

## **Тема 32**

### **Проект - Викторина**

**Изготвил: Ивайло Стоянов, ф.н. 71975**

#### **1. Обхват на модела. Дефиниране на задачата**

Базата от данни за Java приложение представляващо играене на многобройни игри с въпроси и отговори, за които ще се пази информация в нашата база. Ще се съхраняват и данните на потребителите, които ще използват приложението.

За всяка изиграна игра се пази информация номер, с който всяка игра се самоопределя. Също игрите са пряко свързани с въпросите, потребителите и за всяка игра ще се пази история за нея. История на игра представлява информация как и на какво е отговорил потребителя. За всяка игра ще има жокера 50/50, помощ от публиката и помощ от приятел.

Приложението ще разполага с множество въпроси, за които ще се пази информация. Съхранява се текст, който е съдържанието на самия въпрос. Всеки въпрос се определя еднозначно с 5 цифрен уникален номер. Въпросите могат да бъдат два вида – въпрос с 4 възможни отговора, от които един е правилен или въпрос с 2 възможни отговора Истина/Лъжа, от които един е правилен. За всеки въпрос се съдържа информация за сложност на въпроса, която ще определя какво време има играчът да отговори и какви точки ще му донесе въпроса ако отговори правилно.

За отговорите се пази информация дали е правилен или грешен отговор и текст, който е съдържанието на отговора. Също всеки отговор се определя еднозначно с номер и с номер на въпроса, на който принадлежи.

За темите се пази информация името на темата и номер, с който се определя еднозначно темата.

За потребителите се пази информация 5 цифрен уникален номер потребителско име, парола, имейл, точки, които се определят от правилно отговорените въпроси и ще се увеличават след всяка изиграна игра. Всеки потребител се определя еднозначно с потребителското си

име. Те ще отговорят на въпросите и ще се пази история за всяка игра как е отговорил играча.

## **2. Множества от същности и техните атрибути**

- Игра – номер на играч, номер на игра, жокер 50/50, жокер публика, жокер приятел
- Въпроси – номер, текст, сложност
- Отговори – номер, номер определящ въпроса, вид, коректност.
- Потребители – потребителско име, парола, имейл, точки
- Теми – име, номер.

## **3. Домейн за атрибути**

- Игра – номер на потребител: цяло число, номер: цяло число, жокер 50/50: символ, жокер публика: символ, жокер приятел: символ.
- Въпроси – текст: низ, номер: цяло число, сложност: низ
- Отговори – номер: цяло число, номер определящ въпроса: цяло число, вид: цяло число (0 или 1), коректност: цяло число (0 или 1)
- Потребители – потребителско име: низ, парола: низ, имейл: низ, точки: цяло число
- Теми: име: низ, номер: цяло число

## **4. Връзки**

- Всяка игра ще има много отговори.
- Всяка игра се играе от точно 1 потребител
- Всеки отговор може да участва във всяка игра.
- Всеки въпрос има точно 1 тема.
- Всеки въпрос има повече от 1 отговор.
- Всеки отговор принадлежи на точно 1 въпрос.
- Всяка тема може да принадлежи на повече от 1 въпрос.
- Потребителите играят много игри.
- Потребителите отговарят на много въпроси

## 5. Ограничения по единствена стойност, референтна цялостност и друг тип ограничения

- Игра – номер на потребител: еднозначно определя играта.
- Въпроси – номер: еднозначно определя въпроса.
- Отговори – номер на въпрос: еднозначно определя отговора.
- Теми – номер: еднозначно определя темата.
- Потребители – потребителско име: еднозначно потребителя.

## 6. Правила и проверки

Потребителско име на играча и паролата трябва да има поне 1 цифра и да е поне 6 символа дълго.

Потребителският имейл трябва да е валиден.

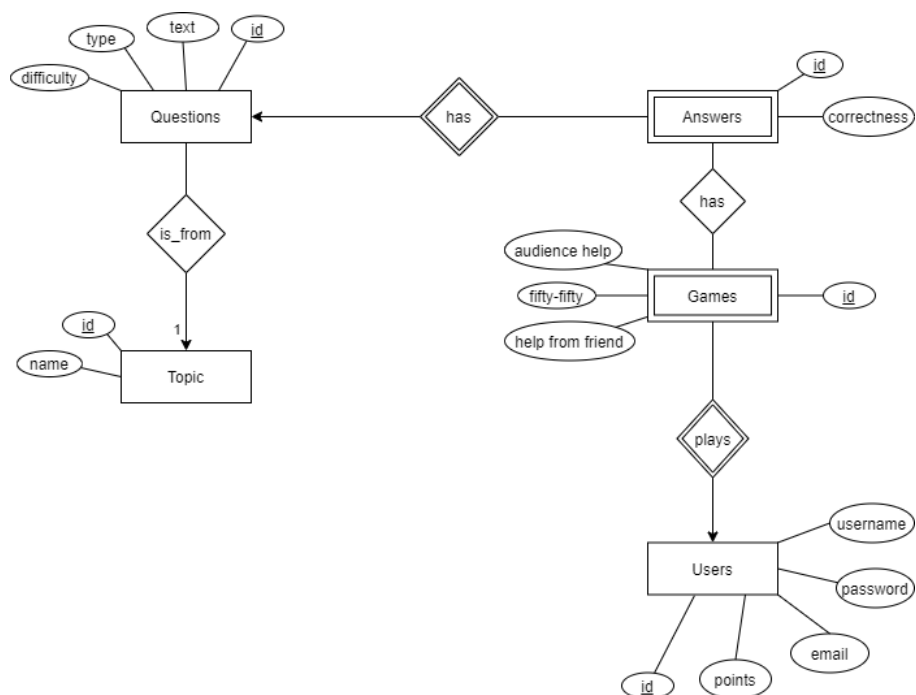
Вида на отговора е символ, който е или „1“ или „2“. Ако е 2 текста трябва да е или „True“ или „False“.

Сложността на въпроса може да е само „hard“, „medium“ или „easy“.

Жокерите са символи, които са или „y“ или „n“.

Коректността на въпрос е символ, който е или „y“ или „n“.

## 7. E/R модел на данни



## 8. Релационен модел на данни

Ще преобразуваме първо силните множества, след това слабите множества, връзките много към много и много към един. Връзките много към един ще бъдат оптимизирани.

Games е слабо множество, което ще определя чрез номер на игра и номер на играча, който ще взема от множеството Users. То ще има връзка с двете множества Questions и Answers тъй като всяка игра има въпроси и отговори и към връзката has слагаме атрибути fifty-fifty, audience help и help from friend, които ще са жокерите на играча и така ще следим, кои са използвани и кои не.

Answers е другото слабо множество, което ще се определя чрез номер на отговор и номер на въпроса

Questions (id, text, type, difficulty, topic\_id) **is\_from** е оптимизирана

Answers (question\_id, answer\_id, correctness)

Topics (id, name)

Users (id, points, email, password, username)

Games (user\_id, game\_id, fifty\_fifty, help\_from\_friend, audience\_help)

Has (game\_id, answer\_id)

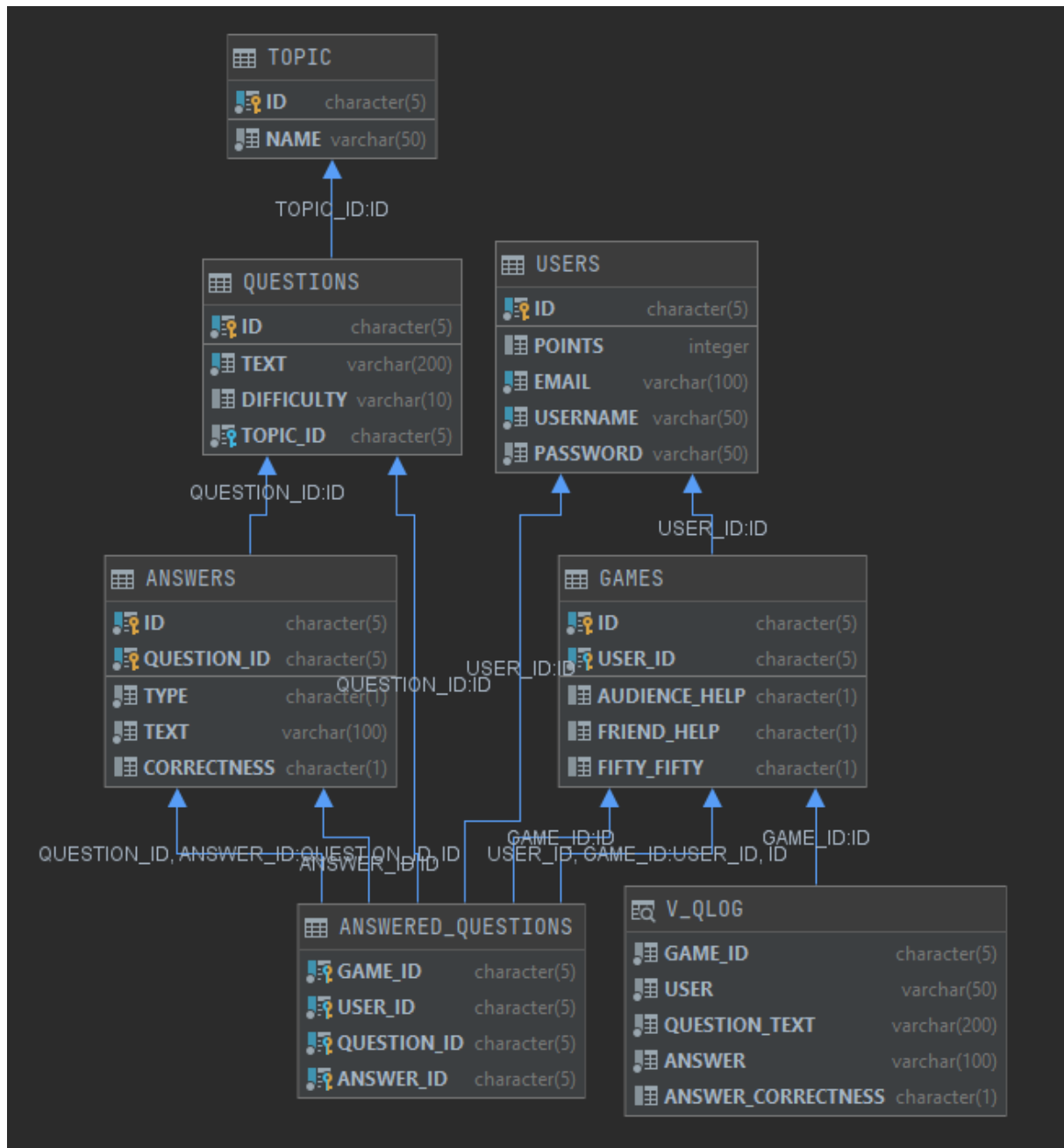
## 9. ФЗ и Нормализация

За горната схема на БД са в сила следните зависимости

- ФЗ-1: id -> text, type, difficulty, topic\_id (Questions)
- ФЗ-2: questions\_id, game\_id -> correctness (Answers)
- ФЗ-3: id -> name (Topics)
- ФЗ-4: id -> username, password, email, points (Users)
- ФЗ-5: user\_id, game\_id -> fifty\_fifty, help\_from\_friend, audience\_help

За всички ФЗ на релациите е в сила, че в лявата част се намира суперключ за релацията, следователно всички релации са в НФБК. Не се налага да правим нищо допълнително.

## 10. Схема на базата от данни



## 11. Приложение

