Conceptual Modeling

1. Formula 1

It is intended to store information about a season of Formula 1 Championship.

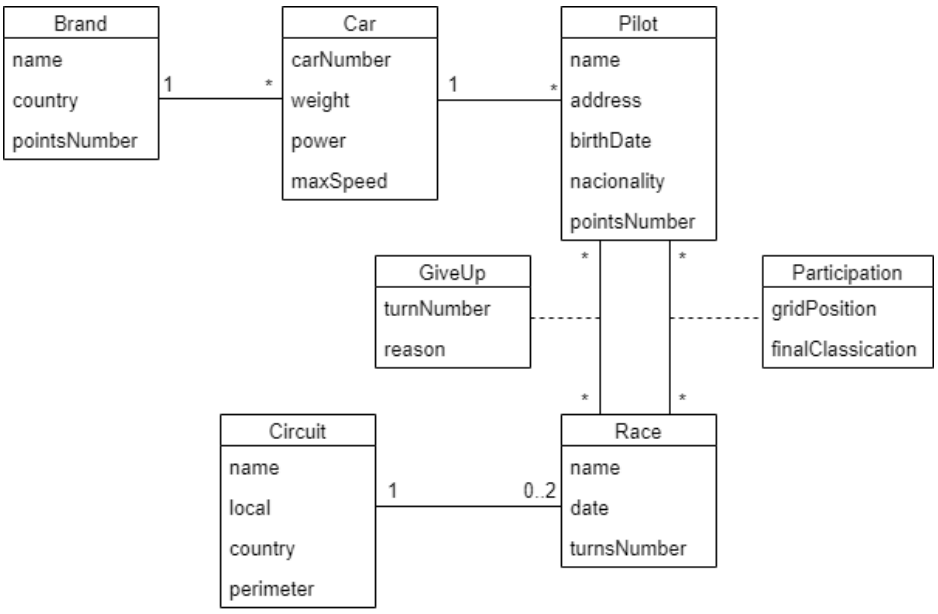
For each tag participant in the season it is intended to store its name, country of origin, current number of championship points and which cars entered. For each car, the weight, the power and the maximum speed are stored.

For participating pilots we need to know his name, address, age, nationality and current number of points in the drivers ' Championship. A pilot can only drive a car throughout the season, although a given car can be driven by more than a pilot. This situation, although not very frequent, can arise, for example, due to the absence of a pilot injured in an accident.

A number of races are held on circuits and dates set at the beginning of the season. For a given race it is possible, throughout the season and for varying reasons, to change the circuit where this takes place. In exceptional situations this can also happen by conducting two races on the same circuit. For each circuit, the name, location, country, number of laps and perimeter are stored.

With regard to the realization of a race matter to know for each pilot that has participated; (1) the position on the starting grid; and (2) the position in the final ranking. Regarding race matters also know what are the riders who dropped out, in which lap has occurred and what the reason for the abandon. [Based in an exercise by António Brito]

I. Conceptual Model



II. Relational Schema

Marca (idMarca, nome, pais, numPontos)

Carro (idCarro, numCarro, peso, potencia, velocidadeMax, idMarca -> Marca)

Piloto (idPiloto, nome, morada, idade, nacionalidade, numPontos, idCarro -> Carro)

Circuito (idCircuito, nome, local, pais, perimetro)

Corrida (idCorrida, nome, data, numVoltas, idCircuito->Circuito)

Participacao (idPiloto -> Piloto, idCorrida -> Corrida, posicaoGrelha, classFinal)

Desistencia (idPiloto -> Piloto, idCorrida -> Corrida, numVolta, motivo)

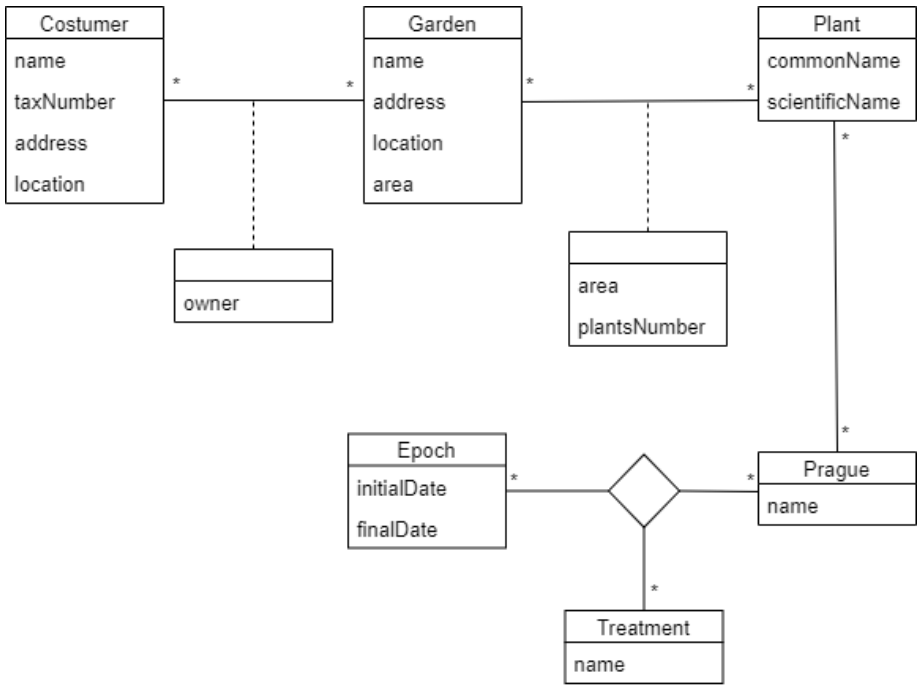
2. Gardener

A gardener is in charge of the maintenance of the gardens of their clients. Thus, he wants to know about each client his name, NIF, address and location. These data are needed for billing purposes.

About each garden it is necessary to know the name, address, location, area (in hectares), the customer responsible for this garden and whether he owns the garden or just explore it. Still about each garden is necessary to know the types of plants that are found there, area occupied by each of these plant types and number of plants of each type. About the types of plant, the name by which is known (for example: orange blossom) and its scientific name (e.g. citrus sinensis) is stored.

For each type of plant it still matters to know the pests that affect it (e.g. the orange trees can have cochineal). For each pest, its name and seasons (one or more) in which treatments must be performed are stored. Seasons are characterised by a start date and an end date. [Based in an exercise by by João Mendes Moreira]

I. Conceptual Model



II. Relational Schema

Cliente (idCliente, nome, nif, morada, localidade)

Jardim (idJardim, nome, morada, localidade, area)

Proprietario (idCliente -> Cliente, idJardim -> Jardim, proprietario?)

TipoPlanta (idTipoPlanta, nomeComum, nomeCientifico)

JardimPlantas (idJardim -> Jardim, idTipoPlanta -> TipoPlanta, area, numPlantas)

Praga (idPraga, nome)

PragaTipoPlanta (idPraga -> Praga, idTipoPlanta ->TipoPlanta)

Tratamento (idTratamento, nome)

Epoca (idEpoca, dataInicio, dataFim)

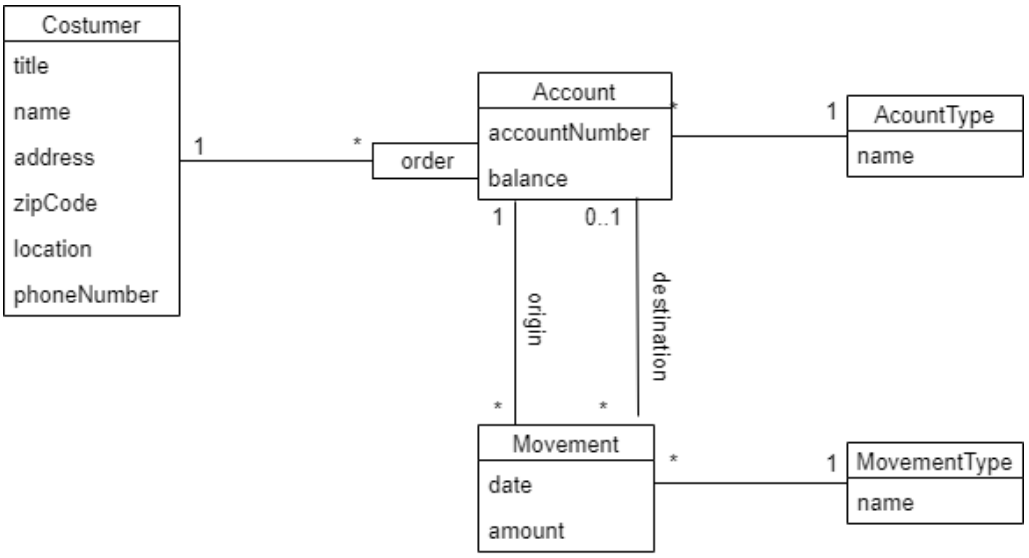
PragaTratamentoEpoca (idPraga -> Praga, idTratamento -> Tratamento, idEpoca -> Epoca)

3. MuitosJuros Bank

Build a conceptual model to represent the reality of the MuitosJuros Bank, taking into account that:

i. customers have a title, name, address, zip code, location, and phone;
ii. the accounts are linked to an account number, an account type and the current balance;
iii. can be carried out movements of various kinds and amounts;
iv. each movement must be associated with a date;
v. each account can have one or more holders with different order (e.g. 1st and 2nd holder);
vi. interested in what are the first holders of a given account, the dates on which an account had movements and who are customers who have an account of a given type. [Exercise by Carla Teixeira Lopes]

I. Conceptual Model



II. Relational Schema

Cliente (idCliente, titulo, nome, morada, telefone)

TipoConta (idTipoConta, nome)

Conta (idConta, numConta, saldoActual, idTipoConta -> TipoConta)

Titular (idConta -> Conta, ordem, idCliente -> Cliente)

TipoMovimento (idTipoMovimento, nome)

Movimento (idMovimento, montante, data, idTipoMovimento -> TipoMovimento, origem -> Conta, destino -> Conta)

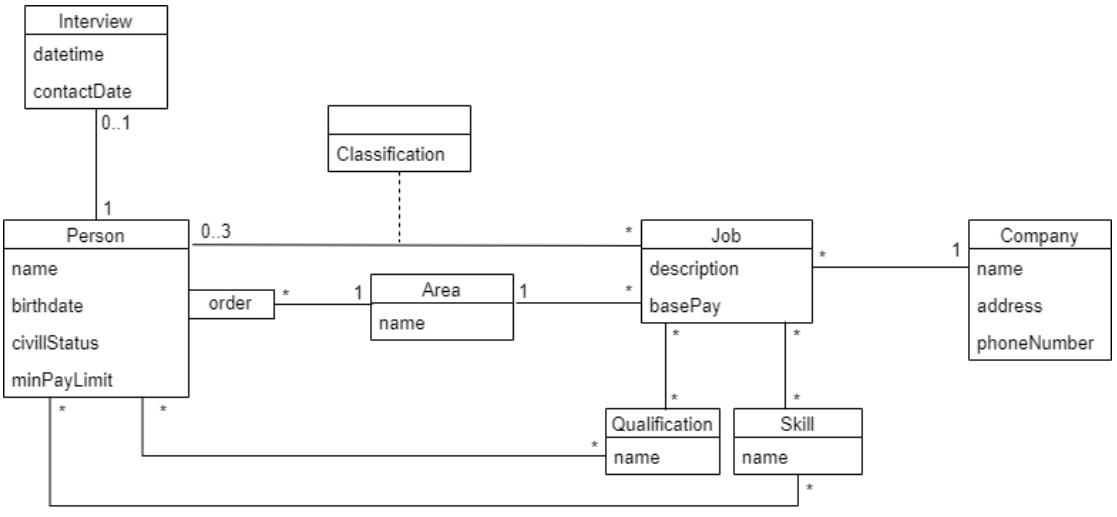
4. Employment agency

An employment agency aims to find suitable personnel to fill vacancies provided by companies. Thus, when a company wants to recruit people, it must provide the Agency with a work area, a description of that work, qualifications and skills required, base compensation and an indication of the applicant company. About the company it is necessary to know the name, address and phone.

When a person uses the agency to seek employment he/she has to provide the name, date of birth, marital status, qualifications, skills, minimum remuneration and a list of the work areas ordered according to the preference of the person. Qualifications, skills and work areas set both by companies and by people are selected from a list defined by the Agency.

The system will automatically assign jobseekers to offers made by companies. Thus, for each offer, three people are assigned, who are contacted to attend individual interviews. The system must record the date of contact, as well as the date and time of the interview. In the case of the person who has been selected for more than one job offer, he/she only goes to one interview. In the interviews a rating is assigned for each of the jobs to which that person has been assigned. [Based in an exercise by João Mendes Moreira]

I. Conceptual Model



II. Relational Schema

Pessoa (idPessoa, nome, dataNascimento, estadoCivil, limMinRemuneracao)

Entrevista (idEntrevista, dataHora, dataContacto, idPessoa -> Pessoa)

Empresa (idEmpresa, nome, morada, telefone)

Area (idArea, nome)

PessoaArea (idPessoa -> Pessoa, ordem, idArea -> Area)

Emprego (idEmprego, descricao, remuneracaoBase, idEmpresa -> Empresa, idArea -> Area)

EmpregoPessoa (idEmprego -> Emprego, idPessoa -> Pessoa, classificacao)

Habilitacao (idHabilitacao, nome)

Competencia (idCompetencia, nome)

PessoaHabilitacao (idPessoa -> Pessoa, idHabilitacao -> Habilitacao)

PessoaCompetencia (idPessoa -> Pessoa, idCompetencia -> Competencia)

EmpregoHabilitacao (idEmprego -> Emprego, idHabilitacao -> Habilitacao)

EmpregoCompetencia (idEmprego -> Emprego, idCompetencia -> Competencia)

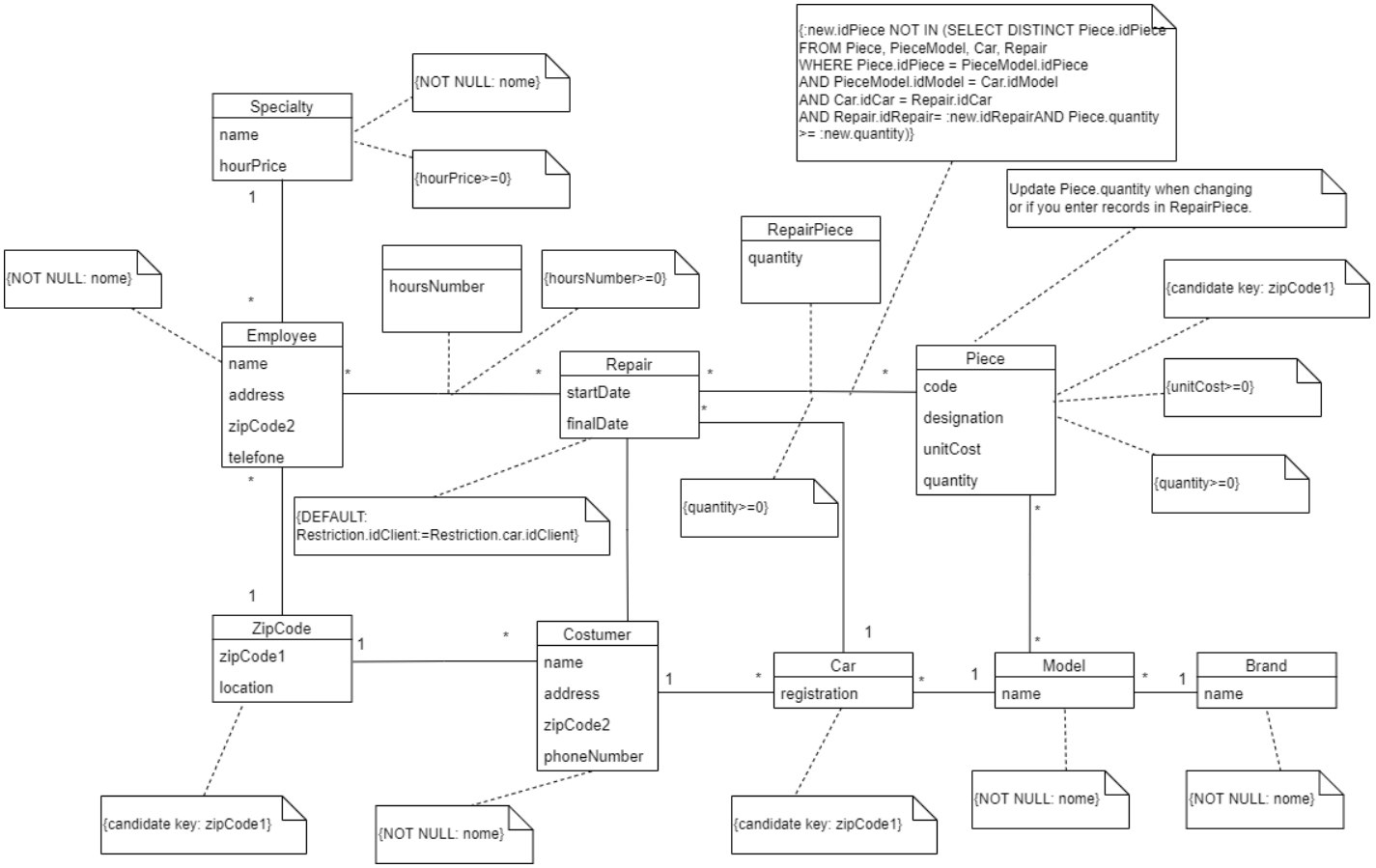
5. Repair Shop

A car dealer plans to computerize its repairs service. Of each client, the name, address and phone number are saved as well as the license number, the brand and model of each car, as well as who is the client owner of this car. On employees of the workshop, the name, the address, the telephone, the specialty of this employee and the cost per hour are stored. It is known that the time cost of the workforce depends on the specialty (for example: the electricians have a price, the plate-worker another price, etc.).

The shop keeps the register of all pieces in stock. For each piece is known: the code, the description, the unit cost and the amount available. It is still necessary to know which car models are compatible with each piece.

For each repair performed is stored the vehicle information, the client (default is the owner of the vehicle, but may be someone else), the start and end dates of the repair, and still: (1) all the pieces used in the repair (obviously, only compatible pieces with the model of the automobile in repair can be used) and its quantity; and (2) the number of hours worked by each employee in the repair. [Based in an exercise by João Mendes Moreira]

I. Conceptual Model



II. Relational schema

Marca (idMarca, nome)

Modelo (idModelo, nome, idMarca -> Marca)

CodPostal (codPostal1, localidade)

Cliente (idCliente, nome, morada, codPostal1 -> CodPostal, codPostal2, telefone)

Carro (idCarro, matricula, idModelo -> Modelo, idCliente -> Cliente)

Reparacao (idReparacao, dataInicio, dataFim, idCliente -> Cliente, idCarro -> Carro)

Peca (idPeca, codigo, designacao, custoUnitario, quantidade)

ReparacaoPeca (idReparacao -> Reparacao, idPeca -> Peca, quantidade)

PecaModelo (idPeca -> Peca, idModelo -> Modelo)

Especialidade (idEspecialidade, nome, custoHorario)

Funcionario (idFuncionario, nome, morada, codPostal1 -> CodPostal, codPostal2, telefone, idEspecialidade -> Especialidade);

FuncionarioReparacao (idFuncionario -> Funcionario, idReparacao -> Reparacao, numHoras)

6. Soccer League

The entity responsible for major football championships of a seaside country aims to computerize their services. To this end, it asked for help from students of FEUP-MIEIC.

In this entity, designated by the League, clubs are inscribed. Each club has several teams and each team participates in a tournament in each season. A Club may have a team for each distinct age class (juveniles, juniors, etc.) and by gender (male or female). Likewise there is, per season, a Championship by age group and by gender. Teams can only participate in the League of the same age group and gender.

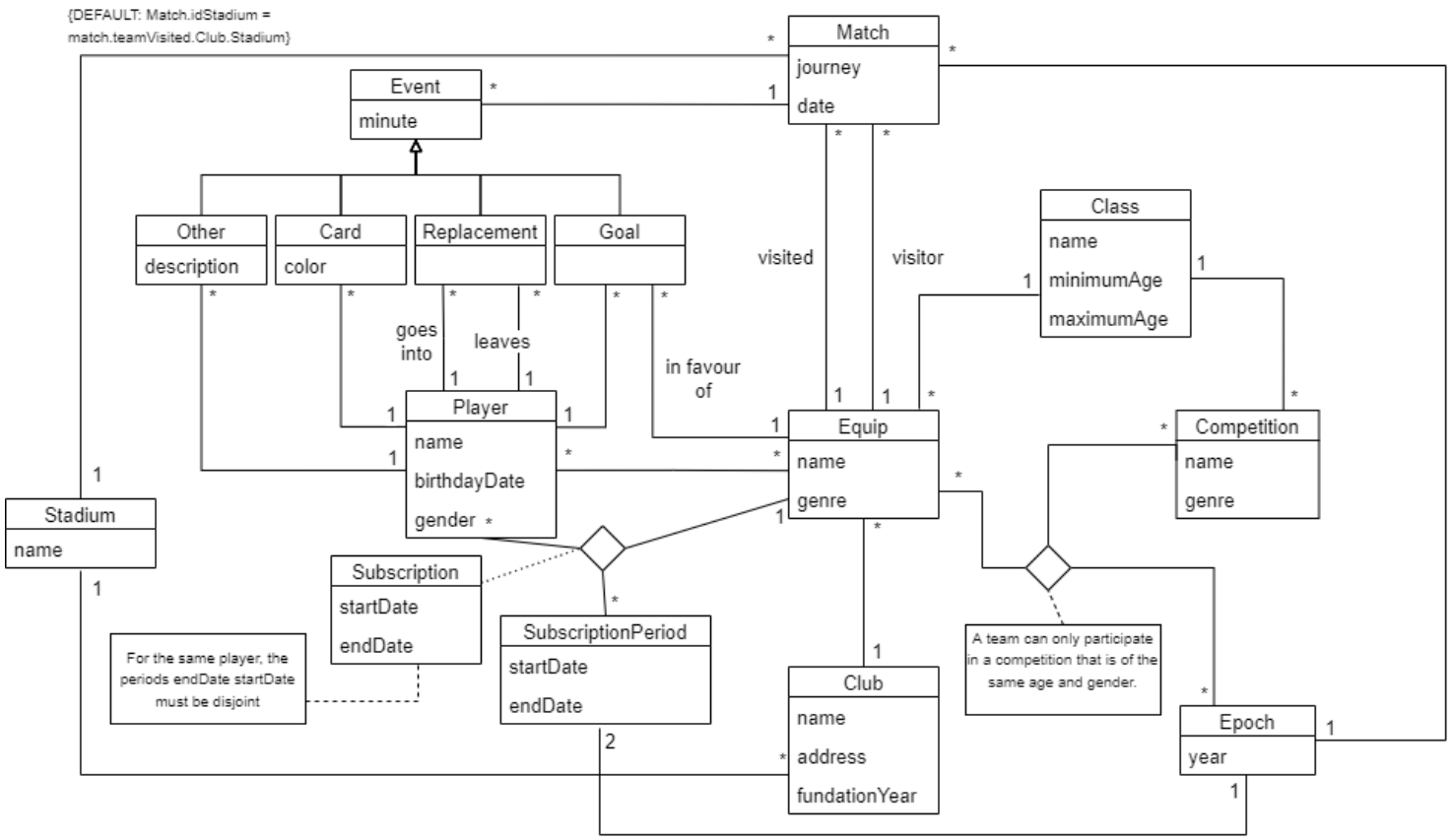
On each team it is necessary to know the players. Each season has two periods for the registration of players. In every moment it must be possible to know which players legally registered in the League and what the Club to which each is bound. At one point a player can only be bound to a club. The team to which the player belongs is determined at the beginning of the season for their age group and by its gender.

At the beginning of the season is set a calendar of games for each Championship. Each game shall include: the journey to which it relates, expected date of completion, the teams in confrontation (visited and visitor) and the stadium where the game takes place and which is, by default, the stadium of the Club visited, but may, for some reason, be another. The date of the game can be modified by request of the League clubs.

About the games it is necessary to keep information about the different events that occur throughout the game. These events are established (e.g., yellow/red card, goal, substitution, etc.). Obviously about: disciplinary cards, it is necessary to know who was shown; about the goals, who scored and in favour of the team (are there own goals!); and about the replacements, who went in and who's out. There may be other types of events for which it is only necessary to know the player involved. Is relevant to all events, the minute of the game it happened.

For each class set the appropriate attributes. For example: the clubs have name, address, year of Foundation, etc.[Based in an exercise by João Mendes Moreira]

I. Conceptual model



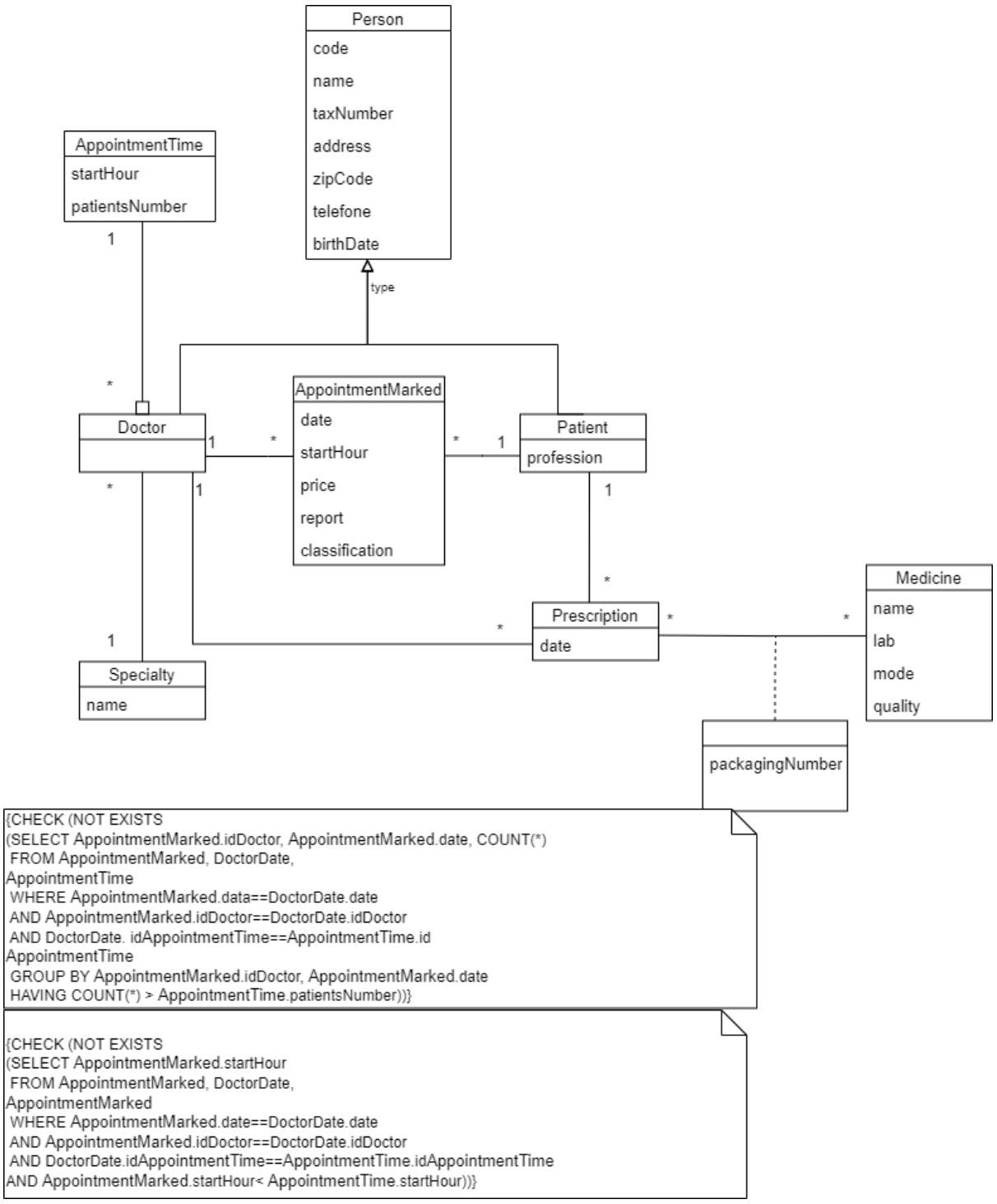
7. Medical Clinic

A medical clinic has at its service a set of doctors and serves a set of patients. Both are people, described by name, TAX ID, address, postcode, telephone number and birthdate. In the context of the clinic, they are identified by a code. Doctors have a specialty and patients a profession.

The work of the clinic is organised according to a schedule that notes what days each doctor gives consultations (e.g., Mondays, Tuesdays, …), the start time of the job and how many patients he/she will see. Patients mark consultancy to a day in which the intended doctor is available giving also the consultancy start time. About each consultancy it is also stored the price paid, the patient status (rating of 1 to 5) and the report query.

It is intended to also keep the prescriptions of medicines made by doctors to patients. For medicines, the name, the laboratory, the method of administration, the amount of drugs on the packaging and the amount of packaging prescribed are stored. [Based in an exercise by Gabriel David]

I. Conceptual Model



II. Relational schema

Especialidade (idEspecialidade, nome)

Medico (idMedico, codigo, nome, nif, morada, codigoPostal, telefone, dataNascimento, idEspecialidade -> Especialidade)

Doente (*idDoente*, codigo, nome, nif, morada, codigoPostal, telefone, dataNascimento)

HorarioConsulta (*idHorarioConsulta*, horaInicio, numDoentes)

MedicoConsulta (*idMedico* -> Medico, data, idHorarioConsulta -> HorarioConsulta)

ConsultasMarcadas (*idConsultaMarcada*, data, horaInicio, preco, relatorio, classificacao, idMedico -> Medico, idDoente -> Doente)

Medicamento (*idMedicamento*, nome, laboratorio, modo, quantidade)

Prescricao (*idPrescricao*, data, idMedico -> Medico, idDoente -> Doente)

PrescricaoMedicamento (*idPrescricao* -> Prescricao, *idMedicamento* -> Medicamento, numEmbalagens)

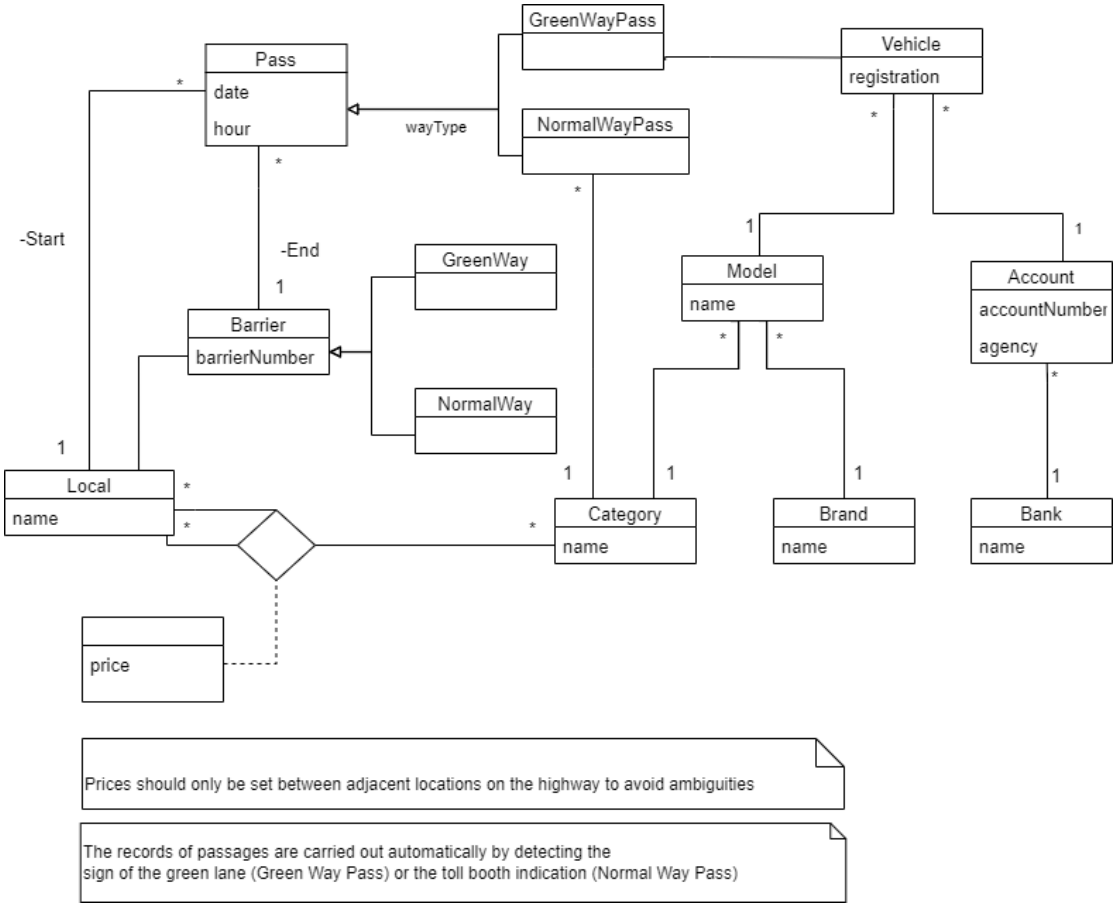
8. Brisa

The company BRISA wants to keep the information on the use of its motorways according to:

i. the existing highways from the A1 (North) to the A8 (Loures);
ii. tolls are charged on the barriers, which have a distinguishing number and are deployed in a site;
iii. the value of the toll depends on the class of the vehicle (from 1 to 4) and is addictive (Porto->Coimbra = Porto->Aveiro + Aveiro -> Coimbra) and symmetrical;
iv. we want to know what day and time the toll was paid, the route to which it relates and the registration and the mark of the respective vehicle.
v. some vehicles are green vehicles, i.e., have a device of automatic identification and it is necessary to know the account number, Bank and agency responsible for the respective payments.

NOTE: it must be possible to represent the costs of the routes independently that someone has already paid that toll. [Based in an exercise by Gabriel David and Ana Paiva]

I. Conceptual model



II. Relational Schema

Barreiras (idBarreiras, numBarreira, tipoVia, idLocal -> Local)

Categoria (idCategoria, nome)

Preco (idCategoria -> Categoria, origem Local, destino -> Local, preco)

Marca (idMarca, nome)

Modelo (idModelo, nome, idMarca -> Marca, idCategoria -> Categoria)

Banco (idBanco, nome)

Conta (idConta, numConta, agencia, idBanco -> Banco)

Veiculo (idVeiculo, matricula, idModelo ->Modelo, idConta ->Conta)

Passagem (idPassagem, data, hora, inicio ->Barreira, fim ->Barreira)

PassagemVerde (idPassagem -> Passagem, idVeiculo -> Veiculo)

PassagemNormal (idPassagem -> Passagem, idCategoria -> Categoria

9. RED BULL AIR RACE

We want to store information about one season of the Red Bull Air Race competition. Pilots participate in this competition and are organized into teams. Each pilot competes throughout the season with a single aircraft model.

Each pilot is identified by a number and it is intended to store his or her first and last name, as well as their nationality and date of birth. For a team, you only need to know their name and country of origin. However, each airplane model is characterized by its power, maximum speed length, width and weight.

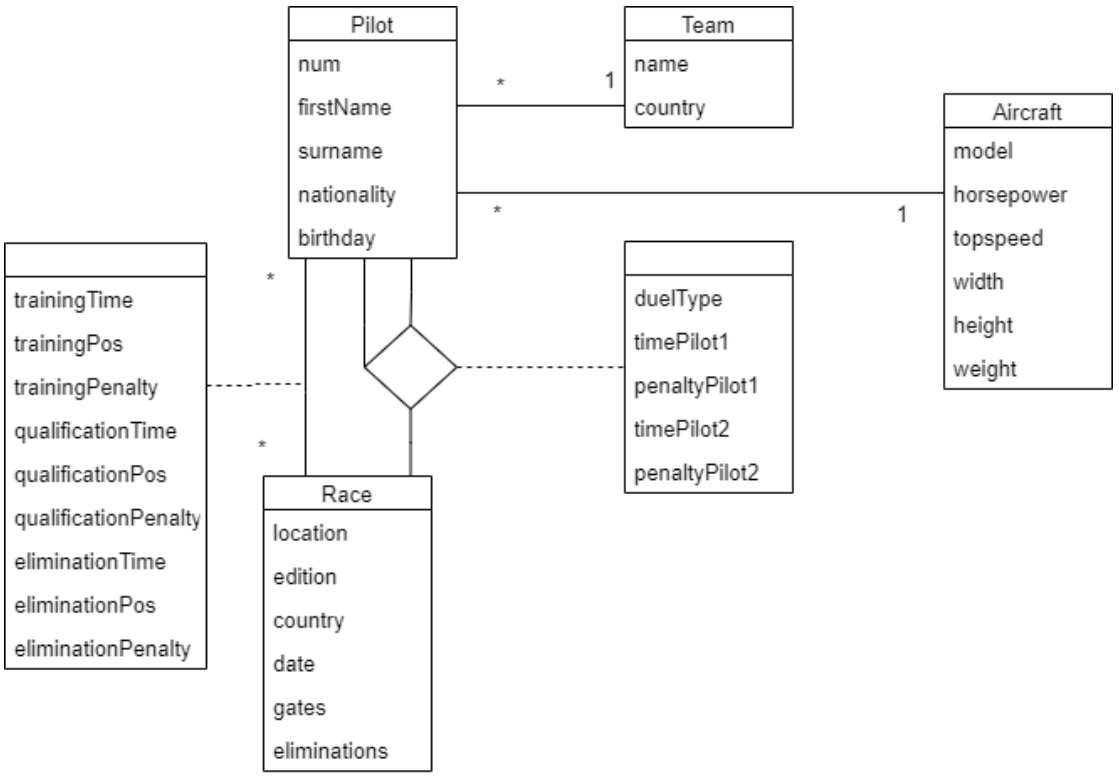
Throughout the season, drivers compete in races that take place in a given city in a given year. It is interesting to know in which country and on which date each race took place as well as the number of pylons in the race and how many drivers were eliminated in the first round.

The races follow a 'two-stage model'. The first phase consists of running the course 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 driver as well as the penalty seconds. The times obtained in the Free Practice phase dictate the order of departure for Qualifying, and the order of departure for Elimination. The driver with the worst time in Qualifying does not go to Elimination and the four worst in Elimination do not go to Stage Two.

The Second Stage is a Duel stage in which the drivers compete directly 2-on-2 from the Quarter Finals to the Final. It is important to keep track of who competed in which duel, in which stage, with what time and with what penalty.

The scoring system is linear, awarding 6 points to the winner up to 1 point to the 6th classified. [Based in an exercise by Gabriel David, Vasco Vinhas and André Restivo]

I. Conceptual Model



II. Relational schema

Team (<u>name</u>, country)

Aircraft (<u>model</u>, horsepower, topspeed, width, height, weight)

Pilot (<u>num</u>, firstname, surname, nationality, birthday, name -> Team, model -> Aircraft)

Race (<u>location</u>, <u>edition</u>, country, date, gates, eliminations)

Participation (num -> Pilot, [<u>location, edition</u>] -> Race, trainingtime, trainingpos, trainingpenalty, qualificationtime, qualificationpos, qualificationpenalty)

Duel (<u>numpilot1</u> -> Pilot, <u>numpilot2</u> -> Pilot, [<u>location, edition</u>] -> Race, dueltype, timepilot1, timepilot2, penaltypilot1, penaltypilot2)