DT228/2, DT282/2 Databases I

## Week 11 Outer Joins and Views

## Olympics Tables

Download DBI-Practical-Week11.SQL in this week’s folder. This will create tables and insert data, for both the Olympics and the Games tables we are using

1. For all sports return the name of the sport and the name of all events in that sport. You should include all sports, whether they have events or not.
2. Amend the SQL for 1 so that rather than showing null in the output, for sports that have no events the statement returns ‘No events found’ (Hint: use NVL)
3. For all sports, find out how many events that sport has (Hint: use a group by statement). You should include all sports whether they have events or not.
4. Amend your SQL for 3 so that it will only show sports that do not have any events.
5. For all events, return the eventname and the number of competitors enrolled. You should include all events, whether they have competitors or not.
6. For all competitors, find out how many events they are enrolled in. You should include competitors whether they are enrolled in events or not.
7. Write the SQL to use a derived table and a left outer join to output the names of any events a competitor is enrolled in or None if they are not enrolled in any events.
8. Write the SQL to create a view called CurrentStandings which has columns called CompName and NumEvents. Use the SQL you wrote for part 6 as a basis. Once the view is created, write a select statement to find the competitors who have not competed in any events.

## Games Tables

### Outer Join

1. We want to find out for all games who has rented them showing the customer id of the person who rented them You should include the game ID in the output. You should include all games, whether they have been rented or not.
2. For all games, find out how many times each game has been rented.

You should include the game title in the output. (Hint: use a group by statement). You should include all games, whether they have been rented or not.

1. Write the SQL to create a view called GameStandings which has the columns GameName (title of the game) and NumRentals (which is the number of times a game has been rented.
2. Find details of all games that have NEVER been rented. (Hint: use a left outer join and a WHERE CLAUSE)
3. Using a right outer join, find details of all customers who have never rented a game.