```
class Flight < ApplicationRecord

  has_many :passengers
  has_many :users, :through => :passengers

end


class Passenger < ApplicationRecord

  belongs_to :flight
  belongs_to :user

end
```

This is the "join model"

```
class User < ApplicationRecord

  has_many :passengers
  has_many :flights, :through => :passengers

end
```

- "Authentication" means "identification."
- "Authorization" means "permission."
- HTTP requests for identify a user-specific or sensitive resource must be authorized

- A browser cookie is an HTTP header that's preserved between requests
- Cookie values are set by the app during a response, and are subsequently transmitted back to the app with every subsequent browser request.
- Cookie data can "expire," which just means we trust the browser to delete the header value
- 80% case: use the **`session`** hash in Rails to read, write, update, and delete cookie data.
- Rails will encrypt and decrypt the **session** cookie value automatically

- You should try to avoid storing user passwords in any format, even if they're hashed.
- If you must store a password, use a one-way hash such as the **bcrypt** algorithm**.**
- Do not allow plaintext passwords to exist on disk, ever.

- "Strong parameters" means "untrusted by default".
- Use **`params.permit(….)`** and **`params.require(….)`** to "whitelist" the params you want to allow
- If you don't use mass assignment in your controllers, you don't need to worry about this at all

- Use the "placeholder" SQL syntax for user-provided query parameters

If you must store passwords in your database, use a one-way hash and follow best practices at all times.

Here is a simple 5-Step Recipe:

1. Add the **bcrypt** gem to your app (and don't forget to **bundle install**)
2. Add a column named **password_digest** to your User model
3. Add **`has_secure_password`** to your User model
4. You can still use **.password=** and **.password_confirmation=** as expected
5. Call **.authenticate()** on a User object to validate a given plain-text password.

app/models/user.rb

```
class User < ApplicationRecord

  has_secure_password

end
```

app/controllers/sessions_controller.rb

```
class SessionsController < ApplicationController

  def create
    user = … # find the user row

    if user.authenticate(params[:password])
       # They have been identified

        session[:user_id] = user.id

        # etc.
    end

  end

end
```