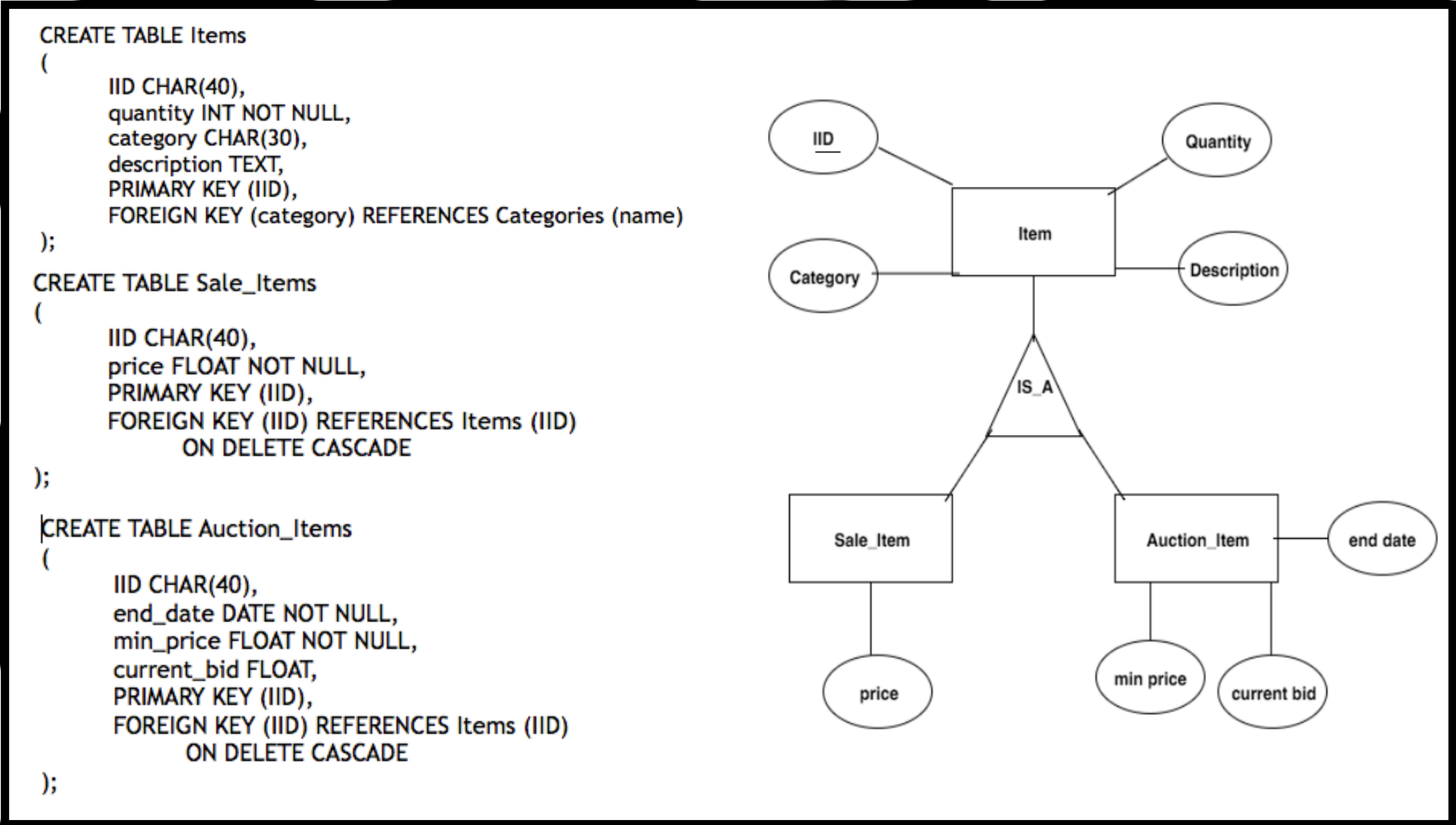
Moogle



Today..

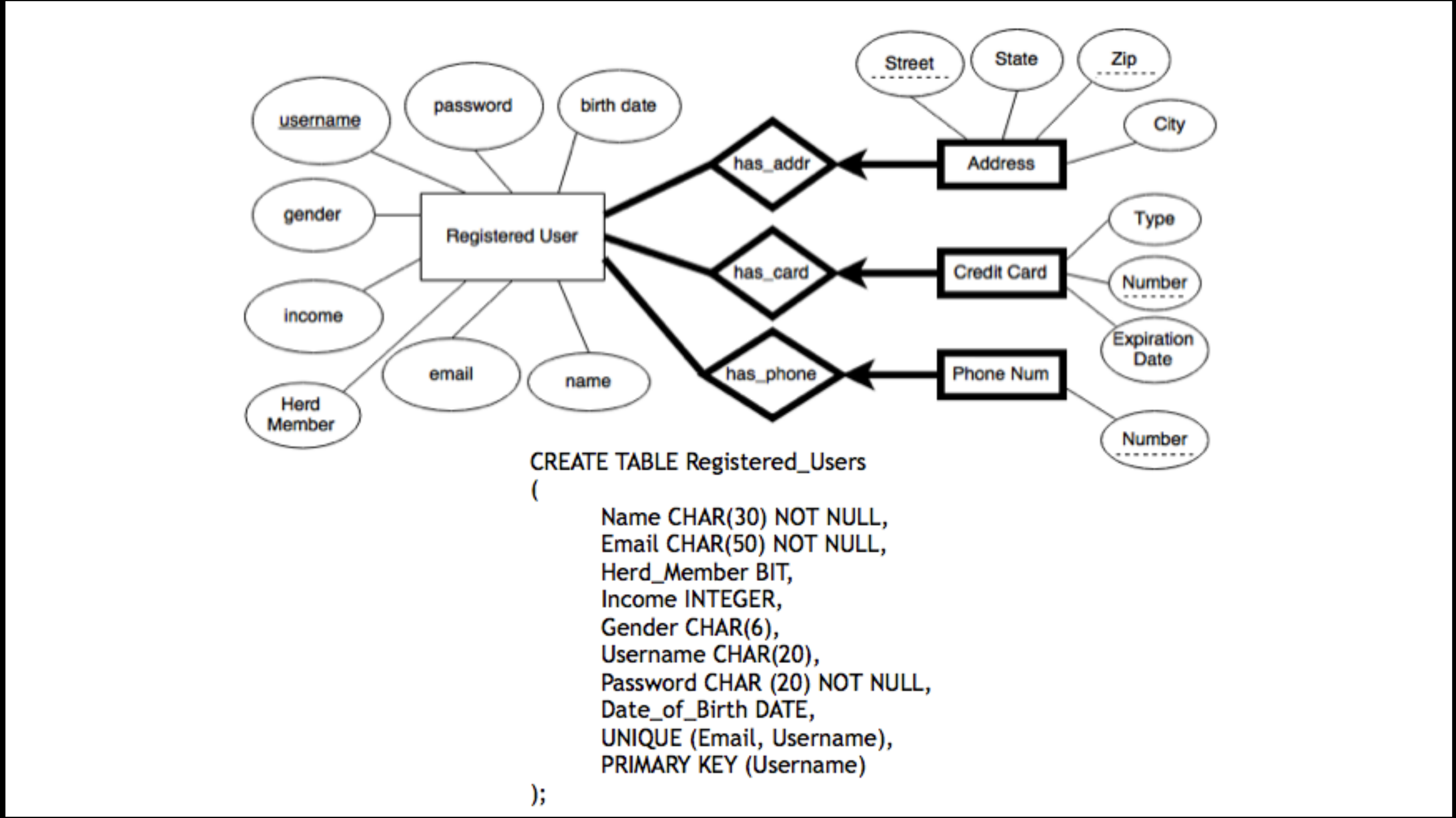
- ER → SQL
- Technology used
- Action plan

ER to SQL: Items



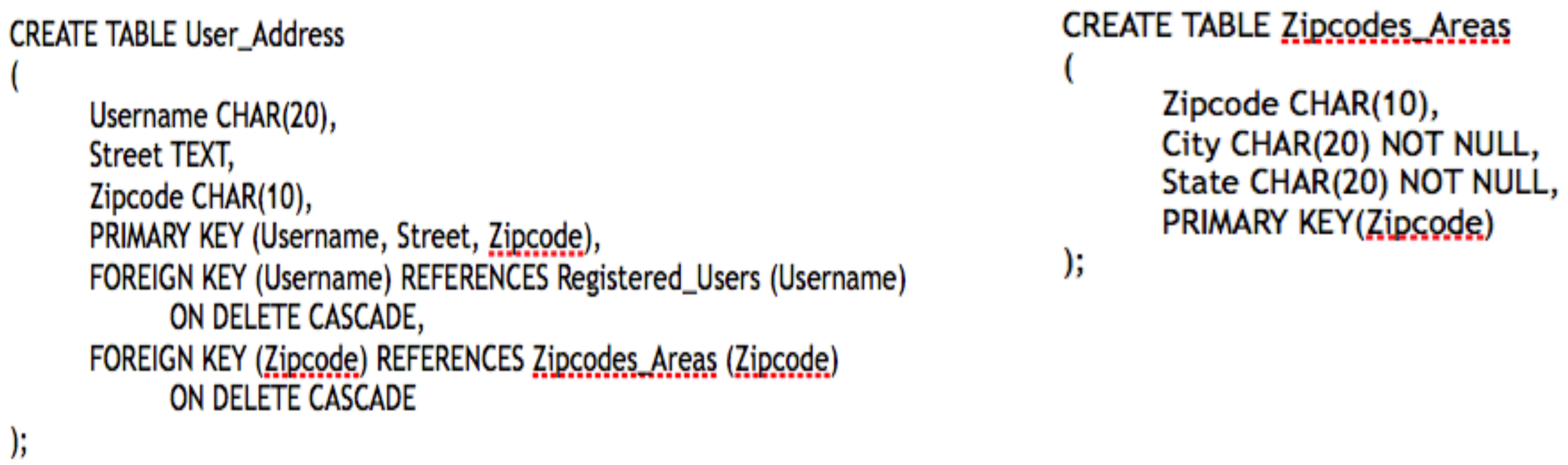
```
CREATE TABLE Items
(
        IID CHAR(40),
        quantity INT NOT NULL,
        category CHAR(30),
        description TEXT,
        PRIMARY KEY (IID),
        FOREIGN KEY (category) REFERENCES Categories (name)
);
CREATE TABLE Sale_Items
(
        IID CHAR(40),
        price FLOAT NOT NULL,
        PRIMARY KEY (IID),
        FOREIGN KEY (IID) REFERENCES Items (IID)
                ON DELETE CASCADE
);

CREATE TABLE Auction_Items
(
        IID CHAR(40),
        end_date DATE NOT NULL,
        min_price FLOAT NOT NULL,
        current_bid FLOAT,
        PRIMARY KEY (IID),
        FOREIGN KEY (IID) REFERENCES Items (IID)
                ON DELETE CASCADE
);
```

ER to SQL: Users



```
CREATE TABLE Registered_Users
(
        Name CHAR(30) NOT NULL,
        Email CHAR(50) NOT NULL,
        Herd_Member BIT,
        Income INTEGER,
        Gender CHAR(6),
        Username CHAR(20),
        Password CHAR (20) NOT NULL,
        Date_of_Birth DATE,
        UNIQUE (Email, Username),
        PRIMARY KEY (Username)
);
```

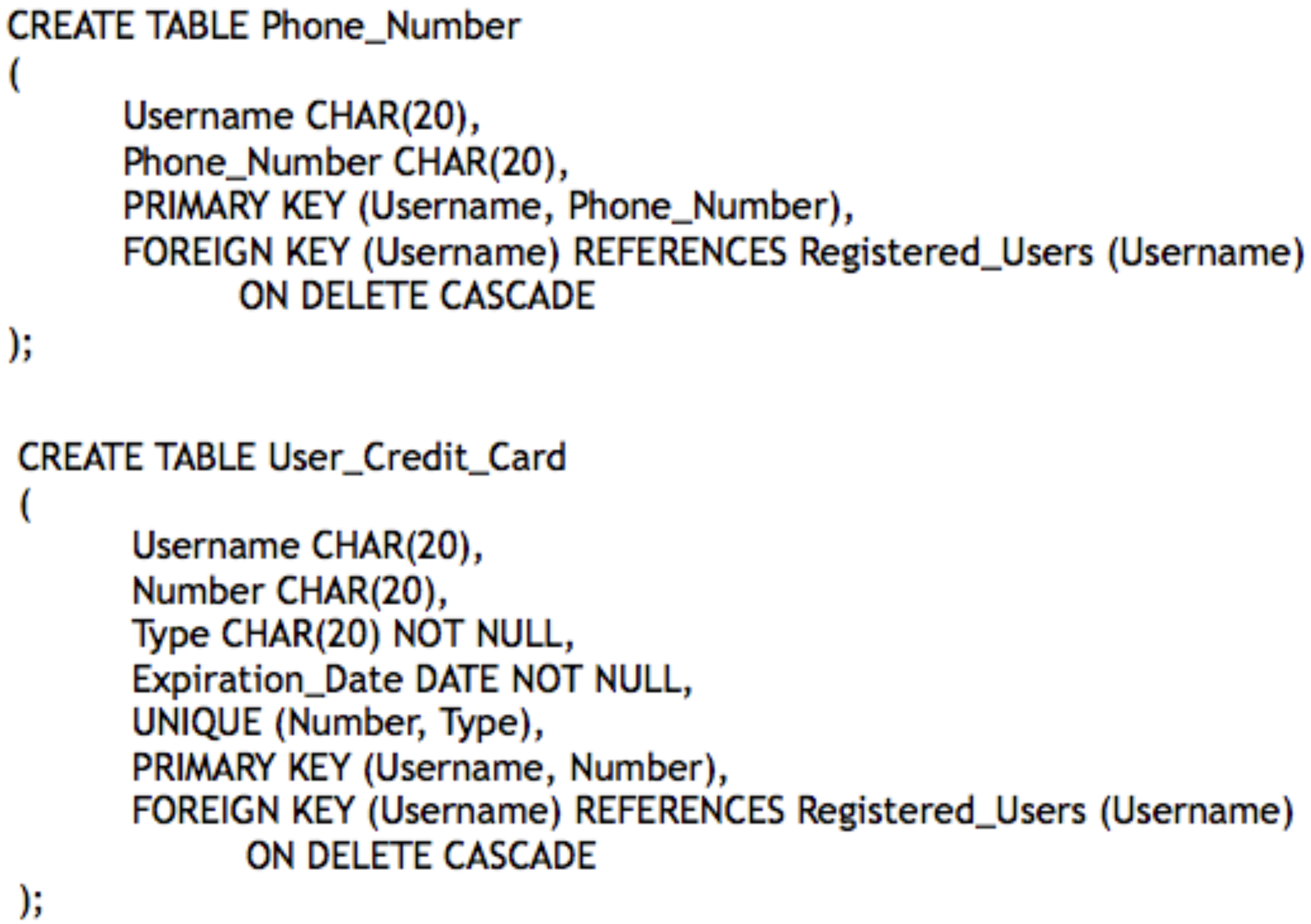
ER to SQL: Users (cont.)



```
CREATE TABLE User_Address
(
        Username CHAR(20),
        Street TEXT,
        Zipcode CHAR(10),
        PRIMARY KEY (Username, Street, Zipcode),
        FOREIGN KEY (Username) REFERENCES Registered_Users (Username)
                ON DELETE CASCADE,
        FOREIGN KEY (Zipcode) REFERENCES Zipcodes_Areas (Zipcode)
                ON DELETE CASCADE
);
```

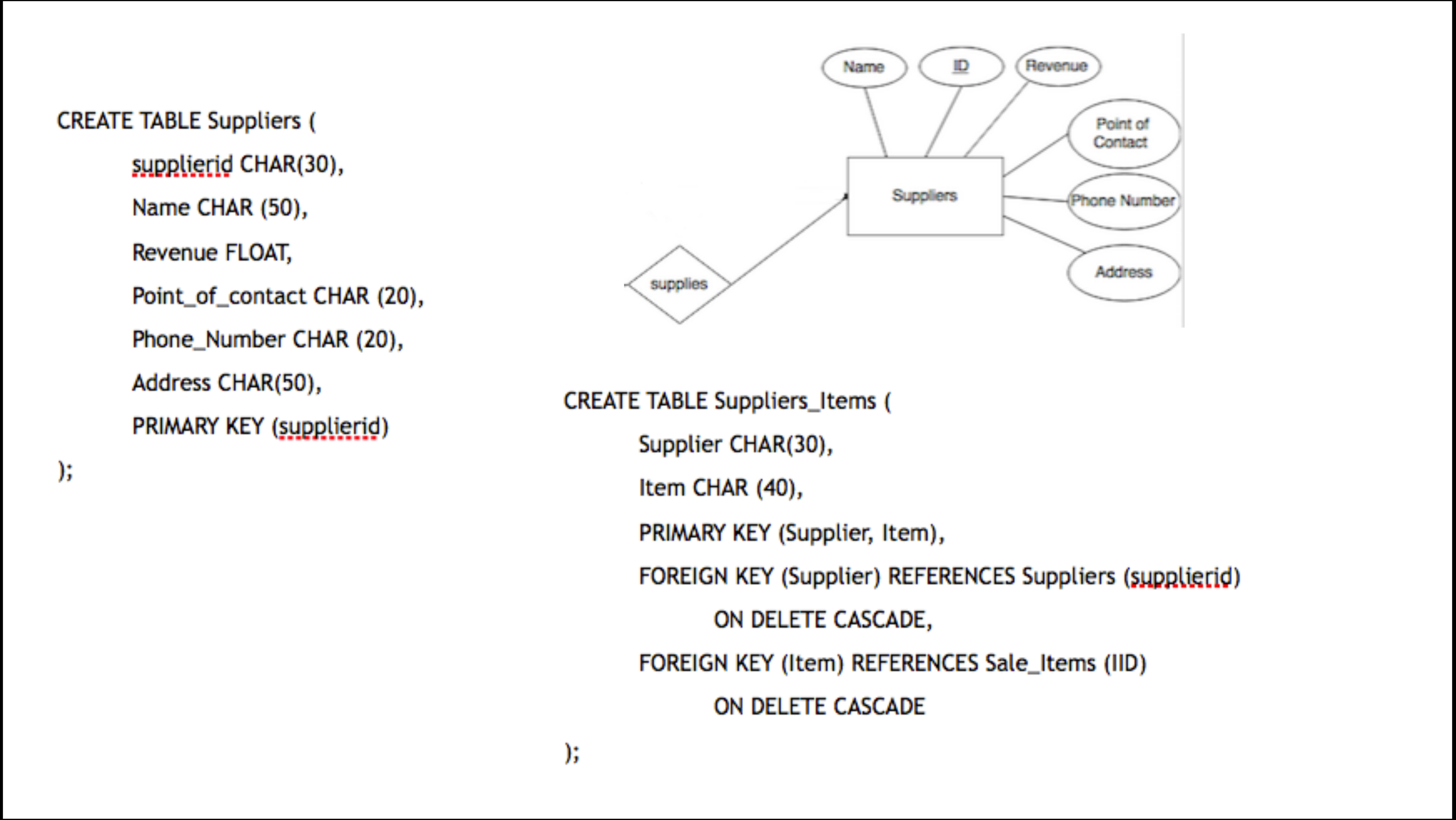
```
CREATE TABLE Zipcodes_Areas
(
        Zipcode CHAR(10),
        City CHAR(20) NOT NULL,
        State CHAR(20) NOT NULL,
        PRIMARY KEY(Zipcode)
);
```

ER to SQL: Users (cont.)



```
CREATE TABLE Phone_Number
(
        Username CHAR(20),
        Phone_Number CHAR(20),
        PRIMARY KEY (Username, Phone_Number),
        FOREIGN KEY (Username) REFERENCES Registered_Users (Username)
                ON DELETE CASCADE
);

 CREATE TABLE User_Credit_Card
 (
        Username CHAR(20),
        Number CHAR(20),
        Type CHAR(20) NOT NULL,
        Expiration_Date DATE NOT NULL,
        UNIQUE (Number, Type),
        PRIMARY KEY (Username, Number),
        FOREIGN KEY (Username) REFERENCES Registered_Users (Username)
                ON DELETE CASCADE
 );
```

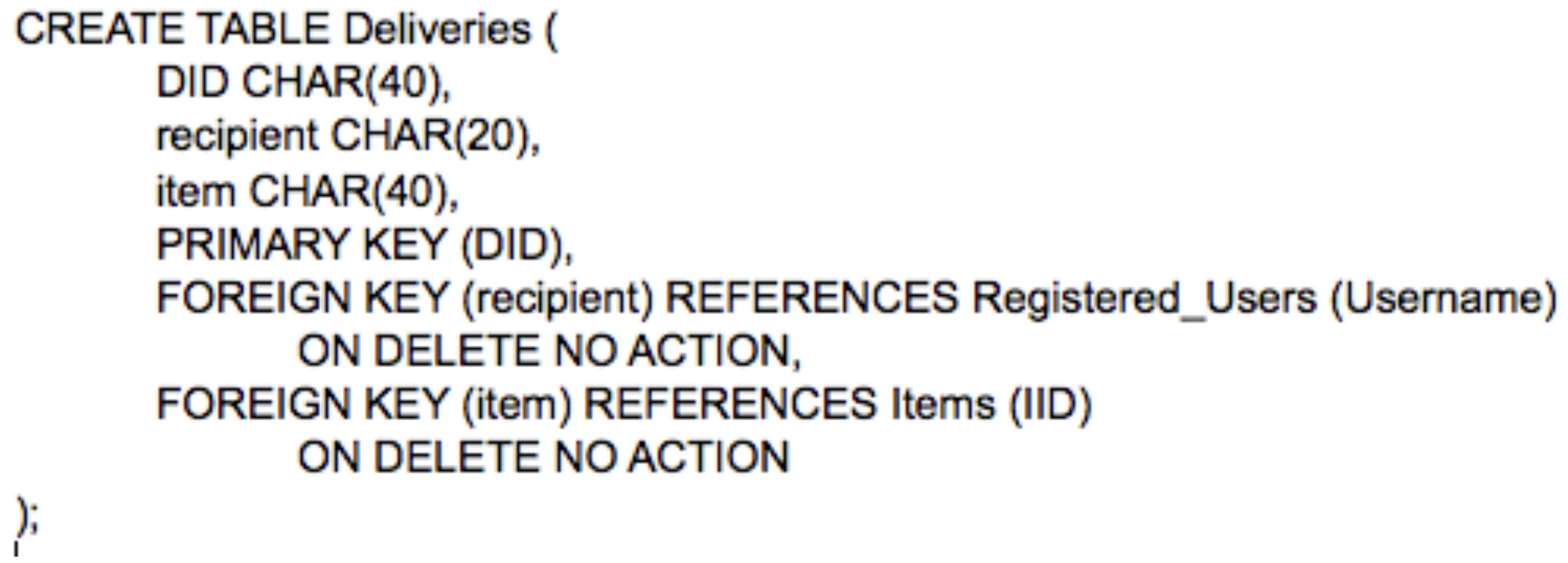
ER to SQL: Suppliers



```
CREATE TABLE Suppliers (
        supplierid CHAR(30),
        Name CHAR (50),
        Revenue FLOAT,
        Point_of_contact CHAR (20),
        Phone_Number CHAR (20),
        Address CHAR(50),
        PRIMARY KEY (supplierid)
);
```

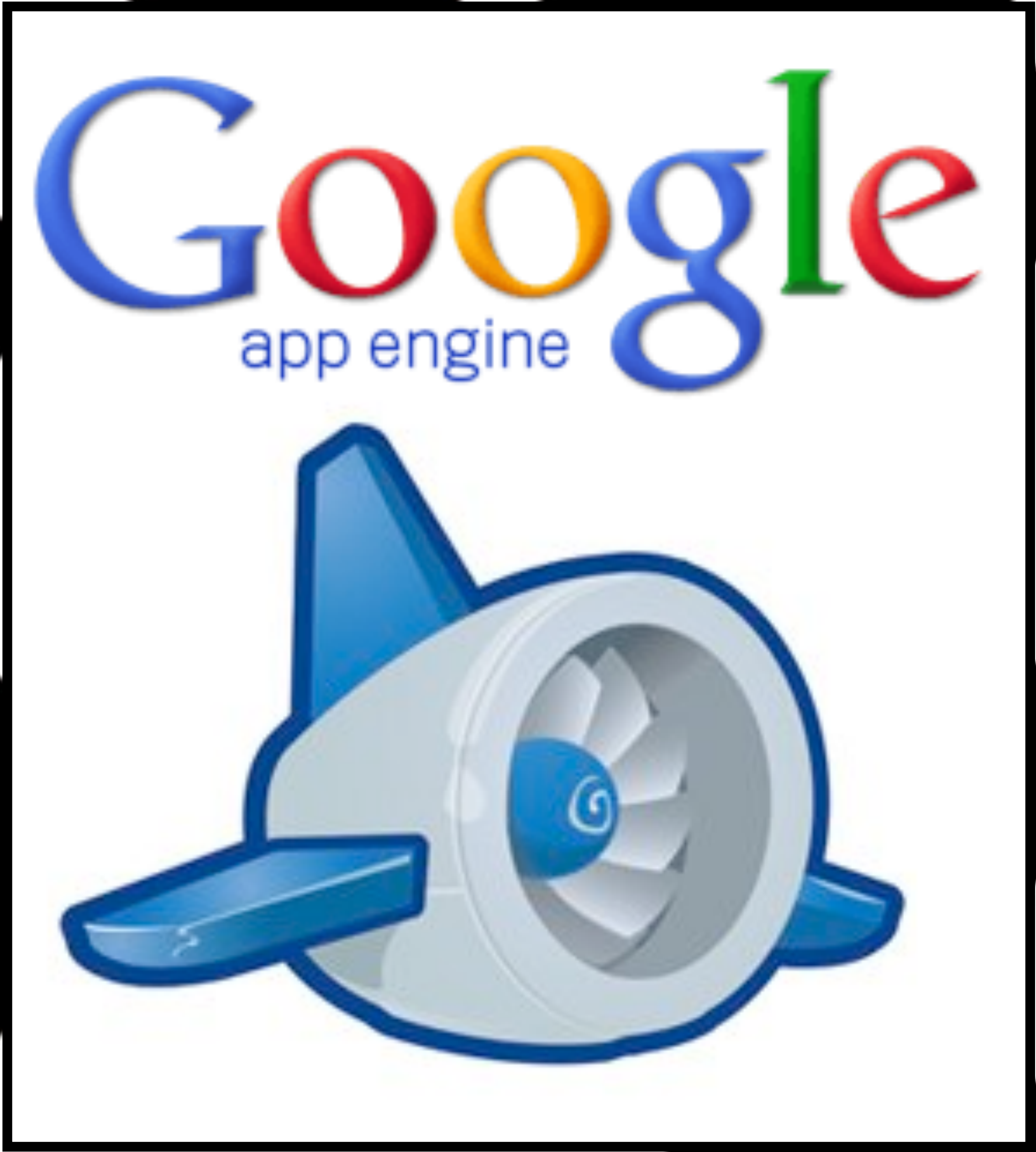
```
CREATE TABLE Suppliers_Items (
        Supplier CHAR(30),
        Item CHAR (40),
        PRIMARY KEY (Supplier, Item),
        FOREIGN KEY (Supplier) REFERENCES Suppliers (supplierid)
                ON DELETE CASCADE,
        FOREIGN KEY (Item) REFERENCES Sale_Items (IID)
                ON DELETE CASCADE
);
```

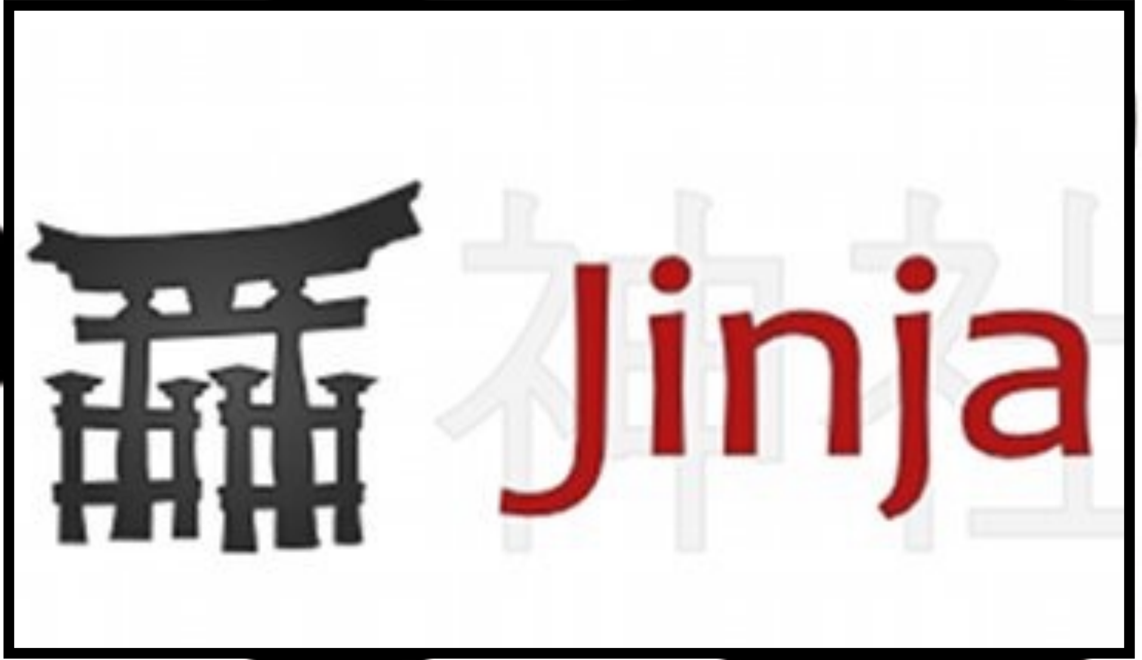
ER to SQL: Deliveries

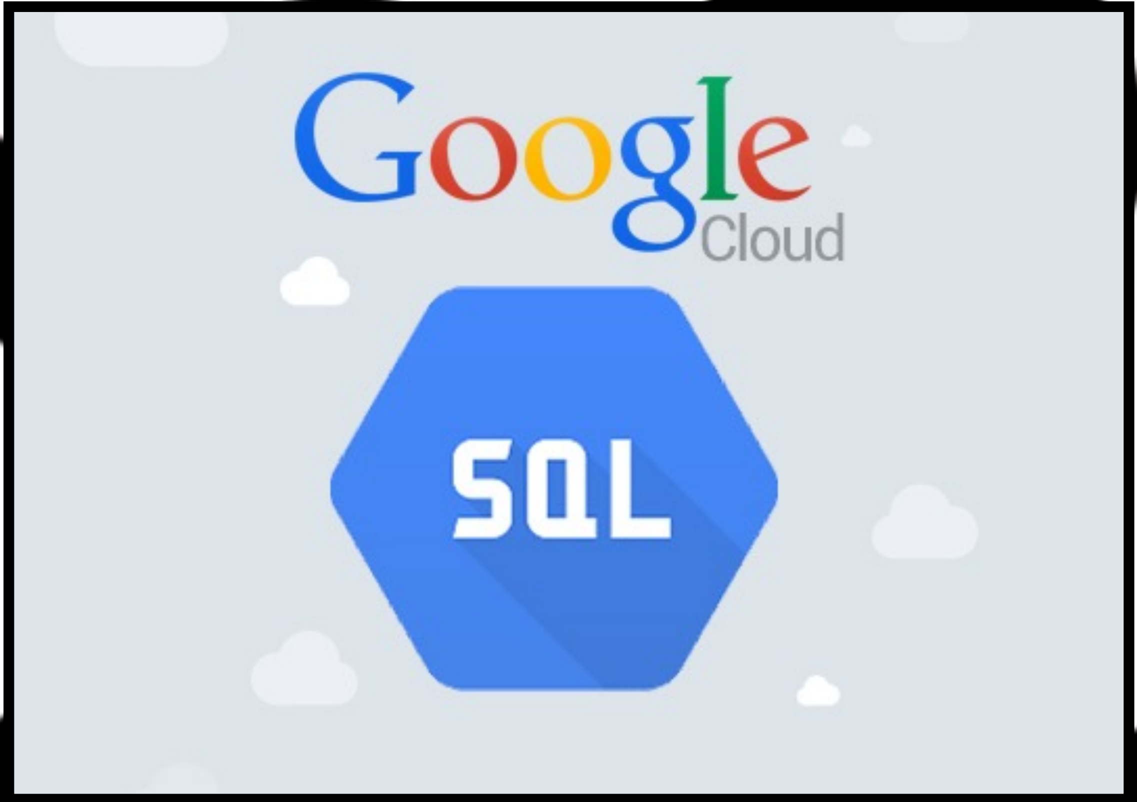


```
CREATE TABLE Deliveries (
        DID CHAR(40),
        recipient CHAR(20),
        item CHAR(40),
        PRIMARY KEY (DID),
        FOREIGN KEY (recipient) REFERENCES Registered_Users (Username)
                ON DELETE NO ACTION,
        FOREIGN KEY (item) REFERENCES Items (IID)
                ON DELETE NO ACTION
);
```

Technology







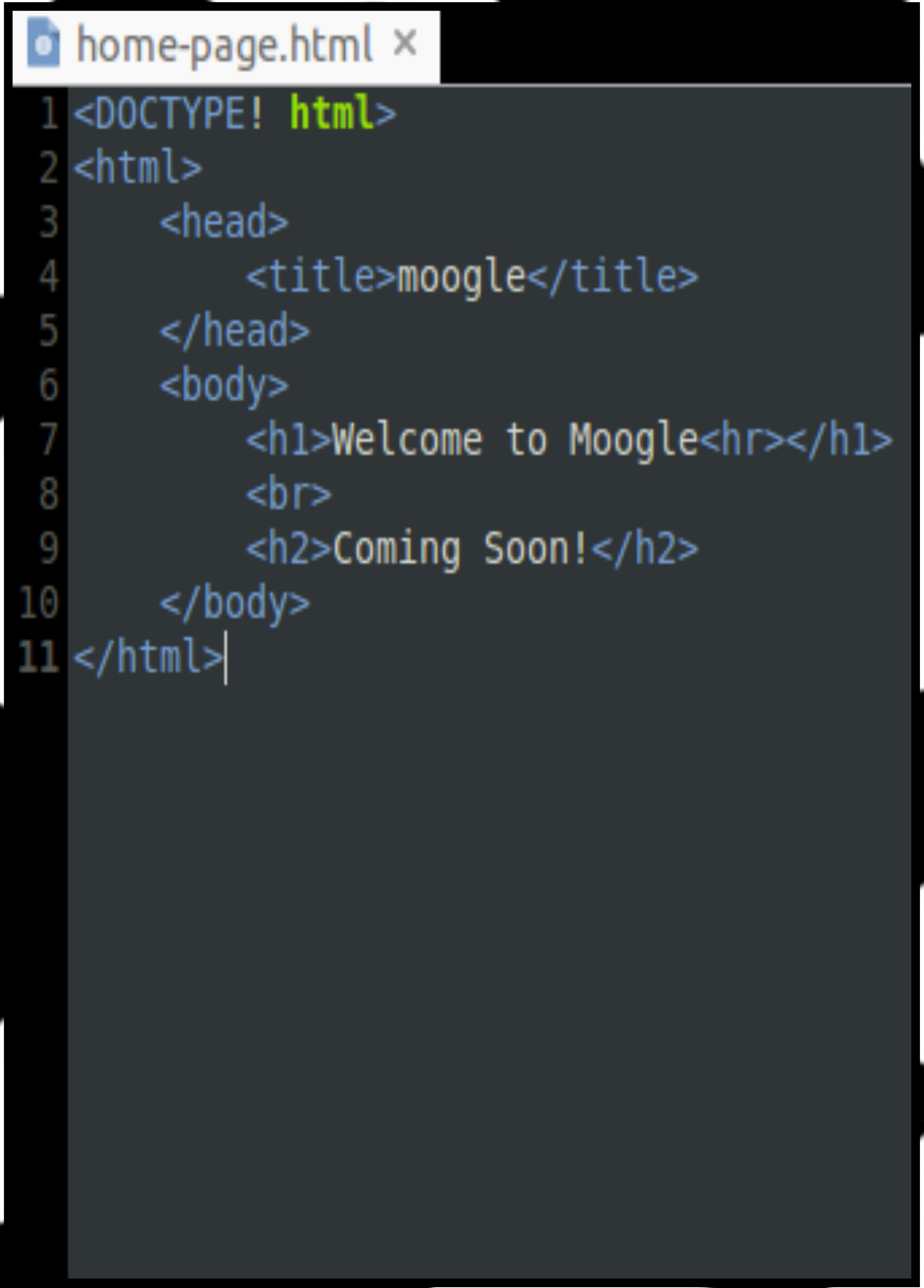
Google App Engine

- “a platform as a service (PaaS) cloud computing platform for developing and hosting web applications in Google-managed data centers”
- Google: “You worry about how you want your web app to look (HTML/CSS) and handle data (Python) and we’ll worry about server issues and scaling”
- Benefits…

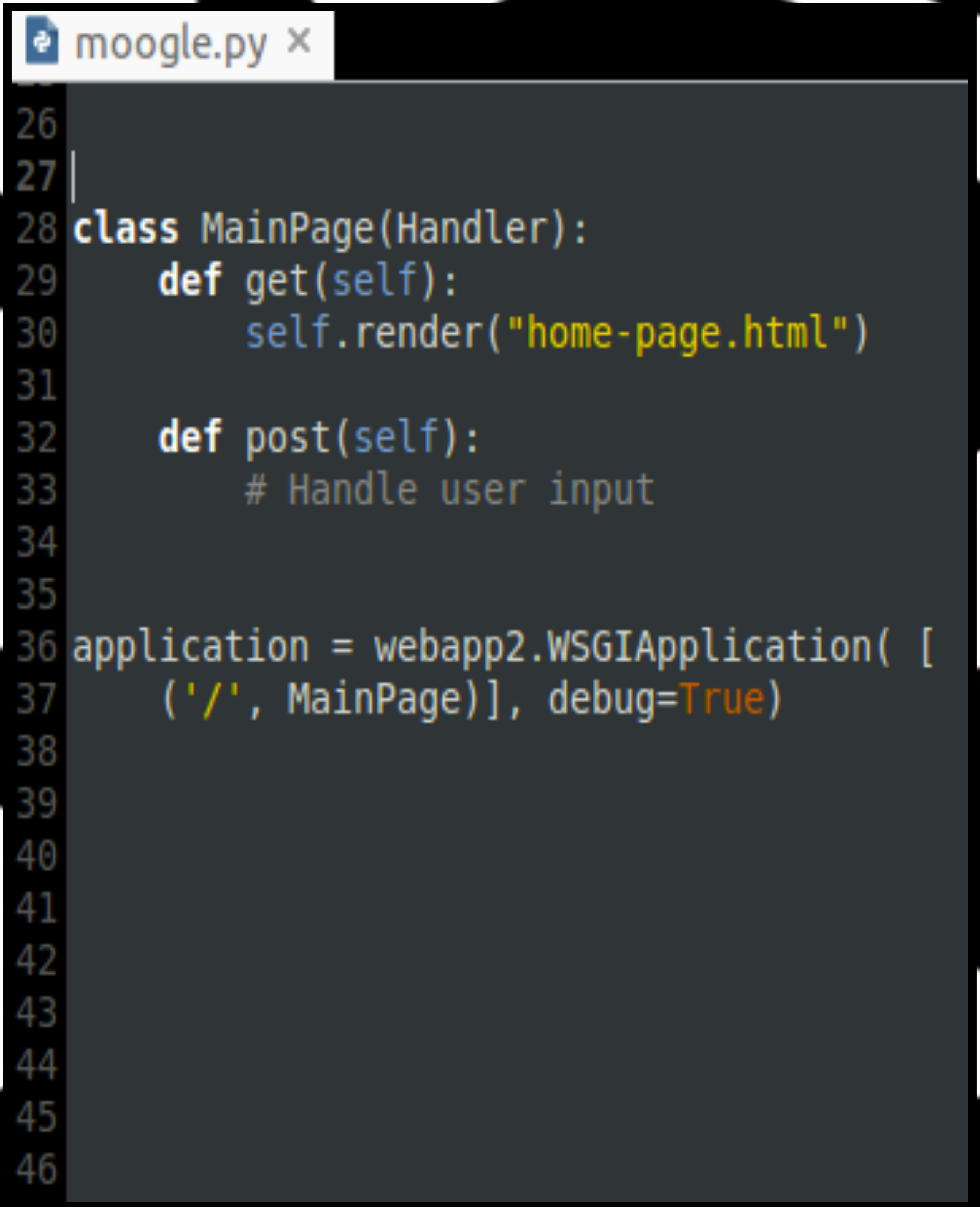
It’s Free

- Google App Engine is free to use up to a certain level of consumed resources
    - Good for a project that probably won’t live past May..
- If your app starts to draw A LOT of requests or consumes a lot of resources, you’ll be charged for additional storage/bandwidth
    - Not a major concern for this project

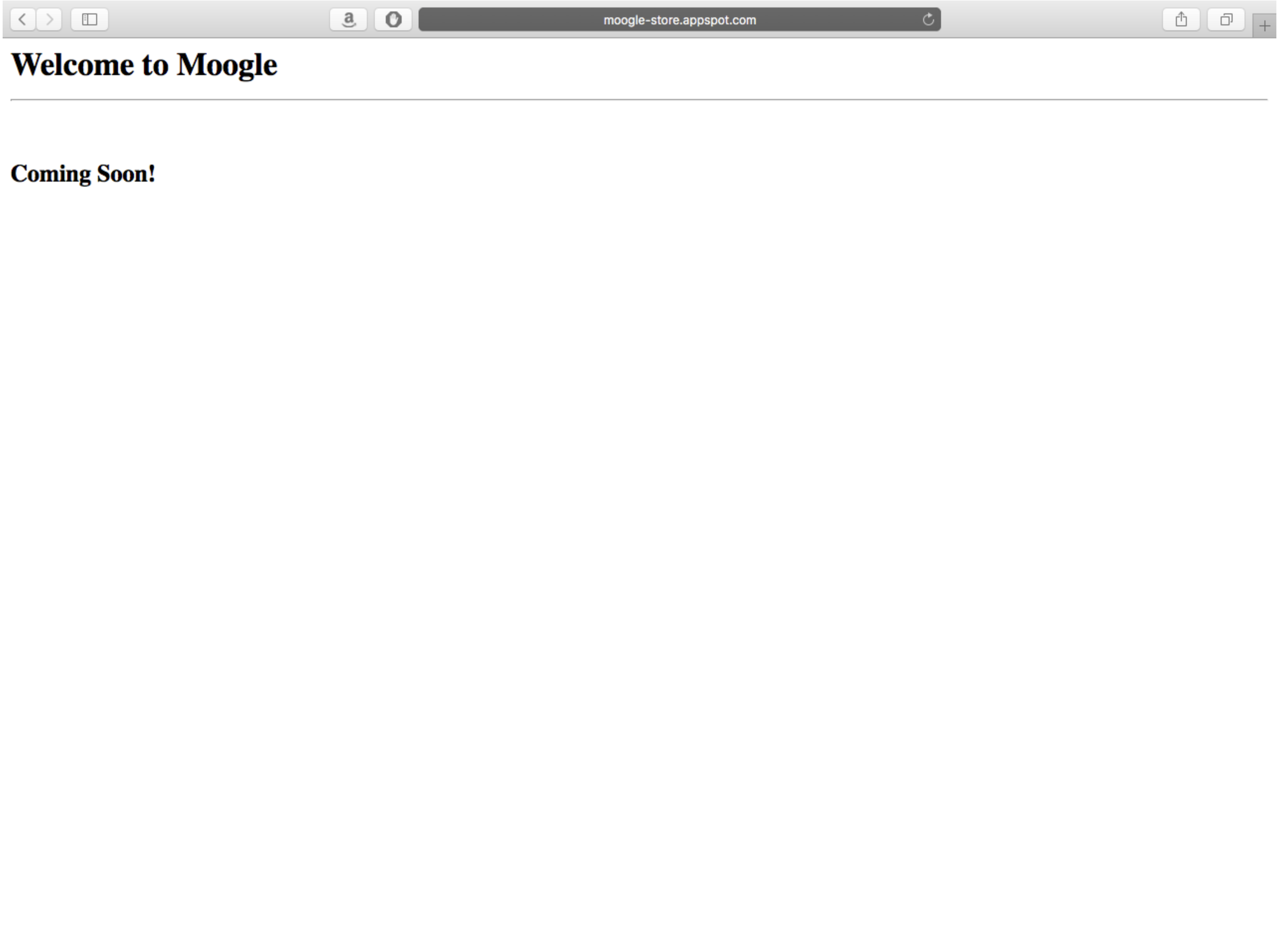
Easy to Use



```html
<DOCTYPE! html>
<html>
    <head>
        <title>moogle</title>
    </head>
    <body>
        <h1>Welcome to Moogle<hr></h1>
        <br>
        <h2>Coming Soon!</h2>
    </body>
</html>
```



```python


class MainPage(Handler):
    def get(self):
        self.render("home-page.html")

    def post(self):
        # Handle user input


application = webapp2.WSGIApplication( [
    ('/', MainPage)], debug=True)
```



Welcome to Moogle

Coming Soon!

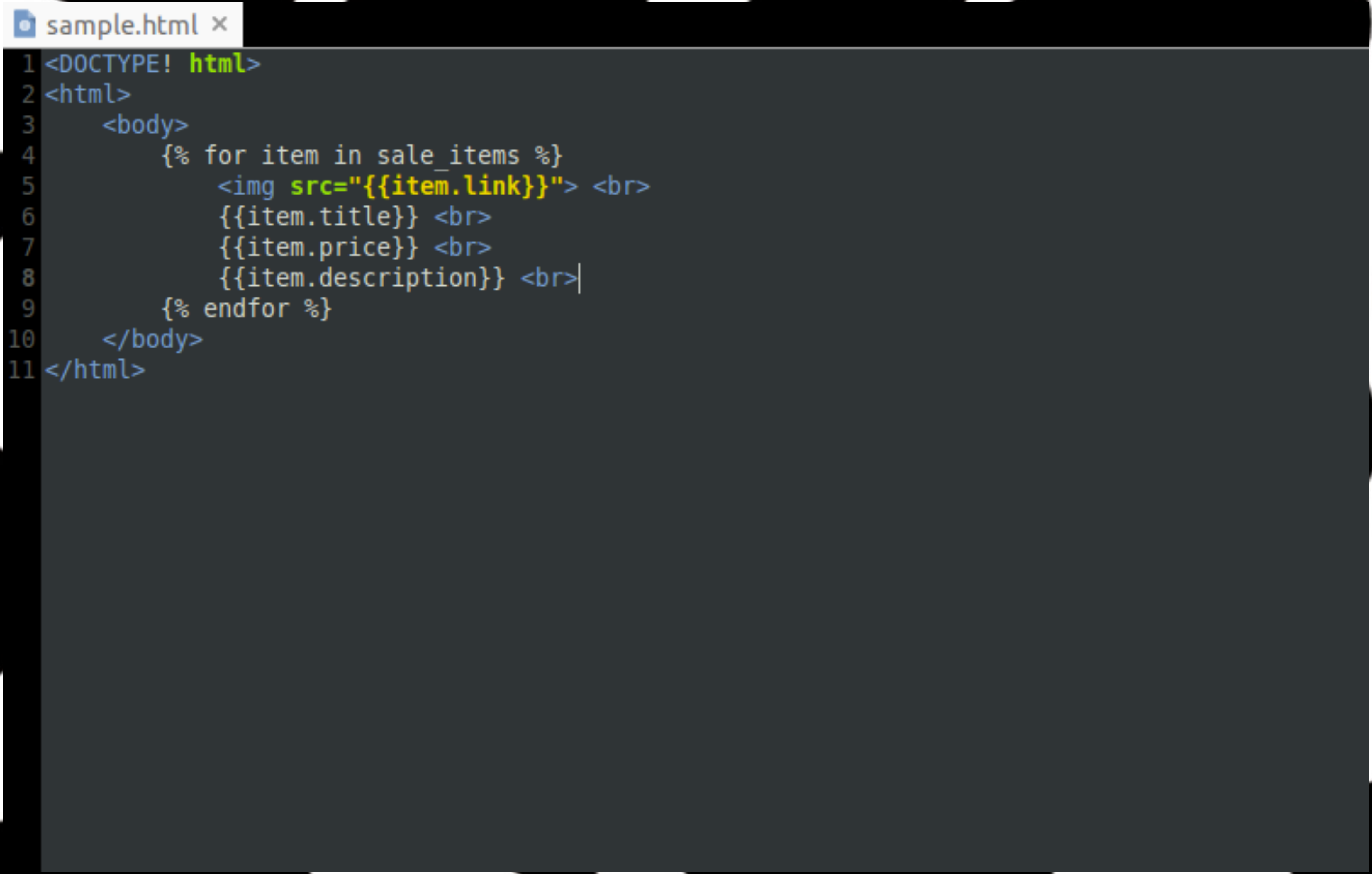
Provides MySQL Support

- Google Cloud SQL
  - Allows you to create DB instance that your App Engine app can access
  - Can connect, post, and get from your database within your Python code
- Google Cloud Datastore
  - Uses SQL-like language, GQL
  - Easier to use within the Google App Engine environment
  - Very similar to SQL, but some differences/restrictions (no “Join”s)

Jinja2

- Template engine for the Python programming language
- Brings Python-like syntax to HTML (for loops, if statements, etc.)
- Automatic HTML escaping
- Template inheritance
- Easy to debug

Using Jinja2



```html
<DOCTYPE! html>
<html>
    <body>
        {% for item in sale_items %}
            <img src="{{item.link}}"> <br>
            {{item.title}} <br>
            {{item.price}} <br>
            {{item.description}} <br>
        {% endfor %}
    </body>
</html>
```

Learn more

- Free course at Udacity
- https://appengine.google.com
- http://jinja.pocoo.org/docs/dev/
- Visit Moogle now at : moogle-store.appspot.com

Action Plan

- Populate database using Scrapy or BeautifulSoup
- Build the website!