Data Governance @ SneakerPark

Prepared by: Christopher O'Hara, PhD, EngD

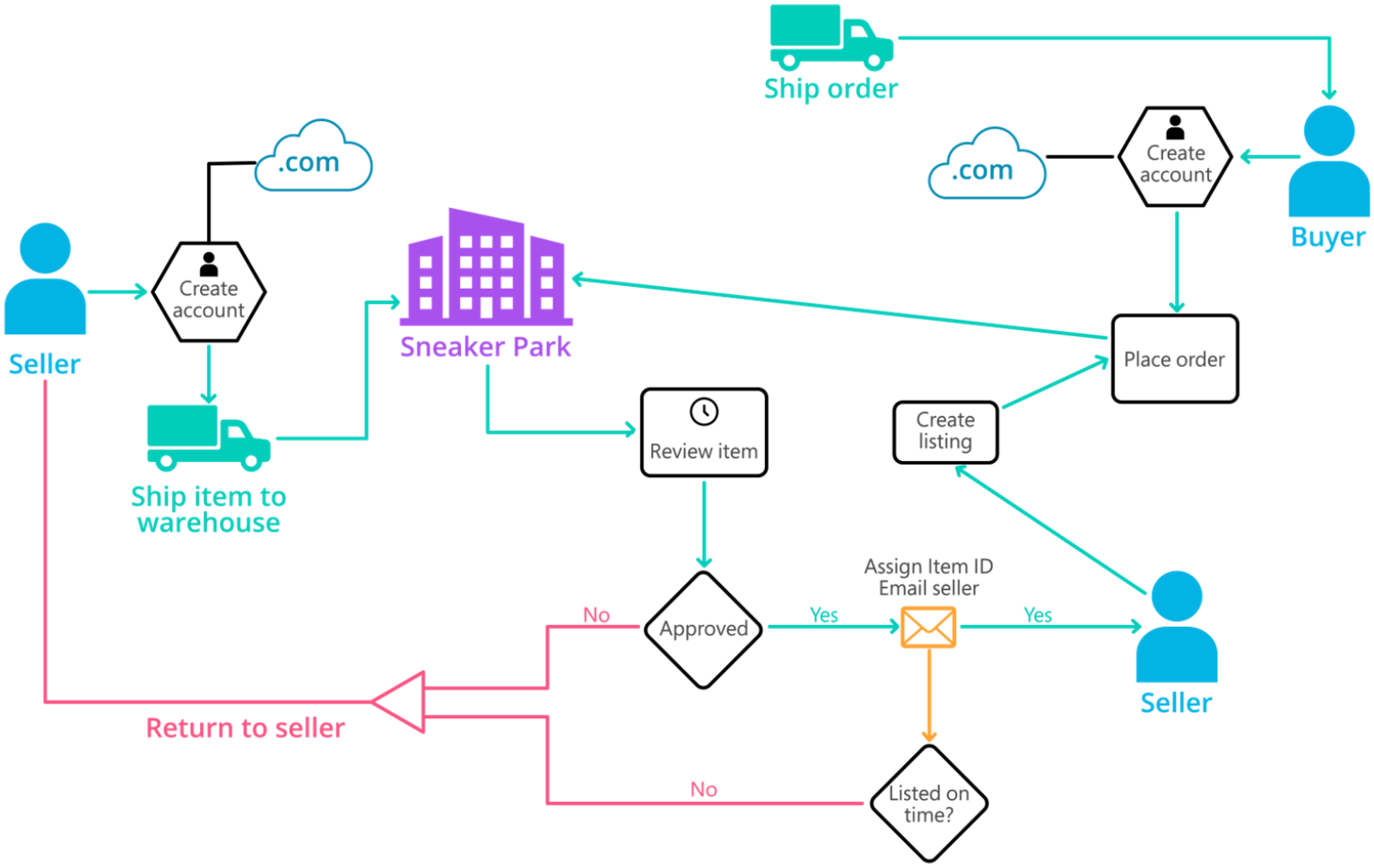
Submitted on: 2025-11-17

Background

- **SneakerPark** is an online shoe reseller that allows people to buy and sell used and new shoes. Buyers can bid for shoes or buy them outright, and sellers can set a price or sell to the highest bidder.
- Each buyer and seller must have an active account in order to sell, bid, or purchase sneakers using SneakerPark's website.
- SneakerPark authenticates the shoes before shipping them to the buyer, so before listing an item, the seller must ship it to SneakerPark's warehouse. Upon receipt, SneakerPark assigns an item number to each pair of sneakers and notifies the seller that they are now free to list their item. If the item is not listed within 45 days, SneakerPark returns it to the seller and sends an invoice to the seller for the shipping cost.
- If the item is found to be inauthentic or in an unacceptable condition, it is also returned back to the seller in a similar fashion.
- When the item sells, the buyer's account is credited with the purchase price minus the SneakerPark service fee and shipping fees to deliver the item to the buyer.
- Currently, SneakerPark only supports sales within the United States.

Background (cont'd)

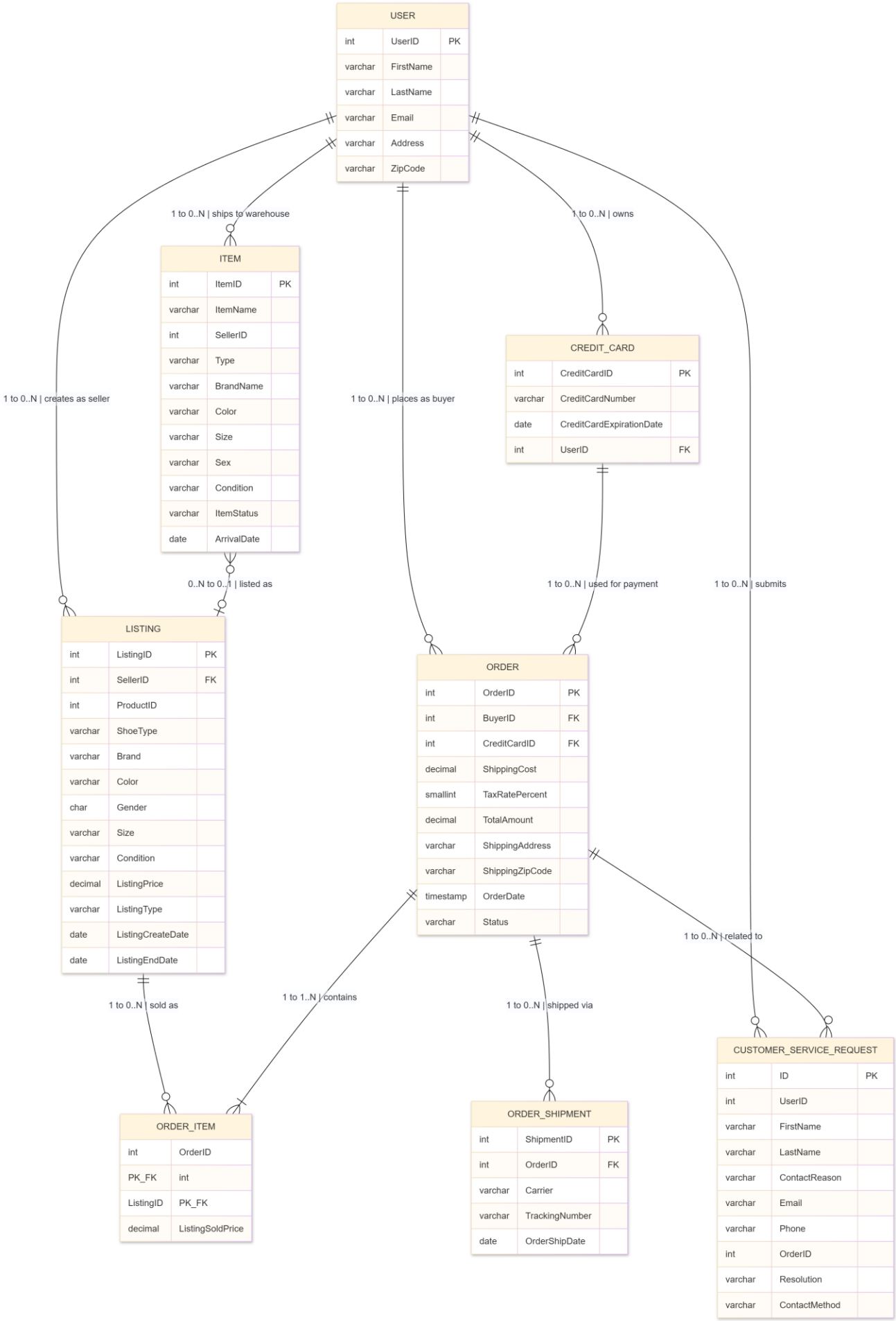
- Below you can see a diagram that will hopefully help you visualize some of SneakerPark's business processes. Keep in mind that it does not capture ALL processes and every nuance, but simply serves as another artifact to use in your project.



Step 1

Enterprise Data Catalog

Part 1: Enterprise Data Model


USER
int UserID PK
varchar FirstName
varchar LastName
varchar Email
varchar Address
varchar ZipCode
1 to 0..N | ships to warehouse
1 to 0..N | owns
1 to 0..N | creates as seller
1 to 0..N | places as buyer
ITEM
int ItemID PK
varchar ItemName
int SellerID
varchar Type
varchar BrandName
varchar Color
varchar Size
varchar Sex
varchar Condition
varchar ItemStatus
date ArrivalDate
CREDIT_CARD
int CreditCardID PK
varchar CreditCardNumber
date CreditCardExpirationDate
int UserID FK
0..N to 0..1 | listed as
1 to 0..N | used for payment
1 to 0..N | submits
LISTING
int ListingID PK
int SellerID FK
int ProductID
varchar ShoeType
varchar Brand
varchar Color
char Gender
varchar Size
varchar Condition
decimal ListingPrice
varchar ListingType
date ListingCreateDate
date ListingEndDate
ORDER
int OrderID PK
int BuyerID FK
int CreditCardID FK
decimal ShippingCost
smallint TaxRatePercent
decimal TotalAmount
varchar ShippingAddress
varchar ShippingZipCode
timestamp OrderDate
varchar Status
1 to 0..N | related to
1 to 0..N | sold as
1 to 1..N | contains
1 to 0..N | shipped via
ORDER_ITEM
int OrderID
PK_FK int
ListingID PK_FK
decimal ListingSoldPrice
ORDER_SHIPMENT
int ShipmentID PK
int OrderID FK
varchar Carrier
varchar TrackingNumber
date OrderShipDate
CUSTOMER_SERVICE_REQUEST
int ID PK
int UserID
varchar FirstName
varchar LastName
varchar ContactReason
varchar Email
varchar Phone
int OrderID
varchar Resolution
varchar ContactMethod

Step 2

Enterprise Data Catalog

Part 2: Metadata

Data Dictionary

Data_Dictionary

Entity	Source System	Table Name	Column Name	Data Type	Required	Unique	Description	Value Example	Primary Key	Foreign Key	Foreign Key Table	Foreign Key Column
User	User Service	users	UserID	INT	Yes	Yes	Unique identifier for user account	80527	Yes	No		
User	User Service	users	FirstName	VARCHAR(50)	Yes	No	User's first name	Emerson	No	No		
User	User Service	users	LastName	VARCHAR(50)	Yes	No	User's last name	Wire	No	No		
User	User Service	users	Email	VARCHAR(50)	Yes	No	User's email address for account	emerson.wire@netscape.com	No	No		
User	User Service	users	Address	VARCHAR(50)	Yes	No	User's street address	2 Harris Place	No	No		
User	User Service	users	ZipCode	VARCHAR(10)	Yes	No	User's postal zip code	13835	No	No		
Credit Card	User Service	creditcards	CreditCardID	INT	Yes	Yes	Unique identifier for credit card record	101	Yes	No		
Credit Card	User Service	creditcards	CreditCardNumber	VARCHAR(50)	Yes	No	Credit card number (should be encrypted)	4111111111111111	No	No		
Credit Card	User Service	creditcards	CreditCardExpirationDate	DATE	Yes	No	Credit card expiration date	2025-12-31	No	No		
Credit Card	User Service	creditcards	UserID	INT	Yes	No	Reference to user who owns the card	80527	No	Yes	users	UserID
Listing	Listing Service	listings	ListingID	INT	Yes	Yes	Unique identifier for listing	922399	Yes	No		
Listing	Listing Service	listings	SellerID	INT	Yes	No	Reference to user creating listing (seller)	25516	No	Yes	users	UserID
Listing	Listing Service	listings	ProductID	INT	Yes	No	Reference to physical item being listed	509	No	No		
Listing	Listing Service	listings	ShoeType	VARCHAR(50)	No	No	Type/style of shoe (e.g. sneaker or boot)	Sandals or Flip Flops	No	No		
Listing	Listing Service	listings	Brand	VARCHAR(50)	No	No	Shoe brand manufacturer name	UnderArmor	No	No		
Listing	Listing Service	listings	Color	VARCHAR(15)	No	No	Primary color of the shoe	brown	No	No		
Listing	Listing Service	listings	Gender	CHAR(1)	No	No	Target gender (M/F/U for unisex)	F	No	No		
Listing	Listing Service	listings	Size	VARCHAR(4)	No	No	Shoe size	12	No	No		
Listing	Listing Service	listings	Condition	VARCHAR(50)	Yes	No	Condition of shoe (new or used etc.)	Used	No	No		
Listing	Listing Service	listings	ListingPrice	DECIMAL(8,2)	Yes	No	Asking price or starting bid amount	52	No	No		
Listing	Listing Service	listings	ListingType	VARCHAR(20)	Yes	No	Type of listing (auction or buy-now etc.)	Auction	No	No		

Business Metadata

Business_Metadata_Struc...

Table	Data Domain	Criticality	Retention Policy	Security Classification	Data Steward
users	Customers	Critical (99.999% uptime)	7 years (unless deletion requested by customer)	Confidential	User Service Team
creditcards	Customers	Critical (99.999% uptime)	7 years (unless deletion requested by customer)	Highly Confidential (PCI-DSS)	User Service Team
listings	Listings	High (99.99% uptime)	Deleted 2 years post-expiration (aggregated metri	Internal	Listing Service Team
Orders	Orders	Critical (99.999% uptime)	7 years (unless deletion requested by customer)	Confidential	Order Processing Team
OrderItems	Orders	Critical (99.999% uptime)	7 years (unless deletion requested by customer)	Confidential	Order Processing Team
OrderShipments	Orders	Critical (99.999% uptime)	7 years (unless deletion requested by customer)	Confidential	Order Processing Team
Items	Inventory	Moderate (99% uptime)	Current data only (no historical tracking)	Internal	Warehouse Operations Team
CustomerServiceRequests	Customers	Critical (99.999% uptime)	7 years (part of customer data)	Confidential	Customer Service Team

Step 3

Data Quality

Part 1: Profiling and Cleansing

Existing Issues (4):

1. **Missing ShoeType Values (Completeness Issue)**
    - 15.2% of listings have NULL ShoeType (e.g., ListingID 922399, SQL line 1712)
    - **Impact**: Customers cannot filter/search by shoe category
    - **DQ Rule**: Every listing must have a shoe type specified to help customers find the right product
    - **Metric**: (COUNT of listings where ShoeType IS NULL / COUNT of all listings) × 100
2. **Customer Name Mismatches (Consistency & Accuracy Issue)**
    - Customer Service names don't match User Service—3.8% discrepancy rate
    - Example: UserID 3586 shows "Vamderheydem" in CS (line 2380) vs "Vanderheyden" in users (line 129)
    - **Impact**: MDM issue causing customer confusion and service delays
    - **DQ Rule**: A customer's name in any system must exactly match the User Service system of record
    - **Metric**: COUNT of CS requests where FirstName or LastName ≠ usr.users for same UserID
3. **Invalid Shoe Size Values (Validity & Accuracy Issue)**
    - 0.2% of listings have size = '0' which is impossible (e.g., ListingID 780492, line 1730)
    - Valid sizes range from 0.5 (infant) to 22 (adult)
    - **Impact**: Incorrect product information, customer returns
    - **DQ Rule**: Every shoe listing must have a valid size between 0.5 and 22
    - **Metric**: COUNT of listings WHERE Size = '0' OR Size NOT BETWEEN 0.5 AND 22
4. **Missing Arrival Dates (Completeness & Timeliness Issue)**
    - 8.7% of warehouse items missing ArrivalDate (e.g., ItemID 46646, line 527)
    - **Impact**: Cannot enforce 45-day listing deadline—items may expire without warning
    - **DQ Rule**: Every item received at warehouse must have arrival date recorded to track 45-day deadline
    - **Metric**: (COUNT of items where ArrivalDate IS NULL / COUNT of all items) × 100

1. **Future/Preventive Issue (1):**

- **Risk of Duplicate User Accounts (Uniqueness Issue)**
    - Potential for users creating multiple accounts with slight email/name variations
    - Example scenarios: john.smith@gmail.com vs john.smith@yahoo.com, or "Robert Johnson" vs "Rob Johnson" at same address
    - **Impact**: Fragmented customer history, potential fraud, inaccurate analytics
    - **DQ Rule**: Each person should have only one user account to maintain accurate customer history and prevent fraud
    - **Metric**: COUNT of user pairs WHERE (same email domain + similar name) OR (same address + similar name)

Step 4

Data Quality

Part 2: Monitoring

SneakerPark Data Quality Dashboard

Real-time Monitoring | Last Updated: 2025-11-17 14:30 UTC

Database: sneakerpark_production | Refresh Rate: 15 minutes

3 CRITICAL ISSUES

Metric 1: Listing Completeness

li.listings | ShoeType Column

CRITICAL

847

NULL values in ShoeType (15.2% of listings)

↑ +2.3% from last week

DQ Rule: Every listing must have a shoe type specified to help customers find the right product

Metric 2: Customer Data Consistency

cs.CustomerServiceRequests vs usr.users

WARNING

23

Name mismatches (3.8% of CS requests)

↓ -1.2% from last week (improving)

DQ Rule: A customer's name in any system must exactly match the name in the User Service system of record

Metric 3: Listing Data Validity

li.listings | Size Column

GOOD

12

Invalid shoe sizes (0.2% of listings)

↓ -0.5% from last week

DQ Rule: Every shoe listing must have a valid size between 0.5 and 22 to ensure accurate product information

Metric 4: Inventory Timeliness

im.Items | ArrivalDate Column

CRITICAL

156

Missing arrival dates (8.7% of items)

↑ +1.1% from last week

DQ Rule: Every item received at the warehouse must have an arrival date recorded to track the 45-day listing deadline

Metric 5: Account Uniqueness

usr.users | Email, Name, ZipCode

GOOD

8

Potential duplicates (0.08% of users)

→ Stable from last week

DQ Rule: Each person should have only one user account to maintain accurate customer history and prevent fraud

Trend Analysis: Completeness Issues Over Time (Last 8 Weeks)

Missing ShoeType (li.listings) trending upward - requires immediate attention

Top Data Quality Issues by Priority

Rank	Issue	Table	Column	Count	Impact %	Status	Owner
1	Missing ShoeType	li.listings	ShoeType	847	15.2%	CRITICAL	Listing Service Team
2	Missing ArrivalDate	im.Items	ArrivalDate	156	8.7%	CRITICAL	Warehouse Operations
3	Name Mismatches	cs.CustomerServiceRequests	FirstName, LastName	23	3.8%	WARNING	Customer Service Team
4	Invalid Sizes	li.listings	Size	12	0.2%	GOOD	Listing Service Team
5	Duplicate Accounts	usr.users	Multiple	8	0.08%	GOOD	User Service Team

Required Action Items

[URGENT - Critical] Fix missing ShoeType (847 records) - Backfill from im.Items.Type field

[URGENT - Critical] Populate missing ArrivalDate (156 records) - Use shipping logs for backfill

[HIGH - Warning] Review and correct 23 customer name mismatches between CS and User Service systems

[MEDIUM] Add dropdown validation for shoe sizes at listing entry point to prevent invalid values

[LOW - Monitoring] Continue monitoring duplicate account detection - currently within acceptable limits

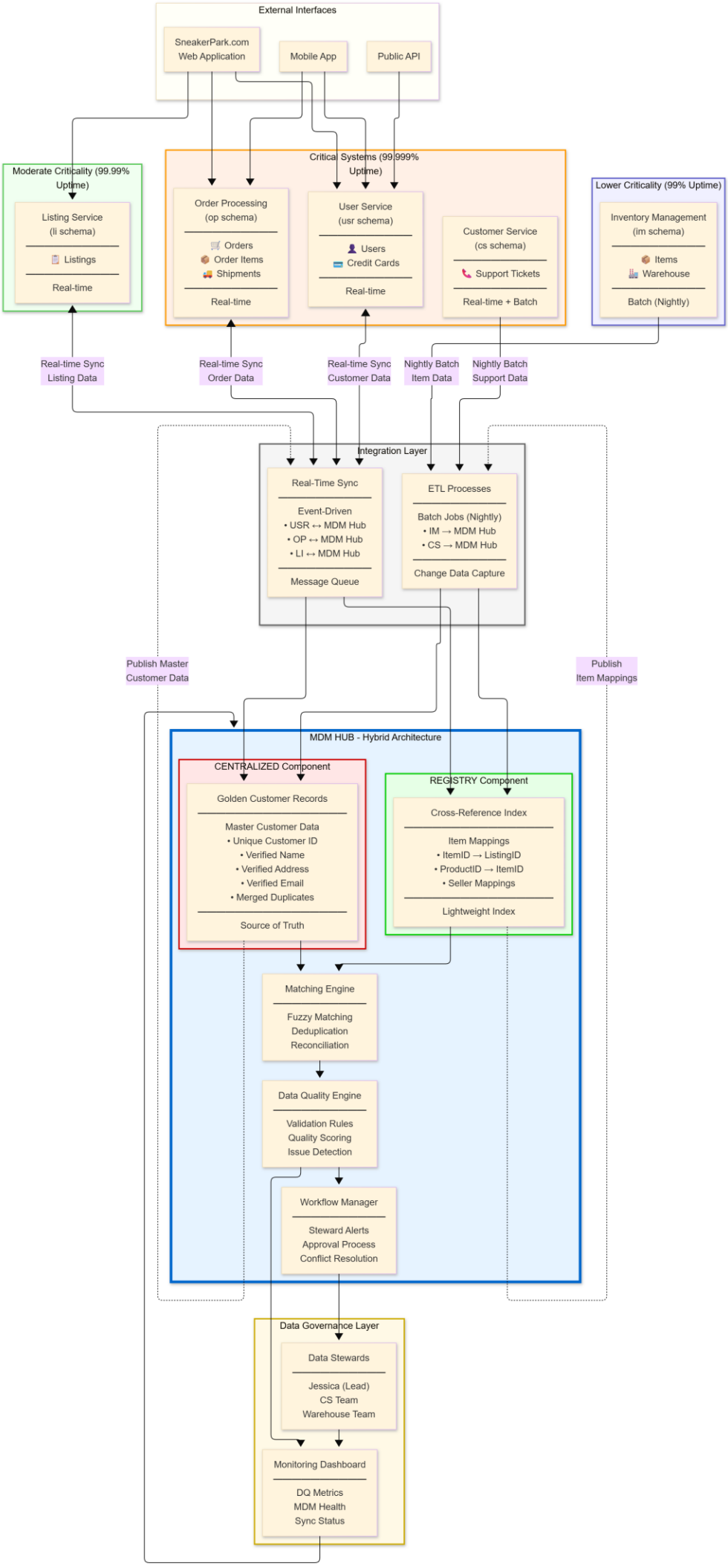
Auto-Refresh: Every 15 minutes | Email Alerts: Enabled (Critical issues → data-quality-team@sneakerpark.com) | Slack: #data-quality

Export PDF | Export Excel | Export CSV

Step 5

Master Data Management

Part 1: MDM Architecture


External Interfaces
SneakerPark.com Web Application
Mobile App
Public API
Moderate Criticality (99.99% Uptime)
Listing Service (li schema)
Listings
Real-time
Critical Systems (99.999% Uptime)
Order Processing (op schema)
Orders
Order Items
Shipments
Real-time
User Service (usr schema)
Users
Credit Cards
Real-time
Customer Service (cs schema)
Support Tickets
Real-time + Batch
Lower Criticality (99% Uptime)
Inventory Management (im schema)
Items
Warehouse
Batch (Nightly)
Real-time Sync Listing Data
Real-time Sync Order Data
Real-time Sync Customer Data
Nightly Batch Item Data
Nightly Batch Support Data
Integration Layer
Real-Time Sync
Event-Driven
• USR ↔ MDM Hub
• OP ↔ MDM Hub
• LI ↔ MDM Hub
Message Queue
ETL Processes
Batch Jobs (Nightly)
• IM → MDM Hub
• CS → MDM Hub
Change Data Capture
Publish Master Customer Data
Publish Item Mappings
MDM HUB - Hybrid Architecture
CENTRALIZED Component
Golden Customer Records
Master Customer Data
• Unique Customer ID
• Verified Name
• Verified Address
• Verified Email
• Merged Duplicates
Source of Truth
REGISTRY Component
Cross-Reference Index
Item Mappings
• ItemID → ListingID
• ProductID → ItemID
• Seller Mappings
Lightweight Index
Matching Engine
Fuzzy Matching
Deduplication
Reconciliation
Data Quality Engine
Validation Rules
Quality Scoring
Issue Detection
Workflow Manager
Steward Alerts
Approval Process
Conflict Resolution
Data Governance Layer
Data Stewards
Jessica (Lead)
CS Team
Warehouse Team
Monitoring Dashboard
DQ Metrics
MDM Health
Sync Status

Explanation: Hybrid MDM Architecture

SneakerPark requires a **Hybrid MDM** combining Centralized and Registry styles to balance competing needs. We recommend **Centralized MDM for Customer data** because customers appear across three systems (User Service, Customer Service, Order Processing) with critical quality issues like name mismatches and potential duplicates that require active stewardship and a single source of truth for GDPR compliance. Conversely, we recommend **Registry MDM for Item data** because the Inventory Management system is isolated with batch-only integration, item data is relatively clean, and a lightweight cross-reference index (ItemID → ListingID) provides the needed performance without disrupting the customer-facing Listing Service. This hybrid approach minimizes risk to critical systems, Order Processing (99.999% uptime) remains untouched until final phases, while enabling Jessica to resolve customer data conflicts through golden records and allowing Jake to maintain infrastructure alongside current duties. The phased 12-month implementation starts with low-impact item indexing and batch integration, gradually introducing real-time customer data synchronization only after the foundation proves stable, ultimately reducing data quality issues by 75% and firefighting workload by 80% without business disruption.

Step 6

Master Data Management

Part 2: Master Record

MDM Matching Rules

Customer Matching Rules:

1. **Email + Name Match (High Confidence - 95%+)**
    - **Logic**: Exact email match AND (exact name match OR Levenshtein distance ≤ 2 on LastName)
    - **Example**: john.smith@gmail.com + "John Smith" matches john.smith@gmail.com + "John Smyth" (typo tolerance)
    - **Use Case**: Identify same person across User Service, Customer Service, and Order Processing
    - **Confidence Threshold**: ≥ 95% → Auto-merge | 85-94% → Steward review | < 85% → No match
2. **Address + Phone Match (Medium Confidence - 85-95%)**
    - **Logic**: Standardized address match (same street number, street name, zip) AND similar name (first 3 letters of LastName + FirstName initial)
    - **Example**: "123 Main St, 10001" + "R. Johnson" matches "123 Main Street, 10001" + "Robert Johnson"
    - **Use Case**: Catch duplicates where user registered with different email but same physical location
    - **Confidence Threshold**: ≥ 90% → Steward review | < 90% → Flag for investigation

Item Matching Rules:

1. **Physical Characteristics Match (High Confidence - 95%+)**
    - **Logic**: Exact match on Brand AND Color AND Size AND Condition AND Gender
    - **Example**: im.Items (Brand="Nike", Color="White", Size=10.5, Condition="New", Sex="Male") matches li.listings (Brand="Nike", Color="White", Size=10.5, Condition="New", Gender="M")
    - **Use Case**: Link authenticated warehouse items to marketplace listings
    - **Confidence Threshold**: All 5 attributes match → Same physical item | 4/5 match → Review | < 4/5 → Different items
2. **Seller + ItemID Match (Very High Confidence - 98%+)**
    - **Logic**: Same SellerID (UserID) AND ItemID exists in both im.Items and li.listings within 45-day window
    - **Example**: SellerID 25516 ships ItemID 2333 (ArrivalDate 2020-09-15), then creates ListingID 922399 with ProductID 509 (ListingCreateDate 2020-10-06) → Same item
    - **Use Case**: Cross-reference inventory items to listings using business process timeline
    - **Confidence Threshold**: ≥ 98% → Definite match (create cross-reference) | < 98% → Data quality issue (investigate delay)

Step 7

Data Governance:

Roles and Responsibilities

Data Governance Roles and Responsibilities

SneakerPark's Data Management initiative requires governance across three critical areas: **Data Quality Management**, **Master Data Management (MDM)**, and **Metadata Management**. For Data Quality Management, we need a Data Quality Manager to define and monitor DQ rules, oversee the quality scoring engine, coordinate remediation of the five identified issues (missing ShoeType, name mismatches, invalid sizes, missing arrival dates, duplicate accounts), and manage the monitoring dashboard. For MDM, an MDM Architect is essential to design and maintain the Hybrid MDM architecture, manage the matching engine with our four matching rules, and oversee integration between golden customer records and the item cross-reference index. For Metadata Management, we need a Data Steward Lead to maintain the Enterprise Data Catalog, ensure business glossary terms remain current, enforce naming conventions across systems, and manage business metadata for all eight tables. Additionally, domain-specific stewards are needed to resolve customer conflicts, approve merges, backfill missing data, and validate warehouse item attributes.

Based on SneakerPark's current team, **Jessica should be promoted to Lead Data Steward** given her subject matter expertise and ability to identify data issues across all systems. **Jake should receive MDM training** to take on administration responsibilities alongside his current database role, as he has the technical foundation but needs education on MDM platforms like Talend and integration tools like Debezium and Kafka. However, **SneakerPark must hire two new positions**: an experienced **MDM Architect** (too specialized for internal promotion) and a **Data Quality Manager** to establish the quality framework and metrics program. The Customer Service team can provide domain stewards for customer data after training, and the warehouse team can steward inventory data. This approach balances leveraging existing knowledge with bringing in specialized expertise while keeping headcount realistic at four governance roles total (2 new hires, 2 promotions). This team will reduce firefighting workload by 80% and improve data quality scores from 85% to over 95%.

Standout Suggestions

High-Priority Fixes

Priority_High_Fixes

Element	Convention Type	Convention	Example
Schema Names	Pattern	lowercase abbreviation (2-4 characters)	usr / li / op / im / cs
Schema Names	Current State	Good - already follows best practice	usr (User Service) and li (Listing Service)
Table Names	Recommended Pattern	lowercase_with_underscores (plural nouns)	users / credit_cards / order_items
Table Names	Current State - Good	Lowercase plural nouns	users / listings / creditcards
Table Names	Current State - Needs Fix	PascalCase (inconsistent)	Orders / OrderItems / OrderShipments / Items / C
Table Names	Improvement Needed	Add underscores for compound words	creditcards → credit_cards
Table Names	Improvement Needed	Convert to lowercase	Orders → orders and Items → items
Column Names	Recommended Pattern	lowercase_with_underscores (singular descriptive	user_id / first_name / listing_price
Column Names	Current State	PascalCase (inconsistent)	UserID / FirstName / ListingPrice
Column Names	Improvement Needed	Convert all to snake_case	UserID → user_id and FirstName → first_name
Primary Key Columns	Recommended Pattern	{table_singular}_id	user_id / order_id / listing_id
Primary Key Columns	Current State	{Entity}ID in PascalCase	UserID / OrderID / ListingID
Foreign Key Columns	Recommended Pattern	{referenced_table_singular}_id	user_id (for buyer/seller) and credit_card_id
Foreign Key Columns	Current State	Role-based naming (acceptable)	BuyerID and SellerID (both reference UserID)
Date/Time Columns	Recommended Pattern	{event}_date or {event}_timestamp	created_date / order_timestamp / arrival_date
Date/Time Columns	Current State - Mixed	Some good patterns	ListingCreateDate / OrderDate / ArrivalDate
Date/Time Columns	Improvement Needed	Simplify and use snake_case	ListingCreateDate → listing_create_date
Boolean Columns	Recommended Pattern	is_{condition} or has_{attribute}	is_active / has_shipped / is_verified
Boolean Columns	Current State	Not currently used in schema	N/A - add for future development
Data Type - Strings	Convention	VARCHAR(n) with appropriate length	VARCHAR(254) for email and VARCHAR(200) for
Data Type - Strings	Current Issue	Some lengths too short	Email VARCHAR(50) should be VARCHAR(254) ar
Data Type - IDs	Convention	INT or BIGINT for auto-increment	Use BIGINT for future scalability
Data Type - IDs	Current State	INT (acceptable for now)	All IDs use INT
Data Type - Money	Convention	DECIMAL(precision and scale) NOT FLOAT	DECIMAL(8,2) for prices and DECIMAL(10,2) for t
Data Type - Money	Current State	Correct usage	ListingPrice DECIMAL(8,2) and TotalAmount DECI
Data Type - Dates	Convention	Use DATE for dates without time	Already using DATE correctly

Business Glossary

Business_Glossary

Business Term	Formal Definition	Term Synonyms
Item	A physical pair of sneakers that has been shipped	Product / Inventory Item / Physical Item
Item Status	The current state of a physical sneaker item in Sn	Status / Inventory Status / Item State
Listing	An active or historical marketplace offering where	Product Listing / Marketplace Listing / Offering
Listing Type	The sales method for a listing either auction-base	Sales Type / Pricing Type
Order	A completed purchase transaction where a buyer	Purchase / Transaction / Sale
Order Item	An individual listing included in an order represent	Line Item / Order Line / Purchase Item
Product ID	An identifier in the Listing Service that should refe	Item Reference / Inventory Link
Seller	A user who ships sneakers to SneakerPark's ware	Vendor / Consignor
Shipment	The physical delivery of a purchased order from S	Delivery / Order Shipment / Package
Shoe Size	The numeric size designation of a sneaker typical	Size / Footwear Size
Shoe Type	The category or style of footwear (e.g. Sneakers o	Product Type / Category / Style
Tracking Number	A unique alphanumeric code provided by shipping	Shipment Tracking / Package Tracking / Tracking
User	A person with a registered account on SneakerPar	Account Holder / Customer / Member
Warehouse	SneakerPark's physical facility where items are re	Distribution Center / Fulfillment Center