**Events** (ID, event\_date, start\_time, end\_time, team\_one, team\_two, created\_at)

PK: ID

FK: team\_one ◊ Teams.ID, team\_two ◊ Teams.ID

**Teams** (ID, team\_name, team\_desc, created\_at, country, *sport\_id*)

PK: ID

FK: sport\_id ◊ Sports.ID

**Sports** (ID, sport\_name, sport\_desc, created\_at)

PK: ID

The logical schema above, unlike the ER diagram, includes referential integrity constraints as well as timestamps and primary key constraints (ID) but does not include relationships mappings.

The tables could be further decomposed; first normal form was violated in the ‘teams’ relation whenever a team was from the US – the country attribute was non-atomic, but due to the scale of the project this is not relevant.

The event calendar, although covering all the required functionalities, has no visual interface apart from input forms and hyperlinks. Multiple events can be added to a single day, they can’t incorporate one or both of the teams that had already played on that day, however.

As the calendar supports a multitude of different sports, potential sport mismatches are accounted for, in order to prevent the user from pitting a hockey club against a football club for example – additionally, another server-side check is added to prevent users from adding events where both teams are the same. To maintain simplicity, the calendar does not include any form of user identification and the added events are therefore not stamped with user details. Finally, update timestamps are also omitted, for the sake of simplicity.