COLLECTION OF REQUIRMENTS:

Create a database for managing shooters who represent Benelli Armi SPA in the shooting world. Shooters can be employees or people from outside the company. Each of them owns their own rifle. Some of the shooters are part of the company's teams, others shoot independently (but with a rifle purchased by the company). The company has two teams: BE TEAM (they shoot in a discipline called Dynamic shooting) and 828U TEAM (they shoot in skeet shooting). The shooters who are not part of these two teams are gathered in the BE PASSION. The shooters participate in competitions (individually, not as a team), national or international, and the results they obtain (if they get on the podium, they have cash prizes) must be taken into account. The rifles with which shooters shoot can be semi-automatic (used in dynamic shooting), pump guns (used in dynamic shooting) or over-and-under (used in skeet shooting). Each of these will have its own particularities (e.g. barrel length, stock length, right or left-handed grip). At the competitions the members of the teams are accompanied by some staff members of the sports function (the head of the function, the client assistance, an external consultant, and an intern).

2.a) **Glossary of terms**:

Terms	Description	Synonyms	Links
Shooters	Shooters can be employees or people from outside the company.		Rifle, Team,Competition
Rifle	Each shooter owns his rifle.		Shooters, Typology, Particularities
Typology	Rifle can be semi-automatic, pump guns or over-and-under.		Rifle, Particularities
Particularities	Each typology can have its own barrel-length, stock length…		Rifle, Typology
Team	Shooter can be part of one of the three teams.		Shooters
Competition	Can be national or international, recording the result of the competition	Tournament	Shooters, Staff
Staff	The staff members goes to the competition with the shooters.		Competition,

*= the prize of the competition is given by the staff to the shooter.

2.b) **Table of operation**:

N°	Operation	Average Frequency (Number of times)	Type
1	Insert a shooter in the team	15 time per year	I
2	Insert a new competition	20-30 times per year	I
3	Insert new staff	1 time every 6 months	I
4	Shooters that participate to competitions	15 per week	I

5	Select podiums of a certain competition	1 per week	I
6	Print the results of the tournament edition of the year	1 per year	B
7	Staff that participate to a competition	4 per competition	I
8	Select shooters of a team that participate at a competition	1 per week	I
9	Print the shooters with the greatest number of partecipation	1 per year	B

3.b) **Table of Volume**

IS: Assuming that for every tournament edition 15 shooter participate (tournament edition are 30 per year)

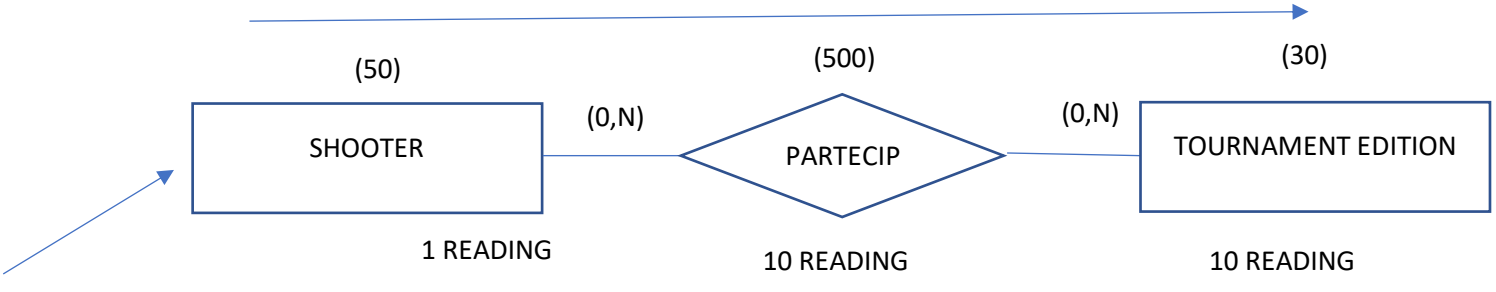
SUPPORT: assuming that for every tournament edition there are 3 staff members.

PARTICULARS: for every rifle there are 3 attachments, but every attachments can be various.

Concept	**Construct**	**Volume**
Shooter	E	50
Team	E	3
Rifle	E	50
Typology	E	3
Particulars	E	3
Tournament	E	30
Tournament edition	E	30
Staff	E	90
Member	R	50
Partecipant	R	500
Is	R	30
Support	R	3
Possesion	R	50
Type	R	50
Different	R	150

3.c) **Cost assessments, Table of accesses and navigation Schema**

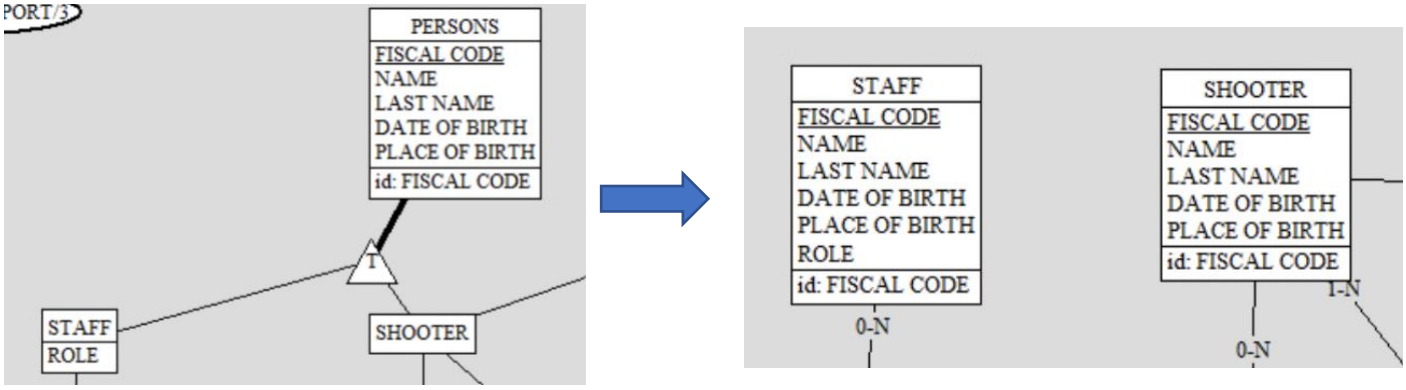
Operation: for each shooter, find the tournament editions he took part in. Assuming that each shooter take place, on average, at 10 tournament editions.



Concept	Type	Accesses	Type
Shooter	E	1	R
Partecipant	R	10	R
Tournament Edition	E	10	R

4.a) Conceptual to logic model

-Deleting generalization hierarchies: In the schema it's present a generalization in which the parent entity is "Person", and the two child entities are "Staff and "Shooter". We decided to delete it collapsing downwards. this decision is motivated by the fact that the two child entities play a different role in the database and so in the main operations the access to them are distinct. This is possible because the coverage of the generalization is total.



All the attributes of the parent entity will then be replicated in the two child entities.

-Selection of key identifiers:

ENTITIES	KEY IDENTIFIERS
STAFF	FISCAL CODE
SHOOTER	FISCAL CODE
RIFLE	RIFLE CODE
TEAM	NAME
TOURNAMENT	NAME, PLACE, ORGANIZER
TOURNAMENT EDITION	EXTERNAL IDENTIFIER: NAME, PLACE, ORGANIZER (TOURNAMENT), YEAR
TYPOLOGY	TIPOLOGY CODE
PARTICULAR	PARTICULAR CODE

4.b) Relational schema

STAFF (fiscal code, name, last name, date of birth, place of birth, role)

SHOOTER (fiscal code, name, last name, date of birth, place of birth, team)

-Fk: team reference TEAM and will assume as values the name of the teams

TEAM (name, discipline)

RIFLE (rifle code, name, shooter, typology)

-fk: shooter reference SHOOTER and will assume as values the fiscal codes of the shooters;

-fk: typology reference TYPOLOGY and will assume as values the codes of the typologies;

DIFFERENT (rifle, particulars, value)

-fk: riflw references RIFLE, and will assume as values the codes of the rifles;

-fk: particulars references PARTICULARS, and will assume as values the code of the particulars;

TYPOLOGY (typology code, name)

PARTICULARS (particular code, description)

TOURNAMENT (name, place, organizer, discipline)

TOURNAMENT EDITION (name, place, organizer, year, cash prize)

-fk: name, place, organizer reference TOURNAMENT

SUPPORT (staff, name, place, organizer, year)

-Fk: staff reference STAFF and will assume as values the fiscal codes of the staff members;

-fK: name, place, organizer, year reference TOURNAMENT EDITION;

PARTECIPANT (shooter, name, place, organizer, year)

Fk: shooter reference SHOOTER and will assume as values the fiscal codes of the shooters;

-fK: name, place, organizer, year reference TOURNAMENT EDITION;

-SQL CREATION SCHEMA

1. **CREATE TABLE** 'Staff' (

'Fiscal Code' VARCHAR (16) PRIMARY KEY,

'Name' VARCHAR (16). NOT NULL,

'Last name' VARCHAR (16) NOT NULL,

'Date of birth' DATE (10) NOT NULL,

'Place of birth' NOT NULL

'Role' DEFAULT 'Technician',

);

2. **CREATE TABLE** 'Shooter' (

'Fiscal Code' VARCHAR (16) PRIMARY KEY,

'Name' VARCHAR (16). NOT NULL,

'Last name' VARCHAR (16) NOT NULL,

'Date of birth' DATE (10) NOT NULL,

'Place of birth' VARCHAR(16) NOT NULL,

'Team' VARCHAR(10 REFERENCES Team (Name)

);

3. **CREATE TABLE** 'Team' (

'Name' VARCHAR (10) PRIMARY KEY,

'Description' VARCHAR (16). NOT NULL,

);

4. **CREATE TABLE** 'Rifle' (

'Rifle Code' VARCHAR (16) PRIMARY KEY,

'Name' VARCHAR (16). NOT NULL,

'Shooter' VARCHAR(16) UNIQUE REFERENCES Shooter (Fiscal code),

'Typology' VARCHAR(10) REFERENCES Typology (Typology code),

''Particulars' VARCHAR(10) REFERENCES Particulars (Particular code),

‘Value’ VARCHAR(10) DEFAULT ‘Technician’,

);

5. **CREATE TABLE** ‘Typology’ (

‘Typology Code’ VARCHAR (10) PRIMARY KEY,

‘name’ VARCHAR (16). NOT NULL,

);

6. **CREATE TABLE** ‘Particulars’ (

‘Particular Code’ VARCHAR (10) PRIMARY KEY,

‘Description’ VARCHAR (16) NOT NULL,

);

7. **CREATE TABLE** ‘Tournament’ (

‘Name VARCHAR (16)

‘Place’ VARCHAR (16),

‘Organizer’ VARCHAR(16),

‘Discipline’ VARCHAR(12) NOT NULL,

PRIMARY KEY (Name, Place, Organizer)

);

8. **CREATE TABLE** ‘Tournament Edition’ (

‘Name’ VARCHAR (16),

‘Place’ VARCHAR (16),

‘Organizer’ VARCHAR(16),

‘Year’ VARCHAR(12),

PRIMARY KEY (Name, Place, Organizer, Year)

);

9. **CREATE TABLE** 'Support' (

'Staff' VARCHAR (16) REFERENCES Staff (Fiscal Code),

'Name' VARCHAR (16) REFERENCES Staff (Tournament Edition),

'Place' VARCHAR (16) REFERENCES Staff (Tournament Edition),

'Organizer' VARCHAR(16) REFERENCES Staff (Tournament Edition),

'Year' VARCHAR(12) REFERENCES Staff (Tournament Edition),

PRIMARY KEY (Name, Place, Organizer, Year)

);

10. **CREATE TABLE** 'Partecipant' (

'Shooter' VARCHAR (16) REFERENCES Shooter (Fiscal Code),

'Name' VARCHAR (16) REFERENCES Staff (Tournament Edition),

'Place' VARCHAR (16) REFERENCES Staff (Tournament Edition),

'Organizer' VARCHAR(16) REFERENCES Staff (Tournament Edition),

'Year' VARCHAR(12) REFERENCES Staff (Tournament Edition),

PRIMARY KEY (Name, Place, Organizer, Year)

);

-SQL QUERIES

1. DISPLAY THE SHOOTERS OF '828U TEAM':

SELECT s.fiscalCode, s.firstName, s.lastName

FROM rifle r JOIN shooter s ON (r.shooter = s.fiscalCode) JOIN team t ON (s.team = t.name)

WHERE t.name = '828U team'

2. FOR EVERY TOURNAMENT EDITION OF THE ‘VALTELLINA TOURNAMENT’, WRITE THE FISCAL CODE OF THE STAFF THAT SUPPORT THE SHOOTERS IN THE TOURNAMENTS

```
SELECT te.year AS Edition, s.staff

FROM tournament t NATURAL JOIN tournamentEdition te NATURAL JOIN support s

WHERE t.name = ‘Valtellina Tournament’

GROUP BY Edition
```

3. COUNT THE NUMBER OF SHOOTERS WITH A “NOVA SPEED” RIFLE

```
SELECT COUNT(r.shooter) as NShootWithNova

FROM rifle r

WHERE r.name = ‘Nova Speed’
```