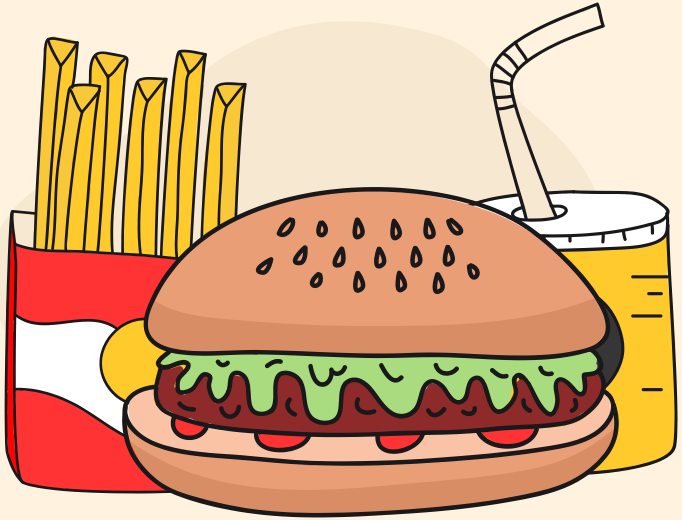


Database Development Project

Shake Shack

Christian Condon, Caleb Esmay, Kristian King, Tini Mai, Danny Ostler

Project Overview

Shake Shack

- American fast casual restaurant chain with many locations in Utah.

Business Model

- **Mission: Stand For Something Good**
  - Sourcing premium ingredients from like-minded partners
  - Make use of animal welfare practices
  - Worry-free dining experience
  - Improve customer services and customer experiences using data

User Requirements

Flexibility

Customers can place orders over the phone

Customers can place orders using available tablets in line

Employees can view orders in the system

Employees can notify customers when their orders are ready

Accessibility

Manager can keep track of menu item prices

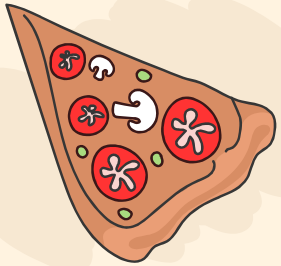
Manager can compare prices among vendors

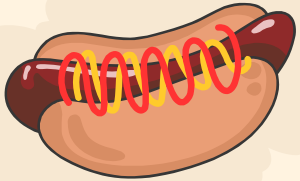
Business Rules

1. Shake Shack has many customers

2. Shake Shack wants to track the name and address of each new customer

3. Each Customer has one or many emails





4. The company tracks which customer email is the primary email

5. Customers can place orders

6. Shake Shack wants to track the date and the time of each order

7. Many Orders can be placed in one and only one Restaurant

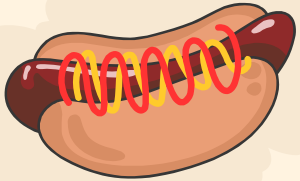
8. Each Order can be placed in one and only one Restaurant

Business Rules (cont.)

9. Employees can fulfill many Orders

10. Each Order is fulfilled by one and only one Employee

11. Shake Shack wants to track the name, address, SSN, Title, and Salary of its employees



12. Each Employee has one or many emails

13. Shake Shack wants to tracks which email is the primary email

14. Orders can have none or many Meals

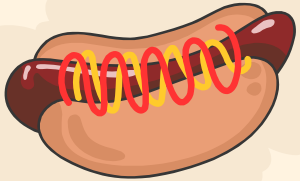
15. Shake Shack wants to track the Meal's name, a description of the meal, and its sales price

16. Each meal comes from a supplier

Business Rules (cont.)

17. Shake Shack wants to track the Unit Cost of the meal

18. Shake Shack wants to track which suppliers are provisioning the meals



20. The company wants to keep track of the quantity of meals sold

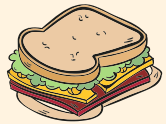
21. Shake Shack wants to track the Price of each meal

19. A Supplier supplies many different meals

Data Outputs/Business Questions

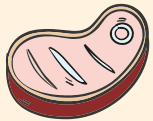
1. What meal is losing us the most money as a company?
2. What meal is the most profitable for our company?

3. What is our overall sales revenue for the past month?
4. How many customers are in our database?

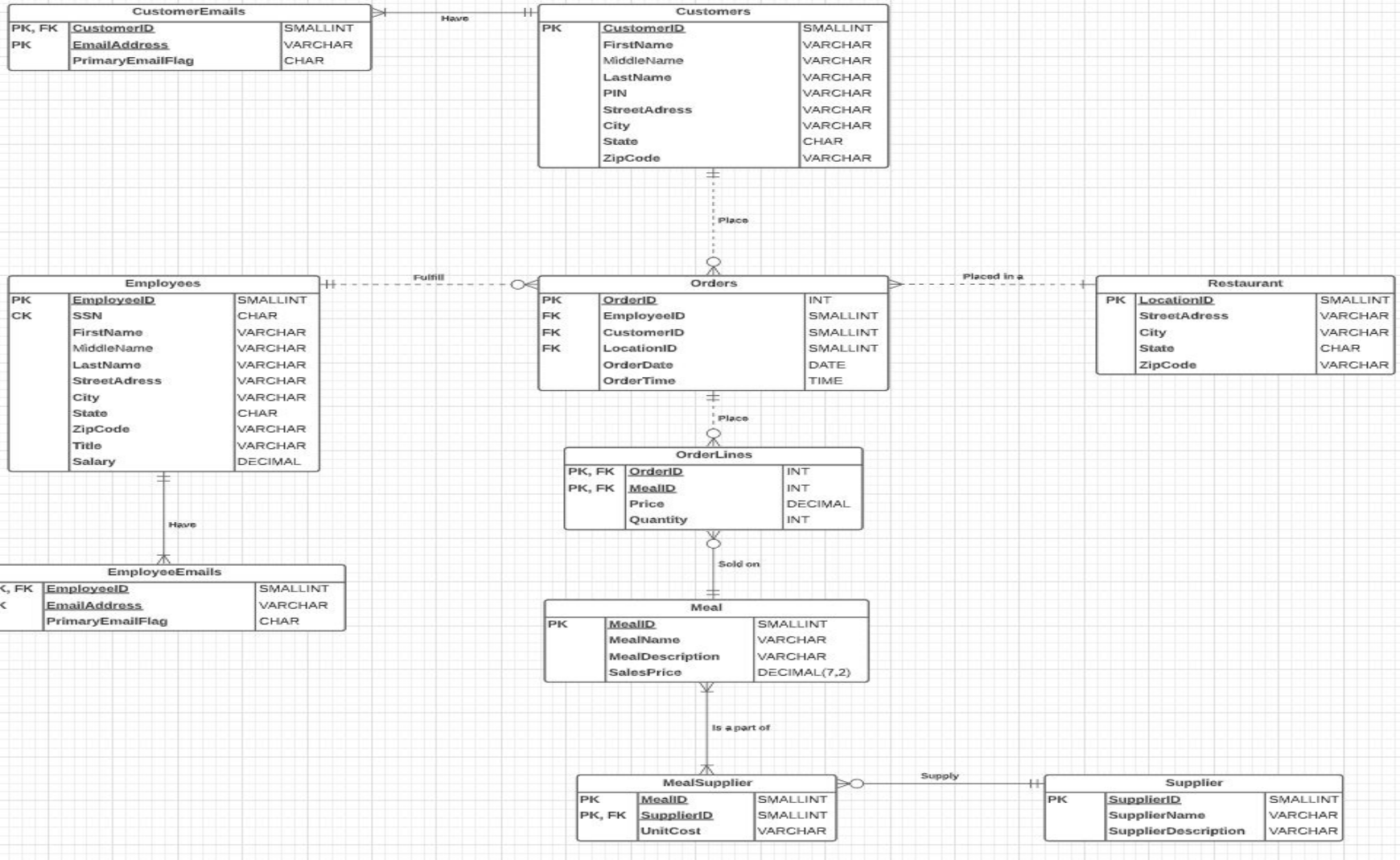


5. Which Restaurant accounts for the most customers?
6. What is the lowest salary for an employee?

7. Which shift needs the most employees?
8. What is the average Salary for a Shake Shack employee?



9. What ingredients are the most expensive to use?
10. How can we track and update employees if they move, switch jobs or salaries?


CustomerEmails
PK, FK CustomerID SMALLINT
PK EmailAddress VARCHAR
PrimaryEmailFlag CHAR
Have
Customers
PK CustomerID SMALLINT
FirstName VARCHAR
MiddleName VARCHAR
LastName VARCHAR
PIN VARCHAR
StreetAdress VARCHAR
City VARCHAR
State CHAR
ZipCode VARCHAR
Place
Employees
PK EmployeeID SMALLINT
CK SSN CHAR
FirstName VARCHAR
MiddleName VARCHAR
LastName VARCHAR
StreetAdress VARCHAR
City VARCHAR
State CHAR
ZipCode VARCHAR
Title VARCHAR
Salary DECIMAL
Fulfill
Orders
PK OrderID INT
FK EmployeeID SMALLINT
FK CustomerID SMALLINT
FK LocationID SMALLINT
OrderDate DATE
OrderTime TIME
Placed in a
Restaurant
PK LocationID SMALLINT
StreetAdress VARCHAR
City VARCHAR
State CHAR
ZipCode VARCHAR
Place
Have
OrderLines
PK, FK OrderID INT
PK, FK MealID INT
Price DECIMAL
Quantity INT
EmployeeEmails
PK, FK EmployeeID SMALLINT
PK EmailAddress VARCHAR
PrimaryEmailFlag CHAR
Sold on
Meal
PK MealID SMALLINT
MealName VARCHAR
MealDescription VARCHAR
SalesPrice DECIMAL(7,2)
Is a part of
MealSupplier
PK MealID SMALLINT
PK, FK SupplierID SMALLINT
UnitCost VARCHAR
Supply
Supplier
PK SupplierID SMALLINT
SupplierName VARCHAR
SupplierDescription VARCHAR

SELECT Queries

```sql
--Select query for most ordered meal and least ordered meal
SELECT M.MealName, M.MealDescription, M.SalesPrice, COUNT(M.MealID) As TimesOrdered
FROM Orders AS O
INNER JOIN OrderLines AS OL ON O.OrderID = OL.OrderID
INNER JOIN Meal AS M ON OL.MealID = M.MealID
GROUP BY M.MealName, M.MealDescription,SalesPrice
ORDER BY TimesOrdered DESC


--Select query for location most frequently visited
SELECT
CONCAT(R.StreetAddress, ' ', R.City, ', ', R.State, ' ', R.ZipCode)AS RestaurantAddress,
```

100 %

Results | Messages

	MealName	MealDescription	SalesPrice	TimesOrdered
1	Buffalo Chicken	Crispy, hand-breaded chicken breast covered in our B...	7.29	10
2	Chicken Shack	Crispy chicken breast over lettuce, pickles, buttermilk h...	6.29	9
3	Hamburger	Griddled patty and your choice of toppings (contains mi...	4.29	9
4	Shack Stack	Cheeseburger topped with a Shroom Burger with lettuc...	9.59	9
5	Bacon Cheeseburger	Cheeseburger topped with applewood-smoked bacon ...	5.29	9
6	Shroom Burger	(Vegetarian) Crisp-fried portobello mushroom filled with ...	6.99	8
7	Cheeseburger	Griddled patty with cheese and your choice of toppings...	4.79	8
8	SmokeShack	Cheeseburger with applewood-smoked bacon, choppe...	8.09	7
9	Shackburger	Cheeseburger with lettuce, tomato, ShackSauce (cont...	5.29	6
10	Grilled Cheese	American cheese melted in a grilled potato bun (contai...	6.59	5

```sql
--Select query for location most frequently visited
SELECT
CONCAT(R.StreetAddress, ' ', R.City, ', ', R.State, ' ', R.ZipCode)AS RestaurantAddress,
COUNT(R.RestaurantID) AS TotalOrders,
SUM(OL.Price) AS TotalRevenue
FROM
Restaurant AS R
INNER JOIN Orders AS O ON R.RestaurantID = O.RestaurantID
INNER JOIN OrderLines AS OL ON O.OrderID = OL.OrderID
GROUP BY R.StreetAddress, R.City, R.State, R.ZipCode

--Select query for Employee ordered by amount of orders

SELECT
CONCAT(E.FirstName, ' ',COALESCE(E.MiddleName,''), ' ', E.LastName) AS FullName,
E.EmployeeID, E.Title, E.Salary, Count(O.OrderID) AS OrderTotal
FROM Employees AS E
INNER JOIN Orders AS O ON E.EmployeeID = O.EmployeeID
GROUP BY E.FirstName, E.LastName, E.MiddleName, E.EmployeeID, E.Title, E.Salary
ORDER BY OrderTotal
```

100 %

Results | Messages

	RestaurantAddress	TotalOrders	TotalRevenue
1	11020 State St Suite B, Sandy Sandy, UT 84070	20	690.28
2	20 Third Avenue New York, NY 10003	5	170.19
3	2115 Westlake Ave. Seattle, WA 98121	3	60.11
4	300 Peter Kirk Lane Kirkland, WA 98033	5	185.14
5	3850 North Terminal Drive Terminal Salt Lake City, U...	19	825.98
6	6123 S State St Murray, UT 84107	21	954.67
7	6201 Hollywood Blvd Suite 104 Los Angeles, CA 90...	7	247.44

```sql
--Select query for Employee ordered by amount of orders

SELECT
CONCAT(E.FirstName, ' ',COALESCE(E.MiddleName,''), ' ', E.LastName) AS FullName,
E.EmployeeID, E.Title, E.Salary, Count(O.OrderID) AS OrderTotal
FROM Employees AS E
INNER JOIN Orders AS O ON E.EmployeeID = O.EmployeeID
GROUP BY E.FirstName, E.LastName, E.MiddleName, E.EmployeeID, E.Title, E.Salary
ORDER BY OrderTotal

--Select query for price of meals less than 7 dollars
SELECT *
FROM Meal
WHERE SalesPrice < 7.00
ORDER BY SalesPrice

-- Select query
SELECT Sum(UnitCost),S.SupplierName, S.SupplierID
```

100 %

Results | Messages

	MealID	MealName	MealDescription	SalesPrice
1	5	Hamburger	Griddled patty and your choice of toppings (contains mi...	4.29
2	6	Cheeseburger	Griddled patty with cheese and your choice of toppings...	4.79
3	7	Bacon Cheeseburger	Cheeseburger topped with applewood-smoked bacon ...	5.29
4	1	Shackburger	Cheeseburger with lettuce, tomato, ShackSauce (cont...	5.29
5	10	Chicken Shack	Crispy chicken breast over lettuce, pickles, buttermilk h...	6.29
6	16	Double Hamburger	2 Griddled patties and your choice of toppings (contain...	6.29
7	8	Grilled Cheese	American cheese melted in a grilled potato bun (contai...	6.59
8	11	Chicken Bites	Crispy, whole white meat bites served with honey must...	6.79
9	14	Double Shroom Burger	(Vegetarian) 2 Crisp-fried portobello mushrooms filled wi...	6.99
10	3	Shroom Burger	(Vegetarian) Crisp-fried portobello mushroom filled with ...	6.99

SELECT Queries (cont.)

```sql
--Select query for Employee ordered by amount of orders

SELECT
CONCAT(E.FirstName, ' ',COALESCE(E.MiddleName,''), ' ', E.LastName) AS FullName,
E.EmployeeID, E.Title, E.Salary, Count(O.OrderID) AS OrderTotal
FROM Employees AS E
INNER JOIN Orders AS O ON E.EmployeeID = O.EmployeeID
GROUP BY E.FirstName, E.LastName, E.MiddleName, E.EmployeeID, E.Title, E.Salary
ORDER BY OrderTotal
```

100 %

Results | Messages

	FullName	EmployeeID	Title	Salary	OrderTotal
1	Svend William Jensen	1	Cashier	135000.00	1
2	Joe Duckhouse	7	Certified Supervisor	115000.00	1
3	Lisa Marie Hansen	8	Restaurant General Manager	85000.00	1
4	Johan Ejs Vestergaard	9	Hourly Manager	85000.00	1
5	Jacob Peter Hanen	11	Director of Restaurant	115000.00	1
6	Alexander Hesel	12	Chef	45000.00	1
7	Thomas Buchhave Jensen	13	Cleaner	40000.00	1
8	Phoebe Anne Bridgers	14	Cashier	30000.00	1
9	Elliot Smith	15	Busser	80000.00	1
10	Mon Laferte	16	Cook	90000.00	1
11	Natalia Elizabeth Lafourcade	17	Crew Member	75000.00	1
12	Kevin Allen Parker	18	Hourly Manager	99000.00	1
13	Hank Green	19	Team Member	56000.00	1
14	April Carl Mae	20	Fry Cook	67000.00	1
15	Mahad Nicle Nielsen	5	Shift Lead	90000.00	1

```sql
--Select query for price of meals less than 7 dollars
SELECT *
FROM Meal
WHERE SalesPrice < 7.00
ORDER BY SalesPrice

-- Select query
SELECT Sum(UnitCost) AS OrderCost,S.SupplierName, S.SupplierID

FROM MealSupplier AS MS
INNER JOIN Supplier AS S ON MS.SupplierID = S.SupplierID
GROUP BY S.SupplierName, S.SupplierID
```

100 %

Results | Messages

	OrderCost	SupplierName	SupplierID
1	2150.30	Harvest Food Distributors	1
2	2241.66	Egg Products Co	2
3	1335.74	Fresh and Honest Foods	3
4	1176.91	Augason Farms	4
5	1166.27	Sevillo Fine Foods	5

Views

```sql
-- View 1: Display employee information without including their SSN or salary number because this is sensitive information.

CREATE VIEW [Nonsensitive Employee Information] AS
 SELECT EmployeeID,
        FirstName,
        MiddleName,
        LastName,
        StreetAddress,
        City,
        State,
        ZipCode,
        Title
 FROM Employees

-- Use View to display the employees who are Cashier's without being able to see their SSN or salary
SELECT *
FROM [Nonsensitive Employee Information]
WHERE Title = 'Cashier'
```

	EmployeeID	FirstName	MiddleName	LastName	StreetAddress	City	State	ZipCode	Title
1	1	Svend	William	Jensen	4452 Maple Drive	Dallas	TX	46283	Cashier
2	14	Phoebe	Anne	Bridgers	54 Punisher Street	San Diego	CA	90211	Cashier

Views (cont.)

```sql
-- View 2: Display the Profit for each meal (SalesPrice - UnitCost)

CREATE VIEW [MealProfit] AS
SELECT M.MealName,
       M.MealDescription,
       SUM (M.SalesPrice - MS.UnitCost) AS Profit
FROM Meal AS M
INNER JOIN MealSupplier as MS ON M.MealID = MS.MealID
GROUP BY M.MealName,
         M.MealDescription

-- Use View to display the meals that are most profitable to least profitable
SELECT *
FROM [MealProfit]
ORDER BY Profit DESC
```

Views (cont.)

	MealName	MealDescription	Profit
1	Buffalo Chicken	Crispy, hand-breaded chicken breast covered in our B...	-297.08
2	Shack Stack	Cheeseburger topped with a Shroom Burger with lettuc...	-419.81
3	Shroom Burger	(Vegetarian) Crisp-fried portobello mushroom filled with ...	-585.39
4	Hamburger	Griddled patty and your choice of toppings (contains mi...	-621.02
5	Bacon Cheeseburger	Cheeseburger topped with applewood-smoked bacon ...	-681.08
6	SmokeShack	Cheeseburger with applewood-smoked bacon, choppe...	-766.30
7	Grilled Cheese	American cheese melted in a grilled potato bun (contai...	-832.37
8	Cheeseburger	Griddled patty with cheese and your choice of toppings...	-964.91
9	Chicken Shack	Crispy chicken breast over lettuce, pickles, buttermilk h...	-1281.61
10	Shackburger	Cheeseburger with lettuce, tomato, ShackSauce (cont...	-1432.81

Stored Procedures

```
-- Stored Procedure 1: This SPROC will allow the company to add a new customer to the database


CREATE PROCEDURE AddNewCustomer
( @FirstName                VARCHAR (25),
  @MiddleName               VARCHAR (25),
  @LastName                 VARCHAR (25),
  @PIN                      CHAR (5),
  @StreetAddress            VARCHAR (35),
  @City                     VARCHAR (25),
  @State                    CHAR (2),
  @ZipCode                  VARCHAR (10)
  )
AS
BEGIN
INSERT INTO Customers (FirstName, MiddleName, LastName, PIN, StreetAddress, City, State, Zipcode )
VALUES ( @FirstName, @MiddleName, @LastName, @PIN, @StreetAddress, @City, @State, @ZipCode );
END

--Now we are ready to add a new customer to the database
EXECUTE AddNewCustomer 'Danny', 'David', 'Ostler', '12367', '3677 S 2455 E','Salt Lake City', 'UT', '84109'
```

Stored Procedures

```
--Stored Procedure 2: Create a SPROC that will allows the company to see SalesRevenue from an Order greatest to smallest

CREATE PROCEDURE SalesRevenue

AS
BEGIN
    SELECT O.OrderID,
           O.EmployeeID,
           O.CustomerID,
           O.OrderDate,
           O.OrderTime,
           SUM (OL.Price * OL. Quantity) AS SalesRevenue
    FROM Orders AS O
    INNER JOIN OrderLines AS OL ON O.OrderID = OL. OrderID
    GROUP BY
    O.OrderID,
    O.EmployeeID,
    O.CustomerID,
    O.OrderDate,
    O.OrderTime
    ORDER BY SalesRevenue DESC
END;
```

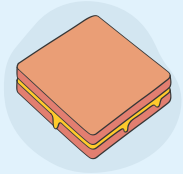
Stored Procedures

```
-- Stored Procedure 3: CREATE SPROC that will update an employee

CREATE PROCEDURE UpdateEmployee
( @EmployeeID       SMALLINT,
  @SSN              CHAR(11),
  @FirstName        VARCHAR(25),
  @MiddleName       VARCHAR(25),
  @LastName         VARCHAR(25),
  @StreetAddress    VARCHAR(35),
  @City             VARCHAR(25),
  @State            CHAR(2),
  @ZipCode          VARCHAR(10),
  @Title            VARCHAR(50),
  @Salary           DECIMAL(8,2)
)
AS
BEGIN
    UPDATE Employees
    SET SSN = @SSN,
        FirstName = @FirstName,
        MiddleName = @MiddleName,
        LastName = @LastName,
        StreetAddress = @StreetAddress,
        City = @City,
        State = @State,
        ZipCode = @ZipCode,
        Title = @Title,
        Salary = @Salary
    WHERE EmployeeID = @EmployeeID
END;

--Now we can use our SPROC to Update an employee record
EXECUTE UpdateEmployee
```

Answering Business Questions

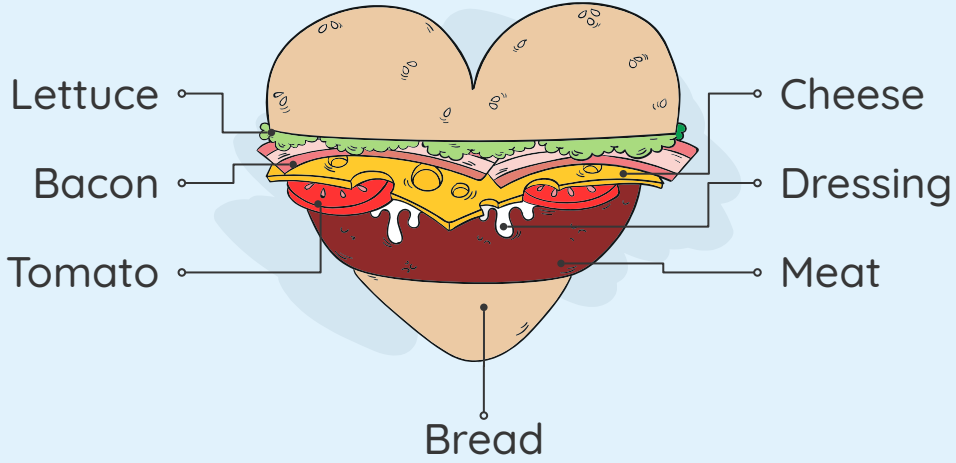


We used Select Queries, Views, and Stored Procedures to answer those business questions.

The Select Queries and Views allow us to generate useful reports for management.

Our stored procedures allow Shake Shack to quickly add new inputs to the database and help reduce user error moving forward.

Thank You!



Shake Shack

Stand For Something Good

ORDER NOW!