**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')