CMSC424 - Database Design
SQL Assignment (Parts A and B)

Fall 2007
Prof. Nick Roussopoulos
TA: Sam Huang

Part A Due Date: Tuesday, October 2, 2007 at 12:30 pm
Part B Due Date: Tuesday, October 9, 2007 at 12:30 pm

Last modified October 7, 2007

This is actually two separate assignments, broken into two parts (Part A and Part B). The due dates are different (see above).

These assignments consists of the student (you) constructing SQL queries for a supplied data set. The queries present in Part A are meant to be introductory, to get you familiar with the Oracle SQL interface, while the Part B is more involved.

The database you will be writing queries about contains information concerning the Association of Tennis Professionals. It contains information about players, matches and tournaments concerning the ATP. A copy of the relationships in the database, as well as a few sample entries, is included at the end of this file. Primary keys are underlined. If you are unfamiliar with tennis terminology (especially concerning scoring), a quick description can be found on Wikipedia at http://en.wikipedia.org/wiki/Tennis#Manner_of_play.

Several assumptions about the database are included below:

- The source of the data used for this database only contained information about the ATP (men's tennis) and not the WTA (women's tennis). All players involved are male; we have no information pertaining to women's singles or doubles, or mixed doubles.

- While it should be clear that Player's `PID` and Tournament's `TID` field should be unique, this is also true for Match's `MID` field; no IDs are re-used throughout different tournaments. Thus, a Match's `MID` field uniquely represents it across all Matches known to exist. This avoids having to compare Tournament IDs as well as Match IDs for several of the queries.

- When a player participates in a tournament, he first has to register. For the purposes of our database, this includes receiving a registration number for the tournament. Because the ATP manages all of the tournaments in our database, we assume that a registration number is unique throughout all of the tournaments. This means that given a only a registration number, we can determine the tournament it corresponds to.

- A Player's ID (Player.`PID`) is only used in our database for registration in tournaments. All other locations the Player is referred to by the `RegistrNum` (notably in the `Winner` attribute of MatchResults).

- Two players participating in doubles tennis as a team share the same registration number. Note that several players register for both singles and doubles tennis in the same tournament, players in this category get two separate registration numbers.

- Both a player registered for singles play or two people registered for doubles play under the same number (i.e. are playing with each other) can both be referred to as a team: a team of one or a team of two.

- In most tournaments, the top fraction of the players registering are seeded, to help spread out the players expected to perform well (this makes the later matches more interesting!). Seed values are given on a per-tournament basis. Players who do not receive a seed in a particular tournament have a `NULL` value instead.

- The Tournament relation stores the number of rounds the tournament has as an attribute, which is an integer. The `Round` attribute of the Match relation is similar. A value of `1` corresponds to the first round, `2` to the second, and so on. Match.`Round` does **not** say "Quarterfinals," "Semifinals," or "Finals." You will have to figure out how to compute what rounds correspond to these. Don't try to hard-code values in (i.e. if you look at the database and realize that the US open singles had a total of 7 rounds, so the Semifinals is round 6). Rather, use a nested query to help.

- Each entity in the Set relation contains information about a particular set of some match. The attributes involving the "Winner" refer to the winner of the winner of the match, **not** necessarily the set (similarly for the "Loser" attributes). In cases where the set was determined by a tie-breaker, the TieBreaker relation has `WinnerTB` and `LoserTB`, which contain the points won in the tie-breaker game. Again, the "Winner" column of the TieBreaker refers to the winner of the match, not necessarily the set.

- Sometimes players will need to retire from a match, which counts as a forfeit. If this happens, we keep records for all *completed* sets of the match. For the set in which the player retired, the `WinnerGames` and `LoserGames` attributes hold the number of games won and lost. The RetiredMatch relation holds all Match IDs where the loser retired.

- Many of the queries will require some date manipulation. Specifically, you may have to extract the year a particular tournament was held. Assume that no tournaments wrap around years (i.e. late December-early January), and only concern yourself with the year of the start date.

Directions for using Oracle: Oracle can be accessed through any WAM or GLUE system. At the prompt, type "`tap oraclient`" to set up all the environmental variables, etc needed to use oracle. It will ask for the database SID, for which you should enter "`dbclass1`." To enter the oracle environment, type "`sqlplus`." You should be prompted for a user-name

and password, which you will be supplied with. After entering these, you will see a prompt that looks like `SQL>`. You're now in the Oracle environment.

The official documentation for Oracle can be found at http://www.oracle.com/technology-/documentation/.

Information specific to this project will be given out through email. Expect to be directed to download a compressed file containing the data set and instructions for loading it into Oracle. In the meantime, you might want to try writing out some of the queries by hand to get a feel for how to structure them.

How to hand in your work: You should be making two separate submissions, one for Part A and one for Part B. Please included solutions to all chosen problems (see below) in a file named `<lastname>-A.sql` for Part A, and a file named `<lastname>-B.sql` for Part B. Use SQL-style comments at the start of each query to specify what question the query will solve. IMPORTANT: If you do **not** signify what problem a query addresses, you will receive no points for that query.

Aside from comments made inline with your SQL queries, all other comments or notes to the grader should be made in a file named `README`. **Do not** include a spool of the output of your queries, we will run your queries to generate the output. You should probably only need to submit at most two files (the `README` and the file containing your SQL queries). Compress your files using `tar`, `gzip`, etc. Call your submission `<lastname>-sqlA.tar.gz` (or whatever extension matches the compression method you used) for Part A, and `<lastname>-sqlB.tar.gz` (or whatever) for Part B.

Submit your work on either WAM or GLUE using the `/usr/local/scripts/submit` script. If you need help using this, try "`submit -h`." If you are having trouble with submission at the last minute, you can email your submission to `cmsc424-staff@cs.umd.edu`.

Part A

Choose 10 of the following 12 questions. Write SQL queries for them (remember to clearly denote the number of each query!). It may help to write out the equivalent relation algebra expressions before you begin.

1. List the names of all the players in the database.

2. List the names of all players who have ever been assigned a seed for any tournament (doubles or singles).

3. List the pairs of players who played doubles tennis at Wimbledon in 2007. Do not include duplicate entries (ex: $\langle personA, personB \rangle$ as well as $\langle personB, personA \rangle$).

4. List the names of all players who have lost to Roger Federer in the finals of any tournament, as well as the name of the tournament they lost in. Include results only for singles tennis.

5. For all final round single matches, list the winner and loser of matches that were between two seeded players, as well as their seeds. Modify the titles of the columns to be something useful, like `WinnerName`, `WinnerSeed`, `LoserName`, `LoserSeed`.

6. List the names of all US players who have participated in at least two tournaments in 2007. Do **not** use any aggregating functions for this problem.

7. List all tournaments having more than 5 rounds. Print the name of the tournament, the tournament type, the start and end dates, and the number of rounds.

8. List all doubles matches that were won because one of the teams retired. Include the winner's names, the loser's names, the tournament name, the year of the tournament, and the round number of the match.

9. Find all singles matches where the loser retired after playing at least one complete set. Include the winner's name, the loser's name, the tournament name, the year of the tournament, and the round number of the match.

10. For all tournaments in the database, list the name, tournament type, surface type, and the number of rounds it has. Sort the results in descending order by the number of rounds.

11. List the names, tournament types, and lengths (in days) of all tournaments that were longer than one week.

12. List the names of all male German players who registered for the 2007 Australian Open singles.

Part B

Choose 6 of the following 8 questions. Write SQL queries for them (remember to clearly denote the number of each query!). It may help to write out the equivalent relation algebra expressions before you begin.

1. For all singles quarterfinal, semifinal, and final round matches that only took 3 sets, list the Tournament Name, Year, Winner of the Match, and the Match score (For the match score, print two columns per set, displaying the number of games each player won. Call these columns something meaningful, and ignore any tiebreaker results.).

2. List the names of all triples of players that have played doubles with each other. Only list each triple once.

3. List the names of all pairs of players that have played against each other in both singles and doubles.

4. Find all singles matches won by James Blake containing a love set (i.e the score of a set was either `6-0` or `0-6`). Print the name of the tournament, the year of the tournament, the round of the tournament, and the name of the player James Blake defeated.

5. List the names of all partners of a player in doubles tennis who also was seeded in a singles tournament at some point. This means that we want all players x who have partners y where partner y was seeded in men's singles.

6. List the name of the player who has registered for the most tournaments, as well as the number of tournaments he has registered for. NOTE: If a player has registered for both singles and doubles play in a particular tournament, this counts as two tournaments.

7. List the names of all Russian, Chilean, and US players Tommy Haas has played against. Include results from both doubles and singles matches.

8. List the names of all players who have made it to at least the quarter finals of the four Grand Slam tournaments (Wimbledon, US Open, French Open, and Australian Open) in 2007 (they could have made it to the finals in either doubles or singles for each tournament considered).

*** Note that the entries displayed here may *not* be the exact values in the database, rather they are meant to give a flavor of the types of values.

Player	PID	Name	Gender	CCode
	920	Roger Federer	M	SUI
	729	James Blake	M	USA

Tournament	TID	Name	Location	StartDate	EndDate	NumRounds	TType	Surface
	5	Wimbledon	London, Great Britain	25-Jun-07	8-Jul-07	6	Doubles	Grass

PlayedIn	TID	RegistrNum	Seed
	5	537	12

Match	MID	TID	Round	RegistrNum1	RegistrNum2
	1001	4	6	456	654

Registration	RegistrNum	PID
	537	777
	537	779

MatchResults	MID	Winner	NumSets
	1001	456	4

CountryCodes	Code	Country
	USA	United States of America
	SUI	Switzerland

SetScore	MID	SetNum	WinnerGames	LoserGames
	1001	1	6	1
	1001	2	2	6
	1001	3	7	6
	1001	4	6	4

RetiredMatch	MID
	324

Tiebreaker	MID	SetNum	WinnerTB	LoserTB
	1001	3	7	2