Vanessa Venkataraman
DATA-202
Milestone 2

**Overview of Database**:

This database represents a cocktail bar. The two main focuses of this database are the employees that work at the cocktail bar and the products sold at the cocktail bar. In terms of the employees that work at the cocktail bar, the database holds their personal contact information, their employment information, and their work shift information. In terms of the products sold at the cocktail bar, the database holds information regarding spirit inventory, cocktail recipes, liquor purchases, and liquor consumption. In terms of the cocktail bar as a whole, the database holds information regarding the financial report of each night that it is open for service. Below are more detailed descriptions of the function of each table in the database and all the variables that are present in each table.

**Overview of Tables:**

cocktail: contains information about ingredients and how to make a cocktail on the menu

- □ cocktail_name - name of the cocktail
- □ spirit_name - name of the main spirit in the cocktail
- □ ounces_used - quantity in ounces of the main spirit used in the cocktail
- □ spec - all ingredients and their quantities used in the cocktail
- □ build_method - how to make the cocktail (usually either shaken or stirred)
- □ glassware - glass that the cocktail goes into
- □ garnish - decoration for the cocktail

spirits: contains information about the spirits that are carried by the bar

- □ spirit_name - name of the spirit
- □ spirit_type - type of spirit
- □ bottle_price - price of an individual bottle
- □ bottle_size_ml - how much of the spirit is in each bottle in millileters
- □ case_price - price of a case of the spirit (always 6 bottles)
- □ order_id - order id number of an order for the spirit

inventory_order: contains information about each and every liquor order for the bar

- □ order_date - date of the order elivery
- □ order_id - order id number of an order for a spirit
- □ spirit_name - name of the spirit ordered
- □ num_bottles - number of bottles of that spirit ordered
- □ bill - cost of the order

jobtitle: contains information about each position at the bar

- □ position_id - unique number representing each unique position
- □ title - name of the position
- □ hourly_wage - hourly rate that an employee is paid for the position
- □ tip_point_average - weight of percentage of total tips received for the position

jposition: contains information that links an employee with a position and provides a daterange of employment

- □ position_id - unique number representing each unique position
- □ employee_id - unique number representing each unique employee
- □ employment - range of dates that an employee is in a particular position (ends in infinity is currently in that position)

staff: contains personal information for each employee

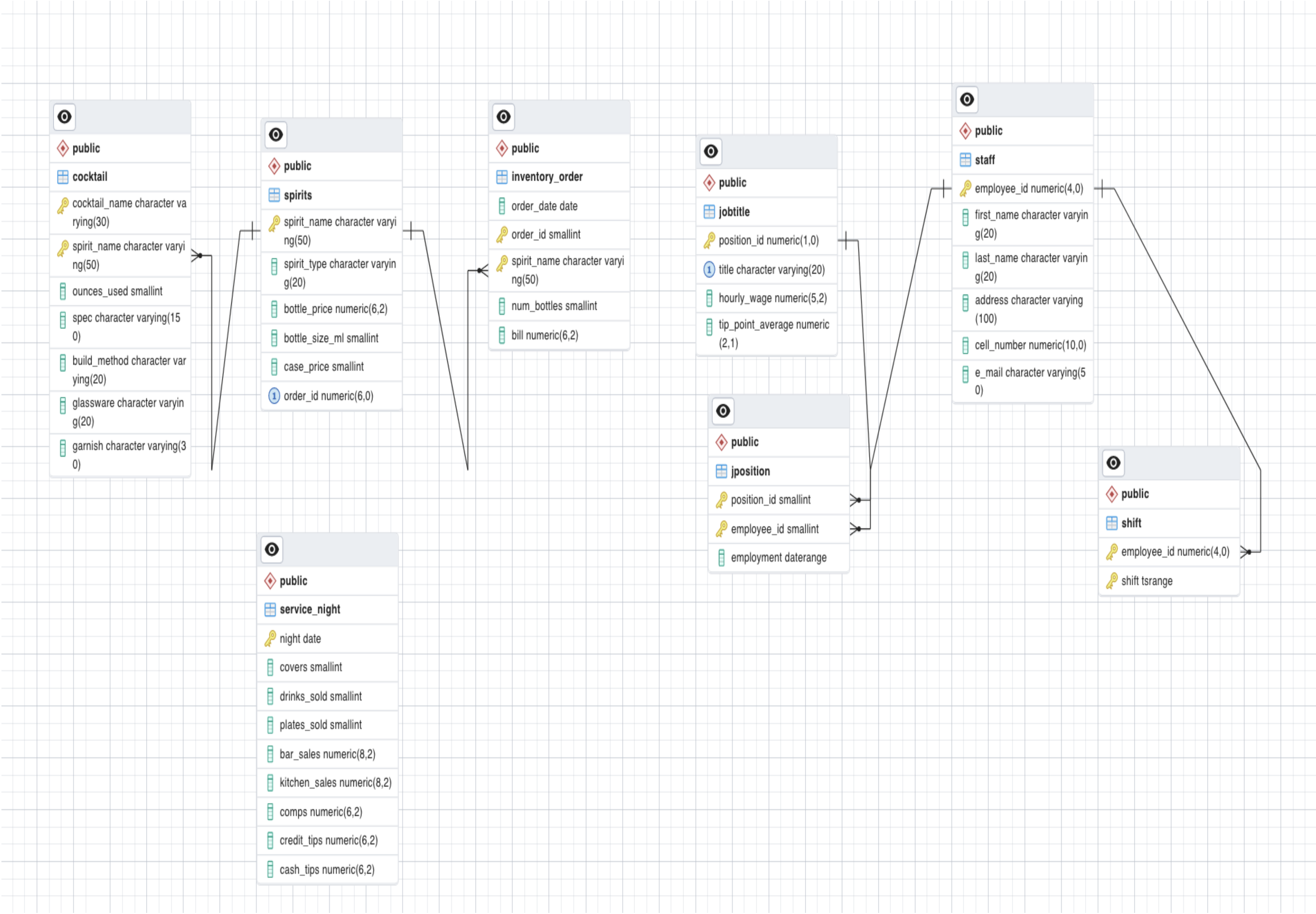
- □ employee_id - unique number representing a unique employee
- □ first_name - first name of employee
- □ last_name - last name of employee
- □ address - address of employee
- □ cell_number - cellphone number of employee
- □ e_mail - e_mail of employee

shift: contains information about each employee and the days/times that they worked

- □ employee_id - unique number representing a unique employee
- □ shift - timestamp range of the entire shift

service_night: contains information about the financials of each night that the bar is open for business

- □ night - date of the service night
- □ covers - amount of people that were guests in the bar that night
- □ drinks_sold - number of beverages sold that night
- □ plates_sold - number of plates sold that night
- □ bar_sales - amount of money in sales collected by the bar
- □ kitchen_sales - amount of money in sales collected by the kitchen
- □ comps - amount of money in sales gifted to guest due to complaints/employee error
- □ credit_tips - amount of money tipped using credit card
- □ cash_tips - amount of money tipped using cash


public
cocktail
cocktail_name character varying(30)
spirit_name character varying(50)
ounces_used smallint
spec character varying(150)
build_method character varying(20)
glassware character varying(20)
garnish character varying(30)
public
spirits
spirit_name character varying(50)
spirit_type character varying(20)
bottle_price numeric(6,2)
bottle_size_ml smallint
case_price smallint
order_id numeric(6,0)
public
inventory_order
order_date date
order_id smallint
spirit_name character varying(50)
num_bottles smallint
bill numeric(6,2)
public
jobtitle
position_id numeric(1,0)
title character varying(20)
hourly_wage numeric(5,2)
tip_point_average numeric(2,1)
public
jposition
position_id smallint
employee_id smallint
employment daterange
public
staff
employee_id numeric(4,0)
first_name character varying(20)
last_name character varying(20)
address character varying(100)
cell_number numeric(10,0)
e_mail character varying(50)
public
shift
employee_id numeric(4,0)
shift tsrange
public
service_night
night date
covers smallint
drinks_sold smallint
plates_sold smallint
bar_sales numeric(8,2)
kitchen_sales numeric(8,2)
comps numeric(6,2)
credit_tips numeric(6,2)
cash_tips numeric(6,2)

I am the owner of the cocktail bar and I have hired an underling named Sebastian to keep the data in the database up to date…

**Adding Data**

*Sebastian: When a new employee is hired, execute the following commands to add the new employee into the database (Note: employee_id will be the last 4 digits of their SSN):*

```
INSERT INTO staff(employee_id,first_name,last_name,address,cell_number,e_mail)
VALUES (...)
```

**In place of the ‘…’ fill in the employee’s info as it appears in the parentheses in the line above**

Ex. (1864,'Chris','Agave','407 Ocean Lane Revere,MA 02151',7289047645,'chrisagave123@gmail.com')

```
INSERT INTO jposition(position_id,employee_id,employment)
VALUES (...)
```

**In place of the ‘...’ fill in the corresponding position id as shown in the jobtitle table, employee id, and a date with start date as day of first shift and end date as infinity**

Ex. (3,1864,'[2023-02-01, infinity)')

*Sebastian: At the end of every night of service, execute the following commands to add all shifts to the database:*

```
INSERT INTO shift(employee_id,shift)
VALUES(...)
```

**In place of the ‘...’ fill in the id of each employee followed by the timestamp range of the entirety of their shift**

Ex. (1864,'[2023-01-01 15:00:00, 2023-01-02 2:00:00]')

*Sebastian: When the restaurant decides to carry a new spirit, execute the following command to enter the new spirit into the database:*

```
INSERT INTO spirits(spirit_name,spirit_type,bottle_price,bottle_size_ml,case_price)
VALUES(...)
```

**In place of the ‘...’ fill in the corresponding info as it appears in the parentheses in the line above**

Ex. ('Mijenta Blanco', 'tequila', 48.99, 750, 240.00)

*Sebastian: At the end of every night of service, execute the following commands to enter the financial report into the database:*

```
INSERT INTO
service_night(night,covers,drinks_sold,plates_sold,bar_sales,kitchen_sales,comps,credit_tips,cash_tips)
VALUES(...)
```

**In place of the ‘...’ fill in the corresponding info as it appears in the parentheses in the line above**

```
Ex. ('2023-01-01',160,310,70,4960,1960,35,1500,25)
```

*Sebastian: If a new position is created, execute the following commands to enter the new position into the database:*

```
INSERT INTO jobtitle(position_id,title, hourly_wage,tip_point_average)
VALUES(..)
```

**In place of the ‘...’ fill in the corresponding info as it appears in the parentheses in the line above**

```
Ex. (6,'somellier',6.50,0.5)
```

*Sebastian: If a new cocktail is added to the menu, execute the following commands to add the cocktail information to the database:*

```
INSERT INTO
cocktail(cocktail_name,spirit_name,ounces_used,spec,build_method,glassware,garnish)
VALUES(...)
```

**In place of the ‘...’ fill in the corresponding info as it appears in the parentheses in the line above**

```
Ex. ('margarita','Mijenta Blanco',1.5,'1.5oz tequila, 0.75oz lime, 0.5oz dry curacao,0.5oz simple','shaken','rocks','salt rim and lime wheel')
```

*Sebastian: If a liquor order is delivered, execute the following commands to add the order information to add the inventory to the database:*

```
INSERT INTO inventory_order(order_date,order_id,spirit_name,num_bottles,bill)
VALUES(...)
```

**In place of the ‘...’ fill in the corresponding info as it appears in the parentheses in the line above**

```
Ex. ('2023-02-01',43,'Ramazotti',6,150.00)
```

**Updates and Deletes**

*Sebastian: If an employee terminates their employment, execute the following commands:*

```
DELETE FROM jposition WHERE employee_id = (...)
Ex. DELETE FROM jposition WHERE employee_id=1864
DELETE FROM shift WHERE employee_id = (...)
Ex. DELETE FROM shift WHERE employee_id=1864
DELETE FROM staff WHERE employee_id = (…)
Ex. DELETE FROM staff WHERE employee_id=1864
```

*Sebastian: If we take a cocktail off of the menu, execute the following command:*

```
DELETE FROM cocktail WHERE cocktail_name = (‘...’)
Ex. DELETE FROM cocktail WHERE cocktail_name = 'margarita'
```

*Sebastian: If an employee switches their position, execute the following commands:*

```
UPDATE jposition SET employment = ([start date, end date]) WHERE employee_id = (‘...’)
Ex. UPDATE jposition SET employment = '[2023-02-01, 2023-02-06]' WHERE employee_id=8273
INSERT INTO jposition (position_id,employee_id,employment)
VALUES (...)
```

**In place of the ‘...’ fill in the corresponding position id as shown in the jobtitle table, employee id, and a date with start date as day of first shift and end date as infinity**

```
Ex. (6,8273,'[2023-02-06, infinity)')
```

**Important Queries**

*Sebastian: To see every employee that worked on a given service night, execute the following query:*

```
SELECT DISTINCT employee_id FROM shift WHERE shift && ‘[Start Date, End Date]’
Ex. SELECT DISTINCT employee_id FROM shift WHERE shift&& '[2023-01-01,2023-01-02]'
```

**This query will return the employee ids of all employees that worked on the date given**

*Sebastian: To see our total sales on a given service night, execute the following query:*

```
SELECT SUM(bar_sales + kitchen_sales) AS total_sales FROM service_night WHERE
date=’date’
Ex. SELECT SUM(bar_sales + kitchen_sales) AS total_sales FROM service_night WHERE night='2023-01-01'
```

**This query wil return the total sales on a given service night**

*Sebastian: To figure out how much cash tips to give each employee from a given night of service, execute the following query:*

```
SELECT s.employee_id, (SUM(n.cash_tips)/(COUNT(DISTINCT s.employee_id)) AS
cash_tips_per_employee FROM shift s
INNER JOIN service_night n ON s.shift::date = n.night WHERE s.shift::date = ‘date’
GROUP BY s.employee_id
```

```
Ex. SELECT s.employee_id, (SUM(n.cash_tips)/(COUNT(DISTINCT s.employee_id))* AS
cash_tips_per_employee FROM shift s
INNER JOIN service_night n ON s.shift::date = n.night WHERE s.shift::date = '2023-01-01'
GROUP BY s.employee_id
```