Simple SQL

CS 327E

Feb 5, 2018

Announcements

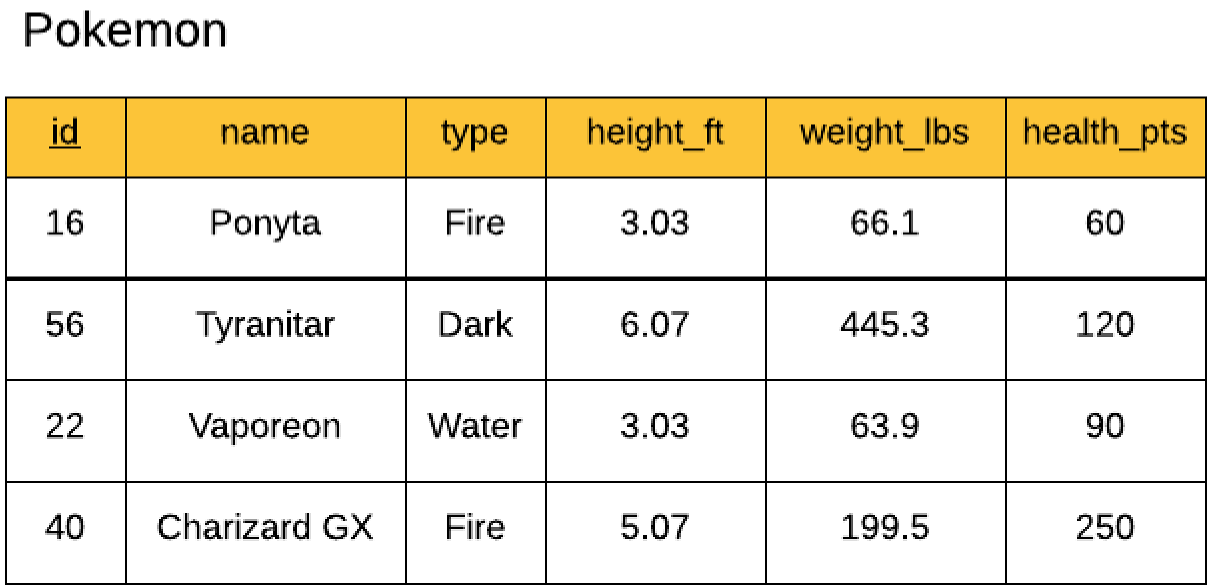
- New Google Cloud coupon for those who didn’t receive one for Lab 1.
- Quiz 1 has been graded. If you don’t see your grade on Canvas, it’s likely an issue with iClicker.

1) What does a SELECT statement do?

a) It retrieves records from the specified table(s)
b) It updates records in the specified table
c) It adds new records to the specified table
d) None of the above

2) How many fields does the following SELECT statement return?

```
SELECT * FROM Pokemon;
```



Pokemon

id	name	type	height_ft	weight_lbs	health_pts
16	Ponyta	Fire	3.03	66.1	60
56	Tyranitar	Dark	6.07	445.3	120
22	Vaporeon	Water	3.03	63.9	90
40	Charizard GX	Fire	5.07	199.5	250

a) 4

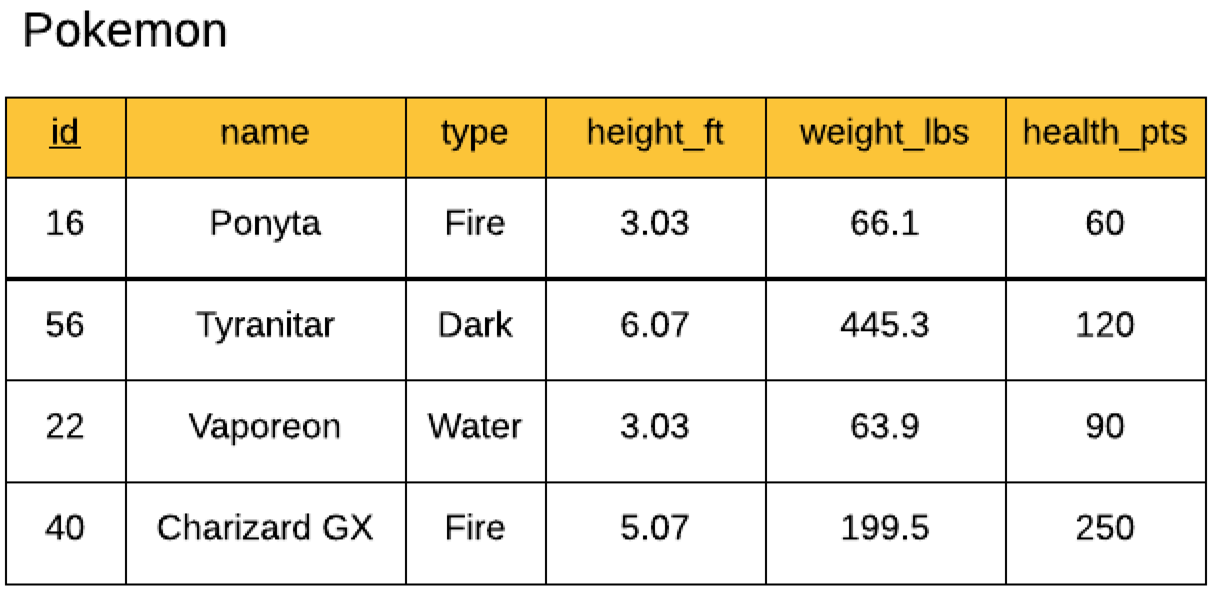
b) 5

c) 6

d) 0

3) How many records does the following SELECT statement return?

```
SELECT * FROM Pokemon
WHERE type = 'Fire' OR type = 'Water';
```



Pokemon

id	name	type	height_ft	weight_lbs	health_pts
16	Ponyta	Fire	3.03	66.1	60
56	Tyranitar	Dark	6.07	445.3	120
22	Vaporeon	Water	3.03	63.9	90
40	Charizard GX	Fire	5.07	199.5	250

a) 0

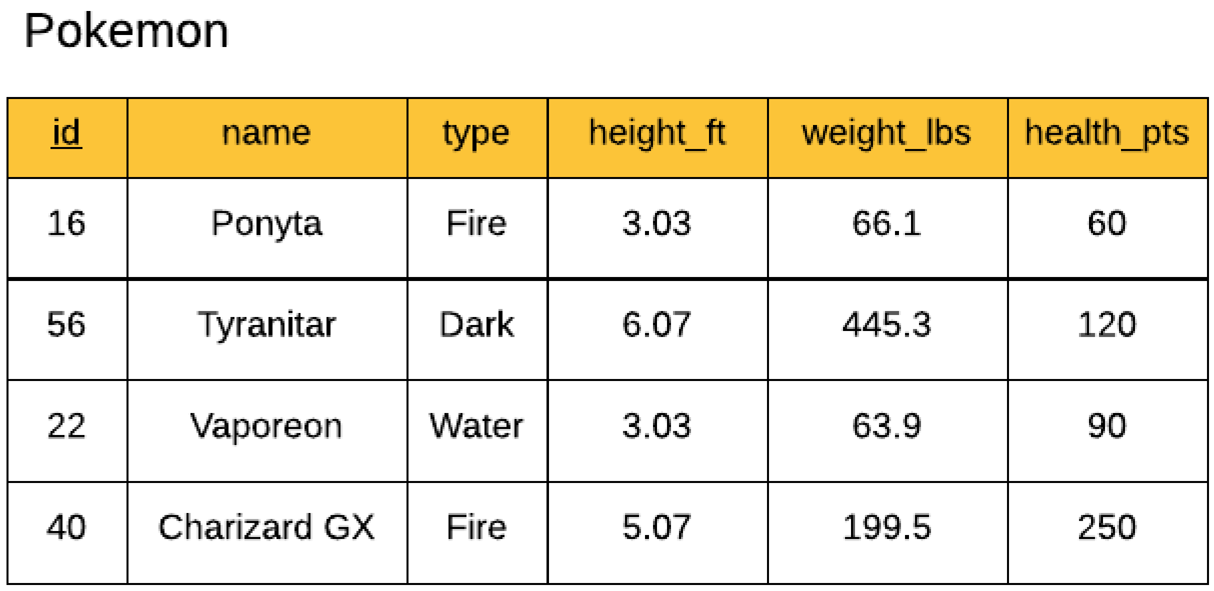
b) 1

c) 2

d) 3

4) How many records does the following SELECT statement return?

```
SELECT * FROM Pokemon
WHERE height_ft = 3.03 AND health_pts >= 100;
```



Pokemon

id	name	type	height_ft	weight_lbs	health_pts
16	Ponyta	Fire	3.03	66.1	60
56	Tyranitar	Dark	6.07	445.3	120
22	Vaporeon	Water	3.03	63.9	90
40	Charizard GX	Fire	5.07	199.5	250

a) 0
b) 1
c) 2
d) 3

5) What does the following CREATE TABLE statement do?

```
CREATE TABLE Pokemon2
  AS SELECT * FROM Pokemon;
```

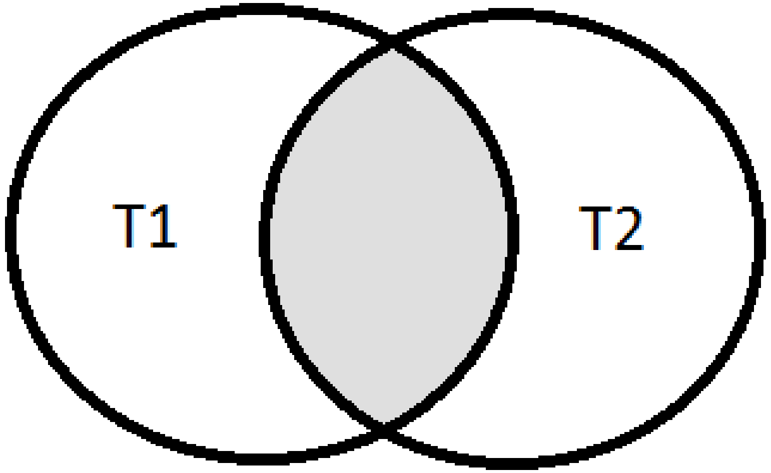
a) It copies the structure and data from `Pokemon2`, naming the copied table `Pokemon`.

b) It copies the structure and data from `Pokemon`, naming the copied table `Pokemon2`.

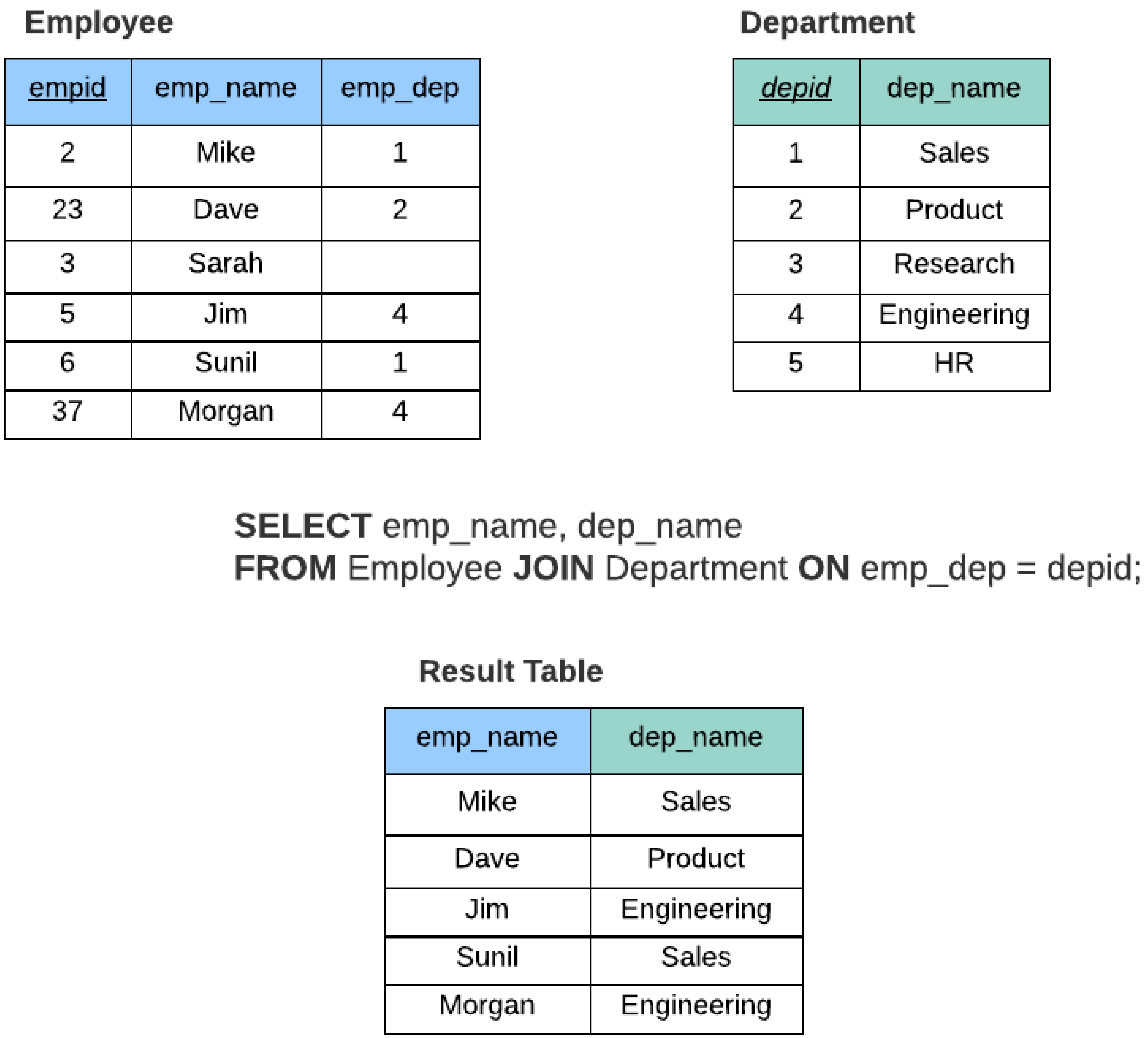
c) None of the above.

Inner Join

```
SELECT *
FROM T1 [INNER] JOIN T2
ON T1.c1 = T2.c2;
```



Inner Join



Employee

empid	emp_name	emp_dep
2	Mike	1
23	Dave	2
3	Sarah	
5	Jim	4
6	Sunil	1
37	Morgan	4

Department

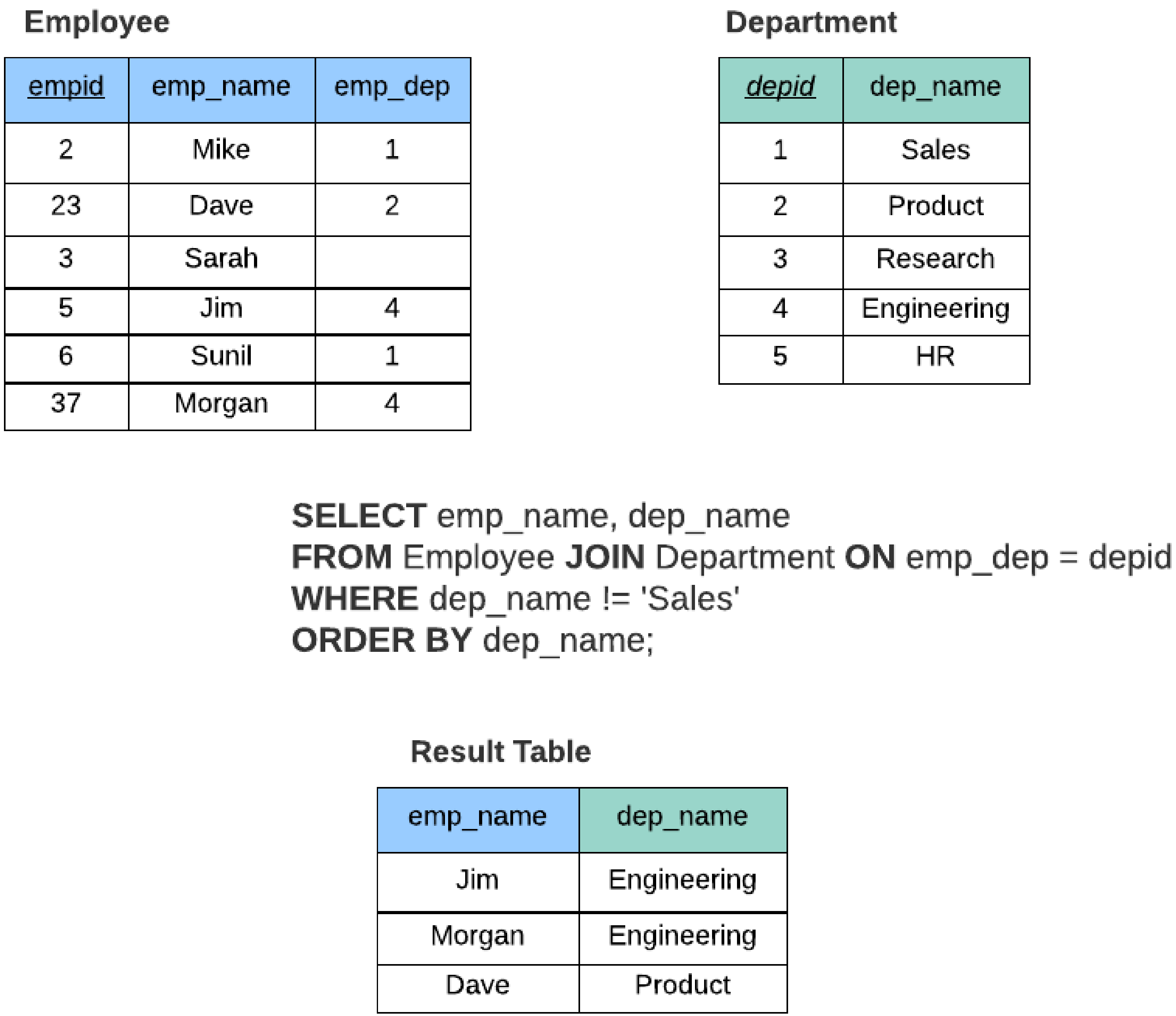
depid	dep_name
1	Sales
2	Product
3	Research
4	Engineering
5	HR

```
SELECT emp_name, dep_name
FROM Employee JOIN Department ON emp_dep = depid;
```

Result Table

emp_name	dep_name
Mike	Sales
Dave	Product
Jim	Engineering
Sunil	Sales
Morgan	Engineering

Inner Join



Employee

<u>empid</u>	emp_name	emp_dep
2	Mike	1
23	Dave	2
3	Sarah	
5	Jim	4
6	Sunil	1
37	Morgan	4

Department

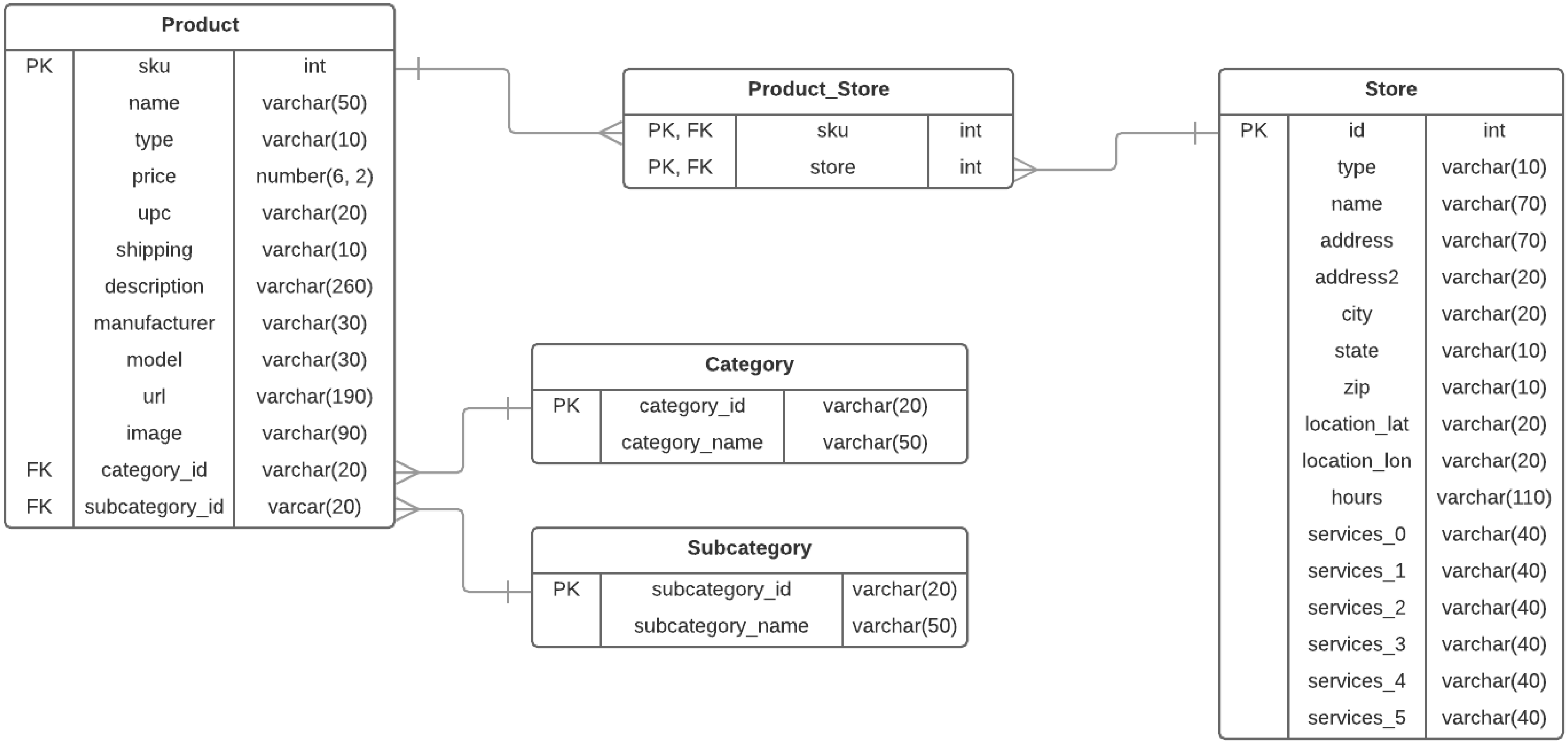
<u>depid</u>	dep_name
1	Sales
2	Product
3	Research
4	Engineering
5	HR

```
SELECT emp_name, dep_name
FROM Employee JOIN Department ON emp_dep = depid
WHERE dep_name != 'Sales'
ORDER BY dep_name;
```

Result Table

emp_name	dep_name
Jim	Engineering
Morgan	Engineering
Dave	Product

Best Buy ERD



Product

PK	sku	int
	name	varchar(50)
	type	varchar(10)
	price	number(6, 2)
	upc	varchar(20)
	shipping	varchar(10)
	description	varchar(260)
	manufacturer	varchar(30)
	model	varchar(30)
	url	varchar(190)
	image	varchar(90)
FK	category_id	varchar(20)
FK	subcategory_id	varcar(20)

Product_Store

PK, FK	sku	int
PK, FK	store	int

Store

PK	id	int
	type	varchar(10)
	name	varchar(70)
	address	varchar(70)
	address2	varchar(20)
	city	varchar(20)
	state	varchar(10)
	zip	varchar(10)
	location_lat	varchar(20)
	location_lon	varchar(20)
	hours	varchar(110)
	services_0	varchar(40)
	services_1	varchar(40)
	services_2	varchar(40)
	services_3	varchar(40)
	services_4	varchar(40)
	services_5	varchar(40)

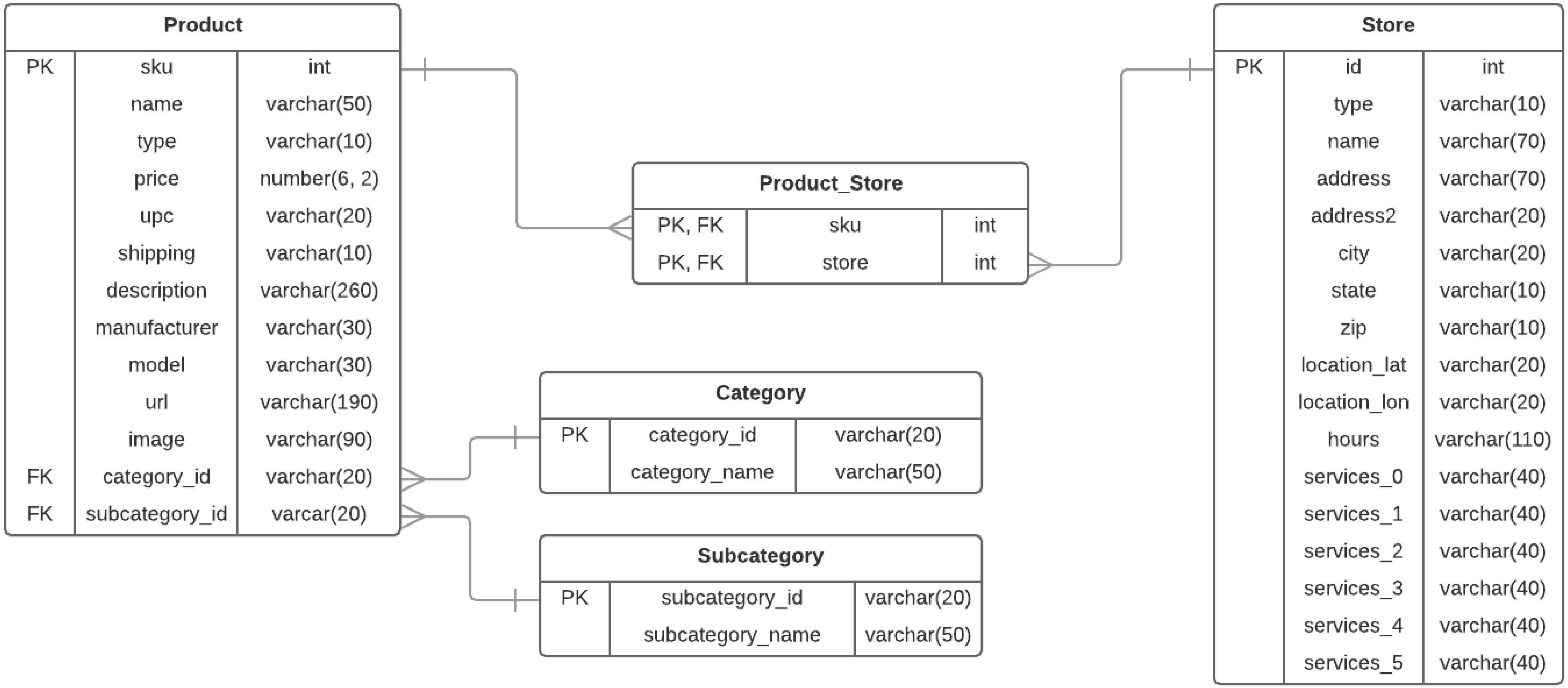
Category

PK	category_id	varchar(20)
	category_name	varchar(50)

Subcategory

PK	subcategory_id	varchar(20)
	subcategory_name	varchar(50)

Practice Problem 1: Find all the Best Buy stores in Austin that carry Apple products. Use subcategory_name = 'Apple'. Return the product name and price, store address and zip. Order the results by product name.



Product

PK	sku	int
	name	varchar(50)
	type	varchar(10)
	price	number(6, 2)
	upc	varchar(20)
	shipping	varchar(10)
	description	varchar(260)
	manufacturer	varchar(30)
	model	varchar(30)
	url	varchar(190)
	image	varchar(90)
FK	category_id	varchar(20)
FK	subcategory_id	varcar(20)

Product_Store

PK, FK	sku	int
PK, FK	store	int

Category

PK	category_id	varchar(20)
	category_name	varchar(50)

Subcategory

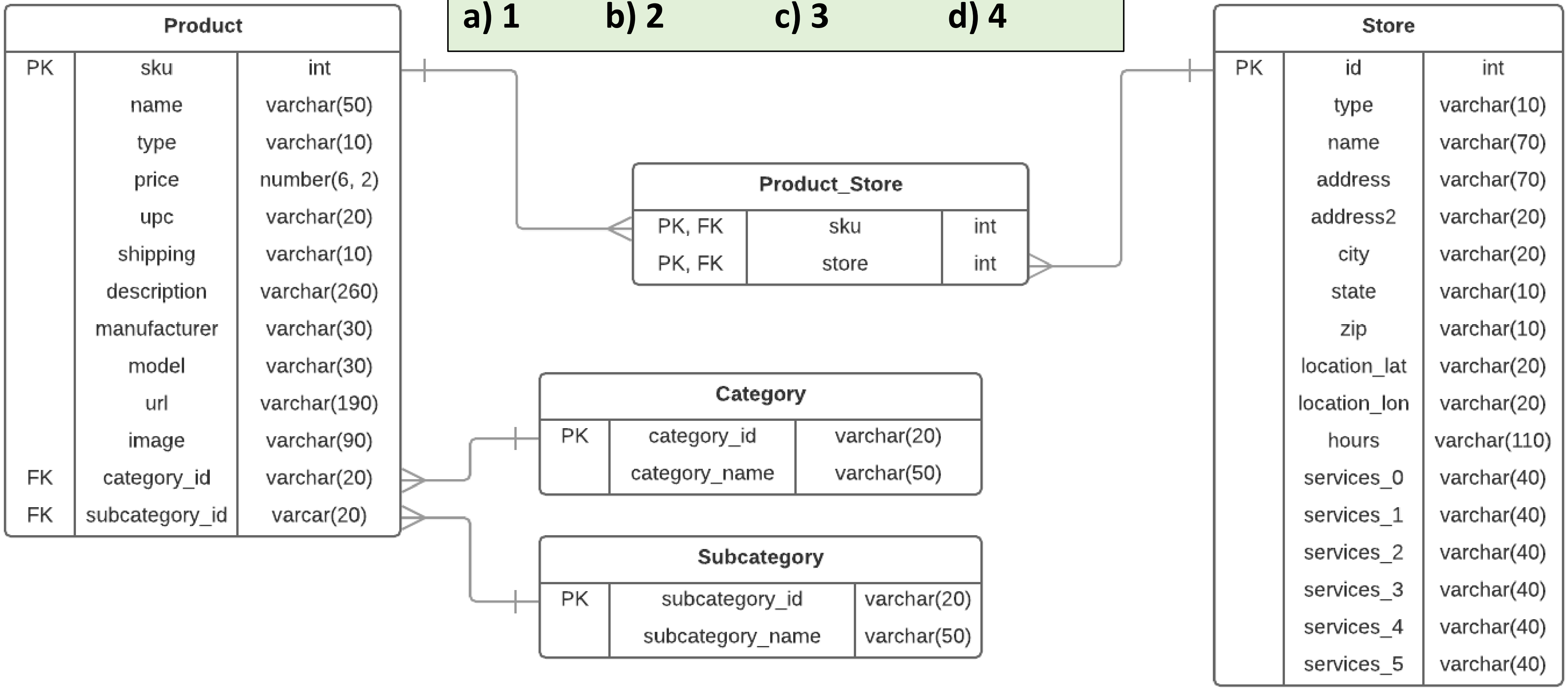
PK	subcategory_id	varchar(20)
	subcategory_name	varchar(50)

Store

PK	id	int
	type	varchar(10)
	name	varchar(70)
	address	varchar(70)
	address2	varchar(20)
	city	varchar(20)
	state	varchar(10)
	zip	varchar(10)
	location_lat	varchar(20)
	location_lon	varchar(20)
	hours	varchar(110)
	services_0	varchar(40)
	services_1	varchar(40)
	services_2	varchar(40)
	services_3	varchar(40)
	services_4	varchar(40)
	services_5	varchar(40)

Practice Problem 1: Find all the Best Buy stores in Austin that carry Apple products. Use subcategory_name = 'Apple'. Return the product name and price, store address and zip. Order the results by product name.

How many joins does this query require?
a) 1 b) 2 c) 3 d) 4



Product		
PK	sku	int
	name	varchar(50)
	type	varchar(10)
	price	number(6, 2)
	upc	varchar(20)
	shipping	varchar(10)
	description	varchar(260)
	manufacturer	varchar(30)
	model	varchar(30)
	url	varchar(190)
	image	varchar(90)
FK	category_id	varchar(20)
FK	subcategory_id	varcar(20)

Product_Store		
PK, FK	sku	int
PK, FK	store	int

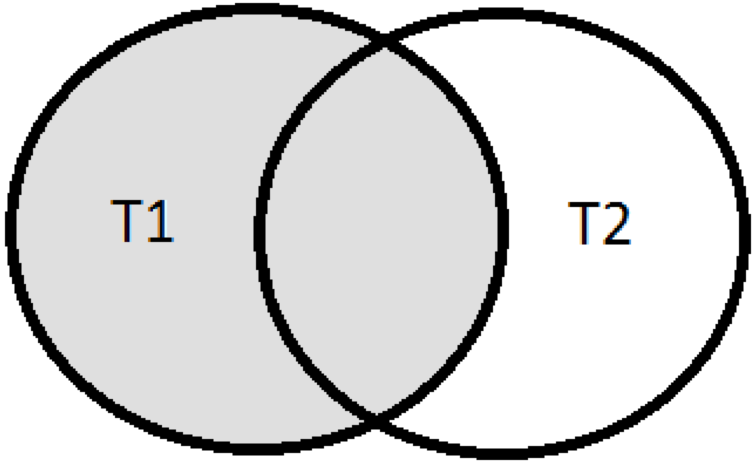
Category		
PK	category_id	varchar(20)
	category_name	varchar(50)

Subcategory		
PK	subcategory_id	varchar(20)
	subcategory_name	varchar(50)

Store		
PK	id	int
	type	varchar(10)
	name	varchar(70)
	address	varchar(70)
	address2	varchar(20)
	city	varchar(20)
	state	varchar(10)
	zip	varchar(10)
	location_lat	varchar(20)
	location_lon	varchar(20)
	hours	varchar(110)
	services_0	varchar(40)
	services_1	varchar(40)
	services_2	varchar(40)
	services_3	varchar(40)
	services_4	varchar(40)
	services_5	varchar(40)

Left Outer Join

```
SELECT *
FROM T1 LEFT [OUTER] JOIN T2
ON T1.c1 = T2.c2;
```



Left Outer Join

Employee

empid	emp_name	emp_dep
2	Mike	1
23	Dave	2
3	Sarah	
5	Jim	4
6	Sunil	1
37	Morgan	4

Department

depid	dep_name
1	Sales
2	Product
3	Research
4	Engineering
5	HR

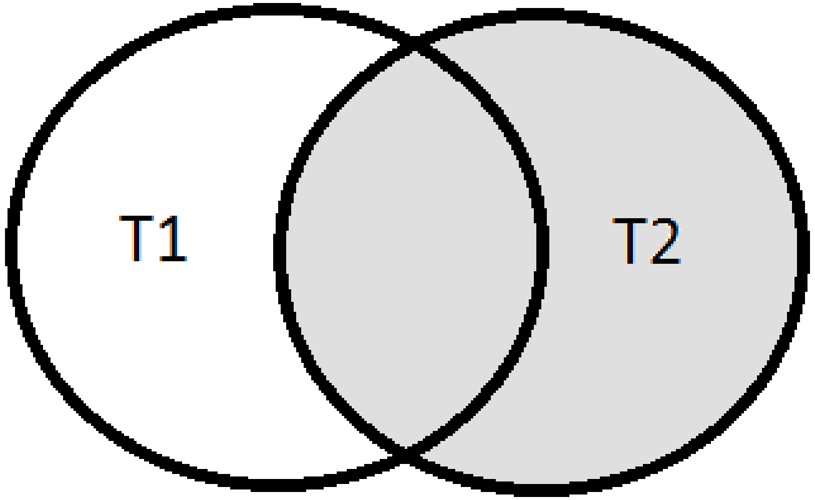
```
SELECT emp_name, dep_name
FROM Employee LEFT JOIN Department ON emp_dep = depid
ORDER BY emp_name;
```

Result Table

emp_name	dep_name
Dave	Product
Jim	Engineering
Mike	Sales
Morgan	Engineering
Sarah	
Sunil	Sales

Right Outer Join

```
SELECT *
FROM T1 RIGHT [OUTER] JOIN T2
ON T1.c1 = T2.c2;
```



Right Outer Join

Employee

empid	emp_name	emp_dep
2	Mike	1
23	Dave	2
3	Sarah	
5	Jim	4
6	Sunil	1
37	Morgan	4

Department

depid	dep_name
1	Sales
2	Product
3	Research
4	Engineering
5	HR

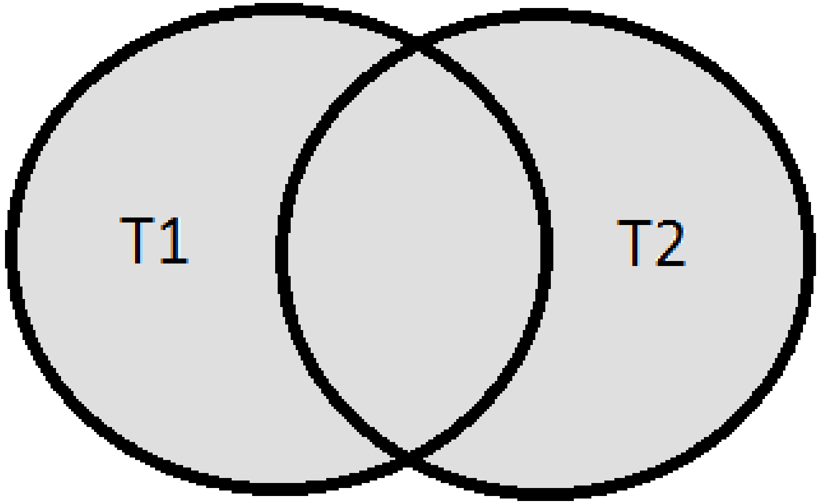
```
SELECT emp_name, dep_name
FROM Employee RIGHT JOIN Department ON emp_dep = depid
ORDER BY dep_name, emp_name;
```

Result Table

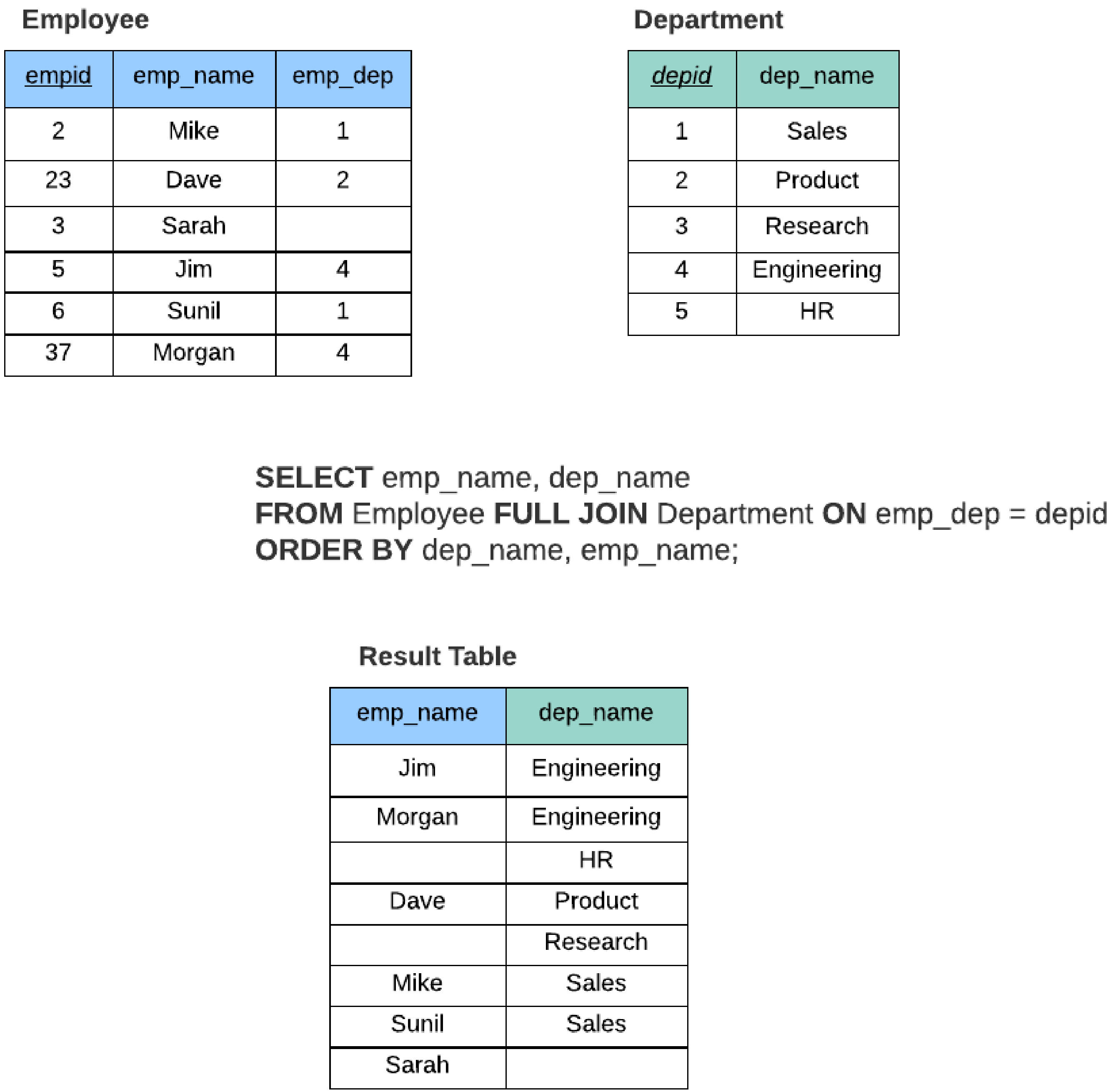
emp_name	dep_name
Jim	Engineering
Morgan	Engineering
	HR
Dave	Product
	Research
Mike	Sales
Sunil	Sales

Full Outer Join

```
SELECT *
FROM T1 FULL [OUTER] JOIN T2
ON T1.c1 = T2.c2;
```



Full Outer Join



Employee

empid	emp_name	emp_dep
2	Mike	1
23	Dave	2
3	Sarah	
5	Jim	4
6	Sunil	1
37	Morgan	4

Department

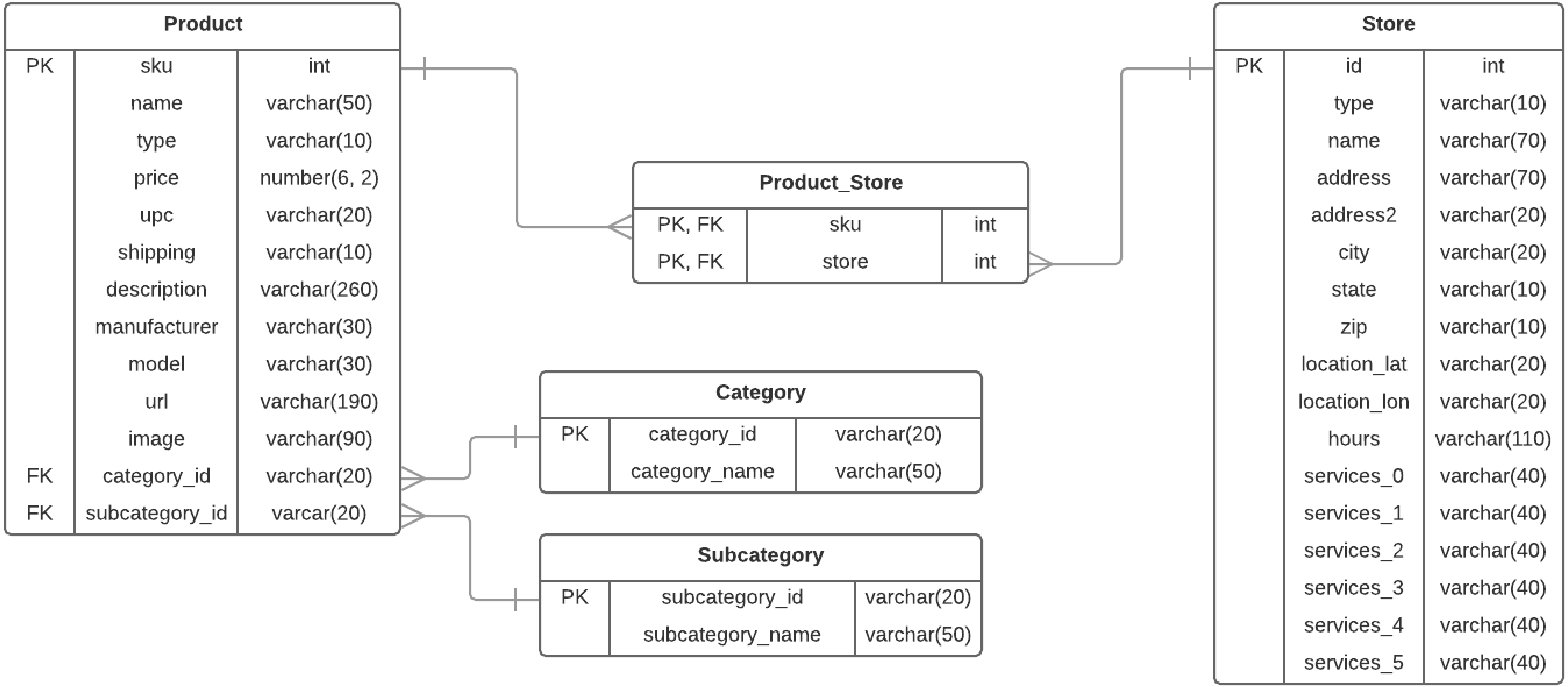
depid	dep_name
1	Sales
2	Product
3	Research
4	Engineering
5	HR

```
SELECT emp_name, dep_name
FROM Employee FULL JOIN Department ON emp_dep = depid
ORDER BY dep_name, emp_name;
```

Result Table

emp_name	dep_name
Jim	Engineering
Morgan	Engineering
	HR
Dave	Product
	Research
Mike	Sales
Sunil	Sales
Sarah	

Practice Problem 2: Find all the products that are **not** sold at any store. Return the product sku, name, and price. Order results by product sku.



Product

PK	sku	int
	name	varchar(50)
	type	varchar(10)
	price	number(6, 2)
	upc	varchar(20)
	shipping	varchar(10)
	description	varchar(260)
	manufacturer	varchar(30)
	model	varchar(30)
	url	varchar(190)
	image	varchar(90)
FK	category_id	varchar(20)
FK	subcategory_id	varcar(20)

Product_Store

PK, FK	sku	int
PK, FK	store	int

Category

PK	category_id	varchar(20)
	category_name	varchar(50)

Subcategory

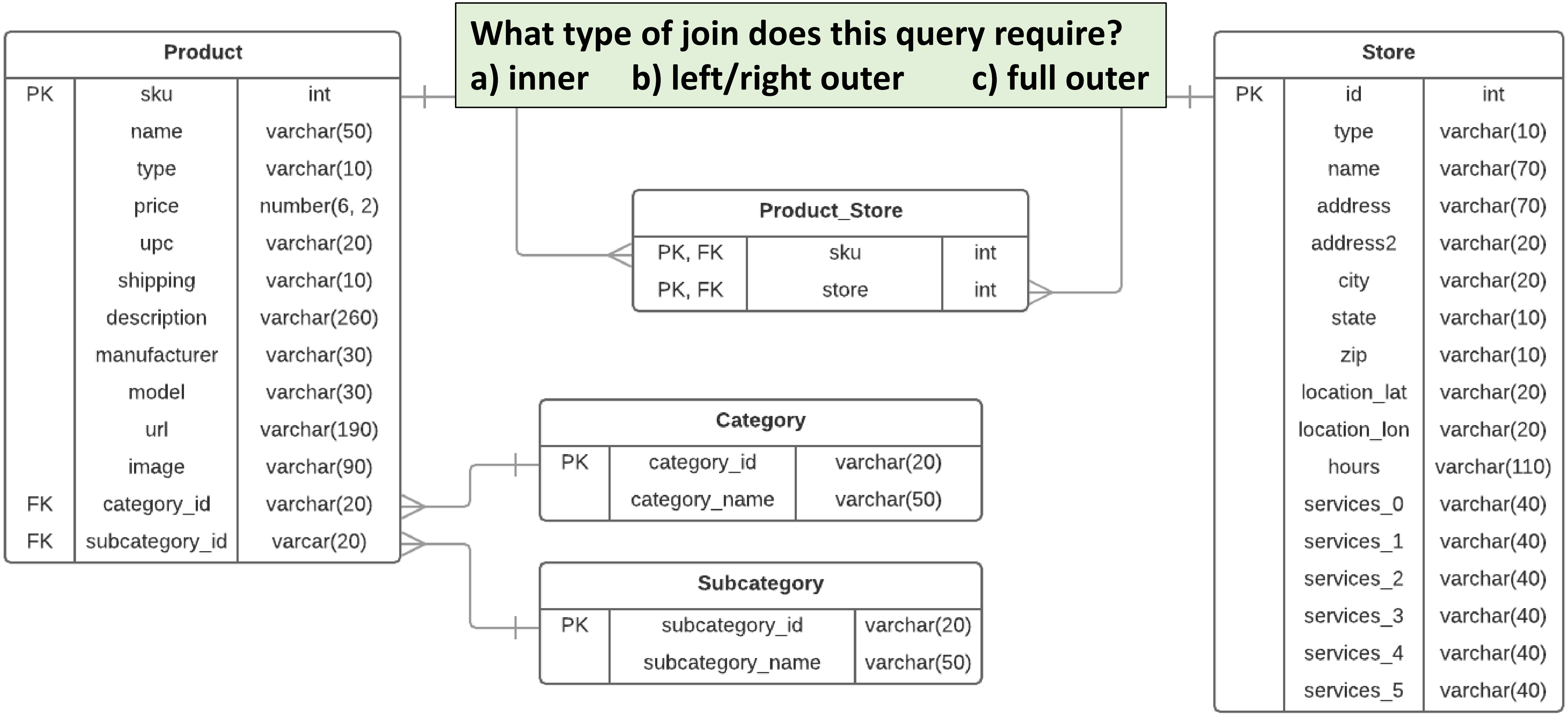
PK	subcategory_id	varchar(20)
	subcategory_name	varchar(50)

Store

PK	id	int
	type	varchar(10)
	name	varchar(70)
	address	varchar(70)
	address2	varchar(20)
	city	varchar(20)
	state	varchar(10)
	zip	varchar(10)
	location_lat	varchar(20)
	location_lon	varchar(20)
	hours	varchar(110)
	services_0	varchar(40)
	services_1	varchar(40)
	services_2	varchar(40)
	services_3	varchar(40)
	services_4	varchar(40)
	services_5	varchar(40)

Practice Problem 2: Find all the products that are **not** sold at any store. Return the product sku, name, and price. Order results by product sku.

What type of join does this query require?
a) inner b) left/right outer c) full outer



Product

PK	sku	int
	name	varchar(50)
	type	varchar(10)
	price	number(6, 2)
	upc	varchar(20)
	shipping	varchar(10)
	description	varchar(260)
	manufacturer	varchar(30)
	model	varchar(30)
	url	varchar(190)
	image	varchar(90)
FK	category_id	varchar(20)
FK	subcategory_id	varcar(20)

Product_Store

PK, FK	sku	int
PK, FK	store	int

Category

PK	category_id	varchar(20)
	category_name	varchar(50)

Subcategory

PK	subcategory_id	varchar(20)
	subcategory_name	varchar(50)

Store

PK	id	int
	type	varchar(10)
	name	varchar(70)
	address	varchar(70)
	address2	varchar(20)
	city	varchar(20)
	state	varchar(10)
	zip	varchar(10)
	location_lat	varchar(20)
	location_lon	varchar(20)
	hours	varchar(110)
	services_0	varchar(40)
	services_1	varchar(40)
	services_2	varchar(40)
	services_3	varchar(40)
	services_4	varchar(40)
	services_5	varchar(40)