Gebze Technical University
Computer Engineering Department
Introduction to Programming
CSE102 Term Project
Fall 2014

Due Date
02.01.2015, 09:00

SüperLig Database

In this project, you will implement a database system which allow users to manage team, player and statistical records in the SüperLig. Users can run queries to extract high level information from the database. Your database system should contain following information:

Table 1 Required tables

Table Name	Fields (at least)
teams	team_name(str), city(str), stadium(str), founding_date, colors(str)
players	player_name(str), team_name(str), birth_date, origin(str), position(str), salary
team_statistics	year, team_name, points, wins, loses, draws, goals_for, goals_against, goals_difference
player_statistics	year, player_name, goals, assists, red_cards, yellow_cards

The proposed system have to fulfill following commands:

Table 2 Query commands

Available Commands	Variants	Description	Sample Query
select	select *field1*, *field2*,.. from *table_name*	Returns *field1*, field2,… of all records from *table_name*	`select team_name from teams`
where	**SELECT_STATEMENT** where *field_a*=y BONUS: **SELECT_STATEMENT** where *field_a*=y & *field_b*>z BONUS OPERATORS: Comparison operators (<,>) and conditional operators (&,|)	Returns *field1*, field2,… of records whose field is y from *table_name*.	`select colors, stadium from teams where team_name=fenerbahce`
sort	**SELECT_STATEMENT** sort by *field1* asc	Sorts the returned data from select statement by specified field. Sort may be in ascending (asc) or descending (desc) order.	`select team_name, founding_date from teams sort by founding_date asc`

smart select	BONUS(Generic smart select):
	
	 smart select *field1*, *field2*,… ***WHERE*_STATEMENT(optional)** ***SORT*_STATEMENT(optional)**
		Returns *field1*, field2,… of all records from proper tables	If you didn't implement generic `smart select` utility, you should only handle following hard coded statements.
			
			 `smart select team_name, stadium,points where points>20&year=2010  sort by points desc`
			
			 `smart select player_name, position, red_cards where position=GK& red_cards>0 sort by red_cards asc`

			smart select player_name,city,goals where city=istanbul& goals>10 sort by goals desc
insert	insert *field1*, *field2*,… to *table_name*	Inserts given element to the database	insert ankaraspor,ankara,19_mayis,2005,mor-sari to teams
update	update *field1*, *field2*,… in *table_name* where *field*=y	Update a record whose field is y in *table_name*	update team_name=osmanlispor, founding_date=2014 in teams where team_name=ankaraspor
delete	delete * from *table_name* where *field*=y	Deletes a record from *table_name* whose field is y	delete * from teams where team_name=ankaraspor
q		Quits.	

Your main function should accept commands in a loop until user hits 'q' key. Keep all data in binary files. Your program should read all the recorded data beginning of the program execution (When user press q and re-execute the system, all recorded data should be available in the database).

Your main function should accept command line arguments (if given).

Table 3 Command line arguments

Argument	Description	Sample Usage
fn	Input query file name.	./exe fn query.txt
r	Deletes all previous saved data in the system.	./exe r
	(All arguments can be used separately or in combined form)	./exe fn query.txt r

Notes:

- Demonstrations will be held on Friday January 2, from 9am - 6pm, in the project laboratory.
- Provide conditional compilation for test and debug modes.
- Use dynamic memory allocation wherever necessary.
- Use at least 2 macro functions.
- Separate headers and implementations parts.
- Write a report in the format shown in your text book.
- Provide a proper makefile.
- You should submit library headers, implementations and main.c files;

- Add all **files into a folder and compress it** for submission. The folder names will be restricted to the following format:
  HW#_studentid_studentname.
    - Example:
      HWTP_121044001_Abdullah_Akay
- Upload soft copy of your homework to Moodle course web page
- **DON'T submit hard copy of your assignment**.
- Don't forget to test your code on linux machines in the project Lab (Room 101).
- Obey good programming rules (Indentation, Documenting, Well Commenting, Avoidıng magic numbers, Non-ascii characters etc.)
- **Strictly follow submission and file, folder naming rules.**