



پروژه دوم برنامه سازی پیشرفته

دکتر فرید فیضی

ترم 14012

مدیریت پایگاه داده



مقدمه

هدف این پروژه پیاده سازی تعدادی از دستورات مرسوم برای انجام چهار عمل اصلی :
CREATE , READ , UPDATE , DELETE که به آن به اصطلاح CRUD می گویند.

هدف

آشنایی با روش های خواندن و نوشتن فایل در جاوا و همچنین آشنایی کوچکی با نحوه عملکرد زبان تحت پایگاه داده Mysql
در پیاده سازی پروژه دقت کنید که تا جای ممکن کدتون رو انعطاف پذیر بنویسید که در پروژه های 3 و 4 هم بتوانید ازش استفاده کنید.

ساختار کلی

در ادامه با دستورات آشنا میشوید .
هر دستور توضیحات و مثالی برای درک بیشتر داره .
- بدیهی است که در ابتدا باید دستور ساخت جداول داده شود و در صورت وارد شدن دستور دیگری با پیامی با محتوا مناسب نمایش داده شود.

Query 1.1

CREATE TABLE

Syntax

```
CREATE TABLE table_name (column1,column2,column3,...);
```

Example

```
CREATE TABLE Persons (PersonID,LastName,FirstName,Address,City);
```

در صورت اجرا دستور باید خروجی مانند زیر در فایلی با پسوند .txt نوشته شود.
برای جداسازی هر فیلد میتوانید از جداکننده هایی مانند "|" در فایل خود استفاده کنید.

PersonID	LastName	FirstName	Address	City

Query 1.2

DROP TABLE

Syntax

```
DROP TABLE table_name;
```

Example

```
DROP TABLE persons;
```

در صورت اجرا دستور باید فایل persons.txt پاک شود.

در ادامه برای اجرای دستورات از اطلاعات لینک زیر استفاده می کنیم :

https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_select_all

وارد لینک بالا شوید و روی Run SQL کلیک کنید تا محتوای فایل را ببینید.

فایل با نام Customers.txt داریم که حاوی اطلاعات بالا می باشد. (دقت شود که به بزرگی و کوچکی حروف حساس نیستیم)

Query 2.1

SELECT

Syntax

```
SELECT column1, column2, ... FROM table_name;
```

Example

```
SELECT CustomerName, City, Country FROM Customers;
```

Output

https://www.w3schools.com/MySQL/trymysql.asp?filename=try_sql_select_columns

Query 2.2

ALIASES

Syntax

```
SELECT column_name AS alias_name  
FROM table_name;
```

Example

```
SELECT CustomerID AS ID, CustomerName AS Customer FROM Customers;
```

Output

https://www.w3schools.com/MySQL/trymysql.asp?filename=try_sql_select_alias_column0

دقت شود که این دستور صرفاً برای نمایش در خروجی کاربرد دارد و تغییری در ساختار جدول‌ها ایجاد نمی‌کند.

Query 2.3

WHERE

Syntax

```
SELECT column1, column2, ... FROM table_name WHERE condition;
```

Example

```
SELECT * FROM Customers WHERE Country='Mexico';
```

Output

https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_select_where

نکته: کاربر با وارد کردن * می تواند تمام ستون ها را انتخاب کند.

Query 2.4

AND

Syntax

```
SELECT column1, column2, ... FROM table_name WHERE condition1 AND  
condition2 AND condition3 ...;
```

Example

```
SELECT * FROM Customers WHERE Country = 'Germany' AND City = 'Berlin';
```

Output

https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_select_where_and

Query 2.5

OR

Syntax

```
SELECT column1, column2, ... FROM table_name WHERE condition1 OR  
condition2 OR condition3 ...;
```

Example

```
SELECT * FROM Customers WHERE City = 'Berlin' OR City = 'Stuttgart';
```

Output

https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_select_where_or

Query 2.6

NOT

Syntax

```
SELECT column1, column2, ... FROM table_name WHERE NOT condition;
```

Example

```
SELECT * FROM Customers WHERE NOT Country = 'Germany';
```

Output

https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_select_where_not

Combining AND, OR and NOT

Example

```
SELECT * FROM Customers WHERE Country = 'Germany' AND (City = 'Berlin' OR  
City = 'Stuttgart');
```

Output

https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_select_where_and_or

Query 3.1

INSERT INTO

Syntax

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

Example

```
INSERT INTO Customers (CustomerName, ContactName, Address, City,  
PostalCode, Country)  
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen  
21', 'Stavanger', '4006', 'Norway');
```

Output

https://www.w3schools.com/sql/trysql.asp?filename=trysql_insert_colname

در مواقعی که بخواهیم به تمام ستون ها مقداری را درج کنیم، می توانیم از این دستور نیز استفاده کنیم:

Query 3.1.1

INSERT INTO

Syntax

```
INSERT INTO table_name  
VALUES (value1, value2, value3, ...);
```

نیازی به نوشتن نام تک تک ستون ها نیست و مقادیر به ترتیب از چپ به راست قرار می گیرند.

Query 4.1

UPDATE

Syntax

```
UPDATE table_name  
SET column1 = value1, column2 = value2...,  
WHERE condition;
```

Example 1

```
UPDATE Customers  
SET ContactName = 'Alfred Schmidt', City = 'Frankfurt'  
WHERE CustomerID = 1;
```

Example 2

```
UPDATE Customers  
SET PostalCode = 00000  
WHERE Country = 'Mexico';
```

قرار گیری کلمه کلیدی WHARE به صورت optional است و دستور می تواند بدون WHERE نیز بیان شود.

Example 3

```
UPDATE Customers  
SET PostalCode = 00000;
```


Query 5.1

DELETE

Syntax

```
DELETE FROM table_name WHERE condition;
```

Example 1

```
DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';
```

برای حذف تمامی رکورد ها از دستور زیر استفاده می شود :

Example 2

```
DELETE FROM Customers;
```

Aggregate Functions

Query 6.1

COUNT

Syntax

```
SELECT COUNT(column_name)  
FROM table_name  
WHERE condition;
```

Example

```
SELECT COUNT(ProductID) FROM Products;
```

Output

https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_select_count

Query 6.2

AVG

Syntax

```
SELECT AVG(column_name)  
FROM table_name  
WHERE condition;
```

Example

```
SELECT AVG(Price) FROM Products;
```

Output

https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_select_avg

Query 6.3

SUM

Syntax

```
SELECT SUM(column_name)  
FROM table_name  
WHERE condition;
```

Example

```
SELECT SUM(Quantity) FROM OrderDetails;
```

Output

https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_select_sum

Query 7.1

LIKE

Syntax

```
SELECT column1, column2...,  
FROM table_name  
WHERE columnN LIKE pattern;
```

Example

```
SELECT * FROM Customers WHERE CustomerName LIKE '%a';
```

Output

https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_select_like_ending

Query 7.2

REGEX

Syntax

```
SELECT * FROM table_name  
WHERE column REGEXP pattern;
```

Example

```
SELECT * FROM author WHERE aut_name REGEXP '^w';
```

Query 8.1

ORDER BY

Syntax

```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1, column2, ... ASC|DESC;
```

Example

```
SELECT * FROM Customers ORDER BY Country;
```

Output

https://www.w3schools.com/MySQL/trymysql.asp?filename=trysql_select_order_by

دقت شود که ASC|DESC می تواند داده نشود و در این صورت به صورت پیش فرض ASC در نظر گرفته می شود.

روند اجرا

در صورت اجرای برنامه، باید منتظر ورود دستورات باشیم و در صورت نیاز خروجی مربوطه را نمایش دهیم و یا تغییرات لازم را در فایل های مربوطه اعمال کنیم.

هر دستور در یک خط وارد می شود و تمامی قسمت های آن در یک خط قرار می گیرند.

نمره مثبت

1- پیاده سازی دستورات max، min، limit، right join، left join، inner join به صورت امتیازی بوده و برای آشنایی با این دستورات می توانید از لینک زیر استفاده کنید :

https://www.w3schools.com/MySQL/mysql_sql.asp

2- برنامه در صورت وارد شدن syntax نادرست دچار قطعی و خطا نشود و پیام مناسبی مانند invalid command نمایش داده شود.

همچنین نیاز است نکاتی تحت عنوان tip به کاربر نمایش داده شود تا اشتباه دستور خود را متوجه شود.

3- کاربر بتواند ادرس فایلی که در آن تعدادی دستور نوشته شده اند و دستورات با سمیکالن از هم جدا شده اند را به برنامه بدهد و همه دستورات گفته شده اجرا شوند.

4- همانطور که در syntax دستورات قابل مشاهده است اکثر دستورات توانایی ترکیب با یکدیگر را دارند. اگر برنامه طوری نوشته شود که توانایی اجرای هر نوع ترکیب از دستورات را داشته باشد (در صورت امکان) دارای نمره مثبت خواهد بود.

نکات :

رابط کاربری گرافیکی فقط باید با Java Swing پیاده سازی شود.

هرگونه ایده و خلاقیت جدید در پروژه با هماهنگی با دستیاران آموزشی نمره مثبت به همراه دارد.

هیچ توجیهی در استفاده از زبان های برنامه نویسی دیگر وجود نداشته و برابر با تحویل ندادن پروژه ، نمره **صفر** به دانشجو تعلق میگیرد.

پروژه تنها به صورت انفرادی قابل انجام بوده و در صورت وجود هرگونه تشابه بین دو کد یا عدم تسلط به روند، **نمره منفی** به دانشجویمان داده می شود.

برای تسلط بهتر و کسب نمره خوب در این درس، به انجام پروژه های اضافه بر پروژه های ترم اقدام کنید.

کامنت نویسی مناسب و رعایت دندانه گذاری الزامی است.

حسین رهبرطلب – مهدی آقائی