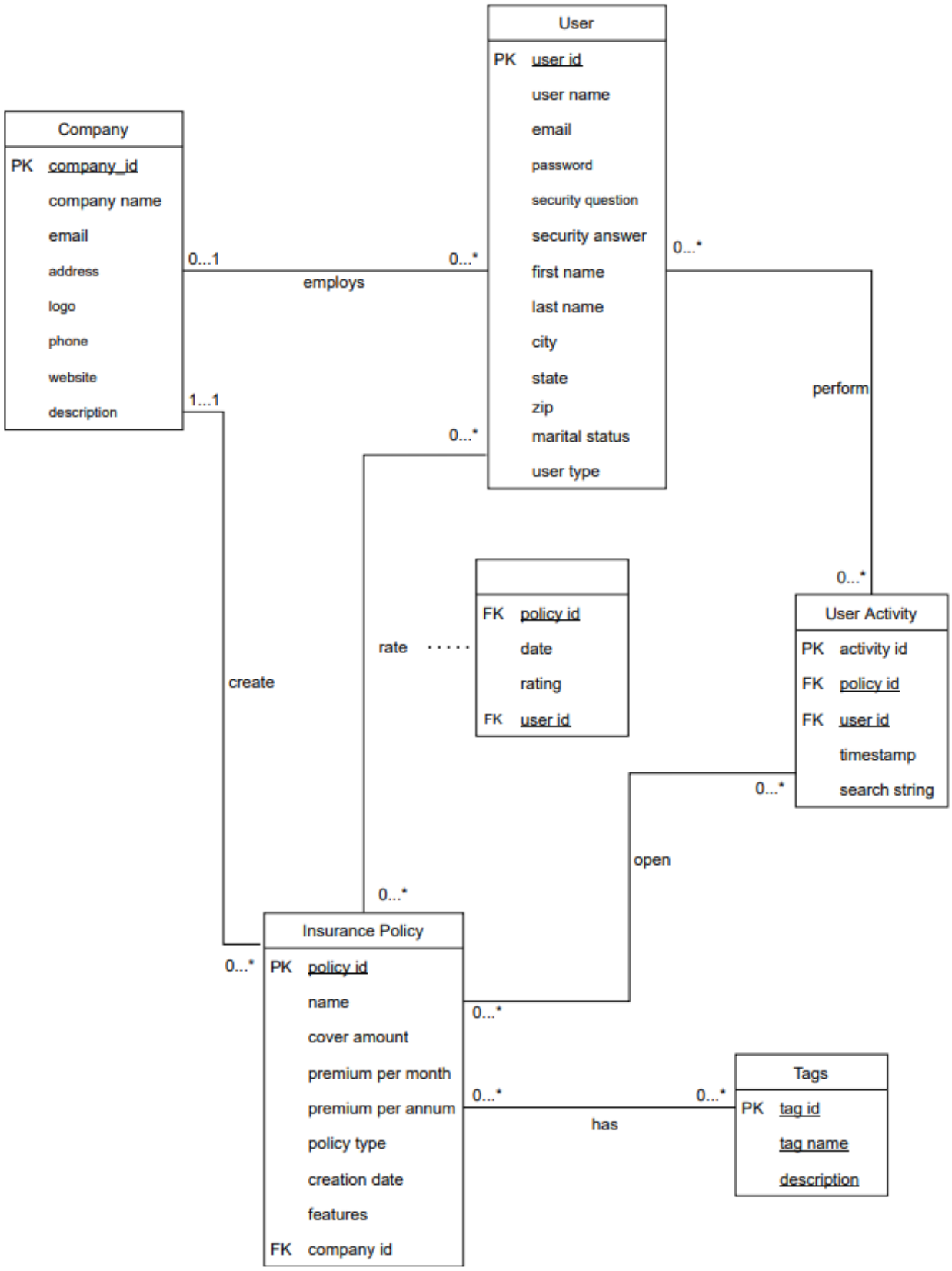
User
PK user id
user name
email
password
security question
security answer
first name
last name
city
state
zip
marital status
user type
Company
PK company_id
company name
email
address
logo
phone
website
description
0...1
employs
0...*
1...1
create
0...*
0...*
perform
0...*
rate
FK policy id
date
rating
FK user id
User Activity
PK activity id
FK policy id
FK user id
timestamp
search string
0...*
open
0...*
Insurance Policy
PK policy id
name
cover amount
premium per month
premium per annum
policy type
creation date
features
FK company id
0...*
0...*
0...*
has
Tags
PK tag id
tag name
description

Our project has 5 entities: User, User Activity, Company, Insurance Policy and Tags.

- The Users table contains information about the user such as login information, username, email, address, user type etc. A user is uniquely identified by their user id.
- The Company Table contains information about the Companies that offer their insurances on our website. We store information such as company name, address, company logo image, description etc.
- The user_type field in the Users Table identifies the persona of the user. This will tell us what type of user it is. The Employs relation between the User and Company table will map the user to a particular company if the user_type is an employee.
- Each user can belong to 0 or 1 company while a company will have many users.
- The Insurance Policy table will contain the details of the policies that are registered with the system. It contains information such as name, cover amount, premium per month etc. Also, each policy is related to exactly one Company.
- We also have a Tags table which contains additional helpful tags that are associated with each policy. This table contains information of the tag name and its description. It has a many to many relationship with the Insurance policy table
- We also maintain a User Activity table which will be used for analytics. It contains metadata such as which user opened which policy, the timestamp and what the user searched for. It has a many to many relationship with Insurance policy and Users table.
- The Ratings relation between Users and Insurance policy keeps track of the ratings. We will use this table for additional search filtering. It has a many to many relation with the Users and Insurance Policy entities.

Relational Schema:

Company(company_id:INT [PK], company_name:VARCHAR(255), email:VARCHAR(255), address:VARCHAR(255), logo:IMAGE, phone:INT, website:VARCHAR(255), description:VARCHAR(255))

User(user_id:INT [PK], user_name:VARCHAR(255), email:VARCHAR(255), password:VARCHAR(255), security_question:VARCHAR(255), security_answer:VARCHAR(255), first_name:VARCHAR(255), last_name:VARCHAR(255), city:VARCHAR(255), state:VARCHAR(255), zip:INT, marital_status:VARCHAR(255), user_type:INT)

Insurance_Policy(policy_id:INT [PK], name:VARCHAR(255), cover_amount:INT, premium_per_month:INT, premium_per_annum:INT, policy_type:INT, creation_date:DATE, features:VARCHAR(255), company_id:INT [FK to Company.company_id])

Ratings(policy_id:INT [FK to Insurance_Policy.policy_id], date:DATE, rating:INT, user_id:INT [FK to User.user_id])

User_Activity(activity_id:INT [PK], policy_id:INT [FK to Insurance_Policy.policy_id], user_id:INT [FK to User.user_id], timestamp:TIMESTAMP, search_string:VARCHAR(255))

Tags(tag_id:INT [PK], tag_name:VARCHAR(255), description:VARCHAR(255))

Employs(user_id:INT [PK], company_id:INT [FK to Company.company_id])

Create(policy_id:INT [PK], company_id:INT [FK to Company.company_id])

Has(tag_id:INT [PK], policy_id:INT[PK], tag_id:INT [FK to Tags.tag_id], policy_id:INT [FK to Insurance_Policy.policy_id])