CIS4301 Notes: Assignment 3 Solutions

Ryan Roden-Corrent

March 23, 2014

1 Olympic Queries

```
-- Query 1
SELECT athlete
FROM Olympics
GROUP BY athlete
ORDER BY sum(total_medals) desc -- athlete with most medals at the top
LIMIT 1;                       -- only keey the top result

-- Query 2
SELECT athlete, sum(total_medals) as medals_won -- sum up all medals
FROM Olympics
GROUP BY athlete;                            -- per athlete

-- Query 3
SELECT DISTINCT athlete -- don't repeat an athlete name
FROM Olympics
WHERE total_medals > 0   -- won at least 1 medal
  AND YEAR > 2010
  AND athlete IS NOT NULL; -- filter out NULL athletes

-- Query 4: 26.5ms
SELECT athlete, sum(gold_medals) AS total_gold
FROM Olympics
GROUP BY athlete
ORDER BY total_gold desc;

-- Query 5: 14.7ms
SELECT sport
FROM Olympics
GROUP BY sport
ORDER BY sum(total_medals) desc
LIMIT 1;
```

```sql
-- or (not sure if correct)
SELECT sport FROM SprotsSumView s
WHERE SUM >= ALL(SELECT medals from SportsSumView)
CREATE NEW SportsSumView AS
SELECT sport, sum(total_medals) as medals
FROM O
GROUP BY sport;

-- Query 6: 5.52ms
SELECT country
FROM Olympics
WHERE year >= 2000
GROUP BY country
ORDER BY sum(gold_medals) desc
LIMIT 3;

-- could use RANK : aggregate function that operates on a window
SELECT country, SUM(gold_medals),
RANK() OVER(ORDER BY SUM(gold_medals) DESC) AS rank
FROM Olympics
WHERE year >= 2000
GROUP BY Country
HAVING rank <= 3;

-- Partition Example (Not part of HW)
SELECT country, SUM(gold_medals),
RANK() OVER(ORDER BY SUM(gold_medals) DESC) AS rank
FROM Olympics
WHERE year >= 2000
GROUP BY Country, year
HAVING rank <= 3;
```