

Part 1: Design

Terps Abode Realtors

Terps Abode Realtors is a medium-sized business based in College Park (Maryland). Terps Abode Realtors assists students in finding suitable housing in and around the University of Maryland (College Park Campus), according to the students' requirements. Terps Abode Realtors needs a database management system that will help it in becoming the best in business and provide the greatest value to its clients, primarily students.

Business Transactions:

We are designing a conceptual schema for Terps Abode Realtors using the entity-relationship model. The transactions include:

- Each unit is described by a unique unit ID, unit type (apartment, townhouse, family home), year of construction, number of beds, number of baths, rent, area, unit number, unit address (street, city, state and zip code).
- Additionally, each unit will also have the distance from campus (Adele H. Stamp Student Union is considered the center of the campus).
- Each owner is described by a unique owner ID, owner type (individual or corporate), owner's first name, owner's last name, phone number, email, and owner address (street, city, state and zip code).
- Each unit is managed by exactly one owner whereas each owner can manage multiple units. An owner can either be an individual (who bought the house and is now leasing it) or a corporation.
- Each unit can have various amenities (in-unit washer, dryer, dishwasher, pool, gym, terrace, or barbeque area) which is described by a unique amenity ID and amenity description.
- Each agent is described by a unique agent ID, title, first name, last name, phone number, and email.
- Each client is described by a unique client ID, first name, last name, phone number, and email.
- Each client is assigned to exactly one agent whereas an agent can have zero or more clients assigned to him/her.
- For each unit that has been leased through an agent, there is a single-client who signs the lease for that unit. There is a unique lease ID, lease start date, and lease end date associated with each lease agreement.
- The company maintains client information even if the client did not end up finding a suitable housing option, and thus not signing a lease through Terps Abode Realtors.

Please note :

1. Some business processes/transactions cannot be captured in the ER diagram.
2. ER Diagram follows in Part 2 (page 4) of this document.

Part 2: Proposal

Mission Statement:

Our consulting firm specializes in providing database design and analytics services. We have experts in the field of data analytics, database design, database development and management, and application development. We have 30+ years of experience in the field of IT consulting with proven expertise in creating robust database management solutions for our clients.

Our client, Terps Abode Realtors is a medium sized business based in College Park (Maryland). Terps Abode Realtors assists students in finding suitable housing in and around the University of Maryland (College Park Campus), according to the students' requirements. Terps Abode Realtors requires a database management system that will help it in becoming the best in business and provide the greatest value to its clients, primarily students.

The mission is for Terps Abode Realtors to become a leading rental pairing agency located in College Park, MD dedicated to creating value for all stakeholders in the student housing rental market ecosystem. They realize this core principle by developing deep relationships with property owners, life-long relationships with renters and by recognizing and acknowledging the impact TAR's agents drive throughout this process.

By empowering agents with owners and client data, visibility and knowledge relating to the housing market, TAR will be able to match clients (mostly students) with the ideal housing situation that fits their needs while ensuring that owners maximize their revenue from renting.

Mission Objectives:

To help TAR achieve its mission statement we propose the following objectives:

- Collate data of all required fields in a structured format and populate the database. Further, design and create a database which will store details of:
 - rental housing properties in and around College Park,
 - the respective owner,
 - amenities within each property,
 - agents working within the organization,
 - the TAR's customers, and
 - leases signed through the client
- Give the client the power to query the database to get insights. Some examples are:
 - the available properties that match a customer's requirements
 - the details of leases signed in the past through agents working within the company
 - details of top performing agents
 - details of properties reaching the end of their lease term

ER Schema:

Entities, Attributes and Primary Keys

Unit(**unitId**, unitType, unitYearOfConstruction, unitBed, unitBath, unitRent, unitArea, unitNumber, unitAddress, -unitStreet, -unitCity, -unitState, -unitZip, unitDistFromCampus)

Owner(**ownId**, ownType, ownFirstName, ownLastName, ownPhone, ownEmail, ownAddress, -ownStreet, -ownCity, -ownState, -ownZip)

Amenity(**amnId**, amnDescription)

Agent(**agtId**, agtTitle, agtFirstName, agtLastName, agtPhone, agtEmail)

Client(**cltId**, cltFirstName, cltLastName, cltPhone, cltEmail)

Relationships, Attributes, Degrees, Participating Entities and Constraints

Assign: binary relationship

1 Client to 1 Agent

1 Agent to 0 or more Clients

Contain: binary relationship

1 Unit to 0 or more Amenities

1 Amenity to 0 or more Units

Manage: binary relationship

1 Unit to 1 Owner

1 Owner to 0 or more Units

Sign (leaseId, leaseStartDate, leaseEndDate): ternary relationship

1 Client and 1 Agent to 1 Unit

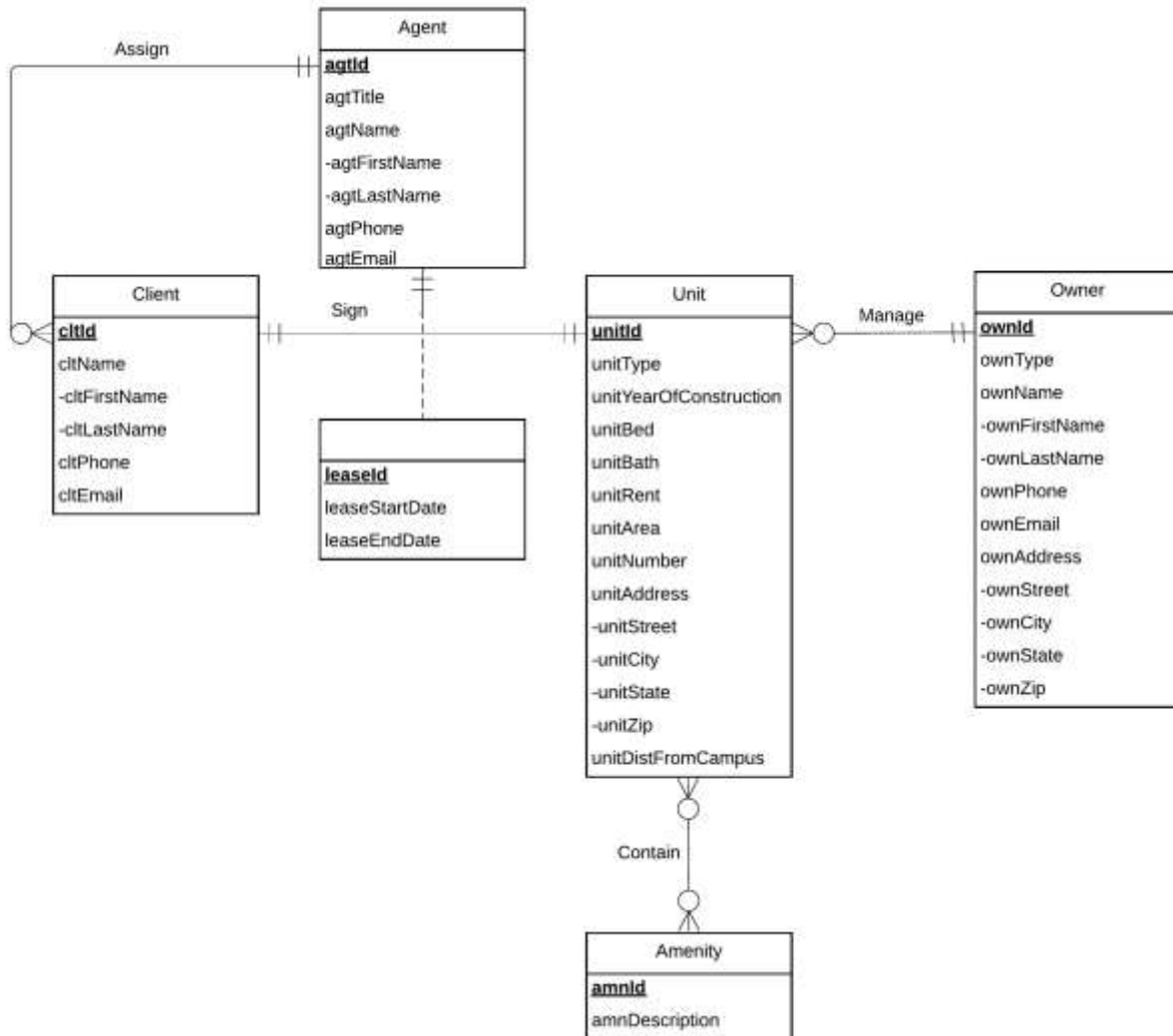
1 Agent and 1 Unit to 1 Client

1 Unit and 1 Client to 1 Agent

Please note: ER Diagram follows on the next page.

ER Diagram

Terps Abode Realtors



Relations:

Unit (unitId, unitType, unitYearOfConstruction, unitBed, unitBath, unitRent, unitArea, unitNumber, unitStreet, unitCity, unitState, unitZip, unitDistFromCampus, ownId)

Owner (ownId, ownType, ownFirstName, ownLastName, ownPhone, ownEmail, ownStreet, ownCity, ownState, ownZip)

Amenity (amnId, amnDescription)

Agent (agtId, agtTitle, agtFirstName, agtLastName, agtPhone, agtEmail)

Client (cltId, cltFirstName, cltLastName, cltPhone, cltEmail, agtId)

Contain (unitId, amnId)

Sign (leaseId, unitId, agtId, cltId, leaseStartDate, leaseEndDate)

Functional Dependencies:

unitId → unitType, unitYearOfConstruction, unitBed, unitBath, unitRent, unitArea,
unitNumber, unitStreet, unitCity, unitState, unitZip, unitDistFromCampus, ownId
ownId → ownType, ownFirstName, ownLastName, ownPhone, ownEmail,
ownStreet, ownCity, ownState, ownZip
amnId → amnDescription
agtId → agtTitle, agtFirstName, agtLastName, agtPhone, agtEmail
cltId → cltFirstName, cltLastName, cltPhone, cltEmail, agtId
unitId, amnId →
leaseId → unitId, agtId, cltId, leaseStartDate, leaseEndDate

Normalization:

Unit (**unitId**, unitType, unitYearOfConstruction, unitBed, unitBath, unitRent,
unitArea, unitNumber, unitStreet, unitCity, unitState, unitZip, unitDistFromCampus,
ownId) = 3NF
Owner (**ownId**, ownType, ownFirstName, ownLastName, ownPhone, ownEmail,
ownStreet, ownCity, ownState, ownZip) = 3NF
Amenity (**amnId**, amnDescription) = 3NF
Agent (**agtId**, agtTitle, agtFirstName, agtLastName, agtPhone, agtEmail) = 3NF
Client (**cltId**, cltFirstName, cltLastName, cltPhone, cltEmail, agtId) = 3NF
Contain (**unitId**, **amnId**) = 3NF
Sign (**leaseId**, unitId, agtId, cltId, leaseStartDate, leaseEndDate) = 3NF

Business rules:

- [R1] When ownership information changes in the database, the corresponding unit information should be changed accordingly.
- [R2] When a unit is owned by an owner, that owner cannot be deleted.
- [R3] When agent information changes in the database, the corresponding client information should be changed accordingly.
- [R4] When an agent is assigned to a client, that agent cannot be deleted.
- [R5] Amenities cannot be deleted or updated.
- [R6] Units cannot be deleted or updated.
- [R7] When a lease is signed for a unit with a client through an agent, the unit, client and agent cannot be deleted or changed in the database.

Please note: Referential Integrity table follows on the next page.

Referential Integrities:

Relation	Foreign Key	Base Relation	Primary Key	Business Rule	Constraint : ON DELETE	Business Rule	Constraint: ON UPDATE
Unit	ownId	Owner	ownId	R2	NO ACTION	R1	CASCADE
Client	agtId	Agent	agtId	R4	NO ACTION	R3	CASCADE
Contains	amnId	Amenity	amnId	R5	NO ACTION	R5	NO ACTION
Contains	unitId	Unit	unitId	R6	NO ACTION	R6	NO ACTION
Sign	unitId	Unit	unitId	R7	NO ACTION	R7	NO ACTION
Sign	cltId	Client	cltId	R7	NO ACTION	R7	NO ACTION
Sign	agtId	Agent	agtId	R7	NO ACTION	R7	NO ACTION

Sample Data:

Please note: We've provided a screenshot of the sample data that was created in an excel file. The excel file has also been inserted after the screenshot.

Unit													
unitId	unitType	unitYearOfConstruction	unitBed	unitBath	unitRm	unitArea	unitNumber	unitStreet	unitCity	unitState	unitZip	unitDistFromCampus	ownId
U001	Townhouse	1996	3	2	3200	1600	12	Stanford St	Hyattsville	MD	20783	7	O006
U002	Apartment	2008	1	1	1450	800	23	Mowatt Lane	College Park	MD	20783	8.8	O011

Owner									
ownId	ownType	ownFirstName	ownLastName	ownPhone	ownEmail	ownStreet	ownCity	ownState	ownZip
O011	Individual	Adam	James	5234567890	adam.james@gmail.com	Tulane Drive	Hyattsville	MD	20783
O006	Individual	Smith	Joe	4569023457	smith.joe123@gmail.com	Mowatt Lane	College Park	MD	20783

Amenity	
amnId	amnDescription
A010	Dishwasher
A002	Laundry

Agent					
agtId	agtTitle	agtFirstName	agtLastName	agtPhone	agtEmail
AG007	Manager	Ruth	Jo	4567812345	ruthjo@gmail.com
AG015	Executive	Kate	Li	451726496	kate@gmail.com

Client					
cltId	cltFirstName	cltLastName	cltPhone	cltEmail	agtId
T100	John	Graham	9912347256	jg@cs.cmu.edu	AG007
T020	Mary	Levine	3865280669	mlewis@cs.cmu.edu	AG015

Contains	
unitId	amnId
U001	A010
U002	A002

Sign					
leaseId	unitId	agtId	unitId	leaseStartDate	leaseEndDate
L005	U001	AG007	T100	4/11/15	6/15/19
L010	U002	AG015	T020	6/3/16	5/4/17

Embedded excel file that contains the sample data:

