

Nexus Schema - Nov 13, 2013

class Client(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

name = models.CharField(max_length=64, blank=True, db_column='Name')

type = models.ForeignKey(ClientType, null=True, blank=True, db_column='Client Type')

notes = models.TextField(max_length=512, null=True, db_column='Notes')

license_requirement = models.ForeignKey(LicenseRequirement, null=True, db_column='License Requirement')

contacts = models.ManyToManyField('ClientContact', null=True, blank=True, db_column='Contacts',
related_name='client_contacts')

portfolios = models.ManyToManyField('Portfolio', null=True, blank=True, db_column='Portfolios')
```

class ClientContact(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

contact_info = models.ForeignKey(ContactInfo, db_column='Contact Info', related_name='client_contact_info')

preferred_report_delivery = models.TextField(max_length=512, null=True, db_column='Preferred Report Delivery')

invoice_delivery_instructions = models.TextField(max_length=512, null=True, db_column='Invoice Delivery Instructions')

client = models.ForeignKey('Client', null=True, blank=True, db_column='Client')

invoice_contact = models.ForeignKey(ContactInfo, null=True, blank=True, db_column='Invoice Contact')

hard_copies = models.IntegerField(null=True, db_column='Hard Copies')
```

class Portfolio(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

name = models.CharField(max_length=64, db_column='Name')

notes = models.TextField(max_length=512, null=True, db_column='Notes')

properties = models.ManyToManyField(Property, null=True, db_column='Properties')

contacts = models.ManyToManyField(ClientContact, null=True, blank=True, db_column='Contacts')
```

class Employee(models.Model):

```
created = models.DateTimeField(auto_now_add=True, editable=False)

changed = models.DateTimeField(auto_now=True, editable=False)

split = models.FloatField(null=True, db_column='Split')

salary = models.DecimalField(max_digits=19, decimal_places=2, null=True, db_column='Salary')

contact_info = models.ForeignKey(ContactInfo, null=True, blank=True, db_column='Contact Info')

office = models.ForeignKey('Office', null=True, db_column='Office')

manager = models.ForeignKey('Employee', null=True, blank=True, db_column='Manager')

user = models.ForeignKey(auth_models.User, related_name='employee_user', db_column='User')

employee_status = models.ForeignKey(EmployeeStatus, null=True, blank=True, db_column='Employee Status')

preferences = models.ManyToManyField('Preference', null=True, db_column='Preferences')

expenses = models.ManyToManyField('Expense', null=True, db_column='Expense')
```

class Preference(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

value = models.TextField(max_length=64, blank=True, db_column='Value')

key = models.ForeignKey(PreferenceKey, null=True, blank=True, db_column='Key')
```

class Office(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

name = models.CharField(max_length=64, null=True, db_column='Name')

contact_info = models.ForeignKey(ContactInfo, null=True, blank=True, db_column='Contact Info')
```

class Receipt(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

name = models.CharField(max_length=64, blank=True, db_column='Name')

notes = models.TextField(max_length=512, null=True, db_column='Notes')

expenses = models.ManyToManyField('Expense', null=True, blank=True, db_column='Expenses')
```

class Expense(models.Model):

```

created = models.DateTimeField(auto_now_add=True, editable=False)

changed = models.DateTimeField(auto_now=True, editable=False)

submittal_date = models.DateTimeField(null=True, db_column='Submittal Date', blank=True)

submitted = models.NullBooleanField(null=True, db_column='Submitted', blank=True)

description = models.CharField(max_length=128, db_column='Description', blank=True)

name = models.CharField(max_length=32, db_column='Name', blank=True)

amount = models.CharField(max_length=32, db_column='Amount', blank=True)

expense_type = models.ForeignKey(ExpenseType, blank=True, null=True, db_column='Expense Type')

submitted_by = models.ForeignKey('Employee', blank=True, null=True, db_column='Submitted By',
related_name='expense_submitted_by')

primary_appraiser = models.ForeignKey('Employee', blank=True, null=True, db_column='Primary Appraiser')

```

class Job(models.Model):

```

created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

due_date = models.DateField(null=True, blank=True, db_column='Due Date')

legacy = models.BooleanField(blank=True, db_column='Legacy', default=False)

job_number = models.CharField(max_length=64, db_column='Job Number')

amount_due = models.CharField(max_length=32, db_column='Amount Due')

repeats = models.CharField(max_length=64, blank=True, db_column='Repeats')

repeats_when = models.CharField(max_length=32, blank=True, db_column='Repeats When')

notes = models.TextField(max_length=512, null=True, db_column='Notes')

client = models.ForeignKey(Client, null=True, blank=True, db_column='Client')

portfolio = models.ForeignKey(Portfolio, null=True, blank=True, db_column='Portfolio')

property = models.ForeignKey(Property, null=True, blank=True, db_column='Property')

factsheet = models.ForeignKey('FactSheet', null=True, blank=True, db_column='Fact Sheet')

job_status = models.ForeignKey(JobStatus, null=True, db_column='Job Status')

invoice = models.ForeignKey('Invoice', null=True, blank=True, db_column='Invoice')

roles = models.ManyToManyField('Role', null=True, blank=True, db_column='Roles')

documents = models.ManyToManyField('Document', null=True, blank=True, db_column='Documents')

receipts = models.ManyToManyField(Receipt, null=True, blank=True, db_column='Receipts')

```

class FactSheet(models.Model):

```

created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

date_inspection = models.DateField(null=True, db_column='Date Inspection')

date_value = models.DateField(null=True, db_column='Date Value')

percent_gross_leasable_area = models.DecimalField(max_digits=5, decimal_places=2, db_column='Percent Gross Leasable Area')
# % of GLA (scraped from excel)

gross_leasable_area = models.DecimalField(max_digits=12, decimal_places=2, null=True, db_column='Gross Leasable Area') # Not
scraped from excel

market_value_as_is = models.DecimalField(max_digits=16, decimal_places=2, null=True, db_column='Market Value As Is')

exposure_time = models.CharField(max_length=64, null=True, db_column='Exposure Time')

marketing_time = models.CharField(max_length=64, null=True, db_column='Marketing Time')

zoning = models.CharField(max_length=64, null=True, db_column='Zoning')

intended_user = models.CharField(max_length=64, null=True, db_column='Intended User')

tax_parcel = models.CharField(max_length=64, null=True, db_column='Tax Parcel')

intended_use = models.TextField(null=True, db_column='Intended Use')

area = models.TextField(null=True, db_column='Area') # square feet

improvements = models.TextField(null=True, db_column='Improvements')

best_use_vacant = models.TextField(null=True, db_column='Best Use Vacant')

best_use_improved = models.TextField(null=True, db_column='Best Use Improved')

property_owner = models.ForeignKey(ContactInfo, null=True, blank=True, db_column='Property Owner')

tenant = models.ForeignKey(Tenant, null=True, blank=True, db_column='Tenant')

```

class Role(models.Model):

```

created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

fee = models.DecimalField(max_digits=8, decimal_places=2, null=True, blank=True, db_column='Fee')

name = models.CharField(max_length=64, db_column='Name')

employee = models.ForeignKey(Employee, db_column='Employee')

role_type = models.ForeignKey(RoleType, null=True, blank=True, db_column='Role Type')

```

class Invoice(models.Model):

```

created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

```

```
name = models.CharField(max_length=64, db_column='Name')

notes = models.TextField(max_length=512, null=True, db_column='Notes')
```

class Property(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

name = models.CharField(max_length=128, db_column='Name')

cid1 = models.CharField(max_length=128, null=True, blank=True, db_column='CID1')

cid2 = models.CharField(max_length=128, null=True, blank=True, db_column='CID2')

notes = models.TextField(max_length=512, null=True, db_column='Notes')

address = models.ForeignKey(Address, null=True, blank=True,)

type = models.ForeignKey(PropertyType, null=True, blank=True, related_name='property_type',
                           db_column='Property Type')

subtype = models.ForeignKey(PropertySubtype, null=True, blank=True, related_name='property_subtype',
                             db_column='Property Subtype')
```

class Tenant(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

name = models.CharField(max_length=64, db_column='Name')

contact_info = models.ForeignKey(ContactInfo, null=True, blank=True, db_column='Contact Info')

tenancy_type = models.ForeignKey(TenancyType, null=True, blank=True, db_column='Tenancy Type')
```

class ZipCode(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

city = models.CharField(max_length=64, null=True, blank=True, db_column='City')

zipcode = models.CharField(max_length=32, null=True, blank=True, db_column='Zip Code')

areacode = models.CharField(max_length=32, null=True, blank=True, db_column='Area Code')

fips = models.CharField(max_length=32, null=True, blank=True, db_column='FIPS')

county = models.CharField(max_length=32, null=True, blank=True, db_column='County')

timezone = models.CharField(max_length=32, null=True, blank=True, db_column='Time Zone')
```

```
dst = models.CharField(max_length=64, null=True, blank=True, db_column='Daylight Savings Time')

latitude = models.CharField(max_length=32, null=True, blank=True, db_column='Latitude')

longitude = models.CharField(max_length=32, null=True, blank=True, db_column='Longitude')

dtype = models.CharField(max_length=32, null=True, blank=True, db_column='DType')

state = models.ForeignKey('State', null=True, db_column='State')

google_address = map_fields.AddressField(max_length=200)

geolocation = map_fields.GeoLocationField(max_length=100)
```

class State(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

name = models.CharField(max_length=8, blank=True, db_column='Name')

full_name = models.CharField(max_length=32, blank=True, db_column='Full Name')

license_requirement = models.ForeignKey('LicenseRequirement', blank=True, null=True, db_column='License Requirement')
```

class Address(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

address1 = models.CharField(max_length=512, null=True, blank=True, db_column='Address1')

address2 = models.CharField(max_length=512, null=True, blank=True, db_column='Address2')

city = models.CharField(max_length=512, null=True, blank=True, db_column='City')

state = models.CharField(max_length=32, null=True, blank=True, db_column='State')

county = models.CharField(max_length=32, null=True, blank=True, db_column='County')

latitude = models.CharField(max_length=32, null=True, blank=True, db_column='Latitude')

longitude = models.CharField(max_length=32, null=True, blank=True, db_column='Longitude')

zipcode = models.ForeignKey('ZipCode', null=True, db_column='Zip Code')

google_address = map_fields.AddressField(max_length=200)

geolocation = map_fields.GeoLocationField(max_length=100)
```

class ContactInfo(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)
```

```
first_name = models.CharField(max_length=128, null=True, blank=True, db_column='First Name')
last_name = models.CharField(max_length=128, null=True, blank=True, db_column='Last Name')
email = models.CharField(max_length=128, null=True, blank=True, db_column='Email')
phone = models.CharField(max_length=64, null=True, blank=True, db_column='Phone')
mobile = models.CharField(max_length=64, null=True, blank=True, db_column='Mobile')
fax = models.CharField(max_length=64, null=True, blank=True, db_column='Fax')
title = models.CharField(max_length=128, null=True, blank=True, db_column='Title')
initials = models.CharField(max_length=64, null=True, blank=True, db_column='Initials')
address = models.ForeignKey('Address', null=True, blank=True, db_column='Address')
```

class Document(models.Model):

```
created = models.DateTimeField(auto_now_add=True)
changed = models.DateTimeField(auto_now=True)
file_name = models.CharField(max_length=64, blank=True, db_column='File Name')
type = models.ForeignKey(DocumentType, null=True, blank=True, db_column='DocumentType')
```

class License(models.Model):

```
created = models.DateTimeField(auto_now_add=True)
changed = models.DateTimeField(auto_now=True)
name = models.CharField(max_length=64, blank=True, db_column='Name')
state = models.ForeignKey('State', null=True, blank=True, db_column='State')
license_type = models.ForeignKey(LicenseType, null=True, blank=True, db_column='License Type')
```

class LicenseRequirement(models.Model):

```
created = models.DateTimeField(auto_now_add=True)
changed = models.DateTimeField(auto_now=True)
reviewers_sign = models.NullBooleanField(db_column='Reviewers Sign')
appraisers_sign = models.NullBooleanField(db_column='Appraisers Sign')
license_reviewers = models.NullBooleanField(db_column='License Reviewer')
license_appraisers = models.NullBooleanField(db_column='License Appraisers')
resource_link = models.CharField(max_length=256, blank=True, db_column='Resource Link')
notes = models.TextField(max_length=512, null=True, db_column='Notes')
```

class ClientType(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

type = models.CharField(max_length=32, blank=True, db_column='Type')
```

class EmployeeStatus(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

status = models.CharField(max_length=32, blank=True, db_column='Status')
```

class PreferenceKey(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

key = models.CharField(max_length=64, blank=True, db_column='Key')
```

class JobStatus(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

status = models.CharField(max_length=64, blank=True, db_column='Status')
```

class DocumentType(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

type = models.CharField(max_length=64, db_column='Type')
```

class RoleType(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

type = models.CharField(max_length=64, blank=True, db_column='Type')
```

class PropertyType(models.Model):


```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

type = models.CharField(max_length=64, blank=True, db_column='Type')

subtype = models.ManyToManyField('PropertySubtype', null=True, blank=True, db_column='Subtype')
```

class PropertySubtype(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

type = models.CharField(max_length=64, blank=True, db_column='type')
```

class TenancyType(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

type = models.CharField(max_length=64, blank=True, db_column='Type')
```

class LicenseType(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

type = models.CharField(max_length=64, blank=True, db_column='Type')
```

class ExpenseType(models.Model):

```
created = models.DateTimeField(auto_now_add=True)

changed = models.DateTimeField(auto_now=True)

type = models.CharField(max_length=64, blank=True, db_column='Type')
```