🚀 Antigravity ETL v5.0 — Technical Documentation

This guide explains the architecture, performance logic, and data protection mechanisms behind the high-speed Google Drive ingestion engine.

⚠️ Critical Operational Requirement

The backend server must be running at all times for the synchronization to work. The ETL engine is part of the backend process. If you close the backend terminal, the "Smart Sync" will stop. To ensure continuous scanning, keep the following command running:

```
python app.py --runserver
```

---

1. System Architecture

The ETL (Extract, Transform, Load) engine is built for **Massive Scale** and **Zero Duplication**. It runs as a persistent background thread alongside the Flask server.

Core Components:

- **Producer (Scanner)**: Monitors Google Drive for any changes across 5,000+ folders.
- **Consumer (Workers)**: 24-32 parallel threads that download, normalize, and save data.
- **Reactive Engine**: Uses the **Google Drive Changes API** to detect movements without rescanning the entire drive.

---

2. How "Reactive Sync" Works

Instead of scanning every folder every minute (which is slow and wastes API calls), the system uses a **Motion Sensor** approach:

1. **Change Alerts**: The engine asks Google Drive: "*Give me everything that changed since Token [X].*"
2. **Immediate Response**: If you add a file, move a folder, or rename a CSV, Google sends an alert.
3. **Targeted Scan**: The engine **only** navigates to the specific folders or files that were modified.
4. **Idle Mode**: If no changes are detected, the system logs: ⚡ `System Idle...` and rests to save CPU and API quota.

---

3. The "Double-Lock" Duplicate Protection

The system ensures that your **1.54 Million+ records** are perfectly unique using two layers of defense.

Layer 1: File Skip (Fingerprint Identity)

- **Identifer**: Every file in GDrive has a permanent `drive_file_id`.
- **Logic**: Before downloading a CSV, the system checks if that `file_id` is in the database.
- **Result**: If the file was already processed, it is skipped instantly. Moving or renaming a file **does not** cause duplicates because the ID never changes.

Layer 2: Record Protection (The unique_business Lock)

- **Identifier**: A combination of `(Name + Address + Phone Number)`.
- **Logic**: The database has a "Unique Index". It is physically impossible for the database to accept two rows where the Name, Address, and Phone all match perfectly.
- **Normalization**: Before checking, the system cleans the data (standardizes phone numbers, trims whitespace, removes symbols).

- **Result**: If a row is a repeat, the database uses `INSERT IGNORE` to safely throw it away.

---

4. Performance Statistics

- **Concurrent Workers**: 32 Parallel Threads.
- **Ingestion Speed**: Optimized for 15,000+ rows per second.
- **Batching**: Saves data in chunks of **2,000 rows** for maximum efficiency.
- **Sync Latency**: New files typically appear on the dashboard within **5-30 seconds** of being uploaded to Drive.

---

5. Understanding the Logs (Terminal)

Log Message	Meaning
🔍 `Change Detected`	Google Drive reported an update (New file, rename, or move).
⏭️ `SKIPPED`	The File ID already exists in the database. No work needed.
🆕 `NEW`	Found a brand new file that hasn't been processed.
✅ `DONE`	File successfully downloaded, normalized, and rows saved.
✨ `Reactive Cycle finished`	The engine has finished catching up with all recent drive changes.
⚡ `System Idle`	Drive and Database are 100% in sync. No new work found.

---

6. Database Metadata

The system tracks its progress in a separate table: * **`etl_metadata`** : Stores the `last_change_token` so the engine remembers its position even if the server restarts. * **`drive_folder_registry`** : Maps the structure of your drive for the Explorer view.

Status: ✅ Production Ready | **Version**: 5.0 | **Engine**: Antigravity High-Speed Parallel ETL