1 Oracle Primer

After you double-click on installation file, and then click-next, click-next, click-next a few dozen times, you've managed to install Oracle. Congratulations!

How the heck do we use it? One approach is to use the Oracle provided GUI tools (such as the oracle's website thingie). Another approach (which as *advanced* users you should be using) is to use the command line. Why is command line better long term? Well, it can be scripted, and you can easily pipe data into and out of commands. Most data warehouse applications of databases value automation and simplicity above one-time use user-friendliness.

2 Oracle Express Edition

Recently (a few years ago), Oracle releasd a free 'Express Edition', which is primarily targetted towards students, developers, and anyone else interested in learning how to use Oracle. It has strict limits on how big the database can be, how many CPUs can be used, and how much RAM Oracle is allowed, etc. Most of these limitations eliminate the Express Edition (XE) from production use within any serious operation. It does however make it a perfect vihicle for learning Oracle (ie: it's a *free* download!).

After installing OracleXE, the server binds to your local IP, ie: `127.0.0.1`, port 1521 (the default), and has a service name of "XE".

The TNS entry (ie: sort of like a URL) for it is:

```
XE = (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = 127.0.0.1) (PORT = 1521))
(CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = XE)))
```

Note that you can add this to the `tnsnames.ora` file (search for it under Oracle home folder), or you can use TNS entry directly when connecting. For example, to login into Oracle from command line you might run:

```
sqlplus "system/manager@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = 127.0.0.1)
(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = XE)))"
```

Note that you cannot have spaces in the TNS part.

For Java database tools, this means a JDBC URL of:

```
jdbc:oracle:thin:@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST=127.0.0.1)
(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = XE)))
```

Again, no spaces in the URL.

The JDBC driver is `oracle.jdbc.driver.OracleDriver`, and is found in `ojdbc14.jar` file, which is somewhere in oracle installation folder (or online).

The `system` user account (ie: oracle's admin account) gets a password during installation (during those click-next, click-next steps).

3 Oracle Regular Edition

Instaling Oracle (regular edition) is just as easy as installing the Express Edition. Slight differences usually revolve around Oracle *requiring* a static IP address to bind to. This can

be a major hassle if you have DHCP client on your computer (ie: get a new IP everytime you reboot).

An easy solution for Windows (Unix folks can figure this one out themselves), is to add a 'dummy' loopback network adapter. You can google "installing a loopback adapter" for exact instructions, but the gist for Windows XP goes something like this: go to control panel to add hardware; 'yes, the hardware is already connected', then pick a 'new hardware device', 'install manually', pick a 'network adapters' category, then pick Microsoft Loopback Adapter. Click 'ok', etc. This should've added another 'LAN' connection in your 'Network Connections' (device name will have 'Microsoft Loopback' in it). Right click it, and configure that adapter to have a non-routable IP, such as 10.10.10.10, etc. That's it.

3.1 Default Password

On older Oracle installers, the user didn't specify the `system` (ie: admin) account password. The default password is `manager`. You are highly encouraged to change it.

4 Working with Users

Playing with Oracle as the `system` user is stupid. Things *will* go wrong! For various safety and security reasons, it's a very good idea to create a low priviledge database account to use.

4.1 Adding User Accounts

To add a user, simply use the 'create user' command:
create user `phreak` **identified by** `abc123` **default tablespace** `users` **temporary tablespace** `temp;`
Here, we've added the 'phreak' user to our database, with password 'abc123', using 'users' tablespace (where data is stored), etc.

In order for the user to login, they must be granted `connect` and `resource` roles:
grant `connect, resource` **to** `phreak;`
Note that some tutorials also grant "create session", which is technically already included in the "connect" role (ie: you shouldn't need to explicitly grant 'create session').

4.2 Altering User Accounts

4.2.1 Changing Passwords

To change password, just do:
alter user `phreak` **identified by** `abc123;`
User can also always type: '`password`' in `sqlplus` to do this interactively.

4.2.2 Expire User Password

Expire current user's password to force user to change password next time they login:
alter user `phreak` **password expire**;

4.2.3 Removing Users

drop user `phreak` **cascade**;

4.2.4 Make User DBA (admin)

grant `dba` **to** `phreak`;

4.2.5 Manage User's Quota

Tablespaces (place where data is stored) can have a quota. To setup no-quota (ie: unlimited) space allowance:
alter user `phreak` **quota** `unlimited` **on** `users`;
This sets up `unlimited` quota for user `phreak` on `users` tablespace.

To specify a limit, just replace the unlimited keyword with some value:
alter user `phreak` **quota** `20M` **on** `users`;

4.2.6 Lock/Unlock Accounts

Locking an account means the user cannot login, etc.
alter user `phreak` **account lock**;
or
alter user `phreak` **account unlock**;

4.2.7 scott/tiger Account

The scott/tiger (username: scott, password: tiger) account is considered a 'test' account, and unless you're in a very unsecure environment, it is recommended that you unlock it (it allows automated scripts to connect to the database and determine whether it's up or down, or do 'anonymous' queries; like asking the database to provide a timestamp, etc.)

4.3 Useful Permissions

Depending on the users, some should be able to create tables, procedures, etc. (or even just select certain tables):

```
grant create table to phreak;
grant create view to phreak;
grant create procedure to phreak;
grant create sequence to phreak;
grant create trigger to phreak;
```

Obviously you can also grant specific permissions for different database objects (tables, views, etc.):
grant `select` **on** `blah` **to** `phreak;`
will let user phreak select data from table (or view) blah.

5 Tablespaces

A tablespace is where all the data is stored. There are a few default ones (users, temp, etc.), but you can create new ones via:
create tablespace `corpdata` **datafile** `'/oradata/CDB/data01.dbf'` **size** `20M` **autoextend on next** `10M` **maxsize** `50M`
This creates a tablespace named 'corpdata', which will be stored in `/oradata/CDB/data01.dbf` file, will start with size 20Mb, go up by 10Mb (as database grows), and finish off at 50Mb.

6 Session

There are a bunch of session level parameters that you can modify. Below are some that seem useful:

6.1 Recyclebin

Recent versions of Oracle appear to have a 'recyclebin' feature. Essentially dropping tables doesn't actually drop them, but renames them (and hides the new name in recyclebin). The space is eventually automaticaly cleaned up, but only when you start to run out of it.

You can really 'drop' a table by purging the recyclebin once in a while:
purge recyclebin;
Or you can turn it off for the whole session right after you login:
alter session set recyclebin=off

6.2 Date Format

Oracle seems to have this weird date format, 'DD-MON-YY', so August 29th, 1997, for example, would be '29-AUG-97'. This may be something *you* enjoy, but it seems the 'YYYYMMDD' format is much easier to deal with programmatically (ie: inside a program). That same date would be '19970829'.

You can tell Oracle that you'd like to use this date format, by altering the session:
alter session set `nls_date_format='YYYYMMDD';`