

Databas

Lagringslösningen har valts till en MySQL-databas. Detta erbjuds genom webbhotellet byethost, vilket är en gratislösning som erbjuder upp till 1000GB i lagringsutrymme. Tjänsten har stöd för lagring i flera MySQL-databaser och erbjuder också möjligheten till att använda phpMyAdmin för att administrera databasen i webbläsaren.

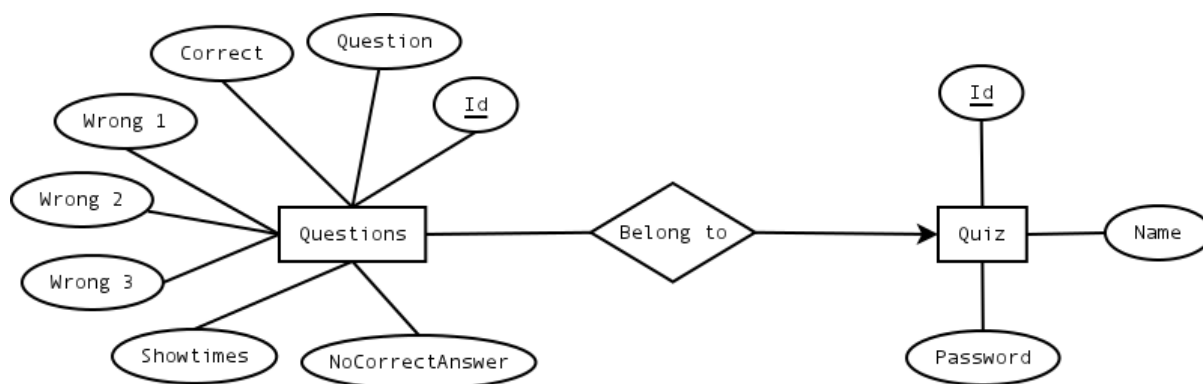
Inloggningsinformation

Nedan återfinns den information som behövs för att komma åt databasen och de olika tabellerna. Efter inloggning hänvisas läsaren till "phpMyadmin" för att därefter ges möjligheten att ansluta till databasen.

- Login page: <http://panel.byethost.com>
- Username: b7_16108318
- Password: ILMkandidat
- Database name: b7_16108318_ILM

Schema för databas

Nedan beskrivs strukturen på databasen som nyttjas av applikationen. Databasen består av två tabeller, en för frågor och en för quiz.



Figur 1. ER-diagram över databasens struktur.

Tabeller:

- Questions
- Quiz

Questions

- _Id:varchar(255)
- Question:varchar(10000)
- Correct:varchar(1000)
- Wrong1:varchar(1000)
- Wrong2:varchar(1000)
- Wrong3:varchar(1000)
- Showtimes:varchar(11)
- NoCorrectAnswer:varchar(11)

- QuizId:varchar(255)

Quiz

- _Id:varchar(255)
- Name:varchar(255)
- Password:varchar(255)

För frågor används *Id* för att identifiera en specifik fråga. *Question* syftar på själva frågan medan *Correct* är det korrekta alternativet. *Wrong 1-3* utgör möjligheten för upp till tre kompletterande alternativ, vilka är felaktiga svar på frågan. *Showtimes* motsvarar antalet gånger en fråga har visats och *NoCorrectAnswer* är antalet gånger en fråga besvarats korrekt. *QuizId* visar vilket specifikt quiz som frågan tillhör och är kopplade till *Id* i Quiz-tabellen. Likt för tabellen med för frågor används *Id* för identifiering. Namnet på ett quiz återfinns i *Name* och *Password* tillåter en quiz-skapare att specificera ett lösenord för ett quiz.

Kommunikation mellan applikation och databas

Applikationen kommunicerar med databasen via PHP-filer med innehållande SQL-kod. PHP-filerna är uppladdade på projektets domän på webbhotellet och kontaktas således från applikationen med hjälp av filernas specifika url. Om exempelvis nedladdning av alla frågor är intressant kontaktas databasen genom PHP-filen `getQuestions.php`, vilken återfinns på <http://www.ilmkandidat.byethost7.com/getQuestions.php>. Klasserna `HttpClient`, `HttpPost`, `HttpResponse` och `HttpEntity` används för att upprätta kopplingen till PHP-filerna, skicka med eventuell data samt hämta den data som PHP-filerna eventuellt plockat ut från databasen. Den data som skickas med representeras av en lista med "namn-värde"-objekt. Listan motsvarar en rad i en tabell i databasen, där namn och värde alltså syftar på kolumnnamn respektive det specifika värdet för raden.

PHP-filerna kan beskrivas som bearbetningspunkter vilka sköter all manipulering av databaser och de ingående tabellerna. Filerna innehåller SQL-kommandon och dessa är utformade utefter PHP-filernas respektive syfte. Efter att data från applikationen behandlats av servern, genom de olika PHP-filerna, returneras svar. Svaren är den data som applikationen har möjlighet att utnyttja om önskat. Svaren representeras inledningsvis genom `HttpEntity`-objekt vilka hanteras genom content streams i Java.

Utan vidare bearbetning är svaren utformade i olika strängar. Dessa är formaterade för att kunna konverteras till `JSONObject`, vilka är oordnade "namn-värde"-objekt. Namn, vilket motsvarar kolumnnamn, och värden, vilka är de specifika värdena för kolumner, är integrerade i strängarna. För att särskilja på olika "namn-värde"-objekt separeras de med kommatecken och de ingående namnen och tillhörande värde särskiljs med kolon. Denna specifika struktur, vilken kan beskrivas som ett `JSONObject`:s externa struktur, möjliggör konverteringen av strängar till `JSONObject`. Detta medför i sin tur interna möjligheter att extrahera specifik data ur `JSONObject` genom `get`-metoder samt manipulera data genom `set`-metoder.

PHP-filer

Nedan återfinns en kort sammanställning av de olika PHP-filerna som används vid kommunikation mellan applikation och databas.

addQuestion.php

- *Syfte*: Manipulerar tabellen Questions genom att lägga till rader
- *Indata*: ID för fråga, fråga, korrekt svar, tre stycken felaktiga svar, antal gånger som frågan visats, antal gånger som frågan besvarats på ett korrekt sätt samt ID för quiz

addQuiz.php

- *Syfte*: Manipulerar tabellen Quiz genom att lägga till rader
- *Indata*: ID för quiz, namn på quiz och lösenord

getQuizQuestions.php

- *Syfte*: Extraherar data från tabellen Questions genom att hämta specifika quizfrågor
- *Indata*: ID för quiz

getQuestions.php

- *Syfte*: Extraherar data från tabellen Questions

getQuizzes.php

- *Syfte*: Extraherar data från tabellen Quiz

updateShow.php

- *Syfte*: Manipulerar tabellen Questions genom att uppdatera rader
- *Indata*: ID för fråga och antal gånger som frågan visats

updateNoCorrectAnswer.php

- *Syfte*: Manipulerar tabellen Questions genom att uppdatera rader
- *Indata*: ID för fråga och antal gånger som frågan besvarats på korrekt sätt