NAIL TECHNICIAN

Presented By

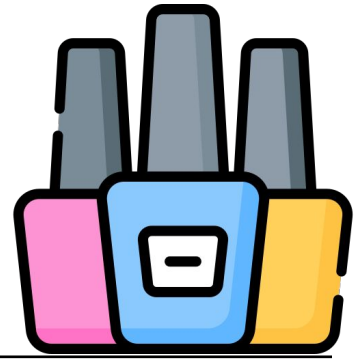
Yoav zucker, Oranit Yogev and Nave maymon

Department of Software Engineering

Shenkar College

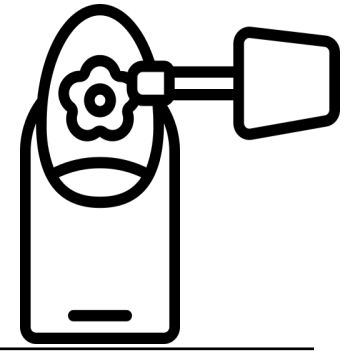
BUSINESS CONTEXT

In the dynamic and growing world of beauty and self-care, nail salons play an important role in creating personalized experiences for customers.
This project uses data to better understand what customers love, improve service quality, and make daily operations smoother — all while making the salon experience even better.

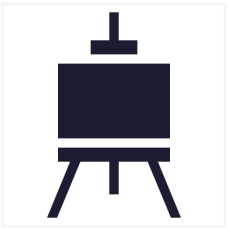


INDUSTRY OVERVIEW & BUSINESS PROBLEM

- **Industry**
  - Beauty & Personal Care – Nail Salons
- **Business Problem**
  - Which colors are preferred by customers
  - Need for smart inventory management
  - Customer retention through predicting return visits



KEY STAKEHOLDERS



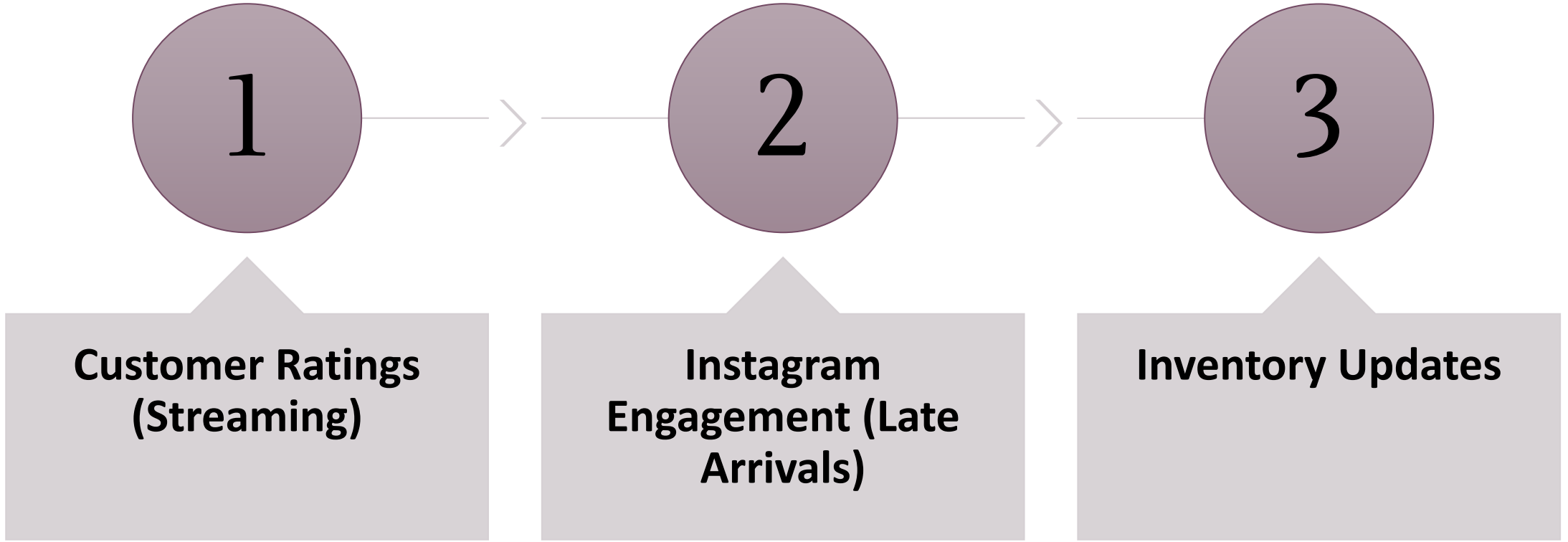
Salon Owners

Marketing Team

Inventory Managers

Nail Technicians

DATA SOURCES EXPLANATION



TECHNICAL DESIGN

DATA PIPELINE ARCHITECTURE

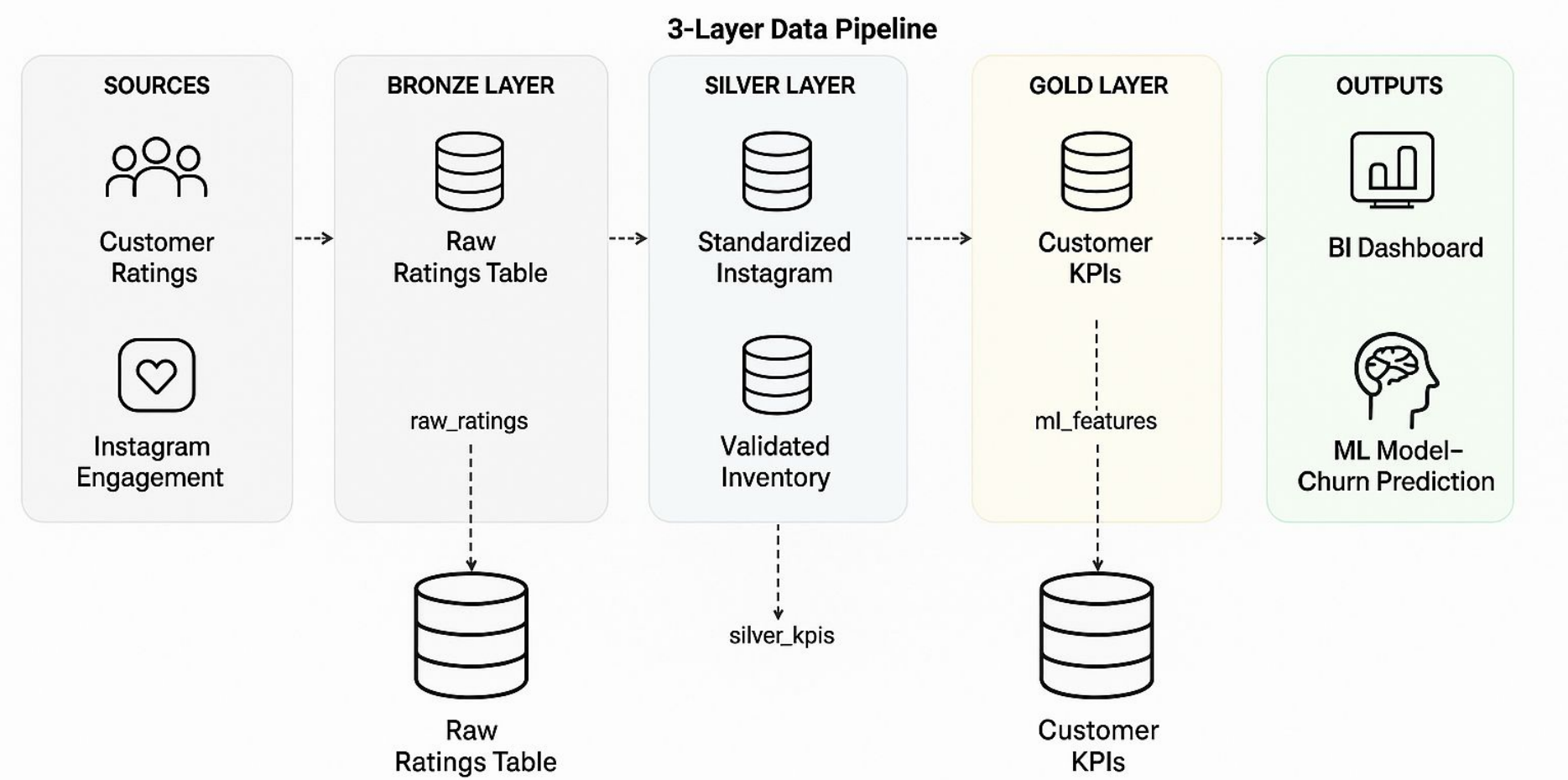


TABLE SCHEMAS

TABLE SCHEMAS - BRONZE - **OLD**

raw_ratings		
INT	rating_id	PK
INT	customer_id	
INT	branch_id	
INT	employee_id	
INT	treatment_id	
FLOAT	rating_value	
STRING	comment	
DATETIME	timestamp	

raw_instagram		
INT	post_id	PK
INT	color_id	
INT	likes	
INT	comments	
DATE	post_date	
DATE	ingestion_date	

raw_inventory		
INT	record_id	PK
INT	branch_id	
INT	color_id	
INT	quantity_used	
TIMESTAMP	timestamp	

TABLE SCHEMAS - BRONZE - **NEW**

raw_ratings	
INT	customer_id
INT	branch_id
INT	employee_id
INT	treatment_id
FLOAT	rating_value
STRING	comment
TIMESTAMP	timestamp
TIMESTAMP	processed_timestamp
STRING	_kafka_topic
INT	_kafka_partition
BIGINT	_kafka_offset

raw_instagram		
INT	post_id	PK
INT	color_id	
INT	likes	
INT	comments	
DATE	post_date	
DATE	ingestion_date	

raw_inventory		
INT	record_id	PK
INT	branch_id	
INT	color_id	
INT	quantity_used	
TIMESTAMP	timestamp	

TABLE SCHEMAS - DIM - **OLD**

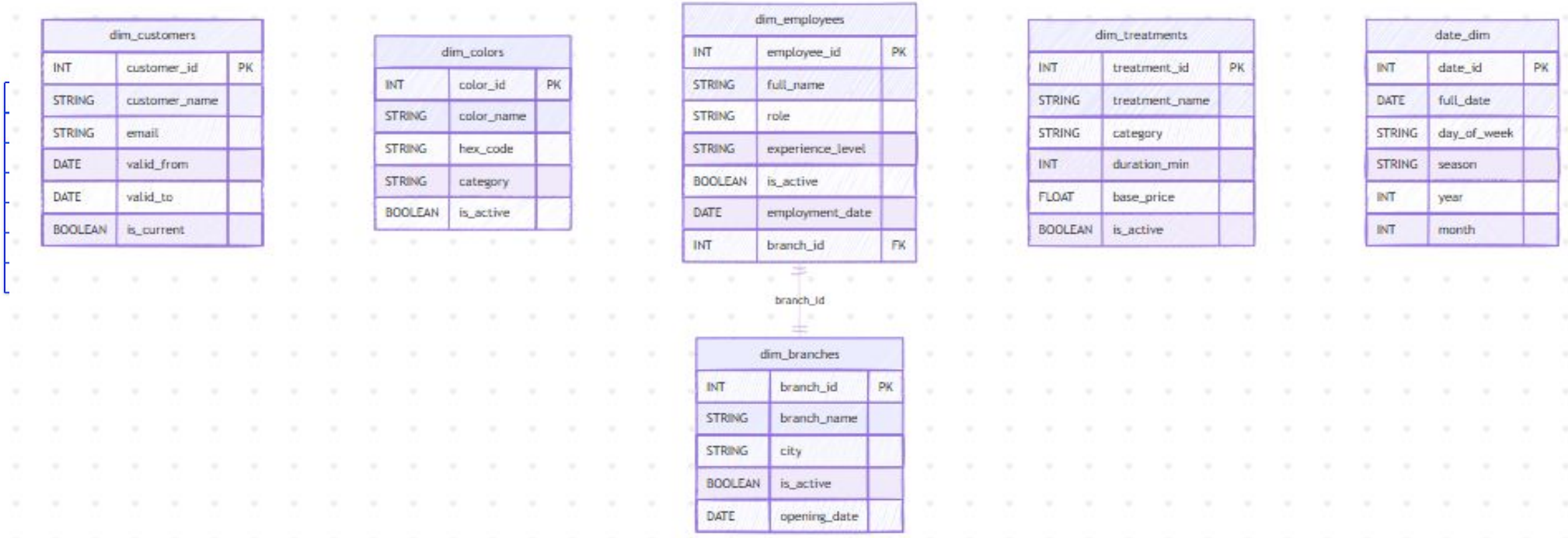


TABLE SCHEMAS - DIM - **NEW**

dim_customers	
INT	customer_id
STRING	first_name
STRING	last_name
STRING	email
STRING	phone
STRING	address
STRING	city
DATE	start_date
DATE	end_date
BOOLEAN	is_current
TIMESTAMP	created_timestamp

dim_colors	
INT	color_id
STRING	name
STRING	hex_code
STRING	category
BOOLEAN	active
TIMESTAMP	created_timestamp

dim_employees	
INT	employee_id
STRING	first_name
STRING	last_name
STRING	role
INT	experience_years
BOOLEAN	active
DATE	hire_date
INT	branch_id
TIMESTAMP	created_timestamp

dim_branches	
INT	branch_id
STRING	name
STRING	city
STRING	address
BOOLEAN	active
DATE	opening_date
TIMESTAMP	created_timestamp

dim_treatments	
INT	treatment_id
STRING	name
DECIMAL	price
INT	duration_minutes
BOOLEAN	active
TIMESTAMP	created_timestamp

dim_date	
STRING	date_id
DATE	date
STRING	day_of_week
INT	day_of_month
STRING	month
INT	month_number
INT	quarter
INT	year
STRING	season
BOOLEAN	is_weekend
BOOLEAN	is_holiday
TIMESTAMP	created_timestamp

TABLE SCHEMAS - SILVER - **OLD**

silver_ratings		
INT	rating_id	PK
INT	customer_id	
INT	branch_id	
INT	employee_id	
INT	treatment_id	
DOUBLE	rating_value	
STRING	comment	
DATE	rating_date	
TIMESTAMP	timestamp	
TIMESTAMP	ingestion_time	

silver_instagram		
INT	post_id	PK
INT	color_id	FK
INT	date_id	FK
INT	likes	
INT	comments	
DATE	post_date	
DATE	ingestion_date	
STRING	season	

silver_inventory		
INT	record_id	PK
INT	branch_id	PK
INT	color_id	PK
INT	quantity_available	
BOOLEAN	is_shortage	
DATE	report_date	
TIMESTAMP	ingestion_time	

TABLE SCHEMAS - SILVER - **NEW**

silver_ratings	
INT	rating_id
INT	customer_id
INT	branch_id
INT	employee_id
INT	treatment_id
FLOAT	rating_value
STRING	comment
DATE	rating_date
TIMESTAMP	timestamp
TIMESTAMP	ingestion_time
FLOAT	data_quality_score

silver_instagram		
INT	post_id	PK
INT	color_id	FK
INT	date_id	FK
INT	likes	
INT	comments	
DATE	post_date	
DATE	ingestion_date	
STRING	season	

silver_inventory	
INT	record_id
INT	branch_id
INT	color_id
INT	quantity_used
DATE	report_date
TIMESTAMP	ingestion_time

TABLE SCHEMAS - GOLD - **OLD**

gold_customer_metrics		
INT	customer_id	PK
FLOAT	avg_rating	
INT	total_sessions	
FLOAT	last_rating	
INT	time_since_last_visit	
BOOLEAN	is_vip	
DATE	last_visit_date	

gold_branch_kpis		
INT	branch_id	PK
STRING	branch_name	
FLOAT	avg_rating	
INT	total_sessions	
DECIMAL	total_revenue	
INT	num_active_employees	

gold_color_engagement		
INT	color_id	PK
INT	total_likes	
INT	total_comments	
INT	times_used	
STRING	top_season	
FLOAT	engagement_score	

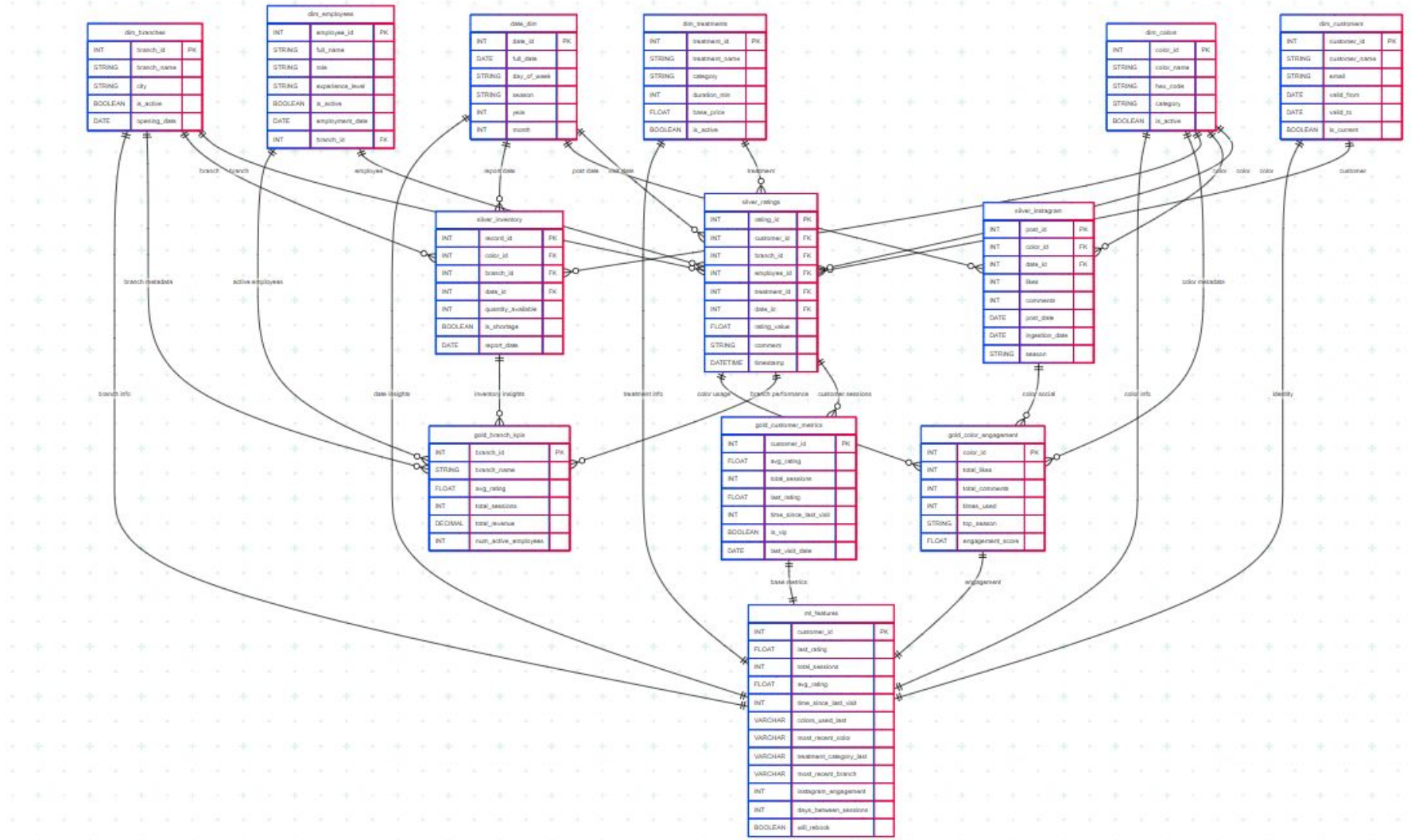
TABLE SCHEMAS - GOLD - **NEW**

gold_customer_metrics	
INT	customer_id
FLOAT	avg_rating
INT	total_ratings
FLOAT	last_rating
BOOLEAN	is_vip

gold_branch_kpis	
INT	branch_id
STRING	branch_name
FLOAT	avg_rating
INT	total_ratings
INT	num_active_employees

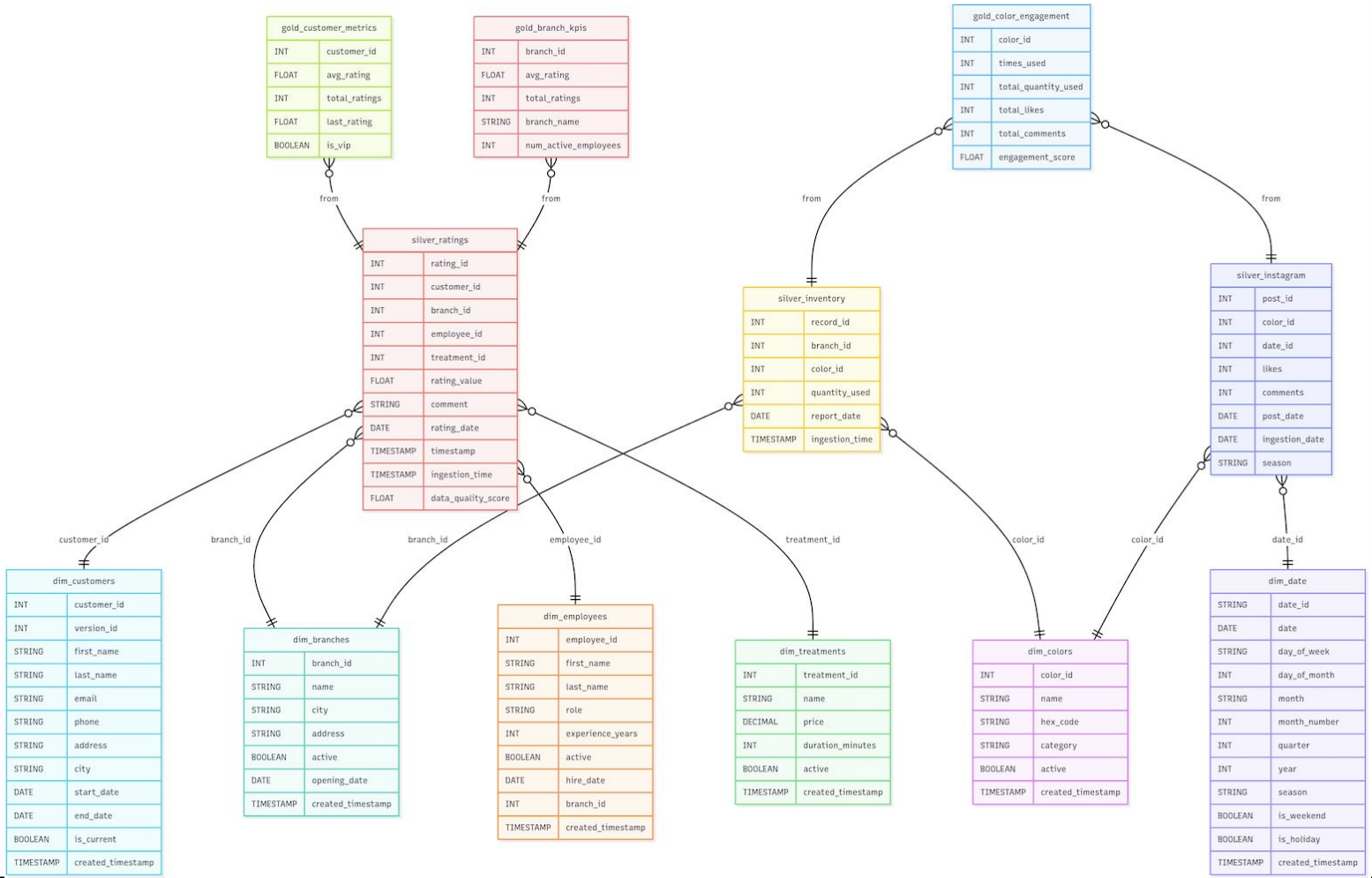
gold_color_engagement		
INT	color_id	PK
INT	total_likes	
INT	total_comments	
INT	times_used	
STRING	top_season	
FLOAT	engagement_score	

ARCHITECTURE DIAGRAM - **OLD**

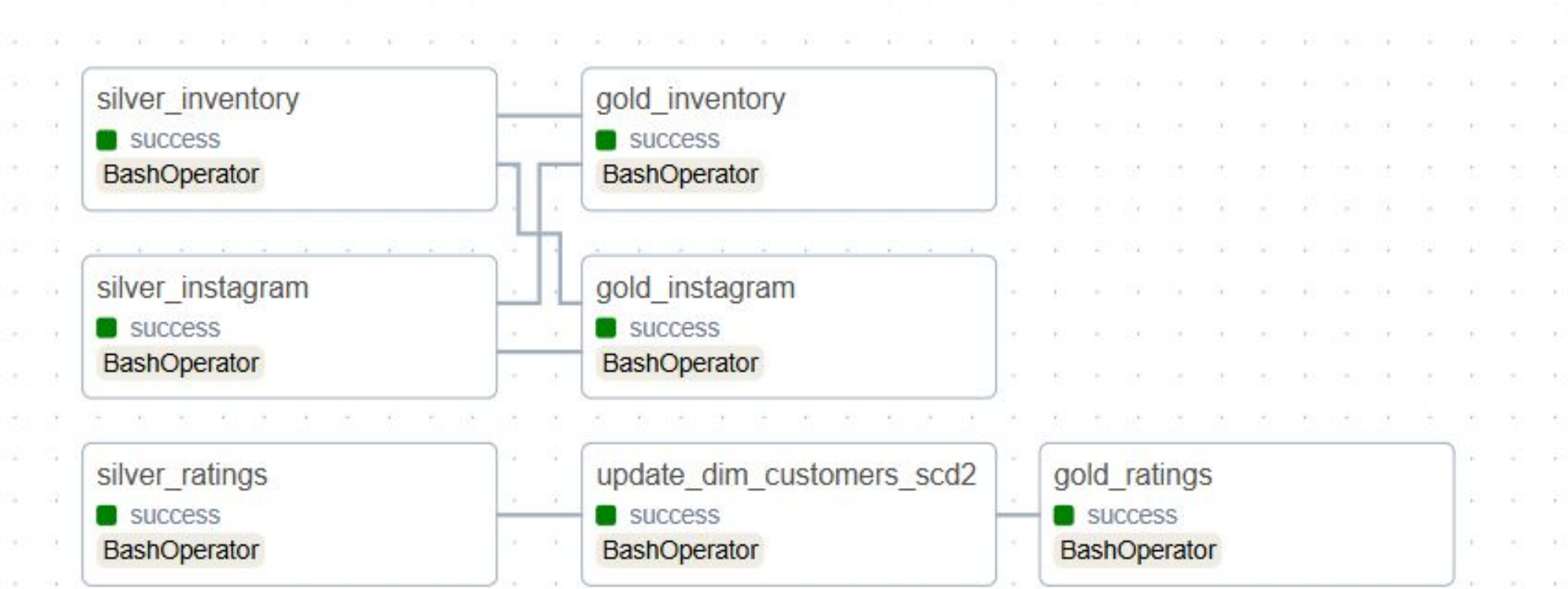


link

ARCHITECTURE DIAGRAM - **NEW**



DAGS


silver_inventory
success
BashOperator
gold_inventory
success
BashOperator
silver_instagram
success
BashOperator
gold_instagram
success
BashOperator
silver_ratings
success
BashOperator
update_dim_customers_scd2
success
BashOperator
gold_ratings
success
BashOperator

DAGS / **Silver Layer**

These tasks transform raw/bronze data into cleaned, analytics-ready "silver" tables

DAGS / Silver Layer Tasks - **silver_inventory**

What it does:

Cleans and processes raw inventory data into the silver_inventory table

What is cleaned:

Removes duplicate inventory records

Handles missing or null values

Standardizes date formats and data types

Corrects inconsistent product or branch names

Filters out invalid or outlier inventory entries

Tables used:

Input: bronze_inventory

Output: silver_inventory

silver_inventory	
INT	record_id
INT	branch_id
INT	color_id
INT	quantity_used
DATE	report_date
TIMESTAMP	ingestion_time

DAGS / Silver Layer Tasks - **silver_instagram**

What it does:

Cleans and processes raw Instagram data into the silver_instagram table

What is cleaned:

Removes duplicate social media posts or records

Handles missing fields

Standardizes timestamp formats

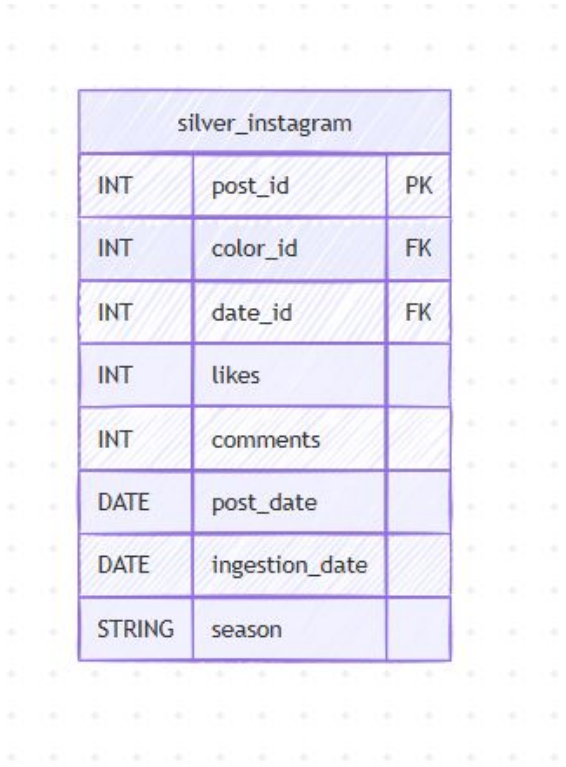
Cleans up text fields

Filters out irrelevant or spammy posts

Tables used:

Input: bronze_instagram

Output: silver_instagram



silver_instagram		
INT	post_id	PK
INT	color_id	FK
INT	date_id	FK
INT	likes	
INT	comments	
DATE	post_date	
DATE	ingestion_date	
STRING	season	

DAGS / Silver Layer Tasks - **silver_ratings**

What it does:

Cleans and processes raw ratings data into the silver_ratings table

What is cleaned:

Removes duplicate ratings

Handles missing or null values

Ensures ratings are within valid ranges

Standardizes date/time formats

Filters out invalid or suspicious ratings

Tables used:

Input: bronze_ratings

Output: silver_ratings

silver_ratings	
INT	rating_id
INT	customer_id
INT	branch_id
INT	employee_id
INT	treatment_id
FLOAT	rating_value
STRING	comment
DATE	rating_date
TIMESTAMP	timestamp
TIMESTAMP	ingestion_time
FLOAT	data_quality_score

DAGS / **scd2**

Tracks and preserves historical changes in customer data by applying Slowly Changing Dimension Type 2 (SCD2) logic to the customer dimension table

DAGS / **dim_customers_scd2**

What it does:

Updates the dim_customers table using Slowly Changing Dimension Type 2 logic to track historical changes in customer data.

What is cleaned and processed:

Identifies changes in customer attributes by comparing new data from silver_ratings with existing records in dim_customers. For any customer whose attributes have changed, the current record in dim_customers is marked as "inactive".

Inserts a new record for the customer with the updated attributes, marked as "active" (with a new start date and no end date).

Ensures there are no duplicate active records for the same customer.

Tables used:

Input: silver_ratings, dim_customers

Output: dim_customers

dim_customers	
INT	customer_id
STRING	first_name
STRING	last_name
STRING	email
STRING	phone
STRING	address
STRING	city
DATE	start_date
DATE	end_date
BOOLEAN	is_current
TIMESTAMP	created_timestamp

DAGS / **Gold**

Aggregates and enriches cleaned data to produce high-level business metrics and insights for analytics and reporting

DAGS / **gold_inventory**

What it does:

Calculates key performance indicators for each branch based on inventory data and stores them in the gold_branch_kpis table

What is aggregated/calculated:

Aggregates inventory data by branch

Calculates metrics such as average inventory levels, out-of-stock events, and inventory value

Joins with dimension tables for branch details

Ensures all metrics are up-to-date and consistent for business reporting

Tables used:

Input: silver_inventory, dimension tables

Output: gold_branch_kpis

gold_branch_kpis	
INT	branch_id
STRING	branch_name
FLOAT	avg_rating
INT	total_ratings
INT	num_active_employees

DAGS / **gold_ratings**

What it does:

Calculates customer metrics based on ratings and customer dimension data, storing results in the gold_customer_metrics table

What is aggregated/calculated:

Aggregates ratings data by customer

Calculates customer satisfaction scores and loyalty indicators

Joins with the updated dim_customers table for customer details and history

Identifies high-value or at-risk customers based on their rating patterns

Tables used:

Input: silver_ratings, dim_customers

Output: gold_customer_metrics

gold_customer_metrics	
INT	customer_id
FLOAT	avg_rating
INT	total_ratings
FLOAT	last_rating
BOOLEAN	is_vip

DAGS / **gold_instagram**

What it does:

Computes engagement metrics for each color based on Instagram/social media data and stores them in the gold_color_engagement table

What is aggregated/calculated:

Aggregates Instagram data by color
Calculates engagement rates and trends for each color

Joins with dimension tables for color details
Identifies top-performing colors and social media trends

Tables used:

Input: silver instagram, dimension tables
Output: gold_color_engagemen

gold_color_engagement		
INT	color_id	PK
INT	total_likes	
INT	total_comments	
INT	times_used	
STRING	top_season	
FLOAT	engagement_score	

Implementation **challenges** and **solutions**

Debugging Data Quality Issues:

- Challenge: We often found unexpected duplicates and missing values in our raw data, which caused errors in later pipeline stages.
- Solution: We added thorough data cleaning and validation steps in the Silver layer, and used print statements and small test runs to catch issues early.

Managing Docker and Spark Integration:

- Challenge: Setting up Spark jobs to run smoothly inside Docker containers and orchestrating them with Airflow was confusing at first.
- Solution: We followed online tutorials, carefully mapped volumes and dependencies, and used Airflow logs to debug and fix integration problems.