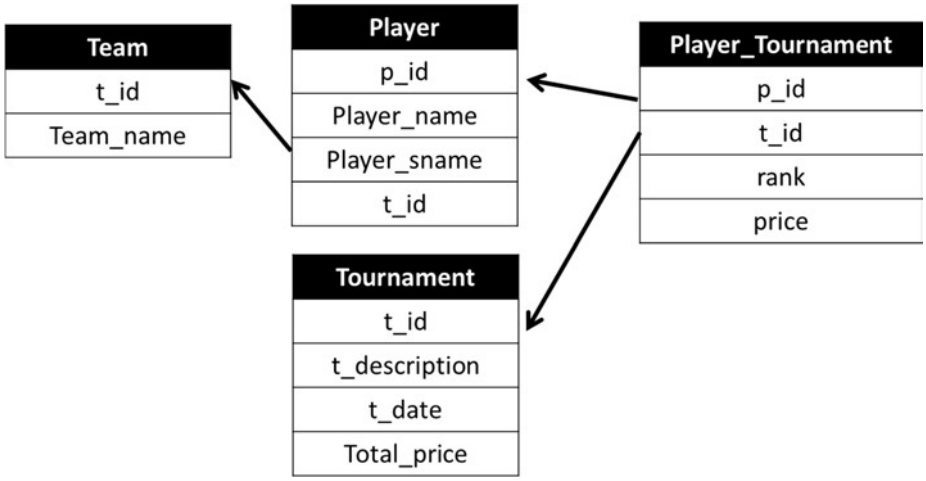
**Advanced Databases, Dr. Pierpaolo Dondio**
**Lab – week 8**
**Dimensional Model and ETL**

---

Execute the script *golf.sql*

The script generates two relational models for two *golf results* databases. Each golf player belongs to a team. Each player plays one or more tournament, and the DB stores player's ranking in the tournament and the price won for each tournament. The DB has a tournaments list as well.

**ER DIAGRAM:**




**An relational schema of the two starting databases**

Each DB registers results from distinct tournaments, but the same player can be in both DBs. Assume that a player name and surname identifies always the same person. The same is for Team names, i.e. same team name across the two DBs is the same team.
The data format is the same with only two exceptions:

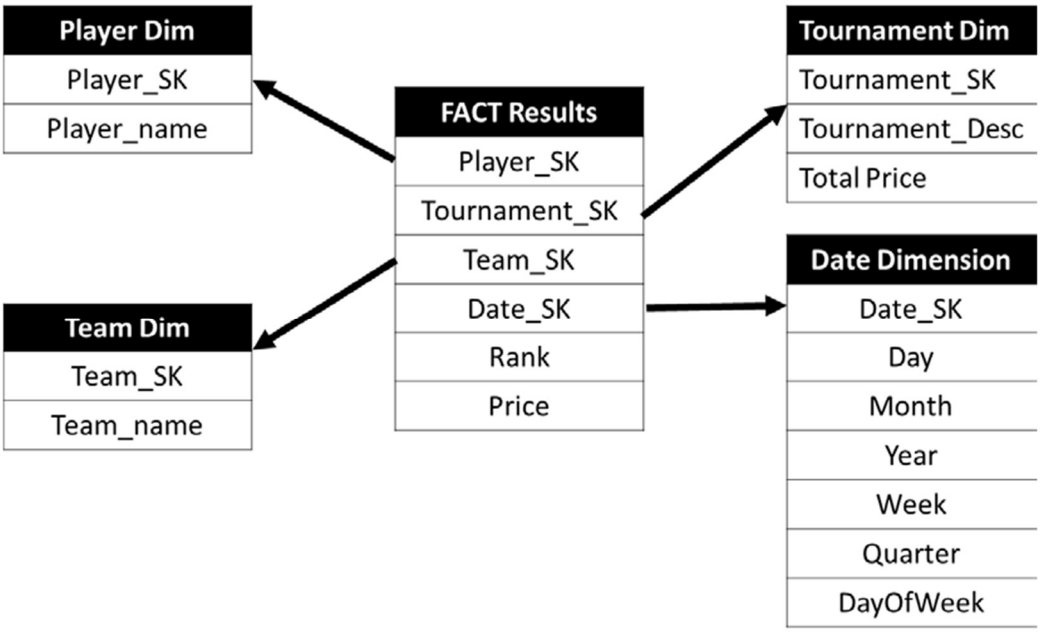
1. DB1 stores prices in EURO while DB stores prices in $. Assume a fix conversion rate 1 EURO = 1.3 DOLLAR and set all the values to EURO.

   You are required to:

1. Create an SQL script to create the following dimensional model (next page)
2. Load data from the two DBs into the Dimensional model, using surrogate keys and applying proper data transformations to address the two above differences in data representation. Note that the date dimension is storing, as integer fields, the surrogate key and the day, month, year, week number (1-52) , quarter and day of the week (from 1 to 7, 1=Monday, 2=Tuesday,..,7=Sunday). You need to write a PL/SQL script to convert a date into the required fields.

General Instructions

- The code can include triggers, if you think they could be useful.
- Refer to the presentation "ETL example" (Brightspace, folder "lecture notes") to understand the steps you need to implement
- ASSUME that in each database the name of the player + the surname of the player is unique, it identifies the same person. Therefore, if there is a player with the same name + surname in the other database then it is the same player (that means the two players are the same entity and they need to have the same surrogate keys).




**Dimensional model**

Useful command:

Extract part of a date from a date field ( the year from the *t_date* field in the example) and convert into an integer:

```
select cast(to_char(t_date,'YYYY') as integer) from TOURNAMENT1;
```

Use those kinds of instructions to create a script to populate the date dimensions. You can refer also to the script on the presentation "ETL EXAMPLE" that you can find on Brightspace in the folder Lecture Notes.

**SECOND LOAD**

After a while, the following new data are available in DB1. Execute the following statements to enter new data.

```
INSERT INTO PLAYERS1 (P_ID, P_NAME, P_SNAME, TEAM_ID) VALUES (7, 'Alan', 'Parker', 1);

INSERT INTO PLAYERS1 (P_ID, P_NAME, P_SNAME, TEAM_ID) VALUES (8, 'Martha', 'Bag',2);

INSERT INTO TOURNAMENT1 (T_ID, T_DESCRIPRION, TOTAL_PRICE) VALUES (5, 'SaudiOpen', 500000);



INSERT INTO RESULTS1 (T_ID, P_ID, RANK, PRICE) VALUES (5, 1, 1,60000);
INSERT INTO RESULTS1 (T_ID, P_ID, RANK, PRICE) VALUES (5, 7, 5,20000);
INSERT INTO RESULTS1 (T_ID, P_ID, RANK, PRICE) VALUES (2, 8, 3,1000);
```

3. Produce SQL code to load the above data into the datewarehouse. You should be able to add those new data using the same script defined at point 3