Guide to Database Design

Normalization

The process of "normalizing" a relational database design. Different levels of normalization are possible, each level further seperating concerns and abstracting data over multiple tables.

Entity Relationship Diagram (ERD)

A way to visually describe the schema of a database and the relationships between tables.

One-to-One Relationship

- ***Definition:*** a relationship where one record on a table matches with one record on another table.
- ***Example:*** users and user_details
- ***Common strategy to implement:*** the owner table of the relationship should have a foreign key to the non-owner.

One-to-Many Relationship

- ***Definition:*** a relationship where one record on a table matches with several records on another table.
- ***Example:*** users and posts
- ***Common strategy to implement:*** the table on the many side should have a foreign key pointing to the single owner from the owning-side table.

Many-to-Many Relationship

- ***Definition:*** a relationship where one record on a table matches with several records on another table and a record on the other table matches with several records on the first table.
- ***Example:*** users and events
- ***Common strategy to implement:*** neither table should have a foreign key to the other and instead, an associative (join) table should be used to link the two together via two foreign key columns, each linking to a primary key of both tables in the many-to-many relationship.

DB Design Conventions to Follow

1. All tables except associative tables should have a single primary key called id.

```
users
  id
  username
  email
  password
```

2. All table cells must contain atomic data (a single cell cannot contain multiple values).

```
  users
    id
    username
    phone_numbers

  1 | bob123| 2105554545, 2105556767, 2105553232 <--- AVOID THIS!
```

3. All columns within a single row should be independent from one another (if not, another table may be needed).

```
  users
    id
    name
    favorite_color_1
    favorite_color_2

  A better alternative...

  users
    id
    name

  colors
    id
    name

  color_user
    color_id
    user_id
```

4. When duplicate values exist accross multiple rows, consider breaking out the column(s) into another table.

```
users
  id
  username
  email
  password
  home_state

1 | bob123 | bob@email.com | JKL3J2JHSURNZ931H | Texas
2 | kev123 | kev@email.com | JKL3J2JHSURNZ931H | Texas <--- duplicate!
3 | dan123 | dan@email.com | JKL3J2JHSURNZ931H | Arizona
4 | mat123 | mat@email.com | JKL3J2JHSURNZ931H | Washington
5 | rob123 | rob@email.com | JKL3J2JHSURNZ931H | Texas <--- duplicate!

Consider a states table.
```

Helpful Links

Additional Info

- ERDs Explained: https://www.lucidchart.com/pages/er-diagrams
- Guide to Crow's Foot notation: https://www.vertabelo.com/blog/crow-s-foot-notation/

ERD Creation Apps

- draw.io
- lucidchart
- sqldbm
- Don't forget about good 'ol pencil and paper or a whiteboard!