



THE DATA DYNAMOS GROUP

Prepared for
Enterprise Data
Management University
of Arizona

Adwait Minde, Ojas Pawar, Aswath Nambiar, Nimish Basu, Barathkannan Senthamaraichelvi
Ezhilarasan



TABLE OF CONTENTS

TABLE OF CONTENTS.....	2
Chapter 1: Requirements Analysis	3
Chapter: Conceptual Schema	5
Data Dictionary(Conceptual / for ER Modeling)	6
Chapter 3: Relational Schema	15
Relational Schema	15
Data Dictionary (Relational).....	19
DDL Appendix.....	30
Chapter 4: SQL Queries.....	38
Chapter 5: Triggers and Procedures	46
Chapter 6: User Interface.....	49
Chapter 7: Implementation Plan	55
Appendix A.....	56
References	56



Chapter 1: Requirements Analysis

In the context of McDonald's, an effective data management framework is crucial to efficiently handle diverse facets of its operations, spanning franchise management, supply chain coordination, marketing analysis, and customer engagement. As a global leader in the fast-food industry, McDonald's faces the complexity of managing a vast network of franchises, suppliers, and customers. To address this complexity, a robust database design is imperative to streamline operations and enhance decision-making.

At the core of McDonald's operations is franchise management. For instance, franchise owners, such as the owner of FranchiseID: 101, are responsible for daily operations, including sales and inventory management. These franchisees require real-time visibility into inventory levels (e.g., UnitsInStock, ReorderPoint) and employee schedules (e.g., ShiftTime, EmployeeID). A centralized system allows franchise owners to generate automated reports, such as DailySalesReports or ItemWiseRevenueAnalysis, ensuring efficient operations.

Similarly, supply chain coordinators play a pivotal role in ensuring the seamless flow of raw materials. For example, SupplierID: S001, representing "Fresh Produce Inc.," delivers goods to WarehouseID: W101. Detailed tracking of supplier performance metrics (e.g., DeliveryTimelines, QualityScores) ensures quality and reliability. SupplyDetails (e.g., QuantitySupplied, SupplyPrice) are systematically recorded, allowing real-time updates on inventory flow between warehouses and franchises.

Marketing analysts, such as those overseeing CampaignID: C001 for a new promotional offer, rely on insights derived from customer behavior. Attributes like CampaignROI, RegionalCustomerTrends, and CouponRedemptionRates enable analysts to design targeted marketing strategies. For instance, analytics derived from CustomerFeedback (e.g., FeedbackID: F123) can improve service quality and customer satisfaction.

In addition to internal operations, McDonald's engages with its customers indirectly. Customers, identified by attributes such as CustomerID and OrderHistory, provide valuable data that feeds into loyalty programs and personalized offers. These insights allow the organization to enhance customer retention and optimize service delivery.

System Challenges

The current system presents several challenges, such as:

Supply Chain Inefficiencies: Stockouts at 15% of franchises, such as those caused by delayed deliveries from suppliers like "Fresh Produce Inc." (SupplierID: S001), disrupt operations.

Franchise Operations: Manual payroll processing for EmployeeID: E101 results in inaccuracies and inefficiencies.

Customer Engagement: Fragmented feedback collection, with FeedbackID: F124 from mobile apps being underutilized, leads to missed improvement opportunities.

Marketing Campaign Management: Manual tracking of CouponID: CP567 delays real-time performance analysis, reducing CampaignROI.



Database Design Benefits:

To address these challenges, a well-designed database system offers:

Unified Data Management: A centralized repository consolidates data for all stakeholders. For example, Corporate Managers can view FranchisePerformanceMetrics (e.g., SalesVolume, CustomerSatisfactionScores) in real-time.

Real-Time Analytics: Automated alerts, such as StockoutAlerts for FranchiseID: 105, help address issues proactively.

Enhanced Customer Engagement: By analyzing CouponUsage patterns, campaigns can be optimized to meet customer needs.

Operational Efficiency: Automated payroll systems reduce FranchiseOwner workload by 80%.

Entity Relationships:

Suppliers (e.g., SupplierID: S001) deliver raw materials to Warehouses (WarehouseID: W101) and Franchises (FranchiseID: 101).

Franchises employ Employees (EmployeeID: E101) and serve Customers (CustomerID: C001).

Customers place Orders (OrderID: O123) and participate in Campaigns (CampaignID: C001).

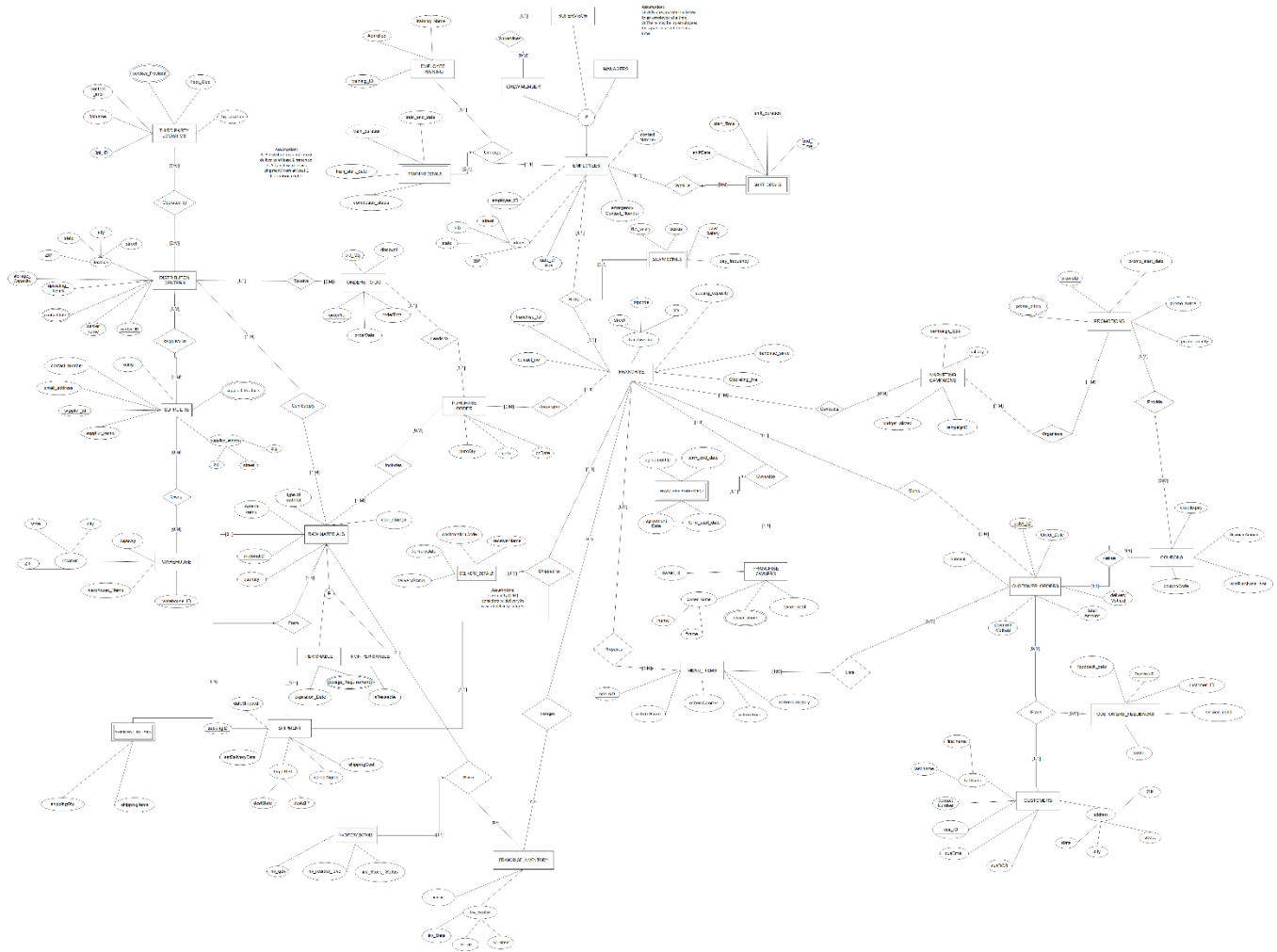
Scalability:

The system is designed to handle daily transactions for over 38,000 franchises, storing 10+ years of historical data for trend analysis. By implementing this database-driven approach, McDonald's will streamline operations, enhance decision-making, and align with its commitment to delivering consistent quality and exceptional service across its global network.



Chapter 2: Conceptual Schema

2A. ER Diagram





2B. Data Dictionary (Conceptual / for ER Modeling)

Data Dictionary

Schema Construct	Construct Meaning	Structure and/or Constraints
Entity Classes		
AVAIL_COUPONS		
promo_ID	Unique identifier for the promotion	Integer
coupon_code	Unique identifier for the coupon	Varchar
CONDUCTED_CAMPAIGNS		
franchise_id	Unique identifier for the franchise.	Integer
campaign_id	Unique identifier for the marketing campaign.	Integer
destState	The state where the shipment is to be delivered	String, NOT NULL
destZip	The zipcode of the address where the shipment is to be delivered	String, NOT NULL
SHIPMENT DETAILS		
shippingItems	The items which are to be shipped in that particular shipment	String, NOT NULL
shippingQty	The total quantity for that particular shipment	Integer, NOT NULL
COUPONS		
coupon_code	The alphanumeric code that customers need to enter or present to redeem the coupon	Primary Key
discountAmount	The monetary amount or percentage off that the coupon provides	Decimal, Not Null
coupExpiry	The date after which the coupon can no longer be used	Date, Not Null
minpurchase_amount	The minimum amount that must be spent to use the coupon	Decimal, Not Null
CREW MEMBER		
SUPERVISOR		
MANAGERS		
CUSTOMER FEEDBACKS		
feedback_id	A unique identifier for each customer feedback entry	Primary Key
cus_id	A foreign key linking to the customer who provided the feedback	Foreign Key
feedback_date	The date when the feedback was	Date, Not Null



	submitted	
rating	A numerical score given by the customer to rate their experience	Integer, Between 1 and 5
service_used	The type of service for which feedback is provided (e.g., dine-in, drive-thru)	String, Not Null
CUSTOMER_ORDERS		
order_id	A unique identifier for each customer order	Primary Key
order date	The date when the order was placed	Date, Not Null
total amount	The total cost of the order, including any taxes and fees	Decimal, Not Null
delivery method	The method of order delivery (e.g., dine-in, takeout, delivery)	String, Not Null
payment method	The method of payment used for the order (e.g., cash, credit card)	String, Not Null
CUSTOMERS		
cus_id	A unique identifier for each customer	Primary Key
cusemail	The email address of the customer for communication and marketing purposes	String, Not Null
cusDOB	The customer's date of birth, which may be used for promotions or age verification	Date
address	The residential address of the customer	String, Not Null
state	State which the customer is in	String, Not Null
city	City which the customer is in	String, Not Null
street	Street which the customer stays on	String, Not Null
ZIP	Zipcode of the address of the customer	Integer, Not Null
contact_number	Mobile number of the customer	Integer, NOT NULL
DELIVERY DETAILS		
delivery_date	The date on which the delivery was made	Date, NOT NULL
confirmation_code	A unique identifier for the delivery	PRIMARY KEY, NOT NULL
receiver_name	The name of the person receiving the delivery	String, NOT NULL
delivery_status	The status for the deliver	NOT NULL
DISTRIBUTION CENTERS		
center_ID	A unique identifier for each distribution center	Primary Key, NOT NULL



center_name	The name of the distribution center	NOT NULL
contact_info	Contact information for the center	Multivalued
operating_hours	The hours of operation for the center	NOT NULL
storage_capacity	The storage capacity of the center	Decimal, NOT NULL
location	The location of the center (state, city, zip, street)	NOT NULL
ZIP	Zipcode information for the address of the center	Integer, Not Null
state	The state which the center is located in	String, Not Null
city	The city which the center is located i	String, Not Null
street	The street which the center is located i	String, Not Null
EMPLOYEES		
employee_ID	A unique identifier for each employee in the franchise	Primary Key
address	The residential address of the employee	String, Not Null
contact_number	The primary phone number to reach the employee	String, Not Null
emergency_contact_number	A phone number for a person to contact in case of an emergency involving the employee	String
DateOfHire	The date when the employee was officially hired	Date, Not Null
street	Street the employee lives on	String, Not Null
city	City the employee lives in	String, Not Null
state	State the employee lives in	String, Not Null
zip	ZIP code for the employees address	Integer, Not Null
EMPLOYEE TRAINING		
training_ID	A unique identifier for each training program	Primary Key
number_enrolled	The number of employees enrolled in the training program	Integer
training_name	The name of the training program being offered to employees	String, Not Null
FRANCHISE		
franchise_id	A unique identifier for each McDonald's franchise location	Primary Key
contact_no	The primary phone number for the franchise location	String, Not Null



street	Street the franchise is located on	String Not Null
zipcode	Zipcode information for the franchise address	Integer Not Null
city	City the franchise is located in	String Not Null
operating_hrs	The hours during which the franchise is open for business	String
franchise_since	The time since the franchise was established and opened to the public	Date, Not Null
seating_capacity	The maximum number of customers that can be seated at one time in the franchise	Integer, Not Null
FRANCHISE_AGREEMENT		
term_end_date	The end date of the franchise agreement	Date, NOT NULL
term_start_date	The start date of the franchise agreement	Date, NOT NULL
agreementid	A unique identifier for each franchise agreement	Primary Key, NOT NULL
agreement_date	The date when the agreement was signed	Date, NOT NULL
FRANCHISE_INVENTORY		
Fran_inv_id	A unique identifier for each inventory item	Primary Key, NOT NULL
inv_state	The state where the inventory is located.	NOT NULL
Inv_zip	The zip code of the inventory's location	Integer, NOT NULL
Inv_street	The street of the inventory's location	Integer, NOT NULL
FRANCHISE_OWNERS		
owner_id	A unique identifier for each franchise owner	Primary Key
owner_name	The name of the individual or entity that owns the franchise	String, Not Null
lname	Last name of the franchise owner	String, Not Null
fname	First name of the franchise owner	String, Not Null
owner_mail	The email address of the franchise owner for communication purposes	String, Not Null
INVENTORY DETAILS		
inventory_id	Unique identifier for each inventory item.	Primary Key Not Null
product_name	Name of the product in the inventory.	String, Not Null



quantity_available	The number of units available in stock.	Not Null
restock_threshold	The minimum quantity of the product before restocking is required.	Not Null
price_per_unit	The cost of a single unit of the product.	Not Null
category	The category to which the product belongs	Varchar
MARKETING CAMPAIGNS		
campaignID	A unique identifier for each marketing campaign	Primary Key
validity	The end date of the marketing campaign, indicating how long it is active	Date, Not Null
budget_alloted	The total budget allocated for the marketing campaign	Decimal, Not Null
campaign_type	The type of campaign being conducted (e.g., digital, television, print)	String, Not Null
MENU ITEMS		
mltemID	A unique identifier for each item	Primary Key
mltemName	Name of the item	String, Not Null
mltemCalories	Calories in each item	Integer
mltemPrice	Cost of the menu item	Integer
mltemCategory	Category of the menu item	String, NOT NULL
NON-PERISHABLE		
recyclable	Whether the non-perishable item is recyclable or not	Boolean, NOT NULLs
PERISHABLE		
expiration_Date	The date for expiry of the perishable item	Date, NOT NULL
ORDERS TO DC		
orderNo	Unique number for the order from distribution center	Interger, NOT NULL
orderDate	Date on which the order to the distribution center was place	Date, NOT NULL
orderTotal	Total price for the order placed	Integer, NOT NULL
ord_Qty	Total quantity for the order placed	Integer, NOT NULL
discount	Discount applied on the order, if any	Integer
PROMOTIONS		
promoid	A unique identifier for each promotion	Primary Key
promo_name	The name of the promotion (e.g., "Buy One Get One Free")	String, Not Null



promo_validity	The end date of the promotional offer	Date, Not Null
promo_start_date	The date when the promotion begins	Date, Not Null
PURCHASE ORDER		
poNo0	Purchase order number	Primary Key
purcQty	Quantity of items purchase in order	Integer, NOT NULL
poDate	Date on which the purchase order was placed	
RAW MATERIALS		
material_id	A unique identifier for the raw material	Primary Key, NOT NULL
material_name	The name of the raw material	NOT NULL
quantity	The available quantity of the raw material	NOT NULL
type_of_material	The type/category of the raw material	NOT NULL
cold_storage	Specifies whether the material requires cold storage (Yes/No)	Boolean, NOT NULL
SALARY DETAILS		
bonus	The bonus amount for the employee	Decimal, NULL
base_salary	The base salary of the employee	Decimal, NOT NULL
pay_frequency	The frequency of salary payments (e.g., monthly, weekly)	NOT NULL
net_salary	The netsalary of the employee	Decimal, NOT NULL
SHIFT_DETAILS		
shiftDate	A unique identifier for each work shift	Primary Key
start time	The start time for the shift	String, Not Null
end time	The end time for the shift	String, Not Null
shift_duration	Calculated shift duration	Interger, Not Null
SHIPMENT DETAILS		
shippingItems	The items which are to be shipped in that particular shipment	String, NOT NULL
shippingQty	The total quantity for that particular shipment	Integer, NOT NULL
SHIPMENT		
trackingID	Unique ID from which each shipment is tracked	Integer, NOT NULL
dateShipped	The date when the item was shipped	Date, NOT NULL
estDeliveryDate	Estimated delivery date for the shipment	Date, NOT NULL
carrierName	Name of the carrier carrying the	String, NOT NULL



	shipment	
shippingCost	Total cost for the shipment	Integer, NOT NULL
destState	The state where the shipment is to be delivered	String, NOT NULL
destZip	The zipcode of the address where the shipment is to be delivered	String, NOT NULL
SUPPLIERS		
supplier_id	A unique identifier for each supplier providing products to McDonald's	Primary Key
supplier_name	The name of the company or individual supplying goods to the franchise	String, Not Null
contact_number	The primary phone number to reach the supplier	String, Not Null
email_address	The email address of the supplier for communication	String, Not Null
supplier_address	The physical address of the supplier	String, Not Null
rating	A numerical score representing the supplier's performance or quality	Decimal, Between 1.0 and 5.0
zip	The zipcode information for the address the supplier is located in	Integer, Not Null
street	The state which the supplier is located in	String, Not Null
city	The city the supplier is located in	String, Not Null
supplied_products	A description or list of products supplied by the supplier	String, Not Null
THIRD PARTY LOGISTICS		
3PL_ID	A unique identifier for each logistics partner	Primary Key, NOT NULL
3PLname	Name of the third-party logistics provider	String, NOT NULL
contact_info	Contact information of the logistics provider	Integer, NOT NULL
services_provided	Services offered by the provider	String, Multivalued
fleet_size	The number of vehicles in the provider's fleet	Integer, NOT NULL
hq_location	The headquarters location of the provider	String, NOT NULL
TRAINING DETAILS		
train_start_date	The start date of the training	Date, NOT NULL



train_end_date	The end date of the training	Date, NOT NULL
train_duration	The duration of the training in hours	NOT NULL
completion_status	Status indicating if the training was completed or not	NOT NULL
WAREHOUSE		
warehouse_id	A unique identifier for each warehouse location	Primary Key
warehouse_name	The name given to the warehouse for identification	String, Not Null
zip	The zipcode information for the warehouse address	Interger, Not Null
state	The state the warehouse is located in	String, Not Null
city	The city the warehouse is located in	String, Not Null
capacity	The total storage capacity of the warehouse, often measured in volume or weight	Integer, Not Null

Relationships:

Operated By	Relationship where Third party logistics are operated by distribution centers
Supplies To	Relationship where Distribution Centers supply materials to suppliers
Owns	Relationship where Suppliers Own Warehouses
Receive	Relationship where Distribution centers receive orders
Can supply	Relationship where DCs can supply raw materials
Form	Relationship between raw materials and shipment details
Undergo	Relationship where employees undergo training
Leads to	Relationship between orders to Dc and purchase orders
Hires	Relationship where Franchise hires employees
Generates	Relationship where franchise generates purchase orders
Work in	Relationship where employees work in shifts
Includes	Relationship where purchase orders includes raw materials
Shipped to	Relationship where shipments are shipped to franchises
Manages	Relationship where franchise manages inventory
Stores	Relationship where inventory stores raw materials
Owned by	Relationship between owners and franchises
Prepares	Relationship where franchises prepare menu items
lists	Relationship where order lists menu items



Conducts	Relationship where franchise conducts campaigns
Serve	Relationship where franchise serves orders
Organises	Relationship where marketing campaigns organise promotions
Provide	Relationship where promotions provide coupons
Validate	Relationship where orders validate coupons
Place	Relationship where customers place orders
Gives	Relationship where customers give feedbacks
Supervises	Relationship where supervisor supervises crew members



Chapter 3: Relational Schema

AVAIL_COUPONS(promo_id, coupon_code)

- FOREIGN KEY promo_id references PROMOTIONS
- FOREIGN KEY coupon_code references COUPONS

CAN_SUPPLY(center_ID, material_id)

- FOREIGN KEY center_ID references DISTRIBUTION CENTERS
- FOREIGN KEY item_code references RAW_MATERIALS

CONDUCTED_CAMPAIGNS(franchise_id, campaign_id)

- FOREIGN KEY franchise_id references FRANCHISE
- FOREIGN KEY campaign_id references MARKETING_CAMPAIGNS

COUPONS(coupon_code, minPurchase_amt, coupExpiry, discountAmount, order_ID)

- FOREIGN KEY order_ID references CUSTOMER_ORDERS

CUSTOMER_FEEDBACKS(feedbackID, feedback_date, rating, service_used, cus_ID)

- FOREIGN KEY cus_ID references CUSTOMERS

CUSTOMER_ORDERS(order_id, orderdate, Paymentmethod, deliverymethod, totalamount, franchise_id, customer_id, couponCode, subtotal)

- FOREIGN KEY franchise_id references FRANCHISE
- FOREIGN KEY customer_id references CUSTOMERS
- FOREIGN KEY couponCode_id references COUPONS

CUSTOMERS(cus_id, cusEmail, cusDOB, State, City, Street, ZIP, firstname, lastname, contactNumber)

DELIVERY_DETAILS(Franchise_ID, trackingID, confirmationCode, DeliveryDate, DeliveryStatus, ReceiverName)

- FOREIGN KEY Franchise_ID references FRANCHISE
- FOREIGN KEY trackingID references SHIPMENT

DISTRIBUTION_CENTERS(center_ID, center_name, operating_hours, storage_capacity, zip, state, city, street)

DISTRIBUTION_CENTERS_CONTACT(center_ID, contactinfo)

- FOREIGN KEY center_ID references DISTRIBUTION CENTERS



EMPLOYEES(employee_ID, contact_number, emergency_contact_number, DateOfHire, state, city, zip, street, franchise_ID)

- FOREIGN KEY franchise_ID references FRANCHISE
- CREW_MEMBER**(employee_ID, supervisor_ID)
- FOREIGN KEY employee_ID references EMPLOYEES
 - FOREIGN KEY supervisor_ID references SUPERVISOR(employee_ID)
- MANAGERS**(employee_ID)
- FOREIGN KEY employee_ID references EMPLOYEES(employee_ID)
- SUPERVISOR**(employee_ID)
- FOREIGN KEY employee_ID references EMPLOYEES(employee_ID)

EMPLOYEE_TRAINING(training_ID, number_enrolled, training_name)

FEEDBACK_INFO(cus_id, order_id, feedbackID)

- FOREIGN KEY cus_id references CUSTOMERS
- FOREIGN KEY order_id references CUSTOMER_ORDERS
- FOREIGN KEY feedback_id references CUSTOMER_FEEDBACKS

FRANCHISE(Franchise_ID, Contact_no, street, zipcode, city, seating_capacity, franchise_since, operating_hrs)

FRANCHISE_AGREEMENT(franchise_id, agreement_id, agreementdate, term_start_date, term_end_date)

- FOREIGN KEY franchise_id references FRANCHISE

FRANCHISE_INVENTORY(fran_inv_id, inv_state, inv_zip, inv_street, franchise_id)

- FOREIGN KEY franchise_id references FRANCHISE

FRANCHISEOWNERS(owner_id, lname, fname, owner_mail)

FRANCHISEOWNER_PHONES(owner_id, owner_phone)

- FOREIGN KEY owner_id references FRANCHISEOWNER

INVENTORY_DETAILS(material_id, inv_id, inv_qoh, inv_reorder_level, inv_stock_status)

- FOREIGN KEY material_id references RAW_MATERIALS
- FOREIGN KEY inv_id references FRANCHISE_INVENTORY

LISTS(order_id, mitemID)

- FOREIGN KEY order_id references CUSTOMERS
- FOREIGN KEY mItemID references MENU_ITEMS



MARKETING_CAMPAIGNS(campaignID, validity, campaign_type, budget_alloted)

MENU_ITEMS(mItemID, mItemName, mitemCalories, mItemPrice, mItemCategory)

OPERATED_BY(3PL_ID, center_ID)

- FOREIGN KEY 3PL_ID references THIRD PARTY LOGISTICS
- FOREIGN KEY center_ID references DISTRIBUTION CENTERS

ORDERS_TO_DC(orderNo, orderDate, orderTotal, ordQty, discount, center_ID, PONo)

- FOREIGN KEY center_ID references DISTRIBUTION CENTERS
- FOREIGN KEY PONo references PURCHASE_ORDER

ORGANISED_PROMO(campaign_id, promo_id)

- FOREIGN KEY campaign_id references MARKETING_CAMPAIGNS
- FOREIGN KEY promo_id references PROMOTIONS

OWNS(supplier_ID, warehouse_id, material_ID)

- FOREIGN KEY supplier_ID references SUPPLIERS
- FOREIGN KEY warehouse_id references WAREHOUSE
- FOREIGN KEY material_ID references RAW_MATERIALS

PREPARES(franchiseID, mItemID)

- FOREIGN KEY franchiseID references FRANCHISE
- FOREIGN KEY mItemID references MENU_ITEMS

PROMOTIONS(promoID, promo_name, promo_validity, promo_start_date)

PURCHASE_ORDER(PONo, PODate, PurcQty, material_id)

- FOREIGN KEY material_id references RAW MATERIALS

PROMO_OFFERS(promoID, promo_offers)

- FOREIGN KEY promoID references PROMOTIONS

RAW_MATERIALS(material_Id, materialName, typeofMaterial, cold_storage, quantity)

NON_PERISHABLE(material_id, recyclable)

- FOREIGN KEY material_id references RAW_MATERIALS

PERISHABLE(material_id, expirationDate)

- FOREIGN KEY material_id references RAW_MATERIALS(material_id)

PERISHABLE_STORAGE(material_id, storage_requirements)

- FOREIGN KEY material_id references RAW_MATERIALS(material_id)



SALARY_DETAILS(employee_ID, franchise_ID, bonus, base_Salary, pay_frequency, net_salary)

- FOREIGN KEY employee_ID references EMPLOYEES
- FOREIGN KEY franchise_ID references FRANCHISE

SHIFTS_DETAILS(Employee_ID, shiftDate, startTime, endTime, shiftDuration)

SHIPMENT_DETAILS(material_id, trackingID, ShippingItems, ShippingQty)

- FOREIGN KEY material_id references RAW MATERIALS
- FOREIGN KEY trackingID references SHIPMENT

SHIPMENT(trackingID, DateShipped, EstDeliveryDate, DestState, DestZIP, carrirerName, ShippingCost)

SUPPLIERS(supplier_ID, supplier_name, email_address, zip, street, city, contact_number, rating)

SUPPLIERS_PRODUCTS(supplier_ID, supplied_Products)

- FOREIGN KEY supplier_ID references SUPPLIERS

THIRD_PARTY_LOGISTICS(3PL_ID, 3PLname, contactinfo, hq_location, fleetsize)

THIRD_PARTY_LOGISTICS_SERVICES(3PL_ID, services_provided)

- FOREIGN KEY 3PL_ID references THIRD PARTY LOGISTICS

TRAINING_DETAILS(training_ID, employee_ID, train_start_date, train_end_date, train_duration, completion_status)

- FOREIGN KEY employee_ID references EMPLOYEES
- FOREIGN KEY training_ID references EMPLOYEE_TRAINING

WAREHOUSE(warehouse_id, warehouse_name, zip, state, city, capacity)



Data Dictionary

AVAIL_COUPONS Table

- **promo_id**: VARCHAR2(16), NOT NULL, CHECK (promo_id LIKE 'PR%')
- **coupon_code**: VARCHAR2(50), NOT NULL
- **CONSTRAINT fk_promo_id FOREIGN KEY (promo_id) REFERENCES PROMOTIONS(promoID)**
- **CONSTRAINT fk_coupon_code FOREIGN KEY (coupon_code) REFERENCES COUPONS(coupon_code)**

CAN_SUPPLY Table

- **center_ID**: VARCHAR2(16), NOT NULL
- **Material_Id**: VARCHAR2(16), NOT NULL
- **PRIMARY KEY (center_ID, Material_Id)**
- **FOREIGN KEY (center_ID) REFERENCES DISTRIBUTION_CENTERS(center_ID)**
- **FOREIGN KEY (Material_Id) REFERENCES RAW_MATERIALS(material_Id)**

CONDUCTED_CAMPAIGNS Table

- **franchise_id**: VARCHAR2(16), NOT NULL, CHECK (franchise_id LIKE 'FA%')
- **campaign_id**: VARCHAR2(16), NOT NULL, CHECK (campaign_id LIKE 'MC%')
- **CONSTRAINT fk_franchise_id FOREIGN KEY (franchise_id) REFERENCES FRANCHISE(Franchise_ID)**
- **CONSTRAINT fk_campaign_id FOREIGN KEY (campaign_id) REFERENCES MARKETING_CAMPAIGNS(campaignID)**

COUPONS Table

- **coupon_code**: VARCHAR2(50), PRIMARY KEY
- **minPurchase_amt**: NUMBER, NOT NULL, CHECK (minPurchase_amt >= 0)
- **coupExpiry**: DATE, NOT NULL
- **coupDiscountAmt**: NUMBER, NOT NULL, CHECK (coupDiscountAmt >= 0)

CREW_MEMBER Table

- **employee_ID**: VARCHAR2(16), NOT NULL
- **supervisor_ID**: VARCHAR2(16)
- **PRIMARY KEY (employee_ID)**
- **FOREIGN KEY (employee_ID) REFERENCES EMPLOYEES(employee_ID)**
- **FOREIGN KEY (supervisor_ID) REFERENCES SUPERVISOR(employee_ID)**



CUSTOMER_FEEDBACKS Table

- **feedbackID:** VARCHAR2(16), PRIMARY KEY, CHECK (feedbackID LIKE 'FB%')
- **feedback_date:** DATE, NOT NULL
- **rating:** NUMBER, NOT NULL, CHECK (rating BETWEEN 1 AND 5)
- **service_used:** VARCHAR2(50), NOT NULL

CUSTOMER_ORDERS Table

- **order_id:** VARCHAR2(16), PRIMARY KEY, CHECK (order_id LIKE 'CO%')
- **orderdate:** DATE, NOT NULL
- **Paymentmethod:** VARCHAR2(50), NOT NULL
- **deliverymethod:** VARCHAR2(50), NOT NULL
- **totalamount:** NUMBER, CHECK (totalamount >= 0)
- **subtotal:** NUMBER, NOT NULL, CHECK (subtotal >= 0)
- **franchise_id:** VARCHAR2(16), NOT NULL
- **coupon_code:** VARCHAR2(50)
- **cus_id:** VARCHAR2(16), NOT NULL
- **CONSTRAINT fk_customer_orders_franchise_id FOREIGN KEY (franchise_id) REFERENCES FRANCHISE(Franchise_ID)**
- **CONSTRAINT fk_customer_orders_cus_id FOREIGN KEY (cus_id) REFERENCES MCDCUSTOMERS(cus_id)**
- **CONSTRAINT fk_customer_orders_coupon_code FOREIGN KEY (coupon_code) REFERENCES COUPONS(coupon_code)**

CUSTOMERS Table

- **cus_id:** VARCHAR2(16), PRIMARY KEY, CHECK (cus_id LIKE 'MCD%')
- **cusEmail:** VARCHAR2(100), NOT NULL, UNIQUE
- **cusDOB:** DATE, NOT NULL
- **state:** VARCHAR2(50), NOT NULL
- **city:** VARCHAR2(50), NOT NULL
- **street:** VARCHAR2(100), NOT NULL
- **zip:** VARCHAR2(10), NOT NULL
- **firstname:** VARCHAR2(50), NOT NULL
- **lastname:** VARCHAR2(50), NOT NULL



- **contactNumber:** VARCHAR2(15), NOT NULL

DELIVERY_DETAILS Table

- **Franchise_ID:** VARCHAR2(16), NOT NULL
- **trackingID:** VARCHAR2(16), NOT NULL
- **confirmationCode:** NUMBER, NOT NULL
- **DeliveryDate:** DATE
- **DeliveryStatus:** VARCHAR2(50)
- **ReceiverName:** VARCHAR2(100)
- **PRIMARY KEY (Franchise_ID, trackingID, confirmationCode)**
- **FOREIGN KEY (Franchise_ID) REFERENCES FRANCHISE(Franchise_ID)**
- **FOREIGN KEY (trackingID) REFERENCES SHIPMENT(trackingID)**

DISTRIBUTION_CENTERS

- **center_ID:** VARCHAR2(16), PRIMARY KEY, CHECK (center_ID LIKE 'DC%')
- **center_name:** VARCHAR2(100), NOT NULL
- **operating_hours:** VARCHAR2(50), NOT NULL
- **storage_capacity:** NUMBER, NOT NULL, CHECK (storage_capacity >= 0)
- **zip:** VARCHAR2(10), NOT NULL
- **state:** VARCHAR2(50), NOT NULL
- **city:** VARCHAR2(50), NOT NULL
- **street:** VARCHAR2(100), NOT NULL

DISTRIBUTION_CENTERS_CONTACT

- **center_ID:** VARCHAR2(16), PRIMARY KEY, CHECK (center_ID LIKE 'DC%')
- **contactinfo:** VARCHAR2(100), NOT NULL
- **FOREIGN KEY REFERENCES DISTRIBUTION_CENTERS(center_ID)**

EMPLOYEES Table

- **employee_ID:** VARCHAR2(16), PRIMARY KEY, CHECK (employee_ID LIKE 'EMP%')
- **contact_number:** VARCHAR2(15), NOT NULL
- **emergency_contact_number:** VARCHAR2(15), NOT NULL
- **DateOfHire:** DATE, NOT NULL
- **state:** VARCHAR2(50), NOT NULL



- **city:** VARCHAR2(50), NOT NULL
- **zip:** VARCHAR2(10), NOT NULL
- **street:** VARCHAR2(100), NOT NULL
- **franchise_ID:** VARCHAR2(16), NOT NULL, FOREIGN KEY REFERENCES FRANCHISE(franchise_id)

EMPLOYEE_TRAINING Table

- **training_ID:** VARCHAR2(16), PRIMARY KEY, CHECK (training_ID LIKE 'ETRN%')
- **number_enrolled:** NUMBER, NOT NULL, CHECK (number_enrolled >= 0)
- **training_name:** VARCHAR2(100), NOT NULL

ESHIFTS_DETAILS Table

- **Employee_ID:** VARCHAR2(16), NOT NULL, CHECK (Employee_ID LIKE 'EM%')
- **shiftDate:** DATE, NOT NULL
- **startTime:** TIMESTAMP, NOT NULL
- **endTime:** TIMESTAMP, NOT NULL
- **shiftDuration:** NUMBER, NOT NULL, CHECK (shiftDuration >= 0)
- **PRIMARY KEY (Employee_ID, shiftDate)**
- **CONSTRAINT fk_eshifts_employee_ID FOREIGN KEY (Employee_ID) REFERENCES EMPLOYEES(employee_ID)**

FEEDBACK_INFO Table

- **cus_id:** VARCHAR2(16), NOT NULL, CHECK (cus_id LIKE 'MCDC%')
- **order_id:** VARCHAR2(16), NOT NULL, CHECK (order_id LIKE 'CO%')
- **feedbackID:** VARCHAR2(16), NOT NULL, CHECK (feedbackID LIKE 'FB%')
- **CONSTRAINT fk_feedback_info_cus_id FOREIGN KEY (cus_id) REFERENCES MCDCUSTOMERS(cus_id)**
- **CONSTRAINT fk_feedback_info_order_id FOREIGN KEY (order_id) REFERENCES CUSTOMER_ORDERS(order_id)**
- **CONSTRAINT fk_feedback_info_feedback_id FOREIGN KEY (feedbackID) REFERENCES CUSTOMER_FEEDBACKS(feedbackID)**

FRANCHISE Table

- **franchise_id:** VARCHAR2(16), PRIMARY KEY, CHECK (franchise_id LIKE 'FA%')



- **contact_no:** VARCHAR2(15), NOT NULL
- **street:** VARCHAR2(100), NOT NULL
- **zipcode:** VARCHAR2(10), NOT NULL
- **city:** VARCHAR2(50), NOT NULL
- **seating_capacity:** NUMBER, NOT NULL
- **franchise_since:** DATE, NOT NULL
- **operating_hrs:** VARCHAR2(50), NOT NULL
- **owner_id:** VARCHAR2(16), NOT NULL, FOREIGN KEY REFERENCES FRANCHISEOWNERS(owner_id)

FRANCHISE_AGREEMENT

- **franchise_id:** VARCHAR2(16), NOT NULL, FOREIGN KEY REFERENCES FRANCHISE(franchise_id)
- **agreement_id:** VARCHAR2(16), PRIMARY KEY, CHECK (agreement_id LIKE 'FAGR%')
- **agreementdate:** DATE, NOT NULL
- **term_start_date:** DATE, NOT NULL
- **term_end_date:** DATE, NOT NULL

FRANCHISE_INVENTORY

- **inv_Id:** VARCHAR2(16), PRIMARY KEY, CHECK (inv_Id LIKE 'FINV%')
- **inv_state:** VARCHAR2(16), NOT NULL
- **inv_zip:** VARCHAR2(16), NOT NULL
- **inv_street:** VARCHAR2(16), NOT NULL
- **franchise_id:** VARCHAR2(16), NOT NULL, FOREIGN KEY REFERENCES FRANCHISE(franchise_id)

FRANCHISEOWNERS Table

- **owner_id:** VARCHAR2(16), PRIMARY KEY, CHECK (owner_id LIKE 'FO%')
- **lname:** VARCHAR2(50), NOT NULL
- **fname:** VARCHAR2(50), NOT NULL
- **owner_mail:** VARCHAR2(100), NOT NULL, UNIQUE

FRANCHISEOWNER_PHONES Table

- **owner_id:** VARCHAR2(16), NOT NULL



- **owner_phone:** VARCHAR2(15)
- **PRIMARY KEY (owner_id)**
- **FOREIGN KEY (owner_id) REFERENCES FRANCHISEOWNERS(owner_id)**

INVENTORY_DETAILS Table

- **Material_Id:** VARCHAR2(16)
- **inv_Id:** VARCHAR2(16)
- **inv_qoh:** NUMBER
- **inv_reorder_level:** NUMBER
- **inv_stock_status:** VARCHAR2(50)
- **CONSTRAINT fk_inv_details_material_id FOREIGN KEY (Material_Id) REFERENCES RAW_MATERIALS(material_id)**
- **CONSTRAINT fk_inv_details_inv_id FOREIGN KEY (inv_Id) REFERENCES FRANCHISE_INVENTORY(inv_Id)**

LISTS Table

- **order_id:** VARCHAR2(16), NOT NULL
- **mitemID:** VARCHAR2(16), NOT NULL
- **PRIMARY KEY (order_id, mitemID)**
- **CONSTRAINT fk_lists_order_id FOREIGN KEY (order_id) REFERENCES CUSTOMER_ORDERS(order_id)**
- **CONSTRAINT fk_lists_mitemID FOREIGN KEY (mitemID) REFERENCES MENU_ITEMS(mItemID)**

MANAGERS

- **employee_ID:** VARCHAR2(16), PRIMARY KEY, CHECK (employee_ID LIKE 'EMP%')
- **FOREIGN KEY REFERENCES EMPLOYEES(employee_ID)**

MARKETING_CAMPAIGNS Table

- **campaignID:** VARCHAR2(16), PRIMARY KEY, CHECK (campaignID LIKE 'MC%')
- **validity:** DATE, NOT NULL
- **campaign_type:** VARCHAR2(50), NOT NULL
- **budget_alloted:** NUMBER, NOT NULL, CHECK (budget_alloted >= 0)

MENU_ITEMS Table

- **mItemID:** VARCHAR2(16), PRIMARY KEY, CHECK (mItemID LIKE 'MI%')
- **mItemName:** VARCHAR2(100), NOT NULL



- **mitemCalories:** NUMBER, NOT NULL, CHECK (mitemCalories >= 0)
- **mItemPrice:** NUMBER, NOT NULL, CHECK (mItemPrice >= 0)
- **mItemCategory:** VARCHAR2(50), NOT NULL

NON_PERISHABLE Table

- **material_id:** VARCHAR2(16), PRIMARY KEY, FOREIGN KEY REFERENCES RAW_MATERIALS(material_id)
- **recyclable:** VARCHAR2(3), CHECK (recyclable IN ('YES', 'NO'))

OPERATED_BY

- **TPL_ID:** VARCHAR2(16), NOT NULL, CHECK (TPL_ID LIKE 'TPL%')
- **center_ID:** VARCHAR2(16), NOT NULL, CHECK (center_ID LIKE 'DC%')
- PRIMARY KEY (TPL_ID, center_ID)
- FOREIGN KEY (TPL_ID) REFERENCES THIRD_PARTY_LOGISTICS(TPL_ID)
- FOREIGN KEY (center_ID) REFERENCES DISTRIBUTION_CENTERS(center_ID)

ORDERS_TO_DC Table

- **orderNo:** VARCHAR2(16), NOT NULL, CHECK (orderNo LIKE 'OTD%')
- **orderDate:** DATE, NOT NULL
- **orderTotal:** NUMBER, NOT NULL
- **ordQty:** NUMBER, NOT NULL
- **discount:** VARCHAR2(16)
- **center_ID:** VARCHAR2(16)
- **PONo:** VARCHAR2(16)
- **CONSTRAINT fk_orders_to_dc_center_ID FOREIGN KEY (center_ID) REFERENCES DISTRIBUTION_CENTERS(center_ID)**
- **CONSTRAINT fk_orders_to_dc_PONo FOREIGN KEY (PONo) REFERENCES PURCHASE_ORDER(PONo)**

ORGANISED_PROMO Table

- **campaign_id:** VARCHAR2(16), NOT NULL, CHECK (campaign_id LIKE 'MC%')
- **promo_id:** VARCHAR2(16), NOT NULL, CHECK (promo_id LIKE 'PR%')
- **PRIMARY KEY (campaign_id, promo_id)**
- **CONSTRAINT fk_organised_promo_campaign_id FOREIGN KEY (campaign_id) REFERENCES MARKETING_CAMPAIGNS(campaignID)**



- **CONSTRAINT fk_organised_promo_promo_id FOREIGN KEY (promo_id) REFERENCES PROMOTIONS(promoID)**

OWNS Table

- **supplier_ID: VARCHAR2(16)**
- **warehouse_id: VARCHAR2(16)**
- **material_ID: VARCHAR2(16)**
- **CONSTRAINT fk_owns_supplier_ID FOREIGN KEY (supplier_ID) REFERENCES SUPPLIERS(supplier_ID)**
- **CONSTRAINT fk_owns_warehouse_id FOREIGN KEY (warehouse_id) REFERENCES WAREHOUSE(warehouse_id)**
- **CONSTRAINT fk_owns_material_ID FOREIGN KEY (material_ID) REFERENCES RAW_MATERIALS(material_ID)**
- **PRIMARY KEY (supplier_ID, warehouse_id, material_ID)**

PERISHABLE Table

- **material_id: VARCHAR2(16), PRIMARY KEY, FOREIGN KEY REFERENCES RAW_MATERIALS(material_id)**
- **expirationDate: DATE, NOT NULL**

PERISHABLE_STORAGE Table

- **material_id: VARCHAR2(16), PRIMARY KEY, FOREIGN KEY REFERENCES RAW_MATERIALS(material_id)**
- **storage_requirements: VARCHAR2(255), NOT NULL**

PREPARES Table

- **franchiseID: VARCHAR2(16), NOT NULL, CHECK (franchiseID LIKE 'FA%')**
- **mItemID: VARCHAR2(16), NOT NULL, CHECK (mItemID LIKE 'MI%')**
- **PRIMARY KEY (franchiseID, mItemID)**
- **CONSTRAINT fk_prepares_franchiseID FOREIGN KEY (franchiseID) REFERENCES FRANCHISE(Franchise_ID)**
- **CONSTRAINT fk_prepares_mItemID FOREIGN KEY (mItemID) REFERENCES MENU_ITEMS(mItemID)**

PROMO_OFFERS Table

- **promoID: VARCHAR2(16), PRIMARY KEY, CHECK (promoID LIKE 'PR%')**
- **promo_offers: VARCHAR2(255), NOT NULL**
- **CONSTRAINT fk_promo_offers_promoID FOREIGN KEY (promoID) REFERENCES PROMOTIONS(promoID)**



PROMOTIONS Table

- **promoID:** VARCHAR2(16), PRIMARY KEY, CHECK (promoID LIKE 'PR%')
- **promo_name:** VARCHAR2(100), NOT NULL
- **promo_validity:** DATE, NOT NULL
- **promo_start_date:** DATE, NOT NULL

PURCHASE_ORDER Table

- **PONo:** VARCHAR2(16), PRIMARY KEY, CHECK (PONo LIKE 'PO%')
- **PODate:** DATE, NOT NULL
- **PurcQty:** NUMBER, NOT NULL
- **Material_Id:** VARCHAR2(16), FOREIGN KEY REFERENCES RAW_MATERIALS(material_Id)

RAW_MATERIALS Table

- **material_Id:** VARCHAR2(16), PRIMARY KEY, CHECK (material_Id LIKE 'RM%')
- **materialName:** VARCHAR2(100), NOT NULL
- **typeofMaterial:** VARCHAR2(50), NOT NULL
- **cold_storage:** CHAR(1), CHECK (cold_storage IN ('Y', 'N'))
- **quantity:** NUMBER, NOT NULL

SALARY_DETAILS

- **employee_ID:** VARCHAR2(16), NOT NULL, CHECK (employee_ID LIKE 'EMP%')
- **franchise_ID:** VARCHAR2(16), NOT NULL, CHECK (franchise_ID LIKE 'FA%')
- **bonus:** NUMBER, NOT NULL, CHECK (bonus >= 0)
- **base_Salary:** NUMBER, NOT NULL, CHECK (base_Salary >= 0)
- **pay_frequency:** VARCHAR2(50), NOT NULL
- PRIMARY KEY (employee_ID, franchise_ID)
- FOREIGN KEY (employee_ID) REFERENCES EMPLOYEES(employee_ID)
- FOREIGN KEY (franchise_ID) REFERENCES FRANCHISE(franchise_id)

SHIPMENT Table

- **trackingID:** VARCHAR2(17), PRIMARY KEY, CHECK (trackingID LIKE 'SPM%')
- **DateShipped:** DATE, NOT NULL
- **EstDeliveryDate:** DATE, NOT NULL
- **DestState:** VARCHAR2(50), NOT NULL



- **DestZIP:** NUMBER, NOT NULL
- **carrierName:** VARCHAR2(100), NOT NULL
- **ShippingCost:** NUMBER, NOT NULL

SHIPMENT_DETAILS Table

- **Material_Id:** VARCHAR2(16)
- **trackingID:** VARCHAR2(16)
- **CONSTRAINT fk_shipment_details_Material_Id FOREIGN KEY (Material_Id) REFERENCES RAW_MATERIALS(material_Id)**
- **CONSTRAINT fk_shipment_details_trackingID FOREIGN KEY (trackingID) REFERENCES SHIPMENT(trackingID)**
- **PRIMARY KEY (Material_Id, trackingID)**

SUPERVISOR

- **employee_ID:** VARCHAR2(16), PRIMARY KEY, CHECK (employee_ID LIKE 'EMP%')
- **FOREIGN KEY REFERENCES EMPLOYEES(employee_ID)**

SUPPLIERS Table

- **supplier_id:** VARCHAR2(16), PRIMARY KEY, CHECK (supplier_id LIKE 'SU%')
- **supplier_name:** VARCHAR2(100), NOT NULL
- **email_address:** VARCHAR2(100), NOT NULL, UNIQUE
- **zip:** NUMBER(5), NOT NULL
- **street:** VARCHAR2(255), NOT NULL
- **city:** VARCHAR2(100), NOT NULL
- **contact_number:** VARCHAR2(15), NOT NULL
- **rating:** NUMBER, NOT NULL, CHECK (rating BETWEEN 0 AND 5)

SUPPLIERS_PRODUCTS Table

- **supplier_ID:** VARCHAR2(16), PRIMARY KEY, FOREIGN KEY REFERENCES SUPPLIERS(supplier_id)
- **supplied_Products:** VARCHAR2(255), NOT NULL

THIRD_PARTY_LOGISTICS

- **TPL_ID:** VARCHAR2(16), PRIMARY KEY, CHECK (TPL_ID LIKE 'TPL%')
- **TPLname:** VARCHAR2(100), NOT NULL
- **contactinfo:** VARCHAR2(255), NOT NULL



- **hq_location:** VARCHAR2(255), NOT NULL
- **fleetsize:** NUMBER, NOT NULL, CHECK (fleetsize >= 0)

THIRD_PARTY_LOGISTICS_SERVICES

- **TPL_ID:** VARCHAR2(16), PRIMARY KEY, CHECK (TPL_ID LIKE 'TPL%')
- **services_provided:** VARCHAR2(255), NOT NULL
- FOREIGN KEY REFERENCES THIRD_PARTY_LOGISTICS(TPL_ID)

TRAINING_DETAILS Table

- **training_ID:** VARCHAR2(16), NOT NULL
- **employee_ID:** VARCHAR2(16), NOT NULL
- **train_start_date:** DATE, NOT NULL
- **train_end_date:** DATE, NOT NULL
- **train_duration:** NUMBER, NOT NULL, CHECK (train_duration >= 0)
- **completion_status:** VARCHAR2(50), NOT NULL
- **PRIMARY KEY (employee_ID, training_ID)**
- **CONSTRAINT fk_training_details_employee_ID FOREIGN KEY (employee_ID) REFERENCES EMPLOYEES(employee_ID)**
- **CONSTRAINT fk_training_details_training_ID FOREIGN KEY (training_ID) REFERENCES EMPLOYEE_TRAINING(training_ID)**

WAREHOUSE Table

- **warehouse_id:** VARCHAR2(16), PRIMARY KEY, CHECK (warehouse_id LIKE 'WA%')
- **warehouse_name:** VARCHAR2(100), NOT NULL
- **zip:** NUMBER(5), NOT NULL
- **state:** VARCHAR2(50), NOT NULL
- **city:** VARCHAR2(100), NOT NULL
- **capacity:** NUMBER, NOT NULL



DDL Appendix

```
CREATE TABLE WAREHOUSE(  
warehouse_id VARCHAR2(16) PRIMARY KEY CHECK (warehouse_id LIKE 'WA%'),  
warehouse_name VARCHAR2(100) NOT NULL,  
zip NUMBER(5) NOT NULL,  
state VARCHAR2(50) NOT NULL,  
city VARCHAR2(100) NOT NULL,  
capacity NUMBER NOT NULL );
```

```
CREATE TABLE MDCCUSTOMERS(  
cus_id VARCHAR2(16) PRIMARY KEY CHECK (cus_id LIKE 'MDCD%'),  
cusEmail VARCHAR2(100) NOT NULL UNIQUE,  
cusDOB DATE NOT NULL,  
state VARCHAR2(50) NOT NULL,  
city VARCHAR2(50) NOT NULL,  
street VARCHAR2(100) NOT NULL,  
zip VARCHAR2(10) NOT NULL,  
firstname VARCHAR2(50) NOT NULL,  
lastname VARCHAR2(50) NOT NULL,  
contactNumber VARCHAR2(15) NOT NULL );
```

```
CREATE TABLE FRANCHISEOWNERS(  
owner_id VARCHAR2(16) PRIMARY KEY CHECK (owner_id LIKE 'FO%'),  
lname VARCHAR2(50) NOT NULL,  
fname VARCHAR2(50) NOT NULL,  
owner_mail VARCHAR2(100) NOT NULL UNIQUE );
```

```
CREATE TABLE FRANCHISE(  
franchise_id VARCHAR2(16) PRIMARY KEY CHECK (franchise_id LIKE 'FA%'),  
contact_no VARCHAR2(15) NOT NULL,  
street VARCHAR2(100) NOT NULL,  
zipcode VARCHAR2(10) NOT NULL,  
city VARCHAR2(50) NOT NULL,  
seating_capacity NUMBER NOT NULL,  
franchise_since DATE NOT NULL,  
operating_hrs VARCHAR2(50) NOT NULL,  
owner_id VARCHAR2(16) NOT NULL,  
CONSTRAINT fk_franchise_owner_id FOREIGN KEY (owner_id) REFERENCES  
FRANCHISEOWNERS(owner_id) );
```

```
CREATE TABLE SUPPLIERS(  
supplier_id VARCHAR2(16) PRIMARY KEY CHECK (supplier_id LIKE 'SU%'),  
supplier_name VARCHAR2(100) NOT NULL,  
email_address VARCHAR2(100) NOT NULL UNIQUE,  
zip NUMBER(5) NOT NULL,  
street VARCHAR2(255) NOT NULL,  
city VARCHAR2(100) NOT NULL,  
contact_number VARCHAR2(15) NOT NULL,  
rating NUMBER CHECK (rating BETWEEN 0 AND 5) NOT NULL );
```

```
CREATE TABLE EMPLOYEE_TRAINING(  
training_ID VARCHAR2(16) PRIMARY KEY CHECK (training_ID LIKE 'ETRN%'),  
number_enrolled NUMBER NOT NULL CHECK (number_enrolled >= 0),  
training_name VARCHAR2(100) NOT NULL );
```



```
CREATE TABLE EMPLOYEES(  
employee_ID VARCHAR2(16) PRIMARY KEY CHECK (employee_ID LIKE 'EMP%'),  
contact_number VARCHAR2(15) NOT NULL,  
emergency_contact_number VARCHAR2(15) NOT NULL,  
DateOfHire DATE NOT NULL,  
state VARCHAR2(50) NOT NULL,  
city VARCHAR2(50) NOT NULL,  
zip VARCHAR2(10) NOT NULL,  
street VARCHAR2(100) NOT NULL,  
franchise_ID VARCHAR2(16) NOT NULL,  
CONSTRAINT fk_employees_franchise_id FOREIGN KEY (franchise_ID) REFERENCES  
FRANCHISE(franchise_id) );  
  
CREATE TABLE SUPPLIERS_PRODUCTS(  
supplier_ID VARCHAR2(16) PRIMARY KEY,  
supplied_Products VARCHAR2(255) NOT NULL,  
CONSTRAINT fk_suppliers_products_supplier_ID FOREIGN KEY (supplier_ID)  
REFERENCES SUPPLIERS(supplier_id) );  
  
CREATE TABLE FRANCHISE_AGREEMENT(  
franchise_id VARCHAR2(16) NOT NULL,  
owner_id VARCHAR2(16) NOT NULL,  
agreement_id VARCHAR2(16) PRIMARY KEY CHECK (agreement_id LIKE 'FAGR%'),  
agreementdate DATE NOT NULL,  
term_start_date DATE NOT NULL,  
term_end_date DATE NOT NULL,  
CONSTRAINT fk_franchise_agreement_franchise_id FOREIGN KEY (franchise_id)  
REFERENCES FRANCHISE(franchise_id),  
CONSTRAINT fk_franchise_agreement_owner_id FOREIGN KEY (owner_id) REFERENCES  
FRANCHISEOWNERS(owner_id) );  
  
CREATE TABLE FRANCHISE_INVENTORY(  
inv_Id VARCHAR2(16) PRIMARY KEY CHECK (inv_Id LIKE 'FINV%'),  
inv_state VARCHAR2(16) NOT NULL,  
inv_zip VARCHAR2(16) NOT NULL,  
inv_street VARCHAR2(16) NOT NULL,  
franchise_id VARCHAR2(16) NOT NULL,  
CONSTRAINT fk_franchise_inventory_franchise_id FOREIGN KEY (franchise_id)  
REFERENCES FRANCHISE(franchise_id) );  
  
CREATE TABLE SALARY_DETAILS(  
employee_ID VARCHAR2(16) NOT NULL CHECK (employee_ID LIKE 'EMP%'),  
franchise_ID VARCHAR2(16) NOT NULL CHECK (franchise_ID LIKE 'FA%'),  
bonus NUMBER NOT NULL CHECK (bonus >= 0),  
base_Salary NUMBER NOT NULL CHECK (base_Salary >= 0),  
pay_frequency VARCHAR2(50) NOT NULL,  
PRIMARY KEY (employee_ID, franchise_ID),  
CONSTRAINT fk_salary_details_employee_ID FOREIGN KEY (employee_ID)  
REFERENCES EMPLOYEES(employee_ID),  
CONSTRAINT fk_salary_details_franchise_ID FOREIGN KEY (franchise_ID)  
REFERENCES FRANCHISE(franchise_id) );  
  
CREATE TABLE MANAGERS(  
employee_ID VARCHAR2(16) PRIMARY KEY CHECK (employee_ID LIKE 'EMP%'),  
CONSTRAINT fk_managers_employee_ID FOREIGN KEY (employee_ID) REFERENCES  
EMPLOYEES(employee_ID) );
```



```
CREATE TABLE SUPERVISOR(  
employee_ID VARCHAR2(16) PRIMARY KEY CHECK (employee_ID LIKE 'EMP%'),  
CONSTRAINT fk_supervisor_employee_ID FOREIGN KEY (employee_ID) REFERENCES  
EMPLOYEES(employee_ID) );  
  
CREATE TABLE DISTRIBUTION_CENTERS(  
center_ID VARCHAR2(16) PRIMARY KEY CHECK (center_ID LIKE 'DC%'),  
center_name VARCHAR2(100) NOT NULL,  
operating_hours VARCHAR2(50) NOT NULL,  
storage_capacity NUMBER NOT NULL CHECK (storage_capacity >= 0),  
zip VARCHAR2(10) NOT NULL,  
state VARCHAR2(50) NOT NULL,  
city VARCHAR2(50) NOT NULL,  
street VARCHAR2(100) NOT NULL );  
  
CREATE TABLE DC_INVENTORY_ITEMS(  
dcinventoryID VARCHAR2(16) PRIMARY KEY CHECK (dcinventoryID LIKE 'DCIN%'),  
inventoryType VARCHAR2(50) NOT NULL,  
dcItem_Quantity NUMBER NOT NULL CHECK (dcItem_Quantity >= 0),  
dcItemName VARCHAR2(100) NOT NULL,  
center_ID VARCHAR2(16) NOT NULL,  
CONSTRAINT fk_dc_inventory_center_id FOREIGN KEY (center_ID) REFERENCES  
DISTRIBUTION_CENTERS(center_ID) );  
  
CREATE TABLE DISTRIBUTION_CENTERS_CONTACT(  
center_ID VARCHAR2(16) PRIMARY KEY CHECK (center_ID LIKE 'DC%'),  
contactinfo VARCHAR2(100) NOT NULL,  
CONSTRAINT fk_distribution_contact_center_id FOREIGN KEY (center_ID)  
REFERENCES DISTRIBUTION_CENTERS(center_ID) );  
  
CREATE TABLE THIRD_PARTY_LOGISTICS(  
TPL_ID VARCHAR2(16) PRIMARY KEY CHECK (TPL_ID LIKE 'TPL%'),  
TPLname VARCHAR2(100) NOT NULL,  
contactinfo VARCHAR2(255) NOT NULL,  
hq_location VARCHAR2(255) NOT NULL,  
fleetsize NUMBER NOT NULL CHECK (fleetsize >= 0) );  
  
CREATE TABLE THIRD_PARTY_LOGISTICS_SERVICES(  
TPL_ID VARCHAR2(16) PRIMARY KEY CHECK (TPL_ID LIKE 'TPL%'),  
services_provided VARCHAR2(255) NOT NULL,  
CONSTRAINT fk_third_party_logistics_services_TPL_ID FOREIGN KEY (TPL_ID)  
REFERENCES THIRD_PARTY_LOGISTICS(TPL_ID) );  
  
CREATE TABLE OPERATED_BY(  
TPL_ID VARCHAR2(16) NOT NULL CHECK (TPL_ID LIKE 'TPL%'),  
center_ID VARCHAR2(16) NOT NULL CHECK (center_ID LIKE 'DC%'),  
PRIMARY KEY (TPL_ID, center_ID),  
CONSTRAINT fk_operated_by_TPL_ID FOREIGN KEY (TPL_ID) REFERENCES  
THIRD_PARTY_LOGISTICS(TPL_ID),  
CONSTRAINT fk_operated_by_center_ID FOREIGN KEY (center_ID) REFERENCES  
DISTRIBUTION_CENTERS(center_ID) );  
  
CREATE TABLE NON_PERISHABLE(  
dcinventoryID VARCHAR2(16) PRIMARY KEY CHECK (dcinventoryID LIKE 'DCIN%'),  
recyclable VARCHAR2(3) CHECK (recyclable IN ('YES', 'NO')),  
CONSTRAINT fk_non_perishable_dcinventoryID FOREIGN KEY (dcinventoryID)  
REFERENCES DC_INVENTORY_ITEMS(dcinventoryID) );
```




```
CREATE TABLE PERISHABLE(  
dcinventoryID VARCHAR2(16) PRIMARY KEY CHECK (dcinventoryID LIKE 'DCIN%'),  
expirationDate DATE NOT NULL,  
CONSTRAINT fk_perishable_dcinventoryID FOREIGN KEY (dcinventoryID)  
REFERENCES DC_INVENTORY_ITEMS(dcinventoryID) );  
  
CREATE TABLE PERISHABLE_STORAGE(  
dcinventoryID VARCHAR2(16) PRIMARY KEY CHECK (dcinventoryID LIKE 'DCIN%'),  
storage_requirements VARCHAR2(255) NOT NULL,  
CONSTRAINT fk_perishable_storage_dcinventoryID FOREIGN KEY (dcinventoryID)  
REFERENCES DC_INVENTORY_ITEMS(dcinventoryID) );  
  
CREATE TABLE MENU_ITEMS(  
mItemID VARCHAR2(16) PRIMARY KEY CHECK (mItemID LIKE 'MI%'),  
mItemName VARCHAR2(100) NOT NULL,  
mitemCalories NUMBER NOT NULL CHECK (mitemCalories >= 0),  
mItemPrice NUMBER NOT NULL CHECK (mItemPrice >= 0),  
mItemCategory VARCHAR2(50) NOT NULL );  
  
CREATE TABLE PREPARES(  
franchiseID VARCHAR2(16) NOT NULL CHECK (franchiseID LIKE 'FA%'),  
mItemID VARCHAR2(16) NOT NULL CHECK (mItemID LIKE 'MI%'),  
PRIMARY KEY (franchiseID, mItemID),  
CONSTRAINT fk_prepares_franchiseID FOREIGN KEY (franchiseID) REFERENCES  
FRANCHISE(Franchise_ID),  
CONSTRAINT fk_prepares_mItemID FOREIGN KEY (mItemID) REFERENCES  
MENU_ITEMS(mItemID) );  
  
CREATE TABLE PROMOTIONS(  
promoID VARCHAR2(16) PRIMARY KEY CHECK (promoID LIKE 'PR%'),  
promo_name VARCHAR2(100) NOT NULL,  
promo_validity DATE NOT NULL,  
promo_start_date DATE NOT NULL );  
  
CREATE TABLE PROMO_OFFERS(  
promoID VARCHAR2(16) PRIMARY KEY CHECK (promoID LIKE 'PR%'),  
promo_offers VARCHAR2(255) NOT NULL,  
CONSTRAINT fk_promo_offers_promoID FOREIGN KEY (promoID) REFERENCES  
PROMOTIONS(promoID) );  
  
CREATE TABLE MARKETING_CAMPAIGNS(  
campaignID VARCHAR2(16) PRIMARY KEY CHECK (campaignID LIKE 'MC%'),  
validity DATE NOT NULL,  
campaign_type VARCHAR2(50) NOT NULL,  
budget_alloted NUMBER NOT NULL CHECK (budget_alloted >= 0) );  
  
CREATE TABLE ORGANISED_PROMO(  
campaign_id VARCHAR2(16) NOT NULL CHECK (campaign_id LIKE 'MC%'),  
promo_id VARCHAR2(16) NOT NULL CHECK (promo_id LIKE 'PR%'),  
PRIMARY KEY (campaign_id, promo_id), CONSTRAINT  
fk_organised_promo_campaign_id FOREIGN KEY (campaign_id) REFERENCES  
MARKETING_CAMPAIGNS(campaignID),  
CONSTRAINT fk_organised_promo_promo_id FOREIGN KEY (promo_id) REFERENCES  
PROMOTIONS(promoID) );  
  
CREATE TABLE CONDUCTED_CAMPAIGNS(  

```



```
franchise_id VARCHAR2(16) NOT NULL CHECK (franchise_id LIKE 'FA%'),
campaign_id VARCHAR2(16) NOT NULL CHECK (campaign_id LIKE 'MC%'),
CONSTRAINT fk_franchise_id FOREIGN KEY (franchise_id) REFERENCES
FRANCHISE(Franchise_ID),
CONSTRAINT fk_campaign_id FOREIGN KEY (campaign_id) REFERENCES
MARKETING_CAMPAIGNS(campaignID) );
```

```
CREATE TABLE COUPONS(
coupon_code VARCHAR2(50) PRIMARY KEY,
minPurchase_amt NUMBER NOT NULL CHECK (minPurchase_amt >= 0),
usage_limit NUMBER NOT NULL CHECK (usage_limit >= 0),
coupon_type VARCHAR2(50) NOT NULL,
coupExpiry DATE NOT NULL,
coupDiscount NUMBER NOT NULL CHECK (coupDiscount >= 0) );
```

```
CREATE TABLE AVAIL_COUPONS(
promo_id VARCHAR2(16) NOT NULL CHECK (promo_id LIKE 'PR%'),
coupon_code VARCHAR2(50) NOT NULL,
CONSTRAINT fk_promo_id FOREIGN KEY (promo_id) REFERENCES
PROMOTIONS(promoID),
CONSTRAINT fk_coupon_code FOREIGN KEY (coupon_code) REFERENCES
COUPONS(coupon_code) );
```

```
CREATE TABLE CUSTOMER_ORDERS (
order_id VARCHAR2(16) PRIMARY KEY CHECK (order_id LIKE 'CO%'),
orderdate DATE NOT NULL,
Paymentmethod VARCHAR2(50) NOT NULL,
deliverymethod VARCHAR2(50) NOT NULL, totalamount NUMBER NOT NULL CHECK
(totalamount >= 0),
franchise_id VARCHAR2(16) NOT NULL,
cus_id VARCHAR2(16) NOT NULL,
CONSTRAINT fk_customer_orders_franchise_id FOREIGN KEY (franchise_id)
REFERENCES FRANCHISE(Franchise_ID),
CONSTRAINT fk_customer_orders_cus_id FOREIGN KEY (cus_id) REFERENCES
MCDCUSTOMERS(cus_id) );
```

```
CREATE TABLE LISTS (
order_id VARCHAR2(16) NOT NULL,
mitemID VARCHAR2(16) NOT NULL, PRIMARY KEY (order_id, mitemID),
CONSTRAINT fk_lists_order_id FOREIGN KEY (order_id) REFERENCES
CUSTOMER_ORDERS(order_id),
CONSTRAINT fk_lists_mitemID FOREIGN KEY (mitemID) REFERENCES
MENU_ITEMS(mItemID) );
```

```
CREATE TABLE CUS_ORDER_DETAIL (
mitemID VARCHAR2(16), order_id VARCHAR2(16) NOT NULL CHECK (order_id LIKE
'CO%'),
ordquantity NUMBER NOT NULL CHECK (ordquantity >= 0),
ordprice NUMBER NOT NULL CHECK (ordprice >= 0),
CONSTRAINT fk_cus_order_mitemid FOREIGN KEY (mitemID) REFERENCES
Menu_items(mitemid),
CONSTRAINT fk_cus_order_order_id FOREIGN KEY (order_id) REFERENCES
CUSTOMER_ORDERS(order_id) );
```

```
CREATE TABLE CUSTOMER_FEEDBACKS (
feedbackID VARCHAR2(16) PRIMARY KEY CHECK (feedbackID LIKE 'FB%'),
feedback_date DATE NOT NULL, rating NUMBER NOT NULL CHECK (rating BETWEEN 1
```



```
AND 5),  
service_used VARCHAR2(50) NOT NULL );
```

```
CREATE TABLE FEEDBACK_INFO (  
cus_id VARCHAR2(16) NOT NULL CHECK (cus_id LIKE 'MCDC%'),  
order_id VARCHAR2(16) NOT NULL CHECK (order_id LIKE 'CO%'),  
feedbackID VARCHAR2(16) NOT NULL CHECK (feedbackID LIKE 'FB%'),  
CONSTRAINT fk_feedback_info_cus_id FOREIGN KEY (cus_id) REFERENCES  
MCDCUSTOMERS(cus_id), CONSTRAINT fk_feedback_info_order_id FOREIGN KEY  
(order_id) REFERENCES CUSTOMER_ORDERS(order_id),  
CONSTRAINT fk_feedback_info_feedback_id FOREIGN KEY (feedbackID) REFERENCES  
CUSTOMER_FEEDBACKS(feedbackID) );
```

```
CREATE TABLE TRAINING_DETAILS (  
training_ID VARCHAR2(16) NOT NULL,  
employee_ID VARCHAR2(16) NOT NULL,  
train_start_date DATE NOT NULL,  
train_end_date DATE NOT NULL,  
train_duration NUMBER NOT NULL CHECK (train_duration >= 0),  
completion_status VARCHAR2(50) NOT NULL,  
PRIMARY KEY (employee_ID, training_ID),  
CONSTRAINT fk_training_details_employee_ID FOREIGN KEY (employee_ID)  
REFERENCES EMPLOYEES(employee_ID),  
CONSTRAINT fk_training_details_training_ID FOREIGN KEY (training_ID)  
REFERENCES EMPLOYEE_TRAINING(training_ID) );
```

```
CREATE TABLE ESHIFTS_DETAILS (  
Employee_ID VARCHAR2(16) NOT NULL CHECK (Employee_ID LIKE 'EM%'),  
shiftDate DATE NOT NULL,  
startTime TIMESTAMP NOT NULL,  
endTime TIMESTAMP NOT NULL,  
shiftDuration NUMBER NOT NULL CHECK (shiftDuration >= 0),  
PRIMARY KEY (Employee_ID, shiftDate),  
CONSTRAINT fk_eshifts_employee_ID FOREIGN KEY (Employee_ID) REFERENCES  
EMPLOYEES(employee_ID) );
```

```
CREATE TABLE SHIPMENT (  
trackingID VARCHAR2(16) PRIMARY KEY CHECK(trackingID LIKE 'SPM%'),  
DateShipped DATE NOT NULL,  
EstDeliveryDate DATE NOT NULL,  
DestState VARCHAR2(50) NOT NULL,  
DestZIP NUMBER NOT NULL,  
carrierName VARCHAR2(100) NOT NULL,  
ShippingCost NUMBER NOT NULL );
```

```
CREATE TABLE RAW_MATERIALS (  
material_Id VARCHAR2(16) PRIMARY KEY CHECK(material_Id LIKE 'RM%'),  
materialName VARCHAR2(100) NOT NULL,  
typeofMaterial VARCHAR2(50) NOT NULL,  
cold_storage CHAR(1) CHECK (cold_storage IN ('Y', 'N')),  
quantity NUMBER NOT NULL );
```

```
CREATE TABLE SHIPMENT_DETAILS (  
Material_Id VARCHAR2(16),  
trackingID VARCHAR2(16),  
CONSTRAINT fk_shipment_details_Material_Id FOREIGN KEY (Material_Id)  
REFERENCES RAW_MATERIALS(material_Id),
```



```
CONSTRAINT fk_shipment_details_trackingID FOREIGN KEY (trackingID)
REFERENCES SHIPMENT(trackingID),
PRIMARY KEY (Material_Id, trackingID) );

CREATE TABLE PURCHASE_ORDER (
PONo VARCHAR2(16) PRIMARY KEY CHECK (PONo LIKE 'PO%'),
PODate DATE NOT NULL,
PurcQty NUMBER NOT NULL,
Material_Id VARCHAR2(16),
CONSTRAINT fk_purchase_order_Material_Id FOREIGN KEY (Material_Id)
REFERENCES RAW_MATERIALS(material_Id) );

CREATE TABLE ORDERS_TO_DC (
orderNo VARCHAR2(16) NOT NULL CHECK (orderNo LIKE 'OTD%'),
orderDate DATE NOT NULL,
orderTotal NUMBER NOT NULL,
ordQty NUMBER NOT NULL, discount VARCHAR2(16),
center_ID VARCHAR2(16),
PONo VARCHAR2(16),
CONSTRAINT fk_orders_to_dc_center_ID FOREIGN KEY (center_ID) REFERENCES
DISTRIBUTION_CENTERS(center_ID),
CONSTRAINT fk_orders_to_dc_PONo FOREIGN KEY (PONo) REFERENCES
PURCHASE_ORDER(PONo) );

CREATE TABLE OWNS (
supplier_ID VARCHAR2(16),
warehouse_id VARCHAR2(16),
material_ID VARCHAR2(16),
CONSTRAINT fk_owns_supplier_ID FOREIGN KEY (supplier_ID) REFERENCES
SUPPLIERS(supplier_ID),
CONSTRAINT fk_owns_warehouse_id FOREIGN KEY (warehouse_id) REFERENCES
WAREHOUSE(warehouse_id),
CONSTRAINT fk_owns_material_ID FOREIGN KEY (material_ID) REFERENCES
RAW_MATERIALS(material_ID),
PRIMARY KEY (supplier_ID, warehouse_id, material_ID) );

CREATE TABLE CAN_SUPPLY (
center_ID VARCHAR2(16) NOT NULL,
Material_Id VARCHAR2(16) NOT NULL,
PRIMARY KEY (center_ID, Material_Id),
FOREIGN KEY (center_ID) REFERENCES DISTRIBUTION_CENTERS(center_ID),
FOREIGN KEY (Material_Id) REFERENCES RAW_MATERIALS(material_Id) );

CREATE TABLE CREW_MEMBER (
employee_ID VARCHAR2(16) NOT NULL,
supervisor_ID VARCHAR2(16),
PRIMARY KEY (employee_ID),
FOREIGN KEY (employee_ID) REFERENCES EMPLOYEES(employee_ID),
FOREIGN KEY (supervisor_ID) REFERENCES SUPERVISOR(employee_ID) );

CREATE TABLE DELIVERY_DETAILS (
Franchise_ID VARCHAR2(16) NOT NULL,
trackingID VARCHAR2(16) NOT NULL,
confirmationCode NUMBER NOT NULL,
DeliveryDate DATE,
DeliveryStatus VARCHAR2(50),
ReceiverName VARCHAR2(100),
```



```
PRIMARY KEY (Franchise_ID, trackingID, confirmationCode),
FOREIGN KEY (Franchise_ID) REFERENCES FRANCHISE(Franchise_ID),
FOREIGN KEY (trackingID) REFERENCES SHIPMENT(trackingID) );

CREATE TABLE FRANCHISEOWNER_PHONES (
owner_id VARCHAR2(16) NOT NULL,
owner_phone VARCHAR2(15),
PRIMARY KEY (owner_id),
FOREIGN KEY (owner_id) REFERENCES FRANCHISEOWNERS(owner_id) );

CREATE TABLE INVENTORY_DETAILS (
Material_Id VARCHAR2(16),
inv_Id VARCHAR2(16),
inv_qoh NUMBER,
inv_reorder_level NUMBER,
inv_stock_status VARCHAR2(50),
CONSTRAINT fk_inv_details_material_id FOREIGN KEY (Material_Id) REFERENCES
    RAW_MATERIALS(material_id),
CONSTRAINT fk_inv_details_inv_id FOREIGN KEY (inv_Id) REFERENCES
    FRANCHISE_INVENTORY(inv_Id) )
```



Chapter 4: SQL Queries

Query 1 : This query provides a ranked analysis of menu item feedback for each franchise, based on customer ratings. It calculates the average feedback rating and total feedback count for each menu item at a franchise, ranks the items within each franchise by their ratings, and filters out items with no feedback. The results help franchises identify their top-rated menu items, assess customer satisfaction, and make informed decisions about menu optimization, promotions, or potential adjustments to poorly rated items.

```
WITH MenuFeedback AS (
    SELECT
        F.franchise_id,
        MI.mItemName,
        AVG(CF.rating) AS avg_feedback_rating,
        COUNT(CF.feedbackID) AS total_feedbacks
    FROM
        MENU_ITEMS MI
    LEFT JOIN
        LISTS L ON MI.mItemID = L.mitemID
    LEFT JOIN
        CUSTOMER_ORDERS CO ON L.order_id = CO.order_id
    LEFT JOIN
        FEEDBACK_INFO FI ON CO.order_id = FI.order_id
    LEFT JOIN
        CUSTOMER_FEEDBACKS CF ON FI.feedbackID = CF.feedbackID
    LEFT JOIN
        FRANCHISE F ON CO.franchise_id = F.franchise_id
    GROUP BY
        F.franchise_id, MI.mItemName
    HAVING
        F.franchise_id IS NOT NULL AND MI.mItemName IS NOT NULL
)
SELECT
    franchise_id,
    mItemName,
    avg_feedback_rating,
    total_feedbacks,
    ROW_NUMBER() OVER (PARTITION BY franchise_id ORDER BY
avg_feedback_rating DESC) AS feedback_rank
FROM
    MenuFeedback
WHERE
    total_feedbacks > 0
ORDER BY
    franchise_id, feedback_rank;
```

Query 2: This query evaluates yearly revenue growth for each franchise by comparing current and previous year revenues. It calculates the revenue difference and growth percentage year-over-year. The results enable



franchises to track financial performance trends, identify periods of significant growth or decline, and make data-driven decisions for strategic planning, such as investing in high-performing franchises or addressing issues in underperforming ones.

```
WITH yearly_revenue AS (
    SELECT
        franchise_id,
        EXTRACT(YEAR FROM orderdate) AS year,
        SUM(totalamount) AS total_revenue
    FROM
        CUSTOMER_ORDERS
    GROUP BY
        franchise_id, EXTRACT(YEAR FROM orderdate)
),
growth_calculation AS (
    SELECT
        franchise_id,
        year,
        total_revenue,
        LAG(total_revenue) OVER (PARTITION BY franchise_id ORDER BY
year) AS previous_year_revenue
    FROM
        yearly_revenue
)
SELECT
    franchise_id,
    year,
    total_revenue,
    COALESCE(previous_year_revenue, 0) AS previous_year_revenue,
    (total_revenue - COALESCE(previous_year_revenue, 0)) AS growth,
    CASE
        WHEN previous_year_revenue IS NULL OR previous_year_revenue
= 0 THEN '0.00%'
        ELSE TO_CHAR(((total_revenue - previous_year_revenue) /
previous_year_revenue) * 100, 'FM999990.00') || '%'
    END AS growth_percentage
FROM
    growth_calculation;
```

Query 3: This query analyzes and ranks franchise performance based on two key metrics: the duration of franchise agreements and the total inventory managed. It provides insights into franchise stability (measured by agreement duration) and operational scale (measured by inventory). By ranking franchises on these metrics, businesses can identify long-term stable franchises, assess inventory management capabilities, and prioritize support or investments in franchises with high operational demands or significant tenure. This aids in strategic decision-making for resource allocation and franchise relationship management.

```
WITH AgreementDurations AS (
```



```
SELECT
    F.franchise_id,
    FO.lname AS owner_last_name,
    FO.fname AS owner_first_name,
    F.franchise_since,
    FA.term_start_date,
    FA.term_end_date,
    ROUND((FA.term_end_date - FA.term_start_date) / 365, 2) AS
agreement_years
FROM
    FRANCHISE_AGREEMENT FA
JOIN
    FRANCHISE F ON FA.franchise_id = F.franchise_id
JOIN
    FRANCHISEOWNERS FO ON F.owner_id = FO.owner_id
),
InventoryCounts AS (
    SELECT
        FI.franchise_id,
        COUNT(FI.inv_id) AS total_inventory
    FROM
        FRANCHISE_INVENTORY FI
    GROUP BY
        FI.franchise_id
),
RankedFranchises AS (
    SELECT
        AD.franchise_id,
        AD.owner_last_name,
        AD.owner_first_name,
        AD.agreement_years,
        IC.total_inventory,
        RANK() OVER (ORDER BY AD.agreement_years DESC,
AD.franchise_id ASC ) AS agreement_rank,
        RANK() OVER (ORDER BY IC.total_inventory DESC,
AD.franchise_id ASC) AS inventory_rank
    FROM
        AgreementDurations AD
    LEFT JOIN
        InventoryCounts IC ON AD.franchise_id = IC.franchise_id
)
SELECT
    franchise_id,
    owner_last_name || ', ' || owner_first_name AS franchise_owner,
    agreement_years,
    total_inventory,
    agreement_rank,
```




```

        inventory_rank
FROM
    RankedFranchises
ORDER BY
    inventory_rank, agreement_rank;

```

Query 4: This query analyzes employee shift patterns across franchises by categorizing shifts as "Short," "Regular," or "Long" based on their durations. It calculates the total shifts, average shift duration, and total hours worked for each franchise and shift type, with results also including aggregated data across all franchises due to the CUBE operation. The business application is to monitor workforce utilization, identify labor trends, optimize staffing schedules, and ensure compliance with labor regulations while maintaining operational efficiency.

```

SELECT
F.franchise_id,
CASE
WHEN ES.shiftDuration < 4 THEN 'Short Shift'
WHEN ES.shiftDuration BETWEEN 4 AND 8 THEN 'Regular Shift'
ELSE 'Long Shift'
END AS shift_type,
COUNT(*) AS total_shifts,
AVG(ES.shiftDuration) AS avg_shift_duration,
SUM(ES.shiftDuration) AS total_shift_hours
FROM FRANCHISE F
JOIN EMPLOYEES E ON F.franchise_id = E.franchise_ID
JOIN ESHIFTS_DETAILS ES ON E.employee_ID = ES.Employee_ID
GROUP BY CUBE (F.franchise_id,
CASE
WHEN ES.shiftDuration < 4 THEN 'Short Shift'
WHEN ES.shiftDuration BETWEEN 4 AND 8 THEN 'Regular Shift'
ELSE 'Long Shift'
END);

```

Query 5: This query calculates the profitability score for each franchise by comparing its seating capacity to the average employee cost (base salary plus bonus). The profitability score helps assess how efficiently each franchise utilizes its workforce relative to its size, providing insights into labor cost management and operational efficiency. This analysis can guide decision-making for resource allocation, staffing strategies, and identifying franchises that are performing well or need improvement in cost management.

```

WITH FranchiseProfit AS (
    SELECT f.franchise_id, f.city, f.seating_capacity,
           AVG(s.base_salary + COALESCE(s.bonus, 0)) AS
avg_employee_cost
    FROM FRANCHISE f
    JOIN SALARY_DETAILS s ON f.franchise_id = s.franchise_id
    GROUP BY f.franchise_id, f.city, f.seating_capacity
)

```



```
SELECT franchise_id, city, seating_capacity, avg_employee_cost,
       (round((seating_capacity / avg_employee_cost),5)) AS
profitability_score
FROM FranchiseProfit
ORDER BY profitability_score DESC;
```

Query 6: This query evaluates the performance of franchise owners by calculating the total revenue generated by their franchises. It ranks owners based on their total revenue, providing a clear picture of which owners are performing best. The business application is to identify top-performing franchise owners, reward successful ones, and assess potential areas for improvement or support for lower-performing owners. This information helps in making decisions related to incentives, resource allocation, and franchise growth strategies.

```
WITH FranchisePerformance AS (
    SELECT
        F.franchise_id,
        F.owner_id,
        SUM(O.totalamount) AS total_revenue
    FROM
        FRANCHISE F
    JOIN
        CUSTOMER_ORDERS O ON F.franchise_id = O.franchise_id
    GROUP BY
        F.franchise_id, F.owner_id
),
OwnerPerformance AS (
    SELECT
        FP.owner_id,
        FO.fname || ' ' || FO.lname AS owner_name, -- Concatenation
        SUM(FP.total_revenue) AS total_owner_revenue,
        RANK() OVER (ORDER BY SUM(FP.total_revenue) DESC) AS
owner_rank
    FROM
        FranchisePerformance FP
    JOIN
        FRANCHISEOWNERS FO ON FP.owner_id = FO.owner_id
    GROUP BY
        FP.owner_id, FO.fname, FO.lname
)
SELECT
    owner_id,
    owner_name,
    total_owner_revenue,
    owner_rank
FROM
    OwnerPerformance
;
```



Query 7: This query calculates each franchise's revenue share as a percentage of the total company revenue. By summing up the revenue generated by each franchise and comparing it to the overall total revenue, it helps assess the contribution of individual franchises to the business. The business application is to identify high-performing franchises, track revenue distribution across locations, and inform strategic decisions regarding resource allocation, support for underperforming franchises, and potential expansion opportunities.

```
WITH TotalRevenue AS (  
    SELECT  
        SUM(CO.totalamount) AS total_revenue  
    FROM  
        CUSTOMER_ORDERS CO  
)  
SELECT  
    F.franchise_id,  
    F.city,  
    SUM(CO.totalamount) AS franchise_revenue,  
    ROUND((SUM(CO.totalamount) * 100.0 / (SELECT total_revenue FROM  
TotalRevenue)), 2) AS revenue_share_percentage  
FROM  
    CUSTOMER_ORDERS CO  
INNER JOIN  
    FRANCHISE F ON CO.franchise_id = F.franchise_id  
GROUP BY  
    F.franchise_id, F.city  
ORDER BY  
    franchise_revenue DESC;
```

Query 8: This query calculates the staff-to-seating ratio for each franchise, which compares the number of employees to the seating capacity of the franchise. The business application is to assess whether a franchise is overstaffed or understaffed relative to its size, helping optimize labor costs and staffing efficiency. It can guide decisions on workforce adjustments, improve operational management, and ensure better customer service by aligning staff levels with franchise needs.

```
WITH FranchiseStaffing AS (  
    SELECT f.franchise_id, f.city, f.seating_capacity,  
        COUNT(e.employee_id) AS staff_count  
    FROM FRANCHISE f  
    JOIN EMPLOYEES e ON f.franchise_id = e.franchise_id  
    GROUP BY f.franchise_id, f.city, f.seating_capacity  
)  
SELECT franchise_id, city, seating_capacity, staff_count,  
    (staff_count / seating_capacity) AS staff_to_seating_ratio  
FROM FranchiseStaffing  
ORDER BY staff_to_seating_ratio DESC;
```



Query 9: This query calculates the revenue per employee for each franchise by dividing total revenue by the number of employees. The business application is to measure labor productivity and efficiency across franchises. By identifying franchises with higher revenue per employee, the business can recognize top-performing locations, optimize staffing levels, and assess where improvements in employee performance or resource allocation might be necessary. This metric helps in making decisions regarding staffing, operational improvements, and resource distribution.

```
WITH FranchiseRevenue AS (  
    SELECT  
        CO.franchise_id,  
        SUM(CO.totalamount) AS total_revenue  
    FROM  
        CUSTOMER_ORDERS CO  
    GROUP BY  
        CO.franchise_id  
) ,  
FranchiseStaff AS (  
    SELECT  
        F.franchise_id,  
        COUNT(E.employee_id) AS num_employees  
    FROM  
        FRANCHISE F  
    INNER JOIN  
        EMPLOYEES E ON F.franchise_id = E.franchise_id  
    GROUP BY  
        F.franchise_id  
)  
SELECT  
    R.franchise_id,  
    F.city,  
    R.total_revenue,  
    S.num_employees,  
    ROUND(R.total_revenue / S.num_employees, 2) AS  
revenue_per_employee  
FROM  
    FranchiseRevenue R  
INNER JOIN  
    FranchiseStaff S ON R.franchise_id = S.franchise_id  
INNER JOIN  
    FRANCHISE F ON R.franchise_id = F.franchise_id  
WHERE  
    S.num_employees > 0  
ORDER BY  
    revenue_per_employee DESC;
```

Query 10: This query evaluates franchise performance by analyzing material purchases relative to seating capacity over the last six months. It calculates the ratio of purchases to seating capacity (purchases_per_seat)



for each franchise and compares it to the average across all franchises. Based on this comparison, it categorizes franchises as either "Needs expansion" (if their ratio significantly exceeds the average) or "Adequate." This analysis helps identify high-performing locations that may require capacity upgrades to meet demand and optimize resource allocation.

```
WITH franchise_purchases AS (  
    SELECT dd.Franchise_ID, SUM(po.PurcQty) AS total_purchases  
    FROM DELIVERY_DETAILS dd  
    JOIN SHIPMENT_DETAILS sd ON dd.trackingID = sd.trackingID  
    JOIN PURCHASE_ORDER po ON sd.Material_Id = po.Material_Id  
    WHERE dd.DeliveryDate >= ADD_MONTHS(SYSDATE, -6)  
    GROUP BY dd.Franchise_ID  
)  
SELECT f.franchise_id, f.city, f.seating_capacity,  
       fp.total_purchases,  
       ROUND((fp.total_purchases / f.seating_capacity),2) AS  
purchases_per_seat,  
       ROUND(AVG(fp.total_purchases / f.seating_capacity) OVER (),  
2) AS avg_purchases_per_seat,  
       CASE  
           WHEN (fp.total_purchases / f.seating_capacity) >  
                (AVG(fp.total_purchases / f.seating_capacity) OVER  
( ) * 1.5)  
           THEN 'Needs expansion'  
           ELSE 'Adequate'  
       END AS capacity_status  
FROM FRANCHISE f  
JOIN franchise_purchases fp ON f.franchise_id = fp.Franchise_ID  
ORDER BY purchases_per_seat DESC;
```



Chapter 5: Triggers and Procedure

Trigger 1: The provided trigger calculates the total amount for a customer order by checking if a coupon code is applied. If a valid coupon is provided (meeting the minimum purchase amount and not expired), the corresponding discount is applied to the subtotal. Otherwise, the total amount remains equal to the subtotal. This ensures that discounts are only granted under valid conditions.

```
CREATE OR REPLACE TRIGGER trg_calculate_totalamount
BEFORE INSERT OR UPDATE ON CUSTOMER_ORDERS
FOR EACH ROW
DECLARE
    v_minPurchase_amt NUMBER;
    v_coupExpiry DATE;
    v_coupDiscountAmt NUMBER;
BEGIN
    -- Check if coupon code is provided
    IF :NEW.coupon_code IS NOT NULL THEN
        -- Fetch coupon details
        SELECT minPurchase_amt, coupExpiry, coupDiscountAmt
        INTO v_minPurchase_amt, v_coupExpiry, v_coupDiscountAmt
        FROM COUPONS
        WHERE coupon_code = :NEW.coupon_code;
        -- Validate the coupon
        IF :NEW.subtotal >= v_minPurchase_amt AND SYSDATE <=
v_coupExpiry THEN
            -- Valid coupon: apply discount
            :NEW.totalamount := :NEW.subtotal - v_coupDiscountAmt;
        ELSE
            -- Invalid coupon: no discount applied
            :NEW.totalamount := :NEW.subtotal;
        END IF;
    ELSE
        -- No coupon provided: totalamount = subtotal
        :NEW.totalamount := :NEW.subtotal;
    END IF;
END;
/
```

Trigger 2: This trigger automatically updates the inv_stock_status field in the INVENTORY_DETAILS table based on the quantity on hand (inv_qoh) relative to the reorder level (inv_reorder_level). It sets the status to "Low" if stock is at or below the reorder level, "Decent" if it's up to twice the reorder level, and "High" otherwise.

```
CREATE OR REPLACE TRIGGER trg_update_stock_status
BEFORE INSERT OR UPDATE ON INVENTORY_DETAILS
FOR EACH ROW
BEGIN
    -- Evaluate the stock status based on inv_qoh and
```



```

inv_reorder_level
  IF :NEW.inv_qoh <= :NEW.inv_reorder_level THEN
    :NEW.inv_stock_status := 'Low';
  ELSIF :NEW.inv_qoh <= 2 * :NEW.inv_reorder_level THEN
    :NEW.inv_stock_status := 'Decent';
  ELSE
    :NEW.inv_stock_status := 'High';
  END IF;
END;
/

```

Trigger 3: This trigger ensures efficient inventory management by validating stock availability before processing new shipments. When a shipment is recorded, it checks if sufficient stock exists for the raw material. If stock is unavailable, it prevents the shipment and raises an error, avoiding overselling or overbooking issues. Additionally, it immediately reduces the stock count upon successful validation, maintaining real-time inventory accuracy and enabling better supply chain decisions.

```

CREATE OR REPLACE TRIGGER trg_validate_stock_on_shipment
BEFORE INSERT ON SHIPMENT_DETAILS
FOR EACH ROW
DECLARE
  available_quantity NUMBER;
BEGIN
  -- Get the current stock for the raw material
  SELECT quantity
  INTO available_quantity
  FROM RAW_MATERIALS
  WHERE material_Id = :NEW.Material_Id;

  -- Validate stock availability
  IF available_quantity <= 0 THEN
    RAISE_APPLICATION_ERROR(-20005, 'Insufficient stock for the
material.');
```

material.');

```

  END IF;

  -- Reserve stock (reduce immediately to avoid overbooking)
  UPDATE RAW_MATERIALS
  SET quantity = quantity - 1
  WHERE material_Id = :NEW.Material_Id;
END;
/

```

Procedure: The ProcessPurchaseOrder function automates the handling of raw material purchase orders by validating the material's existence in the RAW_MATERIALS table, inserting the purchase order details into the PURCHASE_ORDER table, and updating the inventory to reflect the new stock. It also includes error handling to address issues like missing materials or unexpected failures, ensuring that changes are rolled back in case of errors.

```

CREATE OR REPLACE PROCEDURE ProcessPurchaseOrder(
  p_po_no IN PURCHASE_ORDER.PONo%TYPE,

```



```
p_material_id IN RAW_MATERIALS.material_Id%TYPE,
p_purchase_qty IN PURCHASE_ORDER.PurcQty%TYPE,
p_po_date IN PURCHASE_ORDER.PODate%TYPE
) AS
v_current_quantity RAW_MATERIALS.quantity%TYPE;
v_new_quantity RAW_MATERIALS.quantity%TYPE;
BEGIN
-- Validate if the material exists
SELECT quantity INTO v_current_quantity
FROM RAW_MATERIALS
WHERE material_Id = p_material_id;

-- Calculate the new quantity after purchase
v_new_quantity := v_current_quantity + p_purchase_qty;

-- Insert the purchase order details
INSERT INTO PURCHASE_ORDER (
    PONO, PODate, PurcQty, Material_Id
) VALUES (
    p_po_no, p_po_date, p_purchase_qty, p_material_id
);

-- Update the quantity in RAW_MATERIALS
UPDATE RAW_MATERIALS
SET quantity = v_new_quantity
WHERE material_Id = p_material_id;

-- Commit the transaction
COMMIT;

DBMS_OUTPUT.PUT_LINE('Purchase Order Processed Successfully: ' ||
p_po_no);
DBMS_OUTPUT.PUT_LINE('Material ID: ' || p_material_id || ' Updated
Quantity: ' || v_new_quantity);

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20002, 'Material ID not found in
RAW_MATERIALS table. ');
    WHEN OTHERS THEN
        ROLLBACK;
        RAISE_APPLICATION_ERROR(-20003, 'An unexpected error occurred: '
|| SQLERRM);
END ProcessPurchaseOrder;
/
```

How to execute:

```
BEGIN
    ProcessPurchaseOrder('PO0000000013', 'RM0000000001', 500, SYSDATE);
END;
/
```


Chapter 6: User Interface

URL : <https://apex.oracle.com/pls/apex/r/mis531project/mcdonald-s/home?session=264153979629>

Credentials: Username: faizcurrim

Password: mis531@123

User walk-through

Users can access the website by clicking the provided link, which directs them to the login page. After entering the supplied credentials, users can log in and reach the homepage. The navbar, located on the left side of the page, enables navigation through various sections of the website. Users can utilize this navbar to access SQL query reports and forms designed for performing CRUD (Create, Read, Update, Delete) operations. The accompanying screenshots illustrate a comprehensive, screen-by-screen navigation of the website, showcasing the CRUD operations and corresponding query results.

Screenshots:

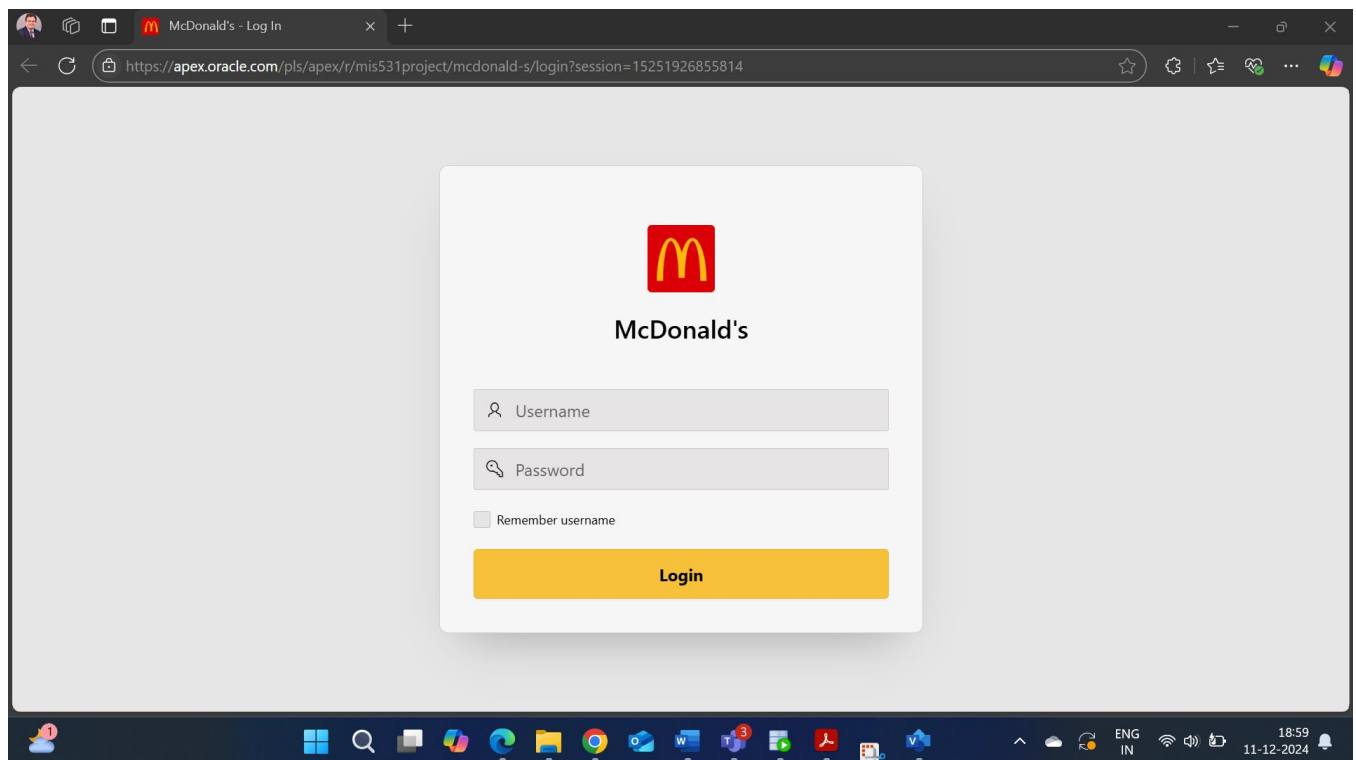


Figure 1: Login Page

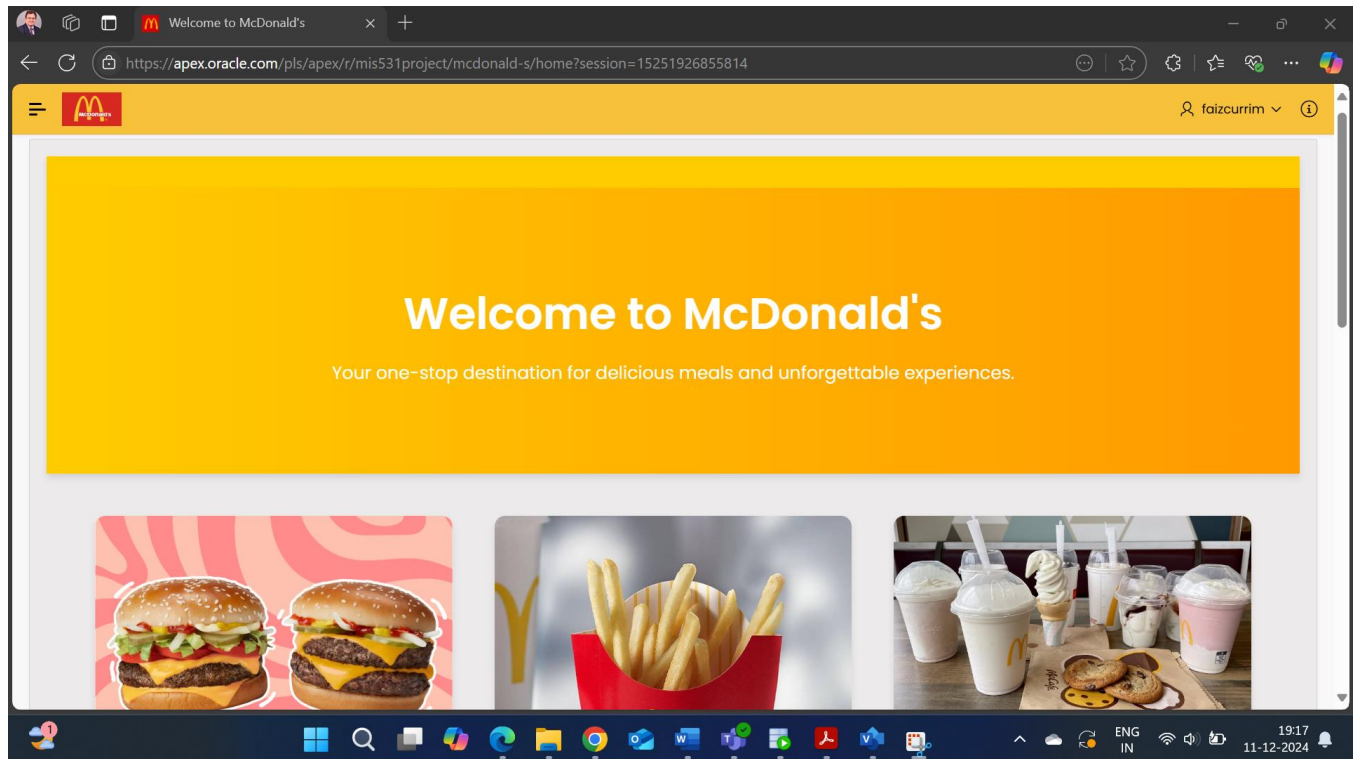


Figure 2: Homepage

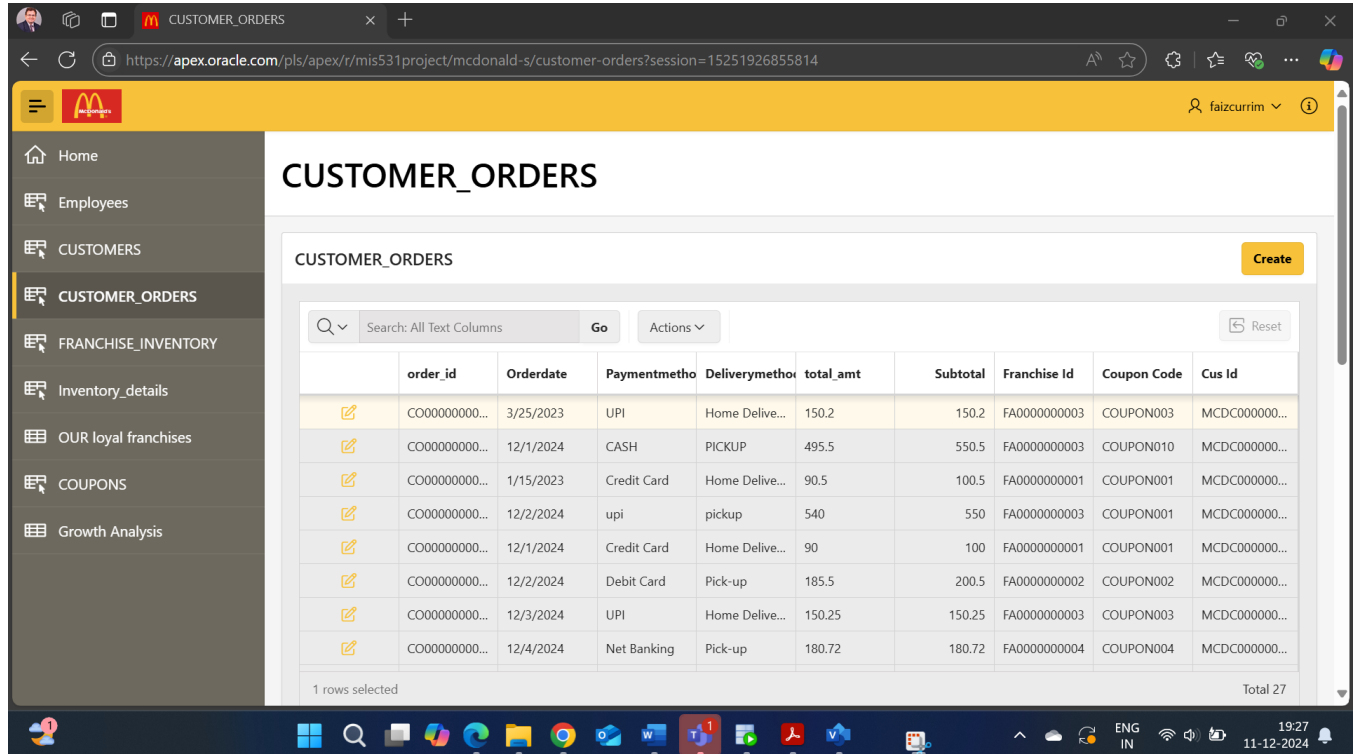


Figure 3: Customer_Orders Table(CRUD Table view)



ORDERS_FORM

order_id

Orderdate

Paymentmethod

Deliverymethod

Subtotal

Franchise Id

Coupon Code

Cancel Create

Id	Coupon Code	Cus Id
0003	COUPON003	MCDC000000...
0003	COUPON010	MCDC000000...
0001	COUPON001	MCDC000000...
0003	COUPON001	MCDC000000...
0001	COUPON001	MCDC000000...
0002	COUPON002	MCDC000000...
0003	COUPON003	MCDC000000...
0004	COUPON004	MCDC000000...

Total 27

Figure 4: Customer_Orders Table (Insert form view)

Employees

Employees

Search: All Text Columns Go Actions

	Employee_id	Contact Number	Emergency Contact	Date of Hire	State	City	Zip	Street	Franchise Id
✎	EMP00000000...	555-1111	555-2222	2/15/2015	California	Los Angeles	90001	123 Main St	FA00000000001
✎	EMP00000000...	555-3333	555-4444	3/22/2016	Texas	Austin	73301	456 Oak Ave	FA00000000002
✎	EMP00000000...	555-5555	555-6666	5/10/2017	Florida	Miami	33101	789 Pine Blvd	FA00000000003
✎	EMP00000000...	555-7777	555-8888	7/1/2014	New York	New York	10001	101 Broadwa...	FA00000000004
✎	EMP00000000...	555-9999	555-1010	8/15/2018	Illinois	Chicago	60601	202 Maple St	FA00000000005
✎	EMP00000000...	555-1212	555-1313	9/22/2019	Arizona	Phoenix	85001	303 Cedar Ln	FA00000000006
✎	EMP00000000...	555-1414	555-1515	10/30/2013	Nevada	Las Vegas	89501	404 Birch Rd	FA00000000007
✎	EMP00000000...	555-1616	555-1717	1/25/2020	Ohio	Cleveland	44101	505 Elm St	FA00000000008

1 rows selected Total 11

Figure 5: Employee Table



CUSTOMERS

1 rows selected

	Customer_id..	Cusemail	Cusdob	State	City	Street	Zip	Firstname	Lastname	Contactnumber
	MCDC0000...	nimishbhas...	12/4/2024	ARIZONA	TUCSON	1150 E. 8th...	85719	Nimish Bha...	Vasagiri	5202309474
	MCDC0000...	john.doe@...	5/12/1985	California	Los Angeles	1234 Suns...	90001	John	Doe	555-1234
	MCDC0000...	jane.smith...	11/23/1990	Texas	Austin	4567 Oak St	73301	Jane	Smith	555-5678
	MCDC0000...	michael.br...	3/15/1978	Florida	Miami	789 Pine Ave	33101	Michael	Brown	555-8765
	MCDC0000...	lisa.jones@...	7/20/1995	New York	New York	101 Broad...	10001	Lisa	Jones	555-4321
	MCDC0000...	chris.white...	9/9/1983	Illinois	Chicago	202 Maple ...	60601	Chris	White	555-2233
	MCDC0000...	emily.mille...	12/30/1989	Arizona	Phoenix	303 Cedar ...	85001	Emily	Miller	555-3344
	MCDC0000...	david.wilso...	6/5/1975	Nevada	Las Vegas	404 Birch Rd	89501	David	Wilson	555-5566

Total 12

Figure 6: Customer Table

FRANCHISE_INVENTORY

1 rows selected

	inv_id	Inv State	Inv Zip	Inv Street	Franchise Id
	FINV000000001	California	90001	Sunset Blvd	FA00000000001
	FINV000000002	Texas	73301	Congress Ave	FA00000000002
	FINV000000003	Florida	33101	Ocean Drive	FA00000000003
	FINV000000004	New York	10001	5th Avenue	FA00000000004
	FINV000000005	Nevada	89001	Fremont St	FA00000000005
	FINV000000006	Illinois	60601	Michigan Ave	FA00000000006
	FINV000000007	Georgia	30301	Peachtree St	FA00000000007
	FINV000000008	Arizona	85001	Camelback Rd	FA00000000008

Total 15

Figure 7: Franchise Inventory Table



Inventory_details

INVENTORY_DETAILS

Create

Search: All Text Columns Go Actions Reset

	Material_id	Inventory_id	Inv Qoh	Inv Reorder Level	Inv Stock Status
✎	RM0000000005	FINV000000001	500	486	Decent
✎	RM0000000001	FINV000000005	57	140	Low
✎	RM0000000004	FINV000000001	400	40	High
✎	RM0000000004	FINV000000001	10	13	Low
✎	RM0000000001	FINV000000001	120	50	High
✎	RM0000000002	FINV000000001	200	250	Low

1 rows selected Total 6

Figure 8: Inventory Details Table

COUPONS

COUPONS

Create

Search: All Text Columns Go Actions Reset

	Minpurchase Amt	Coupexpiry	Coupdiscountamt
✎	100	1/15/2025	10
✎	150	1/20/2025	15
✎	200	1/25/2025	20
✎	250	1/30/2025	25
✎	300	2/5/2025	30
✎	350	2/10/2025	35
✎	400	2/15/2025	40
✎	450	2/20/2025	45

1 rows selected Total 10

Figure 9: Coupons Table



OUR loyal franchises

Franchise Id	Franchise Owner	Agreement Years	Total Inventory	Agreement Rank	Inventory Rank
FA0000000001	Johnson, Michael	10.01	4	1	1
FA0000000002	Brown, Linda	10.01	2	2	2
FA0000000003	Williams, David	10.01	2	3	3
FA0000000004	Jones, Jennifer	10.01	1	4	4
FA0000000006	Davis, Emily	10.01	1	5	5
FA0000000007	Taylor, Brian	10.01	1	6	6
FA0000000008	Martinez, Sophia	10.01	1	7	7
FA0000000009	Garcia, James	10.01	1	8	8
FA0000000010	Rodriguez, Sarah	10.01	1	9	9

1 - 9

Figure 10: Franchise ranking query result

Growth Analysis

Franchise Id	Year	Total Revenue	Previous Year Revenue	Growth	Growth Percentage
FA0000000001	2023	90.5	0	90.5	0.00%
FA0000000001	2024	90	90.5	-.5	-0.55%
FA0000000002	2022	165.9	0	165.9	0.00%
FA0000000002	2024	185.5	165.9	19.6	11.81%
FA0000000003	2022	160.45	0	160.45	0.00%
FA0000000003	2023	150.2	160.45	-10.25	-6.39%
FA0000000003	2024	1185.75	150.2	1035.55	689.45%
FA0000000004	2022	190.3	0	190.3	0.00%
FA0000000004	2023	180	190.3	-10.3	-5.41%
FA0000000004	2024	180.72	180	.72	0.40%
FA0000000005	2022	210.5	0	210.5	0.00%

Figure 11: Growth analysis query result



Chapter 7: Implementation Plan

The financial implementation plan for the McDonald's database modeling project includes key cost components. **Personnel costs** account for \$127,500, calculated based on a 14-week project duration with a 10-member team working approximately 560 person-hours at an average rate of \$125/hour. **Cloud infrastructure costs** for database management and application hosting are estimated at **\$24,510.60 for 12 months**, covering AWS EC2 and RDS services. **Software licenses and tools** required for DBMS, testing, and analytics are projected to cost \$5,000. Additionally, **ongoing support and maintenance** for six months post-deployment, including database optimization and updates, is estimated at \$30,000, calculated at \$5,000/month.

Total Project Cost: \$187,010.60

This revised estimate reflects a comprehensive and detailed financial breakdown for implementing the database system in a real-world environment.

The screenshot displays the AWS Pricing Calculator interface. The browser address bar shows 'calculator.aws/#/estimate'. The page title is 'My Estimate - AWS Pricing Calculator'. The main content area is titled 'My Estimate' and includes an 'Edit' link. The 'Estimate summary' section shows the following costs:

Category	Cost
Upfront cost	0.00 USD
Monthly cost	2,042.55 USD
Total 12 months cost	24,510.60 USD

The total cost includes the upfront cost. The 'Getting Started with AWS' section provides links to 'Get started for free' and 'Contact Sales'. Below the summary, the 'My Estimate' section lists the services included in the estimate:

Service Name	Status	Upfront cost	Monthly cost	Description	Region	Config Summary
Amazon EC2	-	0.00 USD	1,203.41 USD	-	US West (Oregon)	Tenancy (Shared Instances), Operating system (Windows Server w...)
Amazon RDS C...	-	0.00 USD	839.14 USD	rds est	US West (Oregon)	Storage for each RDS Custom instance (General Purpose SSD (gp2...)

The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 23:10 on 11-12-2024.

Appendix A

The project on creating an ER model for McDonald's supply chain and franchise data management provided invaluable lessons about handling complex, real-world systems. Developing the ER model deepened our understanding of the intricate relationships among suppliers, franchises, and operations, emphasizing the importance of data abstraction. Through normalization, we learned how eliminating redundancy and maintaining data integrity can simplify updates, ensure consistency, and improve database performance, especially in large-scale environments like McDonald's. Running queries for data analysis underscored the need for query optimization to handle vast datasets efficiently and extract actionable insights. Implementing triggers showcased the potential of automation in improving operational efficiency, such as automating inventory restocking or updating franchise royalties in real-time. Furthermore, deploying the database on the web highlighted challenges related to system integration, security, and performance, teaching us the importance of designing for concurrent requests while ensuring a seamless user experience.

Insights from other groups during presentations added further depth to our learning. We observed innovative approaches to modeling complex relationships, such as the use of graph-based techniques to represent hierarchical data. Many groups prioritized scalability, which highlighted strategies like partitioning and indexing to design databases that can grow with evolving requirements. Additionally, the focus on user-centric deployment in some projects underscored the importance of intuitive dashboards and responsive designs, inspiring us to prioritize usability alongside functionality. Overall, this project provided a comprehensive learning experience, blending technical skills, problem-solving, and exposure to diverse perspectives, which will be immensely beneficial in addressing real-world challenges.

References

AWS Pricing Calculator: <https://calculator.aws/#/>

<https://corporate.mcdonalds.com/corpmcd/our-purpose-and-impact/food-quality-and-sourcing/responsible-sourcing.html>

Li, S. (n.d.). The Supply Chain Management of McDonald's. *Business Management, University of Leeds*, Leeds, LS2 9JT, United Kingdom.

Verma, Rakesh. (2019). ERC in Fast Food Supply Chain Master student. 10.13140/RG.2.2.31050.72648.