

# CASE STUDY: TRANSFORMING CUSTOMER EXPERIENCE & OPERATIONS AT SWIGGY USING SNOWFLAKE CORTEX AI\_SQL

## 1 BUSINESS CONTEXT: WHY SWIGGY NEEDED AI\_SQL

Swiggy — India's largest food-delivery platform — handles millions of daily interactions between customers, delivery partners, and restaurants.

### **Business challenges:**

- Exploding unstructured data — customer chats, support calls, feedback, tweets, invoices.
- Manual SQL & reporting led to slow insights and reactive responses.
- Difficult to detect emerging issues (late delivery, cold food, app crashes).
- Need to democratize data access — allowing non-technical users to query with natural language.

Swiggy adopted Snowflake Cortex AI\_SQL to build an AI-powered unified analytics platform that converts raw text, voice, and social data into structured, actionable insights — all within Snowflake.

## 2 OBJECTIVE

To enable Swiggy's business, CX, and data teams to:

- Analyse feedback across all regions & channels conversationally.
  - Automatically classify and summarize complaints.
  - Measure customer sentiment in real time.
  - Transcribe call-center audio and extract key topics.
  - Translate and unify multilingual feedback (Hindi, Tamil, Telugu, Bengali).
  - Identify similar complaints for quicker resolution.
  - Generate AI-powered narratives for leadership reports.
-



### 3 SOLUTION ARCHITECTURE

#### Snowflake Components:

- Database: SWIGGY\_AI\_SQL\_DB
- Schema: CUSTOMER\_EXPERIENCE
- Stage: SWIGGY\_STAGE (CSV uploads + media)
- Role: ANALYST\_ROLE with SNOWFLAKE.CORTEX\_USER
- Warehouse: AI\_WH

#### Data Flow:

1. Raw feedback ingestion → CSV upload to stage → table load.
2. AI\_SQL transformations → sentiment, classification, summarization, translation, transcription, document parsing.
3. Aggregated analytics views → insights by region, channel, sentiment.
4. Narrative generation → auto reports via AI\_COMPLETE.

### 4 UNIFIED DATASET

👉 Download `swiggy_ai_sql_unified_dataset.csv`  
10,000 rows of synthetic but realistic Swiggy data.

#### Schema (aligned with Snowflake table):

Column	Description
interaction_id	Unique record ID
ticket_id / order_id	Customer & order identifiers
interaction_date	Date of interaction
customer_type	CONSUMER / DELIVERY_PARTNER / RESTAURANT
channel	APP / CALL_CENTER / WHATSAPP / EMAIL / TWITTER
region / state / city	Indian geographic context
restaurant_name / cuisine_type	Food source metadata
order_value / discount / net_value	Order metrics
feedback_language	en / hi / ta / te / bn / ml



Column	Description
feedback_text	Natural-language feedback
social_post_text	Optional tweet or post
complaint_flag	0/1 for issue
complaint_category_manual	Complaint type
sentiment_manual	Positive / Negative / Neutral
resolution_status	OPEN / RESOLVED / ESCALATED
call_audio_stage_path / invoice_pdf_stage_path	File paths for multimodal AI

## 5 TABLE CREATION SCRIPT (DDL)

```

USE ROLE ACCOUNTADMIN;
CREATE OR REPLACE DATABASE SWIGGY_AI_SQL_DB;
CREATE OR REPLACE SCHEMA CUSTOMER_EXPERIENCE;

USE DATABASE SWIGGY_AI_SQL_DB;
USE SCHEMA CUSTOMER_EXPERIENCE;

CREATE OR REPLACE FILE FORMAT FF_SWIGGY_CSV
  TYPE = CSV
  SKIP_HEADER = 1
  FIELD_DELIMITER = ','
  FIELD_OPTIONALLY_ENCLOSED_BY = '"'
  NULL_IF = ('', 'NULL');

CREATE OR REPLACE STAGE SWIGGY_STAGE FILE_FORMAT =
FF_SWIGGY_CSV;

CREATE OR REPLACE TABLE SWIGGY_INTERACTIONS (
  interaction_id NUMBER,
  ticket_id STRING,
  order_id STRING,
  interaction_date DATE,
  customer_id NUMBER,
  customer_name STRING,

```



```
customer_type STRING,  
channel STRING,  
region STRING,  
state STRING,  
city STRING,  
pin_code STRING,  
restaurant_name STRING,  
cuisine_type STRING,  
order_value NUMBER(10,2),  
discount_amount NUMBER(10,2),  
net_value NUMBER(10,2),  
feedback_language STRING,  
feedback_text STRING,  
social_post_text STRING,  
complaint_flag NUMBER(1),  
complaint_category_manual STRING,  
sentiment_manual STRING,  
resolution_status STRING,  
call_audio_stage_path STRING,  
invoice_pdf_stage_path STRING  
);
```

## 6 LOAD DATA INTO SNOWFLAKE

```
COPY INTO SWIGGY_INTERACTIONS  
FROM @SWIGGY_STAGE/swiggy_ai_sql_unified_dataset.csv  
FILE_FORMAT = (FORMAT_NAME = FF_SWIGGY_CSV)  
ON_ERROR = 'ABORT_STATEMENT';
```

## 7 GRANT ACCESS TO CORTEX AI

```
GRANT DATABASE ROLE SNOWFLAKE.CORTEX_USER TO ROLE ANALYST_ROLE;  
USE ROLE ANALYST_ROLE;  
USE DATABASE SWIGGY_AI_SQL_DB;  
USE SCHEMA CUSTOMER_EXPERIENCE;
```

---



## BUSINESS LOGIC & AI\_SQL ANALYTICS

### SENTIMENT ANALYSIS

```
SELECT
  interaction_id,
  region,
  feedback_channel,
  feedback_text,
  sentiment_manual,
  AI_SENTIMENT(feedback_text) AS sentiment_ai
FROM SWIGGY_INTERACTIONS
SAMPLE (0.1);
```

### COMPLAINT CLASSIFICATION

```
SELECT
  interaction_id,
  feedback_text,
  complaint_category_manual,
  AI_CLASSIFY(
    feedback_text,
    ['Late Delivery', 'Cold Food', 'Wrong Item', 'Refund
Delay', 'Rude Partner', 'Hygiene Issue', 'App Crash', 'General
Feedback']
  ) AS complaint_category_ai
FROM SWIGGY_INTERACTIONS
WHERE complaint_flag = 1
SAMPLE (0.2);
```

### FILTERING FOR CRITICAL ISSUES

```
SELECT
  interaction_id,
  region,
  feedback_channel,
```



```
feedback_text
FROM SWIGGY_INTERACTIONS
WHERE AI_FILTER(
    'Return TRUE if text indicates severe delivery issue, hygiene
risk, or refund delay.',
    feedback_text
) = TRUE;
```

## MONTHLY SUMMARY AGGREGATION

```
SELECT
    DATE_TRUNC('month', interaction_date) AS month,
    AI_SUMMARIZE_AGG(feedback_text) AS feedback_summary
FROM SWIGGY_INTERACTIONS
GROUP BY month
ORDER BY month;
```

## REGIONAL AGGREGATION WITH AI\_AGG

```
SELECT
    AI_AGG(
        feedback_text,
        'Summarize top 5 recurring issues in South region in bullet
points.'
    ) AS south_summary
FROM SWIGGY_INTERACTIONS
WHERE region = 'South' AND complaint_flag = 1;
```

## EMBEDDING + SIMILARITY SEARCH

```
CREATE OR REPLACE TABLE SWIGGY_COMPLAINT_EMBEDDINGS AS
SELECT
    interaction_id,
    complaint_category_manual,
    feedback_text,
    AI_EMBED(feedback_text) AS feedback_embedding
```



```

FROM SWIGGY_INTERACTIONS
WHERE complaint_flag = 1;

SET seed_interaction_id = 500;

WITH seed AS (
    SELECT feedback_text, feedback_embedding
    FROM SWIGGY_COMPLAINT_EMBEDDINGS
    WHERE interaction_id = $seed_interaction_id
),
scored AS (
    SELECT
        c.interaction_id,
        c.complaint_category_manual,
        c.feedback_text,
        AI_SIMILARITY(seed.feedback_embedding,
c.feedback_embedding) AS similarity_score
    FROM SWIGGY_COMPLAINT_EMBEDDINGS c
    CROSS JOIN seed
    WHERE c.interaction_id <> $seed_interaction_id
)
SELECT * FROM scored ORDER BY similarity_score DESC LIMIT 10;

```

## INFORMATION EXTRACTION

```

SELECT
    interaction_id,
    feedback_text,
    AI_EXTRACT(feedback_text,
        'Extract JSON: issue_type, delivery_delay_minutes,
mentions_refund, mentions_competitor.'
    ) AS structured_json
FROM SWIGGY_INTERACTIONS
WHERE complaint_flag = 1
SAMPLE (0.1);

```



## TRANSLATION

```
SELECT
  interaction_id,
  feedback_language,
  feedback_text,
  AI_TRANSLATE(feedback_text, 'auto', 'en') AS
translated_feedback
FROM SWIGGY_INTERACTIONS
WHERE feedback_language <> 'en'
SAMPLE (0.1);
```

## TRANSCRIBE CALL CENTER AUDIO

```
SELECT
  interaction_id,
  call_audio_stage_path,
  AI_TRANSCRIBE(TO_FILE('@swiggy_media_stage',
  SPLIT_PART(call_audio_stage_path, '/', -1))) AS transcript
FROM SWIGGY_INTERACTIONS
WHERE feedback_channel = 'CALL_CENTER' AND
call_audio_stage_path IS NOT NULL
SAMPLE (0.05);
```

## PARSE INVOICES

```
SELECT
  interaction_id,
  invoice_pdf_stage_path,
  AI_PARSE_DOCUMENT(TO_FILE('@swiggy_media_stage',
  SPLIT_PART(invoice_pdf_stage_path, '/', -1)), 'LAYOUT') AS
parsed_invoice
FROM SWIGGY_INTERACTIONS
SAMPLE (0.01);
```





## TOKEN ESTIMATION

```
SELECT
  AI_COUNT_TOKENS(
    'ai_complete',
    'llama3.3-70b',
    (SELECT AI_AGG(feedback_text, 'Combine South region
feedback into one narrative')
  FROM SWIGGY_INTERACTIONS WHERE region = 'South')
  ) AS token_estimate;
```



## NARRATIVE GENERATION (EXECUTIVE REPORT)

```
SELECT AI_COMPLETE(
  'openai-gpt-4.1',
  OBJECT_CONSTRUCT(
    'prompt',
    CONCAT(
      'You are Swiggy India CX head. Based on total complaints
(',
      (SELECT TO_VARCHAR(COUNT(*)) FROM SWIGGY_INTERACTIONS
WHERE complaint_flag=1),
      ') and negative sentiment share (',
      (SELECT TO_VARCHAR(ROUND(100.0 *
COUNT_IF(sentiment_manual='Negative') / COUNT(*), 2))
  FROM SWIGGY_INTERACTIONS),
      '%), write a 200-word summary of major issues, affected
regions, and improvement actions.'
    )
  )
) AS executive_summary;
```



## REGION SENTIMENT OVERVIEW

```
SELECT
  region,
  AI_SENTIMENT(
    AI_AGG(feedback_text, 'Summarize complaints for this region
```



```
in 3 sentences.')
) AS region_sentiment_summary
FROM SWIGGY_INTERACTIONS
GROUP BY region;
```

## 9 KEY KPIS & AI-DRIVEN INSIGHTS

KPI	Definition	AI Function Used
Customer Sentiment Index	% positive feedback	AI_SENTIMENT
Complaint Classification Accuracy	Match AI vs manual	AI_CLASSIFY
Monthly Summary	Concise trend by month	AI_SUMMARIZE_AGG
Similar Complaint Clusters	Related ticket groups	AI_EMBED + AI_SIMILARITY
Translation Coverage	% feedback normalized to English	AI_TRANSLATE
Voice Analytics	Call sentiment + transcriptions	AI_TRANSCRIBE, AI_SENTIMENT
Invoice Quality Extraction	Structured metrics from PDFs	AI_PARSE_DOCUMENT, AI_EXTRACT

## GOVERNANCE & SECURITY SUMMARY

- Data stays within Snowflake, not exposed to public APIs.
- All AI\_SQL calls governed by RBAC + masking policies.
- Analysts granted SNOWFLAKE.CORTEX\_USER.
- Model usage metered under Snowflake AI token billing.
- Supports auditability & lineage through query history.

## FINAL IMPACT SUMMARY

Impact Area	Before	After
Complaint Classification	Manual & inconsistent	AI_SQL auto-classified in real-time



Impact Area	Before	After
Sentiment Tracking	Only survey-based	Real-time sentiment via AI_SENTIMENT
Call Analytics	External tools	Transcribed + analyzed in Snowflake
Language Handling	English only	Multilingual via AI_TRANSLATE
Reporting Time	Days	Minutes via AI_COMPLETE narratives

## EXECUTIVE TAKEAWAY

Snowflake Cortex AI\_SQL transformed Swiggy's feedback ecosystem from reactive to intelligent.

Every interaction — text, tweet, call, invoice — became analyzable with simple SQL. Analysts now ask, “Why are deliveries late in Hyderabad?” and get AI-powered insights in seconds — securely, at scale, and without leaving Snowflake.