

به نام خدا

پروژه پایگاه داده 1404

40231011

مهدیه تهامی

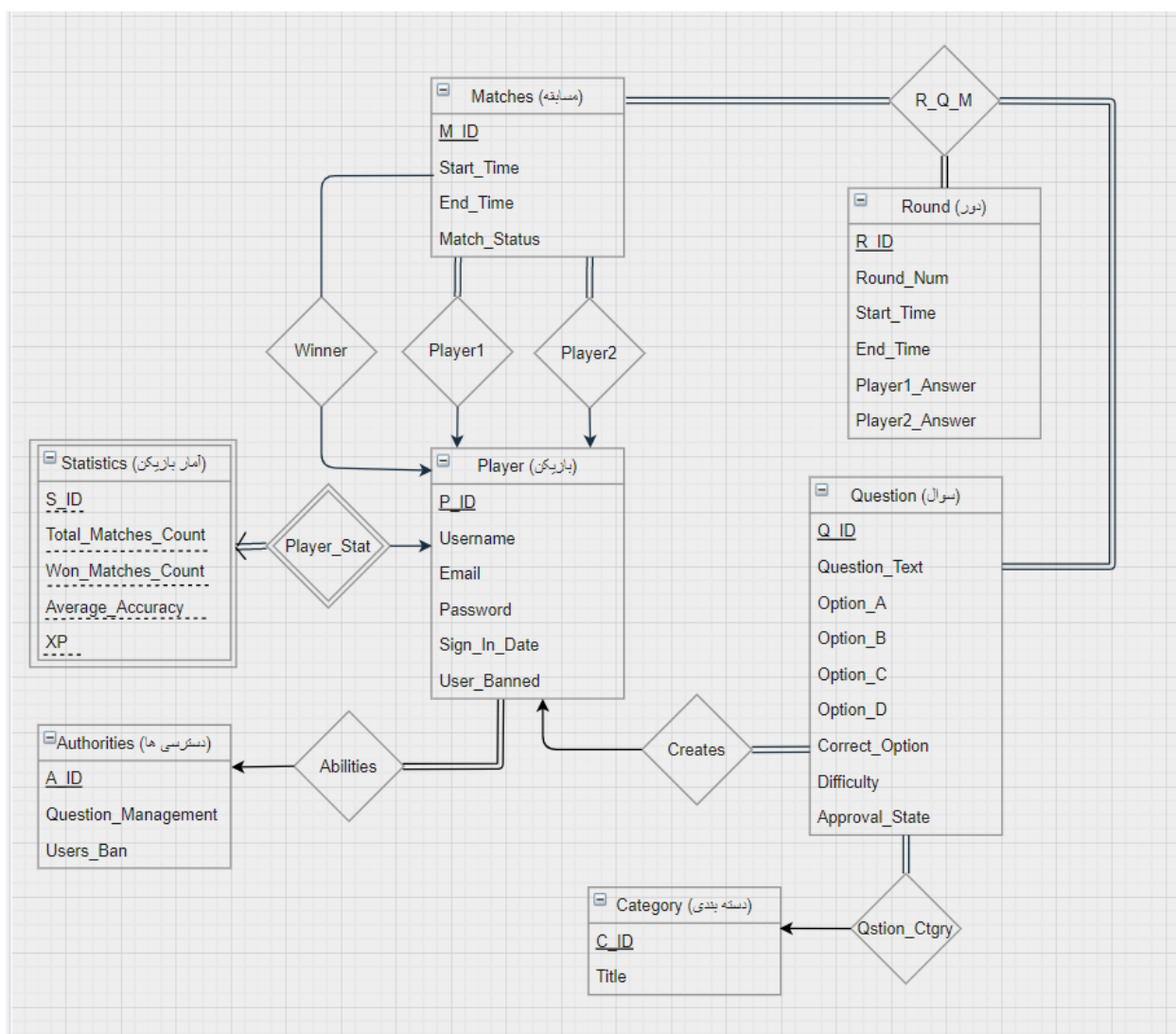
گزارش پروژه

هدف پروژه:

هدف از این پروژه پیاده سازی پایگاه داده بازی مشابه با Quiz Of Kings با استفاده از زبان SQL و طراحی نمودار ER برای آن است.

مراحل پیاده سازی:

در مرحله اول، بر اساس توضیحات داده شده در فایل پروژه، نمودار ER مربوط به پایگاه داده را ترسیم میکنیم:



سپس در MySQL پایگاه داده جدیدی به نام QuizGameDB ساخته و پس از تبدیل نمودار ER به شمای رابطه ای، جداول مرتبط با پایگاه داده را ایجاد میکنیم. در همین حین شروط لازم برای بررسی درستی اطلاعات ذخیره شده در پایگاه داده را در جداول تعریف میکنیم:

CREATE TABLE Authorities (

A_ID INT AUTO_INCREMENT PRIMARY KEY,
Question_Management BOOLEAN NOT NULL,
Users_Ban BOOLEAN NOT NULL

);

CREATE TABLE Player (

P_ID INT AUTO_INCREMENT PRIMARY KEY,
A_ID INT NOT NULL DEFAULT 1,
Username VARCHAR(50) NOT NULL,
Email VARCHAR(100) NOT NULL UNIQUE,
Pass VARCHAR(255) NOT NULL,
User_Banned BOOLEAN NOT NULL DEFAULT false,
FOREIGN KEY (A_ID) REFERENCES Authorities(A_ID),
Sign_In_Date DATE

);

CREATE TABLE Category (

C_ID INT AUTO_INCREMENT PRIMARY KEY,
Title VARCHAR(50) NOT NULL UNIQUE

);

CREATE TABLE Question (

Q_ID INT AUTO_INCREMENT PRIMARY KEY,
C_ID INT NOT NULL,
Question_Text TEXT NOT NULL,

```
Option_A VARCHAR(100) NOT NULL,  
Option_B VARCHAR(100) NOT NULL,  
Option_C VARCHAR(100) NOT NULL,  
Option_D VARCHAR(100) NOT NULL,  
Correct_Option CHAR NOT NULL,  
Creator_ID INT NOT NULL,  
Difficulty VARCHAR(10) NOT NULL,  
Approval_State BOOLEAN NOT NULL DEFAULT false,  
FOREIGN KEY (Creator_ID) REFERENCES player (P_ID),  
FOREIGN KEY (C_ID) REFERENCES Category (C_ID),  
CHECK (Correct_Option IN ('A', 'B', 'C', 'D')),  
CHECK (Difficulty IN ('Easy', 'Medium', 'Hard'))  
);
```

```
CREATE TABLE Round (  
    R_ID INT AUTO_INCREMENT PRIMARY KEY,  
    Round_Num INT NOT NULL,  
    P1_Answer CHAR,  
    P2_Answer CHAR,  
    Start_Time TIMESTAMP NOT NULL,  
    End_Time TIMESTAMP,  
    CHECK (P1_Answer IN ('A', 'B', 'C', 'D')),  
    CHECK (P2_Answer IN ('A', 'B', 'C', 'D')),  
    CHECK (Round_Num > 0 AND Round_Num < 4)  
);
```

```
CREATE TABLE Matches (  
    M_ID INT AUTO_INCREMENT PRIMARY KEY,
```

```

P1_ID INT NOT NULL,

P2_ID INT,

winner_ID INT DEFAULT NULL,

Start_Time TIMESTAMP NOT NULL,

End_Time TIMESTAMP,

Match_Active BOOLEAN NOT NULL DEFAULT TRUE,

CHECK (P1_ID != P2_ID),

FOREIGN KEY (P1_ID) REFERENCES player (P_ID),

FOREIGN KEY (P2_ID) REFERENCES player (P_ID),

FOREIGN KEY (winner_ID) REFERENCES player (P_ID)

);

CREATE TABLE R_Q_M (

    Q_ID INT,

    R_ID INT,

    M_ID INT,

    PRIMARY KEY (Q_ID, M_ID, R_ID),

    FOREIGN KEY (Q_ID) REFERENCES Question (Q_ID),

    FOREIGN KEY (R_ID) REFERENCES Round (R_ID),

    FOREIGN KEY (M_ID) REFERENCES Matches (M_ID)

);

CREATE TABLE Statistics (

    S_ID INT AUTO_INCREMENT PRIMARY KEY,

    P_ID INT UNIQUE NOT NULL,

    Total_Matches_Count INT NOT NULL DEFAULT 0,

    Won_Matches_Count INT NOT NULL DEFAULT 0,

    Accuracy FLOAT NOT NULL DEFAULT 1,

```

```
XP INT NOT NULL DEFAULT 50,  
FOREIGN KEY (P_ID) REFERENCES Player (P_ID) ON DELETE CASCADE  
);
```

بعد از تعریف جداول مربوطه، اطلاعات اولیه برای امکان استفاده از پایگاه داده را در جداول **insert** میکنیم. این کار شامل اضافه کردن دسترسی ها، بازیکنان، سوالات، دسته بندی ها، مسابقات و دوره های مسابقات می شود.

در مرحله بعد فایل اولیه **Client** را ایجاد کرده و صفحه ورود به بازی را ایجاد میکنیم. پس از برقراری **connection** با پایگاه داده، میتوان با نوشتن **query** های ایمن به اطلاعات پایگاه داده دست پیدا کرد. برای پیاده سازی بهینه تر برخی از اهداف پروژه، لازم به تعیین توابع در پایگاه داده است؛ توابعی که کاربرد جامعی دارند و استفاده آن ها فراتر از مرز هر کلاس در برنامه **client** است. پس از طراحی چند تابع، مشاهده میکنیم که چند ستون بسیار پرکاربرد برای یافتن بازیکن ها و مسابقات وجود دارد، که برای آن ها **index** تعریف میکنیم:

```
CREATE INDEX email_index ON Player(email);
```

```
CREATE INDEX pid_index ON Player(p_id);
```

```
CREATE INDEX id_index ON Question(Q_ID);
```

```
CREATE INDEX rid_index ON Round(R_ID);
```

علاوه بر توابع، با توجه به نیاز به بروزرسانی اطلاعات جداول به طور خودکار (مثلا غیرفعال کردن مسابقه بعد از تعیین زمان پایان برای آن) نیاز به تعریف چند **trigger** وجود دارد که حساس به آپدیت و یا درج اطلاعات در جداول باشند. از جمله این **trigger** ها، **update_match_status**، **update_round_status** و **update_statistics** هستند که به ترتیب مربوط به آپدیت وضعیت فعال یا غیر فعال بودن بازی بعد از تعیین زمان پایان، آپدیت زمان پایان دور بعد از پاسخگویی هر دو بازیکن به سوالات، و آپدیت آمار بازیکن ها بعد از پایان بازی میباشند.

کلاس های تعریف شده در فایل **client** به شرح زیر است:

- ActiveMatchesPage
- DatabaseAction
- DatabaseConnection
- HomePage
- Main
- MainMenu
- NewQuestionPage
- QuestionManagementPage
- RankingPage
- SignInPage
- SignUpPage
- StartNewMatchPage
- UserBanPage

کلاس DatabaseAction مربوط به برقراری اتصال به پایگاه داده است. کلاس DatabaseAction وظیفه اجرای query ها و دریافت اطلاعات ارسالی از طرف پایگاه داده را برعهده دارد. کلاس MainMenu صفحه اصلی بازی را نمایش میدهد. کلاس های SignInPage و SignUpPage به ترتیب صفحه های ورود و ثبت نام کاربران را نمایش میدهند. کلاس HomePage صفحه اول برنامه است، که پس از ورود یا ثبت نام کاربر نمایش داده میشود و گزینه های در دسترس کاربر را نمایش میدهد. کلاس های ActiveMatchesCount و StartNewMatchPage به ترتیب صفحه بازی های فعال و شروع بازی جدید را پشتیبانی میکنند. کلاس های NewQuestionPage و RankingPage به ترتیب برای طراحی سوال جدید و مشاهده رتبه بندی جهانی استفاده میشوند. دو کلاس UserBanPage و QuestionManagementPage تنها برای کاربر مدیری که دسترسی محدود کردن کاربران و مدیریت سوالات را داشته باشد استفاده میشود و صفحات مذکور را برای او نمایش میدهد.

در مرحله آخر، کد client و ساختار database بازیابی شده و خطاهای احتمالی رفع شدند.