database design: keys

Daniel Jackson
why not one big table?

<table>
<thead>
<tr>
<th>about</th>
<th>rating</th>
<th>content</th>
<th>by</th>
<th>email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clover</td>
<td>5</td>
<td>yummy!</td>
<td>Alice Alert</td>
<td><a href="mailto:aa@mit.edu">aa@mit.edu</a></td>
</tr>
<tr>
<td>Clover</td>
<td>4</td>
<td>so trendy</td>
<td>Ben Bitdiddle</td>
<td><a href="mailto:bb@mit.edu">bb@mit.edu</a></td>
</tr>
<tr>
<td>6170</td>
<td>5</td>
<td>most fun ever!</td>
<td>Alice Alert</td>
<td><a href="mailto:aa@mit.edu">aa@mit.edu</a></td>
</tr>
<tr>
<td>6170</td>
<td>4</td>
<td>go closures!</td>
<td>Chloe Closure</td>
<td><a href="mailto:cc@mit.edu">cc@mit.edu</a></td>
</tr>
<tr>
<td>Lucid</td>
<td>5</td>
<td>great software</td>
<td>Alice Alert</td>
<td><a href="mailto:aa@mit.edu">aa@mit.edu</a></td>
</tr>
</tbody>
</table>
problems with one big table

wasted space
› lots of needless repetitions (eg, Alice’s email)

inexpressive
› how to record users without reviews?

inconsistencies
› add a new review with a different email for Alice?

not modular
› any change to schema may change all queries

so we need...
› a strategy for dividing into multiple tables
› fundamental idea: keys
what’s a key?

in a database table
› field can have same value in >1 row
› because no set-valued entries

eg, two reviews might have
› same rating, same by
› same rating, same about

but some values occur at most once
› review.id uniquely identifies tuple
› field that identifies tuple is a key

a key can be a combination of fields
› (by, about) may be a key
natural and surrogate keys

natural key
› from problem domain
› eg, **email** is key for **users**
› eg, **SSN** is key for **employee**

surrogate key
› generated by implementation
› eg, **reviews.id**

beware using natural keys
› cost of lookup: integer id smaller
› may become non-keys
  eg, **email** may be used by >1 family member
what’s a foreign key?

use of one table’s key in another table
› subjects.id is a key for subjects
› reviews.about is matched to subjects.id
› so about is a foreign key in reviews
how are keys declared?

programmer gives
› one key as **primary**
› additional keys as **unique**
› foreign keys with **references**

primary vs unique
› primary keys must be non null
› but unclear in SQLite

```sql
CREATE TABLE reviews (  
id integer primary key,  
by integer references users(id),  
content text,  
rating integer,  
about integer references subjects(id),  
unique (about, by)
);
```
what does declaring keys do?

indexes
› creates indexes for optimizing queries

checking constraints
› on every update or delete
› exception thrown if broken

maintaining constraints
› propagate changes on foreign keys
› user specifies action:
  restrict: disallows change, no exception
  cascade: deletes or updates associated row

see http://www.sqlite.org/foreignkeys.html