NOTE:
- Field name were shortened to facilitate
  putting into the database design
- By notation, all relation and field names
  must be in small caps

==========

01. hoa

- name* << db designer decided
  that this is the
  best field to use as
  identifier since
  there is only one
  record of hoa and it
  can be identified
  through its name
- ofcaddress
  - streetno
  - streetname
  - barangay
  - city
  - province
  - region
  - zipcode
  - coordx
  - coordy
- estabyear

- website
- subdname
- aofincorporation_fileid   (FK) << rel 1
- bylaws_fileid   (FK) << rel 2
- minutesga_fileid   (FK) << rel 3
- attendance_fileid   (FK) << rel 4
- certification_filedid   (FK) << rel 6
- coethics_fileid   (FK) << rel 7
- hasotherhoa (Y/N) <<   deleted, if the
  otherhoaname field
  has a value, it
  automatically means
  there exists other
  hoa
- otherhoaname
- otherhoaaddr
  - streetno
  - streetname
  - barangay
  - city
  - province
  - region
  - zipcode
  - coordx

- coordy
- regmondues
- regcolday (1..20)

==========

02. files
- fileid*
- filename
- description
- filelocation
- filetype
- submitdate
- uploader
- fileowner << deleted, fileowner is the hoa and the decision of the designer is to have the relation hoa implement the FK instead of the relation file, as reflected in relationships 1,2,3,4,6, and 7. This decision is because, the files being submitted by the hoa is

more controlled rather than
if the relation file
implements the FK, where
it is possible that many
files (more than the
required) may be recorded.

NOTE:

- The designer decided to use FILEID as identifier instead of filelocation and filename to simplify the implementation of FK in the relation hoa.
- however, the combination of filelocation and filename will still be implemented as unique to avoid the problem of having two exactly the same file be recorded in the relation. Since in MYSQL Workbench, composite attributes cannot be declared as unique, this will be implemented in SQL DDL instead.

```
CREATE TABLE files (
    fileid          INT(7),
    filelocation    VARCHAR(45),
    filename        VARCHAR(45),
    description          VARCHAR(45),
```

```
    filetype        VARCHAR(45),
    submitdate      DATE,
    uploader        VARCHAR(45),
    primary key (fileid),
    UNIQUE (filelocation, filename)
);
```

==========

03. homeowner
    - homeownerid*
    - hoaname   (FK) << relationship 5
    - completename
      - lastname
      - firstname
      - middlename
    - yearsasho
    - properties[] << removed since
                           relationship 8 will
                           make the homeownerid
                           is instead recorded
                           in properties, rather
                           than the properties
                           recorded in homeowner

- birthday
- gender (M/F)
- email
- mobileno []
- fburl
- picture [U]   << filename should
                  be recorded instead
                  of actual picture
- undertaking (Y/N)
- expression (Y/N)
- otheraddr
  - streetno
  - streetname
  - barangay
  - city
  - province
  - region
  - zipcode
  - coordx
  - coordy
- othercontact
  - email
  - mobileno

NOTE:

- The designer took note of the reality
  that when picture files are saved to the
  physical disk, the filenames should be unique.
  this was noted with a [U].

Functional Dependencies

03.1. province   --> region

03.2. barangay,city,province,region --> zipcode

==========

04. property

- propertycode*
- homeownerid   (FK) << relationship 8
- hoaname   (FK) << indicates the
                            if property is
                            owned by hoa
                            and is deemed
                            asset
- size
- turnoverdate
- classification (R/C)   << DB designer
                    decided not to

remove this even
if it is
automatically
deemed
commercial if
there is a value
of commercial
type since the
relation can be
a candidate for
specialization.
The DB Designer
took note that
only residential
properties have
residents, not
commercial
properties.

- commercialtype
- maximumtenants

NOTE:

- The db designer has analyzed that a property owned by the hoa and are assets, are not

residential or commercial properties based on the samples provided in the case. Properties owned by omeowners therefore are either residential or commercial, and based on the case only residential properties have household.

==========

05. household

- householdid*
- propertycode (FK) << DB designer, based on relationship 11, decided to have the propertycode in the household instead of the other way around because properties are recorded first than households. this means that

a property may be recorded without a household and will need to be updated when the household record is created. This is will result to extra processing of the data.

- receiptno (FK)
- billid (FK)

==========

06. resident

- residentid*
- homeownerid (FK) << implementing relationship 9. Homeowners are recorded first than the residents.
- householdid (FK) << implementing

relationship 13

- isauthorized (Y/N)
- completename
  - lastname
  - firstname
  - middlename
- renter (Y/N)
- email
- birthday
- gender (M/F)
- mobileno
- fburl
- picture [U] << filename should
  be recorded instead
  of actual picture
- relationship
- undertaking (Y/N)
- lastupdatedate << to determine if
  record is more than
  a year already
  - businessid (FK)
  - billid (FK)

NOTE:

- The designer took note of the reality

that when picture files are saved to the physical disk, the filenames should be unique. this was noted with a [U].

- The designer took note that the resident and homeowner are relations of the same nature having common identifiers and common attributes
- The designer took note of the consequence of the decision to implement relationship 14 this way. It is possible that in the records, the resident is recorded as a homeowner, but it so happens that the household it belongs to is not the property of the homeowner. This is the reality of db design, there are cases that the limitations of the relational model will not allow certain conditions of the data to be completely fulfilled, and this has to be taken note of that the software component of the application system will check.

==========

07. residentidcard
    - cardno*

    - residentid   (FK) << relationship 14
    - cancelled (Y/N)
    - requestdate
    - reason
    - providedate
    - authorizingofficer (FK) relationship 15
      - homeownerid
      - position
      - electiondate
    - ornumber [U]
    - cardfee

NOTE:
- The designer took note of the legal
  requirement that official receipt numbers are
  unique. This was noted with a [U].

==========

08. vehicle
    - plateno*   << by law, plateno
                  is unique per vehicle
    - owner
      - lastname

```
      - firstname
      - middlename
    - residentid  (FK)
    - homeownerid (FK)   << relationship 10
    - classification
    - type
    - orcr     [U]  << filename should
                    be recorded instead
                    of actual picture
    - orcrupdated (Y/N)
    - regdate
    - regfee
    - ownertype (R/N)    << removed, if there
                    is a value on
                    residentid, then it
                    automatically means
                    the owner is a
                    resident

NOTE:
- The designer took note of the reality
  that when picture files are saved to the
  physical disk, the filenames should be unique.
  this was noted with a [U].
```

==========

09. sticker
    - stickerid*
    - validityyear
    - plateno (FK) << relationship 16
                vehicles receive
                   multiple stickers
                since it says every
                year
    - authorizingofficer (FK) << relationship 17
      - homeownerid
      - position
      - electiondate

==========

10. officer
    - homeownerid*
    - position (P/V/T/A/S)*
    - election
      - electiondate*   << the DB designer
                decided to use the

election date as
part of the
identifier of officer
to represent the
election since no two
elections will
definitely be held on
the same date, that
makes the date to be
unique for every
election. Election is
a highly composite
attribute, and it's
entirety is not best
to be used as part of
PK.

- venue
- quorum (Y/N)
- witness
  - completename
    - lastname
    - firstname
    - middlename
- mobileno

- startdate
- enddate
- availdays
- availtime (M/A)

Functional Dependencies

10.1. electiondate --> venue, quorum,
witness

10.2. witnesscompletename --> mobileno

==========

11. asset
- assetid*
- propertycode (FK) << db designer decided
to implement
relationship 12
by recording the
propertycode in
asset rather than
the other way around
since properties are
recorded first
before assets.

- assetname
- description
- acquisitiondate
- forrent (Y/N)
- value
- type (P/E/F/O)
- status (W/D/R/P/S)
- locX
- locY
- assetid []   << relationship 19

==========

12. Rental   << relationship 18
                  is a * to *
                  relationship, this
                  will require another
                     relation to record
                  the resident and the
                  assets rented. It is
                  a reality that assets
                  can be rented
                  multiple time of the
                  same resident, but of

different dates of course, this would mean that a field on rent date should be part of the identifier and not only the asset and resident that rented it.

- assetid* (FK)
- residentid* (FK)
- rentdate*

==========

13. business

- businessname
- completeownername
- isownerresident (Y,N)
- address
    - streetno
    - streetname
    - barangay

    - city_municipality
    - province
    - region
    - zipcode
    - locx
    - locy
- businessemail
- description
- businesstype (S, P, C)
- startdate
- operatingschedule
- website
- copyofpermit
- permitexpirydate
- authorizing_officer (FK)
    - officer_homeownerid*
    - officer_position (controlled)
    - officer_electiondate
- enddate
- businessID**
- billID (FK)

14. employee

- completename

    - lastname
    - firstname
    - middlename
- mobileno
- position (controlled)
- isauthorized (Y/N)
- employeeid**

15. monthlydue
- regulardue
- otherdues
    - penalties
    - damagestoproperties
- monthlydueID**
- billID (FK)

16. billing
- unpaiddue
- incentive
- discountawarded
- penaltyincurred
- totalamounttocollect
- billID*
- monthofbill

- yearofbill
- dategenerated
- collectionday

17. payment

- dateofpayment
- isfullpayment (Y/N)
- isadvance (Y/N)
- amountpaid
- personpaying
- receivingofficer
- ornumber* (controlled)
- billID (FK)
- businessID (FK)

18. incident

- incidentdate
- incidentdescription
- nameofperson []
- ispersonresident (Y/N)
- nameofbusinessinvolved
- investigatingofficer
- penaltyimposed
- ruleno []

- secondingofficer
- incidentID *
- businessID (FK)

19. evidence
- evidenceID*
- evidencename
- evidencedescription
- evidencefilename
- residentthatsubmitted
- officerthataccepted
- datesubmitted
- incidentID (FK)

20. assetactivity
- activitydate
- activitydescription
- personincharge
- mobileno
- activityloc
- authorizing_officer (FK)
    - officer_homeownerid*
    - officer_position (controlled)
    - officer_electiondate

- tentativeschedule
    - startdate
    - enddate
- actualstart
- actualend
- costofactivity
- officialreceipt
- activitystatus (S, O, C, D)
- activityno*
- assetID (FK)