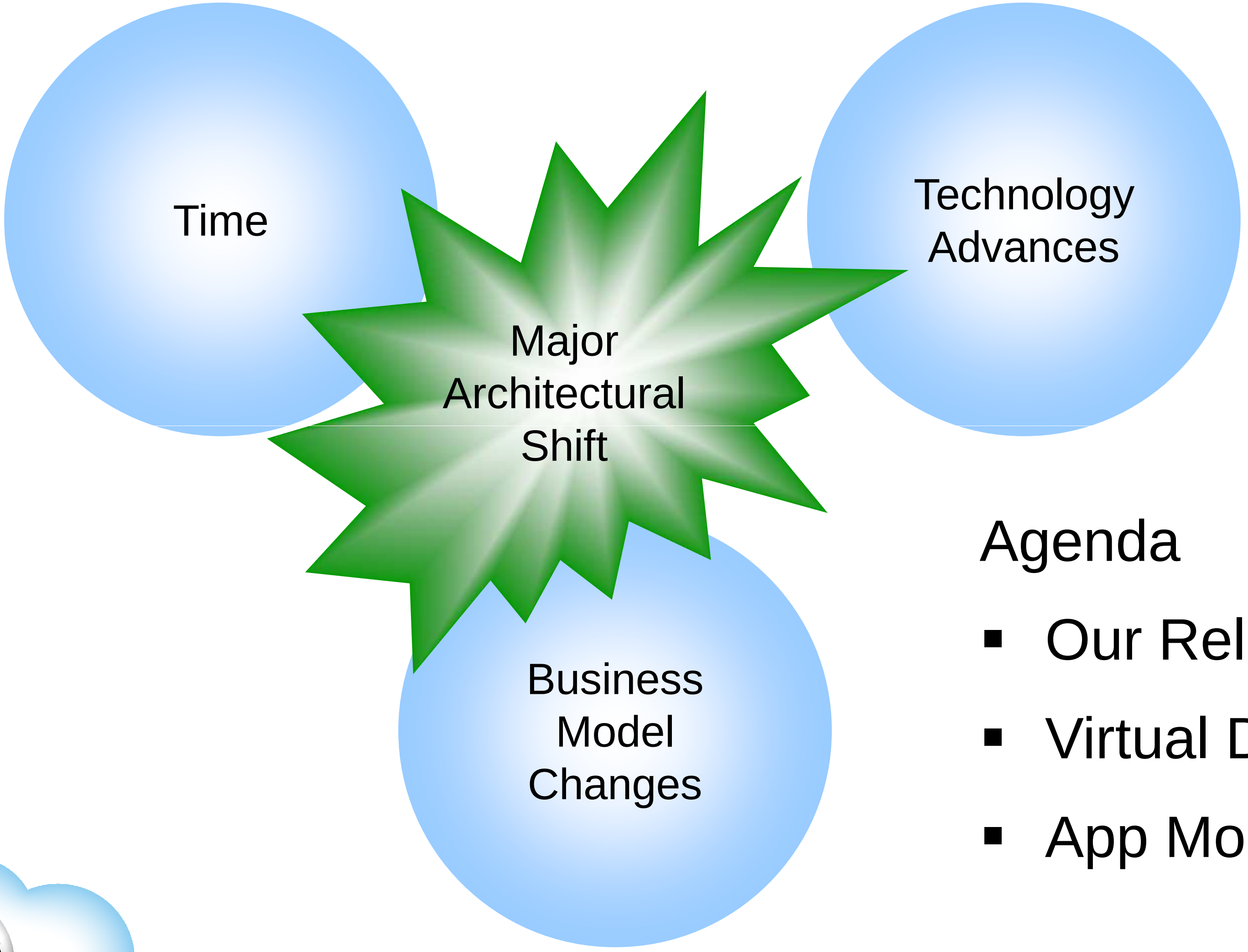
The Internal Design of Force.com's Multi-Tenant Architecture

Craig Weissman, Chief Software Architect
salesforce.com

True Multi-Tenancy is our Religion



Time

Technology Advances

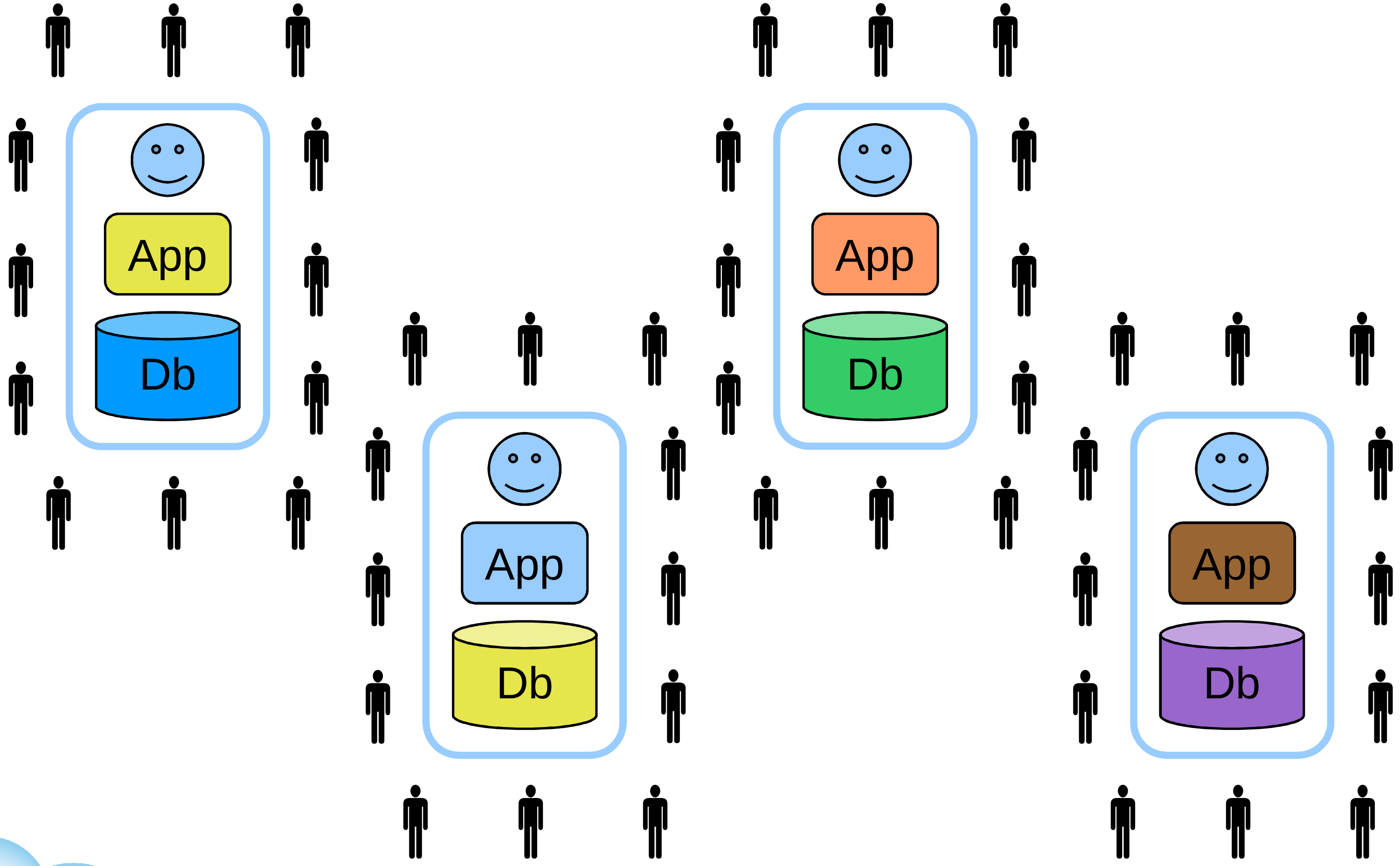
Major Architectural Shift

Business Model Changes

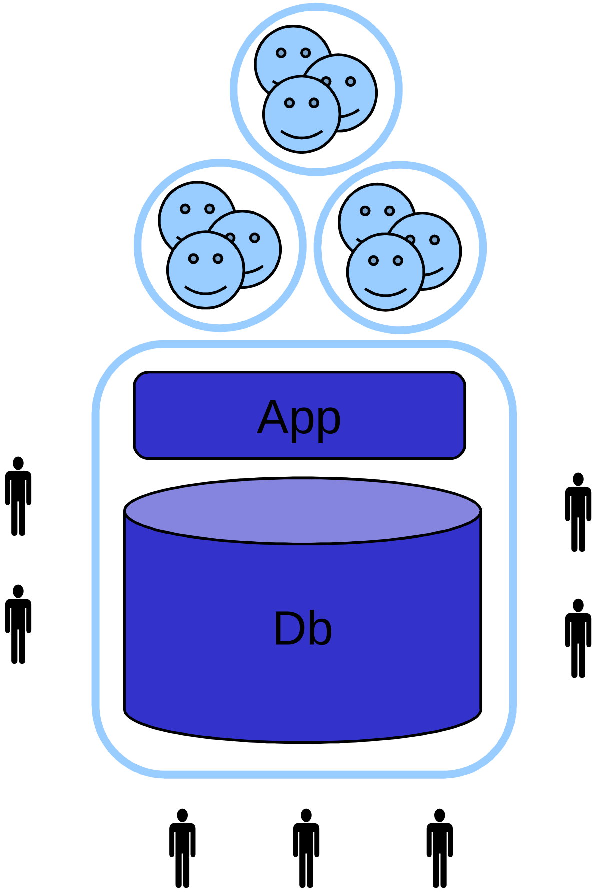
Agenda

- Our Religion
- Virtual Database
- App Model

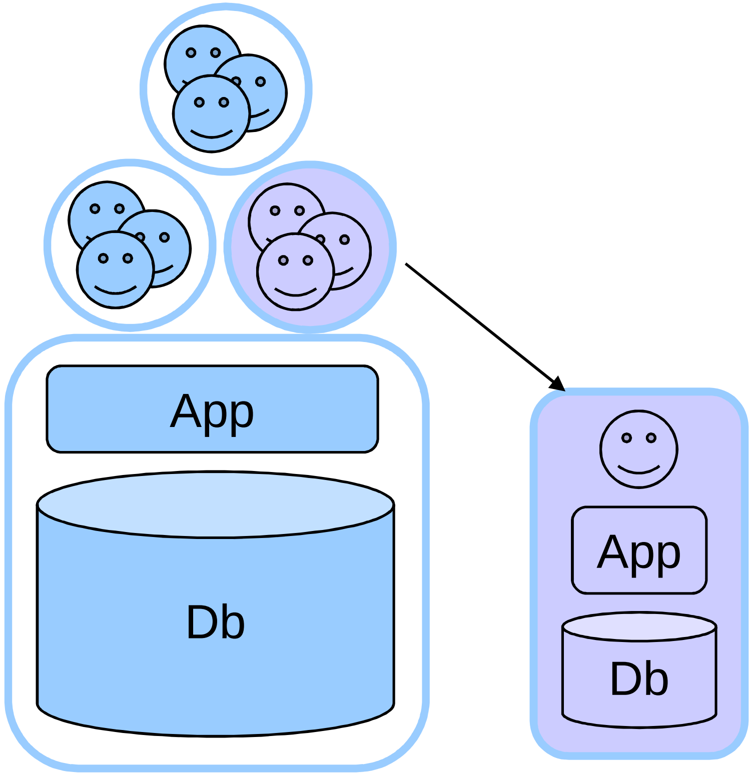
Single tenant applications: lots of waste



Multi-tenancy benefits are self-evident
But isolation is much easier said than done…

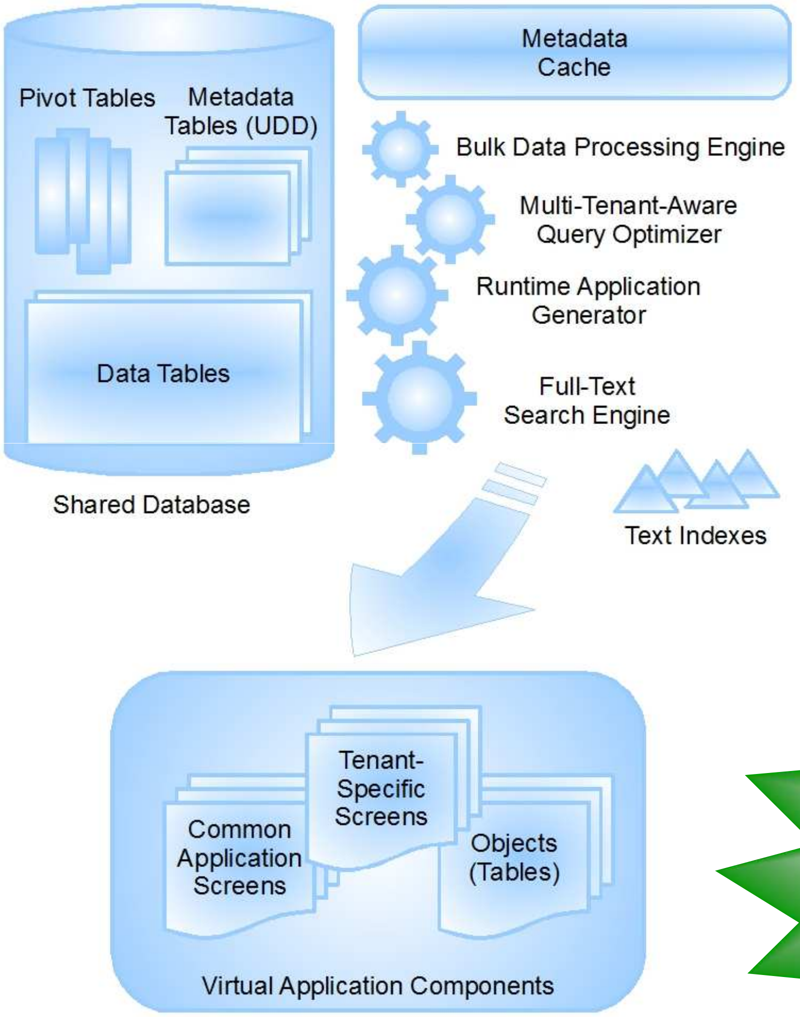


Our religion:
Not all "multi-tenant" designs are created equal



"Can't we create a separate stack for just this one customer? I promise it's just this one…"

Introducing the Force.com metadata-driven, multi-tenant, Internet application platform



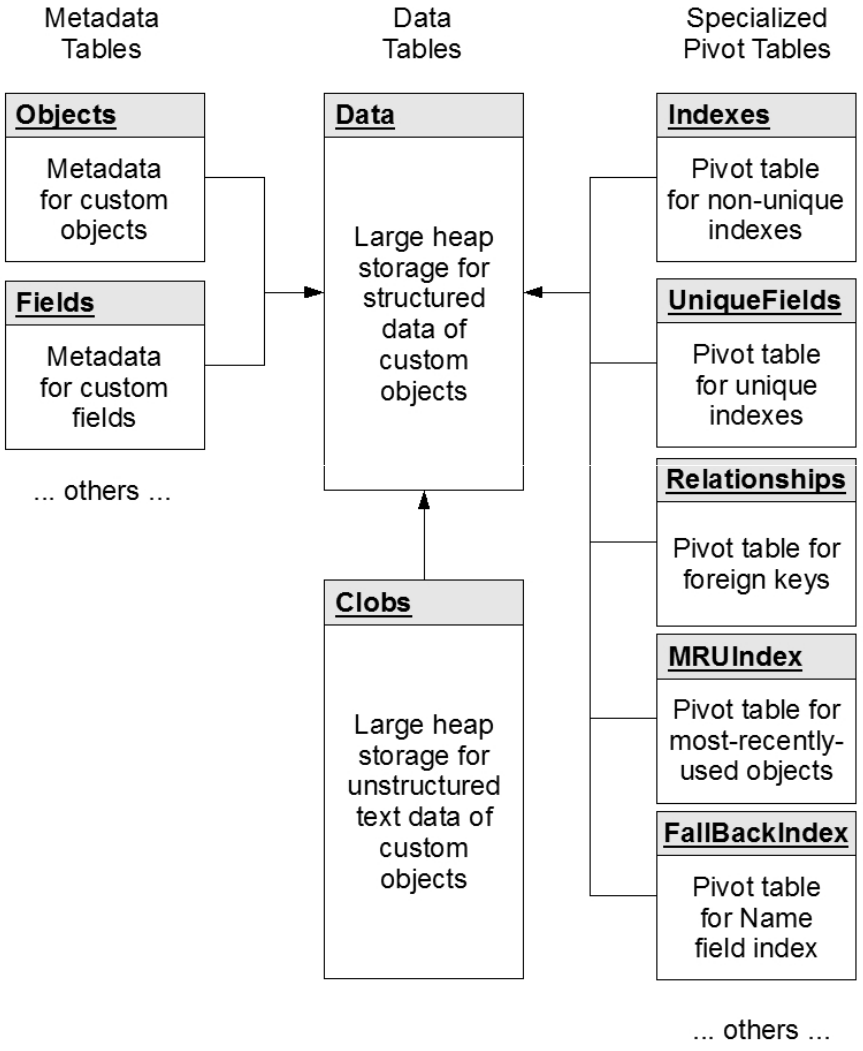
Poly-Morphic Application

SOFTWARE

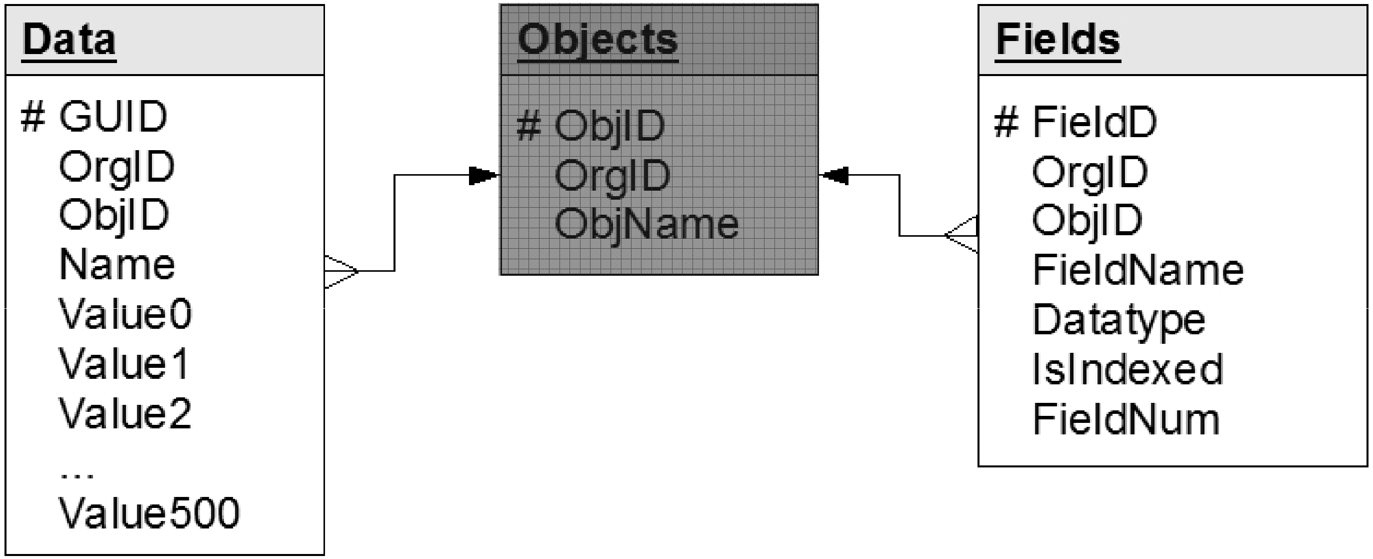
Key Architectural Principles

- Stateless AppServers
- Database system of record
- No DDL
- All tables partitioned by OrgId
- Smart PKs, Polymorphic FKs
- Creative de-normalization and pivoting
- Use every RDBMS feature/trick

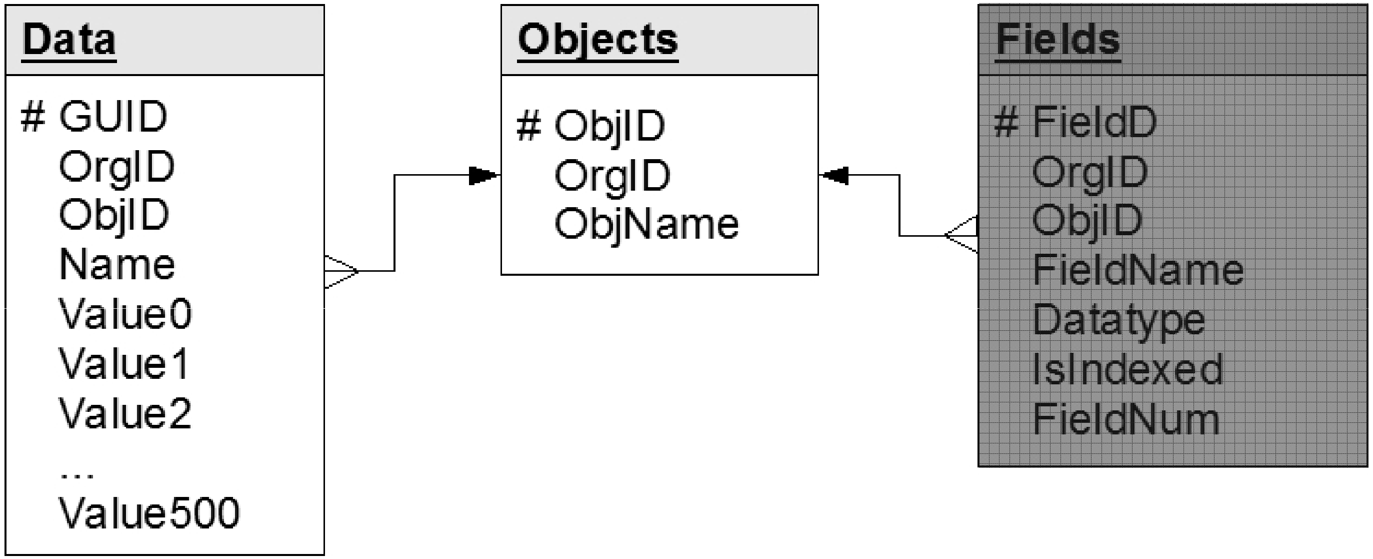
Metadata, data, and pivot table structures store data corresponding to virtual data structures


Metadata Tables
Data Tables
Specialized Pivot Tables
Objects
Metadata for custom objects
Fields
Metadata for custom fields
... others ...
Data
Large heap storage for structured data of custom objects
Clobs
Large heap storage for unstructured text data of custom objects
Indexes
Pivot table for non-unique indexes
UniqueFields
Pivot table for unique indexes
Relationships
Pivot table for foreign keys
MRUIndex
Pivot table for most-recently-used objects
FallBackIndex
Pivot table for Name field index
... others ...

The Objects table stores metadata about custom objects (tables)



The Fields table stores metadata about custom fields (columns)



Data

- # GUID
- OrgID
- ObjID
- Name
- Value0
- Value1
- Value2
- ...
- Value500

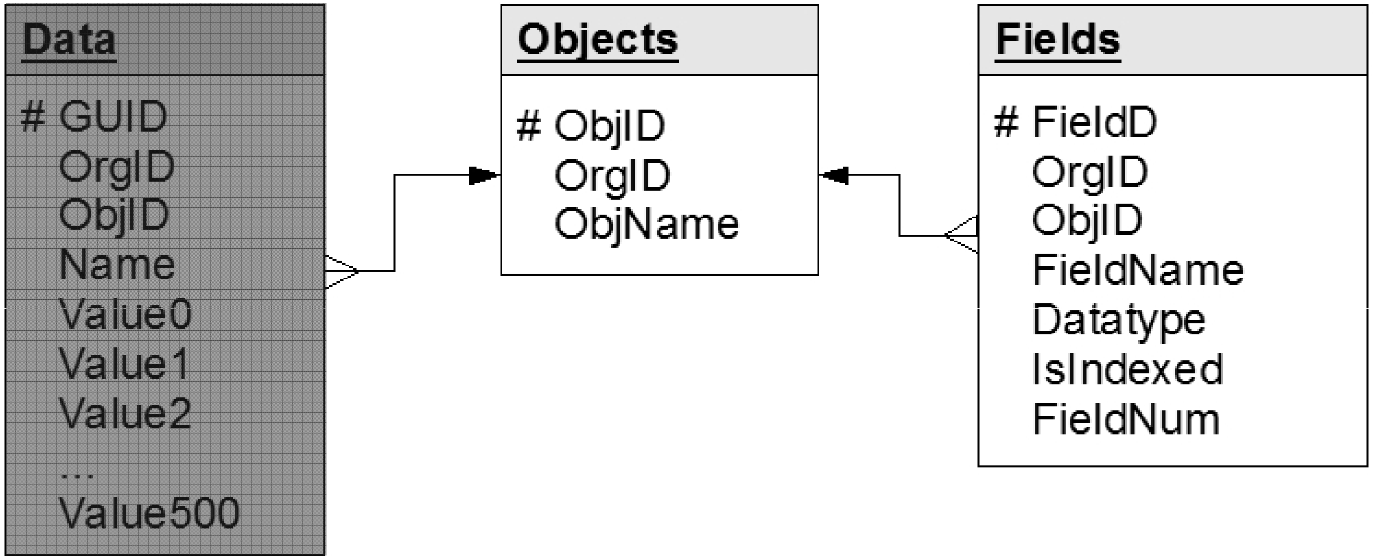
Objects

- # ObjID
- OrgID
- ObjName

Fields

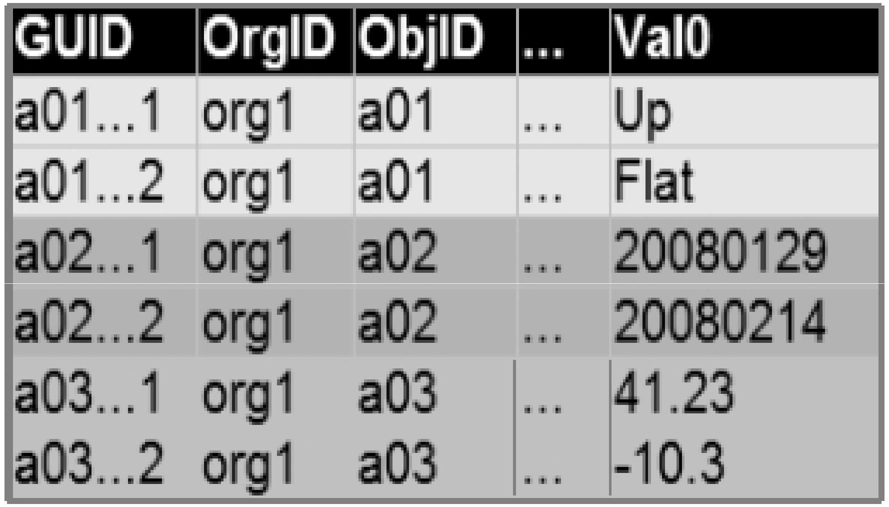
- # FieldD
- OrgID
- ObjID
- FieldName
- Datatype
- IsIndexed
- FieldNum

The Data heap table stores all structured data corresponding to custom objects



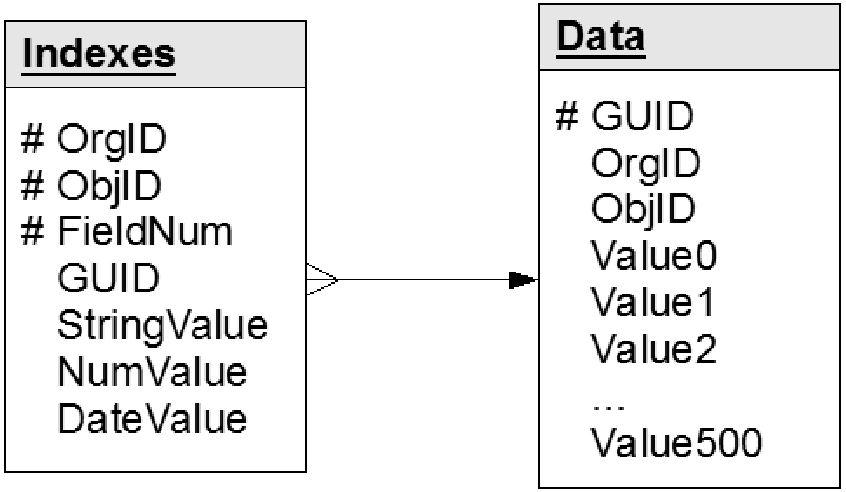
A single slot can store various types of data that originate from different objects

Data

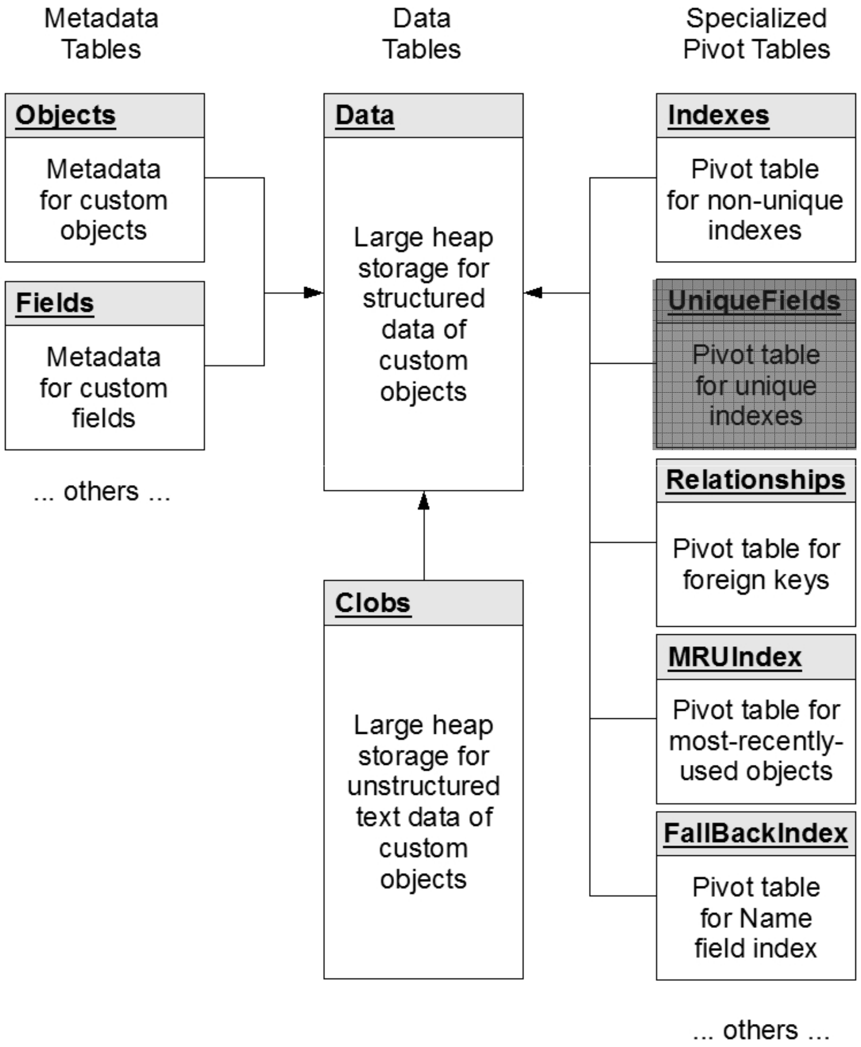


GUID	OrgID	ObjID	...	Val0
a01...1	org1	a01	...	Up
a01...2	org1	a01	...	Flat
a02...1	org1	a02	...	20080129
a02...2	org1	a02	...	20080214
a03...1	org1	a03	...	41.23
a03...2	org1	a03	...	-10.3

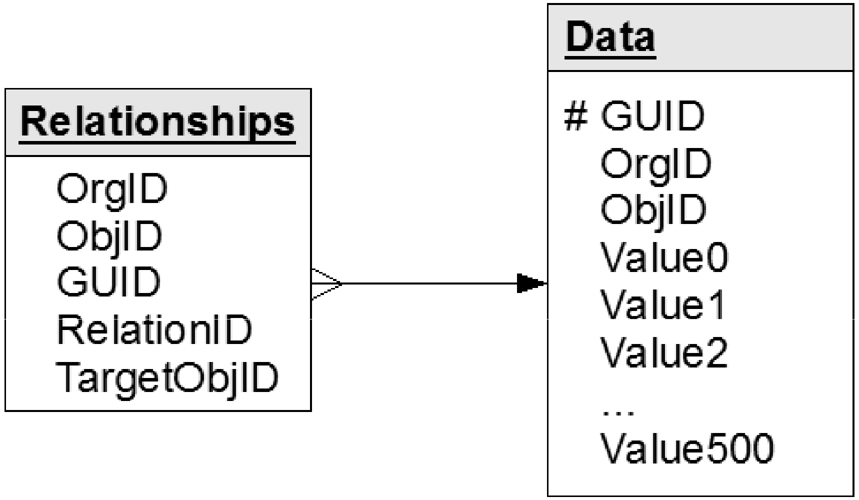
The Indexes pivot table manages tenant-specific selective indexes



The UniqueFields pivot table facilitates uniqueness for custom fields


Metadata Tables
Data Tables
Specialized Pivot Tables
Objects
Metadata for custom objects
Fields
Metadata for custom fields
... others ...
Data
Large heap storage for structured data of custom objects
Clobs
Large heap storage for unstructured text data of custom objects
Indexes
Pivot table for non-unique indexes
UniqueFields
Pivot table for unique indexes
Relationships
Pivot table for foreign keys
MRUIndex
Pivot table for most-recently-used objects
FallBackIndex
Pivot table for Name field index
... others ...

The Relationships pivot table facilitates referential integrity and optimizes joins



Relationships

OrgID
ObjID
GUID
RelationID
TargetObjID

Data

# GUID
OrgID
ObjID
Value0
Value1
Value2
...
Value500

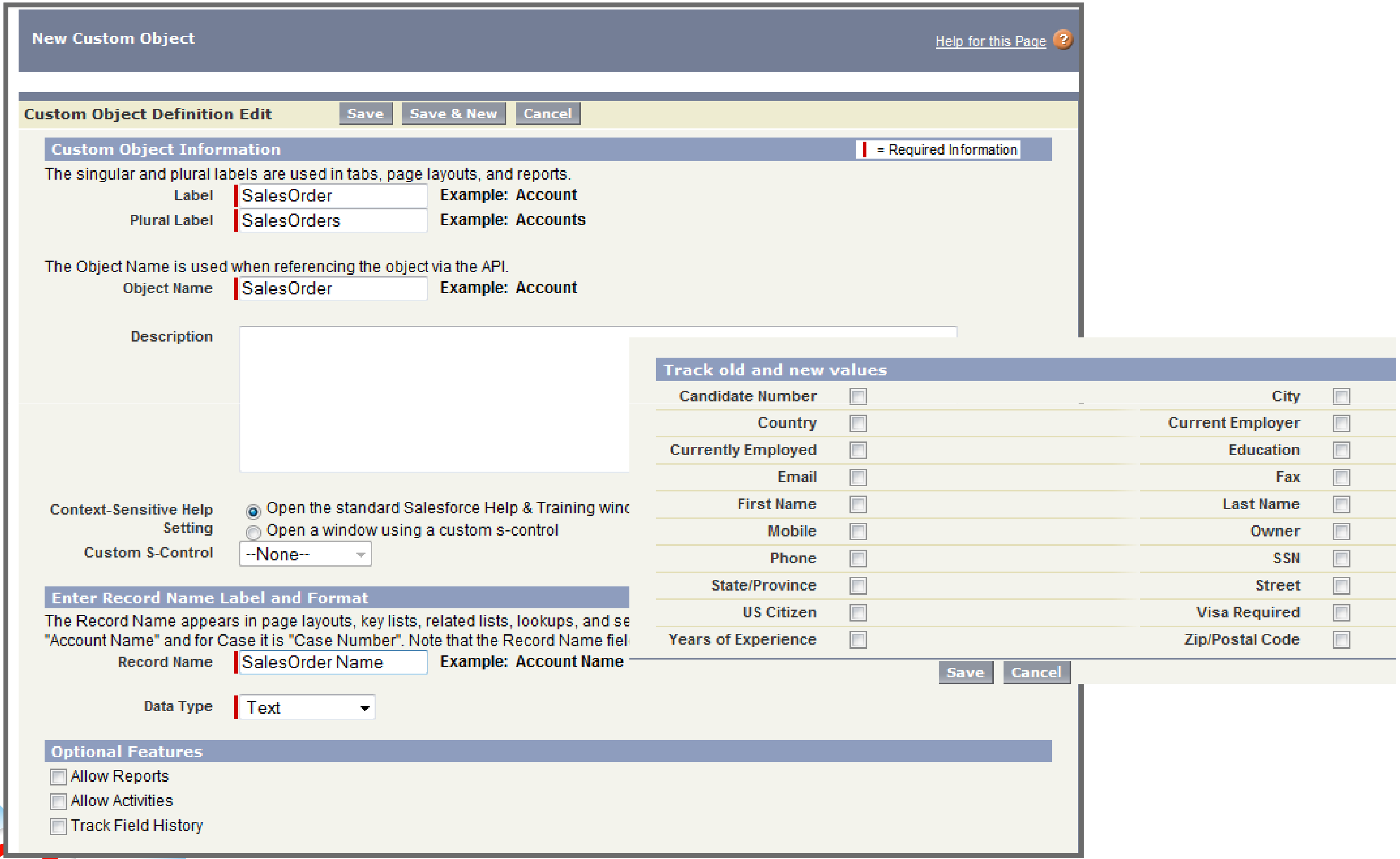
All data & metadata structures are partitioned to improve performance and manageability

- Tables hash partitioned by OrgId
- Separate conn pools point to physical hosts
- App tier is also dynamically partitioned by OrgId
- Distributed metadata cache w/transactional invalidation

Application Framework: a whole lot for free

- Native Declarative features
- Bulk Processing
- The Recycle Bin
- Full Text Search
- Smart Bulk DML
- Web Services APIs

Force.com’s native Application Framework provides declarative development, no coding



Custom Object Information

The singular and plural labels are used in tabs, page layouts, and reports.

Label	SalesOrder	Example: Account
Plural Label	SalesOrders	Example: Accounts

The Object Name is used when referencing the object via the API.

Object Name	SalesOrder	Example: Account

Description

Context-Sensitive Help Setting
● Open the standard Salesforce Help & Training winc
○ Open a window using a custom s-control

Custom S-Control: --None--

Enter Record Name Label and Format

The Record Name appears in page layouts, key lists, related lists, lookups, and se
"Account Name" and for Case it is "Case Number". Note that the Record Name fiel

Record Name	SalesOrder Name	Example: Account Name
Data Type	Text	

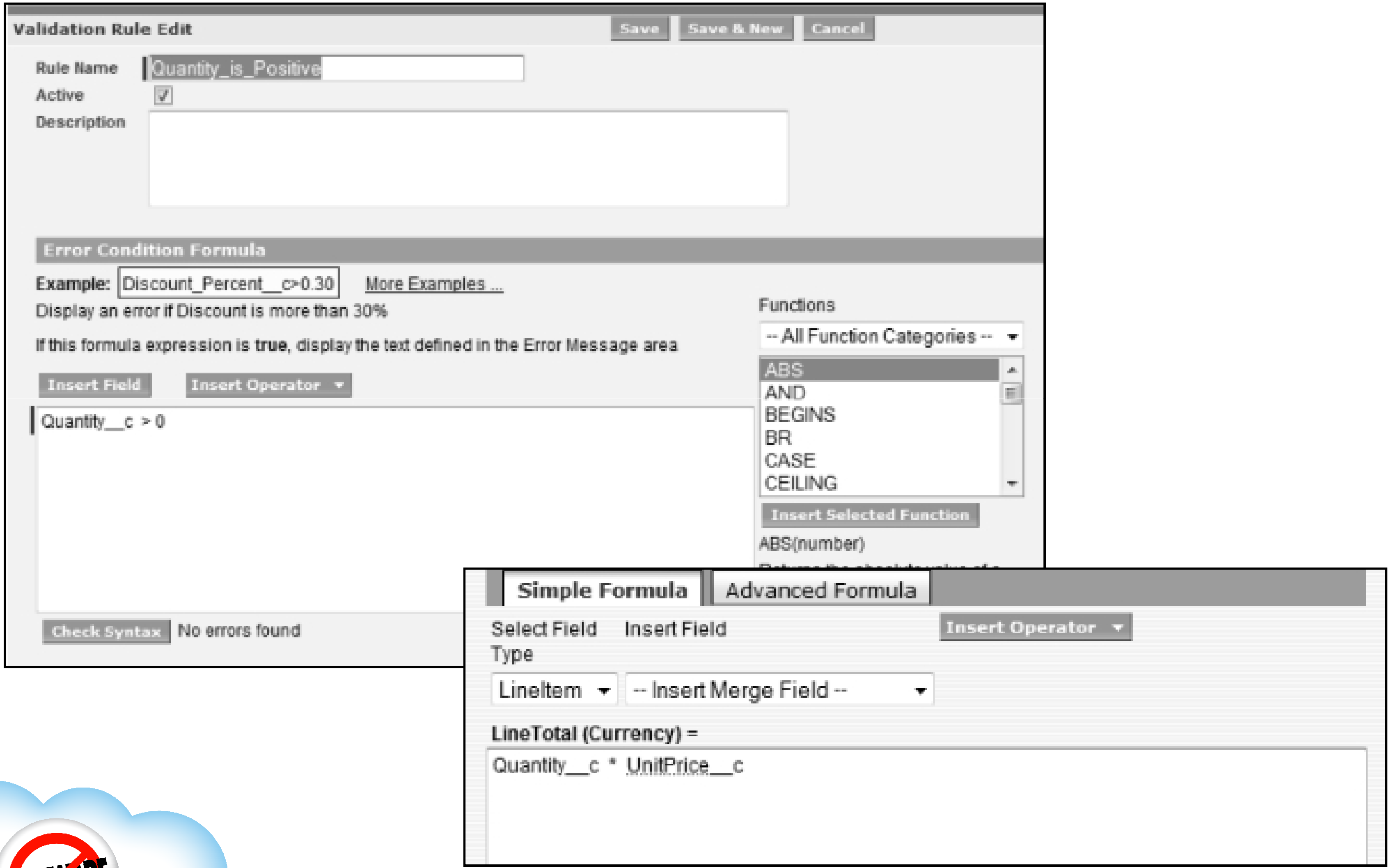
Optional Features

□ Allow Reports
□ Allow Activities
□ Track Field History

Track old and new values

Candidate Number	□	City	□
Country	□	Current Employer	□
Currently Employed	□	Education	□
Email	□	Fax	□
First Name	□	Last Name	□
Mobile	□	Owner	□
Phone	□	SSN	□
State/Province	□	Street	□
US Citizen	□	Visa Required	□
Years of Experience	□	Zip/Postal Code	□

Validation rules and simple formulas: Business analysts can "code" these



Validation Rule Edit

Rule Name: Quantity_is_Positive

Active: ☑

Description:

Error Condition Formula

Example: Discount_Percent__c>0.30 More Examples ...

Display an error if Discount is more than 30%

If this formula expression is **true**, display the text defined in the Error Message area

```
Quantity__c > 0
```

ABS(number)

No errors found

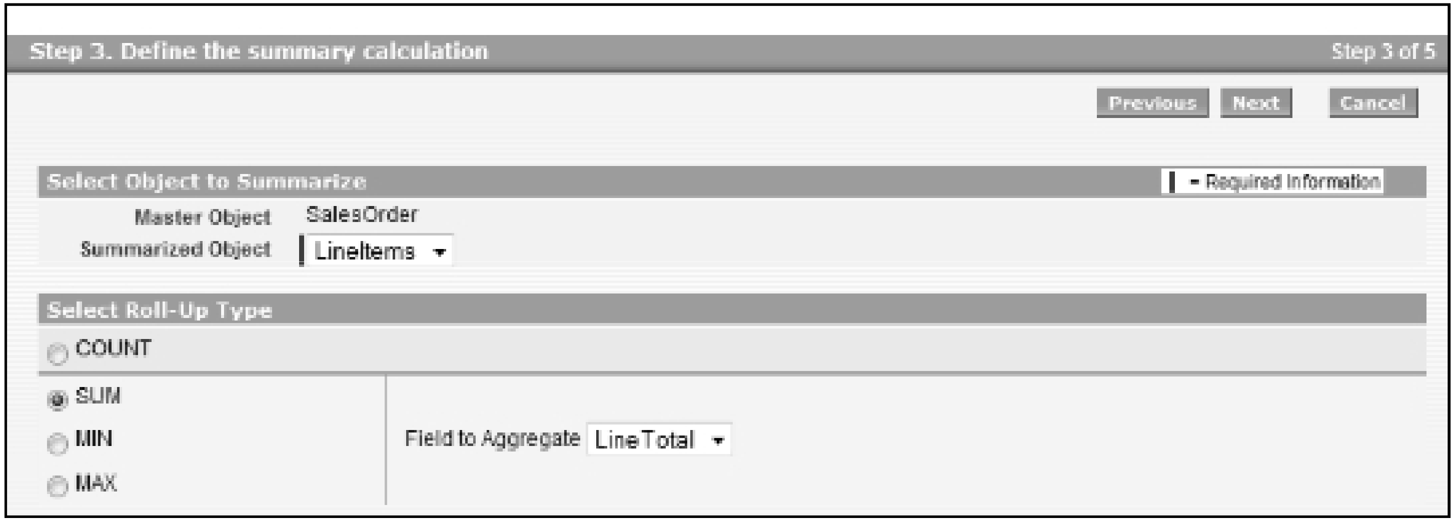
Select Field Type: LineItem

Insert Field: -- Insert Merge Field --

LineTotal (Currency) =

```
Quantity__c * UnitPrice__c
```

Not so simple: Rollup-summary fields provide for easy cross-object summaries



Step 3. Define the summary calculation

Select Object to Summarize

Master Object: SalesOrder

Summarized Object: LineItems

Select Roll-Up Type

○ COUNT

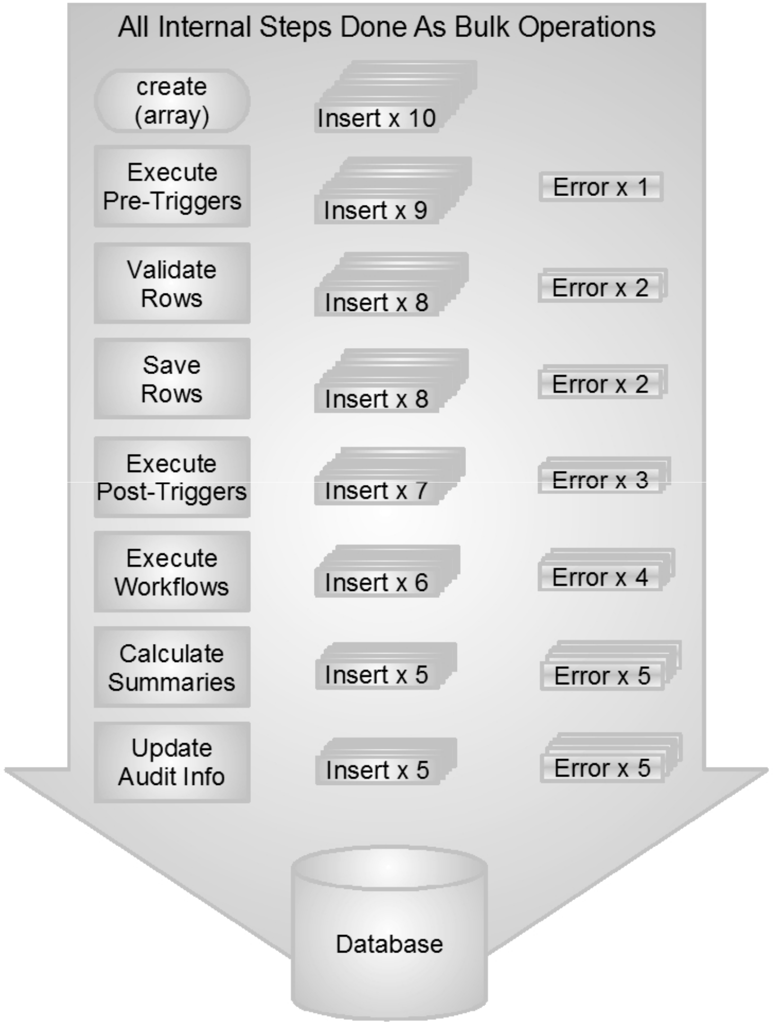
● SUM

○ MIN

○ MAX

Field to Aggregate: LineTotal

Force.com's bulk processing optimizations reduce overhead for data loads



Data definition processing is optimized to avoid performance hits or concurrency limits

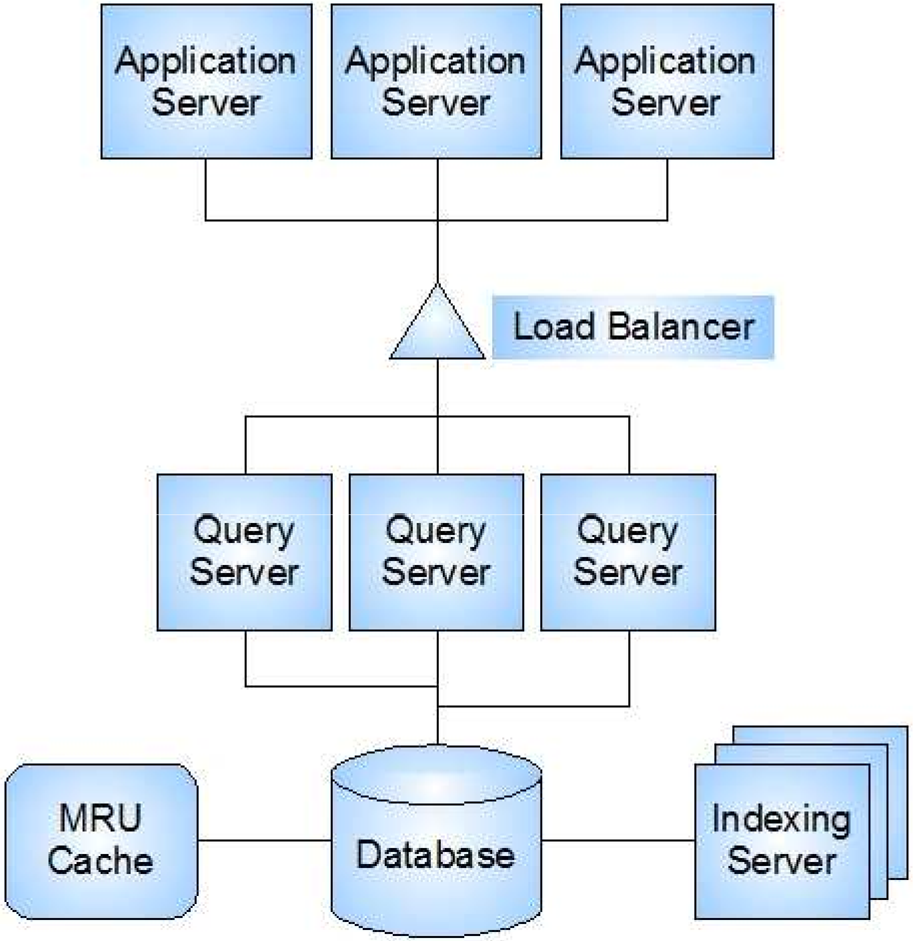
Examples:

- Sort all records by primary key before attempting DML
- Operate on tables in deterministic order
- Slot reallocation for field datatype change
- Deferred calculation for new rollup-summary field
- Background processing of mass changes

The Recycle Bin: Smart Undeletes

- Individual object instances (records)
- Related object instances (parent/child records)
- Entire fields and objects (dropped columns and tables)

Force.com's full-text search engine

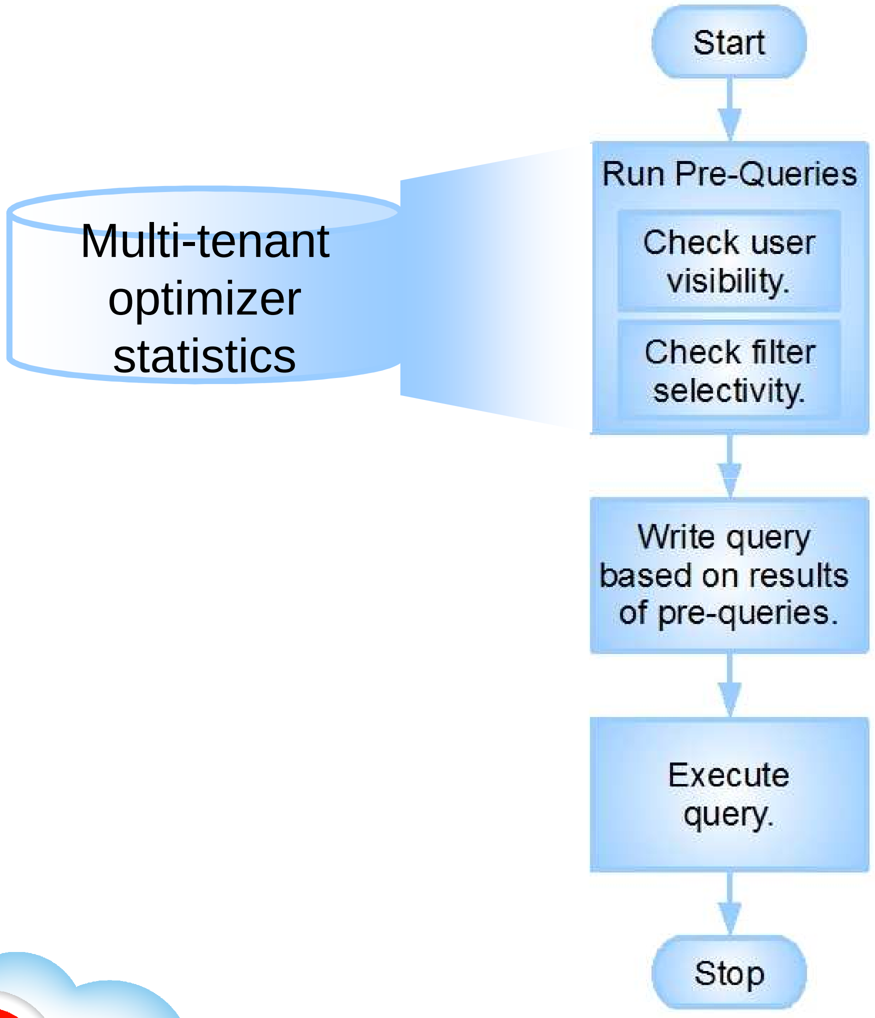


- Asynchronously maintains indexes for all text fields
- MRU caches contain recently updated objects
- Optimizes ranking of search result records based on current user, modification history, and weighting preferences

Multi-tenant Query Optimization Principles

- Consistent SQL generation across the application
- Deep awareness of pivot table structure
  - Flex schema does impose a cost
- Tenant, user, object, fields statistics are crucial
- No runaway queries allowed
- Deep integration with the sharing model

Force.com’s query optimizer writes optimal queries for internal data access operations



Multi-tenant optimizer statistics

Start

Run Pre-Queries

Check user visibility.

Check filter selectivity.

Write query based on results of pre-queries.

Execute query.

Stop

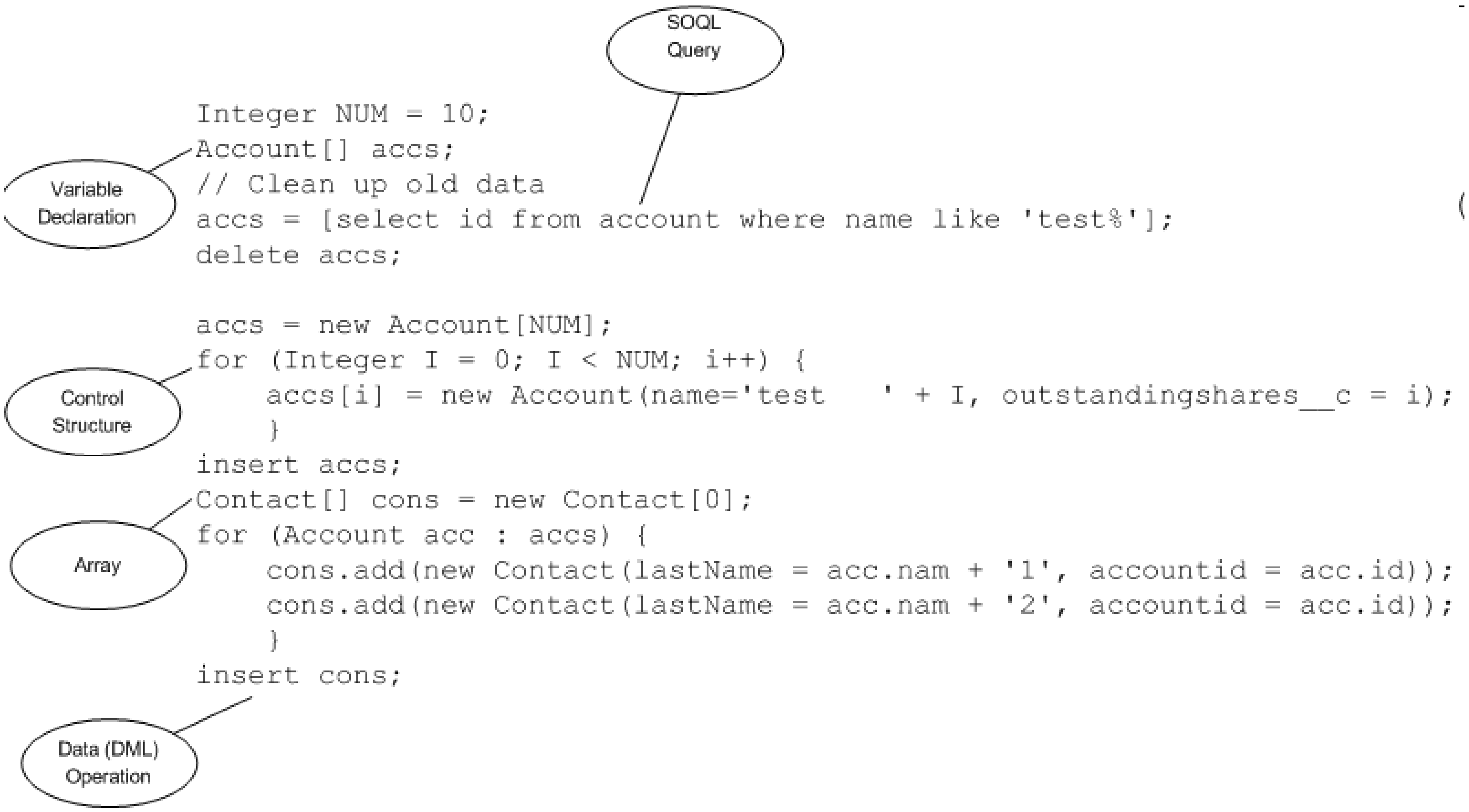
user visibility = number of rows that the user can access

filter selectivity = index corresponding to filter column

The optimizer considers pre-query selectivity measurements when writing a query

Pre-Query Selectivity Measurements		**Write final database access query, forcing …**
User	**Filter**	
Low	Low	… nested loops join; drive using view of rows that the user can see.
Low	High	… use of index related to filter.
High	Low	… ordered hash join; drive using Data table.
High	High	… use of index related to filter.

Apex: Force.com’s procedural frontier



SOQL Query

Variable Declaration

Control Structure

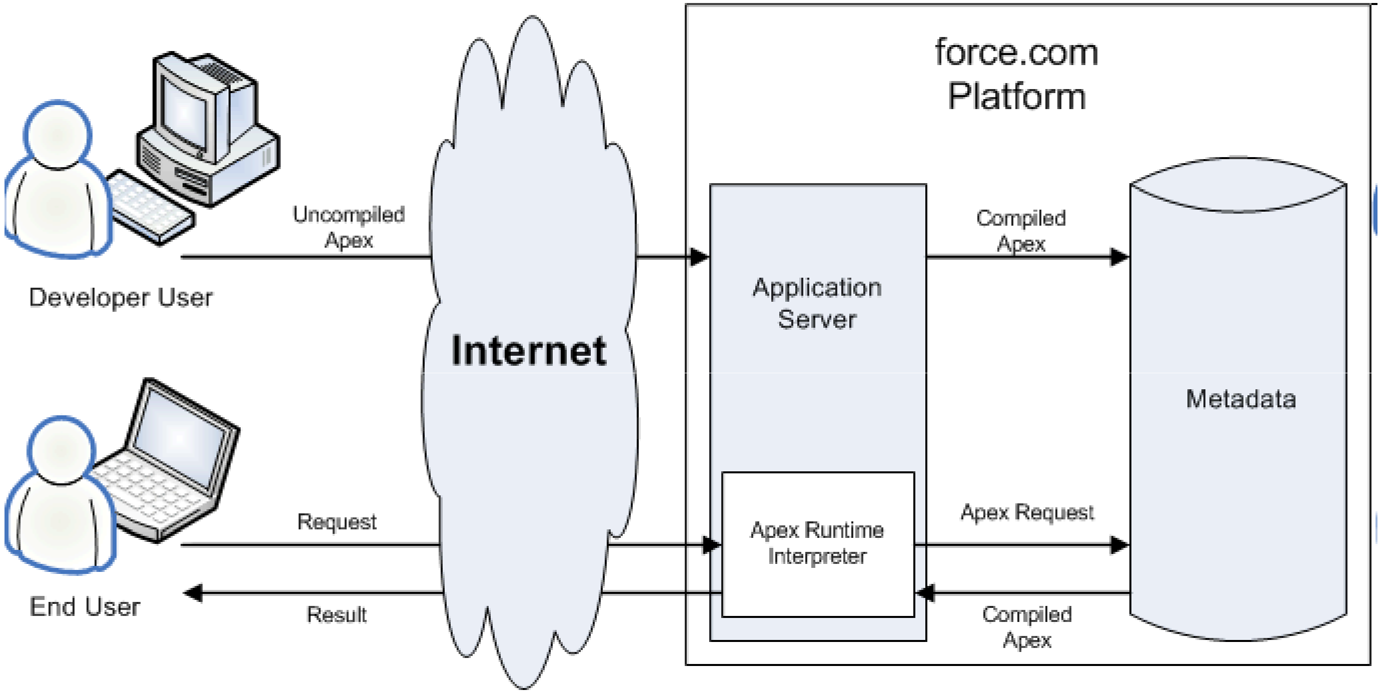
Array

Data (DML) Operation

```
Integer NUM = 10;
Account[] accs;
// Clean up old data
accs = [select id from account where name like 'test%'];
delete accs;

accs = new Account[NUM];
for (Integer I = 0; I < NUM; i++) {
    accs[i] = new Account(name='test    ' + I, outstandingshares__c = i);
    }
insert accs;
Contact[] cons = new Contact[0];
for (Account acc : accs) {
    cons.add(new Contact(lastName = acc.nam + '1', accountid = acc.id));
    cons.add(new Contact(lastName = acc.nam + '2', accountid = acc.id));
    }
insert cons;
```

Apex code is stored as metadata, interpreted at runtime, and cached for scalability



Apex is deeply integrated with platform features

- Bulk DML
- Email and messaging
- Asynchronous processing (Futures)
- XmlStream / HTTP (RESTful) services classes
- Declarative exposure as new Web Services

Force.com governs Apex code execution

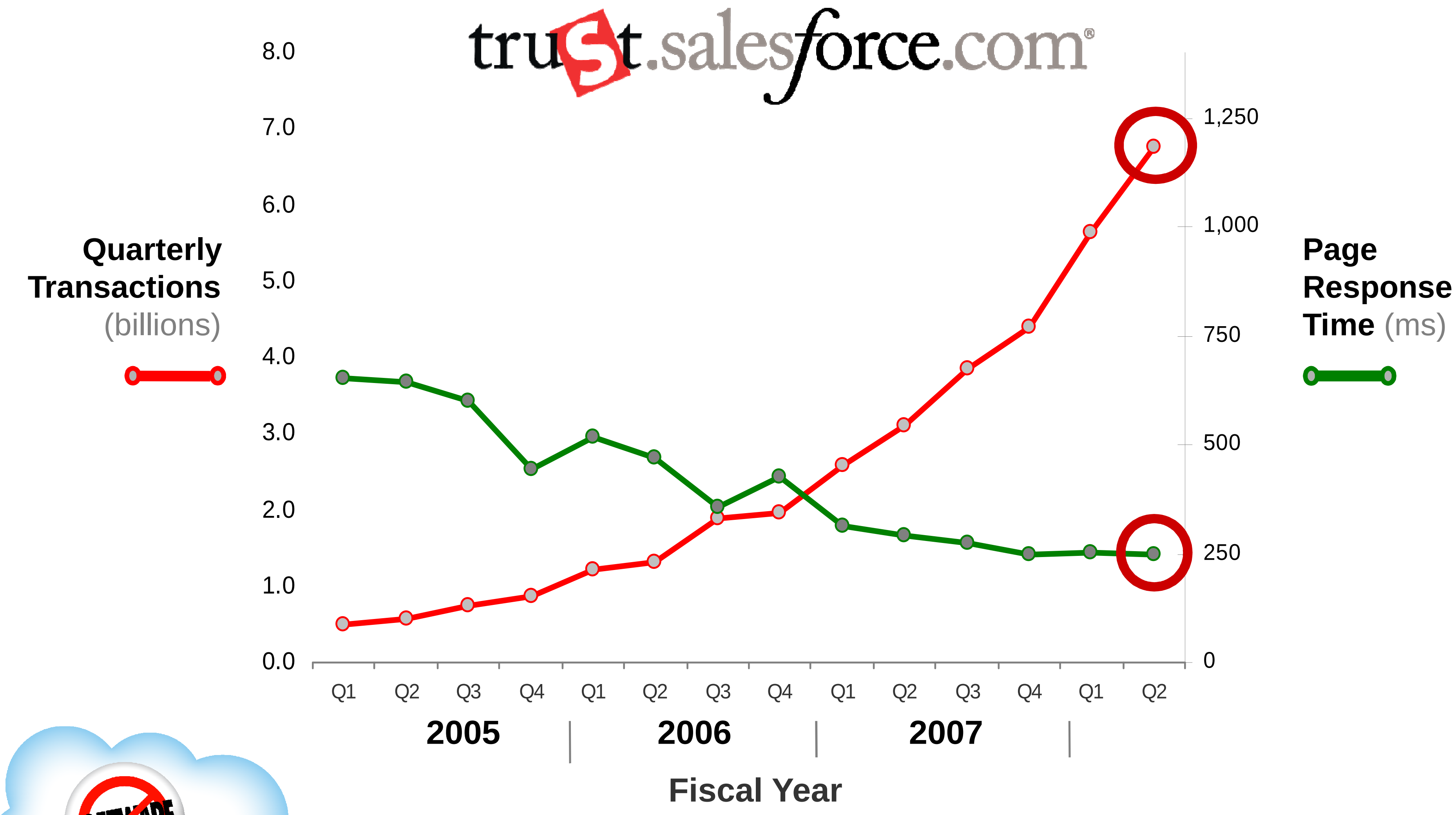
Limits on:

- CPU
- Memory
- # of DML statements
- # calculations
- # web service calls
- … and more

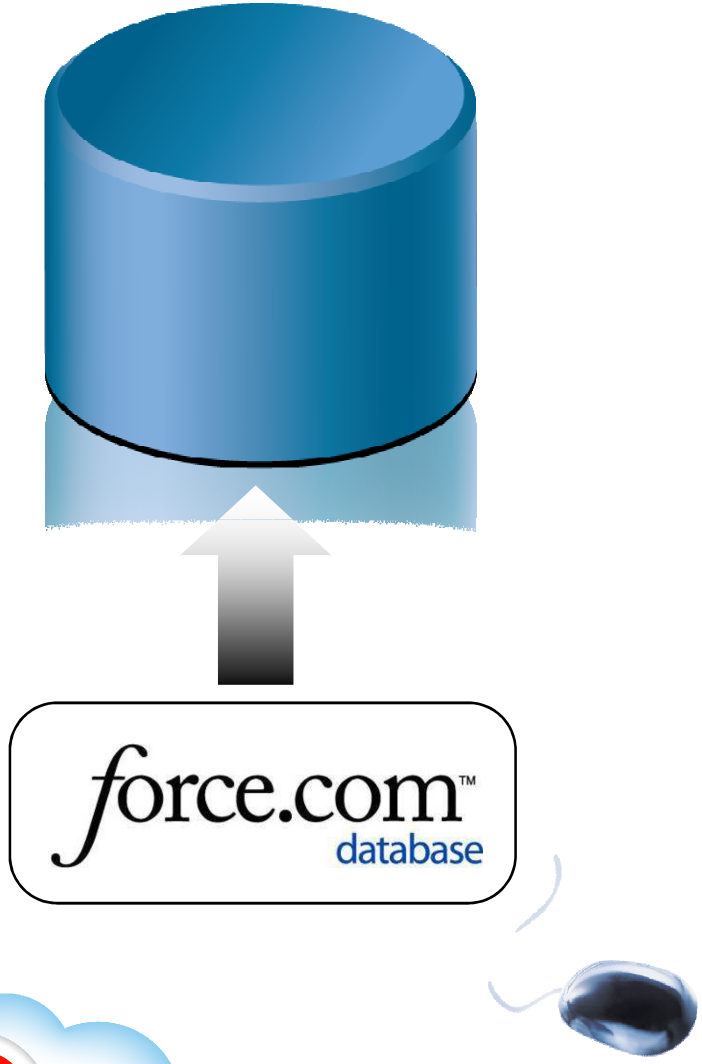
Unit tests must accompany Apex code

- Required 75% code coverage
- Profiling is built into the platform
- Run during application install
- All tests are run before each platform release by us

Force.com is a proven multi-tenant application platform that performs and scales



Concluding Remarks



- PaaS is a major architectural shifts
- PaaS is Application focused, high level of abstraction
- Force.com is the most mature, proven PaaS offering available today
- Optimized for fast, secure, and reliable multi-tenant application development and deployment