

Q1) ShopSmart – Centralized Analytics Data Warehouse

You have been appointed as a data engineer at **ShopSmart**, a retail organization that operates over 100 physical stores along with an online shopping website. The company wants to develop a **central analytics warehouse** to study sales trends, customer behavior, and inventory across all platforms.

Step 1: Data Sources

The data required for this project will come from the following sources:

- **Point of Sale (POS) systems** in physical stores
 - **E-commerce platform** – orders, cart activity, browsing history
 - **Customer database** – demographics, loyalty programs, contact details
 - **Inventory systems** – warehouse stock and store-level stock data
 - **External sources** – promotional campaigns, festive seasons, competitor pricing data
-

Step 2: Storage Architecture

To manage and organize data efficiently, a combination of storage systems is suggested:

Storage System	Purpose
Data Lake	Stores raw, unprocessed, and semi/unstructured data such as JSON logs from the online platform
Data Warehouse	Contains cleaned, structured, analytics-ready data for reporting and BI tools

Schema Design:

- **Star Schema** → Faster performance for analytics and dashboards
- **Snowflake Schema** → More normalized design for hierarchical data

Example Fact Table – sales_fact:

Field	Description
sales_id	Unique sale entry
store_id	Store code
product_id	Product reference
customer_id	Customer reference
date_id	Date key
quantity_sold	Items sold
revenue	Total sale amount

Field	Description
discount_applied	Discount value
sales_channel	Store or online
payment_method	Cash/Card/UPI
cost_of_goods_sold	Product cost
profit_margin	Profit earned

Example Dimension Tables:

- **customer_dim:** customer_id, first_name, last_name, email, phone, age, gender, loyalty_score, city, state, country
 - **product_dim:** product_id, product_name, category, brand, supplier_id, price, cost_price, launch_date, color, material
 - **store_dim:** store_id, store_name, city, state, region, manager_name, opening_date
 - **date_dim:** date_id, date, day, week, month, quarter, year, is_holiday, is_weekend
-

Step 3: ETL Pipeline

1. Extract Data Example:

```
SELECT store_id, product_id, sale_date, quantity, total_amount
FROM pos_sales
WHERE sale_date = CURRENT_DATE;
```

2. Transform Data:

- Fill missing values
- Standardize formats (currency, dates)
- Aggregate: sales & quantity per product per store

```
SELECT store_id, product_id, SUM(quantity) AS total_quantity,
SUM(total_amount) AS total_revenue
FROM pos_sales
WHERE sale_date BETWEEN CURRENT_DATE - INTERVAL '7' DAY AND CURRENT_DATE
GROUP BY store_id, product_id;
```

3. Load into Warehouse:

```
INSERT INTO sales_fact (store_id, product_id, date, quantity_sold, revenue)
SELECT store_id, product_id, sale_date, SUM(quantity), SUM(total_amount)
FROM pos_sales_transformed
GROUP BY store_id, product_id, sale_date;
```

Step 4: Analytics & AI Use Cases

- Find **top-performing products** by store
- **Customer segmentation** based on buying frequency
- Revenue forecasting using **AI algorithms**

- Personalized **recommendation systems**
 - **Dynamic pricing** strategies
-

Step 5: Business Impact

Outcome	Benefit
Sales Insight	Identify high-revenue products & stores
Customer Retention	Personalized promotions, loyalty programs
Inventory Optimization	Reduce stockouts and excess stock
AI Capabilities	Demand forecasting, recommendations

Q2) QuickEats – Scalable Real-Time Data Architecture

You are working as a data engineer at **QuickEats**, an online food delivery company that operates across several cities. The organization is facing challenges like slow analytics, poor scalability, and inability to process real-time data efficiently. To solve these issues, a **hybrid data architecture** is recommended.

1. Recommended Architecture: Hybrid Real-Time & Batch Model

This architecture combines both real-time streaming and traditional batch processing to ensure fast, scalable, and reliable analytics.

(A) Real-Time Data Ingestion

- Continuously captures **orders, delivery updates, GPS data, and user actions** as they occur.
 - Technologies:
 - **Apache Kafka or AWS Kinesis**
 - **Benefits:**
 - ✓ Real-time delivery tracking
 - ✓ Faster allocation of delivery partners
 - ✓ Instant fraud detection or order failure alerts
-

(B) Scalable Data Storage Layer

Component	Description	Usage
Data Lake	Stores raw, semi-structured data (clickstreams, mobile logs, GPS routes)	Future analysis and ML
Data Warehouse	Contains cleaned, structured, analytics-ready tables	Reporting, dashboards, BI tools
Schema Design	Star Schema	Optimized for analytics

Example: Orders Fact Table (orders_fact)

Column	Description
order_id	Unique order number
customer_id	Customer reference
restaurant_id	Restaurant reference
delivery_partner_id	Rider assigned
date_id	Time dimension
order_time	Order timestamp
delivery_time	Actual delivery time
total_amount	Revenue generated
discount_applied	Discount value
payment_method	Cash/Wallet/Card
city	Location of customer/restaurant
profit_margin	Revenue - cost_of_food - delivery_fee

2. Analytics & Business Intelligence Layer

- Used for **real-time dashboards**, KPI tracking, operational monitoring
- Examples of metrics:
 - Active orders in real-time
 - Average delivery time per city
 - Daily revenue per restaurant
 - Top-selling food items

Sample Analytics Query

```
SELECT r.restaurant_name, DATE_TRUNC('month', o.order_time) AS month,
```

```

        SUM(o.total_amount) AS total_revenue
FROM orders_fact o
JOIN restaurant_dim r ON o.restaurant_id = r.restaurant_id
GROUP BY r.restaurant_name, month
ORDER BY total_revenue DESC;

```

3. ETL / ELT Pipeline Description

Step	Description
Extract	Retrieve data from orders, user app interactions, payment APIs, delivery GPS logs
Transform	Remove duplicates, fix timestamps, convert currencies, calculate delivery duration
Load	Insert into analytics data warehouse or real-time database

Example Transformation Query:

```

SELECT order_id, customer_id, restaurant_id, delivery_partner_id,
       TIMESTAMPDIFF(MINUTE, order_time, delivery_time) AS delivery_duration_minutes,
       total_amount - cost_of_food - delivery_fee AS profit
FROM orders_raw
WHERE order_time >= CURRENT_DATE - INTERVAL '1' DAY;

```

4. Enhanced Dimension Tables

Table Name	Purpose
customer_dim	Stores personal details, loyalty score, preferred cuisine
restaurant_dim	Contains restaurant name, cuisine type, ratings, owner info
delivery_partner_dim	Tracks rider details, average delivery time, vehicle type
menu_item_dim	List of food items, price, ingredients, veg/non-veg type
date_dim	Holds formatted date values for analytics (day, week, month, etc.)

5. Final Business Benefits

- ✓ Handles **thousands of orders per hour** efficiently
- ✓ Enables **real-time tracking & dashboards**
- ✓ Enhances **customer satisfaction** via faster deliveries
- ✓ Supports **AI features** such as delivery time prediction and personalized food recommendations

Q3) StreamFlix – Real-Time Streaming Analytics & AI-Ready Data Warehouse

You are a data engineer for **StreamFlix**, a global video streaming service (similar to Netflix). The platform handles **millions of user events every day**, such as play, pause, search, likes, comments, and more. StreamFlix wants to build a **high-performance data warehouse** that can support:

- Real-time viewer engagement analytics
 - Top trending videos in each region
 - AI-powered recommendation systems
-

1. Recommended Architecture: Real-Time + Analytics Optimized System

To meet these needs, a **hybrid architecture** combining streaming + batch processing is proposed.

A) Real-Time Data Ingestion

Feature	Description
Event Types	play, pause, rewind, like, comment, search, share
Tools	Apache Kafka, AWS Kinesis
Purpose	Capture events instantly to power live dashboards and alerts

✓ Enables:

- ✓ Live streaming analytics (who is watching what, right now)
 - ✓ Trending videos detection
 - ✓ Real-time personalized recommendations
-

B) Storage Layer: Data Lake + Data Warehouse

Component	Purpose
Data Lake	Stores raw event logs, JSON clickstreams, device data
Data Warehouse	Stores cleaned, structured fact + dimension tables for analytics

Schema Type:

- **Star Schema** → Central fact table (events) + multiple dimension tables
- Optimized for fast aggregation and reporting

2. ETL / ELT Pipeline

Step	Description
Extract	Collect data from streaming events, user profiles, video catalog, device info, region metadata
Transform	Clean timestamps, standardize region names, calculate total watch time, remove duplicate events
Load	Insert into events_fact table and related dimension tables (user, video, region, device, date)

Sample Transformation Query:

```
SELECT user_id, video_id, SUM(duration_watched) AS total_watch_time,  
       COUNT(event_id) AS play_count  
FROM events_raw  
WHERE event_type = 'play'  
GROUP BY user_id, video_id;
```

3. Data Warehouse Schema Design

❖ Fact Table – events_fact:

Column	Description
event_id	Unique event record
user_id	Viewer reference
video_id	Video reference
device_id	Device used (mobile, TV, tablet)
region_id	Location (country, city, region)
date_id	Time reference
event_type	play, pause, like, search, etc.
event_time	Timestamp of action
duration_watched	Time watched per session
watch_percentage	% of video watched
like_flag	User liked or not
comment_id	Comment reference if any

Column	Description
subscription_type	Basic/Premium plan

❖ Dimension Tables:

Table	Key Details
user_dim	Name, email, age, gender, region, subscription type, preferred genres
video_dim	Title, genre, duration, language, cast, release date, content rating
device_dim	Device type (TV/mobile), OS, screen resolution, app version
region_dim	Region name, country, state, city, timezone, user population
date_dim	Date, day, month, quarter, year, is_weekend, is_holiday

4. Example Analytics Query – Top Trending Videos by Region

```

SELECT v.video_title, r.region_name, COUNT(e.event_id) AS view_count
FROM events_fact e
JOIN video_dim v ON e.video_id = v.video_id
JOIN region_dim r ON e.region_id = r.region_id
WHERE e.event_type = 'play'
    AND e.event_time >= CURRENT_TIMESTAMP - INTERVAL '24' HOUR
GROUP BY v.video_title, r.region_name
ORDER BY r.region_name, view_count DESC;

```

5. Business & AI Benefits

Benefit	Impact
Real-time Monitoring	Track user activity instantly across regions
Trending Content Detection	Recognize viral videos within minutes
Better User Engagement	Personalized recommendations using AI
Improved Retention	Suggest relevant shows, reduce churn
AI/ML Readiness	Enables models like "Next Watch Prediction", "Churn Prediction", etc.