References

Amazon SimpleDB Developer Guide  API Version 2007-11-07

Programming Amazon Web Services, James Murty O'Reilly Media, Inc, Chapter 13
SimpleDB

Simple database

Accessed via REST or SOAP

No SQL

No database schema

Simple queries
Data Model

Domains
Items
Attributes
Domain

Just a collection of items
   Sort of like a table
       ArrayList of item objects is better analogy

Items in a domain do not have to be related

Queries are on a single domain

No foreign keys to other domains
Item & Attribute

Item
   Collection of attributes
   Have names

Attribute
   Has a name
   Multiple text values

Items in the same domain can have different attributes

Only text values
No numbers, etc.
### Domain Example

<table>
<thead>
<tr>
<th>Item</th>
<th>Category</th>
<th>Subcat.</th>
<th>Name</th>
<th>Color</th>
<th>Size</th>
<th>Make</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item_01</td>
<td>Clothes</td>
<td>Sweater</td>
<td>Cathair Sweater</td>
<td>Siamese</td>
<td>Small, Medium, Large</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Item_02</td>
<td>Clothes</td>
<td>Pants</td>
<td>Designer Jeans</td>
<td>Paisley, Acid, Wash</td>
<td>30x32, 32x32, 32x34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Item_03</td>
<td>Car Parts</td>
<td>Engine</td>
<td>Turbos</td>
<td></td>
<td></td>
<td>Audi</td>
<td>S4</td>
</tr>
<tr>
<td>Item_04</td>
<td>Car Parts</td>
<td>Emissions</td>
<td>02 Sensor</td>
<td></td>
<td></td>
<td>Audi</td>
<td>S4</td>
</tr>
<tr>
<td>Item_05</td>
<td>Motorcycle Parts, Clothing</td>
<td>Bodywork</td>
<td>Fender Eliminator</td>
<td>Blue</td>
<td></td>
<td>Yamaha</td>
<td>R1</td>
</tr>
<tr>
<td>Item_06</td>
<td>Motorcycle Parts, Clothing</td>
<td>Clothing</td>
<td>Leather Pants</td>
<td>Black</td>
<td>Small, Medium</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Row = item  
Column header = attribute name  
Cell = attribute values
Eventual Consistency

SimpleDB domains are replicated on multiple machines

There can be a delay in propagating any changes
## Some Limits

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain size</td>
<td>10 GB per domain</td>
</tr>
<tr>
<td>Domain size</td>
<td>250,000,000 attribute name-value pairs</td>
</tr>
<tr>
<td>Domains per account</td>
<td>100</td>
</tr>
<tr>
<td>Attribute name-value pairs per item</td>
<td>256</td>
</tr>
<tr>
<td>Attribute name length</td>
<td>1024 bytes</td>
</tr>
<tr>
<td>Attribute value length</td>
<td>1024 bytes</td>
</tr>
<tr>
<td>Item name length</td>
<td>1024 bytes</td>
</tr>
<tr>
<td>Maximum query execution time</td>
<td>5 seconds</td>
</tr>
</tbody>
</table>
SimpleDB Operations

Create Domain
Delete Domain
List Domains
Put
  Add, modify, or remove data
Delete
  Remove items, attributes, or attribute values from your domain
Get
  Retrieve the attributes and values of any item ID that you specify
Query
  Query a domain against attribute values
QueryWithAttributes
  Query a domain and retrieve the results for matching items
GetAttributes

Returns all of the attributes associated with the item.

Can restrict response to a particular attribute

Request Parameters

<table>
<thead>
<tr>
<th>Item name</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain name</td>
<td>Required</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Optional</td>
</tr>
</tbody>
</table>
Queries

Basic Syntax

[attributeName comparison value]

['title' starts-with 'Erlang']
['size' > 10]
['name' != 'John']

Operators

=  
!=
>
>=
<
<=
starts-with
does-not-start-with
and & or

Can and/or queries on the same attribute

['Year' > '1975' and 'Year' < '2008']

['Rating' = '***' or 'Rating' = '*****']

['Year' > '1950' and 'Year' < '1960' or 'Year' starts-with '193' or 'Year' = '2007']
Set Operators

intersection
union
not

['first name' = 'John'] intersection ['last name' = 'Smith']
['tag' starts-with 'Amazon'] union ['description' = 'SimpleDB']
not ['first name' = 'John']
not Verses !=

['Keyword' != 'Book']

Find all items that contain attribute 'Keyword'

Return those that don't have value 'Book'

All items returned have attribute 'Keyword'

not['Keyword' = 'Book']

Find all items that contain attribute 'Keyword'

And have a value equal to 'Book'

Return all other items

Will return items without attribute 'Keyword'
Numbers

"This provides application designers with the flexibility of enforcing data restrictions at the application layer without the data store enforcing constraints."

Translation

All that useful stuff with numbers you normally use does not work

Example - Comparing numbers

<table>
<thead>
<tr>
<th>As Number</th>
<th>As String</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 &lt; 10</td>
<td>2 &gt; 10</td>
</tr>
</tbody>
</table>

All data in SimpleDb is a string

Quote is from Amazon SimpleDB Developer Guide  API Version 2007-11-07
Number Problems

Any query on "numerical" data will give wrong results

['age' > '10']

will return item with age = '2'
Amazon's Recommendation

Negative Numbers Offsets
Make all number positive
Add a large integer
14.58, -12536.791, 20071109
+ 100,000
100014.58, 87463.209, 20171109

Zero Padding
00100014.58, 00087463.209, 20171109

Convert all queries
['attribute' > '500'] → ['attribute' > '00100500']

Convert all data when using it
00100014.58 → 14.58
- 100,000
Dates and other Types

Everything is a string

It is your job to
  Convert between types
  Structure strings so queries work
<table>
<thead>
<tr>
<th>Query &amp; QueryWithAttributes</th>
<th>Requests sent to SimpleDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAP or Rest</td>
<td>Can request first N results</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query</th>
<th>Returns item names the satisfy query</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueryWithAttributes</td>
<td>Returns attributes the satisfy query</td>
</tr>
</tbody>
</table>
BoxUsage

Queries that take longer than 5 seconds to execute time out

All results return execution time taken - BoxUsage
Optimizing Queries

"The service also automatically indexes every piece of information stored within it to ensure queries will run as quickly as possible without the need for performance tuning."

The developers guide talks about query tuning

ands are faster than intersection

["attribute1" > "value1" and "attribute1" < "value2"]
["attribute1" > "value1"] intersection ["attribute1" < "value2"]

Quote from Programming Amazon Web Services
Scaling

Amazon claims SimpleDB scales

Not clear what makes SimpleDB scale more than other database

Amazon's replication of domains should help it scale

XML, REST, SOAP are slow