

DROPS
Mobile Delivery System
Alexandra Medina

Table of Contents

Abstract	3
Mission Statement	4
Mission Objectives	4
Major User Views	6
ER Diagram	7
Relational Model	8
Normalization	12
List of Actors	15
Use Cases	16
Joint Queries	31
Testing	34
Conclusion	43

Abstract

Project Title: Drops

Abstract:

Drops delivers high-valued video games, systems, and sneakers, same-day, to your door. With a business model which requires the constant movement of mobile inventory and assets, Drops mobile delivery system allows both staff and customers to keep track of inventory on the go. Through a short series of product inventory reviews, customers can track their orders from their smartphones. The goal of this system is to reduce overall operational costs, minimize shrink, and increase customers visibility.

Mission Statement & Mission Objectives

Mission Statement:

The purpose of this mobile delivery management system is to maintain data that is used and generated to support the moving inventory for our suppliers and to facilitate the cooperation and sharing of information between markets.

Mission Objectives:

- To maintain (add, update, delete) data on markets
 - To maintain (add, update, delete) data on staff
 - To maintain (add, update, delete) data on orders
 - To maintain (add, update, delete) data on games
 - To maintain (add, update, delete) data on sneakers
 - To maintain (add, update, delete) data on fleet
 - To maintain (add, update, delete) data on deliveries
 - To maintain (add, update, delete) data on customers
-
- To perform searches on markets
 - To perform searches on staff
 - To perform searches on orders
 - To perform searches on games
 - To perform searches on sneakers
 - To perform searches on fleet
 - To perform searches on deliveries
 - To perform searches on customers

- To track the status of games
- To track the status of sneakers
- To track the status of orders
- To track the status of customers
- To track the status of deliveries

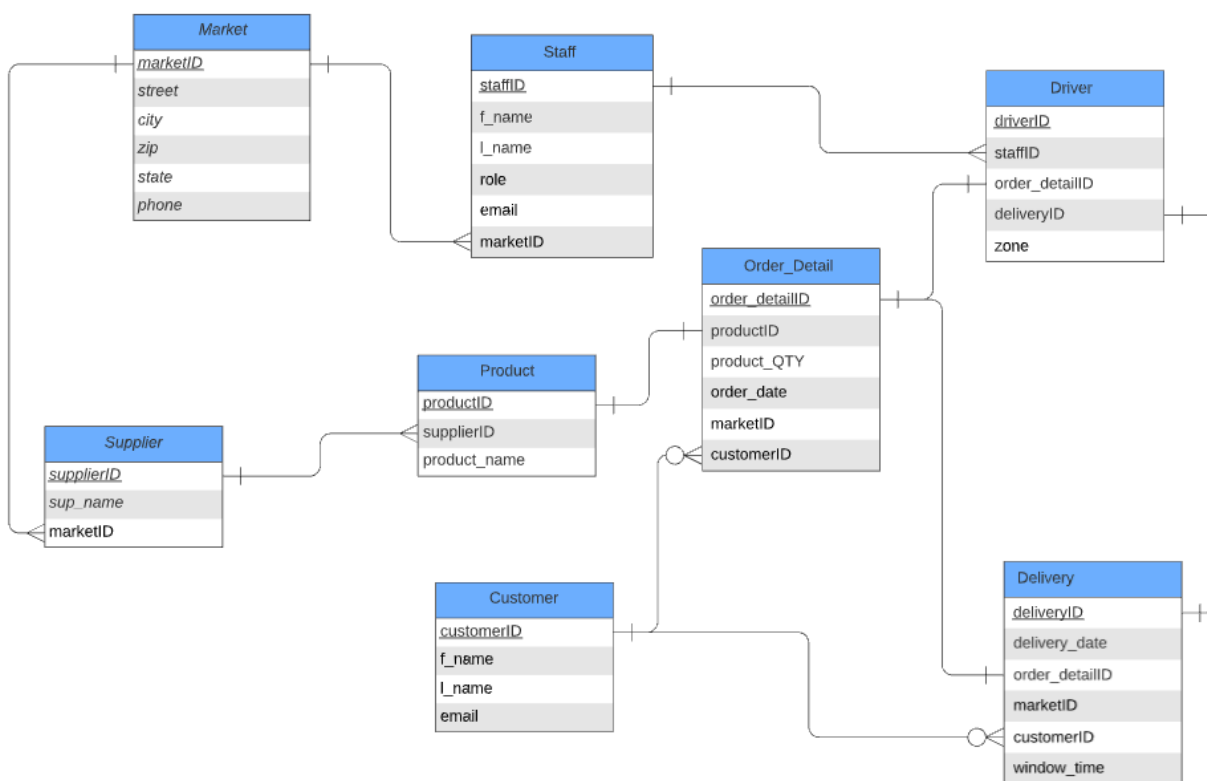
- To report on markets
- To report on staff
- To report on orders
- To report on games
- To report on sneakers
- To report on fleet
- To report on deliveries
- To report on customers

Major User Views

Major User Views

Data	Access Type	Company Director	HR	Billing	Logistics Team	Fleet Manager	Customer
Markets	maintain query report		X				
		X	X				
			X				
Staff	maintain query report		X				
		X	X				
			X				
Orders	maintain query report			X			
		X		X			
				X	X		
Games	maintain query report				X		
		X			X		X
					X		
Sneakers	maintain query report				X		
		X			X		X
					X		
Fleet	maintain query report					X	
		X				X	
					X	X	
Deliveries	maintain query report			X			
		X		X			X
				X	X		X
Customers	maintain query report			X			
		X		X			X
				X	X		X

ER Diagram



Relational Models

Entity: Markets

Table	Attributes	Data Type	Constraints	Default
markets	marketID	INT	NOT NULL	
	street	VARCHAR (50)	NULL	
	city	VARCHAR (50)	NULL	
	zip	VARCHAR (50)	NULL	
	state	VARCHAR (50)	NULL	
	phone	VARCHAR (50)	NULL	

Primary Key = marketID

Candidate Key = marketID

Foreign Key = N/A

marketID → (street, city, zip, state, phone)

Entity: Staff

Table	Attributes	Data Type	Constraints	Default
Staff	staffID	INT	NOT NULL	
	f_name	VARCHAR (50)	NULL	
	l_name	VARCHAR (50)	NULL	
	role	VARCHAR (50)	NULL	
	email	VARCHAR (50)	NULL	
	marketID	INT	NULL	

Primary Key = staffID

Candidate Key = staffID, f_name, l_name, role, marketID

Foreign Key = marketID

staffID → (f_name, l_name, role, email, marketID)

Entity: Orders

Table	Attributes	Data Type	Constraints	Default
Orders	order_detailID	INT	NOT NULL	
	productID	VARCHAR (50)	NULL	
	product_QTY	INT	NULL	
	order_date	VARCHAR (50)	NULL	
	marketID	INT	NULL	
	customerID	INT	NULL	

Primary Key = order_detailID

Candidate Key = order_detailID, productID, product_QTY, order_date, marketID, customerID

Foreign Key = productID, marketID, customerID

order_detailID → (productID, product_QTY, order_date, marketID, customerID)

Entity: Fleet

Table	Attributes	Data Type	Constraints	Default
Fleet	fleetID	INT	NOT NULL	
	make	VARCHAR (50)	NULL	
	model	VARCHAR (50)	NULL	
	vin	INT	NULL	
	plate	VARCHAR (10)	NULL	
	enabled	BOOL	NULL	
	marketID	INT	NULL	

Primary Key = fleetID

Candidate Key = fleetID, make, model, enabled, marketID

Foreign Key = marketID

fleetID → (make, model, vin, plate, enabled, marketID)

Entity: Games

Table	Attributes	Data Type	Constraints	Default
Games	gameID	INT	NOT NULL	
	title	INT	NULL	
	model	VARCHAR (50)	NULL	
	publisher	VARCHAR (50)	NULL	
	sku	VARCHAR (50)	NULL	
	release_date	VARCHAR (25)	NULL	
	price	FLOAT	NULL	
	rating	VARCHAR (25)	NULL	
	platform	VARCHAR (25)	NULL	
	inStock	BOOL	NULL	
	surplus	INT	NULL	
	reserved	INT	NULL	
	supplierID	INT	NULL	

Primary Key = gameID

Candidate Key = gameID, title, publisher, sku, release_date, price, rating, platform, inStock, surplus, reserved, supplierID

Foreign Key = supplierID

gameID → (title, model, publisher, sku, release_date, price, rating, platform, inStock, surplus, reserved)

Entity: Sneakers

Table	Attributes	Data Type	Constraints	Default
Sneakers	shoeID	INT	NOT NULL	
	brand	VARCHAR (50)	NULL	
	name	VARCHAR (50)	NULL	
	style	VARCHAR (50)	NULL	
	color	VARCHAR (50)	NULL	
	size_type	VARCHAR (50)	NULL	
	size	VARCHAR (25)	NULL	
	price	FLOAT	NULL	
	release_year	VARCHAR (25)	NULL	
	inStock	BOOL	NULL	
	surplus	INT	NULL	
	reserved	INT	NULL	
	supplierID	INT	NULL	

Primary Key = shoeID

Candidate Key = shoeID, brand, name, style, color, size_type, size, price, release_year, inStock, surplus, reserved, supplierID

Foreign Key = supplierID

shoeID → (brand, name, style, color, size_type, size, price, release_year, inStock, surplus, reserved, supplierID)

Entity: Delivery

Table	Attributes	Data Type	Constraints	Default
Delivery	deliveryID	INT	NOT NULL	
	delivery_date	VARCHAR (25)	NULL	
	driverID	INT	NULL	
	order_detailID	INT	NULL	
	marketID	BOOL	NULL	
	customerID	INT	NULL	
	window	VARCHAR (25)	NULL	
	isCompleted	BOOL	NULL	

Primary Key = deliveryID

Candidate Key = deliveryID, delivery_date, driverID, order_detailID, marketID, customerID, window, isCompleted

Foreign Key = driverID, order_detailID, marketID, customerID

deliveryID → (delivery_date, driverID, order_detailID, marketID, customerID, window, isCompleted)

Entity: Customer

Table	Attributes	Data Type	Constraints	Default
Customer	customerID	INT	NOT NULL	
	f_name	VARCHAR (50)	NULL	
	l_name	VARCHAR (50)	NULL	
	email	VARCHAR (50)	NULL	

Primary Key = customerID

Candidate Key = customerID, f_name, l_name

Foreign Key = N/A

customerID → (f_name, l_name, email)

Use Cases

Entity: Staff

Use Case Name: Looking up staff

Actor/User: Admin

Steps:

1. User clicks on "Search Staff" button
2. Prompt user to enter the staff's First and Last Name or their Staff ID
3. All information is displayed

```
SELECT*  
FROM Staff  
WHERE firstName = ' ' AND lastName = ' ' OR staffID = ' ' ;
```

Use Case Name: Entering new staff

Actor/User: HR

Steps:

1. User clicks on "Add Staff"
2. Prompt user to enter assigned ID ,first name, last name, role, salary, address
3. User enters information and then clicks on "Submit" button
4. Staff information is saved and user is redirected back to start page

```
INSERT INTO Staff  
VALUES (staffID, firstName, lastName, role, salary, address);
```

Entity: Staff (cont.)

Use Case Name: Removing staff

Actor/User: HR

Steps:

1. User clicks on "Remove Staff" button
2. Prompt user to enter Staff ID
3. Staff information is displayed
4. User clicks "Delete Staff" button
5. Prompt user to confirm action
6. User clicks "Confirm" button or "Cancel" button
7. Information is saved and user is redirected back to start page

DELETE FROM Staff

WHERE staffID = ' ';

Use Case Name: Update Staff

Actor/User: HR

Steps:

1. User clicks on "Update Staff" button
2. Prompt user to enter Staff ID
3. Customer information is displayed
4. User clicks "Edit Customer" button
5. User enters information that needs to change
6. User clicks "Confirm" button or "Cancel" button
7. Information is saved and user is redirected back to start page

UPDATE Staff

**SET staffID = ' ', firstName= ' ', lastName= ' ', role = ' ', salary = ' ', address = ' '
WHERE staffID = ' '**

Entity: Customer

Use Case Name: Looking up customers

Actor/User: Admin & Billing

Steps:

1. User clicks on "Search Customer" button
2. Prompt the user to enter Customer's First and Last Name or Phone Number
3. All information is displayed

SELECT*

FROM Customer

WHERE firstName = ' ' AND lastName = ' ' OR phoneNumber = ' ';

Use Case Name: Add new Customer

Actor/User: Admin & Billing

Steps:

1. User clicks "Add New Customer" button
2. Prompt user to enter Name, Email, Address, & Phone Number
3. User clicks "Add" button
4. Customer ID is auto generated
5. Information is saved and user is redirected back to start page

INSERT INTO Customer

VALUES (customerID, firstName, lastName email, address, phoneNumber);

Use Case Name: Removing Customers

Actor/User: Admin & Billing

Steps:

1. User clicks on "Remove Customer" button
2. Prompt user to enter Customer ID
3. Customer information is displayed

4. User clicks “Delete Customer” button
5. Prompt user to confirm action
6. User clicks “Confirm” button or “Cancel” button
7. Information is saved and user is redirected back to start page

DELETE FROM Customer
WHERE customerID = ‘ ‘;

Use Case Name: Updating Customers

Actor/User: Billing

Steps:

1. User clicks on “Update Customer” button
2. Prompt user to enter Customer ID
3. Customer information is displayed
4. User clicks “Edit Customer” button
5. User enters information that needs to change
6. User clicks “Confirm” button or “Cancel” button
7. Information is saved and user is redirected back to start page

UPDATE Customer
SET customerID = ‘ ‘, firstName = ‘ ‘, lastName = ‘ ‘, email = ‘ ‘, address = ‘ ‘,
phoneNumber = ‘ ‘, order_detailID = ‘ ‘;
WHERE customerID = ‘ ‘;