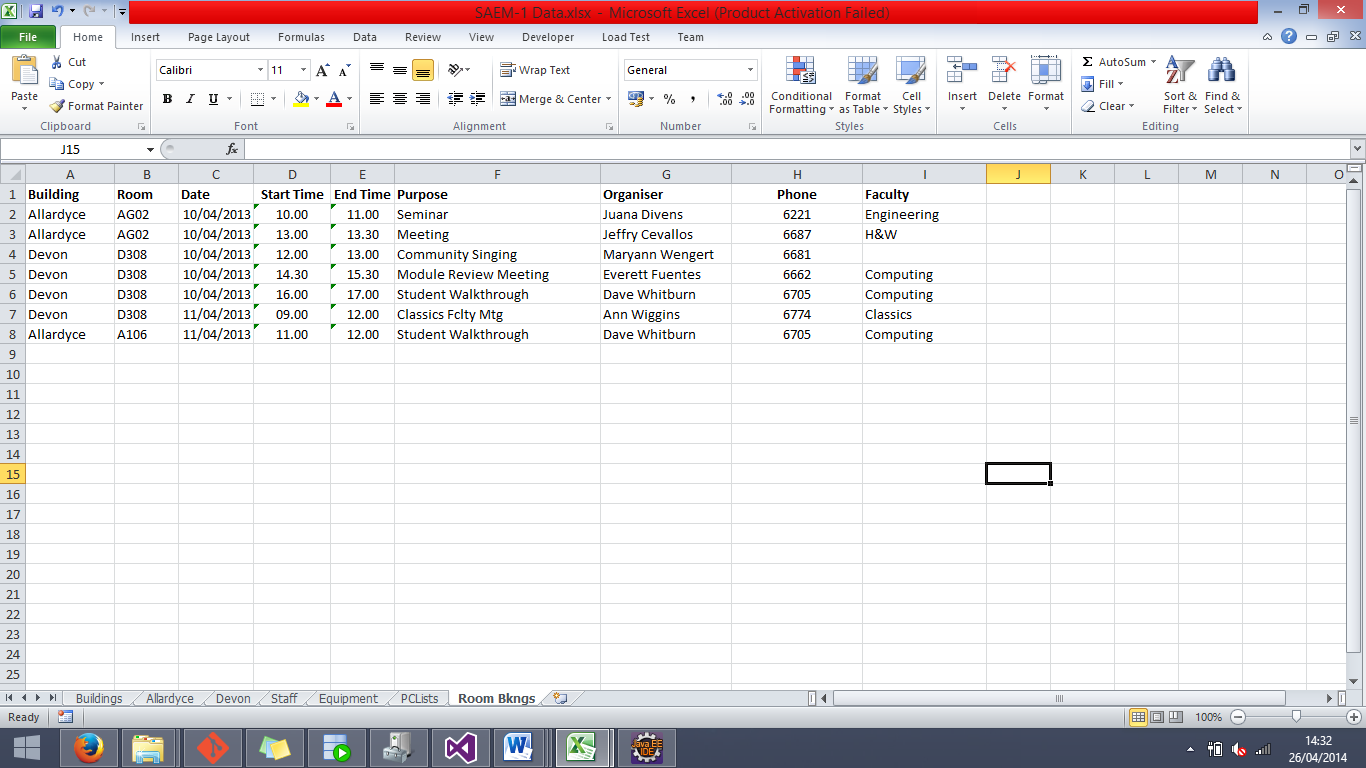
**Data Take-on Testing**

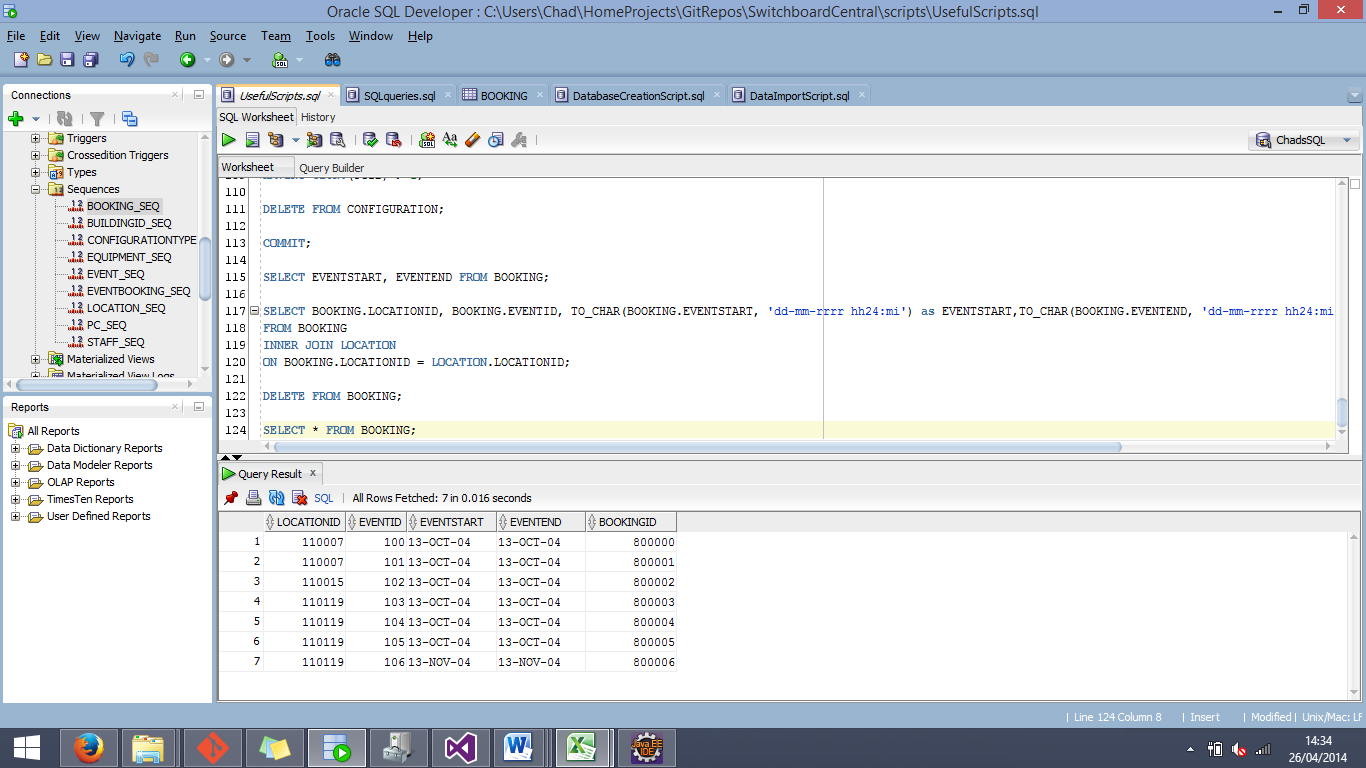
The first action we took when starting to test the data that we loaded was to look back on the source data that we were given and see if there were any aspects of it that would be able to be tested in the final database implementation to see if they were the same as they were in the source data.

As a result of that, we have compiled a series of tests that we performed on the loaded data inside of the database that could be compared to the raw data that we received in the form of an excel document. From the results of these tests, we will be able to see if the data that we have loaded is valid and correct.

This testing will be split up into two sections, manual testing, and automated testing. The manual testing will largely be done by hand, performing actions such as simply checking the row counts or checking to see if a certain piece of equipment for a certain location has been loaded correctly. The automated testing will mostly be in the form of SQL scripts that can check things that would take a much longer time if they were to be done manually.

**Manual Testing**

The first test that is performed on the loaded data is to check that the correct number of event bookings have been implemented into the database. This will be done by simply manually checking that the number of rows in the database is equal to the amount of rows in the source data.



As can be seen from the screenshots above, the amount of events in the final database is equal to the amount of data rows in the source document. There are 8 rows in the source document (8 minus the one row for column headers), and a total of 7 rows in the BOOKING database entity.

In an attempt to weave automated testing in with the manual testing, we made SQL scripts along the way that were able to perform the same action in the database, in order to more easily compare the data in the database with the source data.

SQL Script:

SELECT COUNT(\*) FROM EVENT;

------

The next series of manual tests revolved around making sure that the total amount of locations in the database is correct, and checking that each building had its correct number of locations

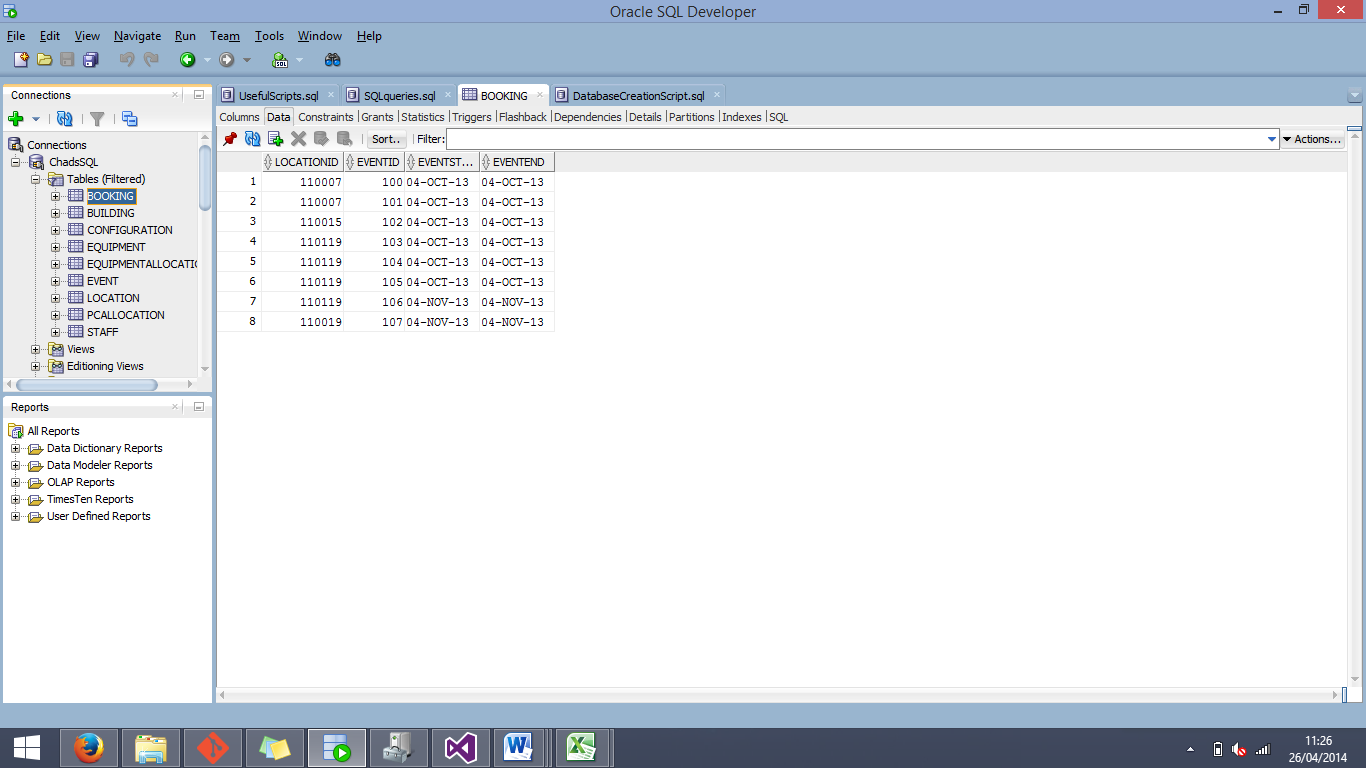
The first test was to compare the total amount of locations in the database to the cumulated amount of locations that were between the different building sheets in the excel source file.

------

Another test that we performed was one exclusively on the database, and it was to check and see if it was possible to extract the date that was inserted into the database in a suitable format.

The data from the database that we used for this exercise was the EVENTSTART and EVENTEND fields from the BOOKING entity. To aid us in this test we created a script that would need to be used if we were going to be able to successfully extract the dates in a suitable format.

The default format of the date field data is shown below.

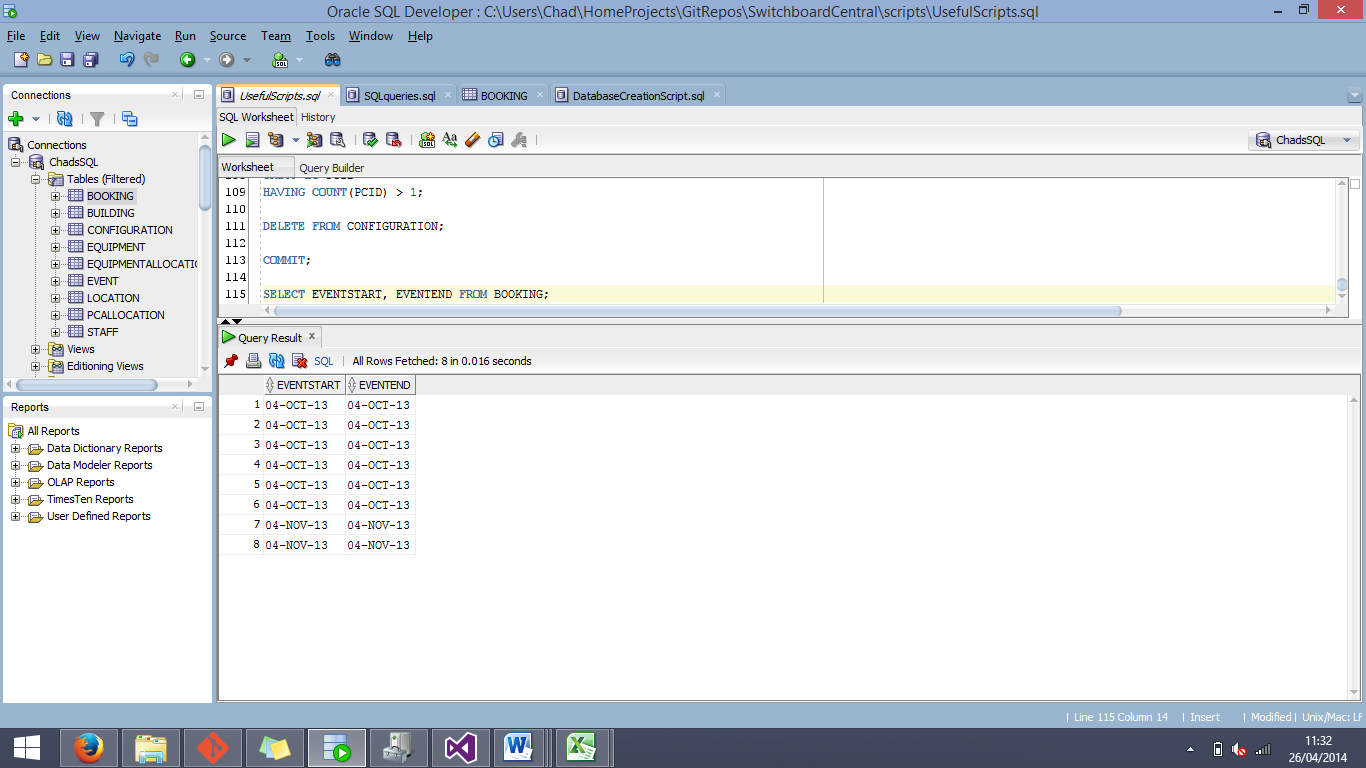


In order to see if this data would be able to show correctly by simply selecting the field, we first tested the following script:

SQL Script:

SELECT EVENTSTART, EVENTEND FROM BOOKING;

The result that was output is the following:



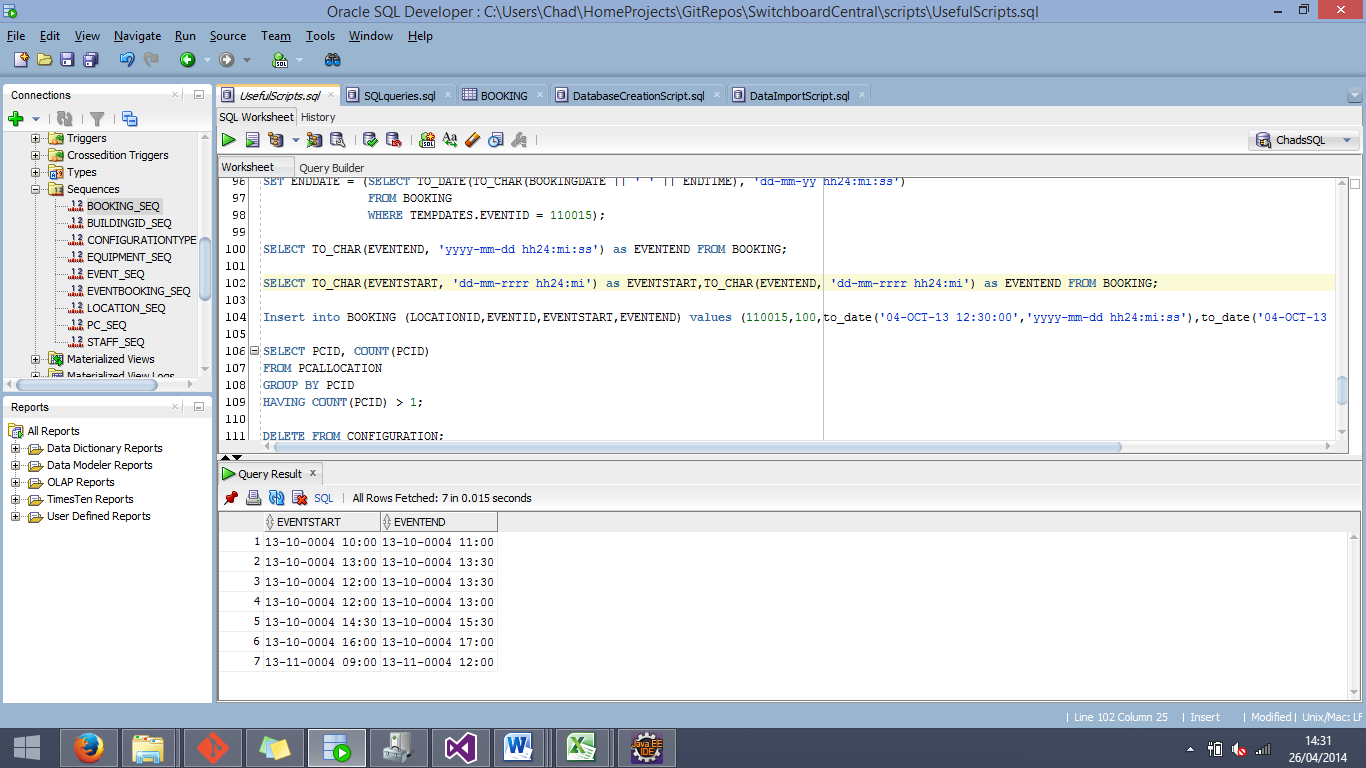
Now we know that we cannot extract the dates by normally selecting them from the BOOKING entity, we must test special methods of extracting, using functions to format the date on the way out.

To do this, we employed the TO\_CHAR() SQL function to format the outputted data to a format that would include the time as well as the date for the field.

SQL Script:

SELECT TO\_CHAR(EVENTSTART, 'dd-mm-rrrr hh24:mi') as EVENTSTART,TO\_CHAR(EVENTEND, 'dd-mm-rrrr hh24:mi') as EVENTEND FROM BOOKING;

As a result from this query, it can be seen that the data can be extrapolated in a valid format from the given date fields.



Now that it is confirmed that we are able to successfully extract the date and time from the BOOKING entity without errors, the next step will be to check and see if the data in the database correctly matches what was loaded from the source data.

To do this, we are going to combine manual checking with an SQL script to select all of the valid data from the database that would be needed to prove the authenticity of the test.

SQL Script:

SELECT BOOKINGID, EVENT.EVENTPURPOSE, TO\_CHAR(BOOKING.EVENTSTART, 'dd-mm-rrrr hh24:mi') as EVENTSTART,TO\_CHAR(BOOKING.EVENTEND, 'dd-mm-rrrr hh24:mi') as EVENTEND, LOCATION.ROOMNO

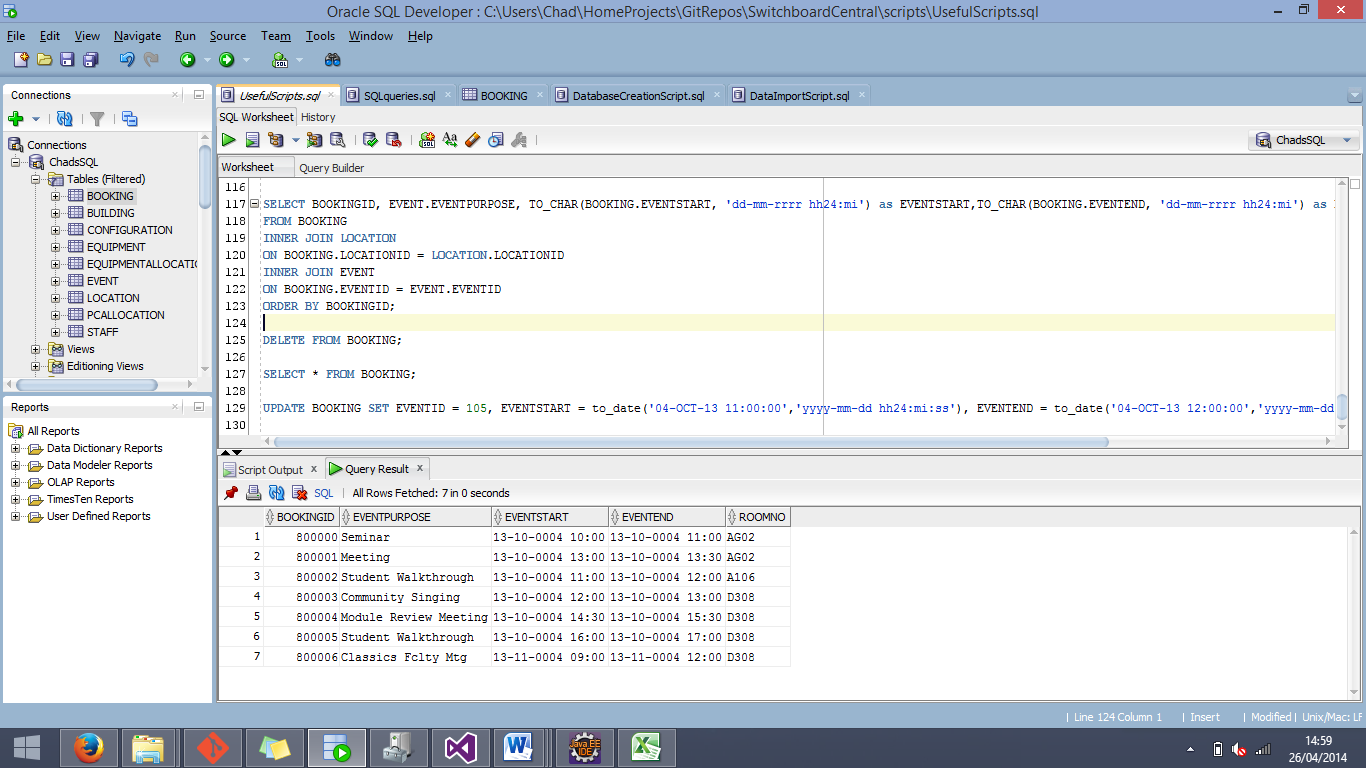
FROM BOOKING

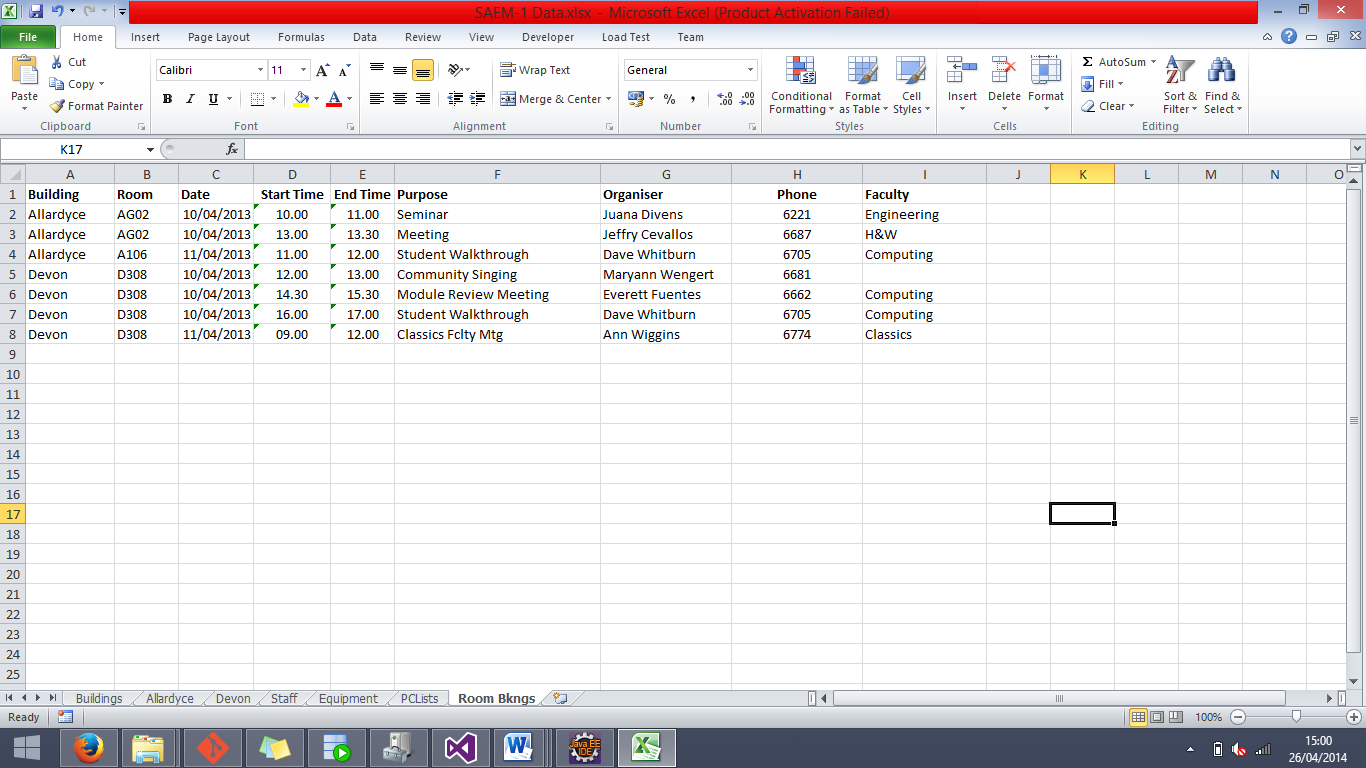
INNER JOIN LOCATION

ON BOOKING.LOCATIONID = LOCATION.LOCATIONID

INNER JOIN EVENT

ON BOOKING.EVENTID = EVENT.EVENTID;

From this script, we are able to see the ROOMNO, EVENTSTART, EVENTEND, and the IDs of the event and location that each of these bookings corresponds to.



The data from the source excel document and the data from the final database implementation is a match, proving that the loaded data is accurate in this section.