# Amazon & Myntra Data Analyst Interview Experience.

# (Expeience= 2+ years)

# CTC= 29 LPA



🛒 **orders table:**

| order_id | user_id | product_id | quantity | order_date |
|---|---|---|---|---|
| 1 | 101 | P01 | 2 | 2025-04-10 10:00AM |
| 2 | 102 | P03 | 1 | 2025-04-12 04:00PM |
| 3 | 101 | P01 | 1 | 2025-04-13 03:00PM |
| 4 | 103 | P02 | 3 | 2025-04-13 02:00PM |
| 5 | 101 | P01 | 1 | 2025-04-14 01:00PM |

🧻 **products table:**

| product_id | product_name |
|---|---|
| P01 | iPhone 15 |
| P02 | Nike Shoes |
| P03 | Boat Earbuds |

---

## ✅ 1. Write a query to find the top 3 best-selling products in the last 30 days. (Focus on joins, GROUP BY, ORDER BY, and LIMIT)

🧠 **Logic:**

- Join orders and products tables
- Filter last 30 days
- Sum quantities per product
- Order by total sold descending
- Limit to top 3

✅ **SQL Query:**

```
SELECT
  p.product_id,
  p.product_name,
  SUM(o.quantity) AS total_sold
FROM
  orders o
JOIN
  products p ON o.product_id = p.product_id
WHERE
  o.order_date >= CURRENT_DATE - INTERVAL 30 DAY
GROUP BY
```

```
  p.product_id, p.product_name
ORDER BY
  total_sold DESC
LIMIT 3;
```

---

## ✅ 2. How would you write a query to get users who purchased more than 3 times in a week?

🧠 **Logic:**

- Group by user_id, YEAR(order_date), WEEK(order_date)
- Count number of purchases
- Filter users with count > 3

✅ **SQL Query:**

```
SELECT
  user_id,
  YEAR(order_date) AS purchase_year,
  WEEK(order_date) AS purchase_week,
  COUNT(*) AS num_purchases
FROM
  orders
GROUP BY
  user_id, YEAR(order_date), WEEK(order_date)
HAVING
  COUNT(*) > 3;
```

📌 **Note:** WEEK() is used to group weekly purchases. You can replace it with DATE_TRUNC('week', order_date) in PostgreSQL.

---

## ✅ 3 .Given a table with user purchases, identify returning customers vs. one-time buyers. give me detailed solutions of the above 3 questions with examples .

🧠 **Logic:**

- Count total distinct orders per user
- If count = 1 → One-time buyer
- If count > 1 → Returning customer

✅ **SQL Query:**

```
SELECT
  user_id,
  COUNT(DISTINCT order_id) AS total_orders,
  CASE
    WHEN COUNT(DISTINCT order_id) = 1 THEN 'One-time buyer'
    ELSE 'Returning customer'
  END AS customer_type
FROM
  orders
GROUP BY
  user_id;
```

📊 **Sample Output:**

| user_id | total_orders | customer_type |
|---|---|---|
| 101 | 3 | Returning customer |
| 102 | 1 | One-time buyer |
| 103 | 1 | One-time buyer |

---

## ✅ 4. Find the cumulative revenue per day for the last 7 days.

🧠 **Logic:**

- First, filter orders from the last 7 days.
- Calculate **daily revenue**: quantity * price
- Use **window function** (SUM() OVER) to compute **cumulative revenue** ordered by order_date.

---

✅ **Assumed Schema:**

Let's say we now have a price field in products.

🛒 orders:

| order_id | user_id | product_id | quantity | order_date |
|---|---|---|---|---|
| 1 | 101 | P01 | 2 | 2025-04-08 |
| 2 | 102 | P02 | 1 | 2025-04-09 |
| 3 | 101 | P01 | 1 | 2025-04-10 |

📦 products:

| product_id | product_name | price |
|---|---|---|
| P01 | iPhone | 50000 |
| P02 | Shoes | 3000 |

---

✅ **SQL Query:**

```
SELECT
  order_date,
  SUM(o.quantity * p.price) AS daily_revenue,
```

```
  SUM(SUM(o.quantity * p.price)) OVER (
    ORDER BY order_date
    ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
  ) AS cumulative_revenue
FROM
  orders o
JOIN
  products p ON o.product_id = p.product_id
WHERE
  o.order_date >= CURRENT_DATE - INTERVAL 7 DAY
GROUP BY
  order_date
ORDER BY
  order_date;
```

---

## ✅ 5.Explain how ROW_NUMBER() vs. RANK() works. When would you use each?

✅ 🧠 **Key Differences:**

| Feature | ROW_NUMBER() | RANK() |
|---|---|---|
| Uniqueness | Always unique | May have duplicates (ties) |
| Skips Rank? | ❌ No skipping | ✅ Skips after tie |
| Use Case | Unique index, pagination | Leaderboards, rankings |

---

🎯 **Example Table:**

| user_id | score |
|---|---|
| 101 | 90 |
| 102 | 90 |
| 103 | 80 |

✅ **Using ROW_NUMBER():**

```
SELECT
  user_id, score,
  ROW_NUMBER() OVER (ORDER BY score DESC) AS rn
FROM test_scores;
```

**Output:**

| user_id | score | rn |
|---|---|---|
| 101 | 90 | 1 |
| 102 | 90 | 2 |
| 103 | 80 | 3 |

---

✅ **Using RANK():**

```
SELECT
  user_id, score,
  RANK() OVER (ORDER BY score DESC) AS rk
FROM test_scores;
```

**Output:**

| user_id | score | rk |
|---|---|---|
| 101 | 90 | 1 |

| user_id | score | rk |
|---|---|---|
| 102 | 90 | 1 |
| 103 | 80 | 3 |

---

📝 **When to Use:**

- **ROW_NUMBER()**: For deduplication, pagination, top N per group.
- **RANK()**: For fair ranking where same values should share ranks.

---

## ✅ 6. How Would You Define and Calculate GMV (Gross Merchandise Value)?

🧠 **Definition:**

**GMV** is the total monetary value of merchandise sold over a period of time on a platform, **before discounts, returns, or fees**.

---

✅ **GMV Formula:**

GMV=∑(Price×Quantity)\text{GMV} = \sum (\text{Price} \times \text{Quantity})GMV=∑(Price×Quantity)

---

✅ **SQL Query:**

```
SELECT
  SUM(o.quantity * p.price) AS GMV
FROM
  orders o
JOIN
  products p ON o.product_id = p.product_id
```

```
WHERE
  o.order_date BETWEEN '2025-04-01' AND '2025-04-07';
```

---

🎯 **Use Cases:**

- Measures platform performance
- Input for investor reporting
- Helps track sales momentum

---

## ✅ 7. Explain How would you measure the success of a new feature on the app?

🧠 **Structured Approach:**

✅ **Step 1: Define the Feature Clearly**

- What is the feature?
- Who are the target users?
- What problem does it solve?

📌 *Example:* Let's say the feature is "One-Tap Buy" on product pages.

---

✅ **Step 2: Define Success Metrics (Primary + Secondary)**

- **Primary Metrics** (core success):
  - Conversion rate (visits → purchase)
  - Feature adoption rate
  - Daily active users of the feature
- **Secondary Metrics** (side effects):
  - Cart abandonment

- Time on app
- Bounce rate

---

✅ **Step 3: Use A/B Testing**

- Split users into **Control (no feature)** and **Treatment (with feature)** groups.
- Run the experiment for 2-4 weeks.
- Ensure randomness, sample size, and statistical significance.

---

✅ **Step 4: Post-Launch Monitoring**

- Analyze trends in key metrics.
- Segment by device, geography, traffic source, etc.
- Look for **regression in any existing KPIs**.

---

✅ **Tools: SQL, Power BI, Tableau, A/B testing platforms, Python (for stats)**

---

## ✅ 8. If Myntra's cart abandonment rate increases suddenly, how would you investigate?

🧠 **Structured Investigation:**

✅ **Step 1: Confirm the Spike**

- Is it a data issue?
- Is the increase consistent across:
  - Devices (iOS, Android, Web)?
  - Regions?
  - User cohorts?

---

### ✅ Step 2: Analyze Funnel Drop-off

Use event-level tracking (like AddToCart → Checkout → Payment → Purchase) to identify:

- Where the drop is happening
- Drop-off percentage at each stage

---

### ✅ Step 3: Check for Recent Changes

- Any new UI updates?
- Payment gateway issues?
- Server latency?
- Pricing bugs or errors?

---

### ✅ Step 4: Segment Affected Users

- Are new users facing more issues than old ones?
- Are discount-seekers abandoning more?

---

### ✅ Step 5: Conduct Qualitative Research

- Heatmaps / session replays
- App store complaints
- Survey users via push notifications or emails

---

### 📊 Metrics to Monitor:

- Cart abandonment rate = 1 - (Purchases / Add to Cart)
- Avg. time to checkout
- Drop-off by device/browser

---

## ✅ 9. How would you estimate the impact of delayed delivery on customer retention?

🧠 **Goal:**

Understand whether customers who experienced delayed deliveries are **less likely to return or make future purchases**.

---

✅ **Step-by-Step Approach:**

✅ **Step 1: Create User Cohorts**

- **Delayed delivery group**: Users whose orders were delayed by >X days
- **On-time delivery group**: Users whose orders were delivered on time

---

✅ **Step 2: Define Retention Metric**

- % of users who made a repeat purchase within 30/60/90 days
- Retention rate = Returning users / Total users in cohort

---

✅ **Step 3: Compare Retention Rates**

```
SELECT
  delivery_status,
  COUNT(DISTINCT user_id) AS total_users,
  COUNT(DISTINCT CASE WHEN repeat_purchase_date IS NOT NULL THEN user_id END) AS retained_users,
  ROUND(COUNT(DISTINCT CASE WHEN repeat_purchase_date IS NOT NULL THEN user_id END) * 100.0 / COUNT(DISTINCT user_id), 2) AS retention_rate
FROM user_order_data
GROUP BY delivery_status;
```

---

✅ **Step 4: Control for Confounders**

- Segment by region, product type, first-time user vs. repeat, etc.
- You could use **propensity score matching** or **logistic regression** in Python for advanced causal inference.

---

✅ **Step 5: Share Impact with Business**

"Delayed delivery reduced 30-day retention by 18%, especially among first-time users. Immediate action on logistics optimization could improve LTV and repeat rate."

---

## ✅ 10. What Metrics Would You Track for the Homepage of the Myntra App?

🧠 **Goal:**

The homepage is key to **user engagement**, **discovery**, and **funnel entry**—so metrics should reflect both **user behavior** and **business objectives**.

---

✅ **1. Traffic & Engagement Metrics**

- 🧍 Unique Visitors / Daily Active Users (DAU)
- ⏱️ Time Spent on Homepage
- 🖱️ Click-Through Rate (CTR) on banners, categories, carousels
- 🔄 Scroll Depth (how far users scroll)

---

✅ **2. Behavioral Metrics**

- 🛒 Add-to-Cart Rate from homepage clicks
- 🔍 Search Rate (users clicking into search from homepage)
- 📦 Product Click Rate from homepage recommendations

---

✅ **3. Conversion Metrics**

- 🛍️ Conversion Rate (Homepage visit → Purchase)
- Funnel Drop-off Rate (Homepage → Product Page → Checkout)
- Revenue per visitor/session

---

✅ **4. Tech/UX Metrics**

- ⏳ Page Load Time (esp. for personalized feeds)
- 🔁 Bounce Rate from homepage
- ⚠️ Error rate / crashes (esp. if images fail to load)

---

✅ **5. Personalization Metrics (if personalized homepage)**

- Hit rate of recommendation engine
- Engagement with personalized modules vs. generic ones

---

## ✅ 11. How Would You Design an A/B Test for a New Product Sorting Algorithm?

Let's say the team wants to test a new **"Relevance + Popularity" sorting** algorithm vs. the default "Popularity-based" one.

---

✅ **Step-by-Step A/B Test Plan:**

✅ **Step 1: Define Objective**

- Primary: Increase conversion rate or average order value
- Secondary: Improve CTR on top-ranked products, reduce bounce rate

---

✅ **Step 2: Define Control and Treatment**

- **Control Group**: Sees current sorting algorithm (Popularity-based)
- **Treatment Group**: Sees new sorting algorithm (Relevance + Popularity)

---

✅ **Step 3: Random Assignment**

- Randomly split users (50/50 or 80/20)
- Use **user-level bucketing**, not session-level (to avoid cross-contamination)

---

✅ **Step 4: Choose Metrics**

- Conversion Rate (Product View → Purchase)
- Average Time on Product Listing Page
- Product Clickthrough Rate (CTR)
- Add-to-Cart Rate
- Revenue per session

---

✅ **Step 5: Monitor & Analyze**

- Ensure large enough **sample size** for statistical significance (use t-test or Z-test)
- Use visualization tools (Power BI / Tableau / Python) to compare trends

---

✅ **Step 6: Post-Test Checks**

- Check for p-value < 0.05 and confidence intervals
- Analyze lift in key metrics
- Segment results by user type, platform, etc.

---

## ✅ 12. A Test Shows No Significant Difference in Conversion Rates. What Could Be the Reasons?

🧠 **Possible Reasons & Root Cause Analysis:**

---

✅ **1. Insufficient Sample Size / Power**

- Too few users → no statistical power
- High variance in user behavior → wide confidence intervals

📌 Fix: Calculate sample size before running the test

---

✅ **2. Short Duration**

- Ran test only for a few days → did not capture full user cycle

📌 Fix: Run for at least 2-3 weeks, include weekdays + weekends

---

✅ **3. Wrong Primary Metric**

- The new feature affects engagement, not conversion
- Misalignment between what's measured and what's impacted

📌 Fix: Choose better-aligned KPIs (e.g., CTR instead of conversion)

---

✅ **4. Implementation Bugs**

- Feature not properly applied to treatment group
- Sorting algorithm not deployed or malfunctioning

📌 Fix: QA logs, confirm via internal testing

---

✅ **5. External Noise**

- Campaigns, outages, price changes during test window

- A major sale skewing buying behavior

📌 Fix: Use calendar normalization and segment-wise checks

---

✅ **6. Feature Just Didn't Work**

- Sometimes, the idea just doesn't make a meaningful difference.

📌 Fix: Accept the null hypothesis and move to next iteration.

## ✅ 13. Explain P-value and Confidence Interval in the Context of Experiments

---

🎯 **P-value (Probability Value)**

📌 **Definition:**

The **p-value** tells you the **probability of observing the experiment result (or more extreme)**, assuming the **null hypothesis is true**.

🧠 **Context in A/B Testing:**

- **Null Hypothesis (H0)**: There is **no difference** between control and treatment.
- If you get a **p-value < 0.05**, you **reject the null hypothesis** → the difference is **statistically significant**.

✅ **Example:**

You're testing a new product sorting algorithm.

- **Conversion rate** (Control): 5.2%
- **Conversion rate** (Treatment): 5.6%
- **p-value = 0.03**

Interpretation: There is only a **3% chance** of seeing this 0.4% uplift in conversion **if the new algorithm had no real effect**.
So, the uplift is **statistically significant**.

---

📊 **Confidence Interval (CI)**

📌 **Definition:**

A **95% confidence interval** means we are 95% confident that the **true difference in metrics** (like conversion) lies within this range.

✅ **Example:**

- Uplift in conversion = **0.4%**
- 95% CI = **[0.1%, 0.7%]**

Interpretation: We are 95% confident that the **true conversion lift** is somewhere between **0.1% and 0.7%**.

✅ If the **entire CI > 0**, it indicates **statistical significance**.

---

🔁 **Key Difference:**

| Term | Tells You... |
|---|---|
| **P-value** | If a result is statistically significant |
| **Confidence Interval** | The **range** of possible values for the effect size |

---

## ✅ 14. How Would You Calculate Sample Size for an A/B Test with 80% Power and 95% Confidence?

---

🧠 **Key Inputs Required:**

1. **Baseline Conversion Rate (p1)** – e.g., 5%
2. **Minimum Detectable Effect (MDE)** – e.g., 0.5% uplift
3. **Confidence Level (α)** – typically 95% → Z-score = **1.96**
4. **Statistical Power (1 - β)** – typically 80% → Z-score = **0.84**

✅ **Sample Size Formula (for each group):**

$$n = \left[ \frac{(Z_{\alpha/2} + Z_{\beta})^2 \times (p_1(1-p_1) + p_2(1-p_2))}{(p_2 - p_1)^2} \right]$$

🔢 **Example:**

- Baseline rate (p1) = 5% = 0.05
- MDE (p2) = 5.5% = 0.055
- So, difference = 0.005
- Z(95%) = 1.96, Z(80%) = 0.84

Now,

$$n = \frac{(1.96 + 0.84)^2 \times [0.05(0.95) + 0.055(0.945)]}{(0.005)^2}$$

$$n = \frac{7.84 \times (0.0475 + 0.051975)}{0.000025} \approx \frac{7.84 \times 0.0995}{0.000025}$$

$$n \approx \frac{0.77968}{0.000025} \approx 31,187 \text{ users per group}$$

✅ **Tools You Can Use:**

- Python: statsmodels.stats.power.tt_ind_solve_power
- Online calculator: Evan Miller A/B test calculator
- Excel: Use built-in NORM.S.INV() for Z-scores

---

🔁 **Final Summary:**

| Term | Meaning |
|---|---|
| **P-value** | Probability that the observed effect is due to random chance |
| **Confidence Interval** | Likely range of the true effect (e.g., uplift in conversions) |
| **Sample Size** | Number of users needed to detect an effect with confidence |

---

## ✅ 15. What Are Some Pitfalls to Avoid When Analyzing an A/B Test?

Here are the most **important A/B test pitfalls** you must **avoid (with real-world impact examples):**

---

### ❌ 1. Stopping the Test Too Early

- You see a good uplift in Day 2 and rush to call it a win.
- **Problem**: You may be acting on a statistical **fluke**.

✅ **Fix**: Pre-define the **test duration and sample size**. Stick to it unless something breaks.

---

### ❌ 2. Peeking Too Often (P-Hacking)

- Checking p-values multiple times during the test.
- **Problem**: Increases the chance of a **false positive** (Type I error).

✅ **Fix**: Use sequential testing techniques or set fixed checkpoints (e.g., Day 7, Day 14).

---

### ❌ 3. Wrong Metric Choice

- Measuring CTR when the goal is Purchase Rate.
- **Problem**: Misleads the analysis and decision-making.

✅ **Fix**: Align your **primary metric** with the **business goal** (e.g., Conversion Rate, AOV, Bounce Rate).

---

### ❌ 4. Ignoring Segment Differences

- Average treatment effect shows no uplift → you reject it.
- **Problem**: It may be working **for high-intent users**, but not for low-intent ones.

✅ **Fix**: Always analyze by **platform, user segment, geography**, etc.

---

❌ **5. Not Randomizing Properly**

- Assigning users instead of sessions randomly, or vice versa.
- **Problem**: Results get biased (same user may enter both groups).

✅ **Fix**: Use **user-level randomization**, especially in login-based apps.

---

❌ **6. External Influences Not Accounted For**

- Running a test during **Diwali sale**, but not accounting for that spike.
- **Problem**: Test gets **confounded** by other variables.

✅ **Fix**: Normalize results across time, or avoid running tests during abnormal periods.

---

❌ **7. Data Quality Issues**

- Tracking bug means treatment group's revenue was never logged.
- **Problem**: You'll conclude treatment failed—**false negative**.

✅ **Fix**: Always do a **pre-test QA check** to validate events are tracked properly.

---

❌ **8. Not Considering Business Cost**

- Test shows 0.5% uplift in conversions, but slows down app by 2 seconds.
- **Problem**: Business loss outweighs the gain.

✅ **Fix**: Include **performance, UX, and financial impact** in the final analysis.

---

❌ **9. Looking Only at Statistical Significance**

- You get a p-value of 0.04 and celebrate.
- **Problem**: The **effect size** might still be too small to matter.

✅ **Fix**: Always check the **confidence interval** and **practical significance** (Is the lift meaningful?).

---

❌ **10. Running Too Many Simultaneous Tests**

- Multiple tests on same page at once.
- **Problem**: **Cross-test interference** → muddy results.

✅ **Fix**: Use **mutually exclusive user buckets** or coordinate test layers.

---

🧠 **Pro Tip for Interviews:**

Wrap it up by saying something like:

“At every stage, I ensure alignment with business goals, proper tracking, and disciplined statistical rigor to avoid misleading conclusions.”