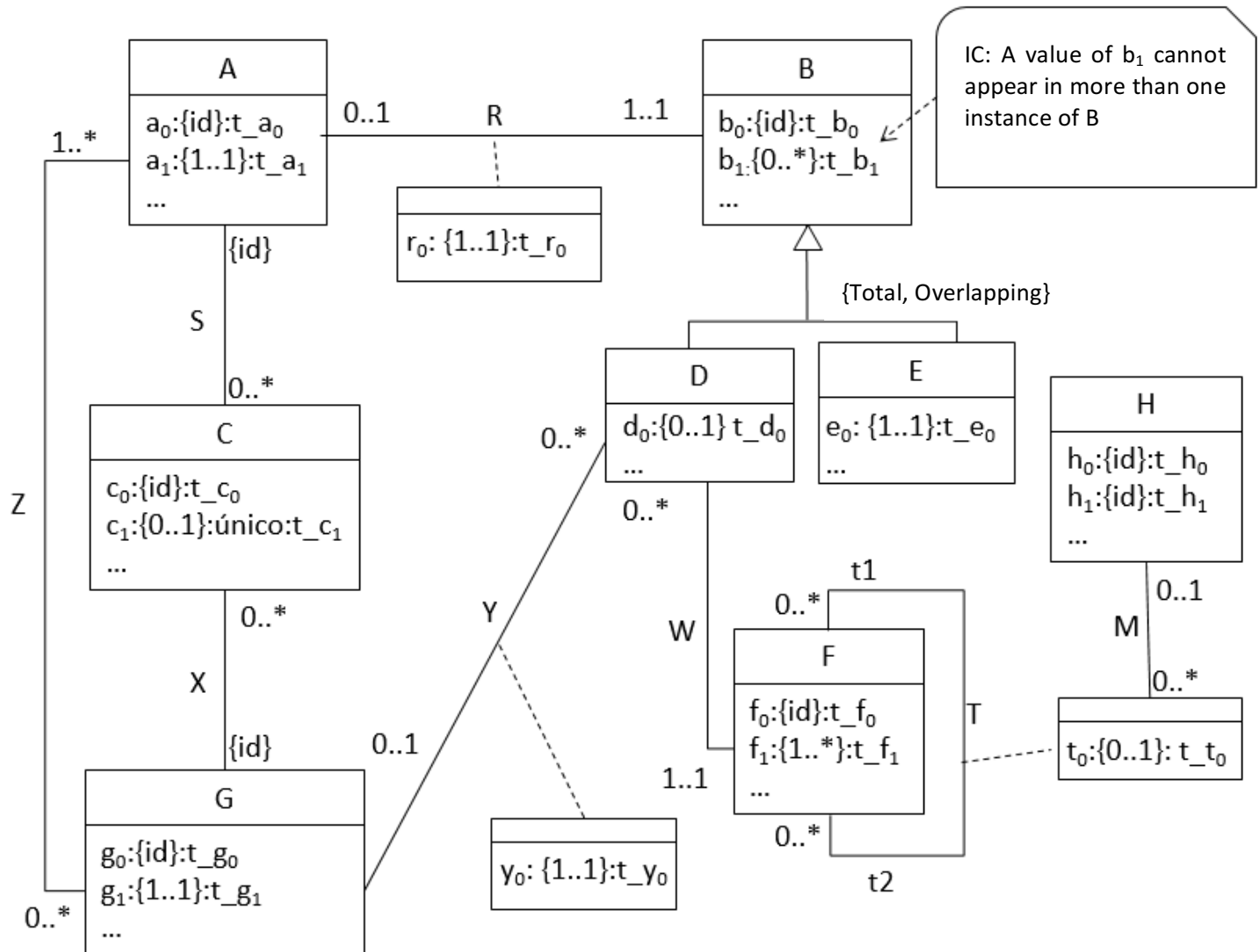


1. Perform the logical design of the following UML class diagram in order to obtain the equivalent set of relations in the relational model. Those constraints that cannot be expressed graphically can be written in natural language (English or Spanish). **(1.5 points)**.



2. Consider the following relational schema:

$R(A: \text{int}, B: \text{int}, C: \text{char}, D: \text{int}, E: \text{txt}, F: \text{char}, G: \text{int}, H: \text{char})$

PK: {A, B, C}

NNV: {D, E, F, G, H}

From the dependencies shown below, transform the relation to a set of relations in third normal form. **(0.5 points)**.

$\{A\} \rightarrow \{F\}$

$\{B, C\} \rightarrow \{D\}$

$\{F\} \rightarrow \{G\}$

$\{D\} \rightarrow \{H\}$

3. Design a UML class diagram for the following information system. Those constraints that cannot be expressed graphically, can be written in natural language (English or Spanish). **(1.5 points)**

We want to save the information of the different seasons of a popular running circuit. A season of the running circuit consists of 10 races. The set of 10 races may vary from one season to another. The technical staff will be responsible for registering the data of each new season of the running circuit: the year of the season (which identifies it) and the races that are part of the circuit during that season. If any of the races is new, it will be registered in the system with a name (which is unique), and a location (that is mandatory). Each edition of a race is identified by an edition number (that is unique for each race). Each edition of a race has also a date, a starting time, its distance in meters, and the organizing entity. Of the organizing entity, we want to store its name (which will be used as identifier) and a contact person. A person will have a SSN, a name, a last name, some telephone numbers, and an email address (all these values are mandatory).

A runner can register to one season of the running circuit paying the entry fee before the end of the registration period of the season. The registration is valid for all the races in the same season of the running circuit. Both the entry fee and the registration period must be known for each season of the circuit. During registration, the runner will have to introduce his/her personal data (SSN, name, last name, date of birth, and gender), contact information (telephones and email address) and the number of his/her credit card (which cannot be used by another runner). All these data are required. If all the information is valid, the system will assign a category to the runner (depending on his/her gender and age), and his number (dorsal) and chip number for the whole season of the running circuit.

Runners may register for a season of the running circuit as members of a club. We want to store the club number and the representative person of the club. Some clubs are members of the International Running Association and therefore, the club has an id number and an official address. During a season, a runner cannot change to other club, but he/she can register through different clubs in different seasons of the running circuit.

The technical staff will publish the results of each race at the end of it. For each participating runner, they will publish his/her position in the classification of the race, his position in the category, his official time, and his real time. The classification can be consulted both by the runners and by the technical staff.

4. Consider the following relational schema

```
CREATE TABLE Assistant (
  SSN CHAR(10) CONSTRAINT pk_assistant PRIMARY KEY DEFERRABLE,
  name VARCHAR(50) NOT NULL,
  phone CHAR(10) NOT NULL,
  num INTEGER DEFAULT 0 NOT NULL)

CREATE TABLE Trip (
  id char(5) CONSTRAINT pk_trip PRIMARY KEY DEFERRABLE,
  SSN CHAR(10) CONSTRAINT fk_trip_assistant REFERENCES Assistant(SSN) DEFERRABLE,
  tdate DATE NOT NULL)
```

where:

- The **ASSISTANT** relation stores the trip assistants data: *SSN*, *name*, *phone*, and *num* (where *num* contains the total amount of trips assigned to the assistant).
- the **TRIP** relation stores the information associated to the trips: *id* of the trip, *SSN* of the assistant assigned to the trip, and the date (*tdate*) of the trip.

Implement a trigger in Oracle PL-SQL to update the *num* attribute of the ASSISTANT relation when a new TRIP is INSERTED. **(0.5 points)**

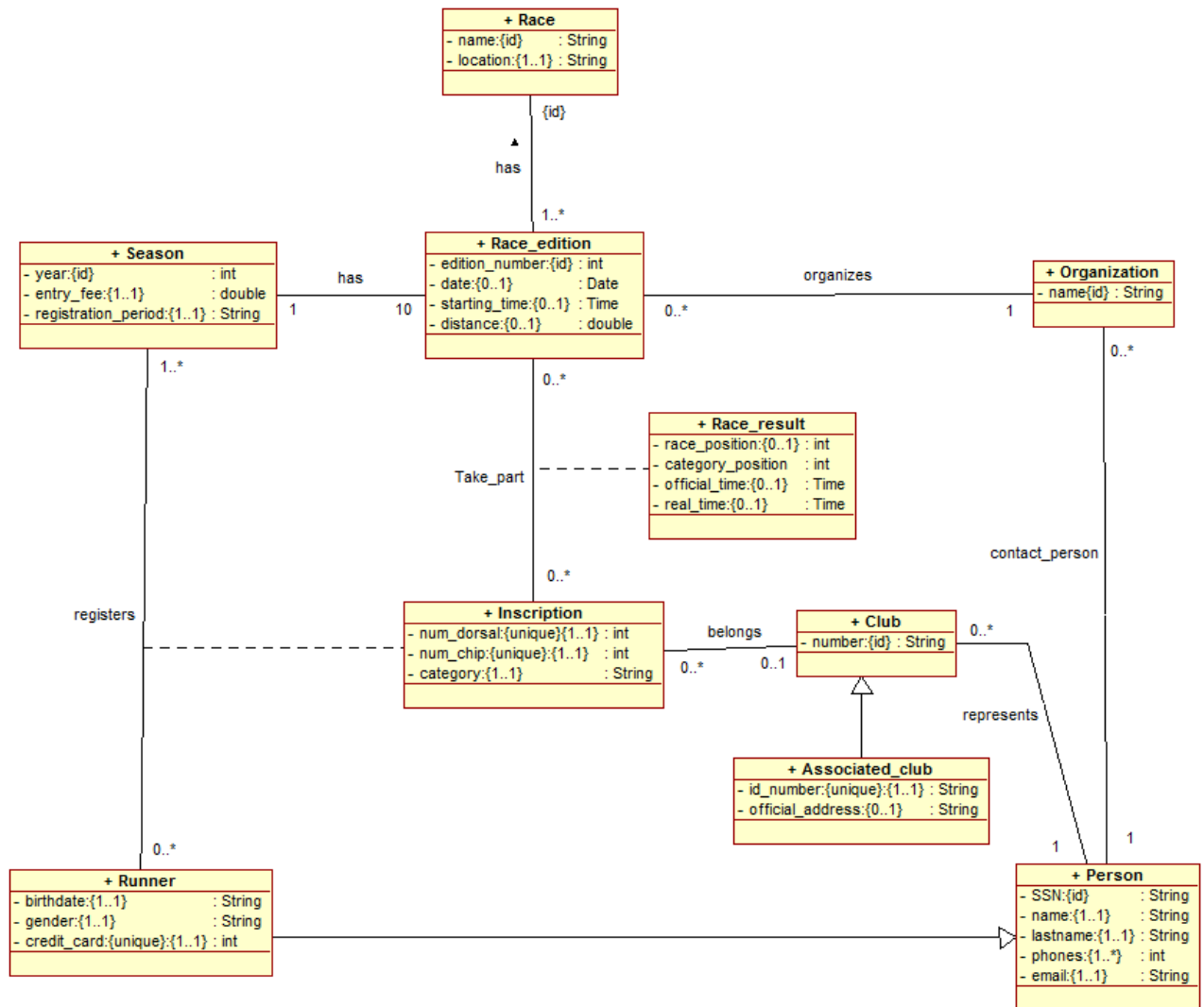
1.

<p>A(a0:t_a0, a1:t_a1, ..., b0:t_b0, r0:t_r0) PK:{a0} NNV:{a1, b0, r0} FK:{b0} → B Uni:{b0}</p> <p>B(b0:t_b0,...) PK:{b0}</p> <p>B1(b1:t_b1, b0:t_b0) PK:{b1} NNV:{b0} FK:{b0} → B</p> <p>C(c0:t_c0, c1:t_c1, ..., a0:t_a0, g0:t_g0) PK{c0, a0, g0} Uni:{c1} FK:{a0} → A FK:{g0} → G</p> <p>D(b0:t_b0, d0:t_d0, ..., f0:t_f0) PK:{b0} FK:{b0} → B FK:{f0} → F(f0) NNV:{f0}</p> <p>E(b0:t_a0, e0:t_e0, ...) PK:{b0} FK:{b0} → B NNV:{e0}</p> <p>G(g0:t_g0, g1:t_g1, ...) PK{g0} NNV:{g1}</p>	<p>H(h0:t_h0, h1:t_h1, ...) PK{h0, h1}</p> <p>F(f0:t_f0, ...,) PK{f0}</p> <p>F1(f1:t_f1, f0:t_f0) PK:{f1, f0} FK:{f0} → F</p> <p>Y(g0:t_g0, b0:t_b0, y0:t_y0) PK:{b0} FK:{g0} → G(g0) FK:{b0} → D(b0) NNV:{g0, y0}</p> <p>Z(a0:t_a0, g0:t_g0) PK{a0, g0} FK:{a0} → A(a0) FK:{g0} → G(g0)</p> <p>T(f0_t1:t_f0, f0_t2:t_f0, t0:t_t0, h0:t_h0, h1:t_h1) PK{f0_t1, f0_t2} FK:{f0_t1} → F(f0) FK:{f0_t2} → F(f0) FK:{h0, h1} → H(h0, h1)</p> <p>RI_{min_G_in_Z}: Every value in g0 of G must appear in g0 of Z.</p> <p>RI_{min_f1}: Every value in f0 of F must appear in f0 of F1.</p> <p>RI_{Total}: Every value which appears in the attribute b0 of B must appear in the attribute b0 of D or E.</p>
---	--

2.

<p>R(A: int, B: int, C: char, E: txt) PK: {A, B, C} NNV: {E} FK: {A}→R1 FK: {B,C}→R2</p> <p>R1(A: int, F: char) PK: {A} NNV: {F} FK: {F}→R12</p> <p>R12(F: char, G: int) PK: {F} NNV: {G}</p>	<p>R2(B: int, C: char, D: int) PK: {B, C} NNV: {D} FK: {D}→R22</p> <p>R22(D: int, H: char) PK: {D} NNV: {H}</p> <p>All the values of {A} in R1 are also in R All the values of {F} in R12 are also in R1 All the pairs of values of {B,C} in R2 are also in R. All the values of {D} in R22 are also in R2</p>
---	--

3.



4.

```

CREATE OR REPLACE TRIGGER T_trip_ins
AFTER INSERT ON Trip
FOR EACH ROW
WHEN new.SSN IS NOT NULL
BEGIN
UPDATE Assistant SET num = num+1
WHERE SSN = :new.SSN;
END;

```