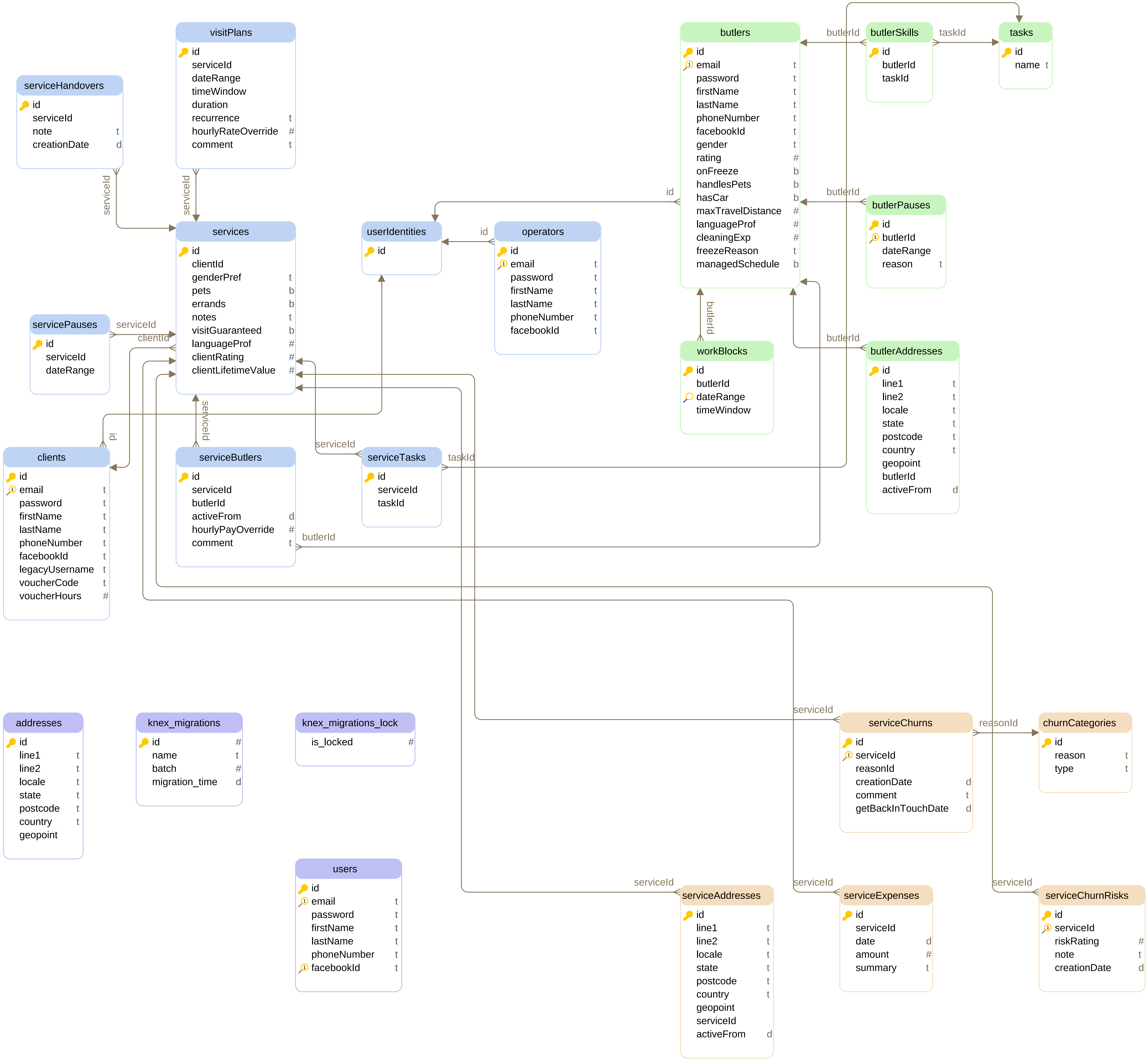
visitPlans
id
serviceId
dateRange
timeWindow
duration
recurrence t
hourlyRateOverride #
comment t
serviceHandovers
id
serviceId
note t
creationDate d
serviceId
serviceId
services
id
clientId
genderPref t
pets b
errands b
notes t
visitGuaranteed b
languageProf #
clientRating #
clientLifetimeValue #
servicePauses
id
serviceId
dateRange
serviceId
clientId
userIdentities
id
id
operators
id
email t
password t
firstName t
lastName t
phoneNumber t
facebookId t
id
butlers
id
email t
password t
firstName t
lastName t
phoneNumber t
facebookId t
gender t
rating #
onFreeze b
handlesPets b
hasCar b
maxTravelDistance #
languageProf #
cleaningExp #
freezeReason t
managedSchedule b
butlerId
butlerSkills
id
butlerId
taskId
taskId
tasks
id
name t
butlerId
butlerPauses
id
butlerId
dateRange
reason t
butlerId
workBlocks
id
butlerId
dateRange
timeWindow
butlerId
butlerAddresses
id
line1 t
line2 t
locale t
state t
postcode t
country t
geopoint
butlerId
activeFrom d
id
clients
id
email t
password t
firstName t
lastName t
phoneNumber t
facebookId t
legacyUsername t
voucherCode t
voucherHours #
serviceId
serviceButlers
id
serviceId
butlerId
activeFrom d
hourlyPayOverride #
comment t
butlerId
serviceId
serviceTasks
id
serviceId
taskId
taskId
addresses
id
line1 t
line2 t
locale t
state t
postcode t
country t
geopoint
knex_migrations
id #
name t
batch #
migration_time d
knex_migrations_lock
is_locked #
users
id
email t
password t
firstName t
lastName t
phoneNumber t
facebookId t
serviceId
serviceChurns
id
serviceId
reasonId
creationDate d
comment t
getBackInTouchDate d
reasonId
churnCategories
id
reason t
type t
serviceId
serviceAddresses
id
line1 t
line2 t
locale t
state t
postcode t
country t
geopoint
serviceId
activeFrom d
serviceId
serviceExpenses
id
serviceId
date d
amount #
summary t
serviceId
serviceChurnRisks
id
serviceId
riskRating #
note t
creationDate d

Default Layout

@DbSchema

Table addresses

* Pk	id	uuid	
*	line1	varchar(255)	
	line2	varchar(255)	
*	locale	varchar(100)	
*	state	varchar(100)	
*	postcode	varchar(10)	
*	country	varchar(100)	
*	geopoint	point	
Indexes			
Pk	addresses_pkey	id	

Table butlerAddresses

* Pk	id	uuid	
*	line1	varchar(255)	
	line2	varchar(255)	
*	locale	varchar(100)	
*	state	varchar(100)	
*	postcode	varchar(10)	
*	country	varchar(100)	
*	geopoint	point	
*	butlerId	uuid	
*	activeFrom	date	
Indexes			
Pk	butlerAddresses_pkey	id	
Foreign Keys			
	butleraddresses_butlerid_foreign (butlerId) ref butlers (id)		

Table butlerPauses

* Pk	id	uuid	
* Unq	butlerId	uuid	
*	dateRange	daterange	
	reason	varchar(255)	
Indexes			
Pk	butlerPauses_pkey	id	
Unq	butlerpauses_butlerid_unique	butlerId	
Foreign Keys			
	butlerpauses_butlerid_foreign (butlerId) ref butlers (id)		
Constraints			
	nonEmptyDateRange	"dateRange" <> 'empty'::daterange	

Table butlerSkills

* Pk	id	uuid	
*	butlerId	uuid	

Table butlerSkills

*	taskId	uuid	
Indexes			
Pk	butlerSkills_pkey	id	
Foreign Keys			
	butlerskills_butlerid_foreign (butlerId) ref butlers (id)		
	butlerskills_taskid_foreign (taskId) ref tasks (id)		

Table butlers

* Pk	id	uuid	
* Unq	email	varchar(255)	
*	password	varchar(255)	
*	firstName	varchar(100)	
*	lastName	varchar(100)	
*	phoneNumber	varchar(15)	
	facebookId	varchar(255)	
*	gender	text	
*	rating	numeric(8,2)	
*	onFreeze	bool	
*	handlesPets	bool	
*	hasCar	bool	
*	maxTravelDistance	numeric(8,2)	
	languageProf	integer	
	cleaningExp	integer	
	freezeReason	varchar(255)	
	managedSchedule	bool	
Indexes			
Pk	butlers_pkey	id	
Unq	butlers_email_unique	email	
Foreign Keys			
	butlers_id_foreign (id) ref userIdentities (id)		
Constraints			
	check_butlers_cleaning_exp	("cleaningExp" >= 0) AND ("cleaningExp" <= 3)	
	check_butlers_language_prof	("languageProf" >= 0) AND ("languageProf" <= 2)	
	butlers_gender_check	gender = ANY (ARRAY['m'::text, 'f'::text])	

Table churnCategories

* Pk	id	uuid	
	reason	varchar(255)	
	type	varchar(255)	
Indexes			
Pk	churnCategories_pkey	id	

Table clients

* Pk	id	uuid	

Table clients

* Unq	email	varchar(255)	
*	password	varchar(255)	
*	firstName	varchar(100)	
*	lastName	varchar(100)	
*	phoneNumber	varchar(15)	
	facebookId	varchar(255)	
	legacyUsername	varchar(255)	
	voucherCode	varchar(255)	
	voucherHours	numeric(12,2)	
Indexes			
Pk	clients_pkey	id	
Unq	clients_email_unique	email	
Foreign Keys			
	clients_id_foreign (id) ref userIdentities (id)		

Table knex_migrations

* Pk	id	serial	
	name	varchar(255)	
	batch	integer	
	migration_time	timestamptz	
Indexes			
Pk	knex_migrations_pkey	id	

Table knex_migrations_lock

	is_locked	integer	

Table operators

* Pk	id	uuid	
* Unq	email	varchar(255)	
*	password	varchar(255)	
*	firstName	varchar(100)	
*	lastName	varchar(100)	
*	phoneNumber	varchar(15)	
	facebookId	varchar(255)	
Indexes			
Pk	operators_pkey	id	
Unq	operators_email_unique	email	
Foreign Keys			
	operators_id_foreign (id) ref userIdentities (id)		

Table serviceAddresses

* Pk	id	uuid	
*	line1	varchar(255)	
	line2	varchar(255)	
*	locale	varchar(100)	
*	state	varchar(100)	
*	postcode	varchar(10)	
*	country	varchar(100)	

Table serviceAddresses

*	geopoint	point	
*	serviceId	uuid	
*	activeFrom	date	
Indexes			
Pk	serviceAddresses_pkey	id	
Foreign Keys			
	serviceaddresses_serviceid_foreign (serviceId) ref services (id)		

Table serviceButlers

* Pk	id	uuid	
*	serviceId	uuid	
*	butlerId	uuid	
*	activeFrom	date	
	hourlyPayOverride	numeric(12,2)	
	comment	varchar(255)	
Indexes			
Pk	serviceButlers_pkey	id	
Foreign Keys			
	servicebutlers_butlerid_foreign (butlerId) ref butlers (id)		
	servicebutlers_serviceid_foreign (serviceId) ref services (id)		

Table serviceChurnRisks

* Pk	id	uuid	
* Unq	serviceId	uuid	
	riskRating	numeric(5,2)	
	note	varchar(255)	
	creationDate	date	
Indexes			
Pk	serviceChurnRisks_pkey	id	
Unq	servicechurnrisks_serviceid_unique	serviceId	
Foreign Keys			
	servicechurnrisks_serviceid_foreign (serviceId) ref services (id)		

Table serviceChurns

* Pk	id	uuid	
* Unq	serviceId	uuid	
	reasonId	uuid	
	creationDate	date	
	comment	varchar(255)	
	getBackInTouchDate	date	
Indexes			
Pk	serviceChurns_pkey	id	
Unq	servicechurns_serviceid_unique	serviceId	
Foreign Keys			

Table serviceChurns

	servicechurns_reasonid_foreign (reasonId) ref churnCategories (id)		
	servicechurns_serviceid_foreign (serviceId) ref services (id)		

Table serviceExpenses

* Pk	id	uuid	
*	serviceId	uuid	
*	date	date	
*	amount	numeric(12,2)	
*	summary	varchar(255)	
Indexes			
Pk	serviceExpenses_pkey	id	
Foreign Keys			
	serviceexpenses_serviceid_foreign (serviceId) ref services (id)		

Table serviceHandovers

* Pk	id	uuid	
*	serviceId	uuid	
	note	varchar(255)	
	creationDate	date	
Indexes			
Pk	serviceHandovers_pkey	id	
Foreign Keys			
	servicehandovers_serviceid_foreign (serviceId) ref services (id)		

Table servicePauses

* Pk	id	uuid	
*	serviceId	uuid	
*	dateRange	daterange	
Indexes			
Pk	servicePauses_pkey	id	
Foreign Keys			
	servicepauses_serviceid_foreign (serviceId) ref services (id)		
Constraints			
	nonEmptyDateRange	"dateRange" <> 'empty'::daterange	

Table serviceTasks

* Pk	id	uuid	
*	serviceId	uuid	
*	taskId	uuid	
Indexes			
Pk	serviceTasks_pkey	id	
Foreign Keys			
	servicetasks_serviceid_foreign (serviceId) ref services (id)		
	servicetasks_taskid_foreign (taskId) ref tasks (id)		

Table services

* Pk	id	uuid	
*	clientId	uuid	
*	genderPref	text	
*	pets	bool	
*	errands	bool	
	notes	text	
	visitGuaranteed	bool	
	languageProf	integer	
*	clientRating	numeric(8,2)	
*	clientLifetimeValue	numeric(13,2)	
Indexes			
Pk	services_pkey	id	
Foreign Keys			
	services_clientid_foreign (clientId) ref clients (id)		
Constraints			
	check_services_language_prof	("languageProf" >= 0) AND ("languageProf" <= 2)	
	services_genderPref_check	"genderPref" = ANY (ARRAY['m'::text, 'f'::text, 'n'::text])	

Table tasks

* Pk	id	uuid	
	name	varchar(255)	
Indexes			
Pk	tasks_pkey	id	

Table userIdentities

* Pk	id	uuid	
Indexes			
Pk	userIdentities_pkey	id	

Table users

* Pk	id	uuid	
* Unq	email	varchar(255)	
*	password	varchar(255)	
*	firstName	varchar(100)	
*	lastName	varchar(100)	
*	phoneNumber	varchar(15)	
Unq	facebookId	varchar(255)	
Indexes			
Pk	users_pkey	id	
Unq	users_email_unique	email	
Unq	users_facebookid_unique	facebookId	

Table visitPlans

* Pk	id	uuid	
*	serviceId	uuid	

Table visitPlans

*	dateRange	daterange	
*	timeWindow	timerange	
*	duration	interval	
*	recurrence	text	
	hourlyRateOverride	numeric(12,2)	
	comment	varchar(255)	
Indexes			
Pk	visitPlans_pkey	id	
Foreign Keys			
	visitplans_serviceid_foreign (serviceId) ref services (id)		
Constraints			
	timeWindowLargeEnough	duration <= (upper("timeWindow") - lower("timeWindow"))	
	nonZeroDuration	duration <> '00:00:00'::interval	
	nonEmptyTimeWindow	"timeWindow" <> 'empty'::timerange	
	nonEmptyDateRange	"dateRange" <> 'empty'::daterange	
	visitPlans_recurrance_check	recurrance = ANY (ARRAY['n'::text, 'w'::text, 'f'::text])	

Table workBlocks

* Pk	id	uuid	
*	butlerId	uuid	
* Idx	dateRange	daterange	
*	timeWindow	timerange	
Idx	(("butlerId")::text)	varchar	
Idx	date_part('dow'::text, lower("dateRange"))	varchar	
Indexes			
Pk	workBlocks_pkey	id	
	nonOverlappingDateRangesOnSameDay	(("butlerId")::text), date_part('dow'::text, lower("dateRange")), dateRange	
Foreign Keys			
	workblocks_butlerid_foreign (butlerId) ref butlers (id)		
Constraints			
	nonEmptyTimeWindow	"timeWindow" <> 'empty'::timerange	
	nonEmptyDateRange	"dateRange" <> 'empty'::daterange	