# 1. Red Bull Air Race

Problem description

You want to store information about one season of the Red Bull Air Race competition. In this competition, the participants are pilots that are organized in teams. Each pilot competes throughout the season with a single model airplane .

Each driver is identified by a number and you should also store his first and last name , his nationality and birth date. Regarding the team, it is only necessary to know its name and country of origin. However, each airplane model is characterized by its power, maximum speed, length, width and weight.

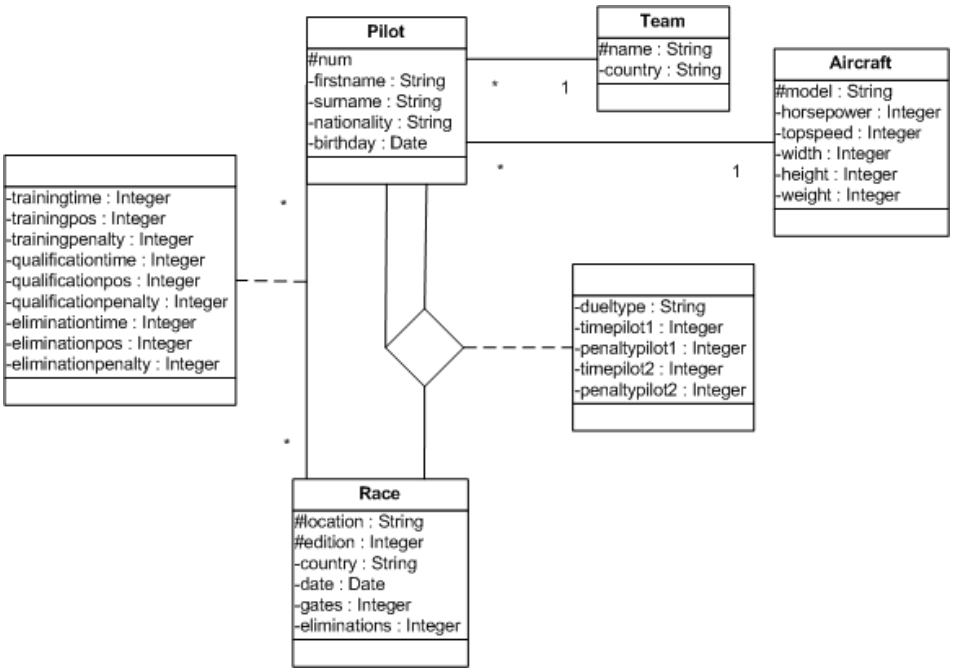
Throughout the season, riders compete in races that take place in a city, in a given year. It is interesting to know in which country and date each race was held, its number of gateways and how many riders are eliminated in the first round.

The races follow a 'two-stroke mechanical'. The first phase consists of running the route in the shortest time possible, fighting only against the clock . This phase is divided into stages: Free Practice, Qualifying and Elimination. For each stage it is interesting to save the time, the relative position of each pilot and the penalty seconds. The times obtained during Free Practice dictate the exit order in Qualification and its order for Elimination. The rider with the worst time of Qualification does not go on the Elimination and the four worst in the Elimination stage do not to follow to the Second Phase.

The Second Stage is a duel stage in which drivers compete directly 2 to 2 since the Quarters of Final until the Final. It is important to keep the information on who participated in each duel, in which stage, with what time and penalty.

The scoring system is linear, giving 6 points to the winner and 1 point to the pilot in the 6th place.

UML Class Diagram



Relational Model

Team (name, country)
Aircraft (model, horsepower, topspeed, width, height, weight)
Pilot (num, firstname, surname, nationality, birthday, name ➔ Team, model ➔ Aircraft)
Race (location, edition, country, date, gates, eliminations)
Participation (num ➔ Pilot, [location, edition] ➔ Race, trainingtime, trainingpos, trainingpenalty, qualificationtime, qualificationpos, qualificationpenalty)
Duel (numpilot1 ➔ Pilot, numpilot2 ➔ Pilot, [location, edition] ➔ Race, dueltype, timepilot1, timepilot2, penaltypilot1, penaltypilot2)

Database construction

Using SQL, build the necessary relations and interity restrictions to the given relational model.

[Exercise by Gabriel David, Vasco Vunhas, André Restivo]

## 2. Auto Center

Using SQL, create the necessary tables and the necessary integrity rules for the UML class diagram built to the car dealer exercise described in BD-P-MC-Res.pdf. Before creating the tables, check if the obtained relations are in the Boyce-Codd Normal Form.

[Exercise by João Mendes Moreira]

## 3. University

Consider a database that stores students grades in several examinations, pertaining several courses of one or more programes, with the relations and data presented next (Aluno=Student; Cadeira=Course; Prof=Professor; Prova=Examination; Nome=Name; Sigla=Acronym; Design=Nome, Curso=Programme, Regente=Responsible Professor; Data=Date; Nota=Grade):

ALUNO

| **nr** | **Nome** |
|---|---|
| 100 | João |
| 110 | Manuel |
| 120 | Rui |
| 130 | Abel |
| 140 | Fernando |
| 150 | Ismael |

PROF

| **sigla** | **Nome** |
|---|---|
| ECO | Eugénio |
| FNF | Fernando |
| JLS | João |

CADEIRA

| **cod** | **Design** | **curso** | **regente** |
|---|---|---|---|
| TS1 | Teoria dos Sistemas 1 | IS | FNF |
| BD | Bases de Dados | IS | ECO |
| EIA | Estruturas de Informação e Algoritmos | IS | ECO |
| EP | Electrónica de Potência | AC | JLS |
| IE | Instalações Eléctricas | AC | JLS |

PROVA

| **nr** | **cod** | **data** | **nota** |
|---|---|---|---|
| 100 | TS1 | 92-02-11 | 8 |
| 100 | TS1 | 93-02-02 | 11 |
| 100 | BD | 93-02-04 | 17 |
| 100 | EIA | 92-01-29 | 16 |
| 100 | EIA | 93-02-02 | 13 |
| 110 | EP | 92-01-30 | 12 |
| 110 | IE | 92-02-05 | 10 |
| 110 | IE | 93-02-01 | 14 |
| 120 | TS1 | 93-01-31 | 15 |
| 120 | EP | 93-02-04 | 13 |
| 130 | BD | 93-02-04 | 12 |
| 130 | EIA | 93-02-02 | 7 |
| 130 | TS1 | 92-02-11 | 8 |
| 140 | TS1 | 93-01-31 | 10 |
| 140 | TS1 | 92-02-11 | 13 |
| 140 | EIA | 93-02-02 | 11 |
| 150 | TS1 | 92-02-11 | 10 |
| 150 | EP | 93-02-02 | 11 |
| 150 | BD | 93-02-04 | 17 |
| 150 | EIA | 92-01-29 | 16 |
| 150 | IE | 93-02-02 | 13 |

Note that the key of the table PROVA has the **nr, cod** e **data** attributes, what allows to store grades of more than one examination per course.

Database construction

Using SQL, create the necessary relations and the necessary integrity rules to the relational model presented above. Fill the relations with the date included in the example.

[Based on an exercise by Gabriel David]