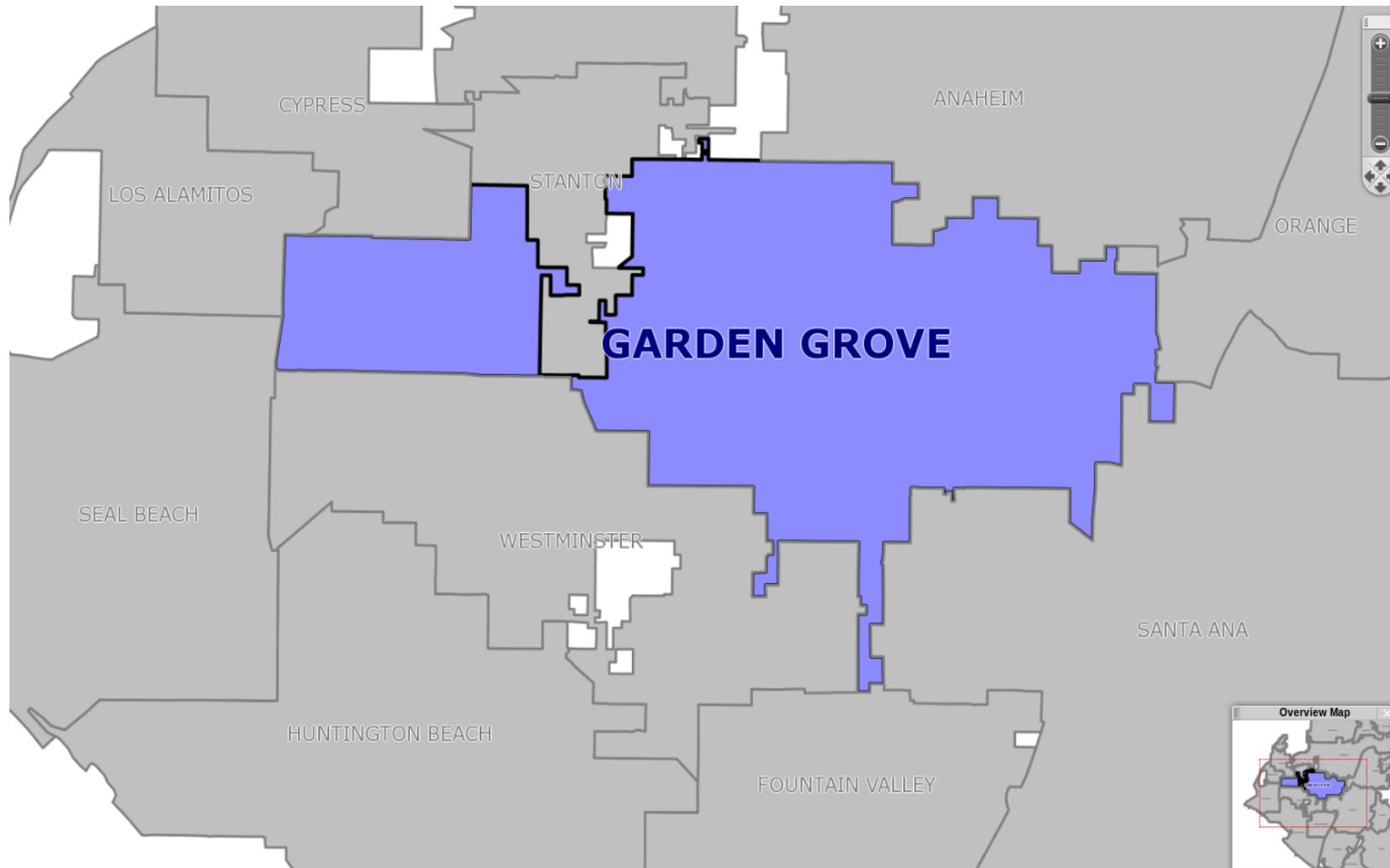


Use of PostgreSQL in Municipal Government

City of Garden Grove, CA

City of Garden Grove

Founded 1874, incorporated 1956.
Population 174,715 (2009)



City of Garden Grove

Direct Services

- Police
- Fire
- Public Works
 - Water and Sewer
 - Storm Drainage
 - Engineering
 - Street Maintenance
 - Parks and Trees
 - Environmental Compliance
- Planning and Permitting
- Redevelopment
- Housing Authority
- Recreation

2010/2011 Budget

- \$87.8 million general
- \$209 million overall



City of Garden Grove

~900 users

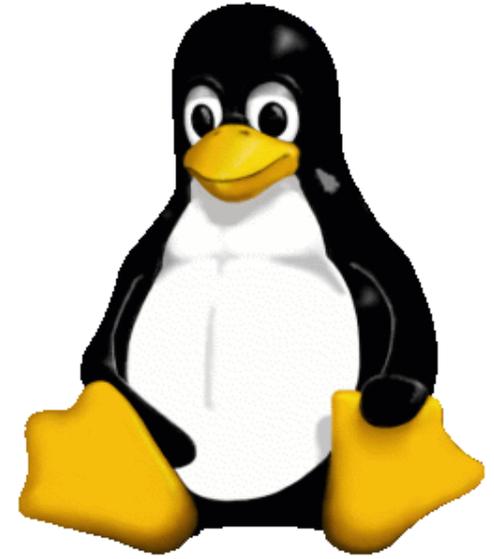
450 desktops

70 mobile computers

Linux primary server OS since 1995

Began using PostgreSQL 7.1 in 2001

Information Technology Budget: 2.1 million



City Server OS History

- Use of computing for finance date back to 1960s
- Pick - database / application environment
 - 1977 Microdata Reality : 24 users
 - 1984 Honeywell/Ultimate : 100 serial devices.
 - 1990 Data General Quad Processor Mini
 - \$400,000 (1990 dollars)
 - ~ 400 users, dumb terminals
 - DGUX/Advanced Pick
 - 1994 x86/SCO
 - Ethernet and TPC/IP
 - 1995 Linux



How PostgreSQL fits in

City database instances in 2000

Brand	Servers	Applications
Pick	2	Accounting, Budget, Payroll, Business License, Building Permits, Utilities , HR, Fire Permits, Police Records, Housing Authority, Public Works, Code Enforcement, Substandard Housing, Land Use, Street Lighting, Trees, Warehouse Inventory and more.
Oracle	2	Computer aided dispatch (CAD)

How PostgreSQL fits in

City database instances in 2011

Brand	Servers	Applications
Pick	1	Accounting, Budget, Payroll, Business License
Oracle	3	computer aided dispatch (CAD), computer aided drawing (CAD) utility network
MS-SQL	3	Jail management, property and evidence, public meeting video and minutes
MySQL	2	OpenAudit, Zimbra
SQLite	1	Wikis
FileMaker-Pro	2	Housing authority, In-car video library
4D	1	Fire inspection, hazardous waste tracking, training
PostgreSQL	5	Police Records Management, Water Billing, Recreation, Code Enforcement, Public Works Work-orders, Environmental Compliance, Utility Maintenance, Time-sheets, Cashiering, Graffiti Tracking, In-car video (2), RT, Drupal, Spam, Passports, Building Permits, Engineering Permits, Scalix, GIS, Location Management/Land Use, Risk Management, Munciple Code, Snort, Document Archiving, and more.

Why PostgreSQL?

City had been using "multi-value" Pick database since 1970s.

In 2001...

MySQL didn't have sub-selects.

DB2 was complicated and didn't have TCL or PHP clients.

Oracle was too laborious.

PostgreSQL

- Free
- Fast
- Easy to install
- Easy to backup
- Worked with many languages
- Good Documentation
- Reliable

Why PostgreSQL?

In 2001 PostgreSQL was missing a few features:

ALTER TABLE .. DROP COLUMN -- added in 7.3, 2002

SCHEMAS -- added in 7.3

ALTER COLUMN TYPE -- added in 8.0, 2005

How PostgreSQL fits in

Commercial Applications - Postgres in the wild

CanIT-Pro : Spam filter

Scalix : Exchange replacement (being replaced)

PermitCity : Building permitting and inspection system

L3 Communication's in-car video system

MapGuide Enterprise : Web based GIS server

How PostgreSQL fits in

Open Source Applications - Postgres as a choice

Drupal : External websites

- some effort required to select modules to work

Squidguard : Web logging

- used after MySQL index corruption

RT : Request Tracker

- Entity attribute model

Snort : Intrusion Detection System

How PostgreSQL fits in

Noteworthy in-house PG applications

Document Archiving : Agreements, Resolutions, Ordinances, Minutes, Deeds, Building Plans, ect.

- PG's full text search

Water : Water, sewer, billing, accounting, and cross connection systems

- 34,000 + services
- ~ 30 million in billings a year
- 12 years of transaction data online
- extensive use of Pg functions, triggers,

Recreation : Facility and class booking application

- Complex scheduling handled through PG
- Written such that other agencies could use

How PostgreSQL fits in

Noteworthy in-house PG applications

Public Works Workorders

- used for all aspects of operations
 - facilities, streets, water distribution, sewer, environmental compliance, signs, trees, graffiti, ect.
- 270,000 workorders
- tracks labor and materials
- Alerts via SMS

Police Records Management System

- events, arrests, people, vehicles, cites, TC,FI
- DNA, calls for service, casefiles, ect.

Municipal Code

- paginated and online output

First uses of PostgreSQL

2001/2002

Homicide case management system

Sewer overflow tracking system

Utility maintenance tracking

- TCL / cgi : bad idea #1
- In-house web application framework : bad idea #2
- Images stored as blobs in PG : bad idea #3
- Paginated reporting using LaTeX : bad idea #4
- Separate PG databases for PD and City : bad idea #5

First uses of PostgreSQL

2003

Evidence, FI, Graffiti Photos

- 100 gigs of blobs
 - interesting to backup
 - worked fine
 - no referential integrity
- Converted to BYTEA
 - takes more space
 - still hard to deal with backups
- Moved images out of database in 2005
 - Now 650GB

Ruby on Rails and PostgreSQL

- Started using in 2007
- Easy to start using
- Takes time to master

 **Code Enforcement System** Thursday, February 2, 2010
[Home](#) | [Search](#) | [Logout](#)

CASE# 141067

Actions: [Re-open](#) | [Print](#)

10131 IMPERIAL AVE - R1

[Reminders](#) | [Other cases at this address](#) | [Create a new case](#)

[Report](#)

[Inspections](#)

[Notice Letters](#)

[Photos](#)

[Documents](#)

[Invoices](#)

[Comments](#)

Reactive Report

Inspector: Rita Cramer **RD#** 105

Date/Time: 07/15/09 - 08:39 AM **N.O.V Date:** 02/04/10

Reinspection: **Closed Date:** 03/25/10

COMPLAINING PARTIES

Name: Anonymous **Phone:** None.

VIOLATION CODE:

01 Inoperative Vehicle

04.1 Storage of Debris

09.1 Living in a Vehicle

27 Yard Setback

```

1 class Customer < ActiveRecord::Base
2   validates_presence_of :name, :address, :city, :state, :zip, :dl, :phones
3   has_many :customer_views
4   has_many :accounts
5   has_many :echecks
6   has_and_belongs_to_many :services
7
8   validates_length_of :state, :is => 2
9
10  def before_save
11    unless self.legacy_id.nil?
12      if self.legacy_id.empty?
13        self.legacy_id = nil
14      end
15    end
16    self.address.upcase!
17  end
18 end

```

Model

```

app/models/customer.rb 16,5 All
54   - if @credit_history
55     %b= @credit_history.size.to_s + " credit issues"
56   = text_field_tag :deposit, number_with_precision(@deposit,:precision =>2), :size =>
57   = label_tag :deposit
58 %p
59 Check one
60 %br
61 = f.check_box :owner
62 = f.label :owner, "owns the property"
63 %br
64 = f.check_box :renter
65 = f.label :renter, "rents or leases"
66 %p
67 = label_tag :open_on
68 %br
69 = calendar_date_select_tag 'open_on', ((@off_order) ? @off_order.order_on : nil), :
70 leave blank to open now. Dates must be in the future.
71 %p
72 = label_tag :remark
73 %br
74 = text_area_tag :remark, nil, :cols => 80, :rows => 2
75 %br
76 %p
77 = f.submit "Create Account"

```

View

```

app/views/accounts/_form.html.haml 77,5 Bot
"app/models/customer.rb" 18L, 418C

```

```
1 class EchecksController < ApplicationController
2   before_filter :require_admin, :except => [:new, :create, :pay, :pay_check, :destroy, :
  show]
3   protect_from_forgery :except => [:pay_check]
4   before_filter :find_echeck
5   layout 'guest'
6
7   ECHECKS_PER_PAGE = 20
8   MAXOVER_BALANCE = 200
9   TIMES_BALANCE = 3
10
11  def create
12    @echeck = Echeck.new(params[:echeck])
13    customer = Customer.find(session[:guest_id])
14    @echeck.customer_id = customer.id
15    respond_to do |format|
16      if @echeck.save
17        flash[:notice] = 'Account was successfully added.'
18        format.html { redirect_to :action=>:pay }
19        format.xml { render :xml => @echeck, :status => :created, :location => @echeck
20      }
21      else
22        format.html { render :action => "new" }
23        format.xml { render :xml => @echeck.errors, :status => :unprocessable_entity }
24      end
25    end
26  end
27 end
```

app/controllers/echecks_controller.rb 4,3 Top

```
57
58  def laterals
59    @material = params[:material]
60    if @material
61      where = "AND lateral_material = E'#{@material}'"
62    else
63      where = ""
64    end
65    bks = Service.find_by_sql("SELECT DISTINCT substr(id,1,2) AS book FROM services ORD
66  ER BY book")
67    @books = Array.new
68    bks.each do |b|
69      @books[b.book.to_i] = Service.find(:all, :conditions => ["substr(id,1,2) = ? #{wh
70  ere}", b.book])
71    end
72    respond_to do |format|
73      format.html
74      format.pdf {
75        render :pdf => 'laterals', :stylesheets => 'lateral'
76      }
77    end
78  end
79  def laterals_stats
```

app/controllers/services_controller.rb 75,7 28%

"app/controllers/echecks_controller.rb" 223L, 7055C written

Controller

Controller

Ruby on Rails and PostgreSQL

- RoR treats DB as a bit bucket (somewhat)
 - You don't have to
 - Models can be views or setof
 - Constraints do not hurt RoR
 - Do counting, sorting, and math in PG
- CPKs
 - Needs plugin
 - Can break REST unless you deal with routes
 - surrender and add SPKs as needed
- Be aware of caching
 - mostly, don't worry
 - can be an issue on same page render while DB data is changing
 - e.g temp table
 - Use "uncached do" block to avoid

Ruby on Rails and PostgreSQL

CASE# 149717

12345 PostgreSQL St

[Go back to case](#)

Notice Letter

Name

[Pull Information](#)

Font family ▾ Font size ▾ Styles ▾ **B** *I* U | [List Icons] | [Undo] [Redo] | ABC ▾ | HTML

February 17, 2011

BRUCE WAYNE
4212 WAYNE MANSION LN
GOTHAM CITY, 21231

Subject: **Municipal Code Violations at 12345 PostgreSQL St**

The City of Garden Grove is committed to working with citizens in a joint effort to preserve and improve the residential neighborhoods. The proper maintenance of homes and neighborhoods will allow all residents to live in a quality environment and will also protect the value of your home. The City has received a complaint concerning your property or has observed the following problem(s):

The Garden Grove Municipal Code does not allow inoperative vehicles to be parked on driveways or in yard areas. An inoperative vehicle is one that has flat tires, engine failure, electrical malfunction, or missing parts. Inoperative vehicles and vehicle parts may only be stored in a fully enclosed garage. Vehicle repairs must be done inside the garage. You are hereby required to repair all inoperative vehicles or move the vehicle and auto parts to a fully enclosed garage. (GGMC 9.32.170(B))

To assure compliance with the Garden Grove Municipal Code, corrective measures should begin now and must be completed within **10 day(s)** from the date of this notice. Failure to correct the above violation(s) may result in issuance of a misdemeanor citation or referral to the City Attorney for

PostGIS: Underground Service Alert

