Data Analyst / Data Engineer Practical Assessment

Overview

This assessment is designed to evaluate your **analytical thinking**, **data reasoning**, **storyboarding** and **problem-solving ability** — across both **logical puzzles** and **real-world data scenarios**.

You will find a mix of questions:

- Some test your ability to **reason and structure logic clearly**.
- Others simulate **practical data challenges**, where you will **clean, transform, and visualize messy data** to produce actionable insights.

You are **not required to solve every question** — the goal is to showcase **depth and clarity**, not volume. A candidate who solves **one question completely, with clean documentation and a traceable thought process**, will be rated **higher** than someone who submits half-finished or disorganized answers to all.

Submission Expectations

- You may attempt **as many questions as you wish**, in **any order**.
- Focus on **quality over quantity**. A **complete, neatly documented, and well-reasoned solution** is far more valuable than multiple incomplete ones.
- Every question should have its own **folder** in your submission with all relevant files and supporting documents.

Use of Tools and Resources

- You may use **any tools** such as **Excel, Python, R, SQL, Power BI, Tableau, or similar**.
- You are **encouraged** to use **online resources or LLMs (like ChatGPT, Gemini, or others)** — just like in real-world problem solving.
- However, you must **understand what you are doing**. Be ready to **explain every step** in a walkthrough if asked.
- Use AI as a **thinking partner**, not as a copy-paste solution.

Submission Format

- Submit **one ZIP file** containing your entire submission.
- Inside the ZIP, create a **separate folder for each question attempted**, named clearly (e.g., Q1_25_Horses, Q2_Optimal_Hiring, Q3_Market_Analysis, etc.).
- Each folder should contain:
    1. **Working files** – Scripts, spreadsheets, notebooks, or dashboards.
    2. **Short documentation** – Explaining what you did, the logic you followed, and any assumptions made.
    3. **Supporting materials** – Screenshots, visuals, or external data sources (if used).
- Use **meaningful filenames** and **organized structure** — submissions that are easy to navigate score higher.

Suggested Time and Effort

- Total time: **3 hours**
- Spend time understanding the **intent behind each question**.
- Look for **hidden logic or trade-offs** — several questions are designed to test reasoning beyond surface-level reading.

Guidelines and Best Practices

Area	**Guideline**
Documentation	Explain your reasoning briefly but clearly. Assume your evaluator has no context and must understand your process just by reading your files.
Data Work	If the task involves data, document how you cleaned, merged, or derived columns. Mention every key assumption.
Logic Puzzles	For non-technical puzzles, make your reasoning visual or step-by-step — use PowerPoint, diagrams, or flowcharts if it helps.
Visualization	Focus on clear storytelling — your charts should make the insights obvious, not cluttered.
Code Quality	Use comments and keep the structure clean; explain what each section does.
File Structure	Keep folders and files neatly named. Consistency and readability count as professionalism.
Citation	If you use public datasets or online content, include the dataset name or URL for transparency.

Evaluation Criteria

Criteria	**Description**
Analytical Thinking	Logical structuring, clarity of reasoning, and ability to handle ambiguity.
Data Preparation Quality	Cleaning, joining, and handling of real-world messy data.
Communication & Clarity	How easily someone else can follow your reasoning and results.
Creativity & Initiative	Smart use of tools, metrics, or additional data enrichment.
Professionalism	Quality of documentation, organization, and overall presentation.

Final Submission

Submit one ZIP file to **careers@dhira.ai**

Tips from the Evaluators

- Go **slow and deep**, not fast and shallow.
- Think before you build — **understand the “why”** behind each step.
- **Label everything clearly**. Your file organization says as much about your discipline as your code does.
- **If you get stuck**, that’s okay — document your reasoning and what you tried. Thoughtful partial progress is better than guesswork.
- Most importantly, **enjoy the process** — treat it like a real consulting assignment, not an exam.

1. The 25 Horses Challenge

You are given **25 horses** and only **5 racetracks**, meaning that at most **5 horses can race at a time**. Your goal is to **identify the 5 fastest horses** — in exact order — **without using a stopwatch**, only based on relative race results.

You need to determine:

1. The **minimum number of races** required to guarantee the 5 fastest horses are correctly identified.
2. A **clear storyboard** that explains how you would conduct each race and decide which horses advance or get eliminated.

Instructions

- Assume all horses run consistently with no variation in performance between races.
- You can only compare results within each race (no timing devices).
- Your final submission must include both:
  a. The **number of races required**, with logical reasoning.
  b. A **pictorial storyboard** that visually explains your approach step by step.

Format

- The storyboard should be self-explanatory — anyone reading it should be able to follow your reasoning without assistance.
- You may use any visual or design tool you prefer, such as **PowerPoint, Paint, Canva, Figma, Miro, Adobe tools or even scanned hand drawn images**.
- Include short **text captions** or labels for each race to describe your decisions and eliminations.
- Use color-coding or clear flow arrows to make the logic easy to understand.

Evaluation Criteria

Criteria	**Description**
Logic and Accuracy	Is the reasoning sound and does it reach the correct minimum number of races?
Clarity of Storyboard	Is the sequence of races and eliminations easy to follow?
Creativity	Is the approach or visualization unique or thoughtfully presented?
Communication	Can a reader understand the full reasoning without needing further explanation?

2. The Executive Dashboard Challenge

Your company is an online retail startup that has been growing quickly, but two major problems have started to show up — **profits are inconsistent** and **customers are unhappy with deliveries**.

As a **Data Analyst**, your job is to explore the company's past data and find **which markets or regions show the best potential for future growth** while also understanding **where the biggest operational risks lie** (for example, delayed deliveries, high refund rates, etc.).

Your Task

You are expected to:

1. **Clean and prepare the raw data** (multiple CSVs such as Orders, Shipments, Products, Customers, Reviews, and Geo Data).
   - Fix missing values, inconsistent formats, wrong joins, etc.
   - You can use any tool or programming language you're comfortable with.
   - Keep a record of all changes made and your reasoning.
2. **Build a full-fledged Executive Dashboard** in **Power BI, Tableau, or any visualization tool** of your choice.
   - The dashboard should help the **CEO quickly understand** which markets or regions are performing well and which are risky.
   - It should include at least **one "success" metric** (e.g., profit, retention, growth) and **one "risk" metric** (e.g., delivery delays, refund rate, negative reviews).
   - Focus on making the dashboard **aesthetically pleasing**, **easy to read**, and **insightful at a glance**.
   - If you can't share the actual dashboard file, attach a **screenshot** of it.
3. **Write a short narrative (max 250 words)** addressed to the CEO.
   - Summarize your **Top 3 recommended markets** for investment or focus.
   - Explain what the dashboard reveals and any trade-offs between success and risk.

What to Submit

Your final submission should include:

1. **Dashboard** – a file or screenshots of your Power BI / Tableau dashboard.
2. **CEO Summary** – a short write-up (max 250 words) explaining key findings.
3. **Data Cleaning Document** – a separate document that lists:
   - What steps you took to clean, merge, or transform data.
   - Any scripts, formulas, or code used (Python, SQL, Excel Power Query, etc.).
   - Any assumptions or fixes applied.

(Optional) You may also include **external public data** (like population, delivery infrastructure, or weather data) to improve your insights. Clearly mention your source.

Evaluation Criteria

Criteria	Description
Data Preparation (25%)	Were data cleaning steps logical, well-documented, and reproducible?
Dashboard Clarity (30%)	Is the dashboard well-designed, interactive, and CEO-friendly?
Insight & Reasoning (25%)	Are the insights meaningful and supported by data?
Narrative & Communication (10%)	Is the CEO summary concise and impactful?
Creativity & Initiative (10%)	Any unique visualizations, thoughtful use of external data, or clever analytical angle.

3. The Biased Interview Strategy

A company interviews **7 candidates** one at a time in the fixed order:

A, B, C, D, E, F, G

Each candidate has a unique hidden ability score from **1 (worst)** to **7 (best)**. After each interview the hiring manager learns **only the relative ranking** of the candidates seen so far (for example: “C is better than A and B so far”), but **not** absolute scores. After each interview you must decide **immediately** to **hire** that candidate or **reject** and move on — once rejected, a candidate cannot be recalled. The goal is to **maximize the probability** of hiring the overall best candidate (score 7).

Special condition: the first five candidates **A–E** arrive in **strictly increasing quality order** (i.e., A < B < C < D < E). The last two candidates (**F and G**) arrive in a **random order** relative to each other and relative to A–E. All rank assignments consistent with this rule are equally likely.

Deliverable (what to write as your answer)

In **no more than 150 words**, do **both** of the following:

1. **State clearly which single candidate you will hire (or the simple rule you will follow)** — e.g., “Hire candidate X on arrival” or “Reject until Y, then hire next best-so-far.”

2. **Give a short, rigorous justification** (concise proof or counting argument) explaining **why that choice maximizes the probability** of selecting the overall best candidate. Your justification should be logically complete (no hand-wavy phrases) and must rely only on the information given.

4. Finding the Best Launch Locations in Your State

SnackFast is a fast-growing **Quick Service Restaurant (QSR)** brand preparing to open its first **delivery-only kitchens** in India. You are the **Data Analyst** in charge of recommending the **Top 3 Pincodes in your state** that show the **best growth potential** with **manageable operational risks**.

Your task is to collect, clean, analyze, and visualize publicly available data to help SnackFast's leadership decide **where to launch first**.

Your Mission

This project simulates an end-to-end analytics task — from **data gathering** to **final executive insights**.

Phase 1: Data Sourcing and Preparation

You'll start by collecting two public datasets related to India and preparing them for analysis.

1. **Data Layer 1 – Market Demand**
   Find a dataset that includes **Pincode**, **District**, **State**, and a measure of **Population** or **Number of Households**.

2. **Data Layer 2 – Competition Landscape**
   Find a dataset that lists **Points of Interest (POIs)** like restaurants, retail stores, or similar service outlets, with **Pincode information**.

Then:

- Filter both datasets to include **only for your state**.
- Clean and standardize the **Pincode** field.
- Merge both layers into a **single master dataset** using Pincode as the key.

Phase 2: Business Logic and Metric Creation

Now, apply logic to create your own meaningful business metrics.

1. **Competitive Density Index (CDI):**

$$CDI = \frac{\text{Number of Competitors (Layer 2)}}{\text{Population (Layer 1)}} \times 10{,}000$$

This indicates how crowded each market is — lower CDI means less competition per capita.

2. **Logistics Risk Score:**
   Create a column Logistics_Risk_Score using a simple rule of your choice.
   For example, assign higher risk to pincodes whose first two digits indicate remote or difficult-to-serve regions.

3. **Overall Viability Tag:**
   Combine CDI and Risk Score to classify each pincode as **High**, **Medium**, or **Low Viability** for store launch.

Phase 3: Dashboard and Recommendation

1. **Executive Summary (max 250 words):**
   Write a short note to the CEO that:
   - Explains your **Top 3 recommended pincodes** for launch.
   - Describes how you balanced demand, competition, and risk.
   - States any **assumptions** made during cleaning or calculations.
2. **Dashboard (max 3 visuals):**
   Create an **Executive Dashboard** in Power BI, Tableau, or any visualization tool.
   It should clearly show:
   - Demand potential (Population or Households)
   - Competition (CDI or similar metric)
   - Logistics Risk (your custom score)

If you can’t submit the dashboard file, attach **screenshots** or a **PDF export**.

Bonus Points (Optional)

To stand out, you can include:

- **A third dataset** — e.g., income levels, delivery density, or traffic data.
- **A simple model or index** that predicts “High Viability” areas.
- **An original custom metric** (like a “Profit Opportunity Index”) and explain its logic.

Final Submission

Submit a single folder or ZIP file containing:

1. **Dashboard** – Power BI / Tableau / Screenshot / PDF.
2. **CEO Summary** – Short business write-up (max 250 words).
3. **Data Prep Document** – Explaining how you:
   - Cleaned and merged the data
   - Created your metrics
   - Applied assumptions or logic

Evaluation Criteria

Criteria	Description
Data Preparation (25%)	Data sourcing, cleaning, and joining steps are clear and logical.
Metric Design (20%)	CDI, Risk, and Viability calculations make business sense.
Dashboard (25%)	Visually clear, aesthetic, and insight-driven.
Executive Summary (20%)	Explains reasoning and trade-offs effectively.
Creativity (10%)	Extra features, third dataset, or custom logic add value.

5. Golden Hospital Discharge Record Design

You are a **Data Engineer** at a hospital tasked with transforming fragmented clinical data from its **Electronic Health Record (EHR)** system into a single **Gold-Standard Analytical Dataset**. The source data is highly unstructured, inconsistent, and messy.

Your goal is to create a **Golden Discharge Record** table where **each row represents one unique discharge event (IPID)**, and all available structured information is cleanly represented as columns.

The Data Provided

You have been given three raw CSV files containing disconnected operational data:

- **discharge_master.csv** – Admission details, patient IDs, dates, and financial summaries.
- **discharge_details.csv** – Itemized breakdowns of charges, amounts, and procedures.
- **clinical_findings.csv** – Free-text medical notes and treatment descriptions.

Assessment Tasks

1. **Data Preparation and Transformation**
   - Understand and clean each dataset to handle missing values, duplicates, and inconsistent identifiers.
   - Merge the files logically using **IPID** as the central key.
   - Extract key features from clinical notes (e.g., keywords, diagnoses, procedures) to enrich the dataset.
2. **Final Deliverables**
   - A single **analytical table or spreadsheet** where each row represents one **discharge event (IPID)** and all relevant details are cleanly included.
   - A short **document** (max one page) describing the major cleaning and transformation steps you performed.
   - An optional **code script** (SQL, Python, or any tool) if you used automation for cleaning or merging.

Format

- You may use **Excel, SQL, Python (Pandas), or any ETL tool** you're comfortable with.
- Keep your column names meaningful and standardized (e.g., Total_Amount, Diagnosis_Category, Stay_Duration_Days).

Evaluation Criteria

Criteria	Description
Logic and Accuracy	Are the joins, cleaning, and transformations logically sound?
Completeness	Does the final table capture maximum usable information from all files?
Documentation Clarity	Are the cleaning and transformation steps easy to understand?
Data Quality	Are duplicates, missing data, and inconsistencies properly handled?

6. The Fast Learner Challenge

Overview

You are a new member of a growing data team that includes both experienced professionals and fresh graduates. The team is exploring new technologies to modernize its data platform, but not everyone is familiar with them yet.

Your manager has asked you to **learn one such technology from scratch** and create a **document that helps everyone in your team — technical or non-technical — understand it clearly.**

The goal is not to copy documentation but to **explain the concept, architecture, and practical use-cases** in a way that **anyone in your organization can understand and apply.** Use simple language, real-world examples, and diagrams if necessary.

You may choose any one topic you are **not already familiar with**, such as:
dbt, Apache Airflow, Apache Kafka, Delta Lake, Apache Superset, Power BI, Snowflake, Dataform, DuckDB, Dagster, Great Expectations, Airbyte, Fivetran, Prefect, Metabase, or Redshift.

Your Task

Create a **document** that introduces your chosen technology, explains how it works conceptually, and demonstrates how it can be useful for your organization.

You have complete freedom to decide the structure — focus on clarity, practical examples, and logical flow rather than technical depth.

Deliverable

Submit your final **Word or PDF document.** You may include visuals, charts, or diagrams if they help make the explanation easier to follow.

Evaluation Criteria

Criteria	**Description**
Understanding	Shows that you genuinely learned and grasped the topic.
Structure and Clarity	The document is organized, readable, and flows logically.
Originality	Written in your own words, not copied or rephrased from online sources.
Communication	Uses clear, relatable explanations and examples for all audiences.
Judgment	Demonstrates what you chose to emphasize or simplify — not everything needs to be included, only what helps understanding.

7. The Conflicting Database Mandates

Scenario: The Conflicting Database Mandates

You are a Junior Data Engineer tasked with designing the structure for the company's new analytical database. You've received two conflicting requests regarding the core sales data:

1. **Finance Team Mandate (Accuracy/Audit):** We need tables designed in a **Normalized format** (like a Third Normal Form - 3NF). We must avoid all data duplication so that every financial report is perfectly accurate and auditable.
2. **Marketing Team Mandate (Speed/Flexibility):** We need data in a **Denormalized format** (one large, flat table) so we can run complex queries for quick customer segmentation and promotional targeting without having to perform slow, complex joins.

You must design **one core analytical data model** that attempts to satisfy both requests by applying the most effective design trade-offs.

1. Define Core Entities and Ambiguity

Based on your knowledge of e-commerce, define the **4 core entities** (tables) required.

- **Ambiguity Test:** In the source transaction log, the primary product identifier is sometimes stored as **Product_ID_Master** and sometimes as **Product_Category_Code**. You must choose **one** key to manage your central Product Dimension and justify why the *other* key cannot serve as the primary lookup.

2. Design the Hybrid Model

Design an analytical model consisting of **two distinct layers** that resolves the conflict.

- **Layer 1: The Audit Layer (Normalized):** Design this layer to meet the Finance Team's audit needs. Use a simple Fact and Dimension approach.
- **Layer 2: The Speed Layer (Denormalized):** Design this second table to meet the Marketing Team's speed needs. This single table should duplicate key descriptive fields from Layer 1.

3. The Judgment Test (Explain the Trade-Offs)

This section tests your ability to apply business logic to technical design.

- **Conflict A: The Freshness Prioritization:** You can only update one layer instantly upon a sale. Explain which layer (Audit or Speed) you would prioritize for instant updates and **quantify** the compromise: How many hours/minutes of data latency will the *other* team experience, and **why is that tolerable** based on their business function?
- **Conflict B: The Name Change Problem (Intricate Details):** The Marketing team wants to rename a product category from **"Shoes"** to **"Footwear"** starting tomorrow.
    - Explain the **exact two columns** you must add to your descriptive tables to manage this change and satisfy *both* the Finance team (historical audit) and the Marketing team (current segmenting).
    - What technique does this represent, and why is it superior to simply updating the name in place?

4. Final Deliverable

Submit a **simple Entity-Relationship (ER) Diagram** (drawn in any basic tool, text, or spreadsheet layout) showing your two layers and how they link. Accompany this with a written **Trade-Off Justification** document that clearly answers all conflicts and ambiguities above.

All the very best!
Approach each question like a problem worth solving — not an exam to pass.
Think clearly, stay curious, and have fun with it.
— *Warm wishes from Team Dhira.ai*