E-Commerce Funnel Analysis Report

Power BI dashboard summary, business recommendations, and SQL appendix

Prepared by: **Yash**
Tools: **Power BI**, **Pandas**, **Excel**, **Matplotlib**, **MySQL 8+**

Overall conversion	**14.66%**	**Visit users**	**116**	**Purchase users**	**17**

Executive summary

This report summarizes an e-commerce user journey from visit to purchase. Out of 116 visit users, 17 completed a purchase, resulting in an overall conversion of 14.66%. The largest leakage occurs close to purchase (checkout to purchase), indicating last-step friction (checkout UX, payment failures, or late cost surprises).

Funnel results

Stage counts and step performance calculated from distinct users at each stage:

Stage	**Users**	**Stage conversion** (vs previous)	**Drop-off** (vs previous)
visit	116	—	—
view_product	81	69.83%	30.17%
add_to_cart	58	71.60%	28.40%
checkout	32	55.17%	44.83%
purchase	17	53.12%	46.88%

Key insights

• Conversion declines steadily across the funnel, indicating friction increases as users move closer to purchase.

• The final step (checkout -> purchase) shows the most severe drop-off; checkout and payment reliability are the highest ROI areas to fix.

• Early-stage engagement is healthier than late-stage completion, so improvements should prioritize completion rate rather than top-of-funnel volume.

Business recommendations

Recommendation	What to do
Checkout simplification (P1)	Reduce form fields, support autofill, and keep validation errors inline and clear.
Cost transparency earlier (P1)	Show shipping/taxes/ETA before checkout to reduce last-minute abandonment.
Payment reliability (P1)	Improve success rate with retries, fallback routing, and clear failure messaging.
Trust signals (P2)	Make returns/support/security visible near payment without cluttering the flow.

Device section (execution guidance)

Use device segmentation to guide execution (keep the main funnel simple). Track overall conversion and checkout completion by device using separate charts, then prioritize fixes where mobile completion is weaker.

• Mobile-first checkout: larger tap targets, fewer fields, autofill, and avoid keyboard overlap.

• Performance hardening: optimize checkout load time on mobile networks; defer non-critical scripts.

• Payment UX: keep mobile-friendly methods prominent (UPI/wallets where relevant) and reduce redirect friction.

• QA matrix: validate end-to-end checkout on Android Chrome and iOS Safari to catch UI breaks and payment issues.

Dashboard sections

Key Insights + Recommendations: Narrative summary of what the funnel shows and what to improve.

KPI Cards: Headline metrics: overall conversion, visit users, and purchase users.

User Funnel Overview: Stage-by-stage user drop-off from visit to purchase.

Funnel Stage Performance: Table view of stage conversion and total users per stage.

Stage Conversion Rate (vs Visit): Trend view to spot the sharpest conversion decline.

Dashboard screenshot

Screenshot included for visual reference of the built report.

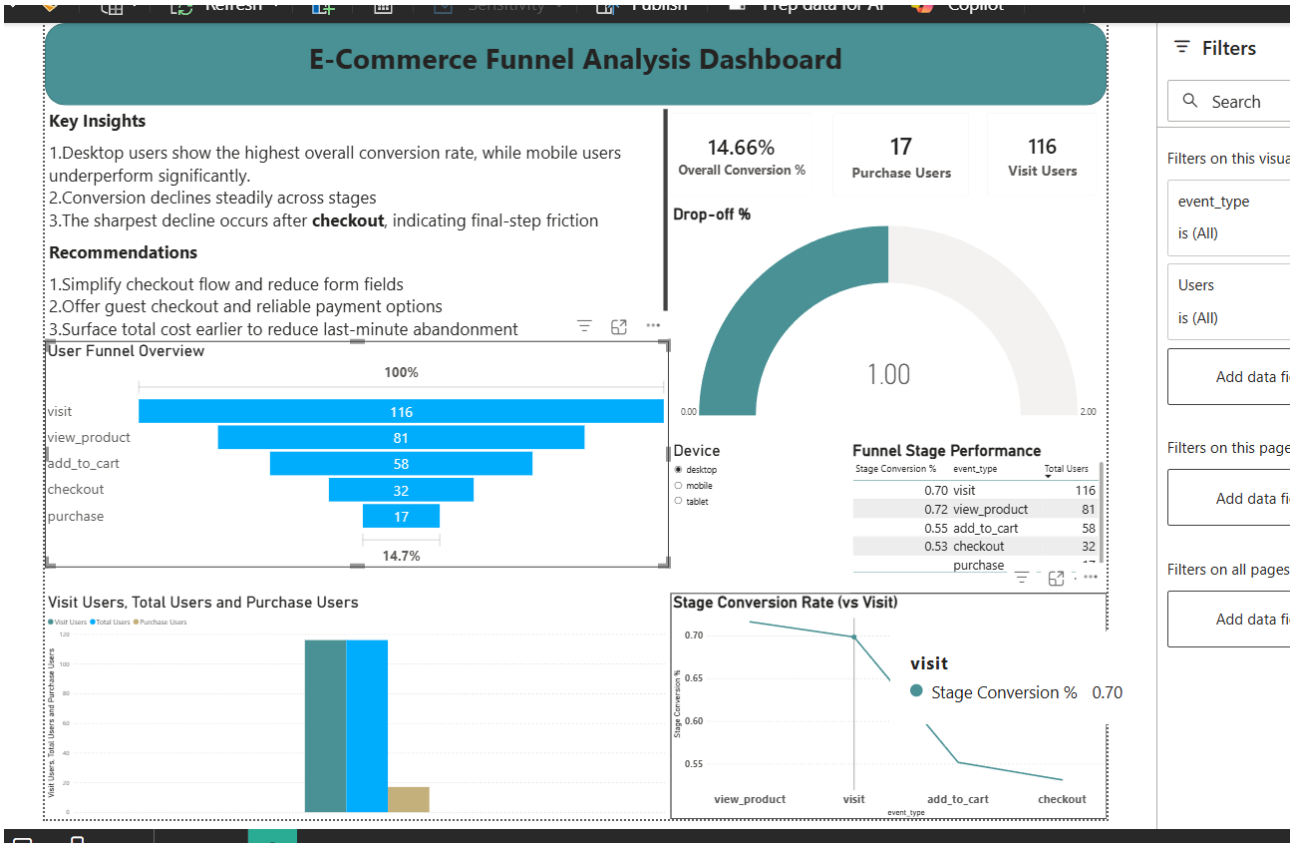

Figure: Power BI E-Commerce Funnel Analysis Dashboard (Canvas + Filters)

Appendix: SQL queries (with business question + answer)

Each query below is mapped to a business question and the expected output. The SQL is written for MySQL 8+.

SQL 1: Stage user counts

Business question: How many unique users reached each funnel stage?

Answer: Returns distinct user counts per stage (visit, view_product, add_to_cart, checkout, purchase). Used to build the funnel bars and the stage totals table.

```
SELECT
  event_type,
  COUNT(DISTINCT user_id) AS users
FROM funnel_events
WHERE event_type IN ('visit','view_product','add_to_cart','checkout','purchase')
GROUP BY event_type
ORDER BY FIELD(event_type,'visit','view_product','add_to_cart','checkout','purchase');
```

SQL 2: Step conversion and drop-off

Business question: What is the conversion rate and drop-off between consecutive stages?

Answer: Computes stage_conversion = current_stage_users / previous_stage_users and drop_off = 1 - stage_conversion. Helps identify the highest-friction step (largest drop-off).

```
WITH stage_counts AS (
  SELECT event_type, COUNT(DISTINCT user_id) AS users
  FROM funnel_events
  WHERE event_type IN ('visit','view_product','add_to_cart','checkout','purchase')
  GROUP BY event_type
),
ordered AS (
  SELECT
    event_type,
    users,
    LAG(users) OVER (
      ORDER BY FIELD(event_type,'visit','view_product','add_to_cart','checkout','purchase')
    ) AS prev_users
  FROM stage_counts
)
SELECT
  event_type,
  users,
  ROUND(users / NULLIF(prev_users,0), 4) AS stage_conversion,
  ROUND(1 - (users / NULLIF(prev_users,0)), 4) AS drop_off
FROM ordered
ORDER BY FIELD(event_type,'visit','view_product','add_to_cart','checkout','purchase');
```

SQL 3: Overall conversion

Business question: What percentage of visitors end up purchasing?

Answer: Calculates overall_conversion = purchase_users / visit_users using distinct users. This should match the headline KPI on the dashboard.

```
WITH counts AS (
  SELECT
    SUM(CASE WHEN event_type='visit' THEN 1 ELSE 0 END) AS visit_users,
    SUM(CASE WHEN event_type='purchase' THEN 1 ELSE 0 END) AS purchase_users
  FROM (
    SELECT event_type, user_id
    FROM funnel_events
    WHERE event_type IN ('visit','purchase')
    GROUP BY event_type, user_id
  ) x
)
SELECT
  purchase_users,
  visit_users,
  ROUND(purchase_users / NULLIF(visit_users,0), 4) AS overall_conversion
FROM counts;
```

SQL 4: Device-wise overall conversion (optional)

Business question: Which device segment converts better overall (desktop vs mobile vs tablet)?

Answer: Assigns one device per user (last seen by event_time), then computes visit_users, purchase_users, and overall conversion per device. Use this to prioritize device-specific fixes.

```
WITH last_device AS (
  SELECT user_id,
         SUBSTRING_INDEX(
           GROUP_CONCAT(device ORDER BY event_time DESC),
           ',', 1
         ) AS user_device
  FROM funnel_events
  GROUP BY user_id
),
user_flags AS (
  SELECT
    user_id,
    MAX(CASE WHEN event_type='visit' THEN 1 ELSE 0 END) AS did_visit,
    MAX(CASE WHEN event_type='purchase' THEN 1 ELSE 0 END) AS did_purchase
  FROM funnel_events
  GROUP BY user_id
)
SELECT
  d.user_device AS device,
  SUM(f.did_visit) AS visit_users,
  SUM(f.did_purchase) AS purchase_users,
  ROUND(SUM(f.did_purchase)/NULLIF(SUM(f.did_visit),0), 4) AS overall_conversion
FROM user_flags f
JOIN last_device d ON d.user_id = f.user_id
GROUP BY d.user_device
ORDER BY overall_conversion DESC;
```