

Università degli Studi di Padova
Dipartimento di Matematica “Tullio Levi - Civita”

Progetto Basi di Dati

Base di dati per un Grest (centro estivo)

Studenti:

Artusi Emanuele (██████████)

Bolzon Nicolò (██████████)

Anno Accademico: 2023/2024

1. ABSTRACT

Durante il periodo estivo è tipico per la Parrocchia di [REDACTED] in collaborazione con l'amministrazione comunale, organizzare il Grest: esperienza di animazione del tempo libero dove ragazzi e ragazze di diverse età imparano, giocano, si divertono e crescono sullo stile degli oratori cattolici, accompagnati e seguiti da animatori. I partecipanti vengono iscritti dall'adulto genitorialmente responsabile per tutte o alcune delle settimane in cui si svolge l'evento e dovrà fornire un proprio contatto da utilizzare in caso di necessità. Inoltre, verranno subito assegnati ad una delle squadre di gioco dell'edizione corrente. Anche gli animatori possono essere organizzati in équipe in base ad alcune mansioni speciali che dovranno svolgere nell'evento; ad altri, invece, è chiesto il compito di animare una squadra di ragazzi. Con il Grest si coinvolgono tutti, dagli animati agli animatori, facenti parte di una storia che si costruisce assieme giorno per giorno, con la collaborazione di adulti e genitori. I ragazzi svolgono nella prima parte della giornata alcuni laboratori, in base alle proprie preferenze, in cui si possono esprimere con attività più manuali per costruire oggetti sbizzarrendo la propria creatività e fantasia, guidati dagli animatori. Tipicamente ogni animato può svolgere un solo laboratorio per settimana tra i vari offerti. Nella seconda parte della giornata si mette da parte la propria creatività per dar spazio al gioco in cui le squadre si sfidano in molteplici partite volte alla collaborazione tra i ragazzi, al rispetto delle regole e dare il massimo di sé, sapendo accettare anche le sconfitte. Al termine di ogni settimana si comunica ai ragazzi la classifica parziale formata con i punteggi accumulati con le varie sfide vinte, per premiare alla fine del centro estivo la prima classificata. Ogni settimana, se il meteo è favorevole, è comune organizzare un'uscita in qualche località preferendo parchi acquatici o parchi divertimento. Durante l'uscita, a cui si partecipa su base volontaria, le altre attività ordinarie sono sospese. Infine, è sempre compito degli animatori effettuare gli ordini dei gadget mancanti necessari allo svolgimento di ogni attività.

2. ANALISI DEI REQUISITI

2.1. Descrizione

Si vuole creare un database per gestire un Grest (o centro estivo) versatile e utilizzabile per ogni edizione e, pertanto, ognuna di esse viene descritta con:

- Un numero progressivo, univoco per ogni edizione
- Un titolo (*solitamente l'edizione ha uno slogan*)
- Una descrizione (*a discrezione degli organizzatori, indica eventuali informazioni aggiuntive, novità o altro*)
- La data di inizio
- La data di fine

Inoltre, ogni evento ha una canzone principale ed è diversa per ogni edizione, simile ad una “colonna sonora”. Questa traccia audio è descritta da un nome, univoco per ogni canzone.

In questo database si devono poter gestire anche tutti gli utenti (ragazzi e animatori) che partecipano a queste esperienze estive affinché ne risulti più semplice la gestione anagrafica.

In particolare, ogni ragazzo partecipante (animato) è descritto da:

- Un Codice Fiscale (*univoco per ogni persona*)
- Il cognome e il nome
- L'indirizzo di residenza
- La data di nascita

Essendo i partecipanti tutti minorenni, di età compresa tra la prima elementare e la terza media, devono necessariamente essere iscritti da un genitore o tutore che può avere uno o più figli partecipanti all'esperienza.

Ogni adulto è descritto da:

- Un Codice Fiscale (*univoco per ogni persona*)
- Il cognome e il nome
- L'indirizzo di residenza
- La data di nascita
- Un numero di cellulare

La fase di iscrizione è solamente a fini anagrafici. Il numero di settimane frequentanti a quella determinata edizione del centro estivo la si indica nel momento in cui il ragazzo viene associato ad una squadra, la quale rimane la stessa per tutta la durata dell'evento.

A tal motivo:

- Se il ragazzo partecipa alla prima e la terza settimana, nel rispettivo attributo si scriverà "13"
- Se il ragazzo partecipa alla prima, seconda e terza settimana, nell'attributo si scriverà "123"
- Se il ragazzo partecipa alla prima e quarta settimana, nel rispettivo attributo si scriverà "14"

e così via.

Se in un'edizione il genitore/tutore ad iscrivere lo stesso partecipante è diverso rispetto alle edizioni precedenti si aggiorneranno i dati dell'adulto dal momento che mantenere lo storico di chi è stato genitore/tutore dei partecipanti non rientra negli obiettivi della base dati in oggetto.

All'esperienza estiva partecipano, in qualità di animatori, anche giovani studenti delle superiori e dell'età degli universitari. Ogni animatore è descritto da:

- Un Codice Fiscale (*univoco per ogni persona*)
- Il cognome e il nome
- L'indirizzo di residenza
- La data di nascita
- Un numero di cellulare
- Un indirizzo Email (*necessario per condividere materiale digitale*)
- Un campo Richiesta Crediti (*per indicare se necessita del certificato per i crediti scolastici*)

Qualche animatore potrebbe afferire a una o più équipes, le quali sono descritte da:

- Un nome (*univoco per l'evento*)
- Una descrizione

Tramite la relazione, si tenga in considerazione che un animatore facente parte dell'équipe potrebbe esserne anche il responsabile.

Ogni animatore può, ma non è obbligatorio, essere associato ad una sola squadra e può collaborare ad un solo laboratorio; entrambe le associazioni rimangono invariate per tutta la durata dell'esperienza estiva. In ogni caso, un animatore deve essere associato almeno ad uno tra squadra, laboratorio o équipe nell'evento. Ogni animato, invece, è associato ad una sola squadra nella quale rimane per tutta la durata del centro estivo e ogni settimana partecipa ad un solo laboratorio che non può essere lo stesso della settimana precedente.

Il laboratorio, così come le squadre e le équipes, sono associati a quella particolare edizione del centro estivo; in questo modo laboratori, squadre ed équipes rimangono univoci e differenti per ogni edizione.

Un laboratorio viene descritto da:

- Un nome (*univoco per l'evento*)
- Una descrizione (*indica cosa si fa, le abilità richieste, l'oggetto da produrre etc.*)

Ogni squadra deve essere composta da animatori e da animati. I membri delle squadre, a differenza dei laboratori nei quali il ragazzo cambia attività ogni settimana, rimangono associati alla stessa per tutta la durata dell'evento. Ogni squadra è descritta da:

- Un nome (*univoco per l'evento*)
- Un punteggio (*accumulato con le varie vittorie*)

Nel centro estivo le squadre competono in diverse partite "squadra contro squadra" per cercare di vincere l'edizione del centro estivo. Al termine dell'edizione si premiano le squadre arrivate nei primi gradini del podio.

Ogni partita è descritta da:

- Data e Ora (*univoci assieme alle due squadre sfidanti*)

Tramite le relazioni tra entità si dovranno indicare anche le due squadre sfidanti (rispettivamente Squadra A e Squadra B) e quale squadra ha vinto la partita.

Ogni partita, inoltre, è arbitrata da un animatore che deve far parte dell'Équipe Arbitri o dell'Équipe Giochi.

Si tenga in considerazione, inoltre, che in un campo si possono svolgere anche più partite poiché quest'ultimo può essere diviso in più terreni attrezzati in base al gioco

Ogni partita ha associato un solo gioco facente parte di un'entità "Gioco" che contiene al suo interno tutti i giochi previsti e preparati. Pertanto, ogni gioco è descritto da:

- Nome (*univoco*)
- Regole
- Punteggio (*quanti punti ottiene la squadra vincitrice*)

Se necessario, ogni gioco prevede uno spareggio. Ogni partita avrà quindi una squadra vincitrice che otterrà interamente il punteggio relativo allo specifico gioco e una squadra perdente che non otterrà alcun punto.

Ogni gioco è associato ad un singolo campo, però più giochi possono avvenire nello stesso campo. Questo significa che, per esempio, nel campo 2 si giocherà a calcio e a fazzoletto e possiamo dire con certezza che in nessun altro campo si giocherà a calcio o a fazzoletto.

Pertanto, ogni singolo campo è descritto da:

- Numero (*univoco*)
- La tipologia di campo (*erboso, sabbioso, etc.*)
- La zona (*definisce velocemente la posizione indicativa del campo all'interno dell'area dell'evento, p.e. "dietro al campanile" o "di fianco al teatro"*)

Una volta a settimana viene proposta un'uscita giornaliera, con rientro lo stesso giorno della partenza, in diverse località preferendo parchi acquatici o parchi divertimento.

Ogni uscita è descritta da:

- Un nome (*univoco assieme alla data*)
- La data (*univoca assieme al nome*)
- Ora di inizio
- Ora di fine
- Il luogo di destinazione
- Una descrizione (*a discrezione della Segreteria, informazioni utili alla corretta organizzazione dell'uscita*)

Infine, si tenga in considerazione la gestione degli ordini di alcuni gadget, solitamente magliette, casacche, palloni, braccialetti in silicone, etc. personalizzati con il logo e simboli del Grest.

Un animatore può effettuare uno o più ordini. Ogni ordine può avere più pezzi solo dello stesso tipo di gadget.

Ogni gadget è descritto da:

- Un nome (*univoco assieme a colore e taglia*)
- Un colore (*univoco assieme a nome e taglia*)
- Una taglia (*univoca assieme a nome e colore*)
- Un prezzo (*cadauno*)

Si salveranno inoltre la data in cui si effettua l'ordine, la quantità di pezzi e il costo totale, dato dalla somma del prezzo di ciascun gadget.

2.2. Glossario

Termine	Descrizione
GrEst	Abbreviazione di Gruppi Estivi, sono esperienze di animazione estiva proposte dalle Parrocchie o altre scuole e/o associazioni di ispirazione cattolica e patrocinati dalle amministrazioni comunali in cui sussistono i vari enti organizzatori.
Équipe	Gruppi di animatori che svolgono alcune attività di animazione aggiuntive rispetto all'ordinario (ad esempio: Équipe Bans si occupa di animare i momenti di danza e ballo, Équipe Giochi si occupa di organizzare i giochi per le diverse giornate e arbitra le partite, ...).
Animato	Ragazzo o ragazza, tipicamente dell'età dalla prima elementare alla terza media, che partecipa al centro estivo.
Animatore	Giovane, tipicamente di età delle scuole secondarie superiori e universitari, che organizza il centro estivo ed è responsabile degli animati.
Ottenere (punti)	Aggiungere i punti definiti dal gioco al punteggio totale della squadra valido per la classifica finale.
Campo (di gioco)	Area adibita allo svolgimento delle partite, può essere diviso in più terreni attrezzati in base al gioco.

3. PROGETTAZIONE CONCETTUALE

3.1. Lista delle Entità

Il database è formato dalle entità sotto riportate e i rispettivi identificatori sono sottolineati e in grassetto.

Tutti i campi sono NOT NULL fatta eccezione ove specificato diversamente.

- **Evento:** rappresenta ogni edizione del Centro Estivo
 - **Edizione:** *int* (progressivo)
 - Titolo: *string*
 - Descrizione: *string*
 - DataInizio: *date*
 - DataFine: *date*
- **Canzone:** rappresenta la traccia (colonna sonora) dell'evento
 - **Nome:** *string*
- **Squadra:** rappresenta l'accorpamento dei ragazzi e alcuni animatori per i giochi
 - **Nome:** *string* (composta con Edizione)
 - Punteggio: *int*
- **Équipe:** rappresenta l'accorpamento di alcuni animatori con ruoli specifici
 - **Nome:** *string* (composta con Edizione)
 - Descrizione: *string*
- **Laboratorio:** rappresenta le attività pratiche che i ragazzi svolgono durante le settimane. Ogni settimana ne svolgono uno diverso. Ai laboratori si associano alcuni animatori incaricati per l'aiuto degli animati iscritti.
 - **Nome:** *string* (composta con Edizione)
 - Descrizione: *string*
- **Persona:** Rappresenta una Persona nel Database (Animatore, Animato, Genitore)
 - **CodiceFiscale:** *string* (lunghezza 16 caratteri)
 - Nome: *string*
 - Cognome: *string*
 - DataNascita: *date*
 - Cellulare: *string*
 - Indirizzo: *attributo composto da: Via, Civico, Comune, CAP, Provincia*
- **Animatore:** rappresenta un animatore
 - Email: *string*
 - NecessitaCrediti: *boolean*
- **Adulto:** rappresenta un adulto che iscrive uno o più ragazzi
- **Partecipante:** rappresenta il partecipante iscritto all'esperienza estiva
- **Uscita:** rappresenta le uscite settimanali
 - **Nome:** *string* (composta con Data)
 - **Data:** *date* (composta con Nome)
 - OraInizio: *time*
 - OraFine: *time*
 - Luogo: *string*
 - Descrizione: *string*
- **Gadget:** rappresenta gli oggetti ordinati dagli animatori
 - **Nome:** *string* (composta con Colore e Taglia)
 - **Colore:** *string* (composta con Nome e Taglia)
 - **Taglia:** *string* (composta con Nome e Colore)
 - Prezzo: *decimal*
- **Partita:** rappresenta le partite giocate dalle squadre di partecipanti
 - **Data:** *date* (composta con Ora, SquadraA, EdizioneSquadraA, SquadraB ed EdizioneSquadraB)
 - **Ora:** *time* (composta con Data, SquadraA, EdizioneSquadraA, SquadraB ed EdizioneSquadraB)
- **Gioco:** rappresenta il tipo di gioco
 - **Nome:** *string*
 - Regole: *string*
 - Punteggio: *int*
- **Campo:** rappresenta il campo in cui si giocheranno le partite
 - **Numero:** *int*
 - Tipologia: *enum*
 - Zona: *string*

3.2. Lista delle Relazioni

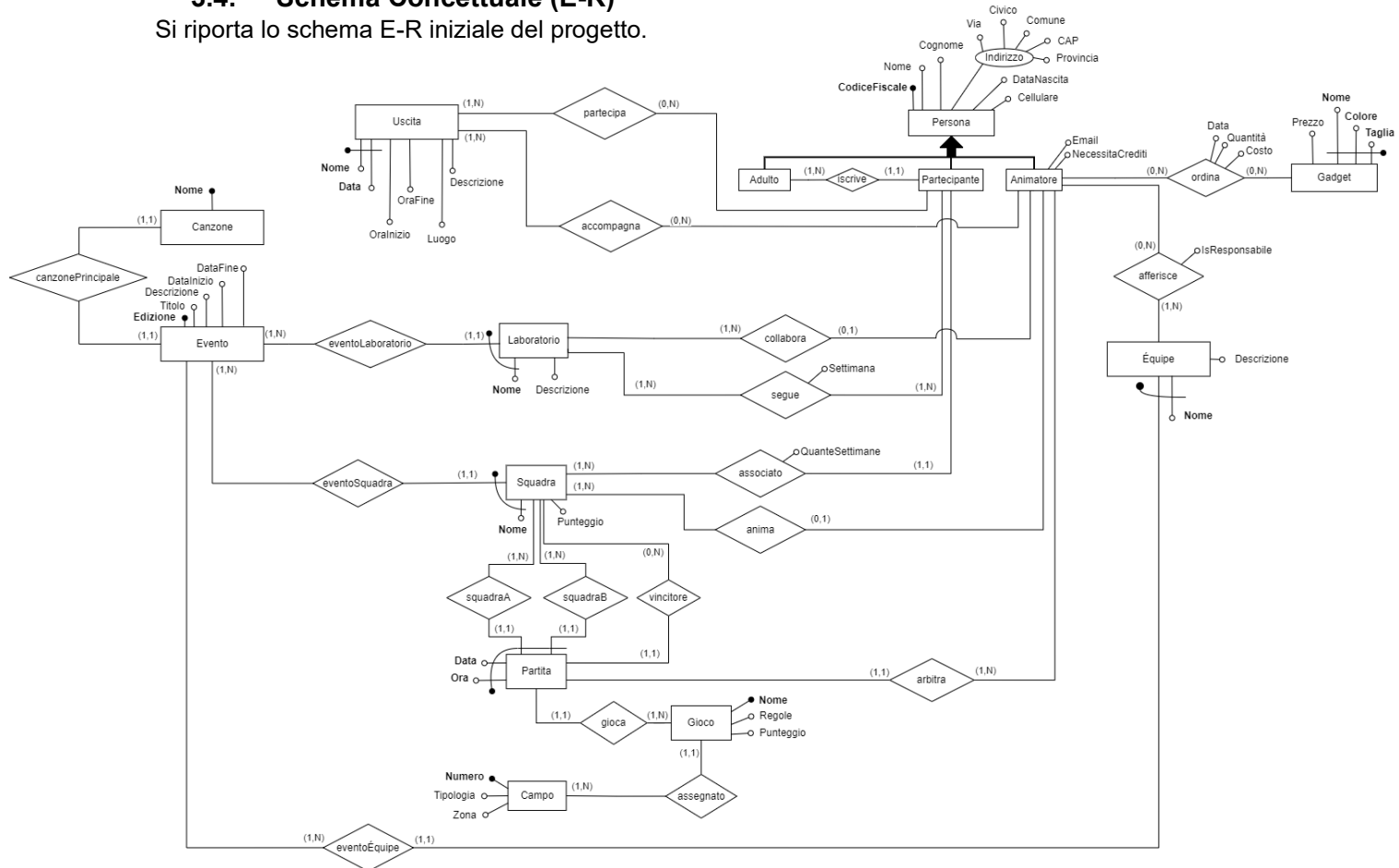
Relazione	Entità Coinvolte	Descrizione	Attributi
canzonePrincipale (1:1)	Evento (1,1) Canzone (1,1)	Ogni evento ha la sua canzone principale ed ogni canzone è tipica di un solo evento	/
eventoSquadra (1:N)	Evento (1,N) Squadra (1,1)	Ogni squadra è associata ad una specifica edizione dell'evento e ad ogni evento appartengono più squadre	/
eventoLab (1:N)	Evento (1,N) Laboratorio (1,1)	Ogni laboratorio è associato ad una specifica edizione dell'evento e ad ogni evento ci sono uno o più laboratori.	/
eventoÉquipe (1:N)	Evento (1,N) Équipe (1,1)	Ogni équipe è associata ad una specifica edizione dell'evento e ad ogni evento ci sono una o più équipes	/
collabora (1:N)	Animatore (0,1) Laboratorio (1,N)	Un animatore può collaborare a zero o al massimo ad un solo laboratorio e ad ogni laboratorio collaborano uno o più animatori	/
segue (N:M)	Partecipante (1,N) Laboratorio (1,N)	Ogni partecipante partecipa a più laboratori, diverso per ogni settimana di frequenza e ad ogni laboratorio partecipano più ragazzi (animati)	Settimana
associato (1:N)	Partecipante (1,1) Squadra (1,N)	Ogni partecipante è associato ad una sola squadra e ad ogni squadra sono associati più animati	QuanteSettimane
anima (1:N)	Animatore (0,1) Squadra (1,N)	Un animatore può animare zero o al massimo una squadra e in ogni squadra ci sono uno o più animatori che ne fanno parte e la animano	/
squadraA (1:N)	Partita (1,1) Squadra (1,N)	Ogni partita ha una squadra A e la squadra può essere classificata come squadra A in diverse partite	/
squadraB (1:N)	Partita (1,1) Squadra (1,N)	Ogni partita ha una squadra B e la squadra può essere classificata come squadra B in diverse partite	/
vincitore (1:N)	Partita (1,1) Squadra (0,N)	Ogni partita ha una squadra vincitrice (squadra A oppure squadra B) e la squadra può vincere zero o più partite	/
arbitra (1:N)	Partita (1,1) Animatore (1,N)	Ogni partita viene arbitrata da un animatore, <i>facente parte dell'équipe Giochi</i> , il quale può arbitrare più partite	/
gioca (1:N)	Partita (1,1) Gioco (1,N)	Ad ogni partita si gioca ad un solo gioco e quel gioco può essere giocato in una o più partite	/
assegnato (1:N)	Gioco (1,1) Campo (1,N)	Ogni gioco è assegnato ad uno specifico campo e in un campo si possono giocare più giochi	/
accompagna (N:M)	Uscita (1,N) Animatore (0,N)	Ad un'uscita possono partecipare più animatori e ogni animatore può partecipare a più uscite	/
partecipa (N:M)	Uscita (1,N) Partecipante (0,N)	Ad un'uscita possono partecipare più partecipanti (animati) e ogni animato può partecipare a più uscite	/
iscrive (1:N)	Adulto (1,N) Partecipante (1,1)	Ogni adulto può iscrivere più figli e ogni animato deve essere iscritti da un adulto genitore o tutore legale	/
afferisce (N:M)	Animatore (0,N) Équipe (1,N)	Un animatore può afferire a zero o più équipes e ad ogni équipe afferiscono più animatori	IsResponsabile
ordina (N:M)	Gadget (0,N) Animatore (0,N)	Un animatore può effettuare zero o più ordini di gadget e ogni gadget può essere ordinato da zero o più animatori.	Data Quantità Costo

3.3. Lista Generalizzazioni

- **Persona** è una generalizzazione totale ed esclusiva di **Adulto**, **Animatore** e **Partecipante**.

3.4. Schema Concettuale (E-R)

Si riporta lo schema E-R iniziale del progetto.



4. PROGETTAZIONE LOGICA

4.1. Analisi delle ridondanze

Analizzando lo schema E-R si può notare che l'attributo **Costo** della relazione "ordina" potrebbe essere calcolato dalla moltiplicazione del **Prezzo** dell'entità "Gadget" per la **Quantità** (relazione "ordina") di articoli ordinati di quel determinato gadget.

Effettuiamo un'analisi sulle operazioni per determinare se può essere eliminato o meno.

- **Operazione 1 (100 alla settimana):** effettuazione di un ordine per l'acquisto di gadget
- **Operazione 2 (2 alla settimana):** rivedere i costi totali sostenuti

Concetto	Costrutto	Volume
Gadget	Entità	100
Ordina	Relazione	10'000

Con ridondanza

- **Operazione 1**

È necessario innanzitutto leggere il prezzo del singolo gadget che sta per essere acquistato; quindi, si possono aggiungere i dati relativi all'ordine:

Concetto	Costrutto	Accesso	Tipo
Gadget	Entità	1	Lettura
Ordina	Relazione	1	Scrittura

Si ricorda che il numero di operazioni è 100 a settimana e la scrittura costa il doppio della lettura.

$$\text{Costo: } 100 \cdot 1 + (100 \cdot 1) \cdot 2 = 300$$

- **Operazione 2**

Ora si accede ad ogni costo di ordine per farne la somma totale tenendo conto del volume indicativo:

Concetto	Costrutto	Accesso	Tipo
Ordina	Relazione	10'000	Lettura

Costo: $2 \cdot 10'000 = 20'000$

Costo settimanale totale: $300 + 20'000 = 20'300$

Senza ridondanza

- **Operazione 1**

In questo caso si aggiunge direttamente l'ordine:

Concetto	Costrutto	Accesso	Tipo
Ordina	Relazione	1	Scrittura

Si ricorda che il numero di operazioni è 100 a settimana e la scrittura costa il doppio della lettura.

Costo: $(100 \cdot 1) \cdot 2 = 200$

- **Operazione 2**

Ora bisogna accedere ad ogni ordine singolarmente per leggere la quantità di pezzi e per ciascun ordine accedere al prezzo del gadget:

Concetto	Costrutto	Accesso	Tipo
Ordina	Relazione	10'000	Lettura
Gadget	Entità	10'000	Lettura

Costo: $2 \cdot 10'000 + 2 \cdot 10'000 = 40'000$

Costo settimanale totale: $200 + 40'000 = 40'200$

Convienne, quindi, mantenere l'attributo **Costo** ed evitare di ricalcolare il costo di ciascun ordine.

Un'altra ridondanza è l'attributo **Punteggio** dell'entità "*Squadra*" che potrebbe essere calcolato dalla relazione "*vincitore*" con l'entità "*Partita*", la quale è relazionata, tramite la relazione "*gioca*", con l'entità "*Gioco*" il quale ha un attributo **Punteggio**.

Come indicato sopra, una squadra ottiene punteggio pieno se vince la partita in cui si gioca quello specifico gioco e non ottiene punti se la perde. Non sono ammessi pareggi.

- **Operazione 1 (180 a settimana):** inserimento partita
- **Operazione 2 (1 a settimana):** visualizzazione classifica

Concetto	Costrutto	Volume
Gioco	Entità	30
Partita	Entità	30'000
Squadra	Entità	5'000

Con ridondanza

- **Operazione 1**

Dopo aver inserito i dati della partita, si controlla il vincitore ed il punteggio da assegnare; successivamente si aggiorna il punteggio della squadra:

Concetto	Costrutto	Accesso	Tipo
Partita	Entità	1	Scrittura
Partita	Entità	1	Lettura
Gioco	Entità	1	Lettura
Squadra	Entità	1	Scrittura

Costo: $(180 \cdot 1) \cdot 2 + 180 \cdot 1 + 180 \cdot 1 + (180 \cdot 1) \cdot 2 = 1'080$

- **Operazione 2**

Ora si ordinano le squadre in base al punteggio per ottenere la classifica:

Concetto	Costrutto	Accesso	Tipo
Squadra	Entità	5'000	Lettura

Costo: $1 \cdot 5'000 = 5'000$

Costo settimanale totale: $1'080 + 5'000 = 6'080$

Senza ridondanza

- **Operazione 1**

Si aggiungono solo i dati della partita:

Concetto	Costrutto	Accesso	Tipo
Partita	Entità	1	Scrittura

Costo: $(180 \cdot 1) \cdot 2 = 360$

- **Operazione 2**

Per ogni partita si deve accedere al punteggio del gioco ed alla squadra vincitrice:

Concetto	Costrutto	Accesso	Tipo
Partita	Entità	30'000	Lettura
Gioco	Entità	30'000	Lettura

Costo: $1 \cdot 30'000 + 1 \cdot 30'000 = 60'000$

Costo settimanale totale: $360 + 60'000 = 60'360$

Chiaramente conviene mantenere anche questa ridondanza.

4.2. Gestione generalizzazioni

Generalizzazione	Risoluzione
Persona \leftarrow Adulto, Partecipante, Animatore	Persona è una generalizzazione totale ed esclusiva. Le entità Adulto e Partecipante sono però messe in relazione tra loro tramite la relazione "iscrive". Inoltre, l'entità Animatore ha altri due attributi propri. Perciò, si è deciso di partizionare la generalizzazione in 3 singole entità, rispettivamente Adulto, Partecipante e Animatore.

4.3. Partizionamento/Accorpamento di Entità e Relazioni

Diverse entità, dopo il partizionamento della generalizzazione, possiederebbero l'attributo "Indirizzo" che diventa poi ridondante. A tal motivo, viene creata un'entità **Indirizzo** che sarà in relazione con **Adulto**, **Partecipante**, **Animatore**.

Osservando la relazione "canzonePrincipale" (1:1) tra l'entità **Evento** e l'entità **Canzone** che ha un solo attributo *nome* (che è anche PK), si è pensato di eliminare l'entità e la rispettiva relazione accorpando l'attributo CanzonePrincipale nell'entità **Evento**.

Si è inoltre notato che la relazione "ordina" (N:M) creava una situazione in cui uno stesso animatore non poteva ordinare lo stesso gadget più volte.

A tal motivo, si è scelto di trasformare la relazione "ordina" (N:M) in due relazioni (1:N) che sono "effettua" e "composto" e l'aggiunta di un'entità **Ordine** avente come chiave primaria un Codice che è univoco per ogni ordine.

In questo modo:

- **Animatore** effettua (0:N) ordini e ogni **Ordine** è effettuato da (1:1) animatore.
- Ogni **Ordine** è composto da (1:1) **Gadget** e ogni Gadget può essere acquistato in (0:N) ordini.

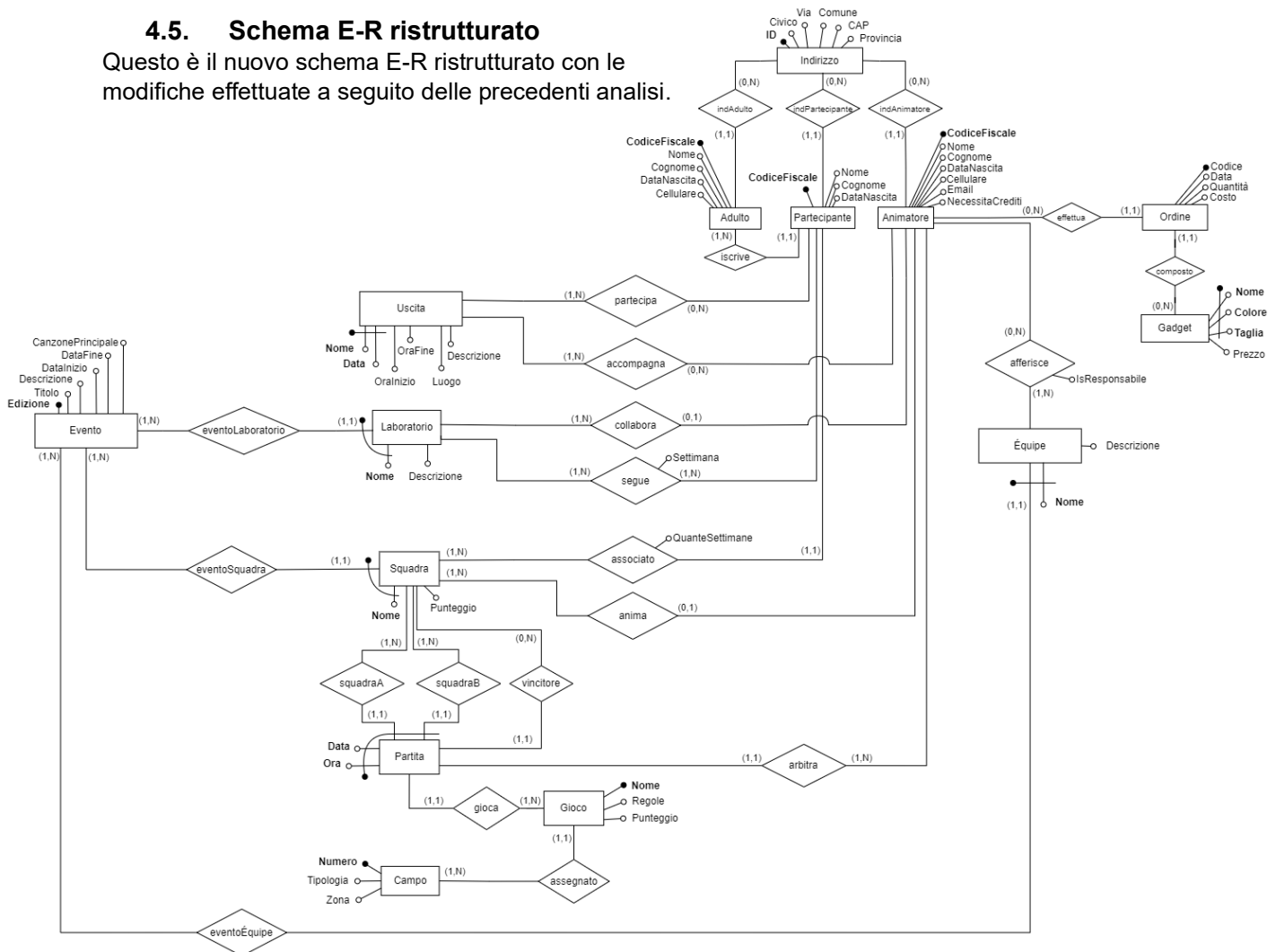
4.4. Scelta di Identificatori Primari

Un identificatore primario non banale è sicuramente la **chiave primaria della tabella Indirizzo**. Si poteva pensare ad una aggregazione di Via, Civico, Comune, CAP, Provincia, ma il tutto avrebbe comportato ad una copia di questi attributi nelle entità Adulto, Partecipante, Animatore relazionate (1:N) con la tabella Indirizzo. Perciò, si è optato per un identificativo numerico che si auto-incrementa. Un'ulteriore analisi la rivolgiamo alla **chiave primaria della tabella Ordine**. Si è deciso di utilizzare "Codice", di tipo stringa

lunga 8 caratteri, invece di una chiave composta da più attributi poiché fondamentale per garantire e implementarne, in una possibile versione web, il tracciamento della spedizione dal sito del corriere.

4.5. Schema E-R ristrutturato

Questo è il nuovo schema E-R ristrutturato con le modifiche effettuate a seguito delle precedenti analisi.



5. IMPLEMENTAZIONE SCHEMA LOGICO

5.1. Modello Logico Relazionale

Evento (Edizione, Titolo, Descrizione, DataInizio, DataFine, CanzonePrincipale)

Equipe (Nome, NumeroEdizione, Descrizione)

Laboratorio (Nome, NumeroEdizione, Descrizione)

Squadra (Nome, NumeroEdizione, Punteggio)

Indirizzo (ID, Civico, Via, Comune, CAP, Provincia)

Adulto (CodiceFiscale, Nome, Cognome, DataNascita, Cellulare, IDIndirizzo)

Partecipante (CodiceFiscale, Nome, Cognome, DataNascita, IDIndirizzo, CFGenitore)

Animatore (CodiceFiscale, Nome, Cognome, DataNascita, Cellulare, Email, NecessitaCrediti, IDIndirizzo)

Uscita (Nome, Data, Orainizio, OraFine, Luogo, Descrizione)

Gadget (Nome, Colore, Taglia, Prezzo)

Ordine (Codice, Data, Quantita, Costo, CFAnimatore, NomeGadget, ColoreGadget, TagliaGadget)

Afferisce (NomeEquipe, NumeroEdizione, CFAnimatore, IsResponsabile)

Campo (Numero, Tipologia, Zona)

Gioco (Nome, Regole, Punteggio, NumeroCampo)

Partita (Data, Ora, NomeSquadraA, EdizioneSquadraA, NomeSquadraB, EdizioneSquadraB, SquadraVincitrice, EdizioneSquadraVincitrice, NomeGioco, CFArbitro)

Partecipa (NomeUscita, DataUscita, CFPartecipante)

Accompagna (NomeUscita, DataUscita, CFAnimatore)

Collabora (NomeLaboratorio, NumeroEdizione, CFAnimatore)

Segue (NomeLaboratorio, NumeroEdizione, CFPartecipante, Settimana)

Associato (NomeSquadra, NumeroEdizione, CFPartecipante, QuanteSettimane)

Anima (NomeSquadra, NumeroEdizione, CFAnimatore)

5.2. Vincoli di Integrità Referenziale

Equipe.NumeroEdizione → Evento.Edizione
Laboratorio.NumeroEdizione → Evento.Edizione
Squadra.NumeroEdizione → Evento.Edizione
Adulto.IDIndirizzo → Indirizzo.ID
Partecipante.IDIndirizzo → Indirizzo.ID
Partecipante.CFGenitore → Adulto.CodiceFiscale
Animatore.IDIndirizzo → Indirizzo.ID
Ordine.CFAnimatore → Animatore.CodiceFiscale
Ordine.NomeGadget → Gadget.Nome
Ordine.ColoreGadget → Gadget.Colore
Ordine.TagliaGadget → Gadget.Taglia
Partita.NomeSquadraA → Squadra.Nome
Partita.EdizioneSquadraA → Evento.Edizione
Partita.NomeSquadraB → Squadra.Nome
Partita.EdizioneSquadraB → Evento.Edizione
Partita.SquadraVincitrice → Squadra.Nome
Partita.EdizioneSquadraVincitrice → Evento.Edizione
Partita.NomeGioco → Gioco.Nome
Partita.CFArbitro → Animatore.CodiceFiscale
Gioco.NumeroCampo → Campo.Numero
Afferisce.NomeEquipe → Equipe.Nome
Afferisce.NumeroEdizione → Evento.Edizione
Afferisce.CFAnimatore → Animatore.CodiceFiscale

Partecipa.NomeUscita → Uscita.Nome
Partecipa.DataUscita → Uscita.Data
Partecipa.CFPartecipante → Partecipante.CodiceFiscale
Accompagna.NomeUscita → Uscita.Nome
Accompagna.DataUscita → Uscita.Data
Accompagna.CFAnimatore → Animatore.CodiceFiscale
Collabora.NomeLaboratorio → Laboratorio.Nome
Collabora.NumeroEdizione → Evento.Edizione
Collabora.CFAnimatore → Animatore.CodiceFiscale
Segue.NomeLaboratorio → Laboratorio.Nome
Segue.NumeroEdizione → Evento.Edizione
Segue.CFPartecipante → Partecipante.CodiceFiscale
Associato.NomeSquadra → Squadra.Nome
Associato.NumeroEdizione → Evento.Edizione
Associato.CFPartecipante →
Partecipante.CodiceFiscale

Anima.NomeSquadra → Squadra.Nome
Anima.NumeroEdizione → Evento.Edizione
Anima.CFAnimatore → Animatore.CodiceFiscale

6. QUERY e INDICI

6.1. Query

6.1.1. Query 1: Si restituiscano i codici fiscali dei partecipanti iscritti a meno di quattro settimane di evento ma che partecipano ad almeno un'uscita.

```
select associato.cfpartecipante as "Codice Fiscale"
from associato
where quantesettimane != '1234' and exists(select * from partecipa where
partecipa.cfpartecipante = associato.cfpartecipante);
```

	Codice Fiscale character	
1	MLTFRN12P03C224M	
2	VLLMRC15Z13C745D	
3	GRNGRN16A14C517R	
4	DLPCRS15S06C224M	
5	GRGFNC17N01C745D	

6.1.2. Query 2: Si restituisca la classifica finale dell'edizione x (dove x è passato per parametro) del Centro Estivo (dalla squadra che ha ottenuto più punti a quella che ne ha ottenuti meno) avendo come output anche il numero di partite vinte. (nella query e nell'output di esempio: x = 2)

```
drop view if exists numerovittorie;

create view numerovittorie as
select partita.squadravincitrice as squadravin, count(partita.squadravincitrice)
as "Partite Vinte"
from partita
where partita.edizionesquadravincitrice = 2
group by partita.squadravincitrice;
```

(continua)

```
select squadra.nome as "Nome Squadra", squadra.punteggio as "Punteggio",
"Partite Vinte"
from squadra inner join numerovittorie on squadra.nome = squadravin order by
squadra.punteggio desc;
```

	Nome Squadra character varying (30)	Punteggio integer	Partite Vinte bigint
1	Luckylama	45	2
2	Team Bros	45	2
3	Green Links	45	2
4	Pro Fortnite	45	2
5	Proplayer	35	1
6	Royale Squad	30	3
7	Creeper	30	3
8	Zelda	30	3
9	Hackerman	10	1
10	Pacman	10	1

6.1.3. Query 3: Per ogni laboratorio che ha x o più partecipanti nella settimana y dell'edizione z (dove x, y e z sono passati per parametro) si mostrino nome, cognome e indirizzo degli animatori che vi collaborano. (nella query e nell'output di esempio: x = 3, y = 2, z = 1)

```
drop view if exists listaanimatori;
drop view if exists listalaboratori;

create view listalaboratori as
select segue.nomelaboratorio as nomelab, segue.numeroedizione as ed from segue
where segue.numeroedizione = 1 and segue.settimana = 2
group by segue.nomelaboratorio, segue.numeroedizione
having count(segue.cfpartecipante) >= 3;

create view listaanimatori as
select collabora.cfanimatore as cf
from collabora inner join listalaboratori on collabora.nomelaboratorio = nomelab
and collabora.numeroedizione = ed;

select animatore.nome, animatore.cognome, indirizzo.via, indirizzo.civico,
indirizzo.comune, indirizzo.cap, indirizzo.provincia
from (animatore inner join indirizzo on animatore.idindirizzo = indirizzo.id)
inner join listaanimatori on animatore.codicefiscale = cf;
```

	nome character varying (12)	cognome character varying (12)	via character varying (20)	civico character varying (5)	comune character varying (20)	cap character	provincia character
1	Annarita	Bilardo	Canova	2	Camposampiero	35012	PD
2	Lorenzo	Bilotta	Palladio	17	Camposampiero	35012	PD
3	Zhe	Bonamente	Tiso	4A	Camposampiero	35012	PD
4	Giovanni	Capobianco	Tentori	1	Camposampiero	35012	PD
5	Luca	Carocci	Canova	5	Camposampiero	35012	PD
6	Anna	Gentilcore	Piave	4	Camposampiero	35012	PD
7	Filippo	Grassini	Vecellio	32	Camposampiero	35012	PD
8	Gaetano	File	Albarella	1	Camposampiero	35012	PD
9	Eduard	Alimena	Piave	2	Camposampiero	35012	PD
10	Darya	El Mestour	Palladio	12B	Camposampiero	35012	PD
11	Gabriele	Melidona	Vecellio	25	Camposampiero	35012	PD
12	Zhe	Bonamente	Tiso	4A	Camposampiero	35012	PD
13	Marta	Periotto	Tiso	3	Camposampiero	35012	PD
14	Francesca	Passuello	Albarella	16A	Camposampiero	35012	PD
15	Anda	Racu	Canova	14	Camposampiero	35012	PD

6.1.4. Query 4: Si mostri il codice dell'ordine meno costoso per ciascuna edizione fatto da un animatore che è oppure è stato responsabile di qualche équipe (null se nessun animatore responsabile ha effettuato ordini nell'edizione).

```
drop view if exists ordinievento;
drop view if exists ordiniresponsabile;

create view ordiniresponsabile as
select ordine.cfanimatore, ordine.data, ordine.codice, ordine.costo
from ordine inner join afferisce on ordine.cfanimatore = afferisce.cfanimatore
where afferisce.isresponsabile = true;
```

(continua)

```

create view ordinievento as
select evento.edizione, ordiniresponsabile.codice as "Codice Ordine",
ordiniresponsabile.costo
from evento left outer join Ordiniresponsabile on evento.datainizio <=
ordiniresponsabile.data and evento.datafine >= ordiniresponsabile.data;

select distinct ordinievento.edizione as "Edizione", ordinievento."Codice
Ordine"
from ordinievento, (select ordinievento.edizione as ed, min(ordinievento.costo)
as minimo
from ordinievento
group by ordinievento.edizione) as ordiniminimi
where ordinievento.edizione = ed and ordinievento.costo = minimo or
ordinievento.costo is null;

```

	Edizione integer	Codice Ordine character
1	1	FH49GML0
2	2	[null]
3	3	2HF8N4MF

6.1.5. Query 5: Si mostri il numero di partecipanti ad ogni squadra e ad ogni laboratorio per edizione.

```

select squadra.nome as "Nome", 'Squ.' as "Tipo", squadra.numeroedizione as
"Numero Edizione", count(associato.cfpartecipante) as "Numero Iscritti"
from squadra
inner join associato on squadra.nome = associato.nomesquadra and
squadra.numeroedizione = associato.numeroedizione
group by squadra.numeroedizione, squadra.nome
union
select laboratorio.nome as "Nome", 'Lab.' as "Tipo", laboratorio.numeroedizione
as "Numero Edizione", count(segue.cfpartecipante) as "Numero Iscritti"
from laboratorio
inner join segue on laboratorio.nome = segue.nomelaboratorio and
laboratorio.numeroedizione = segue.numeroedizione
group by laboratorio.numeroedizione, laboratorio.nome
order by "Numero Edizione", "Tipo", "Nome";

```

	Nome character varying (30)	Tipo text	Numero Edizione integer	Numero Iscritti bigint
1	Decoupage	Lab.	1	8
2	Elettricista	Lab.	1	10
3	Falegname	Lab.	1	9
4	Filografia	Lab.	1	9
5	Magliette Schizzate	Lab.	1	11
6	MasterChef	Lab.	1	10
7	Tecnoled	Lab.	1	16
8	Alice	Squ.	1	4
9	Bruco	Squ.	1	2
10	Cappellaio	Squ.	1	3
11	Dodo	Squ.	1	4
12	Gatto Cheshire	Squ.	1	2
13	Giangi	Squ.	1	3
14	Grillo	Squ.	1	3
15	Regina Bianca	Squ.	1	3
16	Regina Cuori	Squ.	1	4
17	Tartaruga	Squ.	1	2
18	Casette Miniatura	Lab.	2	14
19	Decoupage	Lab.	2	9
20	Falegname	Lab.	2	11
21	Magliette Schizzate	Lab.	2	11
22	MasterChef	Lab.	2	12
23	Pirografia	Lab.	2	7
24	Tecnoled	Lab.	2	14
25	Creeper	Squ.	2	2

26	Green Links	Squ.	2	2
27	Hackerman	Squ.	2	2
28	Luckylama	Squ.	2	2
29	Pacman	Squ.	2	3
30	Pro Fortnite	Squ.	2	2
31	Proplayer	Squ.	2	5
32	Royale Squad	Squ.	2	2
33	Team Bros	Squ.	2	2
34	Zelda	Squ.	2	3
35	Braccialetti	Lab.	3	7
36	Falegname	Lab.	3	10
37	Filografia	Lab.	3	7
38	MasterChef	Lab.	3	12
39	Pirografia	Lab.	3	7
40	Sacche Schizzate	Lab.	3	9
41	Tecnoled	Lab.	3	14
42	Allbabibbia	Squ.	3	2
43	Boss dei Sogni	Squ.	3	2
44	Da cosi a cosi	Squ.	3	2
45	Followme	Squ.	3	3
46	Piccoli Apostoli	Squ.	3	2
47	Principi dei Sogni	Squ.	3	4
48	Profeti di IG	Squ.	3	3
49	Risorgivi	Squ.	3	3
50	Sognatori	Squ.	3	2
51	Sognatori Chill	Squ.	3	2

6.1.6. Query 6: Si mostrino i campi ordinati in base alla media dei punteggi ottenibili dai giochi organizzati in quel campo (dalla più alta alla più bassa).

```
select campo.numero as "Numero", avg(gioco.punteggio) as "Media Punti"
from campo inner join gioco on campo.numero = gioco.numeroocampo
group by campo.numero
order by "Media Punti" desc;
```

	Numero integer	Media Punti numeric
1	61	25.0000000000000000
2	12	25.0000000000000000
3	63	25.0000000000000000
4	13	25.0000000000000000
5	41	25.0000000000000000
6	43	25.0000000000000000
7	32	20.0000000000000000
8	51	20.0000000000000000
9	42	20.0000000000000000
10	62	20.0000000000000000
11	53	20.0000000000000000
12	22	15.0000000000000000
13	21	15.0000000000000000
14	11	15.0000000000000000
15	23	12.5000000000000000
16	31	10.0000000000000000
17	33	10.0000000000000000
18	52	10.0000000000000000

6.2. Indici

Analisi:

Si prenda in considerazione la Query: Mostrare il numero di partecipanti ad ogni squadra e ad ogni laboratorio per edizione. Questa query potrebbe beneficiare di un indice significativo perché coinvolge molteplici tabelle (*squadra*, *associato*, *laboratorio*, *segue*) e operazioni di aggregazione (*count* e *group by*). Un indice adeguato potrebbe migliorare notevolmente le prestazioni di queste operazioni, soprattutto quando si tratta di tabelle con un gran numero di righe.

Implementazione:

Creeremo un indice su *associato.nomesquadra* e *associato.numeroedizione* per ottimizzare le ricerche e le aggregazioni nella prima parte della query. Allo stesso modo, creeremo un indice su *segue.nomelaboratorio* e *segue.numeroedizione* per ottimizzare le ricerche e le aggregazioni nella seconda parte della query.

```
drop index if exists idx_associato_nomesquadra_edizione;
drop index if exists idx_segue_nomelaboratorio_edizione;
```

```
create index idx_associato_nomesquadra_edizione on associato(nomesquadra,
numeroedizione);
create index idx_segue_nomelaboratorio_edizione on segue(nomelaboratorio,
numeroedizione);
```

Commento:

L'indice *idx_associato_nomesquadra_edizione* aiuterà a velocizzare l'operazione di **join** e **group by** tra *squadra* e *associato*, riducendo il tempo di scansione delle righe. Analogamente, l'indice *idx_segue_nomelaboratorio_edizione* migliorerà le prestazioni per l'operazione di **join** e **group by** tra *laboratorio* e *segue*.

Entrambi gli indici contribuiscono a rendere le operazioni di conteggio più efficienti, specialmente in presenza di grandi volumi di dati. Inoltre tutte le tabelle interessate vengono aggiornate principalmente solo all'inizio di ogni edizione dal momento che la maggior parte delle iscrizioni avvengono ad inizio evento. In questo modo si concentra l'overhead dato dall'aggiornamento degli indici.

Migliorie:

Osserviamo le migliorie ottenute grazie all'implementazione degli indici.

```
EXPLAIN
select squadra.nome as "Nome", 'Squ.' as "Tipo", squadra.numeroedizione as "Numero Edizione", count(associato.cfpartecipante) as "Numero Iscritti"
from squadra
inner join associato on squadra.nome = associato.nomesquadra and squadra.numeroedizione = associato.numeroedizione
group by squadra.numeroedizione, squadra.nome
union
select laboratorio.nome as "Nome", 'Lab.' as "Tipo", laboratorio.numeroedizione as "Numero Edizione", count(segue.cfpartecipante) as "Numero Iscritti"
from laboratorio
inner join segue on laboratorio.nome = segue.nomelaboratorio and laboratorio.numeroedizione = segue.numeroedizione
group by laboratorio.numeroedizione, laboratorio.nome
order by "Numero Edizione", "Tipo", "Nome"
```

ta Output Messages Notifications

QUERY PLAN
text
Sort (cost=73.27..73.37 rows=41 width=122)
Sort Key: squadra.numeroedizione, ('Squ.:text'), squadra.nome
-> HashAggregate (cost=71.76..72.17 rows=41 width=122)
Group Key: squadra.nome, ('Squ.:text'), squadra.numeroedizione, (count(associato.cfpartecipante))
-> Append (cost=27.35..71.35 rows=41 width=122)
-> HashAggregate (cost=27.35..27.65 rows=30 width=55)
Group Key: squadra.numeroedizione, squadra.nome
-> Merge Join (cost=0.28..26.75 rows=80 width=32)
Merge Cond: (((squadra.nome)::text = (associato.nomesquadra)::text) AND (squadra.numeroedizione = associato.numeroedizi...
-> Index Only Scan using squadra_pkey on squadra (cost=0.14..12.59 rows=30 width=15)
-> Index Scan using idx_associato_nomesquadra_edizione on associato (cost=0.14..13.34 rows=80 width=31)
-> HashAggregate (cost=42.98..43.09 rows=11 width=56)
Group Key: laboratorio.numeroedizione, laboratorio.nome
-> Hash Join (cost=12.91..41.35 rows=217 width=33)
Hash Cond: (((segue.nomelaboratorio)::text = (laboratorio.nome)::text) AND (segue.numeroedizione = laboratorio.numeroedizi...
-> Index Scan using idx_segue_nomelaboratorio_edizione on segue (cost=0.14..27.34 rows=217 width=33)
-> Hash (cost=12.45..12.45 rows=21 width=16)
-> Index Only Scan using laboratorio_pkey on laboratorio (cost=0.14..12.45 rows=21 width=16)

La query in oggetto ora viene eseguita interamente nell'indice. Infatti, eseguendola, come riportato nel prompt qua sopra, possiamo vedere – ove indicato con le frecce rosse – che utilizza l'**Index Scan** (indici) per restituirci tutti i risultati della query.

7. APPLICAZIONE SOFTWARE PER ACCESSO AL DATABASE

Si allega file .zip contenente:

- Le istruzioni SQL per la creazione delle tabelle, per il popolamento, per la creazione degli indici opportuni e inclusivo delle query
- I file C per eseguire le query da software (con il commento di alcune righe). Nome: query_grest.c
- La cartella *dependencies* contenente le dipendenze e le librerie necessarie per l'esecuzione del programma.

Breve commento al codice:

Dopo aver compilato il programma con il comando

- Linux: gcc query_grest.c -oquery -I /usr/include/postgresql -lpq
- Windows: gcc -I dependencies\include -L dependencies\lib query_grest.c -o query.exe -lpq

lo si esegua. Fare attenzione che nella cartella in cui si trova il file .c ci sia anche la cartella *dependencies*.

Durante l'esecuzione vi verranno chiesti alcuni parametri di autenticazione.

Utilizzare **Username:** eartusi e **Password:** D@_+9R_1;2BQ

Se la connessione avverrà in modo corretto, apparirà un menù visuale (in Command Line Interface) in cui vi verrà chiesto di inserire un numero, rispettivamente:

- **(-1)** per uscire dal programma
- **(0)** per visualizzare una lista delle query
- **(da 1 a 6)** per eseguire la rispettiva query corrispondente a quanto visibile nell'elenco query.

I risultati delle query sono visibili tramite una visualizzazione a tabella generata nel medesimo prompt di esecuzione.

(Il codice è stato eseguito in una macchina LINUX di laboratorio per la connessione locale al server postgresql)