

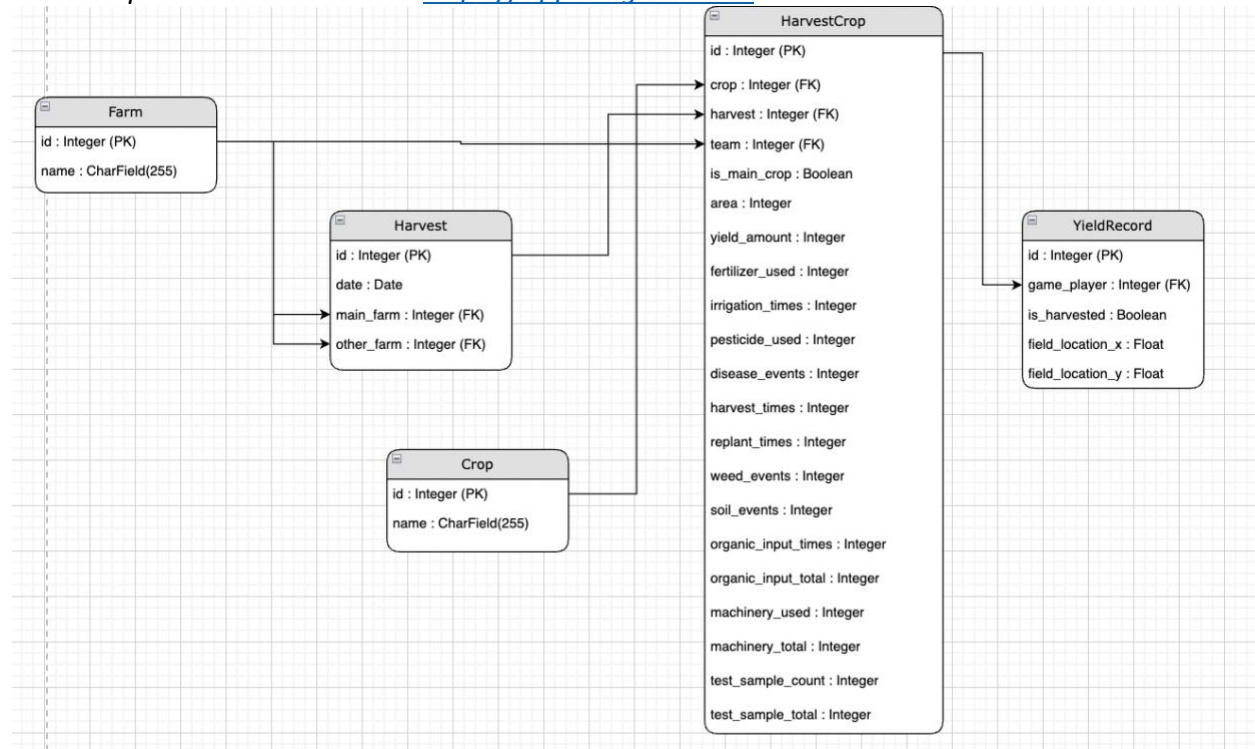
## Database Structure Description – Written Response

Tianye

Based on the raw\_data folder's json files, I created 5 tables. From `crops.json` and `farms.json` I constructed the `Farm` table, which stores basic info about each farm, with farm id as primary key and farm's name. And the `Crop` table which also stores each crop's unique id and its name. In the `harvests.json` there's a Nesting Structure, after formatting it I identified and made 3 different relations. The outer structure is the `Harvest` model, which contain a primary key Integer id, a date field, main\_farm field and other\_farm field. And next is the `HarvestCrop` model, it contains the statistics of a crop's performance on different harvests. The model has 3 foreign keys, pointing to the previous 3 tables and referencing this crop's info, which harvest event the crop was involved at the time and whether it is main or other farm. This table and its connections ensures the many-to-many relationships between crop and harvests, like each crop can participate in several harvests and each harvest could feature several crops. Lastly the `YieldRecord` table has the foreign key harvest\_crop pointing to the HarvestCrop model, to show which crop had made the series of yield records. And contains the info about where the crop made the yield record and whether it successfully grew. In conclusion I think this kind of database design adheres to RDBMS principles and allows the input data's consistency and integrity.

(225 words)

Visual Representation made via <https://app.diagrams.net> :



(One minor feature about design here is by PostgreSQL convention the fields are in snake\_notation, but I think in the jsons they are in camelCase, thus might need a conversion helper function to handle the data after loading.)