



# Database Design for University Marketplace

## Team Members:

Bhavana Reddy, Mohammed Safiuddin, Brandt Green, Lucas Fernandez,  
Ramya Desineedi, Suchit Das

Dec 1, 2021

# Agenda

01 Motivation

02 About UT Marketplace

03 Entities and attributes

04 Entity Relationship Diagram

05 Data Strategy

06 DDL and DML Statements

07 Data Lake and Data Warehouse

08 NoSQL Databases

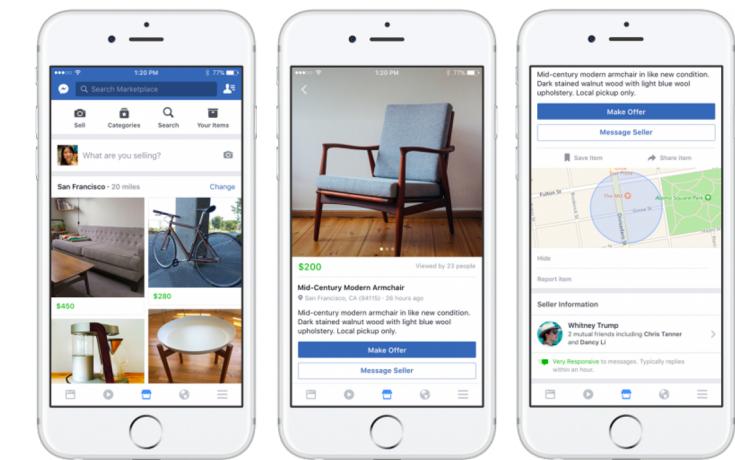
09 Connection to other Courses

10 Learnings

# Motivation

- **74%** of U.S. adults who use Facebook visit the platform at least once per day. (Pew Research Center)
- In 2018, Facebook CEO Mark Zuckerberg claimed Facebook Marketplace attracts over **800 million monthly users** (CNBC)
- Every month, more than **20 million people click** on Facebook's used vehicle listings within the U.S. (Facebook)
- Private sellers have reported a **70% success rate** for selling items on Facebook Marketplace. (CNET)

Then what is the need of having a University specific Marketplace?



# Challenges with existing marketplaces



Existing marketplaces for buying goods is not very trustworthy

Safety of Buyers/Sellers is not guaranteed

Student specific needs not met

# How does UT Marketplace work?

- UT Marketplace is a web / mobile application that allows **Verified UT students and alumni** to buy and sell used products such as electronics, furniture, textbooks, Game Tickets etc.
- As the users are verified and proctored it provides a **safe environment** for both sellers and buyers

Verified Sellers

Verified Buyers

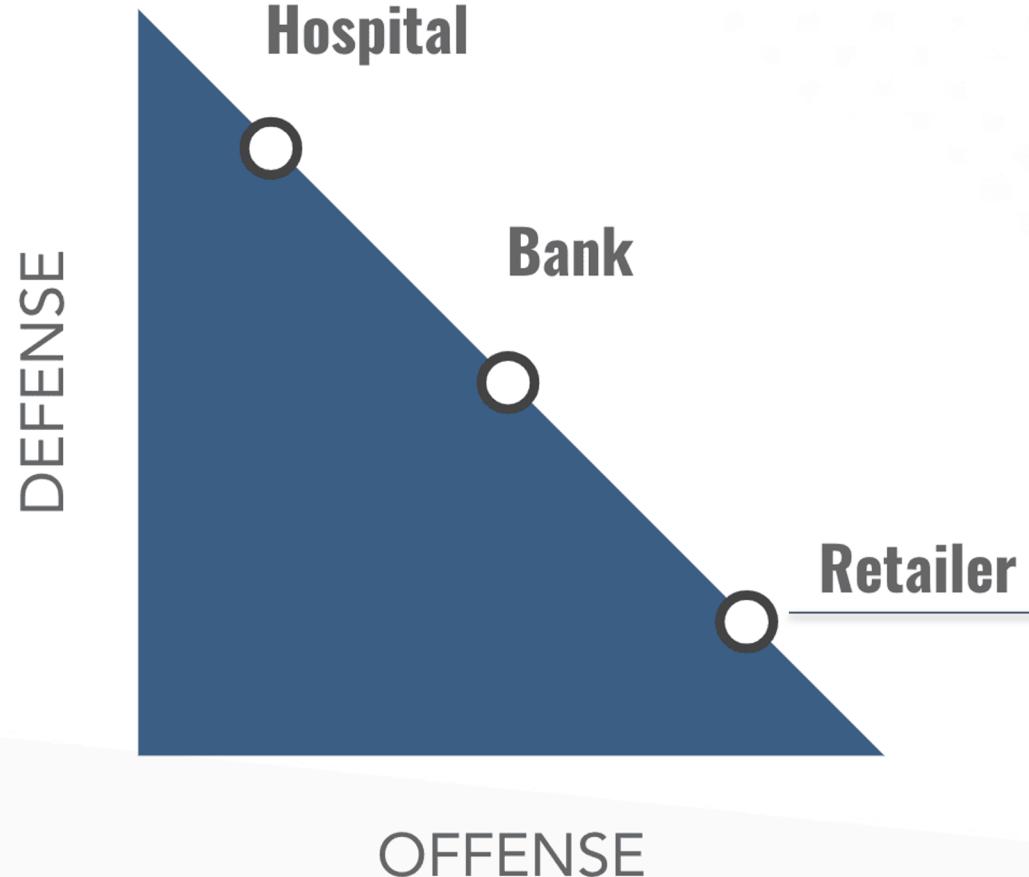
Categorized Products

Private Chat System

Good Vibes

Firsthand sellers  
(in future)

# Data Strategy



Leaning towards offense to improve product recommendations, fraud management

# Entities and Attributes overview

Broadly these are 5 processes that are managed



## User Management

Users  
User details (encrypted)



## Product Management

Product  
Product Categories  
Discount Categories



## Order Management

Cart items  
Order Details  
Order items



## Transaction Management

Payment Details



## Conversation Management

Conversation  
Message  
Report  
User Block List

# User Management Entities

User	
PK	User_ID
	Username
	First_Name
	Last_Name
	Email
	Ut_Eid
	Created_At
	Verified
	Verified_At
	Active
	Gender
	Birthdate
	Seller_Rating
	Buyer_Rating
	Phone
	Address
	Bio
	User_Category
	Image_Url
	Password

Payment Methods	
PK	Payment_Method_ID
FK	UserID
	Cardholder_First_Name
	Cardholder_Mid_Name
	Cardholder_Last_Name
	Card_Type
	Card_Number
	Expiration_Date
	CC_ID
	Billing_Address
	Billing_City
	Billing_State
	Billing_Zip

User Block List	
PK	User_Block_ID
FK	Blocker_ID
FK	Blockee_ID
	Blocked_At
	Active

# Product Management Entities

Product	
PK	Product_ID
FK	Category_ID
FK	Discount_ID
	Name
	Description
	Created_At
	Modified_At
	Status
FK	Seller_ID
	Image_Url
	Price

Product Categories	
PK	Product_Category_ID
	Category_Name
	Parent_Category
	Description
	Active

Discount Categories	
PK	Discount_ID
	Created_At
	Active
	Type
	Percentage_Discount
	Flat_Discount
	Description

# Order Management Entities

Product	
PK	Product_ID
FK	Category_ID
FK	Discount_ID
	Name
	Description
	Created_At
	Modified_At
	Status
FK	Seller_ID
	Image_Url
	Price

Product Cart LK	
FK	Cart_Item_ID
FK	Product_ID

Cart Items	
PK	Cart_Item_ID
FK	User_ID

Order Items	
PK	Order_Item_ID
FK	Order_ID
FK	Product_ID
FK	Conversation_ID

Payment Details	
PK	Payment_ID
FK	Order_ID
	Payment_Amount
	Payment_Status
FK	Payment_Method_ID

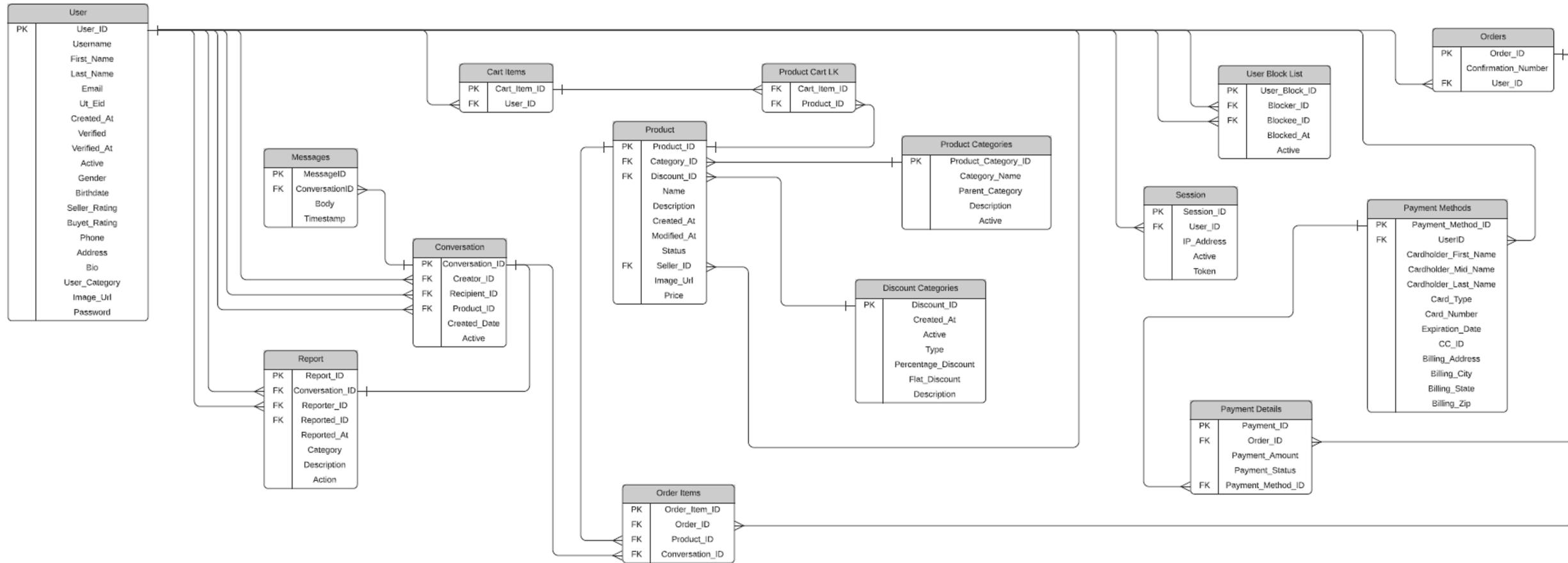
# Conversation Management Entities

Conversation	
PK	Conversation_ID
FK	Creator_ID
FK	Recipient_ID
FK	Product_ID
	Created_Date
	Active

Messages	
PK	MessageID
FK	ConversationID
	Body
	Timestamp

Report	
PK	Report_ID
FK	Conversation_ID
FK	Reporter_ID
FK	Reported_ID
	Reported_At
	Category
	Description
	Action

# Entity Relationship Diagram



# Data Definition Language Statements

## Create Table

```
create table users(
    user_id          number      default user_id_seq.nextval      primary key,
    username         varchar2(100) not null      unique,
    first_name       varchar2(100) not null,
    last_name        varchar2(100) not null,
    email            varchar2(100) not null      unique,
    ut_eid           varchar2(20)  not null      unique,
    created_at       date        default sysdate,
    verified          varchar2(1),  --Y/N
    verified_at      date,
    active            varchar2(1)   not null, -- Y/N/D
    gender            varchar2(1),  -- M/F
    birthdate         date,
    seller_rating     number,
    buyer_rating      number,
    phone             char(12),
    address           varchar2(100),
    bio               varchar2(1500),
    user_category     varchar2(1),  -- S/A/O
    image_url         varchar2(1000),
    password          varchar2(200) not null,
    constraint email_length_checker check (length(email) >= 7),
    constraint email_valid check (email LIKE '%@%'),
    constraint user_verified check (verified IN('Y', 'N')),
    constraint user_active check (active IN('Y', 'N', 'D')),
    constraint user_gender check (gender IN('M', 'F')),
    constraint user_category check (user_category IN('S', 'A', 'O')),
    constraint user_password check (length(password) >= 5)
);
```

# DML, Functions and Procedures

- Highest Rated Sellers
- Number of sellers for each product category
- Product prices sorted low to high for each product category
- Product prices sorted high to low for each product category
- Outstanding reports with no action
- Number of carts with a particular product

```
1 SET SERVEROUTPUT ON;
2
3 CREATE OR REPLACE FUNCTION highest_rated
4 (
5   SELLER_RATING NUMBER
6 )
7 RETURN NUMBER AS HIGHEST_RATED_SELLERS NUMBER;
8
9 BEGIN
10  SELECT USER_ID INTO HIGHEST_RATED_SELLERS
11  FROM USERS
12  WHERE SELLER_RATING IS NOT NULL
13  ORDER BY SELLER_RATING DESC;
14  RETURN HIGHEST_RATED_SELLERS;
15
16 END;
17 /
```

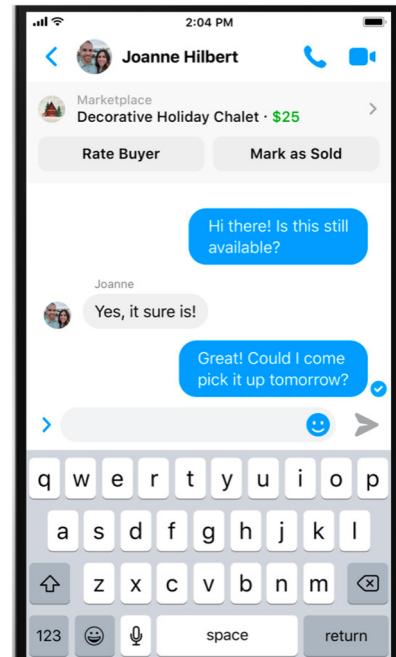
# Unstructured Data

A screenshot of the Facebook Marketplace interface. On the left, there's a sidebar with options like 'Sell Something' highlighted. The main area shows a grid of items for sale, including an L-shaped couch (\$250), an arcade machine (\$400), a grey sofa (\$1), a mountain bike (FREE), a gym equipment (\$60), and a handbag copy (\$FREE). Each item has a small image, its price, and a brief description.

A screenshot of a 'Marketplace appeals' form. It asks for the user's name (First name: Ahmed, Middle name: Maher) and the issue they're reporting. The issue description box contains the text: 'Marketplace icon has suddenly disappeared from my mobile app and I can not use the service anymore. I did not violate Facebook policies by any mean. Please help me restore access to it.' A 'Send' button is at the bottom.

Product Text and Image Data from Sellers

Text and Image Data from Conversation between buyers and sellers



Customer Location

Text and Image Data from concerns reported

Key-value Pairs

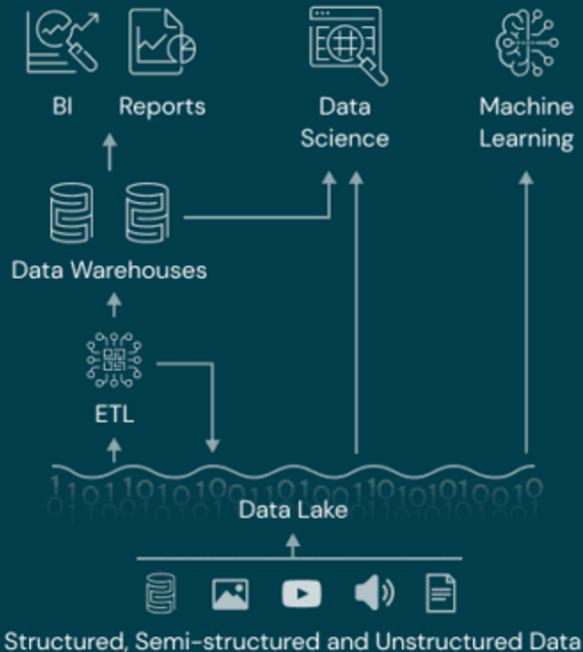


# Data Lake and Warehouse

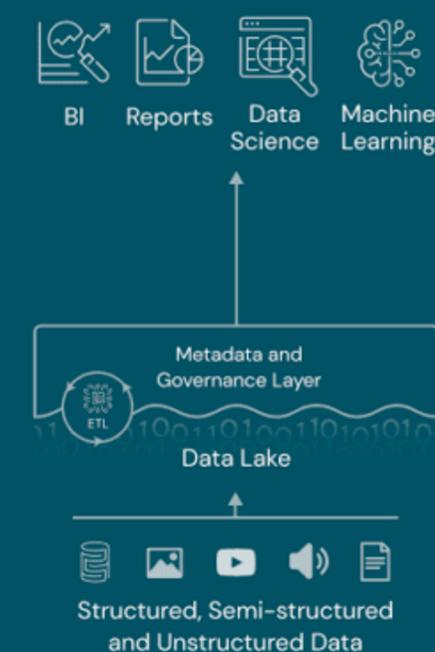
Data Warehouse



Data Lake



Data Lakehouse



Orderliness

Open-mindedness



Business Applications

- Finding out which product categories are seeing growth/decline
- Product popularity for discounts/sales
- Image analytics on products for quality
- Understanding Customer Patterns
- Report conversations data to authorities
- Reporting on fraudulent transactions

# NoSQL-Text Analytics



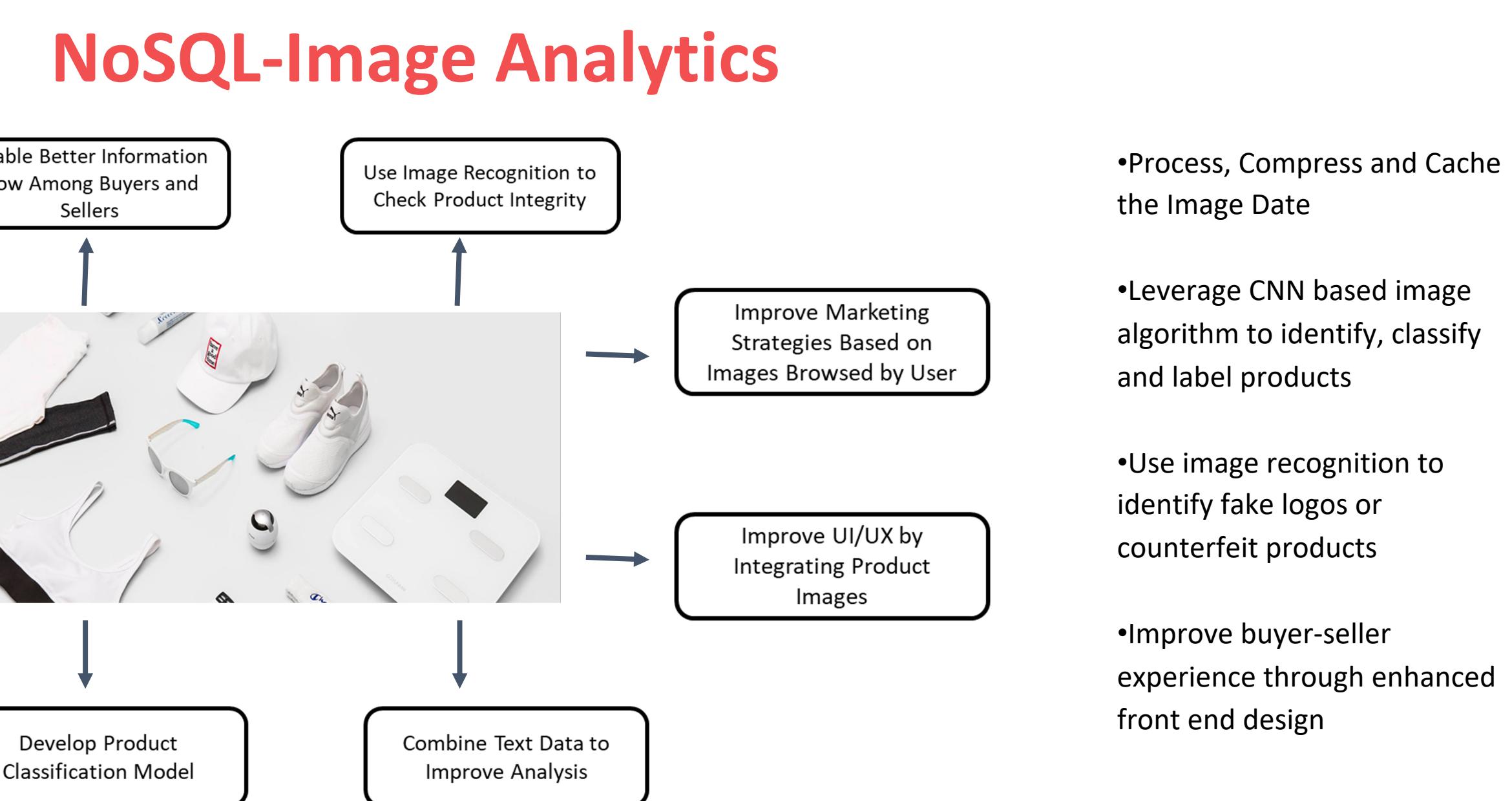
# NoSQL-Image Analytics

Enable Better Information Flow Among Buyers and Sellers

Use Image Recognition to Check Product Integrity

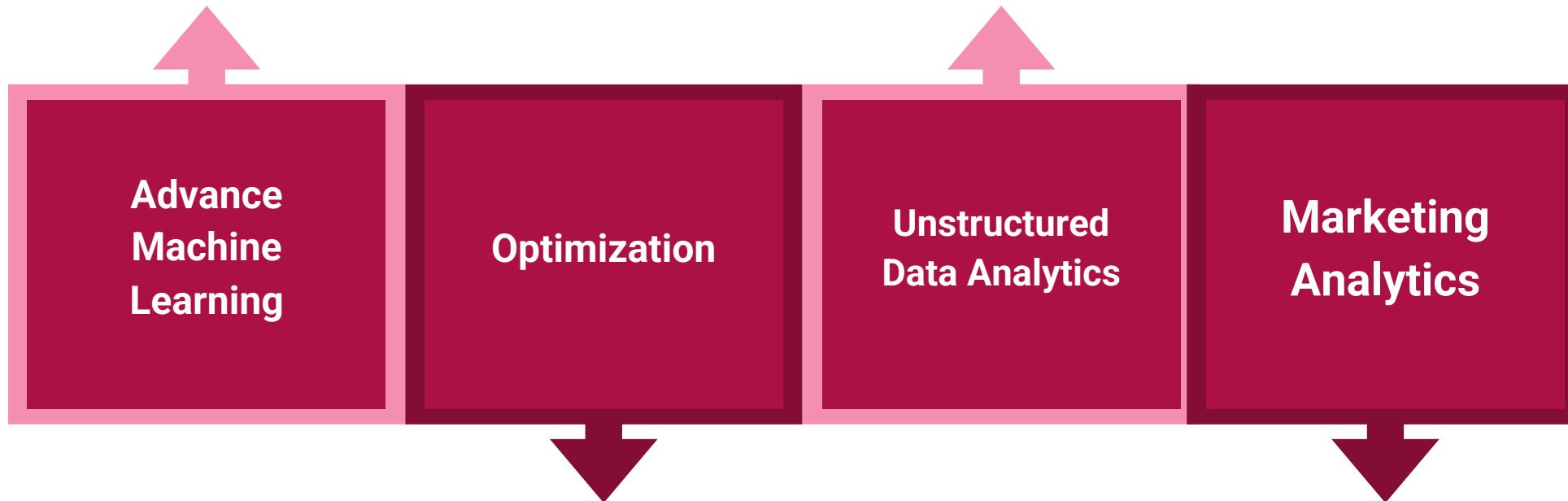
Improve Marketing Strategies Based on Images Browsed by User

Improve UI/UX by Integrating Product Images



# Analysis: Connection to other Courses

Product Image Analytics & Topic Modelling on reported concerns



Product Recommendations based on past purchase history,  
similarity with other users, and from nearest locations

# Learnings & Takeaways

- Difficult to brainstorm future needs accurately enough to create a relational database flexible enough to meet current needs and open to expansion.
- Better to break complex systems into simpler, workable parts
- Create PL/SQL queries and functions for commonly reused queries/transformations
- Growing businesses benefit from understanding structure of their transactions and database that store said transactions
- Mongo looks better after this project because of its flexibility and startup speed
- Database management makes data analytics much easier by keeping data clean

# Thank you!

# Questions?

