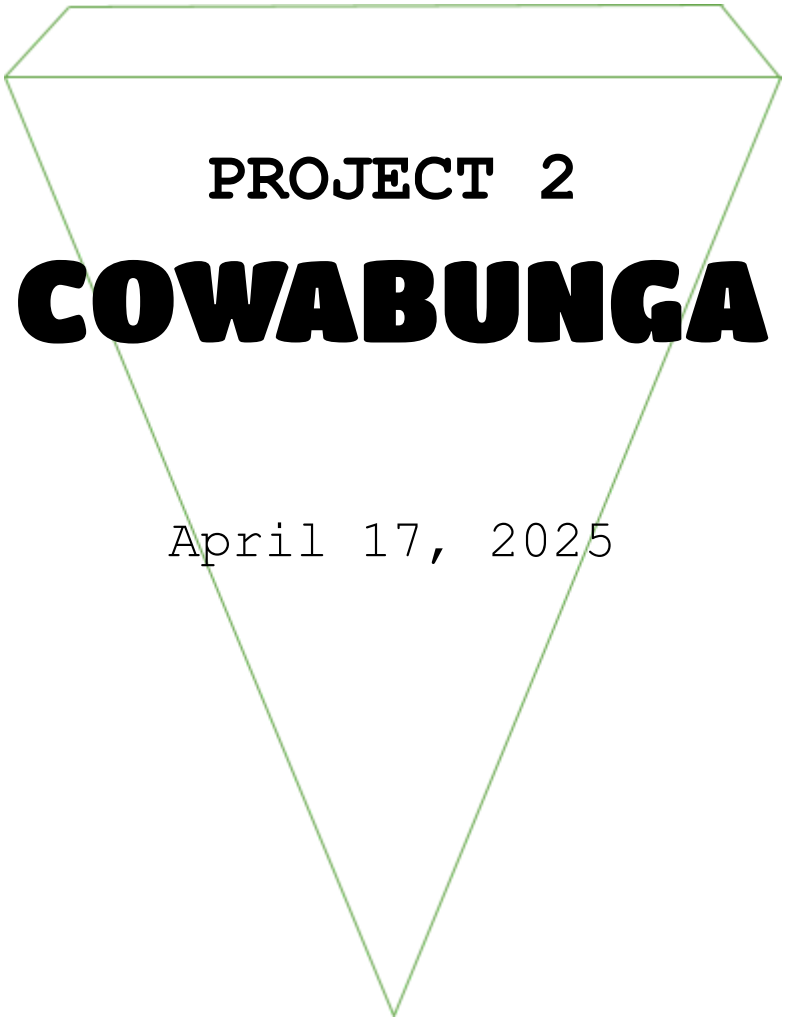


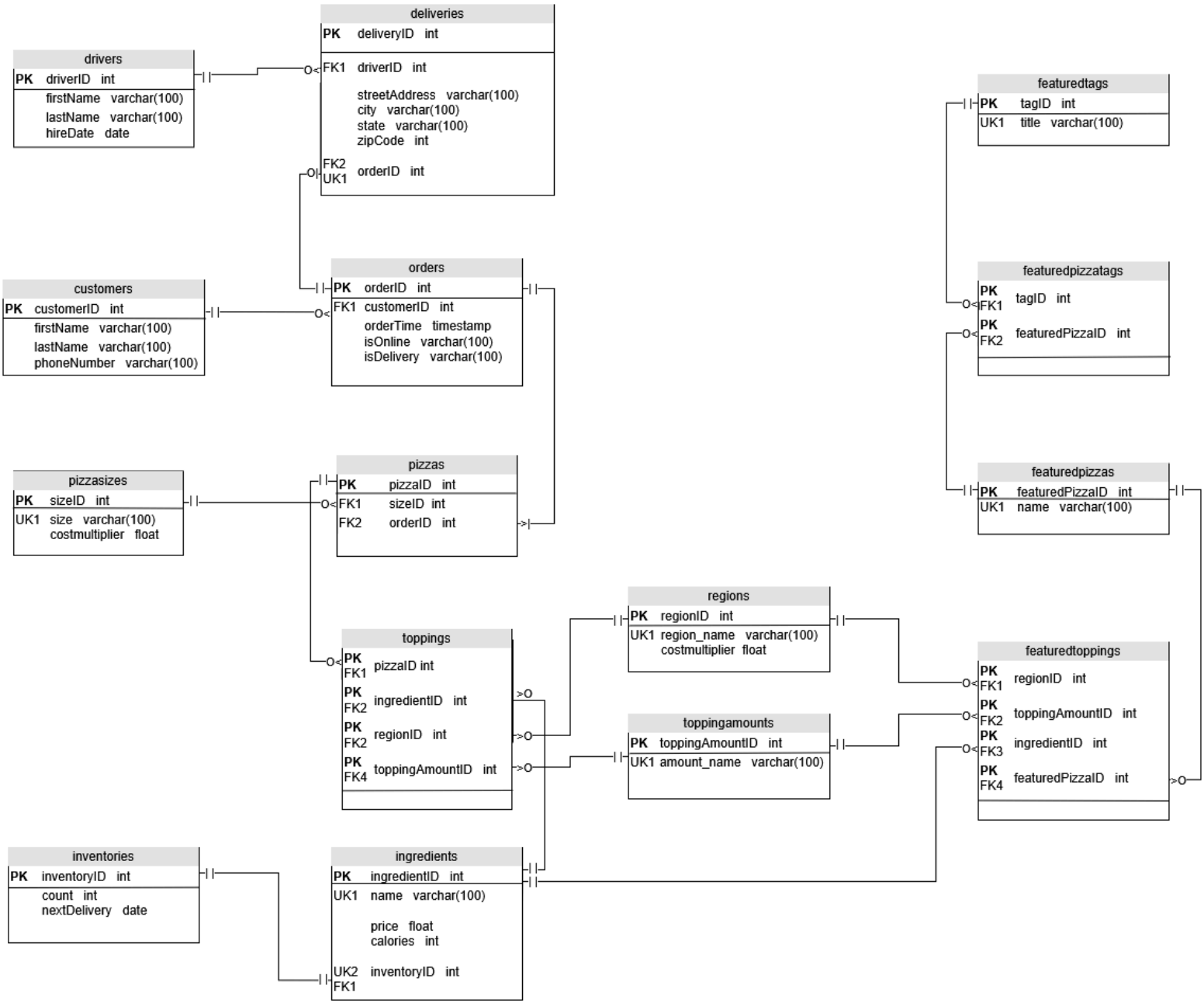
PROJECT 2

COWABUNGA

April 17, 2025

Kimberly Chau

EML:


deliveries
PK deliveryID int
FK1 driverID int
streetAddress varchar(100)
city varchar(100)
state varchar(100)
zipCode int
FK2 UK1 orderID int
drivers
PK driverID int
firstName varchar(100)
lastName varchar(100)
hireDate date
featuredtags
PK tagID int
UK1 title varchar(100)
orders
PK orderID int
FK1 customerID int
orderTime timestamp
isOnline varchar(100)
isDelivery varchar(100)
customers
PK customerID int
firstName varchar(100)
lastName varchar(100)
phoneNumber varchar(100)
featuredpizzatags
PK FK1 tagID int
PK FK2 featuredPizzaID int
pizzas
PK pizzaID int
FK1 sizeID int
FK2 orderID int
pizzasizes
PK sizeID int
UK1 size varchar(100)
costmultiplier float
featuredpizzas
PK featuredPizzaID int
UK1 name varchar(100)
regions
PK regionID int
UK1 region_name varchar(100)
costmultiplier float
toppings
PK FK1 pizzaID int
PK FK2 ingredientID int
PK FK2 regionID int
PK FK4 toppingAmountID int
featuredtoppings
PK FK1 regionID int
PK FK2 toppingAmountID int
PK FK3 ingredientID int
PK FK4 featuredPizzaID int
toppingamounts
PK toppingAmountID int
UK1 amount_name varchar(100)
inventories
PK inventoryID int
count int
nextDelivery date
ingredients
PK ingredientID int
UK1 name varchar(100)
price float
calories int
UK2 FK1 inventoryID int

INSERT STATEMENTS:

Customers

Two Customers:

- Neal Terrell, 562-985-9999
- David Brown, 562-985-0000

```
insert into pizza.customers (customer_id, first_name, last_name, phone_number)
values  ( customer_id 1,  first_name 'Neal',  last_name 'Terrell',  phone_number '562-985-9999'),
        ( customer_id 2,  first_name 'David',  last_name 'Brown',  phone_number '562-985-0000');
```

Deliveries

```
insert into pizza.deliveries (delivery_id, driver_id, order_id, street_address, city, state, zip_code)
values  ( delivery_id 1,  driver_id 1,  order_id 2,  street_address '1250 Bellflower Blvd',  city 'Long Beach',  state 'CA',  zip_code 90815);
```

Drivers

Two Drivers:

- John Doe
- Jane Doe

```
insert into pizza.drivers (driver_id, first_name, last_name, hire_date)
values  ( driver_id 2,  first_name 'Jane',  last_name 'Doe',  hire_date '2021-07-12'),
        ( driver_id 1,  first_name 'John',  last_name 'Doe',  hire_date '2020-04-03');
```

FeaturesPizzas

A Featured Pizza called The Maddy, which has only extra cheese on the whole pizza.

- Tagged as "customer favorite" and "new option".

A Featured Pizza called Meat Lover, which has extra pepperoni and extra sausage on the whole pizza.

- Your Featured Pizzas should not interact with any actual pizzas that have been ordered, even if a Customer based their order on the featured pizza.
- Tagged as "customer favorite".

```
insert into pizza.featuredpizzas (featuredpizza_id, name)
values  ( featuredpizza_id 1,  name 'The Maddy'),
        ( featuredpizza_id 2,  name 'Meat Lover');
```

FeaturedPizzaTags

```
insert into pizza.featuredpizzatags (tag_id, featuredpizza_id)
values  ( tag_id 1,  featuredpizza_id 1),
        ( tag_id 1,  featuredpizza_id 2),
        ( tag_id 2,  featuredpizza_id 2);
```

FeaturedTags

```
insert into pizza.featuredtags (tag_id, title)
values  ( tag_id 1,  title 'Customer Favorite'),
        ( tag_id 2,  title 'New Option');
```

FeaturedToppings

```
insert into pizza.featuredtoppings (region_id, toppingamount_id, ingredient_id, featuredpizza_id)
values  ( region_id 1,  toppingamount_id 3,  ingredient_id 1,  featuredpizza_id 1),
        ( region_id 1,  toppingamount_id 3,  ingredient_id 2,  featuredpizza_id 2),
        ( region_id 1,  toppingamount_id 3,  ingredient_id 5,  featuredpizza_id 2);
```

Ingredients

Five Ingredients:

- Cheese
- Pepperoni
- Mushroom
- Pineapple
- Sausage

```
insert into pizza.ingredients (ingredient_id, name, price, calories, inventory_id)
values  ( ingredient_id 2,  name 'Pepperoni',  price 5.99,  calories 150,  inventory_id 2),
        ( ingredient_id 3,  name 'Mushroom',  price 3.75,  calories 50,  inventory_id 3),
        ( ingredient_id 1,  name 'Cheese',  price 3.49,  calories 113,  inventory_id 1),
        ( ingredient_id 4,  name 'Pineapple',  price 2.34,  calories 452,  inventory_id 4),
        ( ingredient_id 5,  name 'Sausage',  price 4.97,  calories 229,  inventory_id 5);
```

Inventories

10 servings of Cheese and 2 servings of Pepperoni are in the restaurant Inventory.

```
insert into pizza.inventories (inventory_id, count, next_delivery)
values  ( inventory_id 3,  count null,  next_delivery null),
        ( inventory_id 5,  count null,  next_delivery null),
        ( inventory_id 4,  count null,  next_delivery null),
        ( inventory_id 2,  count 2,  next_delivery null),
        ( inventory_id 1,  count 10,  next_delivery null);
```

Orders

An Order for take-out, placed over the phone by Neal Terrell on 2025-02-21 5:00pm:

- A Large pizza, with Regular Cheese on the Whole Pizza, Extra Pepperoni on the Left Half, and Regular Mushrooms on the Right Half.

An Order for delivery to 1250 Bellflower Blvd, Long Beach, CA 90815, placed on the web by David Brown on 2025-02-21 6:00pm:

- A Small pizza, with Extra Cheese on the Whole Pizza.
- A Large pizza, with Regular Cheese on the Whole Pizza, Light Pepperoni on the Whole Pizza, Regular Pineapple on the Whole Pizza, and Extra Mushrooms on the Left Half.
- John Doe will deliver this order.

```
insert into pizza.orders (order_id, customer_id, order_time, is_online, is_delivery)
values  ( order_id 1, customer_id 1, order_time '2025-02-21 17:00:00.000000', is_online 'placed on phone', is_delivery 'take-out'),
        ( order_id 2, customer_id 2, order_time '2025-02-21 18:00:00.000000', is_online 'placed on web', is_delivery 'delivery');
```

Pizzas

```
insert into pizza.pizzas (pizza_id, size_id, order_id)
values  ( pizza_id 1, size_id 3, order_id 1),
        ( pizza_id 2, size_id 1, order_id 2),
        ( pizza_id 3, size_id 3, order_id 2);
```

PizzaSizes

```
insert into pizza.pizzasizes (size_id, size, cost_multiplier)
values  ( size_id 1, size 'small', cost_multiplier 1.1),
        ( size_id 2, size 'medium', cost_multiplier 1.3),
        ( size_id 3, size 'large', cost_multiplier 1.6);
```

Regions

```
insert into pizza.regions (region_id, region_name, cost_multiplier)
values  ( region_id 1, region_name 'whole pizza', cost_multiplier 1),
        ( region_id 2, region_name 'left half only', cost_multiplier 0.7),
        ( region_id 3, region_name 'right half only', cost_multiplier 0.7);
```

ToppingAmounts

```
insert into pizza.toppingamounts (toppingamount_id , amount_name)
values  ( toppingamount_id 1, amount_name 'regular'),
        ( toppingamount_id 2, amount_name 'light'),
        ( toppingamount_id 3, amount_name 'extra');
```

Toppings

```
insert into pizza.toppings (pizza_id, ingredient_id, region_id, toppingamount_id)
values  ( pizza_id 1, ingredient_id 1, region_id 1, toppingamount_id 1),
        ( pizza_id 1, ingredient_id 2, region_id 2, toppingamount_id 3),
        ( pizza_id 1, ingredient_id 3, region_id 3, toppingamount_id 1),
        ( pizza_id 2, ingredient_id 1, region_id 1, toppingamount_id 3),
        ( pizza_id 3, ingredient_id 1, region_id 1, toppingamount_id 1),
        ( pizza_id 3, ingredient_id 2, region_id 1, toppingamount_id 2),
        ( pizza_id 3, ingredient_id 4, region_id 1, toppingamount_id 1),
        ( pizza_id 3, ingredient_id 3, region_id 2, toppingamount_id 3);
```

QUERIES:

Query 1

Select the names of all customers that have placed at least one Order.

```sql
-- 1
select first_name, last_name
from customers
where customer_id IN (
    select customer_id
    from orders
    group by customer_id
    having count(*) >= 1
    );
```

Query 2

Select the names of all Drivers who have never delivered an order.

```sql
--2
select first_name, last_name
from drivers
where driver_id NOT IN (
    select distinct driver_id
    from deliveries
    where driver_id IS NOT NULL
    );
```

Query 3

Determine the name of the Ingredient that has been included on the most Pizzas.

```sql
--3
select i.name, count(*) as max_count
from toppings
    inner join ingredients i 1..n<->1: using (ingredient_id)
group by i.name
having count(*) = (
    select max(count)
    from (
        select ingredient_id, count(*) as count
        from toppings
        group by ingredient_id
         ) as ingredient_count
    );
```

Query 4

For each Pizza, determine the total cost of its Toppings. For each Topping, multiply the cost of its ingredient times the costmultiplier of the pizza's size times the costmultiplier of the topping's region. Sum these values for each Pizza.

```
--4
select pizza_id, SUM(i.price * s.cost_multiplier * r.cost_multiplier) as total_cost
from toppings
inner join ingredients i 1..n<->1: using (ingredient_id)
inner join regions r 1..n<->1: using (region_id)
inner join pizzas p 1..n<->1: using (pizza_id)
inner join pizzasizes s 1..n<->1: using (size_id)
group by pizza_id;
```

Query 5

In a single query, count the number of "Scheduled" and "Unscheduled" orders. A "Scheduled" order is a delivery order that has been assigned to a Driver; an Unscheduled order is a delivery order that has not yet been assigned to a Driver. Your query should produce two rows of output, with *status* and *count* columns.

```
--5
select order_status, count(*) as status_count
from (
    select case
                when driver_id IS NULL then 'Unscheduled'
                else 'Scheduled'
           end as order_status
    from deliveries
     ) as status
group by order_status
```