**Extract and Format Healthcare Contact Data from Large Dataset (.csv, 10.6GB)**

Posted yesterday

Worldwide

Specialized profiles can help you better highlight your expertise when submitting proposals to jobs like these. [Create a specialized profile](https://www.upwork.com/freelancers/).

Summary

I need an experienced data professional to help extract and reformat contact and professional information from a large, dataset of U.S. healthcare professionals. The full file is approximately 10.6 GB with over 300 columns and hundreds of thousands of records. A smaller sample file (200 records) is provided for reference.  
  
📌 Project Objectives:  
- Extract only the relevant contact and professional fields from the raw dataset.  
- Format a clean, structured .csv file with clearly labeled headers.  
- Normalize address entries based on comparison between Business Practice Location and Business Mailing Location:  
  - If the two addresses are identical, output one row and include a new column labeled Address Type with the value BP.  
  - If the two addresses are different, output two rows — one for each — and in the new Address Type column label each Address Type with the value BP or BM.  
- Provide the clean outputted file to me  
- Provide me with counts for unique records for each Taxonomy Code  
- I may need separate files pulled by Taxonomy Codes as well.  
  
✅ Fields to Include in Output:  
  
Basic Identification:  
- NPI  
- Provider First Name  
- Provider Last Name (Legal Name)  
- Provider Credential Text  
- Provider Organization Name (Legal Business Name)  
  
Authorized Official (used when individual name is missing):  
- Authorized Official First Name  
- Authorized Official Last Name  
- Authorized Official Title or Position  
  
Professional Info:  
- Provider Enumeration Date  
- Provider License Number(s)  
- Provider License State(s)  
- Healthcare Provider Taxonomy Code(s)  
  
Location Info (based on Address Type – BP or BM):  
- Street Address  
- City  
- State  
- Zip Code  
- Phone Number  
- Address Type (either BP for Business Practice or BM for Business Mailing)  
  
📝 Output Format Requirements:  
- One row per address (unless BP and BM match, then just one row).  
- Clean and human-readable column headers (e.g., First Name, Last Name, Street Address, City, State, Zip Code, Phone, Address Type, etc.).  
- Consistent formatting and deduplication logic where applicable.  
- Handle missing or empty name fields by falling back to the Authorized Official information.  
  
📚 Special Handling: Healthcare Provider Taxonomy Codes  
Each provider may have up to 15 taxonomy codes, stored in fields named:  
  Healthcare Provider Taxonomy Code\_1  
  Healthcare Provider Taxonomy Code\_2  
  ...  
  Healthcare Provider Taxonomy Code\_15  
  
Some records have gaps (e.g., Taxonomy Code\_1 is empty, but Taxonomy Code\_2 has a value). These codes must be consolidated and shifted up so that:  
- Healthcare Provider Taxonomy Code\_1 always contains the first available code  
- ...\_2, ...\_3, etc. are filled in order with any additional codes  
- All empty columns follow the populated ones (no gaps in the output)  
  
Example:  
Original:  
  Code\_1: (empty)  
  Code\_2: 207Q00000X  
  Code\_4: 207R00000X  
  
Reformatted:  
  Code\_1: 207Q00000X  
  Code\_2: 207R00000X  
  Remaining: (empty)  
  
💾 Provided to You:  
- A sample .csv file (200 records) to preview the column layout and data structure.  
- A list of relevant fields (as above).  
- Clarification on address comparison logic and output expectations.  
  
💡 Ideal Freelancer Will Have:  
- Strong experience with large-scale data wrangling using Python (Pandas), R, or advanced Excel/Power Query.  
- Familiarity with healthcare or NPI datasets (preferred).  
- Ability to deliver accurate, well-formatted, and deduplicated results.  
- Strong attention to edge cases and clean formatting.  
  
Please include a short summary of your approach to:  
1. Comparing two sets of address fields.  
2. Handling missing individual names (e.g., fallback to authorized official).  
3. Ensuring efficient processing for a 10+ GB file.