```
********************
*** 10g
********************


*** In session (1)

SQL> CREATE TABLE bowie_stuff AS SELECT rownum id, 'David Bowie' name FROM
dual CONNECT BY LEVEL <= 10000;

Table created.

SQL> CREATE INDEX bowie_stuff_i ON bowie_stuff(id);

Index created.


*** In other session (2)

SQL> INSERT INTO bowie_stuff VALUES (10001, 'Pink Floyd');

1 row created.


*** In the orginal session (1)

SQL> ALTER INDEX bowie_stuff_i REBUILD ONLINE;

*** session hangs (due to inability to get table lock due to session 2)



*** In yet another session (3)

SQL> INSERT INTO bowie_stuff VALUES (10002, 'Radiohead');

*** It now hangs due to lock from session 1 , as would a transaction on the
bowie_stuff table in session 4 and 5 and 6 and   ...


*** Now commit in session (2)

SQL> COMMIT;

Commit complete.



*** releases the lock in session 3 and the index rebuild is free to proceed
but it will eventually get stuck again as it now requires another lock to
complete the rebuild process ...

*** In session 2, perform another insert before session 3 commits ...

SQL> INSERT INTO bowie_stuff VALUES (10003, 'Iggy Pop');

*** and now it in turn hangs due to the rebuild needing the second table
lock


*** perform the commit in session (3)
```

```
SQL> commit;

Commit complete.
```

and it allows the rebuild in session 1 to finally finish and in turn allows the update in session 2 to then be released and complete as well

*** So a rebuild requires a lock at the start and at the end of the index rebuild process, even if performed ONLINE

*** These locks in turn cause other concurrent transactions on the table to hang as well

****
*** 11g
****

*** In session (1)

```
SQL> CREATE TABLE bowie_stuff AS SELECT rownum id, 'David Bowie' name FROM
dual CONNECT BY LEVEL <= 10000;

Table created.

SQL> CREATE INDEX bowie_stuff_i ON bowie_stuff(id);

Index created.
```

*** In other session (2)

```
SQL> INSERT INTO bowie_stuff VALUES (10001, 'Pink Floyd');

1 row created.
```

*** In the orginal session (1)

```
SQL> ALTER INDEX bowie_stuff_i REBUILD ONLINE;
```

session still hangs (due to inability to get table lock due to session 2)

*** In yet another session (3)

```
SQL> INSERT INTO bowie_stuff VALUES (10002, 'Radiohead');

1 row created.
```

*** Big change. This session is no longer impacted by the rebuild trying to get it's table lock. It can carry on happily ..

*** Performing a Commit in session 2 will allow the rebuild to commence but it will be stuck again with the incomplete transaction in session 3.

```
SQL> commit;

Commit complete.
```

*** Performing another insert in session 2 will complete fine as again the rebuild does not impact other transactions

```
SQL> INSERT INTO bowie_stuff VALUES (10003, 'Iggy Pop');

1 row created.
```

*** commiting the transactions in both session 2 and 3 will allow the rebuild to finally complete

```
SQL> commit; (session 2)

Commit complete.
```

```
Index altered.  (session 1).
```

*** So an online rebuild in 11g can still be impacted by concurrent transactions but it in turn will not cause locking issues for other concurrent transactions on the base table