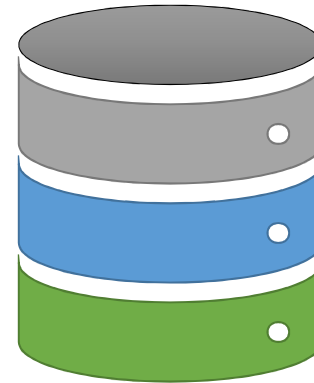
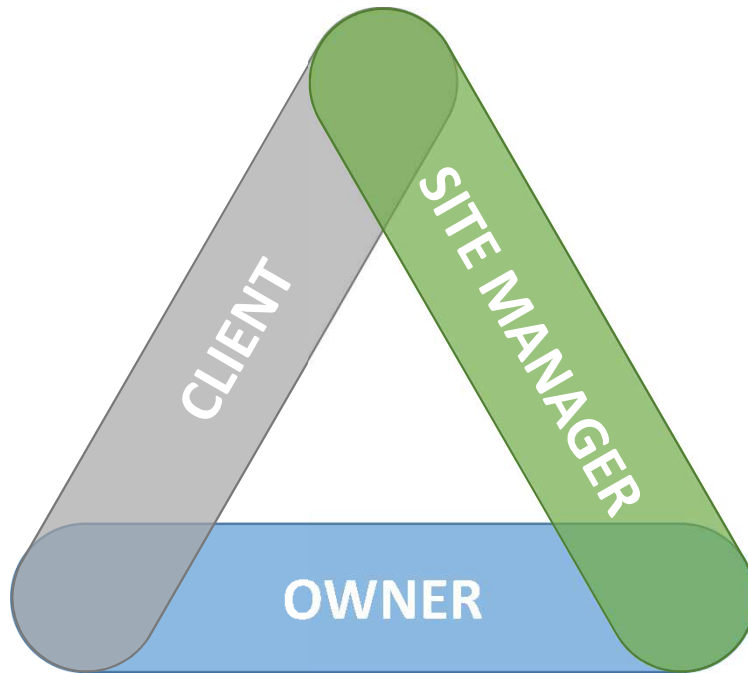


Layers



Layer's functions

CLIENT

- Search for properties based on location and other features.
- Set favorites properties.
- Generate printer-friendly reports from the site.
- Generate requests for property sighting

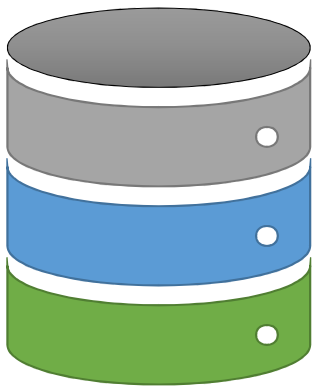
OWNER

- Enter properties
- Contact clients
- Schedule visit/tours

SITE MANAGER

- Client management
- Owner management
- Property items management (not content)
- Ban & publish powers for clients, owner and properties

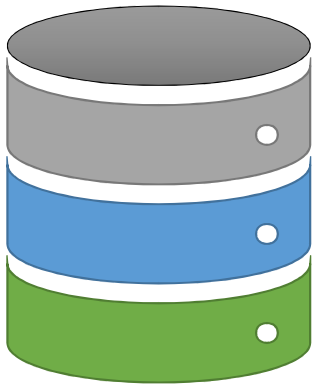
Tables – Client



clients		
Description	Column	Data type and constraints
ID	id	INT NOT NULL AUTO_INCREMENT
First name	firstName	VARCHAR (20) NOT NULL
Last name	lastName	VARCHAR (20) NOT NULL
Email	email	VARCHAR (20) NOT NULL
Phone	phone	VARCHAR (20) NOT NULL

favorites		
Description	Column	Data type and constraints
ID	id	INT NOT NULL AUTO_INCREMENT
List name	listName	VARCHAR (20) NOT NULL

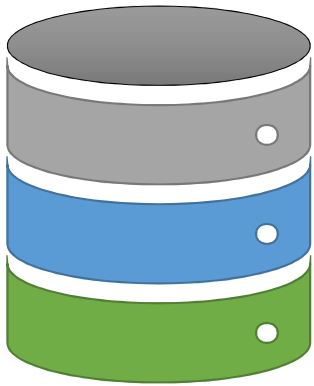
Tables – Owner



owner		
Description	Column	Data type and constraints
ID	id	INT NOT NULL AUTO_INCREMENT
First name	firstName	VARCHAR (20) NOT NULL
Last name	lastName	VARCHAR (20) NOT NULL
Street address 1	address1	VARCHAR (20) NOT NULL
Street address 2	address2	VARCHAR (20) NOT NULL
City	city	VARCHAR (20) NOT NULL
Province or State	prov	VARCHAR (50) NOT NULL
Country	country	VARCHAR (20) NOT NULL
Phone	phone	VARCHAR (20) NOT NULL
Fax	fax	VARCHAR (20)
URL	url	VARCHAR (20)

seenBy		
Description	Column	Data type and constraints
ID	id	INT NOT NULL AUTO_INCREMENT
Property ID		
User ID		
Date seen		

Tables – Owner (Cont.)



properties *					
Description	Column	Data type and constraints	Description	Column	Data type and constraints
ID	id	INT NOT NULL AUTO_INCREMENT	Ownership type	ownerType	TINYINT(1) NOT NULL
Description	description	VARCHAR (255) NOT NULL	View type	viewType	TINYINT(1) NOT NULL
MLS number	mlsNumber	TINYINT(10)	Building type	buildingType	TINYINT(2) NOT NULL
Street address 1	firstName	VARCHAR (20) NOT NULL	Construction style	constStyle	TINYINT(1) NOT NULL
Street address 2	lastName	VARCHAR (20) NOT NULL	Basement type	basementType	TINYINT(1) NOT NULL
City	email	VARCHAR (20) NOT NULL	Heating type	heating	TINYINT(1) NOT NULL
Province or State	phone	VARCHAR (20) NOT NULL	Parking type	parking	TINYINT(1) NOT NULL
Country	country	VARCHAR (20) NOT NULL	Parking spaces	parkingSpaces	TINYINT(1) NOT NULL
Longitude	long	TINYINT(10) UNSIGNED NOT NULL	Air condition	ac	TINYINT(1) NOT NULL
Latitude	lat	TINYINT(10) UNSIGNED NOT NULL	Pool	pool	BOOLEAN DEFAULT false
Price	price	MONEY NOT NULL	Fireplace	fireplace	BOOLEAN DEFAULT false
# of bedroom	beds	TINYINT(3) NOT NULL	Garage	garage	BOOLEAN DEFAULT false
# of bathroom	baths	TINYINT(3) NOT NULL	Waterfront	waterfront	BOOLEAN DEFAULT false
# of half bathrooms	halfBaths	TINYINT(3) NOT NULL	Land size	acres	FLOAT(2) DEFAULT 0
# of storeys	storeys	TINYINT(3) NOT NULL	Images	images	VARCHAR (255) NOT NULL
Property type	propType	TINYINT(1) NOT NULL	Date start	dateIn	DATE NOT NULL
Transaction type	transactionType	TINYINT(1) NOT NULL	Date end	dateOut	DATE NOT NULL

* Based on the npm package **realtorca**



Still to be defined

- **SQL queries:**

- Client creates *favorite* lists.
- Client searches for properties based on straight criteria of the “properties” table.
- Client searches for properties based on a city or address (does **npm realtorca** does this?)
- Owner create new property.
- Owner manages (update/delete) existing property.
- Owner searches for clients that saw a specific property.
- Owner searches for the properties most/least/last viewed.
- Owner searches for sold properties.
- Site manager searches for all users.
- Site manager searches for all realtors.
- Site manager searches for all realtor companies.
- Site manager searches for all properties.

- **Processes**

- A “printer friendly” report of the property is provided (PDF?)

- **About the site:**

- Look of the site
- Name of the site
- Navigation experience of the site

- **Investigate:**

- How to use and implement **npm realtorca**.
- How to use social media authentication as client login (*Oauth2 workflow*)

Navigation

