INLS623

CHADD:

Chapel Hill

Alcoholic Drink Database

Group 4

Gibson Farabow, Felicity Walston, Chenchen Yang

Content

01 / Description & Gathering

02 / Model Diagram

03 / Normalization

04 / SQL

05 / Queries

06 / Conclusion

01

Description & Gathering

Description

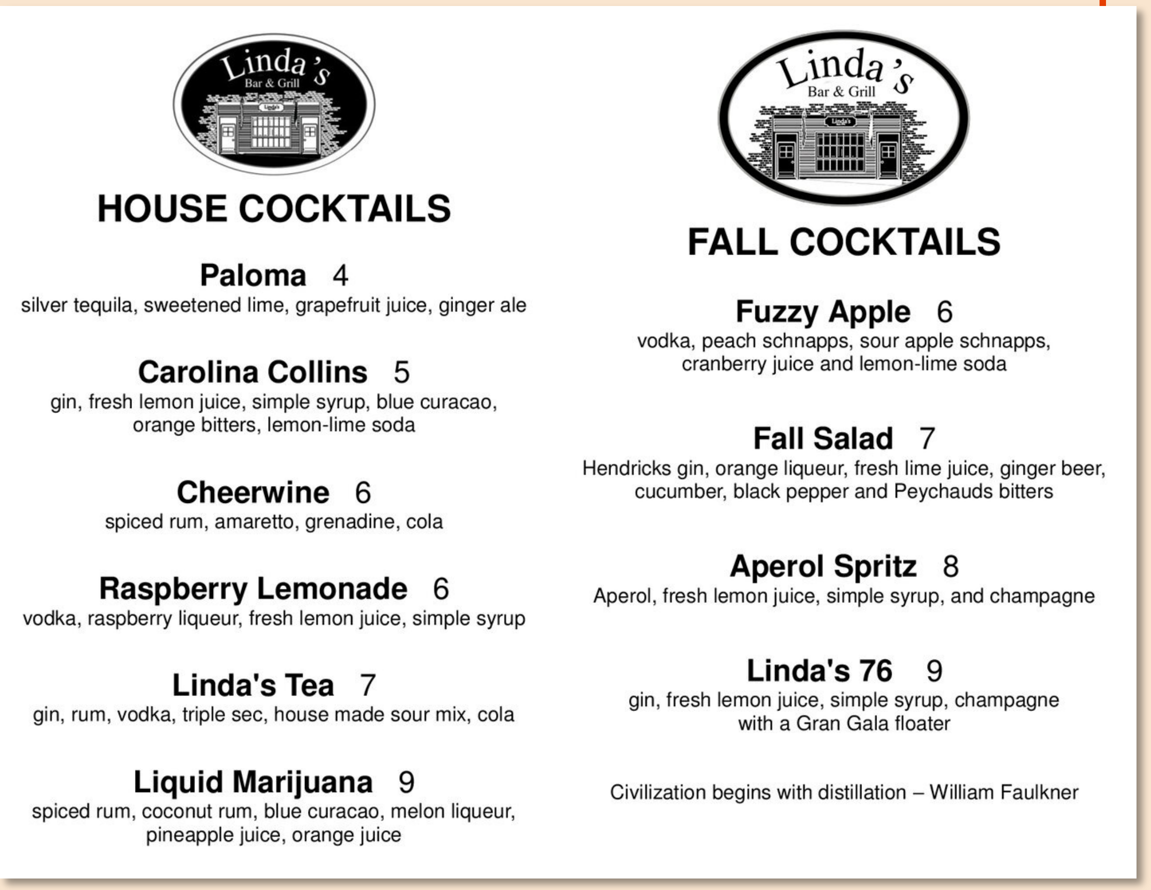
Many people have allergies or other dietary restrictions.

Chapel Hill and Carrboro have lots of restaurants and bars.

Restaurants and stores have a huge range of drink options.

People may want access to menu items in more depth.

Gathering



HOUSE COCKTAILS

Paloma 4
silver tequila, sweetened lime, grapefruit juice, ginger ale

Carolina Collins 5
gin, fresh lemon juice, simple syrup, blue curacao, orange bitters, lemon-lime soda

Cheerwine 6
spiced rum, amaretto, grenadine, cola

Raspberry Lemonade 6
vodka, raspberry liqueur, fresh lemon juice, simple syrup

Linda's Tea 7
gin, rum, vodka, triple sec, house made sour mix, cola

Liquid Marijuana 9
spiced rum, coconut rum, blue curacao, melon liqueur, pineapple juice, orange juice

FALL COCKTAILS

Fuzzy Apple 6
vodka, peach schnapps, sour apple schnapps, cranberry juice and lemon-lime soda

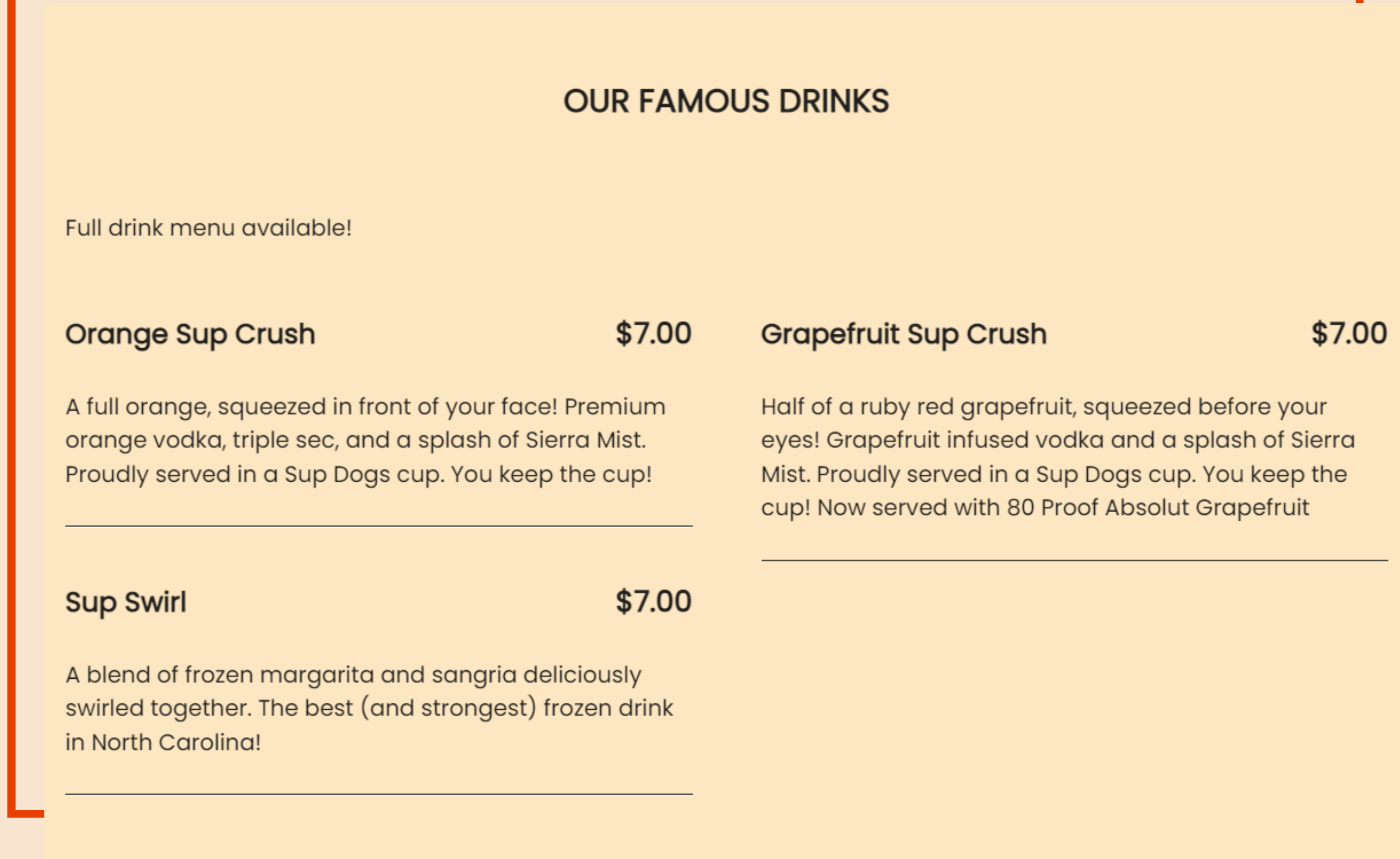
Fall Salad 7
Hendricks gin, orange liqueur, fresh lime juice, ginger beer, cucumber, black pepper and Peychauds bitters

Aperol Spritz 8
Aperol, fresh lemon juice, simple syrup, and champagne

Linda's 76 9
gin, fresh lemon juice, simple syrup, champagne with a Gran Gala floater

Civilization begins with distillation – William Faulkner

Linda's Bar & Gril



OUR FAMOUS DRINKS

Full drink menu available!

Orange Sup Crush $7.00

A full orange, squeezed in front of your face! Premium orange vodka, triple sec, and a splash of Sierra Mist. Proudly served in a Sup Dogs cup. You keep the cup!

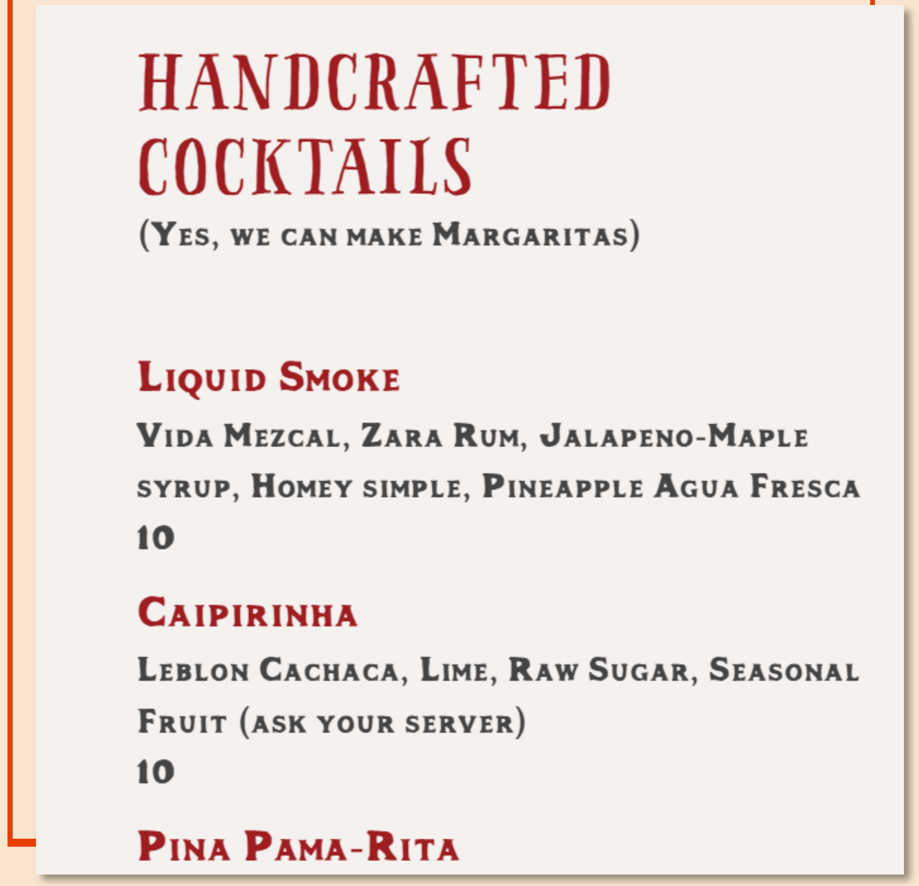
Sup Swirl $7.00

A blend of frozen margarita and sangria deliciously swirled together. The best (and strongest) frozen drink in North Carolina!

Grapefruit Sup Crush $7.00

Half of a ruby red grapefruit, squeezed before your eyes! Grapefruit infused vodka and a splash of Sierra Mist. Proudly served in a Sup Dogs cup. You keep the cup! Now served with 80 Proof Absolut Grapefruit

Sup Dogs



HANDCRAFTED COCKTAILS

(Yes, we can make Margaritas)

Liquid Smoke
Vida Mezcal, Zara Rum, Jalapeno-Maple syrup, Homey simple, Pineapple Agua Fresca
10

Caipirinha
Leblon Cachaca, Lime, Raw Sugar, Seasonal Fruit (ask your server)
10

Pina Pama-Rita

Luna Rotisserie

Sample data was collected using the online menus of restaurants and bars located in Chapel Hill and Carrboro.

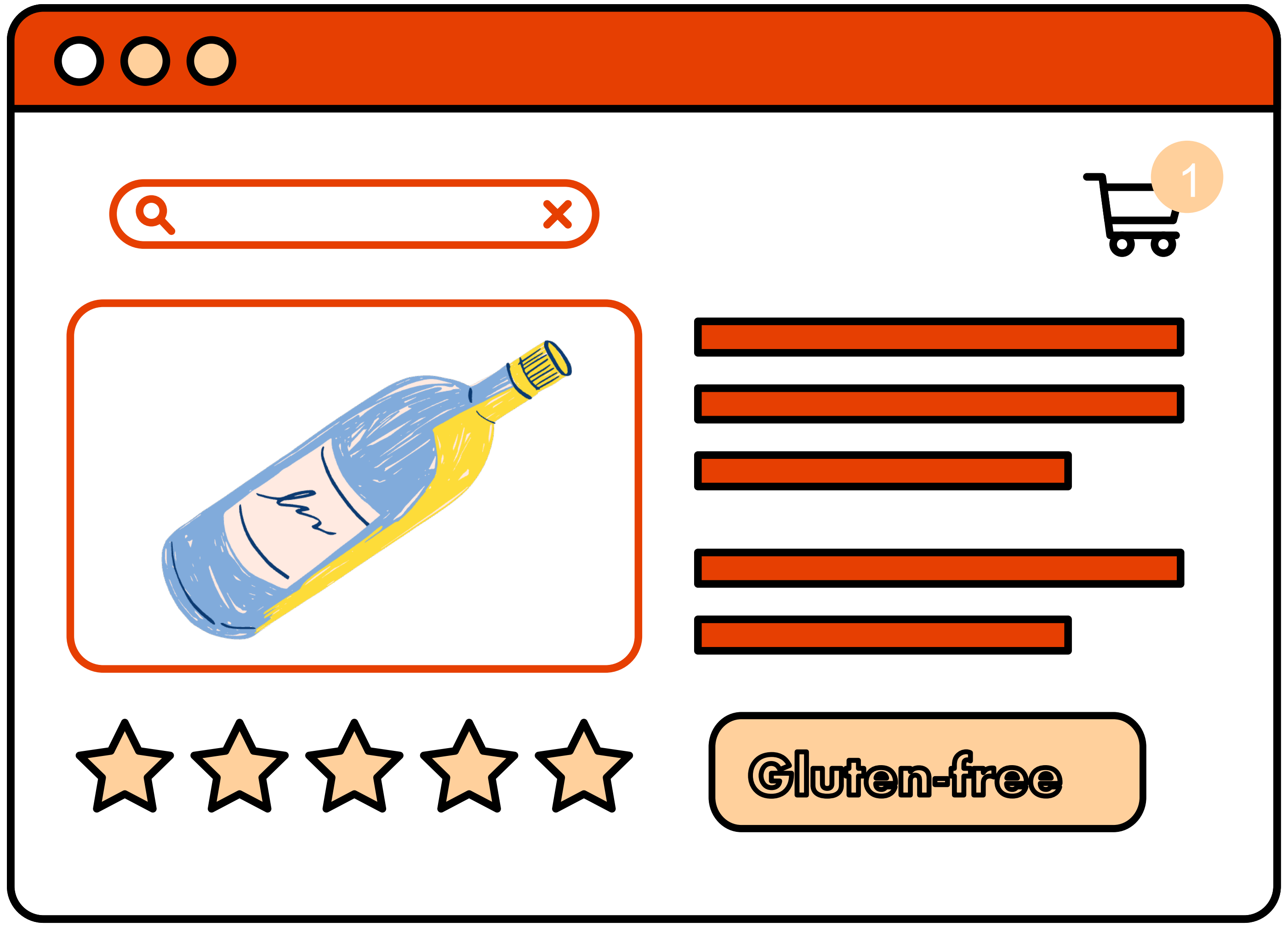
02

Model Diagram

User Stories

- *Callie has Diabetes and wants to see which low-carb drinks in Chapel Hill are the cheapest.*
- *John has a heart condition and takes medication that interacts with grapefruit. He wants to know which drinks to avoid when he goes out with his friends.*
- *Rachel wants to see what mixed drinks in Carrboro have an especially high alcohol content (higher than 20%) to celebrate the end of classes.*
- *Grace is going to the grocery store and wants to know the alcohol content of common drinks beforehand to help her decide what to get.*
- *Margaret is an avid drinker and wants to peruse what ingredients different drinks have to learn about mixing drinks.*

Relationship Sentences

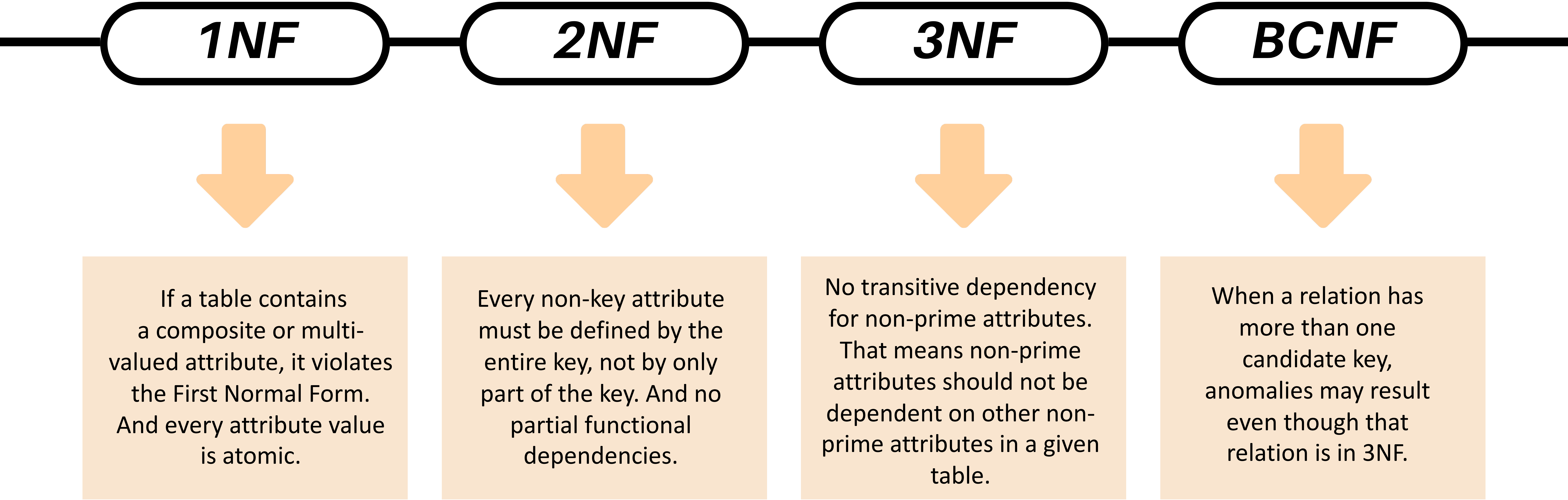


- One **Drink** *may not* accommodate one or more **Dietary_Restriction**s.
- One **Common_Drink** *may not* accommodate one or more **Dietary_Restriction**s.
- One **Dietary_Restriction** *may* apply to one or more **Drink**s.
- One **Dietary_Restriction** *may* apply to one or more **Common_Drink**s.
- One **Drink** *must* consist of one or more **Drink_Attribute**s.
- One **Common_Drink** *must* consist of one or more **Drink_Attribute**s.
- One **Drink_Attribute** *may* apply to one to one or more **Common_Drink**s.
- One **Drink_Attribute** *may* apply to one to one or more **Drink**s.
- One **Drink** *must* be sold at one **Establishment**.
- One **Establishment** *must* sell one or more **Drink**s.

03

Normalization

Normalization



1NF

If a table contains a composite or multi-valued attribute, it violates the First Normal Form. And every attribute value is atomic.

2NF

Every non-key attribute must be defined by the entire key, not by only part of the key. And no partial functional dependencies.

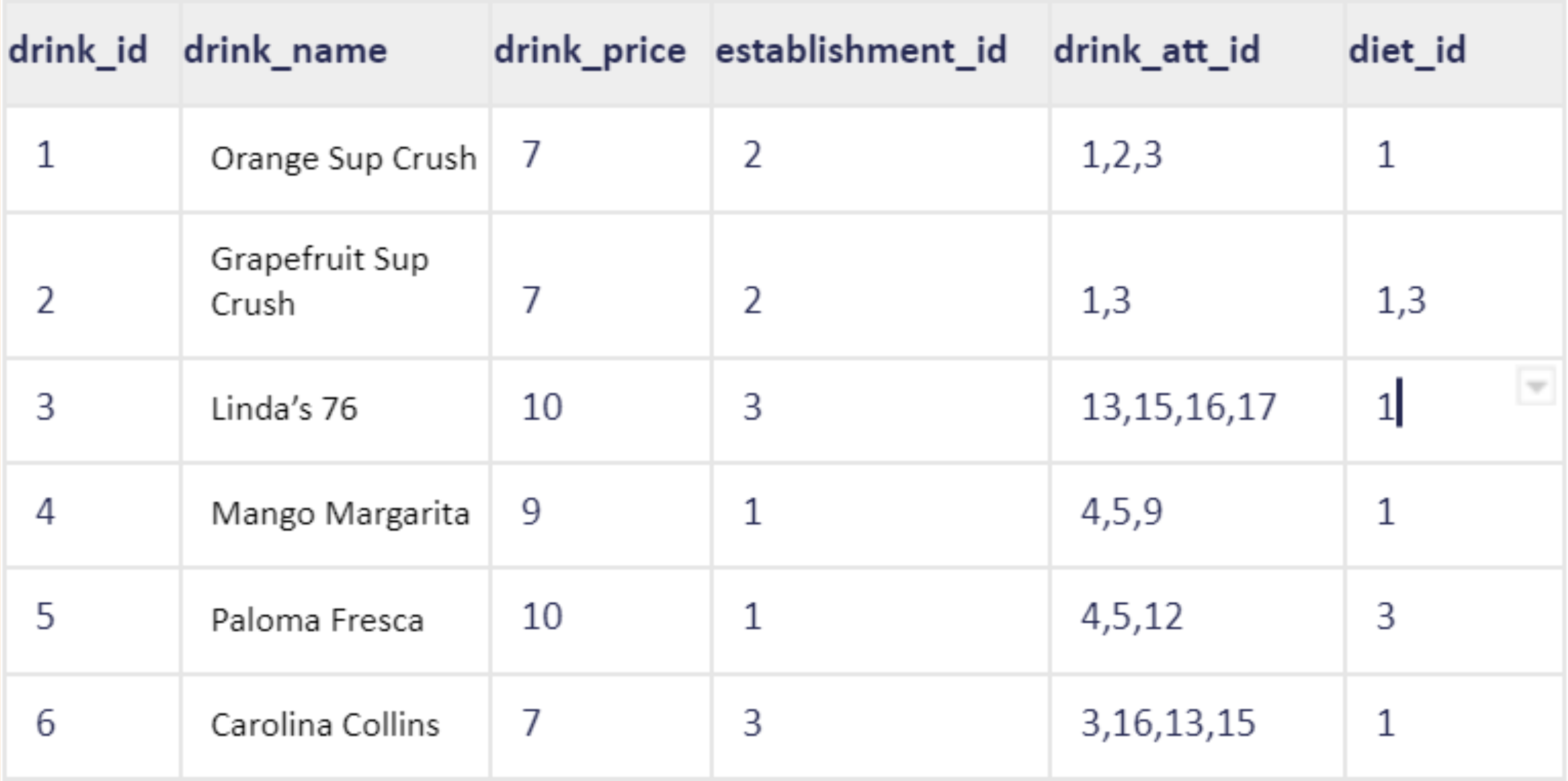
3NF

No transitive dependency for non-prime attributes. That means non-prime attributes should not be dependent on other non-prime attributes in a given table.

BCNF

When a relation has more than one candidate key, anomalies may result even though that relation is in 3NF.

Drink [drink_id (key) , drink_name, drink_price, establishment_id(fk1),drink_att_id(fk2), diet_id(fk3)]



drink_id	drink_name	drink_price	establishment_id	drink_att_id	diet_id
1	Orange Sup Crush	7	2	1,2,3	1
2	Grapefruit Sup Crush	7	2	1,3	1,3
3	Linda's 76	10	3	13,15,16,17	1
4	Mango Margarita	9	1	4,5,9	1
5	Paloma Fresca	10	1	4,5,12	3
6	Carolina Collins	7	3	3,16,13,15	1

Sample Data

FD1: drink_id -> drink-name, drink_price, establishment_id, drink_att_id, diet_id

FD2: drink_att_id -> diet_id

1NF: drink_att_id, diet_id are multi-valued attributes.

Solution: Split out drink_att_id and diet_id into a separate relation.

2NF: No partial Key dependencies

3NF: Transitive dependency exists: drink_id -> drink_att_id -> diet_id

Solution: Split Drink relation into three new relations named Drink, Drink_has_Dietary_Restriction, and Drink_has_Attribute:

Drink [drink_id (key), drink_name, drink_price, establishment_id(fk)]

Key: drink_id

FD1: drink_id -> drink_name, drink_price, establishment_id, drink_att_id

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

Drink_has_Diet [drink_att_id (key), diet_id (key)]

Key: drink_att_id

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

Drink_has_Att [drink_att_id (key), drink_id(key)]

Key: drink_att_id, drink_id

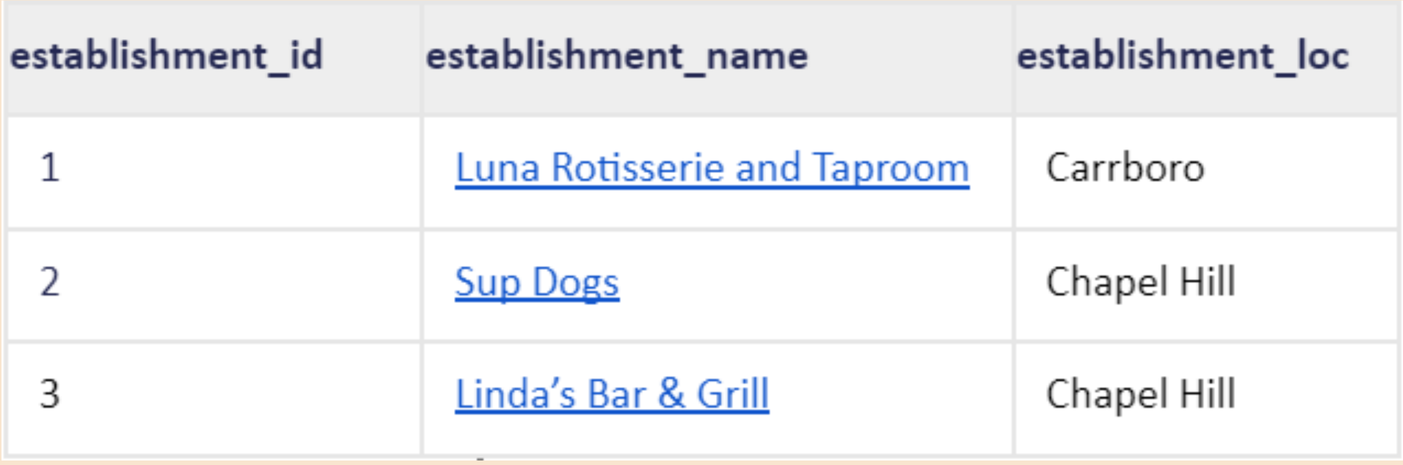
1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

Establishment[establishment_id (key), establishment_name, establishment_loc]



establishment_id	establishment_name	establishment_loc
1	Luna Rotisserie and Taproom	Carrboro
2	Sup Dogs	Chapel Hill
3	Linda’s Bar & Grill	Chapel Hill

Sample Data

FD: establishment_id -> establishment_name, establishment_loc

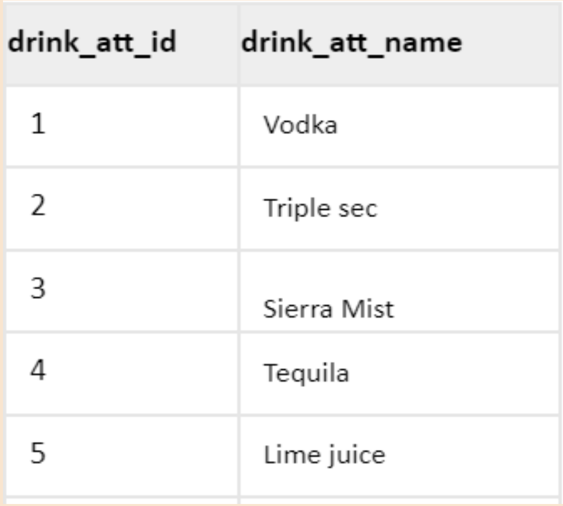
1NF: establishment_loc may be treated as a composite attribute.

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

Drink_Attribute[drink_att_id (key), drink_att_name]



drink_att_id	drink_att_name
1	Vodka
2	Triple sec
3	Sierra Mist
4	Tequila
5	Lime juice

Sample Data

FD: drink_att_id -> drink_att_name

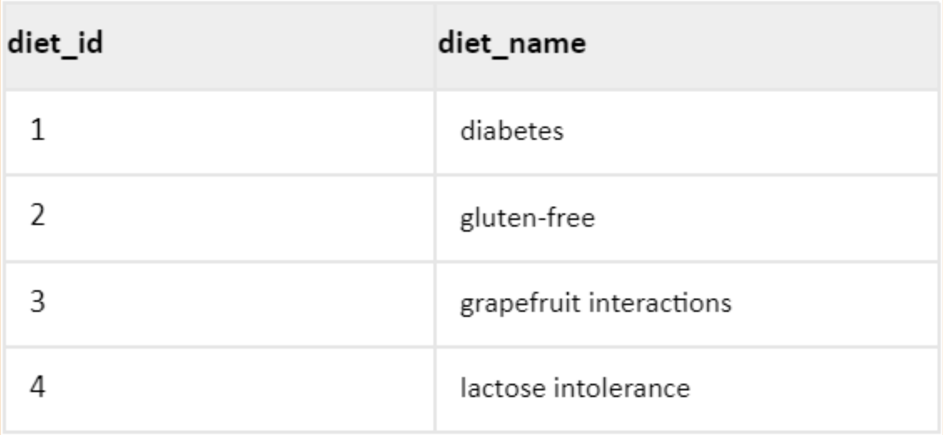
1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

Dietary_Restriction [diet_id(key), diet_name]



diet_id	diet_name
1	diabetes
2	gluten-free
3	grapefruit interactions
4	lactose intolerance

Sample Data

FD: diet_id -> diet_name

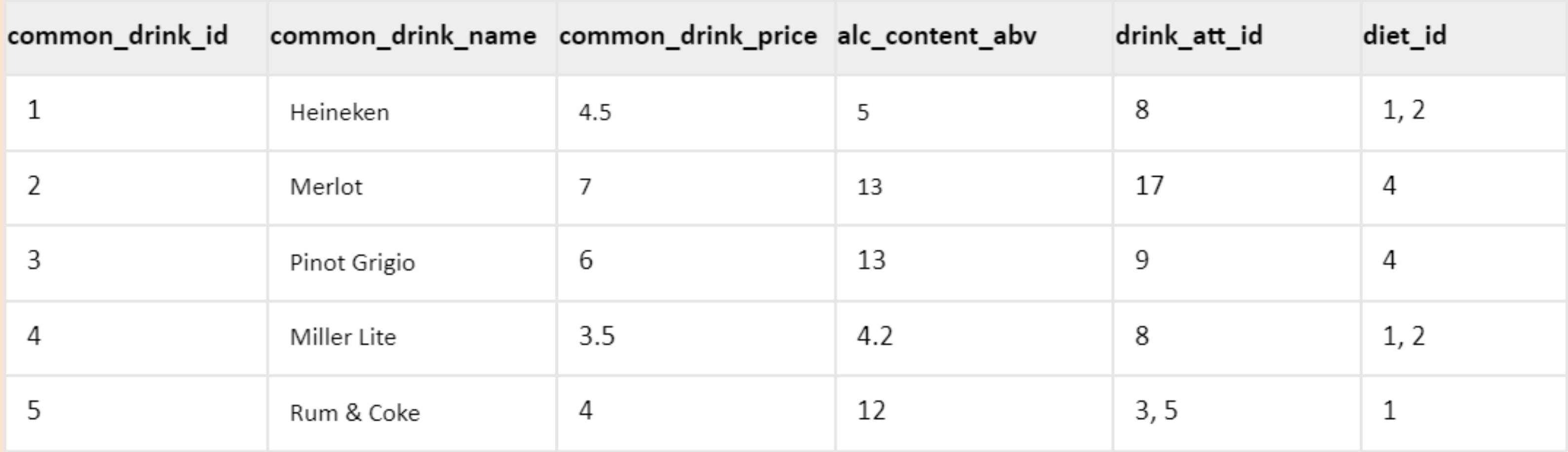
1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

Common_Drink [common_drink_id(key), common_drink_name, common_drink_price, drink_att_id(fk1), diet_id(fk2)]



common_drink_id	common_drink_name	common_drink_price	alc_content_abv	drink_att_id	diet_id
1	Heineken	4.5	5	8	1, 2
2	Merlot	7	13	17	4
3	Pinot Grigio	6	13	9	4
4	Miller Lite	3.5	4.2	8	1, 2
5	Rum & Coke	4	12	3, 5	1

Sample Data

FD1: common_drink_id -> common_drink_name, common_drink_price, drink_att_id, diet_id

FD2: drink_att_id -> diet_id

1NF: drink_att_id, diet_id are multi-valued attributes.

Solution: Split out drink_att_id and diet_id into a separate relation.

2NF: No partial Key dependencies

3NF: Transitive dependency exists: common_drink_id -> drink_att_id -> diet_id

Solution: Split Drink relation into two new relations named Common_Drink and Common_Drink_has_Dietary_Restriction:

Common_Drink [common_drink_id (key), common_drink_name, common_drink_price]

Key: common_drink_id

FD1: common_drink_id -> common_drink_name, common_drink_price, drink_att_id

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

Common_Drink_has_Att [common_drink_id (key), drink_att_id (key)]

Key: common_drink_id, drink_att_id

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate key

Common_Drink_has_Diet[drink_att_id (key), diet_id (key)]

Key: drink_att_id, diet_id

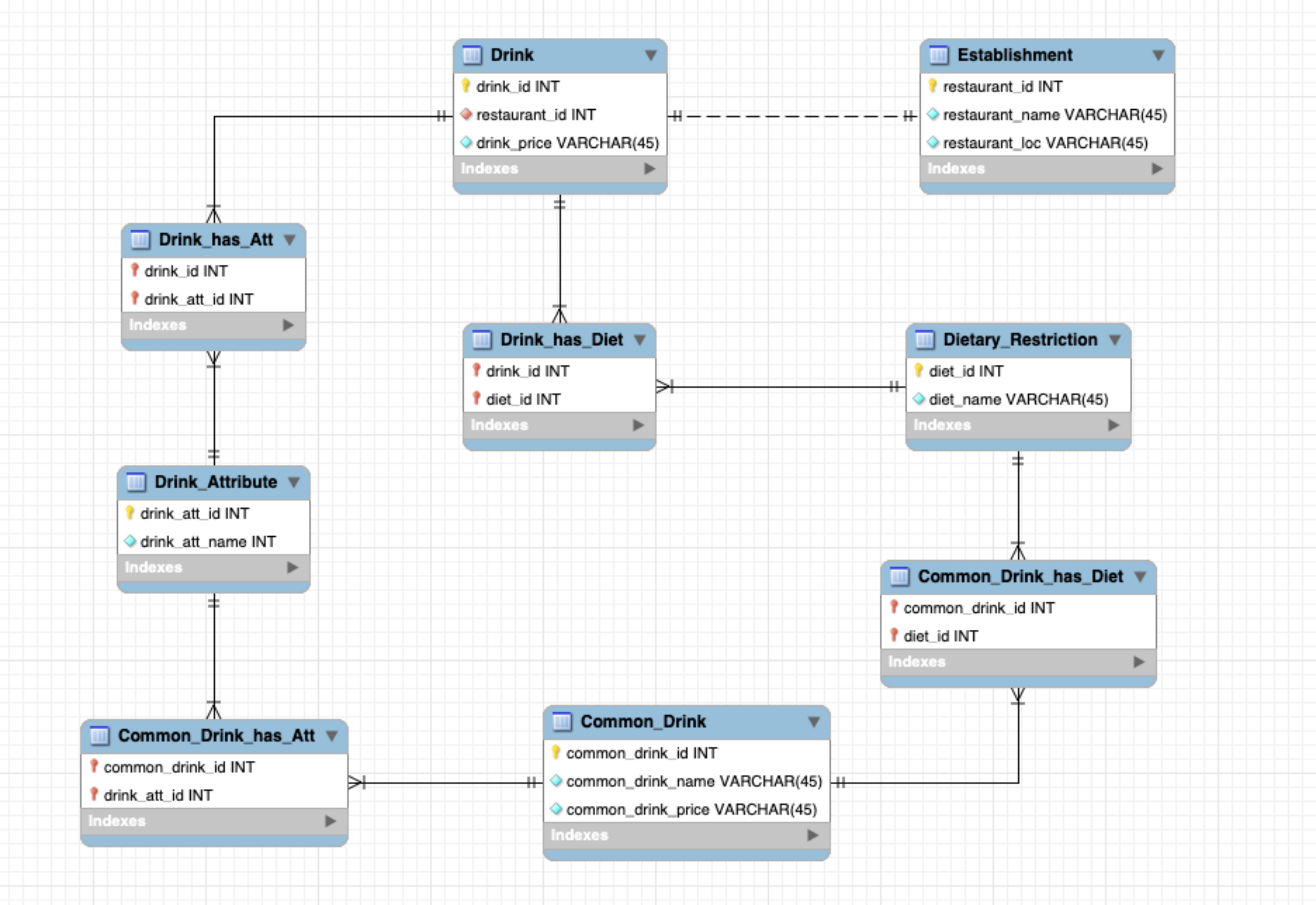
1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate key

Final Relational Model



04

SQL

SQL

```sql
CREATE TABLE IF NOT EXISTS Establishment (
  establishment_id INT(6) NOT NULL,
  establishment_name VARCHAR(45) NOT NULL,
  establishment_loc VARCHAR(45) NOT NULL,
  PRIMARY KEY (establishment_id));

CREATE TABLE IF NOT EXISTS Drink (
  drink_id INT NOT NULL,
  drink_name VARCHAR(45) NOT NULL,
  establishment_id INT(6) NOT NULL,
  drink_price VARCHAR(45) NOT NULL,
  PRIMARY KEY (drink_id),
  FOREIGN KEY (establishment_id) References Establishment(establishment_id));

CREATE TABLE IF NOT EXISTS Common_Drink (
  common_drink_id INT(6) NOT NULL,
  common_drink_name VARCHAR(45) NOT NULL,
  common_drink_price VARCHAR(45) NOT NULL,
  alc_content_abv INT(2) NULL,
  PRIMARY KEY (common_drink_id));

CREATE TABLE IF NOT EXISTS Dietary_Restriction (
  diet_id INT(6) NOT NULL,
  diet_name VARCHAR(45) NOT NULL,
  PRIMARY KEY (diet_id));
```

```sql
CREATE TABLE IF NOT EXISTS Common_Drink_has_Diet(
  common_drink_id INT(6) NOT NULL,
  diet_id INT(6) NOT NULL,
  PRIMARY KEY (common_drink_id, diet_id),
  FOREIGN KEY (common_drink_id) References Common_Drink(common_drink_id),
  FOREIGN KEY (diet_id) References Dietary_Restriction(diet_id));

 CREATE TABLE IF NOT EXISTS Drink_has_Diet (
  drink_id INT(6) NOT NULL,
  diet_id int(6) NOT NULL,
  PRIMARY KEY (drink_id, diet_id),
  FOREIGN KEY (drink_id) References Drink(drink_id),
  FOREIGN KEY (diet_id) References Dietary_Restriction(diet_id));

CREATE TABLE IF NOT EXISTS Drink_Attribute (
  drink_att_id INT(6) NOT NULL,
  drink_att VARCHAR(45) NOT NULL,
  PRIMARY KEY (drink_att_id));

CREATE TABLE IF NOT EXISTS Drink_has_Att (
  drink_id INT(6) NOT NULL,
  drink_att_id INT(6) NOT NULL,
  PRIMARY KEY (drink_att_id, drink_id),
  FOREIGN KEY (drink_id) References Drink(drink_id),
  FOREIGN KEY (drink_att_id) References Drink_Attribute(drink_att_id));

CREATE TABLE IF NOT EXISTS Common_Drink_has_Att (
  common_drink_id INT(6) NOT NULL,
  drink_att_id INT(6) NOT NULL,
  PRIMARY KEY (common_drink_id, drink_att_id),
  FOREIGN KEY (drink_att_id) References Drink_Attribute(drink_att_id),
  FOREIGN KEY (common_drink_id) References Common_Drink(common_drink_id));
```

DDL

```
INSERT INTO Establishment VALUES
(1, 'Luna Rotisserie and Taproom', 'Carrboro'),
(2, 'Sup Dogs', 'Chapel Hill'),
(3, 'Linda''s Bar & Grill', 'Chapel Hill’);

INSERT INTO Common_Drink VALUES
(1, 'Heineken', 4.5, 5),
(2, 'Merlot', 17, 13),
(3, 'Pinot Grigio', 6, 13),
(4, 'Miller Lite', 3.5, 4.2),
(5, 'Rum & Coke', 6, 12);

INSERT INTO Drink VALUES
(1, 'Orange Sup Crush', 2, 7),
(2, 'Grapefruit Sup Crush', 2, 7),
(3, 'Linda''s 76', 3, 10),
(4, 'Mango Margarita', 1, 9),
(5, 'Paloma Fresca', 1, 10),
(6, 'Carolina Collins', 3, 7);

INSERT INTO Drink_Attribute VALUES
(1, 'Vodka'),
(2, 'Triple sec'),
(3, 'Sierra Mist'),
(4, 'Tequila'),
(5, 'Lime juice'),
(6, 'Margarita mix'),
(7, 'Ginger beer'),
(8, 'Wheat'),
(9, 'White wine'),
(10, 'Cola'),
(11, 'Rum'),
(12, 'Grapefruit'),
(13, 'Gin'),
(14, 'Sour mix'),
(15, 'Lemon juice'),
(16, 'Simple syrup'),
(17, 'Red wine');
```

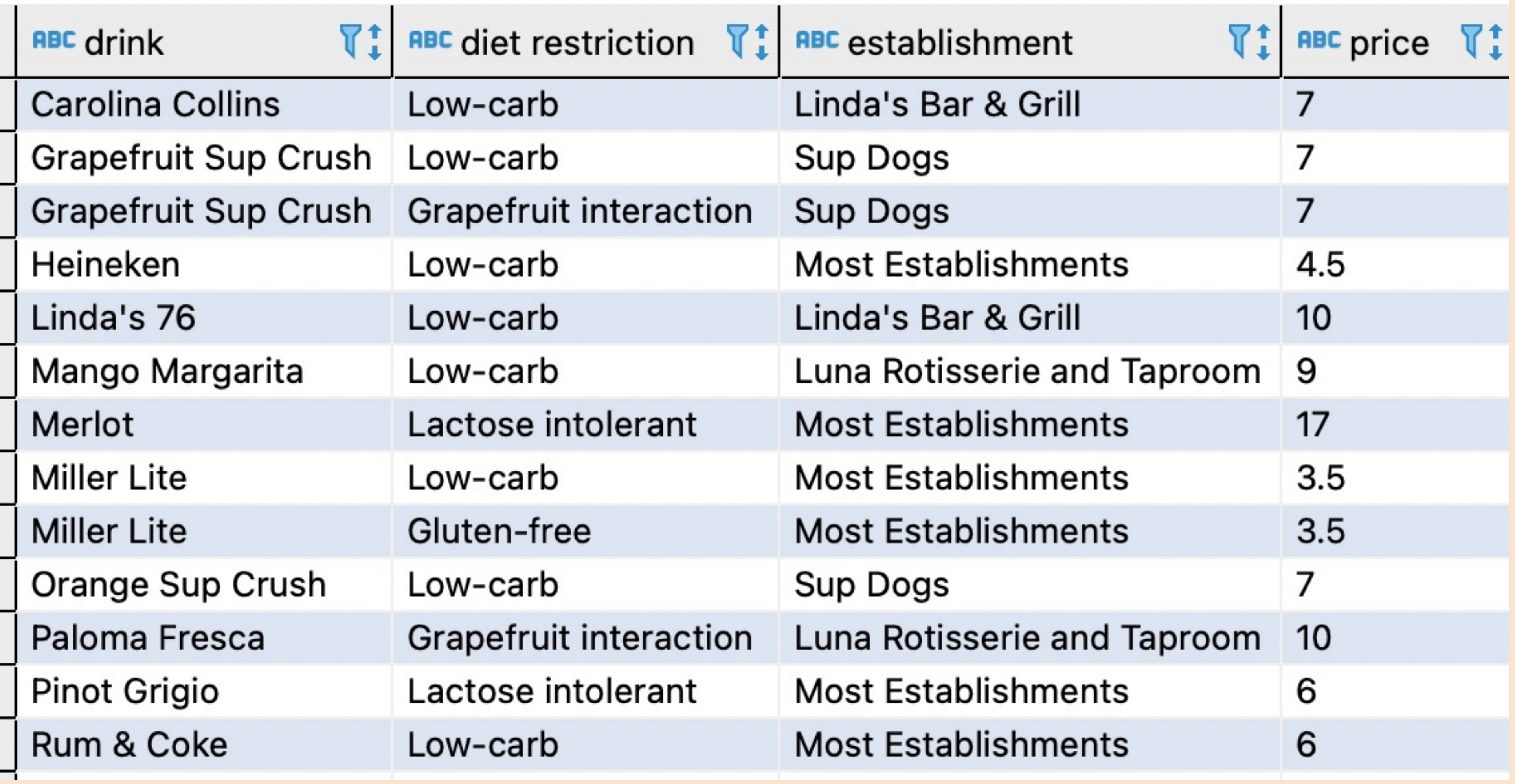
```
INSERT INTO Dietary_Restriction VALUES
(1, ‘Lo'),
(2, 'Gluten-free'),
(3, 'Grapefruit interaction'),
(4, 'Lactose intolerant');

INSERT INTO Drink_has_Att VALUES
(1,1), (1,2), (1,3),
(2,1), (2,3),
(3,13), (3,15), (3,16), (3,17),
(4,4), (4,5),
(5,4), (5,5), (5,12),
(6,3), (6,13), (6,15), (6,16);

INSERT INTO Common_Drink_has_Att VALUES
(1,8),
(2,17),
(3,9),
(4,8),
(5,10), (5,11);

INSERT INTO Drink_has_Diet VALUES
(1,1),
(2,1), (2,3),
(3,1),
(4,1),
(5,3),
(6,1);

INSERT INTO Common_Drink_has_Diet VALUES
(1,1),
(2,4),
(3,4),
(4,1),(4,2),
(5,1);
```

05

Queries

Report 1

Use case: A couple with multiple dietary restrictions is moving to Chapel Hill and wants to learn about drink options in town

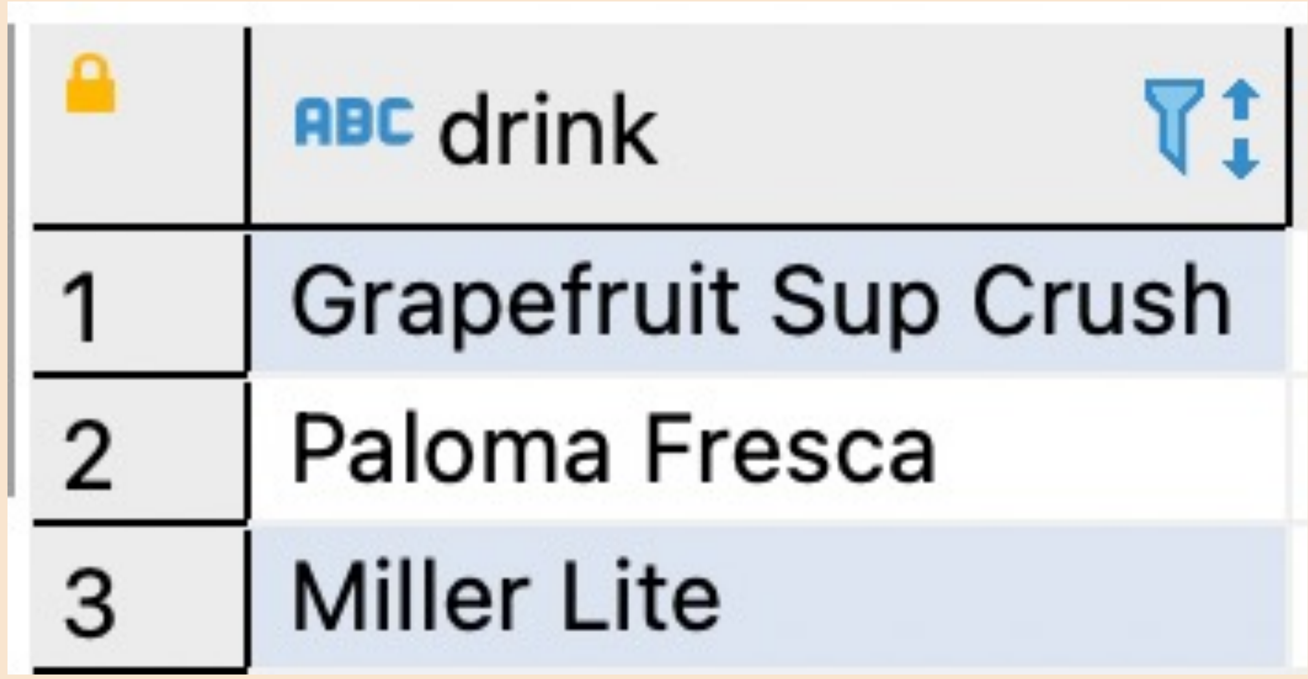


drink	diet restriction	establishment	price
Carolina Collins	Low-carb	Linda's Bar & Grill	7
Grapefruit Sup Crush	Low-carb	Sup Dogs	7
Grapefruit Sup Crush	Grapefruit interaction	Sup Dogs	7
Heineken	Low-carb	Most Establishments	4.5
Linda's 76	Low-carb	Linda's Bar & Grill	10
Mango Margarita	Low-carb	Luna Rotisserie and Taproom	9
Merlot	Lactose intolerant	Most Establishments	17
Miller Lite	Low-carb	Most Establishments	3.5
Miller Lite	Gluten-free	Most Establishments	3.5
Orange Sup Crush	Low-carb	Sup Dogs	7
Paloma Fresca	Grapefruit interaction	Luna Rotisserie and Taproom	10
Pinot Grigio	Lactose intolerant	Most Establishments	6
Rum & Coke	Low-carb	Most Establishments	6

```
SELECT drink_name drink, diet.diet_name ‘diet restriction’, establishment_name establishment, drink_price price
FROM Dietary_Restriction diet join Drink_has_Diet dd on dd.diet_id = diet.diet_id join Drink d on d.drink_id = dd.drink_id join Establishment e on d.establishment_id = e.establishment_id
UNION
SELECT common_drink_name drink, diet.diet_name ‘diet restriction’, ‘Most Establishments', common_drink_price price
FROM Dietary_Restriction diet join Common_Drink_has_Diet cdd on cdd.diet_id = diet.diet_id join Common_Drink cd on cd.common_drink_id = cdd.common_drink_id
ORDER BY drink
```

Report 1

Use case 2: A member of the family has grapefruit and gluten-free dietary restrictions. What drink(s) should they avoid?

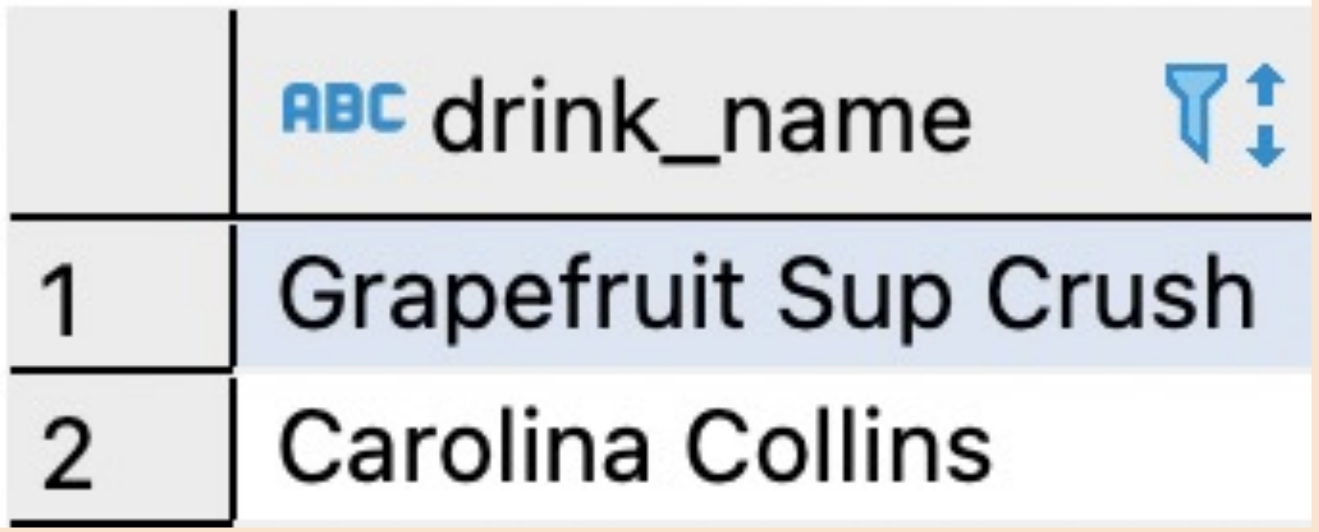


	drink
1	Grapefruit Sup Crush
2	Paloma Fresca
3	Miller Lite

```
SELECT drink FROM (
    SELECT CASE WHEN diet_name in ('Grapefruit
interaction', 'Gluten-free') then drink_name ELSE
NULL END drink FROM Drink d JOIN Drink_has_Diet
dd on d.drink_id = dd.drink_id JOIN
Dietary_Restriction dr on dr.diet_id = dd.diet_id
    UNION
    SELECT CASE WHEN diet_name in ('Grapefruit
interaction', 'Gluten-free') then common_drink_name
ELSE NULL END drink FROM Common_Drink d JOIN
Common_Drink_has_Diet dd on d.common_drink_id
= dd.common_drink_id JOIN Dietary_Restriction dr on
dr.diet_id = dd.diet_id
) cte WHERE cte.drink IS NOT NULL
```

Report 2

Use case: Jessie has a bar cart with vodka, Sierra Mist, gin, simple syrup, cola, and lemon juice. What drinks from Chapel Hill can she recreate?

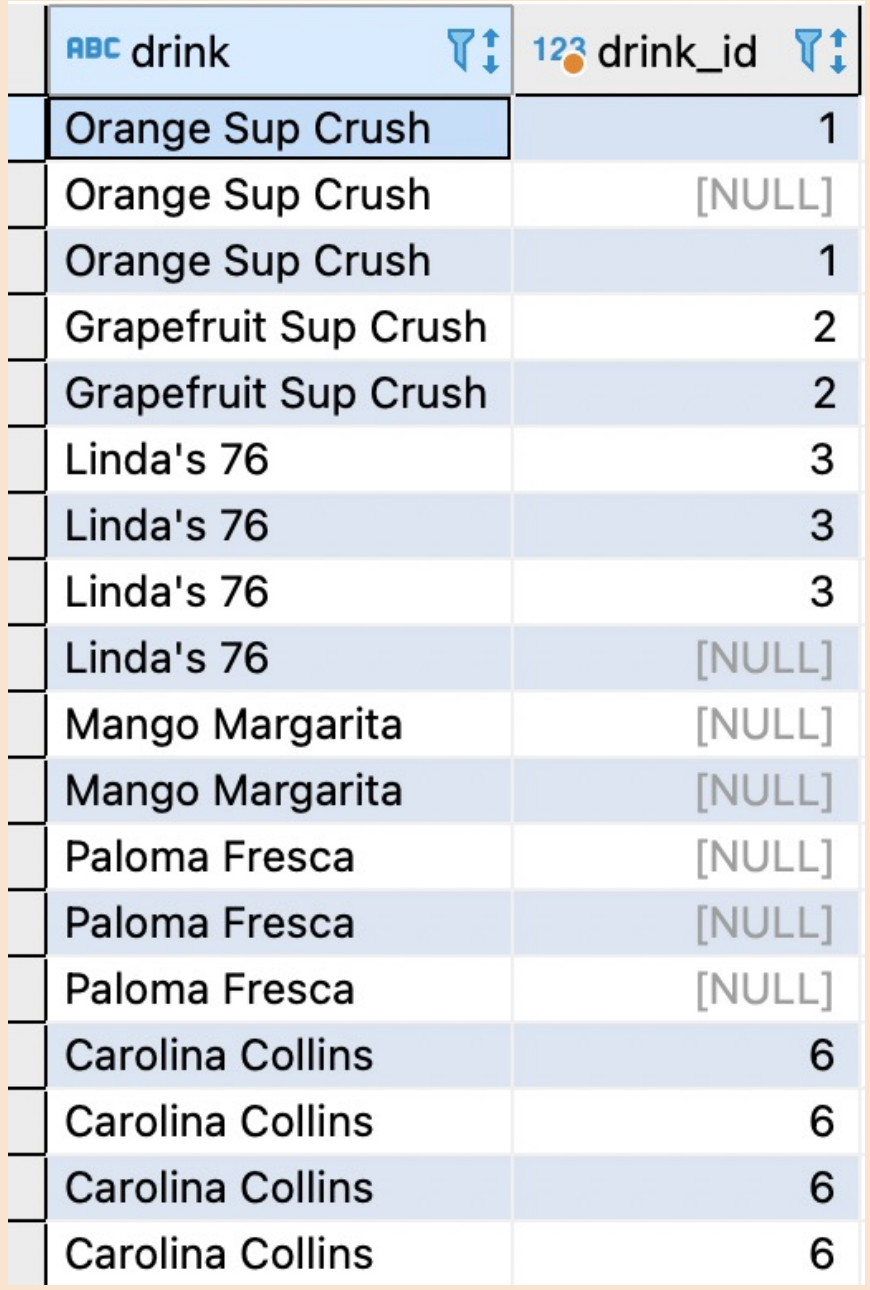


	drink_name
1	Grapefruit Sup Crush
2	Carolina Collins

```
WITH Full_list_of_drinks AS (

SELECT DISTINCT
        placeholder, drink_name
FROM (
        SELECT 'placeholder', drink
        FROM (
                SELECT d.drink_name drink,
                CASE WHEN da.drink_att NOT IN('Vodka', 'Sierra
                Mist', 'Gin', 'Simple syrup', 'Cola', 'Lemon juice')
                THEN NULL ELSE d.drink_id END drink_id

     FROM Drink d JOIN Drink_has_Att dha on d.drink_id = dha.drink_id
     JOIN Drink_Attribute da on da.drink_att_id = dha.drink_att_id )
     subquery WHERE drink_id IS NULL ) wrong_drinks RIGHT JOIN Drink d
     on wrong_drinks.drink = d.drink_name
)

SELECT drink_name FROM Full_list_of_drinks WHERE placeholder IS
NULL;
```

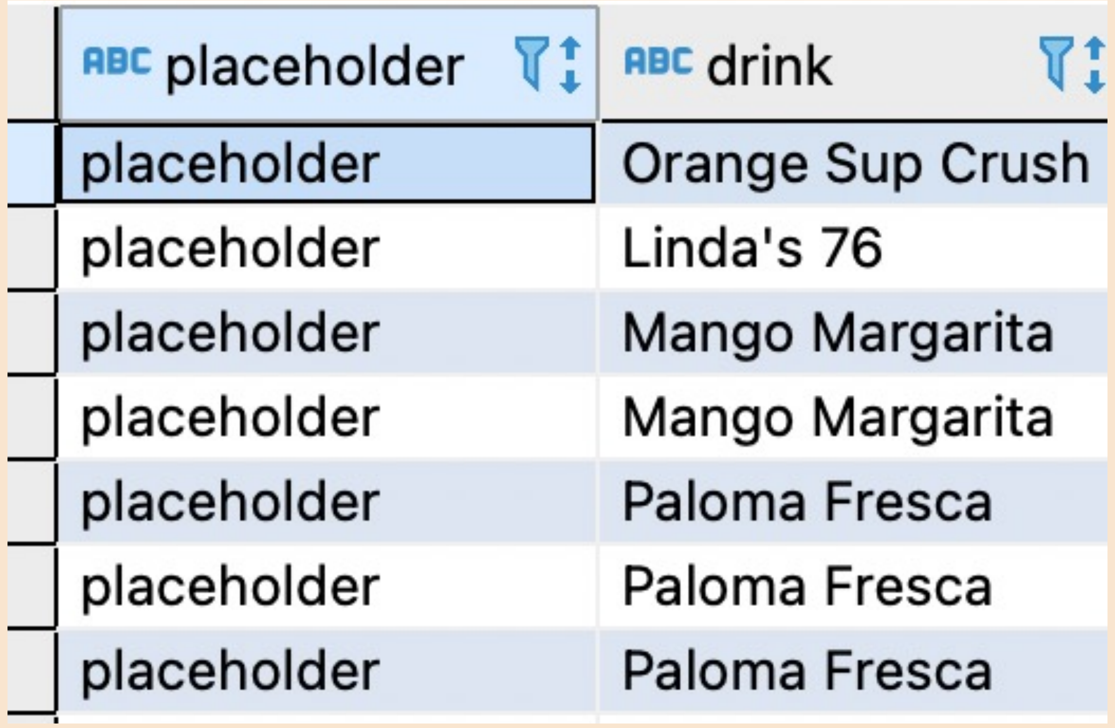
Report 2

```
SELECT d.drink_name drink, CASE WHEN da.drink_att NOT
IN('Vodka', 'Sierra Mist', 'Gin', 'Simple syrup', 'Cola', 'Lemon
juice') THEN d.drink_id = NULL ELSE d.drink_id END drink_id
FROM Drink d JOIN Drink_has_Att dha on d.drink_id =
dha.drink_id JOIN Drink_Attribute da on da.drink_att_id =
dha.drink_att_id
```



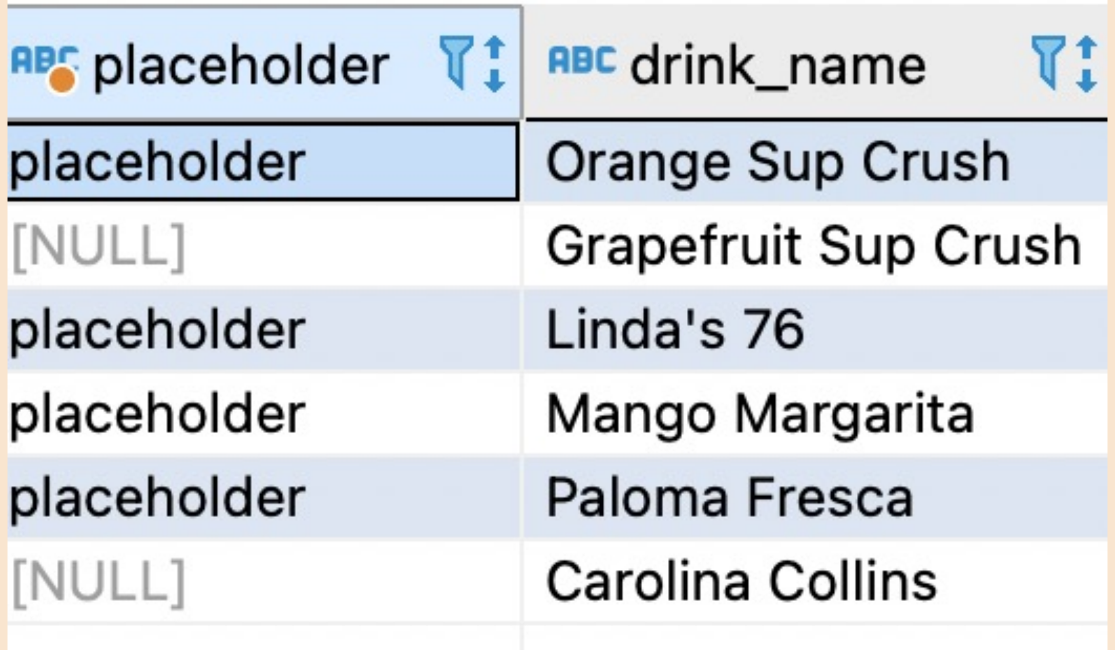
ABC drink	123 drink_id
Orange Sup Crush	1
Orange Sup Crush	[NULL]
Orange Sup Crush	1
Grapefruit Sup Crush	2
Grapefruit Sup Crush	2
Linda's 76	3
Linda's 76	3
Linda's 76	3
Linda's 76	[NULL]
Mango Margarita	[NULL]
Mango Margarita	[NULL]
Paloma Fresca	[NULL]
Paloma Fresca	[NULL]
Paloma Fresca	[NULL]
Carolina Collins	6
Carolina Collins	6
Carolina Collins	6
Carolina Collins	6

```
Select 'placeholder', drink
FROM SUBQUERY_1 where drink_id is null
```



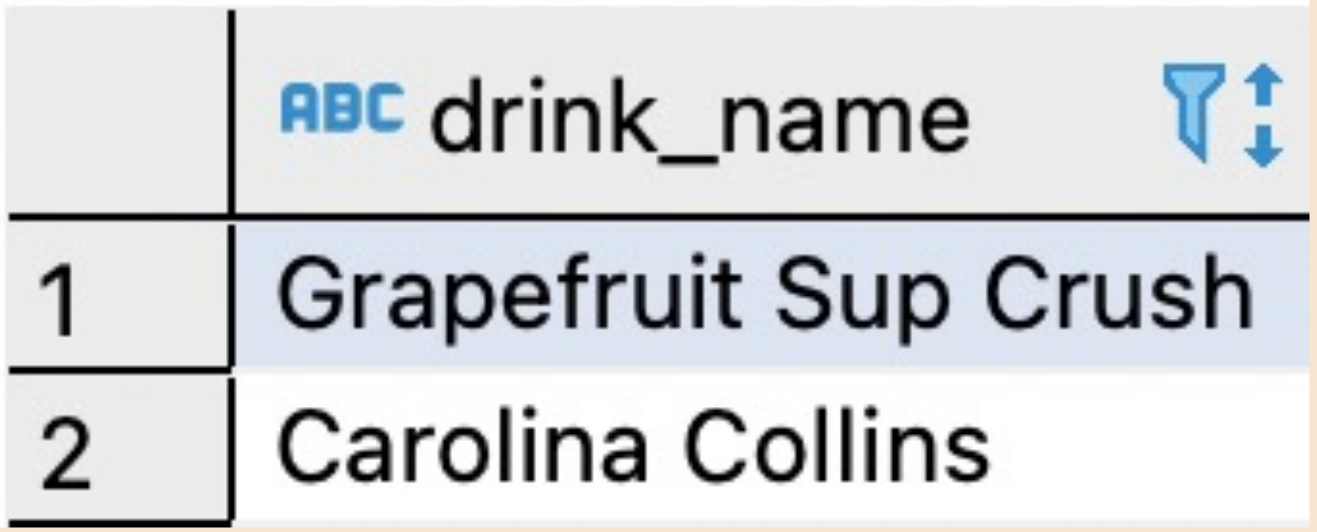
ABC placeholder	ABC drink
placeholder	Orange Sup Crush
placeholder	Linda's 76
placeholder	Mango Margarita
placeholder	Mango Margarita
placeholder	Paloma Fresca
placeholder	Paloma Fresca
placeholder	Paloma Fresca

```
SELECT DISTINCT placeholder, drink_name
FROM SUBQUERY_2 RIGHT JOIN ON drink_name
```



ABC placeholder	ABC drink_name
placeholder	Orange Sup Crush
[NULL]	Grapefruit Sup Crush
placeholder	Linda's 76
placeholder	Mango Margarita
placeholder	Paloma Fresca
[NULL]	Carolina Collins

```
Select drink_name FROM SUBQUERY_3 where placeholder is null
```



	ABC drink_name
1	Grapefruit Sup Crush
2	Carolina Collins

06

Conclusion

What was easy:
-Querying from use cases
-Inputting data

Goal:
To assist in picking restaurants for ordering drinks when a group of people all have different dietary needs.

What was hard:
-Developing functional schema
-Normalizing tables

Questions?

Group4

Thanks