



Università degli Studi di Salerno



Dipartimento di Ingegneria dell'Informazione ed Elettrica e
Matematica Applicata

Corso di Laurea in Ingegneria Informatica

Basi di Dati 2021/2022
Canale I-Z

Project Work

Traccia N. 8 – Rifugiati e Immigrati

Gruppo n. **61 – IZ**

WP	Cognome e Nome	Matricola	e-mail	Responsabile
1	Orlando Palma	0612704791	p.orlando8@studenti.unisa.it	
2	Roberto Gaetano	0612704696	g.roberto6@studenti.unisa.it	
3	Peluso Raffaele	0612704728	r.peluso16@studenti.unisa.it	X
4	Noschese Valerio	0612704650	v.noschese@studenti.unisa.it	

Anno accademico 2021-2022

Sommario

Descrizione della realtà di interesse	3
Analisi della realtà di interesse	4
Analisi delle specifiche	5
Glossario dei termini	5
Strutturazione dei requisiti in frasi	5
Frase di carattere generale	5
Frase relative a ente	5
Frase relative a rifugiato	5
Frase relative a progetto	5
Frase relative a struttura	6
Frase relative a servizio	6
Frase relative a corso	6
Identificazione delle operazioni principali	6
Progettazione Concettuale	7
Schema Concettuale	7
Design Pattern	8
Pattern Storizzazione di Associazione	8
Pattern Instance-of	8
Dizionario dei Dati	9
Regole Aziendali	11
Progettazione Logica	12
Ristrutturazione Schema Concettuale	12
Analisi delle Prestazioni	12
Tavola dei volumi	12
Tavola delle operazioni	12
Analisi delle ridondanze	13
Analisi della ridondanza 1: Numero Partecipanti	13
Valutazione della ridondanza 1	16
Eliminazione delle generalizzazioni	18
Generalizzazione Progetto	18
Partizionamento/Accorpamento Entità e Associazioni	19
Scelta degli identificatori principali	19
Schema ristrutturato finale	20
Schema logico	21
Documentazione dello schema logico	21

Normalizzazione	22
Script Creazione e Popolamento Database	22
Query SQL	29
Query con operatore di aggregazione e join: statistica enti	29
Query nidificata complessa: servizi più utili	29
Query insiemistica: percorso del rifugiato	29
Viste	30
Vista proposta_entri	30
Query con Vista: enti più attivi per ogni vulnerabilità	30
Trigger	31
Trigger inizializzazione: check_lingua	31
Trigger inizializzazione: check_partecipazione_corrente	31
Trigger per vincoli aziendali	32
Trigger1: check_edizione_progetto	32
Trigger2: check_vulnerabilita	32
Trigger3: aggiorna_partecipanti	33
Trigger4: verifica_capacita	34

1. Descrizione della realtà di interesse

Titolo: I-Z - Traccia 8 – Rifugiati e Immigrati

La principale fonte istituzionale di dati sui migranti irregolari arrivati in Italia è il Ministero dell'Interno. Quotidianamente il Viminale pubblica il cruscotto statistico giornaliero, in cui viene riportato il numero di migranti sbarcati nel nostro paese da inizio anno, il confronto con la stessa data degli anni precedenti e la distinzione per Paese di origine dei migranti. Altri dati che possono facilmente essere reperiti sul sito del ministero sono quelli relativi alle richieste di asilo. In particolare, appaiono di attualità dati e informazioni circa i rifugiati politici o di guerra. Rispetto agli arrivi, un'altra fonte autorevole da considerare è l'UNHCR che tiene traccia dei flussi in tutte le principali rotte mediterranee. Ai fini del presente elaborato è di interesse il seguente articolo relativo allo status di rifugiato e l'identità politica dell'accoglienza di Stefano Celentano https://www.questionegiustizia.it/rivista/articolo/lo-status-dirifugiato-e-l-identita-politica-dell-accoglienza_534.php, come altra documentazione di natura anche giuridica facilmente reperibile in rete. Anche in relazione ai recenti fatti relativi all'invasione della Russia in Ucraina, è di interesse la realizzazione di una base di dati che consenta ai familiari, amici e conoscenti di ottenere informazioni relative ai cittadini ucraini (e non solo) che sono scappati dall'Ucraina (e non solo) e si sono rifugiati in luoghi, paesi, centri di accoglienza, ecc. In particolare, sono di interesse informazioni circa:

- Anagrafica del rifugiato e persone con cui è fuggito, località di partenza, mezzi utilizzati, percorso, soste e paese o località target.
- Con riferimento alla località target, l'informazione da memorizzare può essere presente a diversi livelli di dettaglio e di verità. Ovvero, circa la descrizione della località di arrivo e sosta, questa può essere nota con notevole precisione fino ad arrivare a fornire il numero civico e l'interno, nonché eventualmente la famiglia che ospita il rifugiato, mentre in altri casi questa è nota solo in maniera approssimativa, ad esempio si conosce soltanto la Nazione che ospita il rifugiato, o la regione, la provincia, la città, l'ente, ecc; circa il livello di verità, questa è determinata da chi ha fornito l'informazione relativa al target in cui si trova il rifugiato: l'informazione può essere inserita direttamente dal rifugiato che intende far sapere alla comunità il suo stato, il suo viaggio, il suo rifugio, ecc. ; oppure da un'organizzazione o ente; oppure da persone che hanno informazioni (anche approssimate) in tal senso.
- Rispetto alle persone (o organizzazioni) che hanno fornito informazioni circa la località in cui si trova il rifugiato, è di interesse memorizzare le generalità e altri dati, oltre all'eventuale legame con la persona censita. Il livello di relazione può essere quello di un semplice conoscente (ad esempio una persona che ha visto partire il rifugiato, piuttosto che ha viaggiato per un tratto insieme, ecc.) fino ad un familiare che eventualmente è con il rifugiato. Esempi di possibili variazioni: il database può essere specializzato rispetto alla descrizione del rifugiato e del percorso eseguito, nonché ad una dettagliata descrizione degli eventi che hanno determinato la partenza e quelli rilevanti verificatisi durante il viaggio; oppure maggiormente specializzato rispetto alle persone e organizzazioni che forniscono informazioni circa il rifugiato e il suo viaggio. In alternativa alla banca dati dei rifugiati, è possibile articolare un'analoga banca dati (con le dovute modifiche) per gli immigrati clandestini che tipicamente vengono censiti nei centri di accoglienza.

1.1. Analisi della realtà di interesse

L'obiettivo è di progettare una base di dati per memorizzare le informazioni relative al processo di integrazione dei rifugiati, realizzato tramite la promozione di progetti, con lo scopo di offrire la struttura alloggiativa, dei servizi e dei corsi. Si suppone di trattare un database gestito dall'UE e utilizzato da applicazioni web per gli enti locali e per i visitatori con funzioni differenti.

Ai fini di un'accurata descrizione del processo sociale di interesse, è rilevante individuare il contesto culturale e professionale di un rifugiato, del quale si intende memorizzare il nome, il cognome, la data di nascita, il Paese di provenienza, la religione professata, l'eventuale titolo di studio, l'eventuale qualifica professionale, le lingue parlate, l'eventuale vulnerabilità (minore non accompagnato, disabile, donna in gravidanza), le edizioni dei progetti cui partecipa e ha partecipato. Si suppone che il rifugiato, essendo stato già accolto nella fase precedente a quella di interesse, sia in possesso di un documento d'identità.

Agli enti locali possono essere assegnati dei progetti di integrazione per i rifugiati. Per ciascun ente è necessario memorizzare il codice fiscale, la denominazione, il numero di telefono, la mail, la sede con nazione, città e i progetti assegnati.

Per ciascun progetto si intende memorizzare il codice identificativo assegnato dall'UE, il titolo, l'ambito di interesse, l'area territoriale, l'ente responsabile. Inoltre, i progetti possono essere ordinari o adatti all'integrazione di rifugiati con vulnerabilità (minore non accompagnato, disabile, donna in gravidanza). È possibile avere diverse edizioni del progetto, con una data di inizio e fine e il numero dei partecipanti. Ciascuna edizione del progetto deve disporre di una struttura alloggiativa per ospitare i rifugiati partecipanti e può offrire servizi o corsi. Si suppone che il rifugiato che partecipa ad un'edizione di progetto usufruisca dei servizi e corsi proposti in quell'edizione di progetto.

Per la struttura si intende memorizzare il cis (codice identificativo di struttura), il nome, l'indirizzo, la capienza, se offre ospitalità per i disabili, se possiede una mensa.

Per un servizio si intende memorizzare il codice identificativo assegnato dall'UE, la tipologia, l'attività, se è individuale, il luogo dove si svolge.

Per un corso si intende memorizzare il codice identificativo assegnato dall'UE, la descrizione, la durata in ore, la materia, l'eventuale qualifica conseguita, se viene rilasciato un attestato o meno, il luogo dove si svolge.

È bene sottolineare che un rifugiato, in un certo istante di tempo, può partecipare ad una sola edizione di progetto, assegnato a un solo ente. Gli stessi servizi e corsi possono essere offerti in diverse edizioni di progetti.

Per la specializzazione descritta, si intende tralasciare la descrizione dei dettagli relativi al viaggio del rifugiato (tappe, persone con cui è fuggito, dettagli sulla località d'arrivo), ponendo invece l'accento sulla fase successiva all'arrivo, quindi relativa all'integrazione.

2. Analisi delle specifiche

Workpackage	Task	Responsabile
WPO	Analisi delle specifiche	Intero Gruppo

2.1. Glossario dei termini

	Termine	Descrizione	Sinonimi	Collegamenti
1	Ente	Istituzione locale assegnata a dei progetti.	-	Progetto
2	Rifugiato	Partecipante di progetti.	-	Progetto
3	Progetto	Progetti assegnati a enti. Possono avere varie edizioni. Offrono al rifugiato struttura, servizi e corsi.	-	Struttura, Servizio, Corso, Ente, Rifugiato
4	Struttura	Struttura alloggiativa dove vengono ospitati i rifugiati partecipanti a un'edizione di un progetto.	-	Progetto
5	Servizio	Servizi offerti al rifugiato all'interno di un'edizione di un progetto.	-	Progetto
6	Corso	Corsi proposti al rifugiato all'interno di un'edizione di un progetto.	-	Progetto

Tabella 1. Glossario dei Termini

2.2. Strutturazione dei requisiti in frasi

2.2.1. Frasi di carattere generale

L'obiettivo è di progettare una base di dati per memorizzare le informazioni relative al processo di integrazione dei rifugiati, realizzato tramite la promozione di progetti, con lo scopo di offrire la struttura alloggiativa, dei servizi e dei corsi.

2.2.2. Frasi relative a ente

Per ciascun ente, identificato da un codice fiscale, si intende memorizzare la denominazione, il numero di telefono, la mail, la sede con nazione, città e i progetti assegnati.

2.2.3. Frasi relative a rifugiato

Per ciascun rifugiato, identificato dal documento di identità, si intende memorizzare il nome, il cognome, la data di nascita, il Paese di provenienza, la religione professata, l'eventuale titolo di studio, l'eventuale qualifica professionale, le lingue parlate, l'eventuale vulnerabilità (minore non accompagnato, disabile, donna in gravidanza), le edizioni dei progetti cui partecipa e ha partecipato.

2.2.4. Frasi relative a progetto

Per ciascun progetto, identificato dal codice assegnato dall'UE, si intende memorizzare il titolo, l'ambito di interesse, l'area territoriale, l'ente responsabile, le edizioni con una data di inizio e fine e il numero dei partecipanti. Inoltre, i progetti possono essere ordinari o adatti all'integrazione di rifugiati con vulnerabilità (minore non accompagnato, disabile, donna in gravidanza). Un rifugiato, in un certo istante di tempo, può partecipare ad una sola edizione di progetto.

2.2.5. Frasi relative a struttura

Per la struttura, identificata dal cis, si intende memorizzare il nome, l'indirizzo, la capienza, se offre ospitalità per i disabili, se possiede una mensa.

2.2.6. Frasi relative a servizio

Per un servizio, identificato dal codice assegnato dall'UE, si intende memorizzare la tipologia, l'attività svolta, se è individuale, il luogo dove si svolge.

2.2.7. Frasi relative a corso

Per un corso, identificato dal codice assegnato dall'UE, si intende memorizzare la descrizione, la durata in ore, la materia, l'eventuale qualifica conseguita, se viene rilasciato un attestato o meno, il luogo dove si svolge.

2.3. Identificazione delle operazioni principali

Operazione 1: Stampa lo storico dei progetti di un rifugiato, indicando le strutture alloggiative (2000 al giorno).

Operazione 2: Stampa lo storico dei progetti di un rifugiato, indicando i servizi e i corsi offerti (2000 al giorno).

Operazione 3: Individua le edizioni correnti di un progetto per una vulnerabilità, visualizzando anche il numero dei partecipanti (30 al giorno).

Operazione 4: Effettua una statistica sui paesi di provenienza dei rifugiati (1 volta al mese).

Operazione 5: Effettua una statistica sulle religioni dei rifugiati (1 volta al mese).

Operazione 6: Effettua una statistica sui progetti assegnati agli enti di ogni nazione (1 volta all'anno).

3.2. Design Pattern

3.2.1. Pattern Storizzazione di Associazione

Dalla realtà di interesse si evince la volontà di tenere traccia, per un rifugiato, delle edizioni cui partecipa e ha partecipato, quindi si storicizza l'associazione partecipazione secondo il pattern in partecipazione corrente e passata.

In questo modo si tiene conto dello storico delle partecipazioni di un rifugiato alle edizioni di progetti.

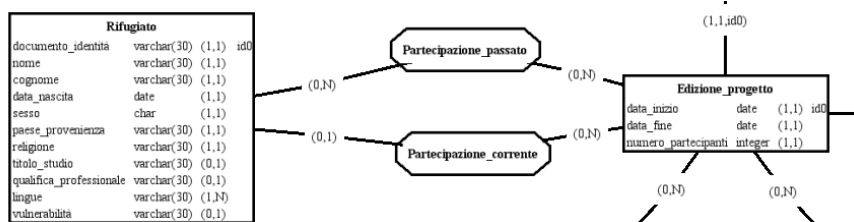


Figura 2. Schema successiva all'applicazione del Pattern Storizzazione di Associazione.

3.2.2. Pattern Instance-of

L'entità Progetto descrive il concetto astratto di progetto con un codice, un titolo, un ambito e l'area territoriale mentre Edizione progetto rappresenta il progetto reale, ovvero l'istanza di un progetto con una determinata data d'inizio, di fine e numero di partecipanti.

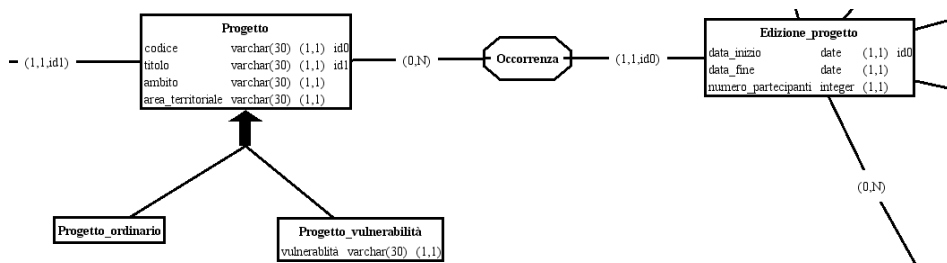


Figura 3. Schema successiva all'applicazione del Pattern Instance of.

3.3. Dizionario dei Dati

Entità	Descrizione	Attributi	Identificatore
Ente	Organizzazione a cui vengono assegnati dei progetti	Codice fiscale, Denominazione, Telefono, Mail, Sede	Codice fiscale
Progetto	Progetti cui prendono parte i rifugiati	Codice, Titolo, Ambito, Area territoriale	Codice, Titolo + Codice fiscale
Progetto ordinario	Particolare tipologia di Progetto	Codice, Titolo, Ambito, Area territoriale	Codice, Titolo + Codice fiscale
Progetto vulnerabilità	Particolare tipologia di Progetto	Codice, Titolo, Ambito, Area territoriale, Vulnerabilità	Codice, Titolo + Codice fiscale
Rifugiato	Persona che prende parte ai progetti	Documento d'identità, Nome, Cognome, Data di nascita, Sesso, Paese di provenienza, Religione, Titolo di studio, Qualifica professionale, Lingue, Vulnerabilità	Documento d'identità
Edizione progetto	Edizione del progetto	Data d'inizio, Data di fine, Numero partecipanti	Data d'inizio + Identificatore(Progetto)
Corso	Corso offerto al rifugiato nell'ambito di un'edizione di un progetto	Codice, Descrizione, Durata ore, Materia, Qualifica conseguita, Rilascio attestato, Luogo	Codice
Servizio	Servizio offerto al rifugiato nell'ambito	Codice, Tipologia,	Codice

	di un'edizione di un progetto	Attività, Individuale, Luogo	
Struttura alloggiativa	Struttura in cui alloggia il rifugiato nell'ambito di un'edizione di un progetto	Cis, Nome, Città, Capienza, Ospitalità disabili, Mensa, Indirizzo	Cis

Tabella 2. Dizionario dei dati – Entità

Relazioni	Descrizione	Entità Coinvolte	Attributi
Assegnazione	Assegnazione di un ente ad un progetto	Ente, Progetto	
Occorrenza	Occorrenza di un progetto in edizione	Progetto, Edizione progetto	
Proposta	Proposta di corsi nell'ambito di un'edizione di un progetto	Edizione progetto, Corso	
Offerta	Offerta di servizi nell'ambito di un'edizione di un progetto	Edizione progetto, Servizio	
Utilizzo	Utilizzo di una struttura alloggiativa nell'ambito di un'edizione di un progetto	Edizione progetto, Struttura alloggiativa	
Partecipazione corrente	Partecipazione corrente di un rifugiato a un'edizione di progetto	Edizione progetto, Rifugiato	
Partecipazione passata	Partecipazione passata di un rifugiato a un'edizione di progetto	Edizione progetto, Rifugiato	

Tabella 3. Dizionario dei dati - Relazioni

Workpackage	Task	Responsabile
WP4	Regole Aziendali	Noschese Valerio

3.4. Regole Aziendali

Regole di Vincolo
(RV1) il numero dei partecipanti di un'edizione di un progetto deve essere minore o uguale della capienza della struttura alloggiativa utilizzata
(RV2) un'edizione di un progetto di vulnerabilità per disabili deve disporre di una struttura attrezzata per tale vulnerabilità
(RV3) un'edizione progetto deve offrire almeno un servizio o un corso

Tabella 4. Regole di vincolo

Regole di derivazione
(RD1) il numero di partecipanti di una certa edizione di progetto si ottiene contando il numero di istanze dell'entità rifugiato che sono legate a questa edizione

Tabella 5. Regole di derivazione

4. Progettazione Logica

Workpackage	Task	Responsabile
WP2	Progettazione Logica	Roberto Gaetano

4.1. Ristrutturazione Schema Concettuale

4.1.1. Analisi delle Prestazioni

4.1.1.1. Tavola dei volumi

Concetto	Tipo	Volume
Rifugiato	E	300000
Ente	E	2500
Progetto	E	5000
Edizione progetto	E	20000
Progetto ordinario	E	4000
Progetto vulnerabilità	E	1000
Struttura alloggiativa	E	5000
Servizio	E	500
Corso	E	500
Assegnazione	R	5000
Occorrenza	R	20000
Offerta	R	100000
Proposta	R	100000
Utilizzo	R	20000
Partecipazione passata	R	50000
Partecipazione corrente	R	100000

Tabella 6. Tavola dei volumi

4.1.1.2. Tavola delle operazioni

Operazione	Tipo	Frequenza
Operazione 1: storico dei progetti con strutture	I	2000 al giorno
Operazione 2: storico dei progetti con servizi e corsi	I	2000 al giorno
Operazione 3: ricerca delle edizioni correnti di un progetto per una vulnerabilità con il numero dei partecipanti	I	30 al giorno
Operazione 4: statistica dei paesi di provenienza dei rifugiati	B	1 volta al mese
Operazione 5: statistica delle religioni dei rifugiati	B	1 volta al mese

Commentato [1]: 300.000 rifugiati di cui 150.000 partecipano in questo momento di cui partecipazione passata
Le 450.000 di partecipazione passata sono : 300.000 due partecipazioni dei 150.000 rifugiati che non stanno facendo progetti in questo momento e 150.000 le passate di quelli che stanno facendo progetti.
20 . 000 edizioni progetti di cui 15 . 000 part. passata di cui 5.000 part. corrente
Ed. progetto dura mediamente 6 mesi.
Rifugiato ha due partecipazioni in un anno
2 progetti per ente

Commentato [2R1]: Ogni edizione di progetto mette a disposizione 5 servizi e 5 corsi. Si può avere una sola edizione del progetto corrente. Quindi abbiamo 4 edizioni in totale: 3 edizioni passate e una corrente. 300.000 rifugiati perchè abbiamo supposto che sia una base di dati europea (1/3 -->100.000 partecipano correntemente ai progetti)
2500 enti
600.000 partecipazioni -> tutti hanno una partecipazione corrente (100.000)
600.000 part / 20.000 edizioni = 30 rifugiati a edizione

Commentato [3]: si considerano i dati nell'arco temporale di un anno

Commentato [4]: è importante che ci sia almeno una struttura per ogni edizione corrente ==> almeno 4000

Commentato [5]: 2 partecipazioni all'anno, più probabilmente passate

Commentato [6]: abbiamo notato che 1/5 delle edizioni progetto sono correnti, quindi anche 1/5 delle partecipazioni totali devono essere partecipazioni correnti

Commentato [7R6]: quindi:
20000 edizioni -> 4000 edizioni correnti
16000 edizioni passate

Commentato [8]: vanno inserite sia le operazioni del wp0 richiamandole sia le operazioni aggiuntive per l'analisi della ridondanza

Commentato [9]: contiene sia le partecipazione passate che corrente

Commentato [10]: per ciascun rifugiato 2/3 storici all'anno, quindi circa 2000 al giorno

Commentato [11]: in un anno 8 edizioni a un ente

Operazione 6: statistica del numero dei progetti assegnati ad ogni nazione	B	1 volta all'anno
Operazione 7: assegna un rifugiato a un'edizione di progetto	I	500 volte al giorno
Operazione 8: stampa le informazioni delle edizioni passate di un progetto con il numero dei partecipanti, le strutture, i servizi, i corsi e il progetto.	I	40 volte al giorno

Tabella 7. Tavola delle operazioni

Commentato [12]: per avere $100000 \cdot 2$ (edizioni) partecipazioni correnti in un anno, occorre fare $200000/365 = 500$ assegnazioni al giorno

Commentato [13]: 2500 enti in in anno * 8 edizioni in un anno /365

4.2. Analisi delle ridondanze

- Ridondanza 1: numero partecipanti (EDIZIONE PROGETTO).
Si tratta di un attributo ridondante perché è possibile ricavare tale valore contando le occorrenze di partecipazione.
RD1: Il numero di partecipanti di una certa edizione di progetto si ottiene contando il numero di istanze dell'entità rifugiato che sono legate a questa edizione.
TIPO: Attributo Derivabile da conteggio di occorrenze

4.2.1. Analisi della ridondanza 1: Numero Partecipanti

- Operazione 3:** ricerca delle edizioni correnti di un progetto per una vulnerabilità con il numero dei partecipanti

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
PROGETTO	E	1	L
PROGETTO VULNERABILITA'	E	1	L
OCCORRENZA	R	1	L
EDIZIONE PROGETTO	E	1	L

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
PROGETTO	E	1	L
PROGETTO VULNERABILITA'	E	1	L
OCCORRENZA	R	1	L
EDIZIONE PROGETTO	E	1	L
PARTECIPAZIONE CORRENTE	R	30	L

- Operazione 7:** assegna un rifugiato a un'edizione di progetto

Commentato [14]: fissato un progetto, delle sue 4 edizioni, 3 sono passate e 1 è corrente

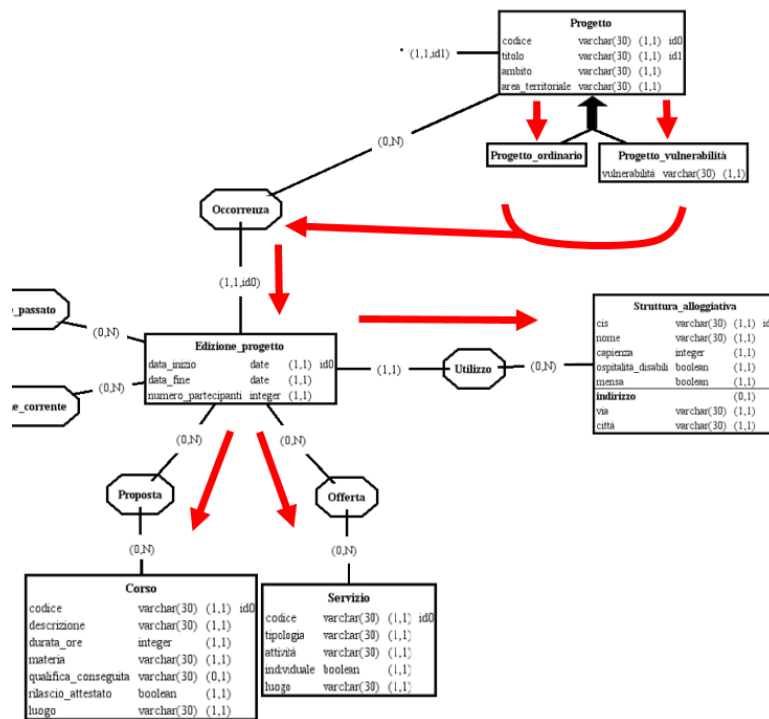
Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
RIFUGIATO	E	1	L
PARTECIPAZIONE CORRENTE	R	1	S
EDIZIONE PROGETTO	E	1	L
EDIZIONE PROGETTO	E	1	S

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
RIFUGIATO	E	1	L
PARTECIPAZIONE CORRENTE	R	1	S

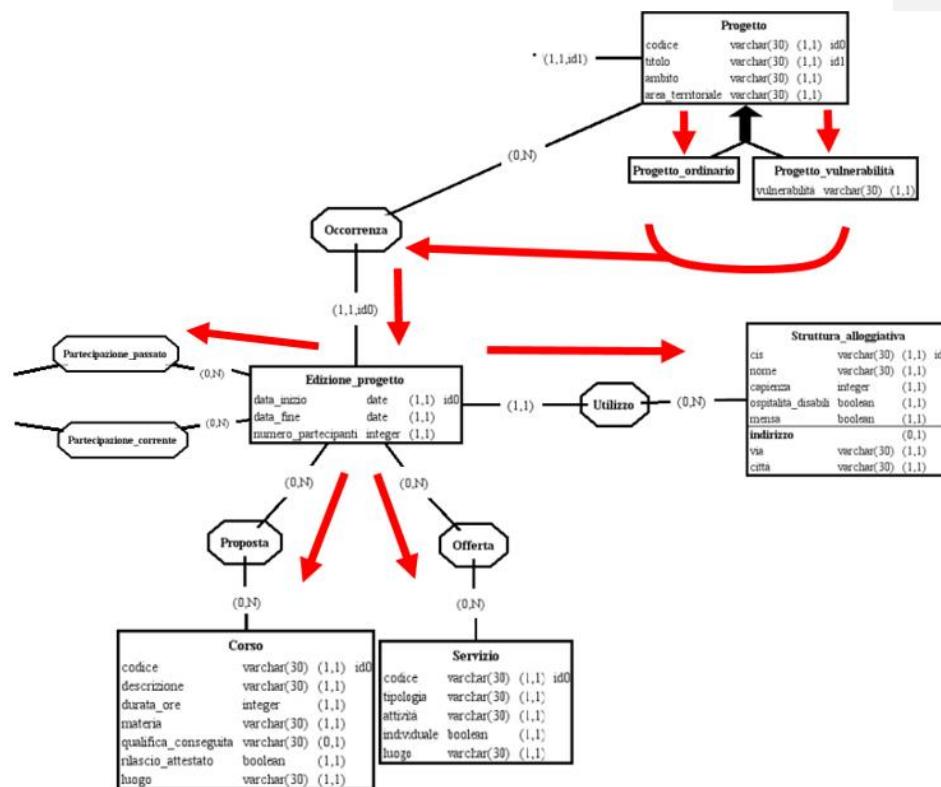
- **Operazione 8** : stampa le informazioni delle edizioni passate di un progetto con il numero dei partecipanti, le strutture, i servizi, i corsi e il progetto.

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
PROGETTO	E	1	L

PROGETTO ORDINARIO/ VULNERABILITA'	E	1	L
OCCORRENZA	R	3	L
EDIZIONE PROGETTO	E	3	L
UTILIZZO	R	3	L
STRUTTURE ALLOGGIATIVA	E	3	L
SERVIZIO	E	15	L
PROPOSTA	R	15	L
CORSO	E	15	L
OFFERTA	R	15	L

Senza Ridondanza



CONCETTO	COSTRUTTO	ACCESSI	TIPO
PROGETTO	E	1	L
PROGETTO ORDINARIO/ VULNERABILITA'	E	1	L

OCCORRENZA	R	3	L
EDIZIONE PROGETTO	E	3	L
UTILIZZO	R	3	L
STRUTTURE ALLOGGIATIVA	E	3	L
SERVIZIO	E	15	L
PROPOSTA	R	15	L
CORSO	E	15	L
OFFERTA	R	15	L
PARTECIPAZIONE PASSATA	R	90	L

4.2.1.1. Valutazione della ridondanza 1

Dopo aver analizzato le operazioni che coinvolgono la ridondanza e moltiplicando accessi per frequenza si osserva che, con il carico considerato:

Costo operazione 3:

- ❖ Con ridondanza
frequenza: 30

CONCETTO	COSTRUTTO	TIPO	ACCESSI
PROGETTO	E	L	30
PROGETTO VULNERABILITA'	E	L	30
OCCORRENZA	R	L	30
EDIZIONE PROGETTO	E	L	30
TOTALE			120

- ❖ Senza ridondanza
frequenza: 30

CONCETTO	COSTRUTTO	TIPO	ACCESSI
PROGETTO	E	L	30
PROGETTO VULNERABILITA'	E	L	30
OCCORRENZA	R	L	30
EDIZIONE PROGETTO	E	L	30
PARTECIPAZIONE CORRENTE	R	L	900
TOTALE			1020

Costo operazione 7:

- ❖ Con ridondanza
frequenza: 500

CONCETTO	COSTRUTTO	TIPO	ACCESSI
RIFUGIATO	E	L	500

PARTECIPAZIONE CORRENTE	R	S	1000
EDIZIONE PROGETTO	E	L	500
EDIZIONE PROGETTO	E	S	1000
TOTALE			3000

- ❖ Senza ridondanza
frequenza: 500

CONCETTO	COSTRUTTO	TIPO	ACCESSI
RIFUGIATO	E	L	500
PARTECIPAZIONE CORRENTE	R	S	1000
TOTALE			1500

Costo operazione 8:

- ❖ Con ridondanza
frequenza: 40

CONCETTO	COSTRUTTO	TIPO	ACCESSI
PROGETTO	E	L	40
PROGETTO ORDINARIO/ VULNERABILITA'	E	L	40
OCCORRENZA	R	L	120
EDIZIONE PROGETTO	E	L	120
UTILIZZO	R	L	120
STRUTTURE ALLOGGIATIVA	E	L	120
SERVIZIO	E	L	600
PROPOSTA	R	L	600
CORSO	E	L	600
OFFERTA	R	L	600
TOTALE			2960

- ❖ Senza ridondanza
frequenza: 40

CONCETTO	COSTRUTTO	TIPO	ACCESSI
PROGETTO	E	L	40
PROGETTO ORDINARIO/ VULNERABILITA'	E	L	40
OCCORRENZA	R	L	120
EDIZIONE PROGETTO	E	L	120
UTILIZZO	R	L	120
STRUTTURE ALLOGGIATIVA	E	L	120
SERVIZIO	E	L	600

PROPOSTA	R	L	600
CORSO	E	L	600
OFFERTA	R	L	600
PARTECIPAZIONE PASSATA	R	L	3600
TOTALE			6560

In conclusione, il costo totale con la ridondanza è:

Operazione	Costo totale
Operazione 3	120
Operazione 7	3000
Operazione 8	2960
TOTALE	6080

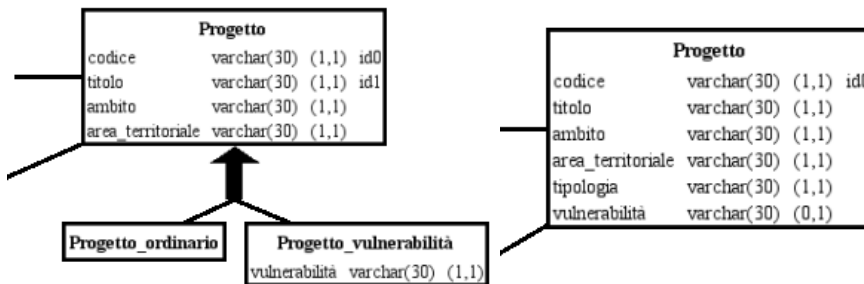
Senza ridondanza:

Operazione	Costo totale
Operazione 3	1020
Operazione 7	1500
Operazione 8	6560
TOTALE	9080

- In presenza di ridondanza il costo delle operazioni è di circa 6080 accessi giornalieri
 - L'occupazione di memoria è circa $4 \text{ byte} * 20.000 = 80.000 \text{ byte}$, \Rightarrow meno di 100 KB.
 - In assenza di ridondanza il costo delle operazioni è di 9080 accessi giornalieri.
- Pertanto, si decide di mantenere la ridondanza in quanto riduce il numero di accessi.

4.3. Eliminazione delle generalizzazioni

4.3.1. Generalizzazione *Progetto*



Per eliminare la generalizzazione di progetto si è deciso di utilizzare la strategia di accorpamento delle figlie nel genitore. In questo caso, occorre inserire in Progetto un attributo “tipologia” e l’attributo “vulnerabilità” dell’entità Progetto Vulnerabilità.

La motivazione principale della strategia utilizzata è data dalla somiglianza delle operazioni tra le occorrenze e tra gli attributi delle entità genitore e figlie.

Suddetta scelta comporta uno spreco di memoria per la presenza di valori nulli, ma assicura un numero minore di accessi rispetto alle altre strategie dove occorrenze e attributi sono distribuiti tra le varie entità.

4.4. Partizionamento/Accorpamento Entità e Associazioni

Si è ragionato sull’eventuale accorpamento di “partecipazione passata” e “partecipazione corrente” in un’unica associazione “partecipazione”, in maniera tale da evitare di trasferire occorrenze da un’associazione ad un’altra quando l’edizione di un progetto termina. Inoltre, le operazioni più frequenti del database, ovvero gli storici, non fanno distinzione tra le due associazioni. Pertanto, conviene effettuare questo accorpamento.

Per la compatibilità con il modello relazionale si è invece prevista l’eliminazione dell’attributo multivalore lingue. Esso diventa una nuova entità “Lingua” identificata dall’attributo “nome” ed è collegata tramite l’associazione molti a molti “conoscenza” all’entità originale Rifugiato, obbligatoria lato rifugiato in quanto si suppone che ogni rifugiato debba conoscere almeno una lingua.

4.5. Scelta degli identificatori principali

Durante la fase di progettazione concettuale si era deciso di identificare l’entità progetto tramite gli identificatori “codice” e “titolo” + “codice ente”. Nella fase di progettazione logica, considerando che sono da preferire identificatori interni e con il minor numero di attributi, la scelta è ricaduta sull’identificatore interno “codice”, che è anche il più comune nella realtà di interesse.

Per quanto riguarda l'edizione progetto, è identificata dall'attributo interno "data inizio" e da "codice" dell'entità progetto. Ciò comporta un identificatore poco efficiente il cui peso ricade su tutte le associazioni ad esclusione di assegnazione. Per questa ragione conviene introdurre un nuovo identificatore per l'entità edizione progetto, ovvero un codice costruito dal codice del progetto, un punto e un numero dell'edizione.

4.6. Schema ristrutturato finale

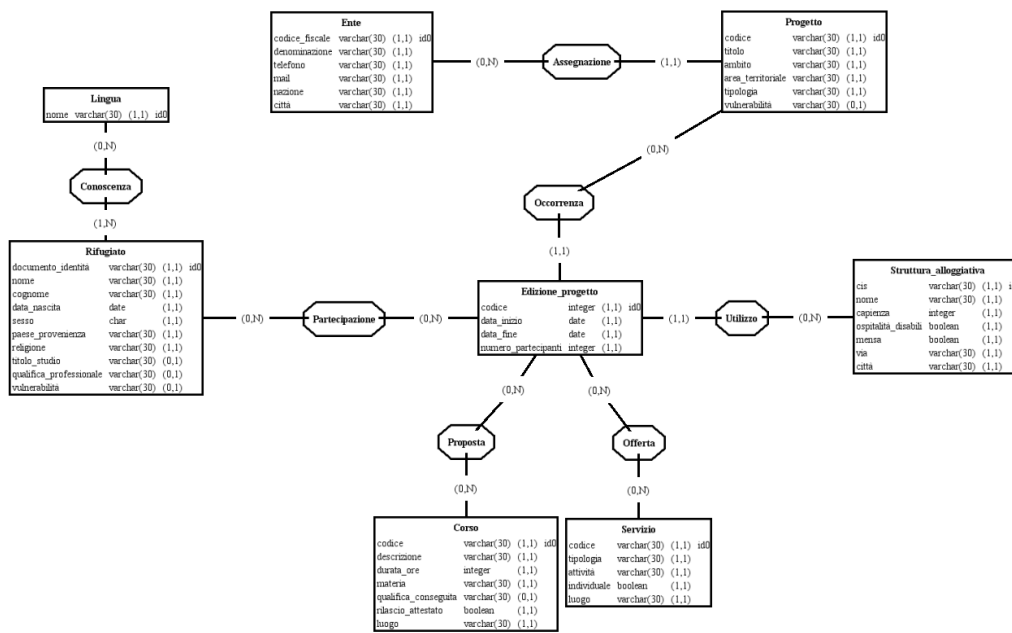


Figura 4. Schema E-R Ristrutturato

4.7. Schema logico

* → Attributo opzionale.

ENTE(codice fiscale, denominazione, telefono, mail, nazione, città)

RIFUGIATO(documento identità, nome, cognome, data nascita, sesso, paese provenienza, religione, titolo studio*, qualifica professionale*, vulnerabilità*)

LINGUA(nome)

CONOSCENZA(lingua, rifugiato)

PROGETTO(codice, titolo, ambito, area territoriale, tipologia, vulnerabilità*, ente assegnatario)

EDIZIONE PROGETTO(codice, progetto, data inizio, data fine, numero partecipanti, struttura)

STRUTTURA ALLOGGIATIVA(cis, nome, città, via, capienza, ospitalità disabili, mensa)

CORSO(codice, descrizione, durata ore, materia, qualifica conseguita*, rilascio attestato, luogo)

SERVIZIO(codice, tipologia, attività, individuale, luogo)

PARTECIPAZIONE(rifugiato, edizione progetto)

OFFERTA(servizio, edizione progetto)

PROPOSTA(corso, edizione progetto)

4.8. Documentazione dello schema logico

In fase di implementazione bisogna tener conto che il titolo del progetto con l'ente assegnatario è una chiave candidata, che deve essere trattata con unique in SQL. Lo stesso ragionamento vale per edizione progetto, in particolare per gli attributi data inizio e progetto.

Si deve aggiungere il vincolo non più esprimibile nello schema che un rifugiato può partecipare ad una sola edizione di progetto in corso.

Per quel che concerne le obbligarietà nelle associazioni uno a molti, esse sono risolvibili aggiungendo un vincolo not null, mentre per l'associazione conoscenza, essendo molti a molti, occorre implementare un trigger.

Vincoli di integrità referenziale:

1. Conoscenza (lingua) → Lingua (nome)
2. Conoscenza (rifugiato) → Rifugiato (documento identità)
3. Progetto (ente assegnatario) → Ente (codice fiscale)
4. Edizione Progetto (progetto) → Progetto (codice)
5. Edizione Progetto (struttura) → Struttura alloggiativa (cis)
6. Offerta (servizio) → Servizio (codice)
7. Offerta (edizione progetto) → Edizione progetto (codice)
8. Proposta (corso) → Corso (codice)
9. Proposta (edizione progetto) → Edizione progetto (codice)
10. Partecipazione (rifugiato) → Rifugiato (documento identità)
11. Partecipazione (edizione progetto) → Edizione progetto (codice)

5. Normalizzazione

Workpackage	Task	Responsabile
WP3	Normalizzazione	Peluso Raffaele

Innanzitutto lo schema logico è certamente in prima forma normale, perché è stato realizzato sulla base del modello relazionale.

Per la verifica della seconda forma normale si considerano le tabelle in cui la chiave è composta da più di un attributo, ossia conoscenza, proposta, offerta, partecipazione. Per le altre è già verificata. Tuttavia, per queste tabelle, non ci sono attributi non chiave. Di conseguenza lo schema è in seconda forma normale.

A questo punto si verifica la BCNF, analizzando tutte le dipendenze funzionali per ciascuna tabella. Per quel che concerne la tabella ente, si è riflettuto su un'eventuale dipendenza funzionale tra nazione e città, cioè città → nazione; tuttavia, esistono città omonime in nazioni differenti (e.g. Friburgo in Svizzera e Friburgo in Germania), dunque è possibile avere una coppia di tuple per cui il valore di città determina diversi valori di nazione. Di conseguenza non si tratta di una dipendenza funzionale.

Per la tabella struttura alloggiativa si è ragionato in maniera analoga sulla relazione tra via → città. Anche in questo caso non si tratta di una dipendenza funzionale, in quanto è comune avere vie uguali in città differenti.

Per la tabella progetto l'inevitabile legame tra la tipologia e l'attributo opzionale vulnerabilità è frutto della progettazione logica, in particolare dell'eliminazione della generalizzazione. Non è da considerarsi come dipendenza funzionale.

Per le altre tabelle non sono state trovate possibili dipendenze funzionali tra attributi non chiave da analizzare.

Quindi, dopo un'attenta analisi, si evince che per ciascuna tabella le uniche dipendenze funzionali trovate hanno per determinante la chiave primaria, che è certamente superchiave per ciascuna tabella. Questo risultato è frutto di una corretta progettazione concettuale, che ha permesso di tenere separati i concetti di interesse.

Di conseguenza si può affermare che lo schema logico è in BCNF.

6. Script Creazione e Popolamento Database

Workpackage	Task	Responsabile
WP2	SQL: Script creazione e popolamento	Roberto Gaetano

```
drop table if exists ente cascade;
drop table if exists rifugiato cascade;
drop table if exists lingua cascade;
drop table if exists conoscenza cascade;
drop table if exists struttura_alloggiativa cascade;
drop table if exists progetto cascade;
drop table if exists edizione_progetto cascade;
drop table if exists corso cascade;
drop table if exists servizio cascade;
drop table if exists partecipazione cascade;
drop table if exists offerta cascade;
drop table if exists proposta cascade;
drop domain if exists dominioVulnerabilita cascade;
```

```
create domain dominioVulnerabilita as varchar(30)
default 'Nessuna vulnerabilita'
check (value in ('Disabilita', 'Donna in gravidanza', 'Minore non accompagnato', 'Nessuna
vulnerabilita'));
```

```
create table ente(
    codice_fiscale varchar(11) primary key,
    denominazione varchar(60) not null,
    telefono varchar(13) not null,
    mail varchar(60) not null,
    nazione varchar(30) not null,
    citta varchar(30) not null
);
```

```
create table rifugiato(
    documento_identita varchar(9) primary key,
    nome varchar(100) not null,
    cognome varchar(100) not null,
    data_nascita date not null,
    sesso char(1) not null check(sesso = 'M' or sesso = 'F'),
    paese_provenienza varchar(30) not null,
    religione varchar(20) not null,
    titolo_studio varchar(100) default 'Nessun titolo',
    qualifica_professionale varchar(100) default 'Nessuna qualifica',
    vulnerabilita dominioVulnerabilita
);
```

```
create table lingua(
```



```
    nome varchar(30) primary key
);

create table conoscenza(
    lingua varchar(30) references lingua(nome) on update restrict on delete restrict,
    rifugiato varchar(10) references rifugiato(documento_identita) on update restrict on
delete restrict deferrable initially deferred,
    primary key(lingua,rifugiato)
);

create table struttura_alloggiativa(
    cis varchar(19) primary key,
    nome varchar(100) not null,
    via varchar(200) not null,
    citta varchar(60) not null,
    capienza integer not null,
    ospitalita_disabili boolean not null,
    mensa boolean not null
);

create table progetto(
    codice varchar(10) primary key,
    titolo varchar(50) not null,
    ambito varchar(50) not null,
    area_territoriale varchar(30) not null,
    tipologia varchar(30) not null check(tipologia = 'Ordinario' or tipologia = 'Vulnerabilita'),
    vulnerabilita dominioVulnerabilita,
    ente_assegnatario varchar(11) not null references ente(codice_fiscale) on update restrict
on delete restrict,
    unique(titolo,ente_assegnatario)
);

create table edizione_progetto(
    codice varchar(12) primary key,
    progetto varchar(10) not null references progetto(codice) on update restrict on delete
restrict,
    data_inizio date not null,
    data_fine date not null,
    numero_partecipanti integer not null,
    struttura varchar(19) not null references struttura_alloggiativa(cis) on update restrict on
delete restrict,
    unique(data_inizio,progetto)
);

create table corso(
    codice varchar(10) primary key,
```

```
    descrizione varchar(500) not null,
    durata_ore integer not null,
    materia varchar(60) not null,
    qualifica_conseguita varchar(40) default 'Nessuna qualifica',
    rilascio_attestato boolean not null,
    luogo varchar(60) not null
);

create table servizio(
    codice varchar(10) primary key,
    tipologia varchar(50) not null,
    attivita varchar(100) not null,
    individuale boolean not null,
    luogo varchar(100) not null
);

create table partecipazione(
    rifugiato varchar(10) references rifugiato(documento_identita) on update cascade on
delete restrict,
    edizione_progetto varchar(12) references edizione_progetto(codice) on update restrict on
delete restrict,
    primary key(rifugiato, edizione_progetto)
);

create table offerta(
    servizio varchar(10) references servizio(codice) on update restrict on delete restrict,
    edizione_progetto varchar(12) references edizione_progetto(codice) on update restrict on
delete restrict deferrable initially deferred,
    primary key(servizio, edizione_progetto)
);

create table proposta(
    corso varchar(10) references corso(codice) on update restrict on delete restrict,
    edizione_progetto varchar(12) references edizione_progetto(codice) on update restrict on
delete restrict deferrable initially deferred,
    primary key(corso, edizione_progetto)
);

INSERT INTO ente(codice_fiscale, denominazione, telefono, mail, nazione, citta) VALUES
('80000330656', 'Comune di Salerno', '089662134', 'comunedisalerno@gmail.com', 'Italia',
'Salerno'),
('80000330723', 'Associazione Arcobaleno', '+393248612339',
'associazionearcobaleno@gmail.com', 'Italia', 'Milano'),
('80000456789', 'Restos du coeur', '+333582367541', 'restosducoeur@gmail.com', 'Francia',
'Caen');
```

```
INSERT INTO lingua(nome) VALUES
```

```
('Italiano'),
('Francese'),
('Spagnolo'),
('Tedesco'),
('Arabo'),
('Ucraino'),
('Russo'),
('Inglese');
```

```
INSERT INTO conoscenza(lingua, rifugiato) VALUES
```

```
('Ucraino', 'IT3476889'),
('Inglese', 'IT3476889'),
('Russo', 'IT3476889'),
('Arabo', 'IT4512373'),
('Inglese', 'IT4512373'),
('Inglese', 'FR3256710'),
('Inglese', 'EN3265898');
```

```
INSERT INTO rifugiato(documento_identita, nome, cognome, data_nascita, sesso,
paese_provenienza, religione, titolo_studio, vulnerabilita) VALUES
('IT3476889', 'Roman', 'Boyko', '12/05/1997', 'M', 'Ucraina', 'Cattolicesimo', 'Diploma', 'Disabilita'),
('EN3265898', 'Elizabeth', 'Swan', '18/05/1988', 'F', 'Inghilterra', 'Anglicanesimo', 'Diploma', 'Donna
in gravidanza');
```

```
INSERT INTO rifugiato(documento_identita, nome, cognome, data_nascita, sesso,
paese_provenienza, religione, vulnerabilita) VALUES
('IT4512373', 'Aisha', 'Ahmed', '13/01/2012', 'F', 'Siria', 'Islam', 'Minore non accompagnato');
INSERT INTO rifugiato(documento_identita, nome, cognome, data_nascita, sesso,
paese_provenienza, religione) VALUES
('FR3256710', 'Alfatih', 'Ali', '14/08/1990', 'M', 'Sudan', 'Islam');
```

```
INSERT INTO struttura_alloggiativa(cis, nome, via, citta, capienza, ospitalita_disabili, mensa)
VALUES
('SA675421312S0000032', 'Residence Mare', 'Via Dante Alighieri 72', 'Salerno', 40, true, true),
('MI345412111S0000067', 'Hotel Marconi', 'Via Marconi 2', 'Milano', 37, true, true),
('NO658966512S0000021', 'CPH Denoyez', 'Rue Denoyez 21', 'Caen', 60, false, false);
```

```
INSERT INTO progetto(codice, titolo, ambito, area_territoriale, tipologia, vulnerabilita,
ente_assegnatario) VALUES
('EU0x123456', 'No limits', 'Sociale', 'Sud Italia', 'Vulnerabilita', 'Disabilita', '80000330656'),
('EU0x789568', 'Casa, mamma e bambino', 'Sociale', 'Nord Italia', 'Vulnerabilita', 'Donna in
gravidanza', '80000330723'),
('EU0x762192', 'Bright future', 'Istruzione', 'Nord Italia', 'Vulnerabilita', 'Minore non
accompagnato', '80000330723');
```

```
INSERT INTO progetto(codice, titolo, ambito, area_territoriale, tipologia, ente_assegnatario)
VALUES
('EU0x540012', 'Restart', 'Lavorativo', 'Nord Francia', 'Ordinario', '80000456789');
```

```
INSERT INTO corso(codice, descrizione, durata_ore, materia, rilascio_attestato, luogo) VALUES
('EUcx547876', 'Corso di creatività artistica', 72, 'Arte', false, 'Associazione Art, Salerno'),
('EUcx349006', 'Corso di narrativa in inglese', 48, 'Inglese', true, 'IC Giovanni Paolo II, Salerno'),
('EUcx399016', 'Corso di italiano base', 72, 'Italiano', true, 'Scuola Galileo, Milano'),
('EUcx419071', 'Corso di matematica base', 72, 'Matematica', true, 'Scuola Galileo, Milano'),
('EUcx111397', 'Corso di francese base', 72, 'Francese', true, 'Institut Lemmonier, Caen'),
('EUcx824163', 'Corso sui diritti del lavoratore', 48, 'Cittadinanza', false, 'Institut Lemmonier, Caen'),
('EUcx550061', 'Corso preparto', 54, 'Sociale', false, 'Ospedale San Raffaele, Milano');
```

```
INSERT INTO corso(codice, descrizione, durata_ore, materia, qualifica_conseguita,
rilascio_attestato, luogo) VALUES
('EUcx643210', 'Corso per operaio metalmeccanico', 200, 'Lavoro', 'Operaio base', true, 'Renault, Caen');
```

```
INSERT INTO servizio(codice, tipologia, attivita, individuale, luogo) VALUES
('EUsx547876', 'Sport', 'Pallacanestro in carrozzina', false, 'Asd Basket, Salerno'),
('EUsx349006', 'Assistenza', 'Seduta psicologica', true, 'Studio de Bellis, Angri'),
('EUsx399016', 'Sport', 'Varie attivita sportive', true, 'Centro sportivo, Pavia'),
('EUsx419071', 'Assistenza', 'Accoglienza in famiglia', false, 'Family first, Milano'),
('EUsx111397', 'Assistenza', 'Seduta psicologica', true, 'Scuola Galileo, Milano'),
('EUsx824163', 'Assistenza', 'Seduta psicologica', false, 'Anne Petite, Rouen'),
('EUsx550061', 'Assistenza', 'Inserimento lavorativo', false, 'Pole Employ, Caen'),
('EUsx643210', 'Assistenza', 'Assistenza legale per il lavoro', true, 'Cabinet pour le travail, Caen');
```

```
INSERT INTO proposta(corso, edizione_progetto) VALUES
('EUcx547876', 'EU0x123456.1'),
('EUcx349006', 'EU0x123456.1'),
('EUcx547876', 'EU0x123456.2'),
('EUcx399016', 'EU0x762192.1'),
('EUcx419071', 'EU0x762192.2'),
('EUcx111397', 'EU0x540012.1'),
('EUcx824163', 'EU0x540012.1'),
('EUcx550061', 'EU0x789568.1'),
('EUcx399016', 'EU0x789568.1'),
('EUcx643210', 'EU0x540012.2');
```

```
INSERT INTO offerta(servizio, edizione_progetto) VALUES
('EUsx547876', 'EU0x123456.2'),
('EUsx349006', 'EU0x123456.1'),
('EUsx349006', 'EU0x123456.2'),
('EUsx399016', 'EU0x762192.1');
```

```
('EU0x419071', 'EU0x762192.1'),
('EU0x399016', 'EU0x762192.2'),
('EU0x111397', 'EU0x762192.2'),
('EU0x111397', 'EU0x789568.1'),
('EU0x824163', 'EU0x540012.1'),
('EU0x824163', 'EU0x540012.2'),
('EU0x550061', 'EU0x540012.1'),
('EU0x550061', 'EU0x540012.2'),
('EU0x643210', 'EU0x540012.2');
```

```
INSERT INTO edizione_progetto(codice, progetto, data_inizio, data_fine, numero_partecipanti,
struttura) VALUES
```

```
('EU0x123456.1', 'EU0x123456', '12/01/2019', '12/03/2019', 1, 'SA675421312S0000032'),
('EU0x123456.2', 'EU0x123456', '20/04/2019', '20/06/2019', 1, 'SA675421312S0000032'),
('EU0x762192.1', 'EU0x762192', '15/09/2020', '31/12/2020', 1, 'MI345412111S0000067'),
('EU0x762192.2', 'EU0x762192', '18/01/2021', '18/05/2021', 1, 'MI345412111S0000067'),
('EU0x540012.1', 'EU0x540012', '01/01/2022', '20/03/2022', 1, 'NO658966512S0000021'),
('EU0x789568.1', 'EU0x789568', '25/03/2022', '28/06/2022', 1, 'MI345412111S0000067'),
('EU0x540012.2', 'EU0x540012', '10/04/2022', '10/06/2022', 1, 'NO658966512S0000021');
```

```
INSERT INTO partecipazione(rifugiato, edizione_progetto) VALUES
```

```
('IT3476889', 'EU0x123456.1'),
('EN3265898', 'EU0x789568.1'),
('IT3476889', 'EU0x123456.2'),
('IT4512373', 'EU0x762192.1'),
('IT4512373', 'EU0x762192.2'),
('FR3256710', 'EU0x540012.1'),
('FR3256710', 'EU0x540012.2');
```

Come evidenziato dalle note del modello ER, è stato implementato un dominio per la vulnerabilità e dei valori di default per gli attributi opzionali.

È stato aggiunto un check sul sesso del rifugiato e sulla tipologia di progetto a causa della generalizzazione totale. Le chiavi candidate sono state trattate con unique.

Per evitare perdita di informazioni si è scelto di utilizzare la policy on delete restrict.

Per quanto riguarda il documento d'identità del rifugiato si è ipotizzato che nel tempo possa cambiare, quindi si è scelto di utilizzare la policy on update cascade nella tabella partecipazione.

Per gli altri attributi interessati da policy si è ipotizzato che non possano cambiare nel tempo, quindi si è scelta la policy on update restrict.

La clausola deferrable initially deferred è stata aggiunta durante la progettazione dei trigger per gestire un controllo differito sugli inserimenti. Anche se non tutti i trigger scattano in fase di inizializzazione, l'ordine degli inserimenti è simulato già in fase di popolamento del db (conoscenza prima di rifugiato poiché per inserire un rifugiato occorre associargli una lingua, servizi e corsi prima di edizione progetto poiché ne deve offrire almeno uno tra i due).

7. Query SQL

Workpackage	Task	Responsabile
WP3	SQL: Query	Peluso Raffaele

7.1. Query con operatore di aggregazione e join: statistica enti

Questa query consente di effettuare una statistica sul numero delle edizioni di progetti nei vari Paesi europei selezionando un anno, ai fini di monitorare il livello di integrazione di ciascuna Nazione.

```
select ent.nazione, count(*) as progetti
from ente ent join progetto prog on ent.codice_fiscale = prog.ente_assegnatario
join edizione_progetto ed on ed.progetto = prog.codice
where extract (year from ed.data_fine) = 2022
group by ent.nazione
```

7.2. Query nidificata complessa: servizi più utili

Questa query consente di trovare i servizi offerti in tutte le edizioni di un progetto, per comprendere quali sono i più utili per i rifugiati.

```
select serv.attivita, serv.tipologia, serv.individuale, serv.luogo
from servizio serv
where exists(select * from edizione_progetto ed where ed.progetto='EU0x762192') and not exists(
  select ed.codice from edizione_progetto ed
  where ed.progetto = 'EU0x762192' and ed.codice not in(
    select off1.edizione_progetto from offerta off1
    where off1.servizio = serv.codice)
);
```

Si tratta di una query nidificata con interpretazione complessa, in quanto nella query interna si fa riferimento al codice del servizio della query esterna; quindi, per ogni riga della tabella servizio, si esegue la query interna.

L'idea alla base della query è di non restituire un servizio nel caso in cui si trovi un'edizione del progetto selezionato che non offra tale servizio. Quindi, prese le edizioni che offrono quel servizio, se un'edizione del progetto scelto non ricade in tale lista, allora questa edizione viene restituita alla not exists, che a sua volta restituisce false. Di conseguenza, il servizio non viene stampato.

La clausola exists è stata aggiunta in and per non restituire servizi nel caso in cui non esistano edizioni del progetto scelto. Infatti, senza tale clausola, il risultato della query interna sarebbe sempre vuoto, quindi la not exists andrebbe a restituire true, di conseguenza sarebbero stampati tutti i servizi.

7.3. Query insiemistica: percorso del rifugiato

Questa query consente di stampare i corsi e i servizi usufruiti da un rifugiato in un certo periodo, per permettere alla famiglia o ai conoscenti di tenere traccia del suo percorso di integrazione.

```

select serv1.attivita as attivita, serv1.luogo, ed1.data_inizio, ed1.data_fine
from rifugiato rif1 join partecipazione part1 on rif1.documento_identita = part1.rifugiato
join edizione_progetto ed1 on part1.edizione_progetto = ed1.codice
join offerta off1 on off1.edizione_progetto = ed1.codice
join servizio serv1 on serv1.codice = off1.servizio
where rif1.documento_identita = 'IT3476889'
union
select cor2.descrizione as attivita, cor2.luogo, ed2.data_inizio, ed2.data_fine
from rifugiato rif2 join partecipazione part2 on rif2.documento_identita = part2.rifugiato
join edizione_progetto ed2 on part2.edizione_progetto = ed2.codice
join proposta prop2 on prop2.edizione_progetto = ed2.codice
join corso cor2 on cor2.codice = prop2.corso
where rif2.documento_identita = 'IT3476889'
order by data_inizio;

```

Per poter utilizzare la union è stato necessario selezionare solo alcuni attributi, i cui tipi sono compatibili tra le tabelle servizio e corso.

8. Viste

Workpackage	Task	Responsabile
WP4	Viste	Noschese Valerio

8.1. Vista *proposta_enti*

Individua il numero di corsi che ogni ente offre a persone affette da una vulnerabilità

```

create view proposta_enti(ente,num_corsi,vulnerabilita) as
select denominazione,count(distinct corso),prog.vulnerabilita
from ente join progetto prog on codice_fiscale=ente_assegnatario
join edizione_progetto ed on prog.codice=ed.progetto
join proposta prop on ed.codice=prop.edizione_progetto
where prog.tipologia<>'Ordinario'
group by codice_fiscale, prog.vulnerabilita ;

```

8.1.1. Query con Vista: enti più attivi per ogni vulnerabilità

Mostra l'ente che ha proposto più corsi per una specifica vulnerabilità.

```

select vulnerabilita,ente,num_corsi
from proposta_enti tab
group by vulnerabilita,ente,num_corsi
having num_corsi >= all(select num_corsi from proposta_enti where
vulnerabilita=tab.vulnerabilita)

```

9. Trigger

9.1. Trigger inizializzazione: *check_lingua*

Il trigger scatta in fase di popolamento del database allo scopo di controllare che, al momento dell'inserimento di un rifugiato, si inserisca obbligatoriamente una lingua conosciuta dal rifugiato.

```
drop trigger if exists check_lingua on rifugiato;
drop function if exists lingua_rifugiato();

create function lingua_rifugiato() returns trigger as $BODY$
begin
    if(not exists(select * from conoscenza where rifugiato = new.documento_identita)) then
        raise exception 'Inserire almeno una lingua conosciuta da %', new.documento_identita;
    end if;
    return new;
end
$BODY$ language plpgsql;

create trigger check_lingua
after insert on rifugiato
for each row execute procedure lingua_rifugiato();
```

9.2. Trigger inizializzazione: *check_partecipazione_corrente*

Il trigger scatta in fase di popolamento del database allo scopo di controllare che, al momento dell'inserimento, si abbia una singola partecipazione corrente per un rifugiato.

```
drop trigger if exists check_partecipazione_corrente on partecipazione;
drop function if exists check_partecipazione_corrente();

create function check_partecipazione_corrente() returns trigger as $BODY$
begin
    if((select data_fine from edizione_progetto where codice=new.edizione_progetto)-
        current_date>0) then
        if(exists(select * from partecipazione join edizione_progetto on
            edizione_progetto=codice where rifugiato=new.rifugiato and
            codice<>new.edizione_progetto and (data_fine-current_date>0))) then
            raise exception 'rifugiato % già assegnato a un progetto', new.rifugiato;
        end if;
    end if;
    return new;
end $BODY$ language plpgsql;

create trigger check_partecipazione_corrente
after insert on partecipazione
for each row execute procedure check_partecipazione_corrente();
```


9.3. Trigger per vincoli aziendali

Workpackage	Task	Responsabile
WP4	Trigger per vincoli aziendali	Noschese Valerio

9.3.1. Trigger1: check_edizione_progetto

Il trigger implementa il vincolo secondo il quale un'edizione di un progetto deve offrire almeno un servizio o proporre almeno un corso

```
drop trigger if exists check_edizione on edizione_progetto;
drop function if exists check_edizione();
```

```
create function check_edizione() returns trigger as $BODY$
```

```
begin
```

```
if(not exists(select * from offerta where edizione_progetto=new.codice) and
```

```
not exists(select* from proposta where edizione_progetto=new.codice)) then
```

```
raise exception 'edizione progetto deve offrire almeno un corso o un servizio ';
```

```
end if;
```

```
return new;
```

```
end $BODY$ language plpgsql;
```

```
create trigger check_edizione
```

```
after insert on edizione_progetto
```

```
for each row execute procedure check_edizione();
```

9.3.2. Trigger2: check_vulnerabilita

Il trigger implementa il vincolo secondo il quale un'edizione di un progetto di vulnerabilità per disabili deve disporre di una struttura attrezzata per tale vulnerabilità

```
drop trigger if exists check_vulnerabilita on edizione_progetto;
```

```
drop function if exists check_vulnerabilita;
```

```
create function check_vulnerabilita() returns trigger as $BODY$
```

```
begin
```

```
if(exists(select * from progetto where codice=new.progetto and tipologia='Vulnerabilita'
and vulnerabilita='Disabilita')) then
```

```
if(select ospitalita_disabili from struttura_alloggiativa where new.struttura=cis) then
return new;
```

```
else
```

```
raise exception 'struttura non in grado di ospitare disabili';
```

```
end if;
```

```
end if;
```

```
return new;
```

```
end $BODY$ language plpgsql;
```

```
create trigger check_vulnerabilita
```

```
after insert on edizione_progetto
```

```
for each row execute procedure check_vulnerabilita();
```

9.3.3. Trigger3: aggiorna_partecipanti

Il trigger aggiorna la ridondanza presente in edizione_progetto, ovvero il numero di partecipanti di un'edizione. Trattandosi di un trigger per una regola di derivazione, è stato inserito nella sezione dei vincoli aziendali. Tuttavia, poiché il database deve essere già popolato, si potrebbe anche pensare di far scattare il trigger in fase di inizializzazione, invece di impostare manualmente il numero iniziale in conformità ai rifugiati inseriti in fase di popolamento.

```
drop trigger if exists aggiorna_partecipanti on partecipazione;
drop function if exists aggiorna_partecipanti();

create function aggiorna_partecipanti() returns trigger as $BODY$
declare
    c integer;
begin

select count(*) into c from partecipazione
    where edizione_progetto = new.edizione_progetto;

update edizione_progetto
set numero_partecipanti = c
where codice = new.edizione_progetto;

return new;
end $BODY$ language plpgsql;

create trigger aggiorna_partecipanti
after insert on partecipazione
for each row execute procedure aggiorna_partecipanti();
```

9.3.4. Trigger4: verifica_capacita

Il trigger implementa il vincolo secondo il quale il numero dei partecipanti di un'edizione di un progetto deve essere minore o uguale della capienza della struttura alloggiativa utilizzata. Il numero partecipanti in edizione_progetto è certamente aggiornato a valle del nuovo inserimento in partecipazione, in quanto il relativo trigger è stato inserito prima del trigger verifica_capacita nel database, quindi è eseguito prima.

```
drop trigger if exists verifica_capacita on partecipazione;  
drop function if exists verifica_capacita();
```

```
create function verifica_capacita() returns trigger as $BODY$
```

```
declare
```

```
    part integer;
```

```
    cap integer;
```

```
begin
```

```
select numero_partecipanti into part from edizione_progetto  
where codice = new.edizione_progetto;
```

```
select capienza into cap from  
    struttura_alloggiativa join edizione_progetto on cis = struttura  
where codice = new.edizione_progetto;
```

```
if(part > cap) then raise exception 'struttura alloggiativa piena';  
end if;
```

```
return new;
```

```
end $BODY$ language plpgsql;
```

```
create trigger verifica_capacita
```

```
after insert on partecipazione
```

```
for each row execute procedure verifica_capacita();
```