Portfolio Assignment:  
Sweet Seoul Korean Bakery

Jenna James  
ICT 4405 Database Design & Implementation  
University of Denver College of Professional Studies  
June 8, 2025  
Cathie Wilson, MS

# **Table of Contents**

[**Table of Contents 1**](#_heading=h.b8auorskysj)

[**Section One: Business Case Abstract 3**](#_heading=)

[Business Overview 3](#_heading=h.4hk6zhvh01r8)

[Business Model 3](#_heading=h.6jaro3tnp0br)

[Customer Base 3](#_heading=h.scbjaosxma5t)

[Current Database Needs 4](#_heading=h.eov6nperl4l4)

[Business Reporting Requirements 4](#_heading=h.a2fbyzsz6ch2)

[**Section Two: Entity Relationship Model and Business Rules 5**](#_heading=h.ir5zbcpipmk)

[Entities 5](#_heading=h.bu4hyg6fu8ni)

[*Core Entities 5*](#_heading=h.ab5jez9epqyt)

[*Product and Inventory Management 6*](#_heading=h.iqod85fhr6t0)

[*Marketing and Promotions 6*](#_heading=h.1qgoru3aeluh)

[*Subscription System 7*](#_heading=h.b3emfcxy4ij8)

[*Customer Experience 8*](#_heading=h.bt0tojanivpt)

[Bi-Directional Business Rules with Cardinality 8](#_heading=h.730n24rze81l)

[*CUSTOMERS Entity Relationships 8*](#_heading=h.oxbh9ljjt7ns)

[*ORDERS Entity Relationships 10*](#_heading=h.i6j3uhdgf6h)

[*PRODUCTS Entity Relationships 12*](#_heading=h.q3nzpufnjbhn)

[*PROMOTIONS Entity Relationships 13*](#_heading=h.r55zee6c8hgh)

[*SUBSCRIPTIONS Entity Relationships 14*](#_heading=h.9gqxw3n6uv5x)

[*MARKETING\_CHANNELS Entity Relationships 15*](#_heading=h.s3fljdfp9d6q)

[*INVENTORY Entity Relationships 15*](#_heading=h.m1j2del2fm6v)

[*Junction Table Relationships 15*](#_heading=h.c2rhrxcc8wwy)

[Assumptions 16](#_heading=h.gnkn0hrve0vl)

[**Section Three: Entity Relationship Diagram and Data Dictionary 25**](#_heading=h.2xhpsvbq2pyc)

[Entity Relationship Diagram 25](#_heading=h.m2wyxqacjjka)

[Data Dictionary 26](#_heading=h.aebjpb7xl9ky)

[**Section Four: Normalization Diagrams and Functional Dependencies 28**](#_heading=h.ln2p50atjrgf)

[Functional Dependencies 28](#_heading=h.1oowktspdrto)

[Normalization Diagram 30](#_heading=h.kxs7ffuh3hf7)

[**Section Five: Create the Database with DDL Statements 30**](#_heading=h.lgjwdpdkepz2)

[Refined DDL Statements and Implementation Confirmation 30](#_heading=h.wbb4dw5mwan)

[Issues with the Prior Model and Resulting SQL 34](#_heading=h.k5rqbgr3qa4c)

[*Implementing the CLOB (Character Large Object) Datatype 34*](#_heading=h.gch873e3k2e4)

[*Design Inconsistencies with the ID Columns 39*](#_heading=h.f8l558b6tegl)

[**Section Six: Populate the Database with DML Statements 41**](#_heading=h.v9eypkqyf4x1)

[**Section Seven: SQL Queries for Business Reporting Requirements 45**](#_heading=h.2vswlc90jugy)

[Requirement One: Product Performance Analytics 46](#_heading=h.j0j8k07noczs)

[*Best-Selling Items Comparison Across Online and In-Store Channels 46*](#_heading=h.au8llq2bgye0)

[*Inventory Turnover Rates 47*](#_heading=h.zd5ntvh6yc05)

[*Seasonal and Limited Edition Product Performance 50*](#_heading=h.4yb5ky9c44i3)

[Requirement Two: Subscription Box Performance Tracking 51](#_heading=h.dctb93p0vy)

[*Subscription Growth and Status Trends Over Time 51*](#_heading=h.iris0r3e0ty1)

[*Subscription Type Performance Analysis 53*](#_heading=h.c7qafbyd16zk)

[*Customer Lifetime Value (CLV) Analysis for Subscriptions 54*](#_heading=h.rnrj5v21e098)

[Requirement Three: Customer Acquisition Cost Analysis 55](#_heading=h.5j3yth21znl2)

[*Marketing Channel Performance Overview 55*](#_heading=h.xbkclop2h0b9)

[*Channel Type Comparison 57*](#_heading=h.b9jzup5o61dh)

[*Customer Acquisition Funnel by Marketing Channel 59*](#_heading=h.n93sdwqcgbdo)

[Requirement Four: Customer Segmentation Reporting 62](#_heading=h.cjhgtrxag959)

[*RFM Customer Segmentation Analysis 62*](#_heading=h.7x2u7brhb54k)

[*Customer Promotion Response Segmentation 63*](#_heading=h.x59x3fk7o3ja)

[*Subscription Customer Segmentation 65*](#_heading=h.9mopq8xk3fv4)

[Requirement Five: Limited-Time Offering Effectiveness Metrics 66](#_heading=h.x03c5g6vqaul)

[*K-Pop/Anime Themed Product Performance Overview 66*](#_heading=h.o74ayonldajx)

[Requirement Six: Cross-Platform Customer Behavior Analysis 68](#_heading=h.vwi3vp46wdwn)

[*Platform-Specific Promotion Effectiveness 68*](#_heading=h.8kyttkq2mmq1)

[*Product Category Preferences by Platform 69*](#_heading=h.4a9m6cud1uaj)

[Requirement Seven: Gift Card Tracking 71](#_heading=h.451d7tqsiy1v)

[*Gift Card Sales Overview Dashboard 71*](#_heading=h.3oc63xfyg4q)

[**Section Eight: User Interface vs. Physical Design 73**](#_heading=h.ue1s041xff41)

[Product Management User Interface 74](#_heading=h.7xlf1y4hdfgv)

[Subscription Management User Interface 76](#_heading=h.ltv9xlft2q2h)

[Orders Management User Interface 78](#_heading=h.tkmbecdhfgy8)

[**Section Nine: Considerations of Maintenance and Evolution 81**](#_heading=h.38hexe3fz2kt)

[Scalability and Performance Optimization 81](#_heading=h.nj5x6tl1ocys)

[Enhanced Analytics and Machine Learning Integration 82](#_heading=h.ch4bnuebkq8t)

[Customer Experience Personalization 82](#_heading=h.vbacmkb1roc4)

[Regulatory Compliance and Data Privacy 83](#_heading=h.hh9pz9owkaae)

[Project Reflection 83](#_heading=h.oatf2qhjaakx)

[**References 85**](#_heading=h.9c2qlhwsbrtp)

# **Section One: Business Case Abstract**

## Business Overview

Sweet Seoul is a specialty Korean bakery located in Denver, Colorado, offering authentic Korean baked goods, snacks, and confections to the local community. The business currently operates from a single brick-and-mortar location but maintains a robust e-commerce platform to reach customers beyond its immediate neighborhood. Plans are underway to open a second physical location within the next 12 months to meet growing demand. This expansion is tentative upon the comparative analysis between in-store and e-commerce sales. They would like to review the performance of promotional offers across both channels. The business may choose to expand their e-commerce endeavors should this channel prove more lucrative.

## Business Model

The bakery generates revenue through multiple streams: in-store retail sales, online orders with home delivery options, and a popular monthly mystery subscription box featuring curated Korean treats. The mystery box subscription service provides a stable monthly revenue stream and introduces customers to new products. Sweet Seoul sources products internationally, emphasizing authentic Korean ingredients and seasonal specialties. The business differentiates itself by offering limited-time products tied to K-Pop bands, manga, and anime franchises, creating urgency and collectability among its target market.

## Customer Base

Sweet Seoul primarily targets teenagers, young adults, and college students with a disposable income for specialty food items in the $15-20 range. Secondary audiences include family members and friends purchasing gifts. Rather than focusing exclusively on Korean expatriates, the bakery appeals to food enthusiasts and curious locals seeking novel culinary experiences. The average transaction value ranges from $15-30, with a growing segment of loyal customers making weekly purchases and subscribing to the monthly mystery box.

## Current Database Needs

As Sweet Seoul prepares for expansion, its current data management system is inadequate for tracking cross-channel customer behavior, inventory from international suppliers, and promotion effectiveness. A comprehensive database system would enhance customer relationship management and support data-driven decision-making for the planned second location.

## Business Reporting Requirements

1. Product performance analytics comparing best-selling items across online and in-store channels, with metrics on inventory turnover rates to identify fast-moving versus slow-moving products.
2. Subscription box performance tracking, including month-over-month subscriber growth, retention rates, and comparison between mystery box value versus average in-store purchase value.
3. Customer acquisition cost analysis by marketing channel, with attribution tracking for social media (TikTok, Instagram, Facebook) and local advertising efforts (billboards, vehicle wraps, events).
4. Customer segmentation reporting to identify purchasing patterns and preferences across platforms, supporting personalized product recommendations and targeted promotions.
5. Limited-time offering effectiveness metrics, particularly for products tied to K-Pop, manga, and anime promotions, including sales velocity and customer engagement metrics.
6. Cross-platform customer behavior analysis comparing online versus in-store purchase patterns, repeat customer rates, and platform-specific promotions effectiveness to inform expansion decisions.
7. Gift card tracking system to monitor sales and redemption patterns across both physical and digital platforms.

# **Section Two: Entity Relationship Model and Business Rules**

## Entities

### *Core Entities*

1. **CUSTOMERS**  
   **customer\_id** (PK) - Generated by Default as Identity  
   first\_name, last\_name, email (unique), phone  
   join\_date, address, city, state, postal\_code  
   is\_online\_customer, is\_in\_store\_customer (NUMBER(1) flags)  
   loyalty\_points, loyalty\_status, preferred\_communication\_method  
   created\_at, updated\_at (TIMESTAMP fields)
2. **ORDERS**  
   **order\_id** (PK) - Generated by Default as Identity  
   **customer\_id** (FK to CUSTOMERS)  
   order\_date, subtotal, tax\_amount, shipping\_cost, discount\_amount  
   total\_amount (calculated field), order\_status, payment\_method  
   **marketing\_channel\_id** (FK to MARKETING\_CHANNELS, nullable)  
   platform ("online"/"in-store"), is\_gift (NUMBER(1) flag)  
   **gift\_card\_id** (FK to GIFT\_CARDS, nullable)  
   shipping\_address, billing\_address (VARCHAR2(500))  
   created\_at, updated\_at
3. **ORDER\_STATUS\_HISTORY**  
   **history\_id** (PK) - Generated by Default as Identity  
   **order\_id** (FK to ORDERS)  
   previous\_status (nullable), new\_status  
   status\_change\_date (TIMESTAMP), updated\_by, change\_reason  
   created\_at, updated\_at
4. **ORDER\_ITEMS**  
   **order\_item\_id** (PK) - Generated by Default as Identity  
   **order\_id** (FK to ORDERS)  
   **product\_id** (FK to PRODUCTS)  
   quantity, unit\_price, discount (item-level)  
   created\_at, updated\_at

### *Product and Inventory Management*

1. **PRODUCTS**  
   **product\_id** (PK) - Generated by Default as Identity  
   product\_name (unique), description (VARCHAR2(4000))  
   category, origin, base\_price  
   seasonal, expiry\_date, limited\_edition (NUMBER(1) flags)  
   kpop\_anime\_tie\_in (VARCHAR2(100), nullable)  
   online\_exclusive, in\_store\_exclusive (mutually exclusive flags)  
   min\_inventory\_level  
   created\_at, updated\_at
2. **INVENTORY**  
   **inventory\_id** (PK) - Generated by Default as Identity  
   **product\_id** (FK to PRODUCTS, unique)  
   current\_stock, reorder\_point, min\_stock\_level, max\_stock\_level  
   last\_restock\_date, expected\_arrival  
   storage\_requirements (CLOB)  
   created\_at, updated\_at

### *Marketing and Promotions*

1. **MARKETING\_CHANNELS**  
   **marketing\_channel\_id** (PK) - Generated by Default as Identity  
   channel\_name (unique), channel\_type, specific\_platform  
   customer\_acquisition\_cost, conversion\_rate  
   created\_at, updated\_at
2. **PROMOTIONS**  
   **promotion\_id** (PK) - Generated by Default as Identity  
   promotion\_name, description (VARCHAR2(4000))  
   start\_date, end\_date, discount\_amount, discount\_type  
   platform ("online"/"in-store"/"both")  
   kpop\_anime\_reference, promotion\_type, min\_purchase  
   is\_gift\_card\_promo (NUMBER(1) flag)  
   created\_at, updated\_at
3. **CUSTOMER\_PROMOTION**  
   **customer\_promotion\_id** (PK) - Generated by Default as Identity  
   **customer\_id** (FK to CUSTOMERS)  
   **promotion\_id** (FK to PROMOTIONS)  
   is\_used (NUMBER(1) flag), date\_offered, expiry\_date  
   times\_used, usage\_limit  
   created\_at, updated\_at
4. **PRODUCT\_PROMOTIONS**  
   **product\_promotion\_id** (PK) - Generated by Default as Identity  
   **product\_id** (FK to PRODUCTS)  
   **promotion\_id** (FK to PROMOTIONS)  
   override\_discount\_amount (nullable)  
   effective\_start\_date, effective\_end\_date (nullable)  
   created\_at, updated\_at

### *Subscription System*

1. **SUBSCRIPTIONS**  
   **subscription\_id** (PK) - Generated by Default as Identity  
   **customer\_id** (FK to CUSTOMERS)  
   start\_date, next\_delivery\_date, end\_date (nullable)  
   subscription\_status, monthly\_price, billing\_cycle\_months  
   subscription\_type ("Standard"/"Premium")  
   is\_gift, auto\_renewal (NUMBER(1) flags)  
   delivery\_address (CLOB)  
   created\_at, updated\_at
2. **SUBSCRIPTION\_ITEMS**  
   **subscription\_item\_id** (PK) - Generated by Default as Identity  
   **subscription\_id** (FK to SUBSCRIPTIONS)  
   **product\_id** (FK to PRODUCTS)  
   delivery\_date, quantity, delivery\_status  
   created\_at, updated\_at

### *Customer Experience*

1. **CUSTOMER\_PREFERENCES**  
   **preference\_id** (PK) - Generated by Default as Identity  
   **customer\_id** (FK to CUSTOMERS, unique)  
   flavor\_preference, product\_category\_preference (CLOB fields)  
   subscription\_interest (NUMBER(1) flag)  
   kpop\_anime\_interest (VARCHAR2(1000))  
   created\_at, updated\_at
2. **FEEDBACK**  
   **feedback\_id** (PK) - Generated by Default as Identity  
   **customer\_id** (FK to CUSTOMERS)  
   **order\_id** (FK to ORDERS, nullable for general feedback)  
   **product\_id** (FK to PRODUCTS, nullable for general feedback)  
   rating (1-5), review\_text (VARCHAR2(4000))  
   feedback\_date, is\_verified\_purchase (NUMBER(1) flag)  
   created\_at, updated\_at
3. **GIFT\_CARDS**  
   **gift\_card\_id** (PK) - Generated by Default as Identity  
   **issuer\_customer\_id** (FK to CUSTOMERS)  
   **recipient\_customer\_id** (FK to CUSTOMERS, nullable)  
   gift\_card\_code (unique), initial\_amount, current\_balance  
   issue\_date, expiry\_date, is\_redeemed (NUMBER(1) flag)  
   created\_at, updated\_at

## Bi-Directional Business Rules with Cardinality

### *CUSTOMERS Entity Relationships*

1. **CUSTOMERS to ORDERS (1:N)**Cardinality: One-to-Many  
   One customer can place zero or many orders  
   Each order must be placed by exactly one customer  
   *Business Rule: Customer deletion restricted if active orders exist  
   Referential Integrity: CASCADE DELETE disabled to preserve order history*
2. **CUSTOMERS to SUBSCRIPTIONS (1:N)**  
   Cardinality: One-to-Many  
   One customer can have zero or many subscriptions  
   Each subscription must belong to exactly one customer  
   *Business Rule: Customers can have multiple subscription types simultaneously  
   Gift Logic: Gift subscriptions link issuer and recipient customers*
3. **CUSTOMERS to CUSTOMER\_PREFERENCES (1:0..1)**  
   Cardinality: One-to-Zero or One  
   One customer can have zero or one customer preferences record  
   Each customer preferences record must belong to exactly one customer  
   *Business Rule: Preferences auto-created on first subscription or feedback  
   CLOB Storage: Flexible preference data using Oracle CLOB field*s
4. **CUSTOMERS to FEEDBACK (1:N)**  
   Cardinality: One-to-Many  
   One customer can provide zero or many feedback records  
   Each feedback record must be provided by exactly one customer  
   *Business Rule: Verified purchase flag set automatically for order-linked feedback  
   Rating System: 1-5 star scale with review text*
5. **CUSTOMERS to CUSTOMER\_PROMOTION (1:N)**  
   Cardinality: One-to-Many  
   One customer can be assigned zero or many promotional offers  
   Each customer promotion assignment must belong to exactly one customer  
   *Business Rule: Individual promotion tracking with usage limits and expiry dates  
   Usage Tracking: times\_used increments automatically with each redemption*
6. **CUSTOMERS to GIFT\_CARDS (as issuer) (1:N)**  
   Cardinality: One-to-Many  
   One customer can issue zero or many gift cards  
   Each gift card must be issued by exactly one customer  
   *Business Rule: Issuer tracked for relationship management and marketing  
   Revenue Recognition: Immediate recognition on gift card purchase*
7. **CUSTOMERS to GIFT\_CARDS (as recipient) (1:N)**  
   Cardinality: One-to-Many (Optional)  
   One customer can receive zero or many gift cards  
   Each gift card may be received by zero or one customer (nullable for physical cards)  
   *Business Rule: Physical gift cards may have null recipient until first use  
   Transfer Logic: Gift cards can be transferred between customers*

### *ORDERS Entity Relationships*

1. **ORDERS to ORDER\_ITEMS (1:N)**Cardinality: One-to-Many  
   One order must contain one or many order items  
   Each order item must belong to exactly one order  
   *Business Rule: Orders cannot exist without at least one order item  
   Inventory Impact: Stock reduced when order moves to "Processing" status*
2. **ORDERS to ORDER\_STATUS\_HISTORY (1:N)**  
   Cardinality: One-to-Many  
   One order must have one or many status history records  
   Each order status history record must belong to exactly one order  
   *Audit Trail: Automatic logging of all status changes with timestamps  
   Business Rule: Initial status record created automatically on order insertion*
3. **ORDERS to MARKETING\_CHANNELS (N:0..1)**  
   Cardinality: Many-to-Zero or One  
   Each order may be attributed to zero or one marketing channel  
   One marketing channel can be attributed to zero or many orders  
   *Attribution Logic: Used for ROI analysis and marketing performance tracking  
   Business Rule: In-store orders may not have marketing channel attribution*
4. **ORDERS to GIFT\_CARDS (N:0..1)**  
   Cardinality: Many-to-Zero or One  
   One order can redeem zero or one gift card  
   One gift card can be redeemed across zero or many orders (partial redemptions)  
   *Balance Logic: Automatic balance reduction and redemption status updates  
   Validation: Gift card balance and expiry checked before order processing*
5. **ORDERS to FEEDBACK (1:N)**  
   Cardinality: One-to-Many  
   One order can generate zero or many feedback records  
   Each order-specific feedback record must belong to exactly one order  
   *Business Rule: Customers can provide feedback on overall order and individual products  
   Verification: Order-linked feedback automatically marked as verified purchase*

### *PRODUCTS Entity Relationships*

1. **PRODUCTS to ORDER\_ITEMS (1:N)**Cardinality: One-to-Many  
   One product can be included in zero or many order items  
   Each order item must reference exactly one product  
   *Business Rule: Products cannot be deleted if referenced in order history  
   Pricing: unit\_price captured at time of order for historical accuracy*
2. **PRODUCTS to SUBSCRIPTION\_ITEMS (1:N)**  
   Cardinality: One-to-Many  
   One product can be featured in zero or many subscription items  
   Each subscription item must feature exactly one product  
   *Curation Logic: Products selected based on customer preferences and K-pop/anime interests  
   Delivery Tracking: Individual delivery status for each subscription item*
3. **PRODUCTS to INVENTORY (1:1)**  
   Cardinality: One-to-One  
   One product must be tracked in exactly one inventory record  
   Each inventory record must track exactly one product  
   *Business Rule: Inventory record automatically created when product is added  
   Stock Management: Real-time tracking with reorder alerts and storage requirements*
4. **PRODUCTS to PRODUCT\_PROMOTIONS (1:N)**  
   Cardinality: One-to-Many  
   One product may be featured in zero or many product promotions  
   Each product promotion must feature exactly one product  
   *Override Logic: Product-specific discounts can override general promotion rates  
   Date Logic: Effective date ranges can differ from parent promotion dates*
5. **PRODUCTS to FEEDBACK (1:N)**  
   Cardinality: One-to-Many  
   One product can receive zero or many feedback records  
   Each product-specific feedback record must reference exactly one product  
   *Rating Aggregation: Average ratings calculated for product recommendations  
   Business Rule: Both general and order-specific product feedback allowed*

### *PROMOTIONS Entity Relationships*

1. **PROMOTIONS to PRODUCT\_PROMOTIONS (1:N)**Cardinality: One-to-Many  
   One promotion may include zero or many product-specific overrides  
   Each product promotion must belong to exactly one promotion  
   *Business Rule: General promotions can apply to all products or be product-specific  
   Override Logic: Product promotions can specify different discount amounts or dates*
2. **PROMOTIONS to CUSTOMER\_PROMOTION (1:N)**  
   Cardinality: One-to-Many  
   One promotion can be assigned to zero or many customers  
   Each customer promotion assignment must belong to exactly one promotion  
   *Assignment Logic: Promotions must be individually assigned to customers  
   Usage Control: Expiry dates, usage limits, and redemption tracking per customer*

### *SUBSCRIPTIONS Entity Relationships*

1. **SUBSCRIPTIONS to SUBSCRIPTION\_ITEMS (1:N)**  
   Cardinality: One-to-Many  
   One subscription must include one or many subscription items (across time)  
   Each subscription item must belong to exactly one subscription  
   *Curation Logic: Items selected based on subscription type and customer preferences  
   Delivery Scheduling: Items scheduled across multiple delivery dates*
2. **SUBSCRIPTIONS to CUSTOMERS (N:1)**  
   Cardinality: Many-to-One  
   Multiple subscriptions can belong to one customer  
   Each subscription must belong to exactly one customer  
   *Business Rule: Customers can have multiple active subscriptions of different types  
   Gift Subscriptions: Special handling for gift subscription delivery and billing*

### *MARKETING\_CHANNELS Entity Relationships*

1. **MARKETING\_CHANNELS to ORDERS (1:N)**  
   Cardinality: One-to-Many  
   One marketing channel can be attributed to zero or many orders  
   Each order may be attributed to zero or one marketing channel  
   *Attribution Logic: Used for ROI calculation and marketing budget allocation  
   Performance Tracking: Conversion rates and customer acquisition costs tracked per channel*

### *INVENTORY Entity Relationships*

1. **INVENTORY to PRODUCTS (1:1)**  
   Cardinality: One-to-One (Reverse of PRODUCTS to INVENTORY)  
   Each inventory record tracks exactly one product  
   Each product must have exactly one inventory record  
   *Stock Automation: Automatic updates via triggers when orders change status  
   Alert System: Reorder alerts when stock hits minimum levels*

### *Junction Table Relationships*

1. **CUSTOMER\_PROMOTION Entity**  
   Bridges: CUSTOMERS (N:1) and PROMOTIONS (N:1)  
   Enables many-to-many relationship between customers and promotions  
   Tracks individual usage, limits, and expiry per customer-promotion pair  
   *Unique Constraint: One record per customer-promotion combination*
2. **PRODUCT\_PROMOTIONS Entity**  
   Bridges: PRODUCTS (N:1) and PROMOTIONS (N:1)  
   Enables many-to-many relationship between products and promotions  
   *Allows product-specific discount overrides and effective date ranges  
   Business Logic: Supports complex promotional strategies*
3. **ORDER\_STATUS\_HISTORY Entity**  
   References: ORDERS (N:1)  
   Provides complete audit trail for order lifecycle  
   Tracks status changes with timestamps and responsible users  
   *Immutable Records: History records cannot be modified, only appended*

## Assumptions

These assumptions form the foundation for Sweet Seoul Korean Bakery's database system. This system is designed with Oracle 19c as the release available in our lab environment. Hence, this is the version of Oracle where the aforementioned entities and relationships will be implemented. With these assumptions in mind we are able to ensure robust support for the bakery’s unique business model combining traditional bakery operations with modern culture, subscription services, and a means of sophisticated customer relationship management (Atzeni and Parker 1982).

1. **Customer Identity Management:**
   1. Unique Email Addresses: Each customer must have a unique email address serving as primary identifier
   2. Multi-Channel Shopping: Customers can be both online and in-store shoppers (not mutually exclusive)
   3. Loyalty Program: Available to all customers with automatic point accumulation and tier progression
   4. Loyalty Tiers: Standard (0-199), Silver (200-399), Gold (400-599), Platinum (600+) points
   5. Communication Preferences: Customers can specify preferred contact method (Email, SMS, Phone, Mail)
   6. Data Integrity: Customer deletion restricted if active orders, subscriptions, or gift cards exist
2. **Order Processing:**
   1. Customer Association: All orders are associated with exactly one customer (no guest orders)
   2. Multi-Channel Orders: Orders can be placed either online or in-store with platform-specific handling
   3. Financial Breakdown: Orders track subtotal, tax\_amount, shipping\_cost, and discount\_amount separately
   4. Total Calculation: total\_amount = subtotal + tax\_amount + shipping\_cost - discount\_amount (enforced by constraints)
   5. Status Tracking: Orders can be tracked through multiple status changes with complete audit trail
   6. Order Items Requirement: Each order must have at least one order item
   7. Gift Orders: Orders can be marked as gifts with special handling and recipient information
   8. Payment Methods: Support for Credit Card, Debit Card, PayPal, Gift Card, Cash, Apple Pay
3. **Product Management:**
   1. Inventory Relationship: Each product must have corresponding inventory information (1:1 relationship)
   2. Product Categories: K-pop Albums, Anime Figures, Merchandise, Accessories, Collectibles, Apparel
   3. Origin Tracking: Products tracked by origin (South Korea, Japan, China, USA, Other)
   4. Seasonal Products: Can be seasonal or limited edition with expiry dates for availability windows
   5. K-pop/Anime Integration: Products can have tie-ins as core marketing and curation strategy
   6. Channel Exclusivity: Products can be exclusive to either online or in-store channels (mutually exclusive)
   7. Inventory Automation: Stock levels automatically adjust based on order status changes via triggers
   8. Storage Requirements: Complex storage instructions stored in CLOB fields for specialized handling
4. **Subscription Model:**
   1. Recurring Billing: Subscriptions support 1, 3, 6, or 12-month billing cycles
   2. Subscription Types: Standard and Premium tiers with different pricing and product selections
   3. Multi-Product Delivery: Each subscription delivery can include multiple products
   4. Gift Subscriptions: Can be gifted to others with separate issuer and recipient tracking
   5. Auto-Renewal: Configurable auto-renewal with different defaults for purchased vs. gift subscriptions
   6. Delivery Tracking: Individual tracking for each product within subscription deliveries
   7. Personalized Curation: Product selection based on customer preferences and K-pop/anime interests
   8. Flexible Addressing: Delivery addresses can differ from customer billing addresses (stored in CLOB)
5. **Promotion Structure:**
   1. Product-Specific Promotions: Promotions can apply to specific products through PRODUCT\_PROMOTIONS junction table
   2. Customer Targeting: Promotions can be individually assigned to customers through CUSTOMER\_PROMOTION junction table
   3. Usage Limitations: Each promotion can have specific usage limits per customer with automatic tracking
   4. Discount Types: Support for both percentage and fixed amount discounts
   5. Platform Targeting: Promotions can target online, in-store, or both channels
   6. Override Pricing: Product promotions can override general promotion discount amounts
   7. Expiry Enforcement: Automatic expiry date checking prevents use of expired promotions
   8. K-pop/Anime Themed: Promotions can reference specific K-pop groups or anime series
6. **Marketing Attribution:**
   1. Single Channel Attribution: Each order can be attributed to at most one marketing channel
   2. Channel Types: Social Media, Email, PPC, SEO, Direct, Referral, Display
   3. Platform Specificity: Channels specify exact platforms (Instagram, TikTok, Google Ads, etc.)
   4. ROI Tracking: Marketing effectiveness tracked through customer acquisition cost and conversion rates
   5. Performance Analytics: Automatic calculation of channel performance metrics via database views
   6. Attribution Optional: In-store orders may not have marketing channel attribution
7. **Gift Card System:**
   1. Issuer Requirement: Gift cards are always issued by a customer (tracked for relationship management)
   2. Optional Recipients: May or may not have recipient customer (physical cards or transfers)
   3. Multi-Order Usage: Gift cards can be used across multiple orders until balance is depleted
   4. Balance Tracking: Automatic balance reduction and redemption status updates
   5. Expiration Enforcement: Gift cards have expiration dates enforced at transaction time
   6. Code Uniqueness: Gift card codes must be globally unique across all cards
   7. Revenue Recognition: Immediate revenue recognition upon gift card purchase
8. **Customer Feedback System:**
   1. Multi-Level Feedback: Can be order-specific, product-specific, or general feedback
   2. Rating Scale: 1-5 star rating system with 1 being lowest satisfaction
   3. Verified Purchases: Automatic verification flag for feedback linked to actual purchases
   4. Review Text: Optional detailed review text in addition to numeric ratings
   5. Temporal Tracking: Feedback date automatically captured for trend analysis
   6. Business Intelligence: Aggregate ratings and satisfaction metrics calculated via database views
9. **Temporal Data Management:**
   1. Universal Audit Trails: All entities track created\_at and updated\_at timestamps via triggers
   2. Immutable History: ORDER\_STATUS\_HISTORY provides immutable audit trail for order changes
   3. Date Validation: Start dates must precede end dates across all temporal data
   4. Status Change Tracking: Complete tracking of who made changes and when
   5. Oracle TIMESTAMP: Precise timestamp tracking using Oracle SYSTIMESTAMP for accuracy
   6. Automatic Updates: updated\_at timestamps automatically maintained by database triggers
10. **Customer Preferences:**
    1. One-to-One Relationship: Each customer can have at most one preference record
    2. CLOB Storage: Flavor and product category preferences stored in flexible CLOB fields
    3. Structured Interests: K-pop/anime interests stored in VARCHAR2(1000) for targeted recommendations
    4. Subscription Interest: Binary flag indicating interest in subscription services
    5. Curation Impact: Preferences directly influence subscription box curation algorithms
    6. Auto-Creation: Preference records can be auto-created on first subscription or feedback
11. **Business Logic Automation:**
    1. Trigger-Based Automation: Inventory updates, timestamp management, and business rule enforcement
    2. Identity Columns: Auto-incrementing primary keys using Oracle GENERATED BY DEFAULT AS IDENTITY
    3. Check Constraints: Complex business rules enforced at database level
    4. Referential Integrity: Foreign key constraints with strategic CASCADE options
    5. CLOB Utilization: Flexible storage for complex data (preferences, storage requirements, addresses)
    6. Real-Time Validation: Gift card balance and expiry checking at transaction time
12. **Business Domain Specialization:**
    1. Cultural Integration: K-pop and anime tie-ins throughout product and marketing systems
    2. Multi-Cultural Products: Support for Korean, Japanese, Chinese, and other origins
    3. Gift-Centric Features: Enhanced gift handling aligning with Korean cultural practices
    4. Seasonal Awareness: Special handling for seasonal products and limited editions
    5. Community Building: Loyalty programs and feedback systems foster customer community
    6. Personalization Focus: Individual customer preferences drive product recommendations and curation
13. **Data Quality and Integrity:**
    1. Unique Constraints: Email addresses, gift card codes, product names must be unique
    2. Null Handling: Strategic use of nullable fields for optional relationships
    3. Business Rule Validation: Database-level validation prevents invalid business states
    4. Financial Accuracy: Monetary calculations enforced by constraints and triggers
    5. Audit Compliance: Complete change tracking supports regulatory and business requirements
14. **Performance and Scalability:**
    1. Strategic Indexing: Indexes on frequently queried fields (loyalty\_status, order\_status, etc.)
    2. View-Based Reporting: Pre-calculated business intelligence views for performance
    3. Normalized Design: Proper normalization reduces data redundancy and ensures consistency
    4. Oracle Optimization: Leverages Oracle 19c features for optimal performance
    5. Scalable Architecture: Design supports growth in customers, orders, and product catalog
15. **Security and Privacy:**
    1. Data Protection: Customer personal information properly secured through access controls
    2. Audit Requirements: Complete audit trails support compliance requirements
    3. Business Continuity: Referential integrity prevents accidental data loss
    4. Privacy Considerations: Customer preferences and feedback properly isolated and protected

# **Section Three: Entity Relationship Diagram and Data Dictionary**

## Entity Relationship Diagram

Since week seven of this course, we have made a series of significant changes to our original entity relationship diagram. The most notable improvement is the enhanced relationship between gift cards and customers. There are connections that facilitate product feedback as well. As we aspire to implement our database in the Oracle 19c lab environment, we moved from generic data types to those optimized for the release alongside precise constraints. For a greater deal of financial granularity and realness, we broke down the order totals into their component parts, i.e. subtotal, tax, shipping, discount. In addition, we’ve added a series of comprehensive database-level constraints and validations to better enforce our business rules. Operation enhancements have been made as deliveries can be tracked; audit audit trails can be displayed. Delivery status management is an additional feature present. Finally, we figured out how to implement those pesky CLOB fields for scalable text storage and more sophisticated field sizing (“Oracle Data Types” 2025). In prior models and implementation attempts, these fields were a consistent technical barrier. This is detailed in [Section 05: Implementing the CLOB (Character Large Object) Datatype](#_heading=h.gch873e3k2e4) of this document. Our refined model is clearly designed for implementation in the Oracle Database 19c environment. It’s important to not completely discount the [previous entity relationship diagram](https://mermaid.live/edit#pako:eNqtWetv4jgQ_1eifG6XstcXfOtSulv1-hClJ92pUmScAXw4dtZx2HK0__uNkxjycNhWpR9Q8YzHM7952qx9KkPw-z6oS0ZmikTPwsO_wdPj-P52OHr0Xl8PD-Xaux9dmm99L-aEQtLC9fj07XEwun4YX9_fGeYknSRUsQkkgZYte-xC8DAaXg1Hw7vB0Gydk7ZDrobDy28XgxujjJJLFraqUxJ9f3tvlMI9Ciiwpd1T3VlifJOHh6-v3mbB6MSZWOSW5NsKUMxprwVEwfV4eGt4qRSaMJFUOI1Q-erdXoxuhuPru-_B4MfF3d3wT7OBaI1IpRrCDVYl-dYFweP4Yvz0GPy4Ro1Hf-M-rQhdJFYl1PfyaTCubbJKMUF5GuIJrIW97MDNrikQnaq2XQjS9d1fw7uyOi7ezPa1_R5UgK2ckO-rhlIBsVO9wqiKQ0vSC8uc51adlH9-v74aB4OL0WUz9lFDgKhkXD3u1vmC-WNCezRNtIxABSz0Hm62tAQ9LWbelKlEB4JE0CBx0kaBiDDeWI3nUpR4Q6LB-1cyEZj_GtwkDBUkSWOdMr1qLCbaJSKWuM4DUz22tImUHIjwWBJIgckCgUXAyYPqIVW5uAx6XK4I16sgRjt0U1lLNvqlTXKsYArKxBSVUZQKRolmUgQR6LkMt-yaRYAiotijCqMQ-Yl2UdM4rFDfakWg5nqpwqbf6zFxdVNzWb6r6rMQKIsI97Q0gJNIpkI3rM03tkFBVhEI3TDdqBMRtQCNXAGdEyGA1_WyMjjRU6ki79nPXdth4jDz3rPvdO6MTXX1KLMSUKLC_AiUJFLOPTb1hNSb3PJ-MT3PeD3DW5a-H1fVK2jNcXNmrFq5XLfxqQsfBUsm06TNBwJ-NUhbjXNK5oIZuHPWGjVZOY3Ki6E7CDVE7zbHELCrhinVLtLPlAhdqRI2PDHDdBArRh2hG7KEVsP2067ctJV1q-4PLi_lVGdlxRaCs0psqkSzMqIWMwwLR-KxGRNNkyckgTocNjsSIIkUhNeSH15ihoGX_f_sF6p6ZIn1nkwYR9Q9EGHG68o5ziJmoIKQOU1YxDIOiEBoA80gKCttRRRFG16woSY4IzkS25ZsB09WTpCBiSVWG5NDHJbA9-j07ZxR8_r2SEec7whnmmKHwMqINtFFlaQgz4-s-9Q8lXVn7KBmV71aF54EmlmgFFsS7uipUhHMdAU_U6bAVOekCWQuvwZiXrRf3MRPI-yYTNfvaBiOVLNUZ6pZol7FJtYTSRnhHS4pYuU30UI02ZTRoNSExmwxlovOtUCDzLWlc4XXEozSRecb43wisXV0QNOyNJuamw5MKKKfZNmCQ0JSa1c4Gy5BJYaoKi7-NMjV0baGb3FlysrQh-cHPFhpV0QKeMF14JiwRYlpolw-uN6qLHYRDsxzvqpXNqMYliiehcWK8t9It27XRITGUdg8I5ZG75okNsW6sKUxy-7VPe6-WvVRS3utO_IDXTZDubq0AqJ-14r30VTt3ajZViPpCsltYy3o72-tuyMWW13LIGynibZZeEMvwiwGRbHAYr3tTNkLhI4K0z7ddiYSXeHv6qbZRQMEhdbozafeDKMdyBXqWvU7k1SEHDpyiuJddcy0iDhVWEuTfVYnx1tI27W2NSp2FapGOJWJJeDSBMJaTJiPIMOjQSoNT9WjMotrwsx6mpgGnA1N-x9Lg52ZlKX9Tux21IZW6GxYZN0Sb4q1Er3PuNi-0zWMs7nwkYiwzyGcLKUJqmY-1cZ3O407eTdTdqX-Cg1mYNuVx02eT2O2eaqs4TQFCCcE57cP5k3pymavz1i2vBkIUOh6K7ZcLRrh9JGNOPgYkJ797uGJowialxVoTMcb48L9jk2lh7n1jpeFOp4sSVIEbResCjMmZuYiUOUqPVHE81XCqJkea48SBRRbFapPYjYrGV6PmeP9ZjuU5jeRCeGkEs4ZpJkNLTeNZtkrFVH7qrIXN_gHPkITERb6fT9zwbOv5yj-2e-bxgVTknJtwHlDVpJq-bgS1O9rlcKBr2Q6m_v9KeEJfsvFF785WBZzdZXqNv9FIvth4sCPifhHyg0) as a conceptual model independent of specific deployment considerations.

#### Figure 01. Sweet Seoul Entity Relationship Diagram for Oracle 19c Implementation. To view a version in the live editor, click [here](https://mermaid.live/edit#pako:eNrNWmtv2zYX_iuEvqQDnL52lqvxdoDjuJ3Rxglsp8CGAAIt0TEXiXIpyo2X9L_vkLpREmXL6To3HwJL4jkiD5_znIv4bDmBS6yuRfgVxQ8c-_cMwV__bjK9uR6MJ-jl5fAweEY34yt51UVLDzskrBk1ubuc9MfD2-nwZiQHh9EsdDidkdAWQY1MesO-HQ_eD8aDUX8gRRe47iXvB4Ory17_o5wMD1bUrZ2Opvrm-kZOCmQ4cQhd1cp8GL6f2v3e-EpOgoZh1GxkUWtRQnv9t-Dw8OUFZTekqEfZY2yfWCwxtXzLS2J4ezgdXMuxTsAEpiwsjJRKgxd03Rt_HEyHow92__feaDT4JAWwEGD_SBA32wFNf7qx9mTam95N7N-HMOPxHyAnOHYeja8pLdslxAflX6lYpAuA1V3d9aelV6RLoMzxIhdEaM1wHUSZ1JxgEfE6KTDpcPR5MNInbxqrlvCcXtuFbdj8hiLuOFlR8jUZG48uQj_ZPONSEgMUoKLNJHmbcY54ufSo5kxlZD7HN-Tf6O76cjBGThSKwCfcpi66_YjurQ-EEY4BDwiHaOgSJqhY31u54OfeGPAzPrJP2mhOeShshn0CkqNAoFHkeXWDPdxwbKfdRsTH1IOBd4x-iUgLbRQ4aqPlImB1iq960wH6K6DMdmFhMOiKzHHkCTT5YyKfmXWenCDsupyEYYO1OWCkBsNCEU9g62oCGOnZkntrRsfbZ3eAguyAAUUQO91KbYFt9Kb97j32QrBg592UR-SXOiVgHRDn36MGtjhYY0-s7SUYW4S6hhbqL4jziH57h9p1q06lpZUiXfpgIjBzMXcPag3GyZxw6Z1O4PsRow4WNGC2T8QicHVVAwmsg3Q6lKEV9gD78F8yuaZ-OrweAO1d3yKHE-kQNhZF7GQjzGLR0m0m9q3E7FUvDbi7i4safPu9FBxLIxEG8TnnhIqjxO_a7inZrttHMo6LABALEjdSPLtRt-26sMBPNvaDiEk7TfETii9aaCt4ClNY0OWSsgcAQCgVTZJrJK931OXS0JFT0KalVpfe31GdskSu6y7GRRf1sedEntrIOSWe28RH4t2pesgtYS6stgrseGgR2jolLfHaBwTlnlKST56j-HlogJmP-SMRyvQLzBjxMrxJ0sIzD1hDQ141Dal1arDNPOC-Pqk3BzHbwUIPKDtUlHVQS2oPdC525TEkhWwH2GbzOvI8p8a07RyTeSDJYJncaqFUea2WGfW8opLL-M5mHfvir3KuWGWzBZXbtt6ZzzIWLJNZzJstBAn01eDTAEis35v0e1eDDeFiRYMozB0p3-Jd_AdUMfI119JANrdv_FQ5zQMxEK5xe3TXTTdo1iD1gBQlfg9gIQyYsp_6AQ6WzCQZ8ZOhKc6Ka0OiIP5_jKNECZSWbuQIk5q0MDBIfYmwmlWGld_qwwZkMcJecuqQfHSDoKXz3dt2bYDaw4ZmBdPzJmvuWIXIYiGVToqLRjXDcRsEocpyOF3KVLFJkg8TegDiqjp68oTWh1l4-EBZVTK-H9Zt6gyHpCEGUKgcWiVhDQOeyvfI05ICGSf8k_OgIoZUpfQz4ySRR30qYUBcmpix4csLO_i4DJY2ZtQntqDEVpYy0VD22qTmIU9QK4d0RUq1BuQhsJkzgiANmKOstsmH0xB1zFlDdewm1ZV5mBUjH_RStgIgy8jnkRXxand0D66Zd0eqvplPe1eiLXNk6pgGrmxGuk4ENR4kq7BDYLcG5WUix0nM-aourbW7cgfVpYCsSr6h7BQm3yGOsjTndKVczzS2_-nmEklUYRWAv0SUE5lUh3o18AnzB4IEeRJpNVDvAApO8RQ3QylL0Z_qhpc07RmHhh7lc7OCY_fmVSq8S9Q4ysXEekkMkSB-iuTT2mAQAmronDq2VuFsaPKcQyDIannsAHxCxbVpqdsk3CeqTqSqAByah1IBLyaduoIZEV8JYcDimLkIOHrPyCj2T6ugSD4lqFj-YzolWyhKEQJk0lxktFHXi2Tg5LZLPIgXeeTN9i3XUeUb5qbDJ9p65X0k72-pJgG9BTNV2wg9R8CkmnUR9DzFD5h).[[1]](#footnote-1)

## Data Dictionary

Alongside this document is a [GitHub repository](https://cherryberrykix66.github.io/sweet-seoul/) we’ve created for this assignment. We've used the magic of GitHub pages to host a series of stylized diagrams that would have otherwise been stale in Microsoft Word. The first of these diagrams is the [Data Dictionary](https://cherryberrykix66.github.io/sweet-seoul/data_dictionary_refined.html) for the Sweet Seoul Korean Bakery database. This data dictionary is a comprehensive, centralized record that serves detailed metadata for all the data elements in the information system (GeeksForGeeks 2024). We can think of this dictionary in terms of its traditional notion as a lookup tool. Although, mere word definitions are not contained here. Instead, the data dictionary provides detailed definitions, specifications, and metadata about every piece of data stored in the organization's systems (GeeksForGeeks 2024).

Looking at the [Sweet Seoul Korean Bakery Data Dictionary](https://cherryberrykix66.github.io/sweet-seoul/data_dictionary_refined.html), there are some features we’d like to point out. The data dictionary captures the unique aspects of the business model as a mix of contemporary culture offerings with traditional bakery operations (GeeksForGeeks 2024). Fields like kpop\_anime\_tie\_in, kpop\_anime\_reference, and kpop\_anime\_interest help the business stakeholders and developers understand the target market of the bakery. The seamless integration of online and in-store operations through fields like platform, online\_exclusive, and is\_in\_store\_customer demonstrate a sophisticated understanding of the business model that extends beyond a typical e-commerce databases. The comprehensive constraint system includes mutual exclusivity rules between online and in-store offerings as well as cross-field validations for minimum and maximum stock level (“Constraint” 2025). Beyond basic stock tracking, the system includes reorder\_point, min\_stock\_level, max\_stock\_level, and storage\_requirements (CLOB) for complex handling instructions, supporting real warehouse operations (Roeser et al., n.d.). Not only is the database designed with the unique business model and operations in mind, marketing attribution is baked into the system for cost and conversion tracking. With the integration of marketing concerns, the information system enables ROI analysis of the business’ advertising efforts.

From a technical standpoint, the transition from generic datatypes to Oracle 19c implementations such as NUMBER(10,2) for currency values, VARCHAR2 with precise sizing and CLOB for scalable text allows for the creation of a implementation-ready database design (“Oracle Data Types” 2025). This is an expression of the database system that is ready for implementation in our Oracle 19c lab environment. We are using Oracle's GENERATED BY DEFAULT AS IDENTITY for all primary keys. This demonstrates our hard-earned understanding of the requirements surrounding an implementation-grade database design as we have eliminated a common ID management issue (codezone 2024). In [Section 05: Design Inconsistencies with the ID Columns](#_heading=h.f8l558b6tegl), we explored this design issue in our previous test implementations throughout the evolution of our database. Achieving this finely-tuned model of our database entailed a series of iterations verified through SQL runs in the lab environment. Each run justified not only an edit of the SQL code but the overall database model and documentation. This experience underlines our understanding of the Database Development Life Cycle as an iterative approach by which one stage in the design process can demand revisions to previous steps (GeeksforGeeks 2024).

# **Section Four: Normalization Diagrams and Functional Dependencies**

## Functional Dependencies

Another document available within our [GitHub repository](https://cherryberrykix66.github.io/sweet-seoul/) are the [Functional Dependency Diagrams](https://cherryberrykix66.github.io/sweet-seoul/functional_dependencies_refined.html). These diagrams illustrate how each primary key functionally determines all other attributes in its table with the single-head arrow notation. It also shows foreign key relationships with the double-headed arrow notation. In this manner, the functional dependencies diagrams show how the tables reference each other. Each table follows the pattern where a single primary key determines all the non-key attributes. This setup is the foundation of proper normalization (Barick 2020).

Based on the functional dependencies shown, our database has achieved Third Normal Form (3NF) and shows characteristics that approach the Boyce-Codd Normal Form (BCNF). We’ve reached the First Normal Form (1NF) as all the attributes contain single, indivisible values. There are no arrays or multiple values within a single field. Uniqueness is ensured as each table has a primary key. Each table exhibits a fixed column structure as it has a consistent set of columns. With 1NF compliance attained, we can move onto the Second Normal Form (2NF) where no parent dependencies are present. That is to say that all on-key attributes depend on the entire primary key. This brings us to the 3NF where transitive dependencies are not present; non-key attributes don't depend on other non-key attributes (Barick 2020).

Our bakery database demonstrates a series of design principles that bring it close to achieving the Boyce-Codd Normal Form (BCNF). To understand this achievement, we need to examine how our database avoids a collection of specific dependency issues. Our decision to use auto-generated identity columns, or surrogate primary keys, throughout the entire database represents a foundational design choice that eliminates the most common source of BCNF violations. In some database designs, many tables will use composite primary keys composed of multiple columns. This setup creates a series of complex dependency scenarios where certain portions of the primary key can create conflicting functional dependencies (Upadhyay 2019). Our decision to use surrogate keys is a practical benefit beyond mere compliance with normalization standards. These auto-generated keys remain stable even if the business data changes. As a result, we can prevent a cascade of update problems that would occur with natural keys, such as email addresses and product codes. When these data points are used as primary keys, any modification on behalf of the business will entail a system update with heavy-lifting involved (“Surrogate Key in DBMS” 2021). With the Oracle identity column implementation (GENERATED BY DEFAULT AS IDENTITY), we can ensure that these keys are automatically managed by the database system. Hence, we reduce the potential for application-level errors in key generation. Referential integrity is maintained across all related tables (codezone 2024).

## Normalization Diagram

Our normalization diagram serves as a comprehensive physical database design blueprint where theoretical normalization principles are translated into concrete table structures. This diagram implements and enforces functional dependencies through its physical structure rather than just documenting them theoretically (GeeksforGeeks 2015).

#### Figure 02. Sweet Seoul Normalization Diagram. To view a version in the live editor, click [here](https://mermaid.live/edit#pako:eNrNWgtv2zgS_iuEgMK7gNNzsnGaGrcLpI7aNdo4gR9d7KGAQEu0zYtEqhSVxJvkv9-Qett6OdlbN0CL2OGMqJlvvnmQj4bNHWIMDCIuKV4J7H1jCH7evEFDLggymaSSkgDdU7lGck2oQIK4WFLOgjX1A3SE5r6DJXHQkgs0DIUgTKKR57vEg9_0ykjncD6dXV-Zkyl6ejo64o_oenKpPg2Q72KbBBWrpvMP0-FkdDMbXY_V4iBcBLagCxJYklfIJF9YNxPzozkxx0NTia5x1UM-mublh4vhZ7UZwe-oU7mdnOrrq2u1KZARxCb0rlLm0-jjzBpeTC6n1mg6nZuXIEKDIGwjMDGH5uirFik-JfofdnE5H86KNrVGM_NKvTBlths6xLEoK1-eN24qtSRYhqJK6ukJjcZfzfHsevInLJYC27c1T4g_Z-ZqfkLRH0KBK10Y_R8jR29n-51tDqijLNhZmVpnOruYzafW76Np_h2KAs8cBJ7Q1cXkszkbjT9Zw98vxmPzi3oClhLwF6ptJQgsSmXu015zCPEKTitCOn6LUlfEDtx2eWLJOhtj33dpLkh2xUqhzJdLInJvlvKBpoINcsiSMqrjPyKFW7LJs4Dk8MZLl9gS2TEZ0FoyeIy-UD_j-dUHcwJygeQeERZ10M1n9M34RBjRMEA4QCOH6J18MzLBrxcT8M_kxDru9RDxMHXRXAnOGf0eki4ac4nGoeuWyvR7aElFIC2GPQJCTYtd3HLtSQ_5a86a1_X7CDuOIEHQ4vG28kLzsgCs3WqHHFa6lkoDFasjr1jHQFkWZy5lxEo8BBKXZIlDV6JelQhlFiwW7YXAwnyDXbmxfE6ZDPLru2i4JvYt-u3Xomz-jRJpZYEwL92ZSswcLJxOpTEAukQo-Nvc80JGbQ1ZyyNyzZ28KlOBrKDn8mJmov_Cji0nsnyydvrnVP0tv3Y2ujKBha5ukC2I5jcsixLpinKxMIq2ZrHnEtrcjTgunH3CrSROPyrBibIeYZDLs_guD75Yg4fFLZGUrSx7jRkjbpmqXQ7uan144ZISnSu6lJYNXi7TlTFzuQ7txMgazV5MAWudqKpEcggkFUPmH4MKmOYlJH6wsMdDJhOZRpQXHgjVl68tBwH8Ig0ODWz1-NfsQr9zpmB-o2x0Ca-PXTt0o7KQEtepCrjI1LuRekOYAy_XqWI3H29UQskCM9oqZegOu-D36PugMs5ha1CuennBnzoRuXW6qEPZkeaszs9VrKZQVkVkuY32Mj_lGF6ZuQx9BcEFdd2Wcoekk-1qapdc1lTZcrM3vaSktB3EEY01ZPWIze8oD4MMX7VGBwlG7nOLS9RnJouWad5akRKyKLV4HsKJzRctsjlk_eg54N6Asx8WC1HZWplfJPH-GRTEstDKOaEty6SThqNW_nuIo6I3oYnfqpkQigVp-YLaJFvdgnzzHPK2V8m5B_Jq2pY91tn2BTV6Iq3L6Pal-mkPZKEdgv7fV4VZm3IZ9rQC-tnNEfFfoEeqkoU_rijblYy-D6p8u8ABaQkFFOiA1nVDaS7RBQl58CkQaEwySdR39cQlVaCiq3RLyKUeVb4nDo2NVp-2lINufe5bmFGPWJISS1uhuuiCh8SdAXmAhjWgd6RYtV-FMsSuu0HZ33XzmDQH5Wk26Rz2VBptpazcBI2U3QE2VT5yyR1xKz10oIjLRiu7IZdtfV8S3SbCJNj2JcS4o1duAZO16MtiOUEiGtcNXaXJNdZ1dw0Fj3rCNuLLAoPY2shC0Luo-i5ZO_xy_QEpKGGdQr-HVOh5RJAvWb9gsSJIkgcZlay1TYYXYRO2WI-itM95qFq-pekHgGDJzOuxXee2ZyLoR2WNEt4zD5xkknLjk12GVt9W0noAoKFLalu5NqBm8HEOlJ62utgG9ASaR5POq03-jlX1lSoOYSwCpUAUq8a8gt4RsPAPAIbiqHIXB_E0Xmfjf3KAoMMfanAhU5KoWsUgpC2HuJAfsiSauinTscsuzEmVN7ZsAMmCLXZ72gtbwhY6VVWDx5lcu5tWhUPaHNob2yWWFi2Q2R9g-MKaBFfHR8cnrfYfx1X).[[2]](#footnote-2)

# **Section Five: Create the Database with DDL Statements**

## Refined DDL Statements and Implementation Confirmation

The [GitHub page](https://cherryberrykix66.github.io/sweet-seoul/DDL_code_samples_v2.html) containing all the DDL statements needed to create the basic structure of the Sweet Seoul Korean Bakery database can be viewed [here](https://cherryberrykix66.github.io/sweet-seoul/DDL_code_samples_v2.html). We also captured a series of screenshots running through the execution of these DDL statements in our Oracle 19c lab environment. These images can be viewed in detail [here](https://github.com/cherryberrykix66/sweet-seoul/tree/main/DDL_oracle_environment). As a form of verification we have included a [summary script](https://cherryberrykix66.github.io/sweet-seoul/DDL_code_samples_v2.html#completion) that confirms the successful completion of our DDL statements as well as a comprehensive overview of the components implemented. These scripts query the user metadata views to generate a complete report of the objects created (Kedem 2015). We’ve included the results of this verification run from our session in the Oracle 19c lab environment on May 30, 2025. This is a confirmation that the basic structure and associated business logic of our database has been successfully created in an actual Oracle environment. Following this evidence, we have included a couple sections entailing some previous implementation issues that led to the final iteration of our database design. Access to the Oracle environment for testing cannot be understated as we utilized this tool to test and refine not only our database script but the design model itself. Along the way we learned a great deal about Oracle-specific implementation practices as well as best practices in database design theory.

#### Figure 03. Tables Verification. Captured by author, May 30, 2025.

#### Figure 04. Views and Triggers Verification. Captured by author, May 30, 2025.

#### Figure 05. Sequences Verification. Captured by author, May 30, 2025.

## Issues with the Prior Model and Resulting SQL

### *Implementing the CLOB (Character Large Object) Datatype*

#### Figure 06. Creating test tables to troubleshoot FEEDBACK table creation failure. Captured by author, May 17, 2025.

#### Figure 07. Verifying user privileges, tablespace quotas and syntax. Captured by author, May 23, 2025.

We were having some trouble adding the FEEDBACK table due to an issue with the comment column. It has a data type of CLOB (Character Large Object). This data type is designed to store large amounts of character data. It can handle more than the VARCHAR2 data type. That said, we don’t appear to have the appropriate permissions to create this column on the FEEDBACK table. So, we checked our system privileges (McDermid and Sarika Surampudi 2025):

SELECT \* FROM user\_sys\_privs WHERE privilege LIKE '%CREATE%';

We also checked the tablespace quota to see if there are any limitations (“QUOTA in Tablespace - Learnomate Technologies” 2024):

SELECT \* FROM user\_ts\_quotas;  
And we tried creating a small test table:

CREATE TABLE test\_table (  
 id NUMBER,  
 large\_text CLOB  
);

If anything, we know we are using the correct syntax and indeed have CREATE TABLE privileges.

#### Figure 08. Verify schema to resolve CLOB column issue. Captured by author, May 23, 2025.

Finally, we verified our current schema and that the table exists therein. Still, we couldn't add the comment column to the FEEDBACK table. After this troubleshooting procedure, we decided to just build the FEEDBACK table incrementally to see if any other columns were affecting the CREATE TABLE statement. Then, we added the constraints, triggers and indexes separately.

#### Figure 09. Incrementally creating the FEEDBACK table. Captured by author, May 17, 2025.

#### Figure 10. Adding constraints, trigger and indexes to the FEEDBACK table. Captured by author, May 17, 2025.

### *Design Inconsistencies with the ID Columns*

#### Figure 11. Creating the CUSTOMERS table without identity for the ID column. Captured by author, May 17, 2025.

#### Figure 12. Dropping and rebuilding the CUSTOMERS table for design consistency. Captured by author, May 17, 2025.

You’ll notice that the CREATE TABLE statement used in [Figure 11](#_heading=h.40kjft9psmjj) doesn't indicate the customer\_id column AS IDENTITY. This means that the CUSTOMERS table will not automatically generate sequential numbers for the customer record IDs. We were noticing a lot of issues when the other tables were added without the ID field as an identity. Therefore, we switched to the design pattern that allows the following:

1. Oracle to automatically create values for the ID column (GENERATED)
2. The database administrator to insert their own values (BY DEFAULT)
3. The ID column to auto-increment (AS IDENTITY) and act as the unique identifier for each row (PRIMARY KEY)

By having the customer\_id be a mere number, we were introducing a host of problems into our database (codezone 2024).

# **Section Six: Populate the Database with DML Statements**

The [GitHub page](https://cherryberrykix66.github.io/sweet-seoul/DML_code_samples_v2.html) containing all the DML statements needed to populate the Sweet Seoul Korean Bakery database can be viewed [here](https://cherryberrykix66.github.io/sweet-seoul/DML_code_samples_v2.html). We also captured a series of screenshots running through the execution of these DML statements in our Oracle 19c lab environment. These images can be viewed in detail [here](https://github.com/cherryberrykix66/sweet-seoul/tree/main/DML_oracle_environment). As a form of verification we have included a series of [Smoke Tests](https://cherryberrykix66.github.io/sweet-seoul/DML_code_samples_v2.html#smoke) (“Smoke Testing | Software Testing” 2019) and [Summary Counts](https://cherryberrykix66.github.io/sweet-seoul/DML_code_samples_v2.html#summary) with error prevention (“Exception Handling in PL/SQL” 2018) that confirm the successful completion of our DML statements. To verify the [business logic and views](https://cherryberrykix66.github.io/sweet-seoul/DDL_code_samples_v2.html#business-logic) previously implemented as a portion of our DDL statements, we created a series of [Business Intelligence Test Queries](https://cherryberrykix66.github.io/sweet-seoul/DML_code_samples_v2.html#business-intel-test). All of our business intelligence tests cleared. Although one did need some troubleshooting as seen below. We actually employed Claude AI from Anthropic to aid us in this matter (Claude 2025).This was a case where the view and business logic was in place; we simply didn’t have applicable test data. With this confirmation in hand, we could confidently move on to creating a series of queries designed for complex business reporting.

#### Figure 13. Testing the inventory alerts view doesn’t yield any results. Troubleshooting with Claude AI from Anthropic. Captured by author, May 30, 2025.

#### Figure 14. Executing a manual calculation to identify if the alert is being triggered with the sample data in place. Troubleshooting with Claude AI from Anthropic. Captured by author, May 30, 2025.

#### Figure 15. Creating an applicable testing scenario, testing the view and re-running the business intelligence testing query. Troubleshooting with Claude AI from Anthropic. Captured by author, May 30, 2025.

# **Section Seven: SQL Queries for Business Reporting Requirements**

Now that we have the basic structure of the database built, the basic business logic intact, and the table records implemented, we can start querying the database to service our business intelligence initiatives. This is the step where the raw data inserted into the database becomes information we can use to make business decisions (Cornel and Morris 2018). In this section, we have written and executed a series of [queries that satisfy the Business Reporting Requirements](https://cherryberrykix66.github.io/sweet-seoul/SQL_queries_business_reporting.html) from [Section One](#_heading=h.a2fbyzsz6ch2).

Before we dive into the individual reporting metrics, it’s important to note that the output for these queries were formatted in a manner similar to the example below:

SET LINESIZE 200  
SET PAGESIZE 50  
SET TRIMOUT ON  
SET TRIMSPOOL ON  
SET TAB OFF  
SET WRAP ON  
SET HEADING ON  
SET UNDERLINE ON  
SET FEEDBACK ON  
SET ECHO OFF  
SET VERIFY OFF  
SET NUMFORMAT 999,999,990.99  
ALTER SESSION SET NLS\_DATE\_FORMAT = 'YYYY-MM-DD';

This formatting script sets up a standardized display for the query results. These commands display the layout according to LINESIZE and PAGESIZE set to a maximum value. The longer data is allowed to wrap to the next line as opposed to truncating. The output is cleaned of trailing spaces and tab characters. Column headers are demarcated with dashed underlines. With the SET ECHO OFF command, the SQL commands are hidden from the output; only the results are shown. Likewise, the SET VERIFY OFF command determines how variables are substituted and displayed. Finally, we have added some data formatting for numbers with commas and decimal places. We also want to view the dates in ISO format (“Formatting Query Results,” n.d.). Commands such as these are typically placed at the beginning of a SQL script to ensure consistent and readable output. In this manner, the output is viewable across a variety of environments and users. While some of the opening formatting statements will vary as we introduce the seven business reporting metrics outlined in our [Business Case Abstract](#_heading=h.a2fbyzsz6ch2), it's important to understand the basic function they serve in all cases.

## Requirement One: Product Performance Analytics

### *Best-Selling Items Comparison Across Online and In-Store Channels*

This first query is a [comprehensive sales channel analysis](https://cherryberrykix66.github.io/sweet-seoul/SQL_queries_business_reporting.html#best-selling). It compares online and in-store performance for each product. When we query sales performance by product across these two channels, we are providing metrics to understand which product performs higher on either of the two platforms. The results are sorted by total revenue with the highest appearing first. Our top-performing product is the “Stray Kids Keychain Set” with a preferred channel of “Online.” Overall, the best performing products featured a K-Pop or Anime reference. By understanding the channel preferences for each product, we can determine inventory allocation, marketing strategies and channel optimization measures. A further note, this query has specialized formatting for the data retrieved; it builds upon the basic formatting previously discussed. Many of the business reporting queries seen here will be accompanied by specific formatting commands (“Formatting Query Results,” n.d.).

#### Figure 16. Query results for Best-Selling Items Comparison Across Online and In-Store Channels. Captured by author, May 31, 2025.

### *Inventory Turnover Rates*

This second query provides a [comprehensive inventory velocity analysis report](https://cherryberrykix66.github.io/sweet-seoul/SQL_queries_business_reporting.html#turnover). Its core purpose is to analyze how quickly each product is selling in relation to the current stock levels. This analysis will allow us to identify which items need attention in terms of inventory management. The product velocity is calculated through a multi-step process that analyzes the relationship between sales volume and current inventory levels. Based on this formula, the query assigns a velocity verification threshold to the associated product as follows:

* If the product has zero current inventory, it is flagged as “Out of Stock” even if there isn’t any sales history.
* If the product hasn’t generated any sales (with the cancellations excluded), it is classified as “No Sales.” This is an indication for an inventory that is potentially obsolete.
* If the sales volume of a product exceeds the current stock by more than 3 times, it is labeled as “Very Fast”. This is indicative of a product in extremely high demand.
* Products with a turnover ratio between 1.5 (exclusive) and 3.0 (inclusive) are considered “Fast-Moving.” Turnover ratios in this range indicate that a product has a strong and consistent demand.
* “Medium-Moving” and “Slow-Moving” products are similarly determined based on the range in which their turnover ratio falls.

We can also see a similar labeling attribution process occurring in regards to the stock\_status column. Here, the min\_stock\_level (within the INVENTORY table) provides a threshold against which the current\_stock value is held. If the current\_stock drops to or below min\_stock\_level, the system flags it as "Critical Stock." Likewise, if the current\_stock exceeds the max\_stock\_level, the product is labeled as “Overstocked.” The CASE statement also considers the reorder\_point attribute from the INVENTORY table. This respective labeling will bring restocking needs to the immediate attention of the store staff. Future orders for overstocked items can be avoided to prevent the bakery’s cashflow from getting tied up in unselling inventory.

Finally, we have a supply forecasting mechanism at play with the estimated\_days\_supply column. The current stock is predicted to last for a certain amount of days based on the formula of current\_stock divided by the daily sales rate. The latter metric is calculated from the historical order patterns present in the database.

There is a lot of business value in this query as it accounts for fast-moving products where increased orders may be necessary. This query provides the business intelligence to prevent items from falling out of stock. In the event of such an occurrence, the system can produce a flag. Likewise, slow-moving products may not be considered in future orders. The business managers may even consider applying a promotion to boost slower sales on certain products. Or they could evaluate discontinuation altogether.

Overall, this report empowers the bakery management with actionable insights into lost sales opportunities, reordering priorities and optimized inventory investment. These analyses would contribute to the bakery’s cashflow; we don’t want it locked up in unmovable inventory.

#### Figure 17. Query results for Inventory Turnover Rates. Captured by author, May 31, 2025.

### *Seasonal and Limited Edition Product Performance*

This third query provides an analysis of [seasonal and limited edition products](https://cherryberrykix66.github.io/sweet-seoul/SQL_queries_business_reporting.html#seasonal-products) with sophisticated time-sensitive metrics. The daily\_sales\_rate calculates average daily sales velocity using either the actual selling period or the count of active sales days. This calculation provides a realistic measure of product movement speed as opposed to a simple average in calendar days. The sell\_through\_percentage estimates what portion of the original inventory has been sold by using the formula: (Units Sold ÷ Estimated Original Inventory) × 100. The estimated original inventory equals current sales plus remaining stock. This percentage metric indicates how successfully the product is moving through its lifecycle.

Having a report on expiration dates within the context of sales rates allows the bakery management to make important strategic decisions such as training staff on upselling techniques. If a product is near expiry, the bakery management will need to enact an urgent clearance sale. They could also bundle the items approaching expiry with more popular products to encourage quicker sales. This report allows the bakery managers to optimize their most time-sensitive products by showing both sales velocity and remaining selling opportunities in terms of time.

#### Figure 18. Query results for Seasonal and Limited Edition Product Performance. Captured by author, May 31, 2025.

## Requirement Two: Subscription Box Performance Tracking

### *Subscription Growth and Status Trends Over Time*

This query produces a dashboard that tracks the [health and growth of the bakery’s subscription box](https://cherryberrykix66.github.io/sweet-seoul/SQL_queries_business_reporting.html#growth-trends) service over time. It is organized according to a month-by-month view of acquisition, retention, and growth trends. Primarily it compares new signups to the previous month via a growth percentage. If the value is positive, there is an indicated percentage of subscription growth whereas a negative value translates to lost subscriptions. The month-over-month growth percentage is a key metric that reveals if the subscription business is accelerating or declining. If negative growth trends start to appear, there is a need to investigate an issue in the marketing approach or a specific product included in the subscription. Growth trends also help identify peak subscription months that may coincide with holidays or promotions.

Another health metric is the active\_rate. This value is a percentage of the subscribers from each cohort, i.e. New, Active, Paused, Cancelled, that remained active. A high rate would indicate good retention. A lower rate might suggest that churn is an issue. The retention rate can indicate as to whether the subscription box model or contents are a good fit for their targeted market. Again, the month-to-month analysis can reveal which period of the year produces positive results as more loyal customers appear. Customer loyalty may be relative to the particular products offered that month. Perhaps the value proposition was positive given the contents or time of year. With these considerations in mind, we can evaluate the success of our subscription model via a comparison of retention and monthly product offerings.

#### Figure 19. Query results for Subscription Growth and Status Trends Over Time. Captured by author, May 31, 2025.

### *Subscription Type Performance Analysis*

This query provides a comprehensive business intelligence analysis of the bakery’s subscription service performance with [segmentation based on subscription type and billing frequency](https://cherryberrykix66.github.io/sweet-seoul/SQL_queries_business_reporting.html#sub-type). By grouping the data by both subscription type and billing cycle, this query reveals which subscription models are successful and profitable. It determines the customer lifetime duration by measuring the time from subscription start to either the end date or current date for completed or active subscriptions respectively. With this information, the bakery will have insight into their customer retention patterns specific to their subscription model. The churn rate calculation shows what percentage of subscriptions have been canceled. This is a critical metric for determining subscription business model health. Predictable revenue streams can be ascertained from the monthly recurring revenue calculation. Additionally, this query analyzes customer behavior patterns by tracking the percentage of gifted subscriptions and auto-renewal rates. The results are intended to help the bakery management identify their most valuable subscription segments and make informed, data-driven decisions around pricing, marketing initiatives and service improvements.

#### Figure 20. Query results for Subscription Type Performance Analysis. Captured by author, May 31, 2025.

### *Customer Lifetime Value (CLV) Analysis for Subscriptions*

This query performs a detailed [customer lifetime value (CLV) analysis](https://cherryberrykix66.github.io/sweet-seoul/SQL_queries_business_reporting.html#clv) for the bakery’s subscription customers. The CLV metric is attained by calculating the total financial value each subscriber has generated throughout their relationship with the service . The total subscription value each customer has contributed is calculated by first computing their subscription durations. Then, this duration is multiplied by the monthly subscription price. This query also tracks operational metrics like delivery completion rates. It accomplished this by analyzing how many subscription box items were successfully delivered versus planned. This information can provide the bakery management with insights into service quality and customer satisfaction (Caldwell 2022). Again, we can see a CASE statement used to categorize customers based on defined ranges resulting in labels, i.e. “High Value”, “Medium Value” and “Low Value”. This labeling allows the bakery to quickly identify their most valuable customers for targeted retention efforts and personalized service. In addition, we predict future revenue and identify at-risk customers with access to their auto-renewal and current subscription statuses. Overall, this query is designed to support strategic business decisions around customer retention programs, premium service offerings, and personalized subscription boxes for different value segments. By understanding the customer lifetime value, the bakery management can optimize not only the content of the subscription boxes; they can determine pricing strategies and invest in their customer experience (Caldwell 2022).

#### Figure 21. Query results for Customer Lifetime Value (CLV) Analysis for Subscriptions. Captured by author, May 31, 2025.

## Requirement Three: Customer Acquisition Cost Analysis

### *Marketing Channel Performance Overview*

This first query provides a [marketing channel and ROI analysis](https://cherryberrykix66.github.io/sweet-seoul/SQL_queries_business_reporting.html#marketing-channel) that evaluates which marketing investments are generating the best returns. Each marketing channel is evaluated for effectiveness by calculating the net ROI ratio. This report will help optimize the fund allocation for the various media channels in the marketing budget. The key metric here is the ROI ratio. This value is derived from the total net revenue (after discounts) generated by customers from that channel by the total customer acquisition cost (CAC) investment for those specific customers. If the ROI is greater than 1.0 (> 1.0), the channel can be considered profitable. The channel would be considered a financial loss if the ROI falls below 1.0 (<1.0). This evaluation will help the bakery management determine the efficacy of their marketing channel spend. The goal is to focus on the media channels that bring valuable customers. If a particular channel continues to underperform, there may be a need to optimize the channel-specific marketing strategies or reallocate some of the marketing budget to a higher performing channel.

Although, returns can take time to appear as some marketing investments entail a warm up period. That’s why this query is designed to include a CAC payback period (cac\_payback\_months) column in terms of months. What we have here is a measurement of channel efficacy in terms of time. A quick payback would be a value under three (<3) months to the acquisition of customers. If a channel takes more than twelve months (>12) to produce a customers base, it could be considered a higher funnel or long-term approach to customer retention.

Beyond a basic ROI analysis, this query provides enhanced business intelligence with the inclusion of the Customer Lifetime Value (LTV) metric. This information can enable the bakery management to evaluate long-term customer relationships as opposed to simple, immediate returns (Caldwell 2022). There is also a crucial distinction between gross revenue and net revenue that is applicable to the bakery’s unique business model. The elements of promotional campaigns, gift card usage, and seasonal discounts could significantly impact true channel profitability. This comprehensive analysis enables informed budget allocation decisions that account for both immediate returns and long-term customer relationships across multiple channels and a subscription business.

#### Figure 22. Query results for Marketing Channel Performance Overview. Captured by author, May 31, 2025.

### *Channel Type Comparison*

This second query creates a report that [groups individual marketing channels into broader strategic categories.](https://cherryberrykix66.github.io/sweet-seoul/SQL_queries_business_reporting.html#channel-type) Through these set categories, we can compare the effectiveness of different marketing audiences. The result is a high-level comparison among the aggregated channels. The various marketing channels are grouped into three strategic categories via the Common Table Expression (CTE) below:

WITH channel\_metrics AS (  
 SELECT  
 CASE   
 WHEN mc.channel\_type IN ('Social Media', 'Social') THEN 'Social Media'  
 WHEN mc.channel\_type IN ('Email', 'PPC', 'SEO', 'Display') THEN 'Digital Marketing'  
 WHEN mc.channel\_type IN ('Direct', 'Referral') THEN 'Organic/Referral'  
 ELSE 'Traditional/Other'  
 END AS marketing\_category,  
 mc.marketing\_channel\_id,  
 mc.customer\_acquisition\_cost,  
 mc.conversion\_rate  
 FROM  
 MARKETING\_CHANNELS mc  
),

The CTE creates a temporary result set named channel\_metrics that can pre-process the data by applying the marketing\_category grouping. The primary query can reference the CTE like a regular table in the database: FROM channel\_metrics cm. By using the CTE, we are avoiding the repetitive use of the CASE statement in the main query. Essentially, the overall query is simplified when the category mapping is pre-calculated. The readability of this query is improved as the data transformation is separated from the aggregation. Hence, CTEs are an excellent means of breaking an overly complex query into logical, digestible steps (“CTE in SQL” 2018).

The primary reason for including this query is the need to identify which route the bakery should pursue: online advertising or local community outreach. Each strategic marketing strategy will require funds that eat at overall profits. So, it’ll be useful to weigh the effectiveness of digital and organic marketing strategies. This is especially pertinent given the bakery’s indecision to expand into a second storefront or grow their e-commerce site. They will need to compare the ROI between these strategic categories.

#### Figure 23. Query results for Channel Type Comparison. Captured by author, May 31, 2025.

### *Customer Acquisition Funnel by Marketing Channel*

This third query produces a [marketing funnel analysis](https://cherryberrykix66.github.io/sweet-seoul/SQL_queries_business_reporting.html#funnel) that breaks the customer acquisition process into detailed steps. The complete marketing funnel is analyzed from the initial impressions stage to final customer acquisition. Along the way, the conversion rates and costs are captured for each stage. Before we dive into the specific of how the query works, we’ll provide a crash course in the stages of the marketing funnel. Impressions are people that merely saw the advertisement. For instance, someone can see a display ad load on their browser screen. Prospects are people that showed interest in the ad by clicking through or interacting in some way. This is known as a conversion. Hence, the conversion rate is the percentage of people who progress from impression to prospect. Customers are people who actually made a purchase. When we refer to the acquisition rate, we are talking about the percentage of prospects that became customers. This query provides a cost analysis that compares impressions and customers (Salsi 2024).

To estimate the impressions by channel type, we created another CTE (“CTE in SQL” 2018) with multipliers based on industry benchmarks (Bond 2023):

WITH channel\_metrics AS (

SELECT

mc.marketing\_channel\_id,

mc.channel\_name,

mc.channel\_type,

mc.conversion\_rate,

mc.customer\_acquisition\_cost,

-- UPDATED: More sophisticated impression estimates based on channel type

CASE

WHEN mc.channel\_type = 'Social Media' THEN mc.customer\_acquisition\_cost \* 800

WHEN mc.channel\_type = 'PPC' THEN mc.customer\_acquisition\_cost \* 500

WHEN mc.channel\_type = 'SEO' THEN mc.customer\_acquisition\_cost \* 1200

WHEN mc.channel\_type = 'Email' THEN mc.customer\_acquisition\_cost \* 1500

WHEN mc.channel\_type = 'Display' THEN mc.customer\_acquisition\_cost \* 2000

WHEN mc.channel\_type = 'Direct' THEN mc.customer\_acquisition\_cost \* 300

WHEN mc.channel\_type = 'Referral' THEN mc.customer\_acquisition\_cost \* 200

ELSE mc.customer\_acquisition\_cost \* 1000

END AS estimated\_impressions

FROM

MARKETING\_CHANNELS mc

),

With impression estimates established, we can extrapolate our marketing funnel stages accordingly. To produce these metrics, we’ll create a series of calculated columns to convey the matriculation of impressions to prospects and prospects to customers. The cost per impression ultimately leads to the cost per customer.

Businesses will typically use queries like this to discern whether a campaign needs to be improved due to a lower conversion or acquisition rate. In the case of an unsatisfactory conversion rate, the ad creative or targeting will need to be adjusted. Likewise, a low acquisition rate is indicative of a need to improve the follow-up process on prospects with better communication or specialized offers. If the cost per impression is high from the start, the bakery management might benefit by exploring different platforms or campaign launch times. In the marketing world, full funnel analysis and attribution is paramount. Media planning can be tweaked with an understanding of the tradeoffs between reach and targeting. An overview of the funnel can reveal where bottlenecks occur in the conversion process. The entire spectrum of a company’s marketing strategies can be determined from creative content to digital distribution. With a detailed cost breakdown, the marketing budget can be clearly justified to company stakeholders (Bond 2023).

#### Figure 24. Query results for Customer Acquisition Funnel by Marketing Channel. Captured by author, May 31, 2025.

## Requirement Four: Customer Segmentation Reporting

### *RFM Customer Segmentation Analysis*

Before we take a look at this next query, we think it would be helpful to briefly define an [RFM customer segmentation analysis](https://cherryberrykix66.github.io/sweet-seoul/SQL_queries_business_reporting.html#rfm). This is a marketing technique where customers are grouped based on three purchasing behaviors:

1. Recency: How recently did the customer make their last purchase?
2. Frequency: How often does the customer make purchases in a given time period?
3. Monetary: How much money does the customer spend as a total or average?

Based on these three dimensions, the customer is scored between 1 and 5. For instance, the highest performing customers will receive a 5-5-5 score as a recent purchaser, with a frequent history or purchases that were of a high monetary value. Meanwhile, the lowest scoring customers will receive a 1-1-1 as a dated, infrequent, and low-spending case. With this analysis in hand, a company can create better targeted marketing strategies. One approach would be to focus on high-touch efforts targeted at highly loyal customers while automating campaigns for lower-value segments. Product strategies could entail premium offerings for high-monetary customers. Customers with a lower total spend might be attracted to entry-level or discounted products. So, a level of personalization can be introduced to marketing campaigns. Customers most likely to churn may be identified through low recency scores. In many ways, marketing resources can be duly allocated when there are clear strategies for each customer group. The goal is to focus marketing efforts where they will have the most impact. This query has the capacity to transform raw transactional data into strategic business intelligence that drives both targeted segments and improves customer retention (Wright 2021).

#### Figure 25. RFM Customer Segmentation Analysis. Captured by author, May 31, 2025.

### *Customer Promotion Response Segmentation*

This query can be used to evaluate how well [individual customers respond to marketing campaigns](https://cherryberrykix66.github.io/sweet-seoul/SQL_queries_business_reporting.html#promo-response). Customers are segmented according to their engagement level with the bakery’s advertisements. From this analysis, we can determine which customers are more likely to respond to promotional offers. Therefore, marketing campaigns can be optimized with semi-personalized promotional strategies. The calculations are pretty straight forward as the promotions\_used (multiplied by 100) is divided by promotion\_offered to determine a rate. Based on the value of promotion\_response\_rate, the customers are segmented into super, high, medium and low responders. High responders are likely to act on exclusive deals and pre-orders. Their value lies in their sensitivity to promotional, limited-time offers. Meanwhile, medium responders may have more select preferences. This is an ideal audience to enact A/B testing with. Low responders may need offerings outside of the usual promotional materials. They may be more price-sensitive. In this query, there is also an ELSE statement intended to capture customers that are minimal responders. This is an audience that would benefit from a retargeted offer to boost their responsiveness. By using the customer promotion response segmentation analysis, the baker can optimize their advertising ROI by targeting the correct customers with the most fitting offers at an acceptable frequency. Less responsive segments will not be subject to advertising fatigue due to inapplicable or unattractive offers that a more engaged audience will readily accept.

#### Figure 26. Query results for Customer Promotion Response Segmentation. Captured by author, May 31, 2025.

### *Subscription Customer Segmentation*

This query performs [customer segmentation analysis for the bakery’s subscription service](https://cherryberrykix66.github.io/sweet-seoul/SQL_queries_business_reporting.html#sub-seg). As we’ve seen in the other customer segmentation queries, we are using a CASE statement to attribute labels to different customers based on the applicable range of their total\_subscription\_value. The customers are segmented into nine distinct categories reflecting different stages of the customer lifecycle. By categorizing subscribers based on their value, loyalty, and engagement patterns, we can determine the best targeted customer relationship management strategies. We can combine financial value (total\_subscription\_value), longevity (subscription\_duration), and operational satisfaction (delivery\_success\_rate) indicators to gain a holistic view of each customer's relationship with the subscription service. This segmentation allows the bakery management to implement targeted retention strategies such as special offers for high-value customers that have recently paused their subscription. Welcome campaigns can be deployed for new subscribers. Churned customers might call for various types of win-back initiatives.

#### Figure 27. Query results for Subscription Customer Segmentation. Captured by author, May 31, 2025.

## Requirement Five: Limited-Time Offering Effectiveness Metrics

### *K-Pop/Anime Themed Product Performance Overview*

This query analyzes how well specialty-themed bakery items are selling within a [limited window of availability](https://cherryberrykix66.github.io/sweet-seoul/SQL_queries_business_reporting.html#themed). The focus is on daily sales velocity (sales\_per\_day) and customer satisfaction (avg\_rating). Both sales rate and customer feedback are used to inform promotional offers around limited edition products. The sales\_per\_day velocity metric is calculated by dividing total units sold by the days\_available figure. With this calculation, the bakery management can receive insight into how quickly the specialty items move through their lifecycle. Meanwhile, verified customer ratings provide a set of authentic feedback metrics from confirmed purchases.

An urgency level classification system appears as a CASE statement is used to categorize products based on the days remaining until expiry in tandem with remaining stock levels. The categories are flagged as Emergency, Critical, High, Medium and Expired. If this CASE statement doesn’t detect a flagable condition, the product is marked as normal. With these flags in pace, the bakery management can be prompted to enact clearance activities and timely promotions to avoid lost sales opportunities.

We can pair the velocity with the average customer rating to discern if a high-selling product could be improved. The bakery may want to contact the vendor and explore similar products of higher quality. Therefore, this analysis not only takes successful sales into account; it looks to customer feedback as an indicator for repeat promotions and ongoing vendor relations. In time, this analysis could yield insight into the optimal duration of a limited edition product promotion. Management responses to the urgency level classification system will aid in this effort.

Relating to the time element of this query, let’s look at a portion of particular pride. There is a somewhat complex CASE statement used to calculate the days\_available for each product:

CASE   
 WHEN p.expiry\_date IS NOT NULL THEN   
 GREATEST(1,   
 TRUNC(LEAST(SYSDATE, p.expiry\_date)) -   
 TRUNC(GREATEST(TRUNC(p.created\_at), TRUNC(SYSDATE - 365)))  
 )  
 ELSE   
 GREATEST(1,   
 TRUNC(SYSDATE) -   
 TRUNC(GREATEST(TRUNC(p.created\_at), TRUNC(SYSDATE - 365)))  
 )  
END AS days\_available

This query uses a calculation to determine days\_available for each product.Product performance is analyzed over a 365-day lookback period. For products with expiry dates, the query calculates availability from either the product creation date or a year ago to the expiry or current date. To prevent calculation errors, there is a minimum value of one day. With this approach, we can realistically assess how long a product has actually been available for purchase according to our window of analysis (“Khan Academy” 2023).

#### Figure 28. K-Pop/Anime Themed Product Performance Overview. Captured by author, May 31, 2025.

## Requirement Six: Cross-Platform Customer Behavior Analysis

### *Platform-Specific Promotion Effectiveness*

This query performs an analysis of [promotional campaign effectiveness](https://cherryberrykix66.github.io/sweet-seoul/SQL_queries_business_reporting.html#platform-specific) across the bakery’s online and in-store platforms. It measures how different types of promotions, i.e. discount, bundle, offer, gift, perform in each channel. This analysis uses a two-step approach starting with Common Table Expressions (CTEs) to identify orders that actually used promotions (“CTE in SQL” 2018). Then, the performance metrics are aggregated to evaluate promotional ROI and customer engagement patterns. The query specifically excludes promotions available on both platforms to yield only platform-specific insights. Recall that the bakery is gauging their dual-platform business model to ascertain if another store location or an expansion of their e-commerce platform is preferable.

#### Figure 29. Query results Product Category Preferences by Platform. Captured by author, May 31, 2025.

### *Product Category Preferences by Platform*

This query analyzes [how product sales perform between the online and in-store platforms](https://cherryberrykix66.github.io/sweet-seoul/SQL_queries_business_reporting.html#pref-plat). The products are categorized for greater insight into greater themes. The query calculates several key business metrics for analysis. A key metric is the revenue share percentage (revenue\_share); it shows what percentage each product category contributes to its platform's total revenue. This metric is achieved through a window function that partitions the data by platform i.e. OVER (PARTITION BY o.platform). This function produces the revenue share calculation by allowing each row to reference the total revenue for its specific platform while also maintaining the granular grouping by category (GeeksforGeeks 2024). With conditional aggregation through the CASE WHEN statement, this query identifies themed products by checking whether the kpop\_anime\_tie\_in field contains data. Only those items are counted toward the themed product percentage (theme\_product\_pct). By looking at the revenue share percentage and theme product percentage together, we can discover several valuable business insights about market positioning and product strategy.

The revenue share percentage (revenue\_share) shows which product categories are driving the most sales within each platform. Categories with high revenue share indicate strong market positions and customer demand. Meanwhile, lower-performing categories might need strategic attention. They might also be candidates for discontinuation. This metric helps identify the core revenue drivers for each platform. From this metric, we can also discern whether certain platforms have more diversified or concentrated sales patterns.

The theme product percentage (theme\_product\_pct) reveals how successfully K-pop and anime merchandise has penetrated different market segments. High percentages indicate that themed products have strong appeal in those categories. This phenomena suggests that we are effectively targeting fan communities. Lower percentages might represent untapped opportunities where themed variants could be introduced. There is the possibility that a lower percentage could also indicate categories where themed products have less appeal.

By viewing these metrics entandem, we can uncover some particularly interesting insights. Product categories with both a high revenue share and a high theme product percentage represent our strongest themed merchandise markets. These are prime opportunities for further investment and expansion. Conversely, product categories with high revenue share but lower theme product percentages might indicate missed opportunities. Therefore, introducing more themed options could capture additional market share from our existing customer base.

#### Figure 30. Query results Product Category Preferences by Platform. Captured by author, May 31, 2025.

## Requirement Seven: Gift Card Tracking

### *Gift Card Sales Overview Dashboard*

This query analyzes the [trends in gift card sales](https://cherryberrykix66.github.io/sweet-seoul/SQL_queries_business_reporting.html#gift-card) over time using a Common Table Expression (CTE) called monthly\_sales (“CTE in SQL” 2018). It’s formatted into monthly reporting periods. The query calculates several key metrics for each month:

1. the total number of cards issued (total\_cards\_issued)
2. the combined value of all cards issued (total\_value\_issued)
3. the average value per card (avg\_card\_value)

By reviewing these monthly metrics, we can gain insights into customer behavior, business performance and strategic opportunities. The total number of cards issued (total\_cards\_issued) reveals seasonal trends that are crucial for inventory planning and marketing strategies. We'll likely see spikes during holidays when customers are more likely to be seeking gifts for others. Understanding these patterns helps optimize our staffing, inventory levels, and promotional campaigns. Additionally, dips in certain months might indicate opportunities for targeted marketing to boost slower sales. With the combined value of all cards issued (total\_value\_issued) for the month, we can infer future revenue since gift cards represent deferred sales that will eventually convert to actual purchases. If we see a month-over-month growth in total value, this phenomena is indicative of business expansion. Likewise, declining trends might signal market saturation and increased competition. The monthly combined value also helps with cash flow planning. With gift cards, we’ll experience an immediate cashflow prior to inventory depletion. The average card value (avg\_card\_value) reveals significant insights about customer demographics and purchasing power. Increasing average values might indicate customers are becoming more comfortable with larger purchases. This growth could be the result of successful upselling strategies. Our marketing efforts may have exposed us to higher-income customer segments. Alternatively, declining average values could suggest increased price sensitivity from customers.

A particularly interesting aspect of this query is its ability to distinguish between gift purchases and self purchases. Gift purchases are identified when the recipient customer ID is different from the issuer customer ID and not null. This is a clear indicator that the customer bought a card for another person. Self purchases occur when the recipient ID is null or matches the issuer ID. It’s a common retail tactic to offer promotional bonuses for gift card purchases. Therefore, the bakery could run a gift card promotion with insight into its success via the self\_purchases metric.

Finally, this query counts unique purchasers to understand customer acquisition patterns. Let’s say the total number of gift cards were to increase month-over-month. This could be driven by existing customers buying more cards as opposed to new customers entering the market. By tracking unique purchasers, we can determine whether our business growth comes from customer acquisition or increased customer engagement. These cases would be indicated by new people buying cards and existing customers buying cards more frequently respectively. This distinction is important due to the fact that acquiring new customers typically costs more than retaining an existing customer base. Each strategy calls for different marketing approaches.

#### Figure 31. Query results Gift Card Sales Overview Dashboard. Captured by author, May 31, 2025.

# **Section Eight: User Interface vs. Physical Design**

For this portion of the assignment, we created a standup development environment in React. This was made possible through the online Integrated Development Environment (IDE), [CodeSandbox.io](http://codesandbox.io). In our [online prototype](https://wxtq5v.csb.app/), we have fleshed out a few use cases of our bakery management application.

## Product Management User Interface

Our first use case appears under the products tab. The product catalog management comes with complete CRUD (create, read, update, delete) functionalities. New products can be added to the database with detailed information including their category as well as their status as a K-pop/Anime Tie-in, limited edition or seasonal item. The existing products are displayed in a tabular format. Under the “Actions” column, the product can be either updated or deleted. We have included search features as well as drop down to filter by category and product type. These search and filter features will be particularly useful as the inventory grows.

Beyond mere listings, the product catalog offers inventory tracking and alerts. Recall that the PRODUCTS table had a min\_inventory\_level attribute. This threshold can be set to trigger automatic warnings when a product is running low on stock. The stock column in the sandbox application offers an at-a-glance view with color-coded indicators, i.e. green for in-stock and yellow for low stock). With this feature, the bakery management can track inventory levels across multiple product lines simultaneously. The streamlined product management provided with this user interface will reduce the manual tracking and enable quick updates for adequate pacing alongside seasonal menu changes.

Another valuable feature of the product catalog management use case is the tagging or banding classification. With this inclusion, the bakery management can easily identify seasonal products that may have limited availability over time. With these limited, seasonal flags, the operations staff can coordinate with the marketing team to deploy timely, promotional campaigns.

#### Figure 32. Add a new product dialog. Captured by author, May 31, 2025.

#### Figure 33. Edit a product dialog. Captured by author, May 31, 2025.

## Subscription Management User Interface

Recall from our business case abstract that the subscription box service is a core element in the bakery’s [business model](#_heading=h.6jaro3tnp0br). Customers can sign up for monthly delivery boxes containing Korean desserts and K-pop themed treats. This service operates on a recurring revenue model with different tiers (Standard and Premium) and flexible billing cycles. To support gifted subscriptions, the system can indicate when one person purchases a subscription or a recipient. Therefore, the customer acquisition model is expanded through gifting. Similar to the Product Management use case, we have a set of CRUD operations through which new subscriptions can be created in detail. These same details can be viewed, edited and deleted via the options in the “Actions” column. The subscription management user interface adequately handles the “Status” of Active, Paused, Cancelled for operation transparency. Subscriptions can be marked as cancelled With proper confirmation workflows, cancelled subscriptions can be handled without record deletion.

This user interface includes some aspects of the customer relationship management (CRM) process. Customer profiles with names, IDs and a subscription history are viewable. Again, the gift recipient tracking is useful from an additional perspective. The customers are properly segmented by subscription tier and status. There are also search and filtering capabilities to aid customer record lookups. In addition to some basic CRM support, there are some metrics and attributes that aid the bakery’s delivery and fulfillment planning. The next delivery date is tracked for each subscription record. Countdown timers display the days remaining before the next shipment. We can also manage the billing cycles at the record level. This will help the bakery management optimize the cashflow of the business. Up top, there is an upcoming delivery dashboard displaying the deliveries for the coming week. Alongside the weekly view are a collection of other useful business intelligence metrics including the total subscribers, monthly revenue, gift subscriptions and growth trends. Not only is this user interface designed with operational efficiencies in mind, there is visibility into the success of the subscription model alongside customer retention tactics.

#### Figure 34. View a customer subscription dialog. Captured by author, May 31, 2025.

## Orders Management User Interface

Our final use case is a comprehensive order management system that handles the complete lifecycle from creation to fulfillment. This tracking occurs in real-time as the statuses of Pending, Processing, Shipped and Delivered are displayed with color-coding alongside an audit trail. Shipping integration is further enhanced with the appearance of tracking numbers.

Additionally, this is an omnichannel order processing system capable of managing order from both the online and in-store channels supported by the bakery. It is important that a unified level of visibility across all platforms be accessible to the bakery staff operating at the physical store front as well as the e-commerce site. The online orders have marketing attributes present as well. With marketing channel tracking, an analysis can reveal which promotional channels drive the most valuable customers. Hence an ROI analysis can be established for each marketing investment. With a diversity in platforms comes a need to support multiple payment types such as credit cards, cash and gift cards. If the order is identified as a gift, there are indicators for special handling.

Like our subscription model interface, there are customer intelligence features as name, IDs and order histories appear. The system tracks the complete customer journey from order placement through delivery. Therefore, a proactive customer service model is enabled. Gift order tracking allows for special handling of purchases intended as presents. This last point is crucial to a business serving non-essential, themed items.

With order composition tracking available through the "View Details” icon under “Actions”, the bakery staff can see detailed line items, quantities, and pricing just as they saw in-depth information concerning subscriptions. By tracking order status and contents, the system helps the bakery staff coordinate inventory needs and production scheduling. In some cases, it’s important to know exactly what products are in processing orders. This helps plan daily production runs for perishable food items.

We also have another operational dashboard from which real-time metrics can be viewed including daily orders, processing queue, shipped items and revenue. There are also performance indicators with percentage changes and trend analysis. The revenue is tracked by weekly comparisons. Through revenue and growth tracking the dashboard enables data-driven business decisions concerning platform expansion, product development, and marketing strategies.

#### Figure 35. View an order dialog. Captured by author, May 31, 2025.

# **Section Nine: Considerations of Maintenance and Evolution**

## Scalability and Performance Optimization

Our current implementation in Oracle 19c provides a solid foundation for the Sweet Seoul Korean Bakery database system. That said, anticipated growth will necessitate performance tuning and infrastructure scaling. With a successful e-commerce platform, the customer base will likely expand beyond the local Denver market. The product catalog is likely to grow with the inclusion of more seasonal, limited-edition and regular items. With this growth, query optimization will become a critical concern. Particularly, the subscription box service will create increased volumes of transactional data in need of enhanced indexing strategies. In time, it’s possible that table partitioning will need to take place for historical data (GeeksforGeeks 2016). As data volumes reach enterprise levels, the information systems team associated with the bakery will need to implement Oracle's Advanced Compression and In-Memory features (“Reduce Database Storage and Optimize Performance” 2023).

## Enhanced Analytics and Machine Learning Integration

To better support the product demand forecasting in place, predictive analytics capabilities are a candidate for future enhancements. This enhancement will serve other current prediction capabilities present within the system including tracking customer churn and personalized product recommendations. With the existing RFM segmentation framework, we have an excellent foundation from which machine learning algorithms can operate. There is potential to automate the identification of customer behavior patterns and optimize subscription box content. In order to implement these machine learning ambitions, it’s recommended that the database system be integrated with Oracle Machine Learning (“Machine Learning for Database and Big Data Environments” 2024). There are other options external to the platform such as Azure ML (Microsoft 2024). These enhancements could enable a sophisticated recommendation engine on the e-commerce website where customer product preferences are leveraged alongside an individual’s purchasing history.

## Customer Experience Personalization

Our previous discussion on machine learning touched up the power of customer preferences. Indeed, the foundation for customer preferences and feedback analysis exists. Although the future system iterations could support dynamic personalization engines. This enhancement would include a real-time recommendation system for subscription boxes. The bakery could deploy personalized marketing campaigns inspired by the behavioral patterns of individual customers. We could loop in the customer segmentation features to explore adaptive pricing strategies. The gifting features present within the system could be enhanced with sharing abilities via social media (“A Complete Guide to Personalization Engines | Braze” 2024).

## Regulatory Compliance and Data Privacy

As the bakery expands into new markets, compliance is likely to become a system requirement. The database architecture should be enhanced to automated consent management and residency requirements. Comprehensive audit trails will need to be implemented to better support regulatory reporting. As the volumes of customer data grow, there will be a need for data masking and encryption strategies. Larger databases are highly prized by malicious actors.

## Project Reflection

This project has really exposed us to the iterative nature of database development. It’s paramount that technical architecture be aligned with business strategy. This is a priority that must take place at every stage within the Database Life Cycle. The opportunity to implement and test our SQL statements in the Oracle 19c environment revealed numerous design considerations. Such concepts were not apparent in the conceptual modeling phase. This is particularly true for the CLOB field implementation and identity column management.

During our development of the React prototype interface, we discovered the importance of considering the end-user experience alongside database design. Features that seemed logical from a data modeling perspective required refinement when seen through operational workflows. This experience emphasizes the interconnected nature of database design and user interface development.

Most importantly, this project demonstrated how an intentional and thoughtful database design can develop into a strategic business asset. The comprehensive analytics capabilities built into our system don't just support current operations. They provide the foundation for data-driven growth strategies. This is the business intelligence the Sweet Seoul Korean Bakery will come to rely on in an increasingly competitive market. Distinguishment in their business domain not only comes from novelty products but an attention to personalized marketing strategies, effective operational workflows, and customer satisfaction.

# **References**

“A Complete Guide to Personalization Engines | Braze.” 2024. Braze.com. 2024.   
 https://www.braze.com/resources/articles/a-complete-  
 guide-to-personalization-engines.

Atzeni, Paolo, and D. Stott Parker. 1982. “Assumptions in Relational Database   
 Theory.” Proceedings of the 1st ACM SIGACT-SIGMOD Symposium   
 on Principles of Database Systems - PODS ’82.   
 https://doi.org/10.1145/588111.588113.

Barick, Ankur. 2020. “Normalization Process in DBMS.” GeeksforGeeks.   
 August 10, 2020. https://www.geeksforgeeks.org/normalization-  
 process-in-dbms/.

Bond, Conor. 2023. “Conversion Rate Benchmarks: Find out How YOUR   
 Conversion Rate Compares.” Wordstream.com. 2023.   
 https://www.wordstream.com/blog/ws/2019/08/19/conversion-  
 rate-benchmarks.

Caldwell, Austin. 2022. “How to Calculate Customer Lifetime Value.” Oracle NetSuite.   
 July 21, 2022. https://www.netsuite.com/portal/resource/articles/ecommerce/  
 customer-lifetime-value-clv.shtml.

Claude (Anthropic). "Troubleshooting Business Intelligence Test: Inventory Alerts View."   
 Claude AI. May 30, 2025. AI-generated code. https://claude.ai.

codezone. 2024. “Understanding GENERATED ALWAYS as IDENTITY vs. GENERATED   
 by DEFAULT as IDENTITY in Oracle.” Medium. October 19, 2024.   
 https://medium.com/@codezone/understanding-generated-always-  
 as-identity-vs-generated-by-default-as-identity-in-oracle-a9a8fd9e8e0d.

“Constraint.” 2025. Oracle Help Center. SQL Language Reference. 2025.   
 https://docs.oracle.com/en/database/oracle/oracle-database/  
 21/sqlrf/constraint.html.

Cornel and Morris. "Data versus Information." In Database Systems Design,   
 Implementation, & Management. 13th ed. Boston, MA: Cengage   
 Learning, 2018.

“CTE in SQL.” 2018. GeeksforGeeks. September 17, 2018.   
 https://www.geeksforgeeks.org/cte-in-sql/.

“Exception Handling in PL/SQL.” 2018. GeeksforGeeks. April 28, 2018.   
 https://www.geeksforgeeks.org/exception-handling-plsql/.

“Formatting Query Results.” n.d. Docs.oracle.com.   
 https://docs.oracle.com/cd/A87860\_01/doc/server.817/a82950/ch4.htm.

GeeksforGeeks. 2015. “Introduction of Database Normalization.” GeeksforGeeks.   
 July 2, 2015. https://www.geeksforgeeks.org/introduction-of-database  
 -normalization/.

GeeksforGeeks. 2016. “Indexing in Databases | Set 1 - GeeksforGeeks.”   
 GeeksforGeeks. July 15, 2016. https://www.geeksforgeeks.org/  
 indexing-in-databases-set-1/.

GeeksforGeeks. 2024. “Database Development Life Cycle.” GeeksforGeeks.   
 September 9, 2024. https://www.geeksforgeeks.org/  
 database-development-life-cycle/.

GeeksforGeeks. 2024. “Difference between PARTITION by and GROUP by in PL/SQL.”   
 GeeksforGeeks. March 18, 2024. https://www.geeksforgeeks.org/difference-between-  
 partition-by-and-group-by-in-plsql/.

GeeksForGeeks. 2024. “What Is Data Dictionary?” GeeksforGeeks. April 23, 2024.   
 https://www.geeksforgeeks.org/what-is-data-dictionary/.

James, Jenna. “Sweet Seoul Entity Relationship Diagram for Oracle 19c   
 Implementation.” Diagram created using Mermaid.live, May 26, 2025.

James, Jenna. “Sweet Seoul Normalization Diagram.” Diagram created using   
 Mermaid.live, May 26, 2025.

Kedem, Ram. 2015. “Oracle String Concatenation.” Upscale Analytics.   
 August 30, 2015. https://ramkedem.com/en/oracle-string-concatenation/.

“Khan Academy.” 2023. Khanacademy.org. 2023. https://www.khanacademy.org/  
 computing/computer-programming/sql/more-advanced-  
 sql-queries/pt/calculating-results-with-case.

“Machine Learning for Database and Big Data Environments.” 2024.   
 Oracle.com. 2024. https://www.oracle.com/artificial-intelligence/  
 database-machine-learning/.

McDermid, David, and Sarika Surampudi. 2025. “USER\_SYS\_PRIVS.” Oracle   
 Help Center. Database Reference. 2025. https://docs.oracle.com/en/database/  
 oracle/oracle-database/19/refrn/USER\_SYS\_PRIVS.html.

Microsoft. 2024. “What Is Azure Machine Learning? - Azure Machine Learning.”   
 Learn.microsoft.com. January 29, 2024. https://learn.microsoft.com/en-us/azure/  
 machine-learning/overview-what-is-azure-machine-learning?view=azureml-api-2.

“Oracle Data Types.” 2025. Oracle.com. 2025. https://docs.oracle.com/cd/  
 B28359\_01/server.111/b28318/datatype.htm.

“QUOTA in Tablespace - Learnomate Technologies.” 2024. Learnomate Technologies.   
 June 12, 2024. https://learnomate.org/qouta-in-tablespace/.

“Reduce Database Storage and Optimize Performance.” 2023.   
 Oracle.com. 2023. https://www.oracle.com/database/  
 advanced-compression/.

Roeser, Mary Beth, Drew Adams, Lance Ashdown, Vikas Arora, Thomas Baby,   
 Hermann Baer, Yasin Baskan, et al. n.d. “Data Types.” Oracle Help Center.   
 https://docs.oracle.com/en/database/oracle/oracle-database/  
 19/sqlrf/Data-Types.html#GUID-7B72E154-677A-4342-A1EA-  
 C74C1EA928E6.

Salsi, Hava. 2024. “The Marketing Funnel: What It Is & How It Works.” Semrush Blog.   
 August 30, 2024. https://www.semrush.com/blog/marketing-funnel/.

“Smoke Testing | Software Testing.” 2019. GeeksforGeeks. April 18, 2019.  
 https://www.geeksforgeeks.org/smoke-testing-software-testing/.

“Surrogate Key in DBMS.” 2021. GeeksforGeeks. June 1, 2021.   
 https://www.geeksforgeeks.org/surrogate-key-in-dbms/.

Upadhyay, Mithlesh . 2019. “Boyce-Codd Normal Form (BCNF) -   
 GeeksforGeeks.” GeeksforGeeks. July 31, 2019.   
 https://www.geeksforgeeks.org/boyce-codd-normal-form-bcnf/.

Wright, Gavin. 2021. “What Is RFM (Recency, Frequency, Monetary) Analysis and   
 How Does It Work?” SearchDataManagement. May 2021.   
 https://www.techtarget.com/searchdatamanagement/definition/RFM-analysis.

1. James, Jenna. “Sweet Seoul Entity Relationship Diagram for Oracle 19c Implementation.” Diagram created using Mermaid.live, May 26, 2025. [↑](#footnote-ref-1)
2. James, Jenna. “Sweet Seoul Normalization Diagram.” Diagram created using Mermaid.live, May 26, 2025. [↑](#footnote-ref-2)