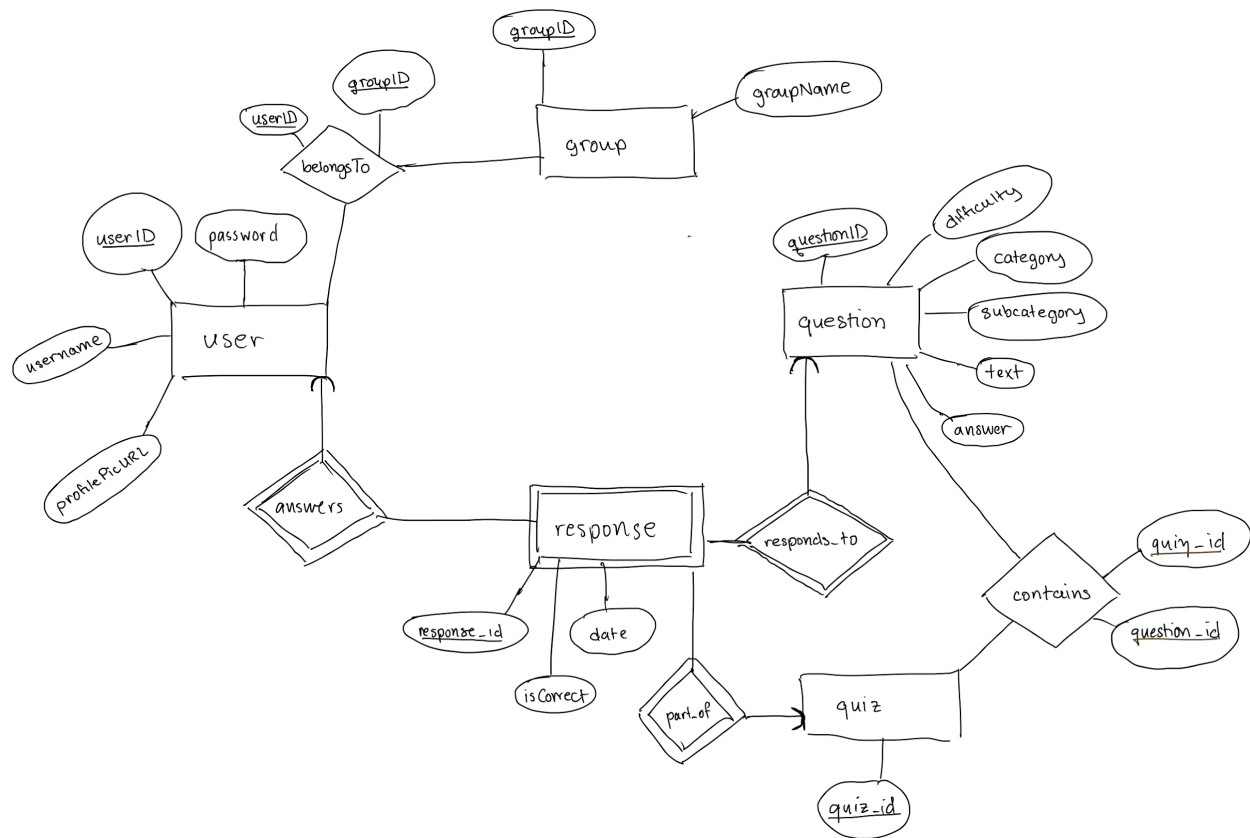


ER-diagram



Assumptions

1. **User to Group** is many-many because a user can be part of different groups, for instance a trivia team and a casual friend group. **Group to User** is also many to many because a **Group** may contain many **Users**.
2. **Response** is weak to **Question**, **User**, and **Quiz** because we wanted to be able to repeat **Questions** in different **Quiz** variations (like PrairieLearn). Without the **Quiz** table, we would not be able to repeat **Questions** for a user without deleting **Responses**. Therefore to uniquely identify a **Response**, we need to know:
 - a. Who answered it (**User**)
 - b. What was the question (**Question**)
 - c. Which quiz variation did it belong to (**Quiz**)
3. The relationship table **Contains** is how we generate **Quiz** variations and repeat **Questions** to **Users**.
4. **Quiz to Question** is many-many because a **Question** may be in multiple **Quizzes** and a **Quiz** has multiple **Questions**.

Relational Schema

```
group(groupID: INT [PK],
      groupName: VARCHAR(20)
)
```

```
belongsTo (
  groupID:INT [PK] [FK to group.groupID],
  userID:INT [PK] [FK to user.userID]
)
```

```
user (
  userID:INT [PK],
  password:VARCHAR(30),
  Username:VARCHAR(30),
  profilePicURL:VARCHAR(2083)
)
```

```
response (
  responseID:INT [PK]
  isCorrect:BIT,
  Date:DATE,
  quizID:INT [PK] [FK to quiz.quizID].
  questionID:INT [PK] [FK to question.questionID]
  userID:INT [PK] [FK to user.userID]
)
```

```
quiz (
  quizID:INT [PK],
)
```

```
contains(
  quizID:INT [PK] [FK to quiz.quizID],
  questionID:INT [PK] [FK to question.questionID],
)
```

```
questions(
  questionID:INT [PK],
  answer:VARCHAR(MAX),
  text:VARCHAR(MAX),
  subcategory:VARCHAR(30),
  category:VARCHAR(30),
  difficulty:VARCHAR(25),
)
```

DDL Database Commands

```
CREATE TABLE group (  
    groupID INT NOT NULL,  
    groupName VARCHAR(20),  
    PRIMARY KEY(groupID)  
)
```

```
CREATE TABLE belongsTo (  
    groupID INT NOT NULL,  
    userID INT NOT NULL,  
    PRIMARY KEY(groupID, userID),  
    FOREIGN KEY (groupID) REFERENCES group(groupID),  
        ON UPDATE CASCADE, ON DELETE CASCADE  
    FOREIGN KEY (userID) REFERENCES user(userID),  
        ON UPDATE CASCADE, ON DELETE CASCADE  
)
```

```
CREATE TABLE user (  
    userID INT NOT NULL,  
    password VARCHAR(30) NOT NULL,  
    Username VARCHAR(30) NOT NULL,  
    profilePicURL VARCHAR(2083),  
    PRIMARY KEY(userID)  
)
```

```
CREATE TABLE response (  
    responseID INT NOT NULL,  
    isCorrect BIT,  
    Date DATE,  
    quizID INT NOT NULL,  
    questionID INT NOT NULL,  
    userID INT NOT NULL,  
    PRIMARY KEY(responseID, quizID, questionID),  
    FOREIGN KEY (quizID) REFERENCES quiz(quizID),  
        ON UPDATE CASCADE, ON DELETE CASCADE  
    FOREIGN KEY (userID) REFERENCES user(userID),  
        ON UPDATE CASCADE, ON DELETE CASCADE  
    FOREIGN KEY (questionID) REFERENCES question(questionID),
```

ON UPDATE CASCADE, ON DELETE CASCADE

)

```
CREATE TABLE quiz (  
    quizID INT NOT NULL,  
    PRIMARY KEY(quizID)
```

)

```
CREATE TABLE contains(  
    quizID INT NOT NULL,  
    questionID INT NOT NULL,  
    PRIMARY KEY( quizID, questionID),  
    FOREIGN KEY (quizID) REFERENCES quiz(quizID),  
        ON UPDATE CASCADE, ON DELETE CASCADE  
    FOREIGN KEY (questionID) REFERENCES question(questionID),  
        ON UPDATE CASCADE, ON DELETE CASCADE
```

)

```
CREATE TABLE questions(  
    questionID INT NOT NULL,  
    answer VARCHAR(MAX),  
    text VARCHAR(MAX),  
    subcategory VARCHAR(30),  
    category VARCHAR(30),  
    difficulty VARCHAR(25),  
    PRIMARY KEY(questionID)
```

)