

Database Engineer Capastone

▼ Er Diagram.

```
erDiagram
    Bookings {
        int booking_id PK
        date booking_date
        int table_number
        int customer_id FK
        int staff_id FK
    }

    Customers {
        int customer_id PK
        string first_name
        string last_name
        string email
        string phone_number
    }

    Orders {
        int order_id PK
        date order_date
        int quantity
        float total_cost
        int booking_id FK
    }

    OrderMenu {
        int order_menu_id PK
        int order_id FK
        int menu_id FK
    }
```

```

        int quantity
    }

    OrderDeliveryStatus {
        int status_id PK
        int order_id FK
        date delivery_date
        string delivery_status
    }

    Menu {
        int menu_id PK
        string item_name
        string item_type
        float price
    }

    Staff {
        int staff_id PK
        string first_name
        string last_name
        string role
        float salary
        string email
        string phone_number
    }

    Bookings ||--o{ Customers : "belongs to"
    Bookings ||--o{ Orders : "includes"
    Orders ||--o{ OrderMenu : "contains"
    OrderMenu }o--|| Menu : "references"
    Orders ||--o{ OrderDeliveryStatus : "has"
    Staff ||--o{ Bookings : "manages"

```

▼ Data model in MySQL workbench


```

        `LittleLemonDB`.`Orders`.`quantity` AS `Quantity`,
        `LittleLemonDB`.`Orders`.`totalCost` AS `Cost`
FROM
    `LittleLemonDB`.`Orders`
WHERE
    (`LittleLemonDB`.`Orders`.`quantity` > 2)

```

- ▼ Extract data about the customer which have order greater then \$150.

```

select
c.customerId as CustomerID,
concat(c.firstName,' ',lastName) as 'FullName',
o.orderId as OrderID,
o.totalCost as Cost,
m.itemName as MenuName,
m.itemType as CourseName
from Orders o
join Bookings b
on b.bookingId = o.bookingId
join Customers c
on c.customerId = b.customerId
join OrderMenu om
on om.orderId = o.orderId
join Menu m
on m.menuId = om.menuId;

```

- ▼ Create a stored Procedure to get the max quantity in Orders.

```

DELIMITER
CREATE PROCEDURE `GetMaxQuantity`()
BEGIN
    select max(quantity) as 'Max Quantity in Orders' from
    Orders;
END

```

```
call GetMaxQuantity();
```

- ▼ Create a prepared Statement call GetOrderDetail to get the order detail by orderId.

```
PREPARE GetOrderDetail FROM 'SELECT orderId as OrderID, order  
quantity as OrderQuantity,  
totalCost as Cost  
FROM Orders WHERE orderId = ?';
```

```
SET @id = 1;  
EXECUTE GetOrderDetail USING @id;
```

- ▼ Create a stored procedure Cancel Order which will delete the order by orderId input.

```
DELIMITER &&  
CREATE PROCEDURE DeleteOrderByID (IN _orderId int)  
BEGIN  
    delete from Orders o  
    where o.orderId = _orderId  
END&&  
DELIMITER ;  
  
call DeleteOrderByID(1);
```

- ▼ Create a procedure to Add booking to add a new table record.

```
CREATE PROCEDURE `AddBooking`(  
    IN p_bookingId int,  
    IN p_bookingDate date,  
    IN p_tableNumber int,  
    IN p_customerId int,
```

```

IN p_staffId int
)
BEGIN
    INSERT into Bookings(bookingId,bookingDate,tableNumber,customerId)
    (
        p_bookingId,
        p_bookingDate,
        p_tableNumber,
        p_customerId,
        p_staffId
    );

    SELECT 'Booking added successfully.' AS Confirmation;
END

```

▼ Create an update booking procedure.

```

CREATE PROCEDURE `UpdateBookingDate`(
IN p_bookingId int,
IN p_bookingDate date
)
BEGIN
    UPDATE Bookings
    SET bookingDate = p_bookingDate
    WHERE bookingId = p_bookingId;

    SELECT concat('Booking ',p_bookingId,' Updated.') as Confirmation;
END

```

▼ Create a cancel booking procedure.

```

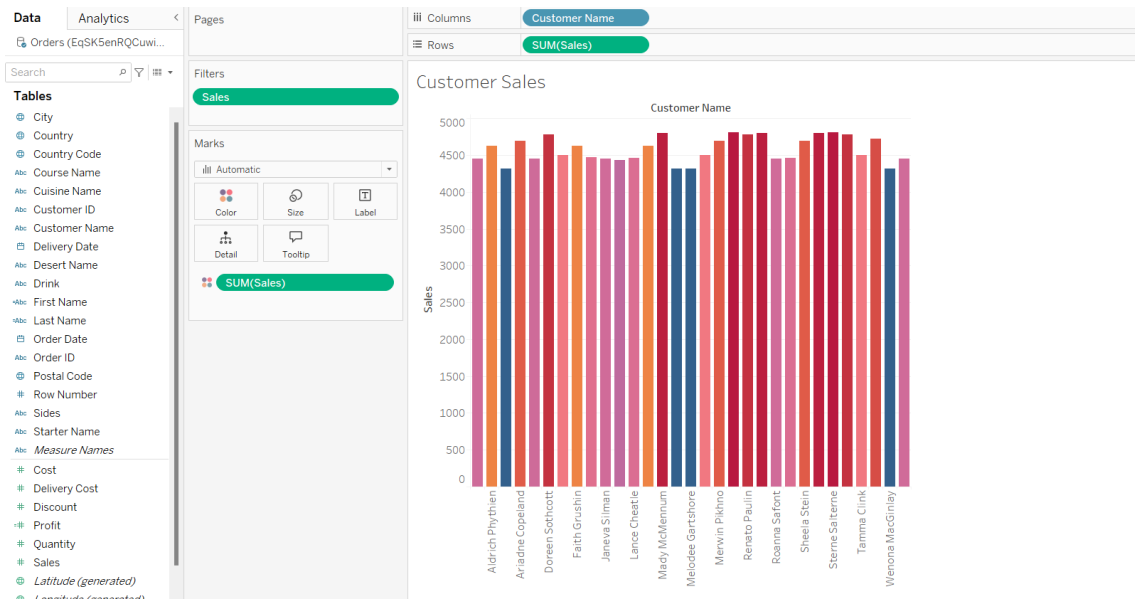
CREATE PROCEDURE `DeleteByBookingId`(IN _orderId int)
BEGIN
    delete from Orders
    where orderId = _orderId;

```

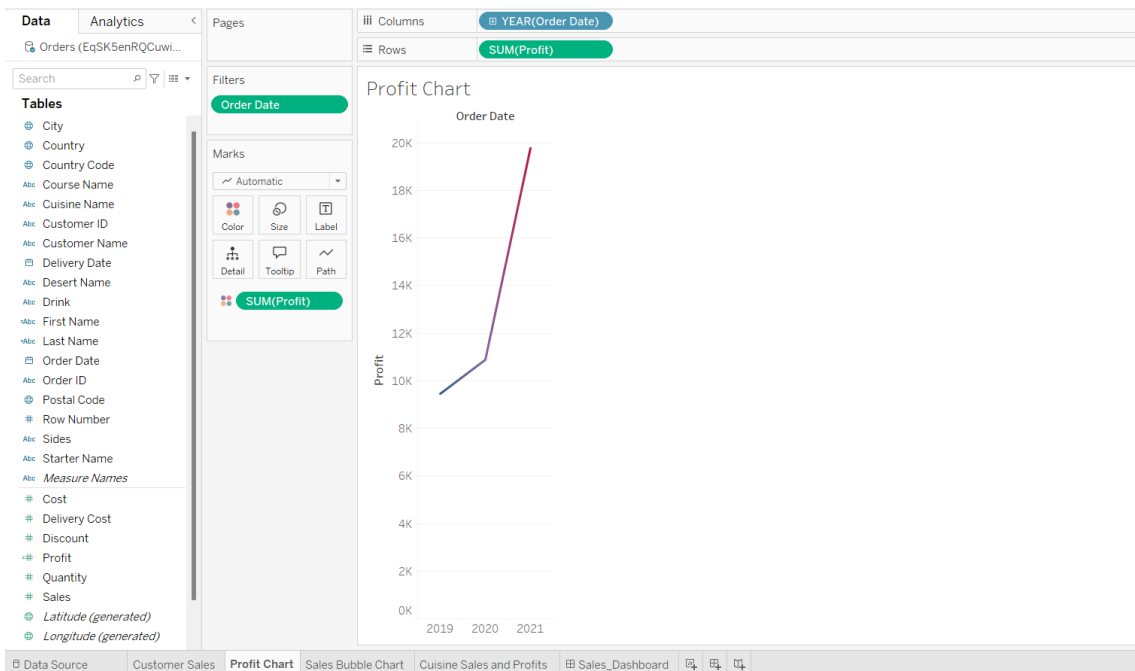
```
SELECT concat('Booking ',p_bookingId,' Cancelled.') as Conf:
END
```

▼ Tableau Analysis

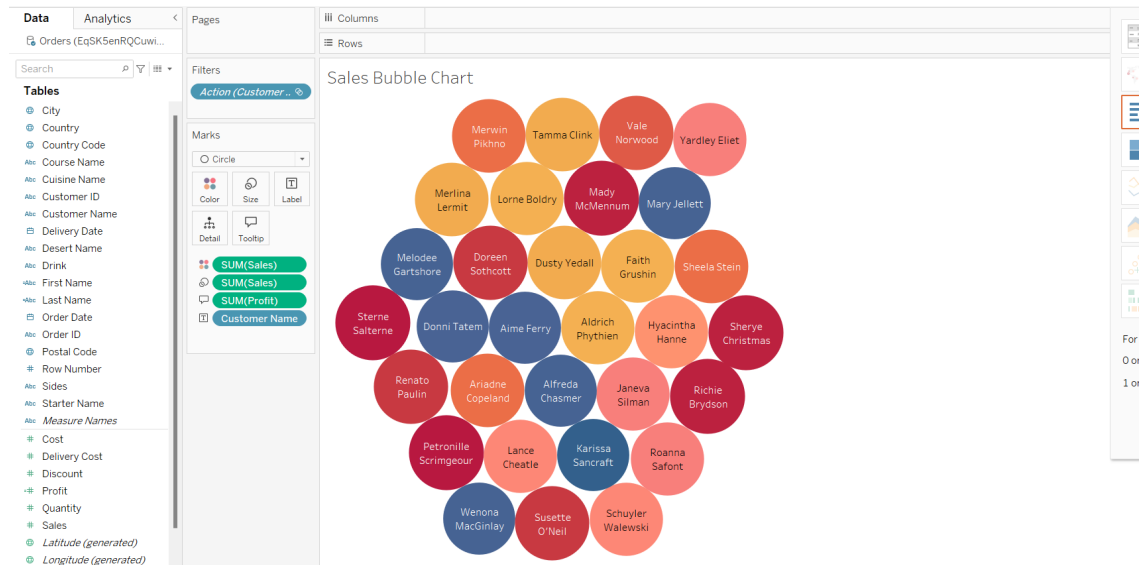
▼ Customer Sales



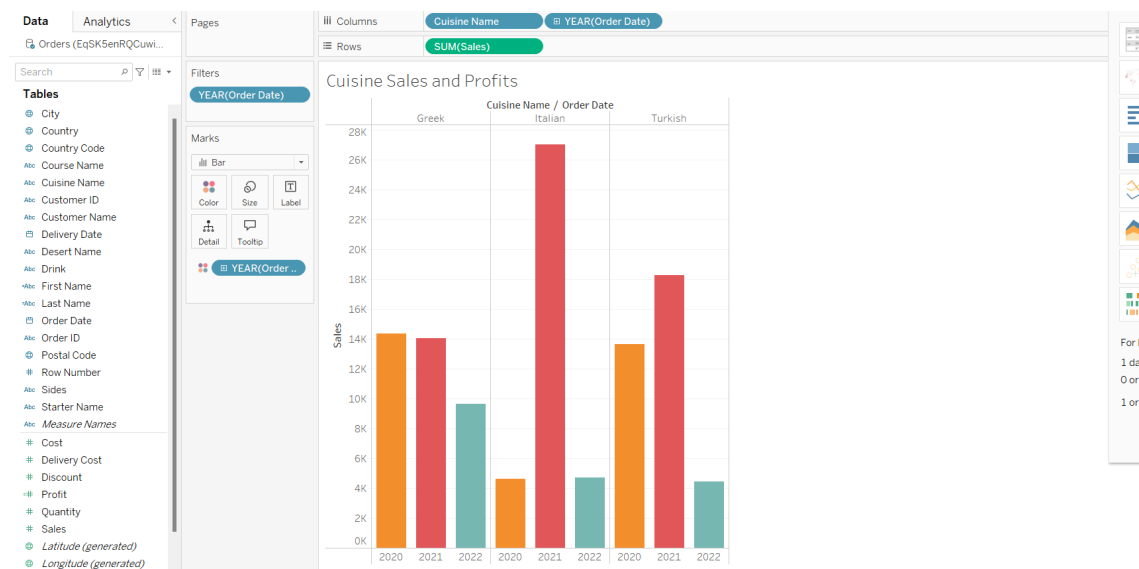
▼ Profit Chart



▼ Sales Bubble Chart



▼ Cuisine Sales and Profits



▼ Sales Dashboard

