

```
CREATE TABLE Departments(  
    DepartmentId VARCHAR(7),  
    Name VARCHAR(255),  
    PRIMARY KEY(DepartmentId)  
);
```

```
CREATE TABLE Courses(  
    CourseId Int,  
    DepartmentId VARCHAR(7),  
    Number Int,  
    Name VARCHAR(255),  
    Credits VARCHAR(50),  
    GenEd VARCHAR(63),  
    FOREIGN KEY(DepartmentId) REFERENCES Departments(DepartmentId),  
    PRIMARY KEY(CourseId)  
);
```

```
CREATE TABLE CourseDescription(  
    CourseId Int,  
    Description VARCHAR(1023),  
    Instructors VARCHAR(255),  
    FOREIGN KEY(CourseId) REFERENCES Courses(CourseId),  
    PRIMARY KEY(CourseId)  
);
```

```
CREATE TABLE User(  
    UserId VARCHAR(31),  
    Email VARCHAR(63),  
    EncryptedPassword VARCHAR(255),  
    DateJoined DATE,  
    MostGuessed Int,  
    AvgGuesses Decimal,  
    FOREIGN KEY(MostGuessed) REFERENCES Courses(CourseId),  
    PRIMARY KEY(UserId)  
);
```

```
CREATE TABLE DailyClass(  
    CurrentDate DATE,  
    CorrectCourseId Int,  
    MostCommonWrong Int,
```

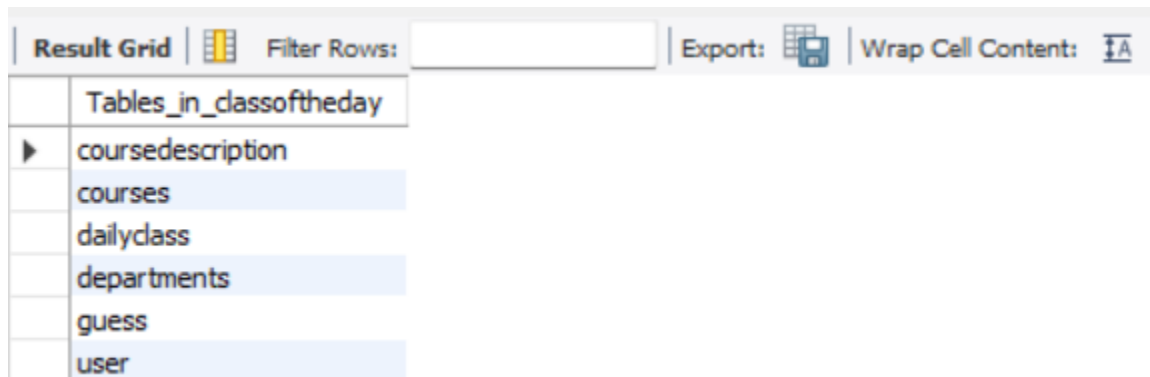
```

    TotalAvgGuesses Decimal,
    FOREIGN KEY(CorrectCourseId) REFERENCES Courses(CourseId),
    PRIMARY KEY(CurrentDate)
);

CREATE TABLE Guess(
    GuessId Int,
    UserId VARCHAR(31),
    CurrentDate DATE,
    CourseId INT,
    FOREIGN KEY(UserId) REFERENCES User(UserId),
    FOREIGN KEY(CurrentDate) REFERENCES DailyClass(CurrentDate),
    FOREIGN KEY(CourseId) REFERENCES Courses(CourseId),
    PRIMARY KEY(GuessID)
);

```

Connection Screenshot:



Query 1: Gets the emails of all users who guessed the correct course on their first try

```

SELECT Email
FROM User
JOIN
Guess ON User.UserId = Guess.UserId
JOIN
DailyClass ON Guess.CurrentDate = DailyClass.CurrentDate
WHERE Guess.CurrentDate = '2024-29-10' AND Guess.CourseId = DailyClass.CorrectCourseId
AND
(SELECT (COUNT(GuessId)) FROM Guess g WHERE g.CurrentDate = Guess.CurrentDate) =
1

```

GROUP BY User.UserId;

Query 2: Gets the most commonly guessed departments on a given day

```
SELECT Departments.DepartmentId, COUNT(GuessId) AS TimesGuessed
FROM Departments
JOIN
Courses ON Departments.DepartmentId = Courses.DepartmentId
JOIN
Guess ON Guess.CourseId = Courses.CourseId
WHERE Guess.CurrentDate = '2024-29-10'
GROUP BY Departments.DepartmentId
ORDER BY TimesGuessed DESC;
```

Query 3: Gets the average number of guesses per user across all days

```
SELECT User.UserId, User.Email, AVG(UserGuess.GuessCount) AS AvgGuesses
FROM User
JOIN
(
SELECT Guess.UserId, Guess.CurrentDate, COUNT(Guess.GuessId) AS GuessCount
FROM Guess
GROUP BY Guess.UserId, Guess.CurrentDate
) AS UserGuess ON User.UserId = UserGuess.UserId
GROUP BY User.UserId;
```

Query 4: Gets the top 5 most frequently guessed courses overall

```
SELECT Courses.CourseId, Courses.Name, Departments.Name, COUNT(Guess.GuessId) AS
FrequentGuess
FROM Courses
JOIN
Guess ON Courses.CourseId = Guess.CourseId
JOIN Departments ON Courses.DepartmentId = Departments.DepartmentId
GROUP BY Courses.CourseId, Courses.Name
ORDER BY FrequentGuess DESC
LIMIT 5;
```

Query 1 - Attempted Indexing Designs:

```
EXPLAIN ANALYZE SELECT Email
FROM User
JOIN
Guess ON User.UserId = Guess.UserId
JOIN
DailyClass ON Guess.CurrentDate = DailyClass.CurrentDate
WHERE Guess.CurrentDate = '2024-29-10' AND Guess.CourseId =
DailyClass.CorrectCourseId AND
(SELECT (COUNT(GuessId)) FROM Guess g WHERE g.CurrentDate =
Guess.CurrentDate) = 1
GROUP BY User.UserId;
```

- CREATE INDEX Current_Date ON Guess(CurrentDate);
- CREATE INDEX Course_Id ON Guess(CourseId);
- CREATE INDEX Correct_Course_Id ON DailyClass(CorrectCourseId);

Query 2 - Attempted Indexing Designs:

```
EXPLAIN ANALYZE SELECT Departments.DepartmentId, COUNT(GuessId)
AS TimesGuessed
FROM Departments
JOIN
Courses ON Departments.DepartmentId = Courses.DepartmentId
JOIN
Guess ON Guess.CourseId = Courses.CourseId
WHERE Guess.CurrentDate = '2024-29-10'
GROUP BY Departments.DepartmentId
ORDER BY TimesGuessed DESC;
```

- CREATE INDEX Current_Date ON Guess(CurrentDate);
- CREATE INDEX Course_Id ON Guess(CourseId);

- CREATE INDEX Dept_Id ON Courses(DepartmentId);

Query 3 - Attempted Indexing Designs:

```
EXPLAIN ANALYZE SELECT User.UserId, User.Email,  
AVG(UserGuess.GuessCount) AS AvgGuesses  
FROM User  
JOIN  
(  
SELECT Guess.UserId, Guess.CurrentDate, COUNT(Guess.GuessId) AS  
GuessCount  
FROM Guess  
GROUP BY Guess.UserId, Guess.CurrentDate  
) AS UserGuess ON User.UserId = UserGuess.UserId  
GROUP BY User.UserId;
```

- CREATE INDEX Current_Date ON Guess(CurrentDate);
- CREATE INDEX User_Id ON Guess(UserId);

Query 4 - Attempted Indexing Designs:

```
EXPLAIN ANALYZE SELECT Courses.CourseId, Courses.Name,  
Departments.Name, COUNT(Guess.GuessId) AS FrequentGuess  
FROM Courses  
JOIN  
Guess ON Courses.CourseId = Guess.CourseId  
JOIN Departments ON Courses.DepartmentId = Departments.DepartmentId  
GROUP BY Courses.CourseId, Courses.Name  
ORDER BY FrequentGuess DESC  
LIMIT 5;
```

- `CREATE INDEX Name ON Courses(Name);`
- `CREATE INDEX Course_Id ON Guess(CourseId);`
- `CREATE INDEX Dept_Id ON Courses(DepartmentId);`