



Basi di Dati
Progetto A.A. 2023/2024

CHAT MULTICANALE

0314881

Stefano Ostili

Indice

1.	Descrizione del Minimondo	2
2.	Analisi dei Requisiti	3
3.	Progettazione concettuale	7
4.	Progettazione logica	11
5.	Progettazione fisica	20

1. Descrizione del Minimondo

1 Si vuole realizzare un sistema informativo per consentire ai lavoratori di una azienda di
2 scambiare messaggi legati ai progetti a cui stanno attualmente lavorando.

3 Il sistema prevede tre livelli di utenza: gli amministratori, i capi progetto, i dipendenti.

4 Gli amministratori hanno la possibilità di gestire quali utenti sono capi progetto.

5 I capi progetto possono creare un numero arbitrario di canali di comunicazione ed invitare al
6 loro interno tutti i dipendenti che cooperano sulle attività del progetto.

7 All'interno di un canale gli utenti possono inviare messaggi che possono essere letti da tutti
8 gli altri appartenenti al canale.

9 I messaggi sono organizzati in pagine e gli utenti possono visualizzare, una per una, le pagine
10 della conversazione.

11 Un utente del sistema ha la possibilità di rispondere pubblicamente in un canale. In questa
12 risposta, può decidere di riferire un messaggio precedentemente inviato, così che il suo
13 messaggio appaia come risposta ad una parte specifica della comunicazione. Allo stesso
14 modo, partendo da un qualsiasi messaggio, l'utilizzatore può decidere di rispondere in modo
15 privato. Questa risposta privata aprirà un canale di discussione privato tra lui e il mittente del
16 messaggio cui si sta rispondendo. I project manager possono sempre accedere (in sola lettura)
17 a tutte le discussioni private nate nei canali dei progetti di cui sono responsabili.

18 Un utente può accedere in qualsiasi momento a tutti i canali di cui fa parte, e a tutte le
19 conversazioni private (organizzate per canale) che ha creato o in cui è stato coinvolto.

2. Analisi dei Requisiti

Identificazione dei termini ambigui e correzioni possibili

Linea	Termine	Nuovo termine	Motivo correzione
3, 6	Dipendenti	Dipendenti base	Il termine dipendenti si riferisce all'insieme degli amministratori, dei capi progetto e dei dipendenti base. Il termine dipendenti base indica l'insieme di dipendenti che non sono né amministratori né capi progetto
4, 7, 9, 11, 18	Utente	Impiegato	Nel sistema, il termine utente si riferisce a qualsiasi utilizzatore, inclusi gli amministratori. Tuttavia, nel contesto specifico della specifica, ci si riferisce esclusivamente ai capi progetto e ai dipendenti base, raggruppati sotto il termine impiegato.
5, 7, 11, 17, 18, 19	Canale	Canale pubblico	Distinzione tra canale pubblico e privato
10, 19	Conversazione	Canale	La conversazione, in questo caso, coincide con il canale
14	Utilizzatore	Utente	Individuano lo stesso concetto, è preferibile utilizzare un solo termine
15	Discussione	Comunicazione	Si riferisce a canale, che in precedenza è stato riferito come canale di comunicazione
16	Project manager	Capi progetto	Individuano lo stesso concetto, è preferibile utilizzare un solo termine
17	Discussioni	Canali	Si intende l'accesso in sola lettura ai canali privati

Specificazione disambiguata

Si vuole realizzare un sistema informativo per consentire ai lavoratori di una azienda di scambiare messaggi legati ai progetti a cui stanno attualmente lavorando.

Il sistema prevede tre livelli di utenza: gli amministratori, i capi progetto, i dipendenti base.

Gli amministratori hanno la possibilità di gestire quali impiegati sono capi progetto.

I capi progetto possono creare un numero arbitrario di canali pubblici di comunicazione ed invitare al loro interno tutti i dipendenti base che cooperano sulle attività del progetto.

All'interno di un canale pubblico gli impiegati possono inviare messaggi che possono essere letti da tutti gli altri appartenenti al canale.

I messaggi sono organizzati in pagine e gli impiegati possono visualizzare, una per una, le pagine del canale.

Un impiegato ha la possibilità di rispondere pubblicamente in un canale pubblico. In questa risposta, può decidere di riferire un messaggio precedentemente inviato, così che il suo messaggio appaia come

risposta ad una parte specifica della comunicazione. Allo stesso modo, partendo da un qualsiasi messaggio, l'impiegato può decidere di rispondere in modo privato. Questa risposta privata aprirà un canale di comunicazione privato tra lui e il mittente del messaggio cui si sta rispondendo. I capi progetto possono sempre accedere (in sola lettura) a tutti i canali privati nati dai canali pubblici dei progetti di cui sono responsabili.

Un impiegato può accedere in qualsiasi momento a tutti i canali pubblici di cui fa parte, e a tutti i canali privati che ha creato o in cui è stato coinvolto.

Glossario dei Termini

Termine	Descrizione	Sinonimi	Collegamenti
Dipendente Base	Impiegato che collabora alle attività dei progetti di cui fa parte	Dipendente	Impiegato
Capo Progetto	Impiegato che gestisce i progetti	Project Manager	Impiegato, Progetto
Amministratore	Lavoratore che gestisce i capi progetto		Impiegato
Impiegato	Colui che utilizza il sistema informativo	Utente, utilizzatore	Progetto, Canale, Messaggio, Amministratore, Capo Progetto, Dipendente Base
Progetto	Insieme di attività lavorative a cui partecipano gli impiegati		Impiegato, Canale, Capo Progetto
Canale	Spazio virtuale di comunicazione tra i membri di un progetto, può essere pubblico o privato	Conversazione, discussione	Impiegato, Progetto, Messaggio
Messaggio	Testo di lunghezza variabile che i partecipanti ad un progetto si scambiano all'interno di un canale	Risposta	Impiegato, Canale

Raggruppamento dei requisiti in insiemi omogenei

Frasi relative a Dipendente Base

Il sistema prevede tre livelli di utenza: gli amministratori, i capi progetto, i dipendenti base.

I capi progetto possono creare un numero arbitrario di canali pubblici di comunicazione ed invitare al loro interno tutti i dipendenti base che cooperano sulle attività del progetto.

Frasi relative a Capo Progetto

Il sistema prevede tre livelli di utenza: gli amministratori, i capi progetto, i dipendenti base.

I capi progetto possono creare un numero arbitrario di canali pubblici di comunicazione ed invitare al loro interno tutti i dipendenti base che cooperano sulle attività del progetto.

I capi progetto possono sempre accedere (in sola lettura) a tutti i canali privati nati dai canali pubblici dei progetti di cui sono responsabili.

Frasi relative ad Amministratore

Il sistema prevede tre livelli di utenza: gli amministratori, i capi progetto, i dipendenti base.

Gli amministratori hanno la possibilità di gestire quali impiegati sono capi progetto.

Frasi relative ad Impiegato

All'interno di un canale pubblico gli impiegati possono inviare messaggi che possono essere letti da tutti gli altri appartenenti al canale.

I messaggi sono organizzati in pagine e gli impiegati possono visualizzare, una per una, le pagine del canale.

Un impiegato ha la possibilità di rispondere pubblicamente in un canale pubblico.

L'impiegato può decidere di rispondere in modo privato.

Un impiegato può accedere in qualsiasi momento a tutti i canali pubblici di cui fa parte, e a tutti i canali privati che ha creato o in cui è stato coinvolto.

Frasi relative a Canale

I capi progetto possono creare un numero arbitrario di canali pubblici di comunicazione.

All'interno di un canale pubblico gli impiegati possono inviare messaggi che possono essere letti da tutti gli altri appartenenti al canale.

Gli impiegati possono visualizzare, una per una, le pagine del canale.

Questa risposta privata aprirà un canale di comunicazione privato tra lui e il mittente del messaggio cui si sta rispondendo.

Frase relative a Progetto

I capi progetto possono creare un numero arbitrario di canali pubblici di comunicazione ed invitare al loro interno tutti i dipendenti base che cooperano sulle attività del progetto.

Frase relative a Messaggio

All'interno di un canale pubblico gli impiegati possono inviare messaggi che possono essere letti da tutti gli altri appartenenti al canale.

I messaggi sono organizzati in pagine e gli impiegati possono visualizzare, una per una, le pagine del canale.

3. Progettazione concettuale

Costruzione dello schema E-R

Per la costruzione dello schema E-R sono partito analizzando il glossario dei termini ed elaborando uno schema scheletro, rappresentato in Figura 1.

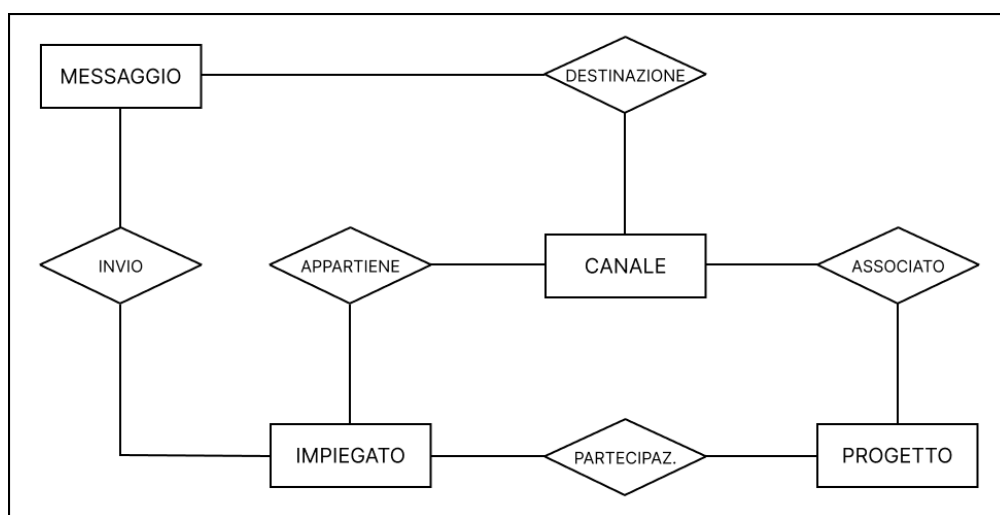


Figura 1: Schema scheletro

Analizzando successivamente le singole entità ho potuto raffinare lo schema scheletro aggiungendo ulteriori entità, relazioni ed attributi.

Per quanto riguarda l'entità IMPIEGATO ho aggiunto gli attributi *Nome*, *Cognome* e *Codice Fiscale*, quest'ultimo è utilizzato come identificatore.

Per quanto riguarda l'entità PROGETTO ho aggiunto gli attributi *Nome*, *Data Scadenza* e *Codice*, quest'ultimo è utilizzato come identificatore.

Ho inoltre generalizzato l'entità IMPIEGATO aggiungendo le entità AMMINISTRATORE, DIPENDENTE BASE e CAPO PROGETTO. In questo modo posso distinguere i diversi ruoli che un impiegato può assumere all'interno dell'azienda.

Successivamente ho inserito la relazione GESTIONE che permette al capo progetto di gestire i singoli progetti di cui è responsabile.

Aggiungendo tutte le cardinalità tra entità e relazioni, lo schema che ne deriva è rappresentato in Figura 2.

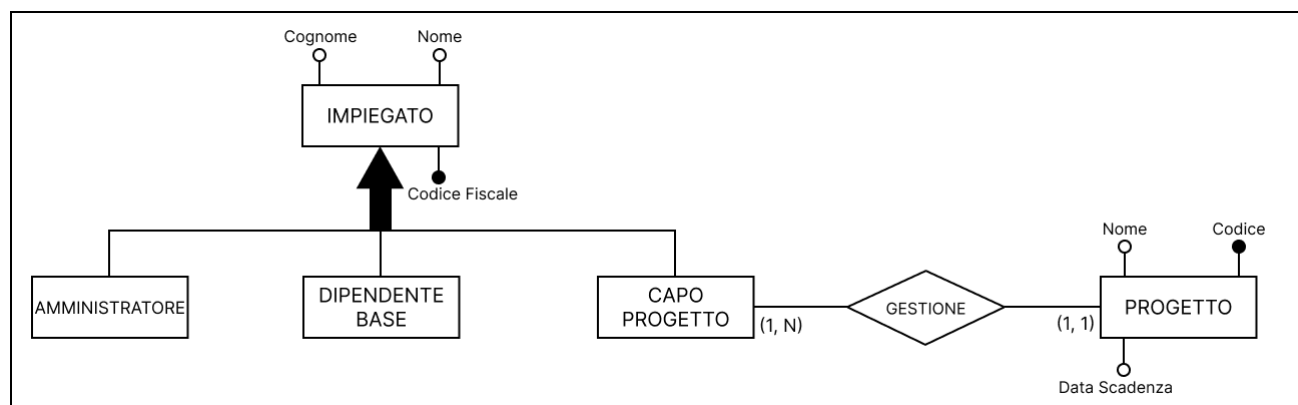


Figura 2: Raffinamento entità UTENTE e PROGETTO

Per quanto riguarda l'entità MESSAGGIO ho aggiunto la relazione ricorsiva RIFERIMENTO per permettere così all'impiegato di poter inviare un messaggio in riferimento ad un altro messaggio presente nel canale, con l'attributo *Tipo* è possibile distinguere se il messaggio debba essere mandato in un canale pubblico o generare un canale privato. Inoltre, ho aggiunto gli attributi *Parole*, *Data Invio* e *Ora Invio*.

Questa entità è identificata dalla data e dall'ora dell'invio e dall'utente che effettua tale operazione.

Per quanto riguarda l'entità CANALE ho aggiunto l'attributo *Nome*.

Questa entità è identificata dal nome del canale e dal progetto alla quale fa riferimento.

Ho inoltre generalizzato l'entità CANALE aggiungendo le entità CANALE PUBBLICO e CANALE PRIVATO, distinguendo così le diverse tipologie di canale.

Aggiungendo tutte le cardinalità tra le relazioni, lo schema che ne deriva è raffigurato in Figura 3.

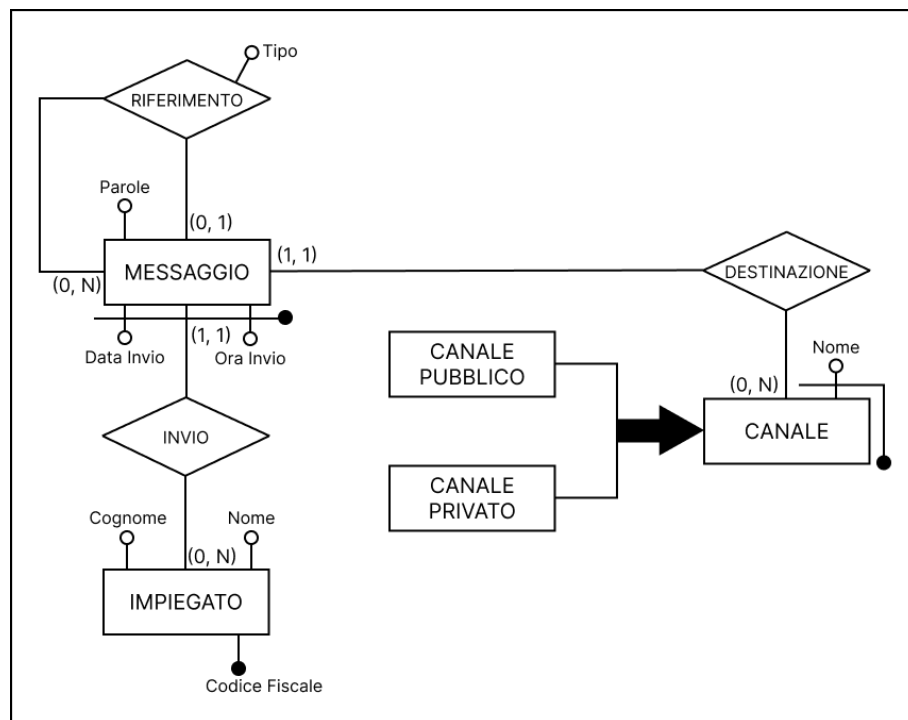


Figura 3: Raffinamento entità MESSAGGIO e CANALE

Integrazione finale

Accorpendo i vari schemi parziali rappresentati precedentemente ho ottenuto lo schema finale rappresentato in Figura 4.

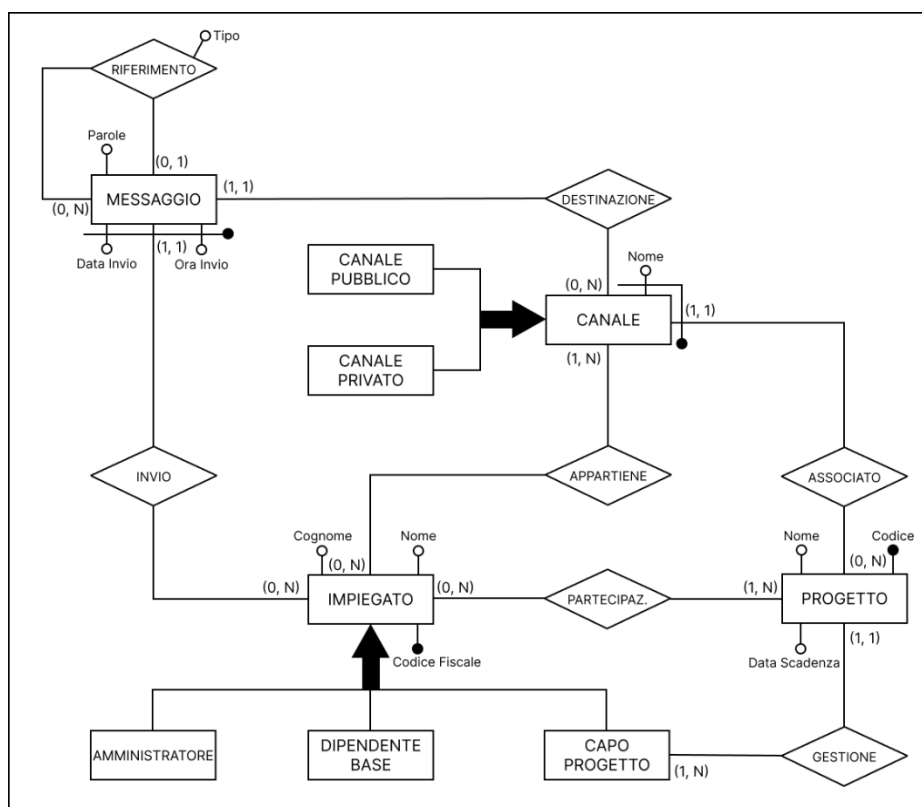


Figura 4: Schema finale

Regole aziendali

I canali privati possono essere creati solo attraverso risposte private ai messaggi esistenti in un canale pubblico e sono accessibili, oltre che al capo progetto, in sola lettura, solo al mittente del messaggio privato e al mittente del messaggio riferito.

Un capo progetto può creare solo canali pubblici relativi ai progetti di cui è responsabile.

La data di scadenza di un progetto non può essere antecedente alla data al momento della creazione di questo.

Il progetto di riferimento di un canale privato è quello del canale pubblico da cui è nato il canale privato.

Non possono coesistere due progetti con lo stesso nome.

Non possono coesistere due canali con lo stesso nome.

I messaggi all'interno di un canale sono visualizzabili in ordine cronologico.

Dizionario dei dati

Entità	Descrizione	Attributi	Identificatori
Dipendente Base	Impiegato che partecipa ai progetti aziendali	Nome, Cognome,	Codice Fiscale
Capo Progetto	Impiegato incaricato di gestire uno o più progetti	Nome, Cognome,	Codice Fiscale
Amministratore	Impiegato incaricato di creare progetti	Nome, Cognome,	Codice Fiscale
Impiegato	Termine generico per rappresentare tutti i lavoratori dell'azienda	Nome, Cognome,	Codice Fiscale
Progetto	Un insieme di attività aziendali gestite da un capo progetto	Nome, Data Scadenza	Codice
Canale	Spazio virtuale di comunicazione		Nome, Associato
Canale Pubblico	Spazio virtuale di comunicazione aperto a tutti gli impiegati inseriti dal capo progetto		Nome, Associato
Canale Privato	Spazio virtuale di comunicazione ristretto creato in risposta a un messaggio di un canale pubblico, si compone di soli due membri		Nome, Associato
Messaggio	Testo inviato da un impiegato in un canale	Parole	Data Invio, Ora Invio, Invio

4. Progettazione logica

Volume dei dati

Un impiegato invia in media 30 messaggi al giorno distribuiti tra i vari canali.

I canali relativi ai progetti scaduti vengono eliminati 12 mesi dopo dalla data di scadenza.

In media il 30% dei messaggi fa riferimento ad altri messaggi.

In media il 5% dei messaggi riferiti ha come destinazione un canale privato.

Concetto nello schema	Tipo ¹	Volume atteso
Impiegato	E	600
Amministratore	E	20
Capo Progetto	E	180
Dipendente Base	E	400
Progetto	E	360
Canale	E	4.320
Canale Pubblico	E	1.080
Canale Privato	E	3.240
Messaggio	E	4.350.000
Partecipazione	R	1.740
Gestione	R	360
Associato	R	1.701
Appartiene	R	4.320
Invio	R	4.350.000
Destinazione	R	4.350.000
Riferimento	R	1.740.000

Tavola delle operazioni

Cod.	Descrizione	Frequenza attesa
AM1	Inserimento impiegato	30/anno
AM2	Assegnazione capo progetto a progetto	348/anno
AM3	Creazione progetto	348/anno
AM4	Visualizza impiegati	678/anno
CP1	Creazione canale pubblico relativo ad un progetto	720/anno
CP2	Inserimento dipendente base in canale pubblico	4.320/anno
CP3	Visualizzazione messaggi canale privato	30/anno

¹ Indicare con E le entità, con R le relazioni

CP4	Inserimento dipendente base in un progetto	1.740/anno
CP5	Visualizza la lista dei progetti di cui si è responsabili	3.480/anno
CP6	Visualizza membri di un progetto	870/anno
CP7	Visualizza impiegati non ancora aggiunti ad un progetto	1.740/anno
CP8	Visualizza impiegati non ancora aggiunti ad un canale	4.320/anno
IM1	Invio messaggio senza riferimento in un canale	10.440/giorno
IM2	Rispondi ad un messaggio in riferimento ad un altro messaggio	6.960/giorno
IM3	Visualizzazione canali di cui un impiegato fa parte	5.800/giorno
IM4	Visualizzazione membri di un canale	1.400/anno
IM5	Visualizzazione messaggi di un canale	17.400/giorno
L1	Login	4.200/giorno

Tutte le operazioni “IM_” sono svolte sia dai dipendenti base sia dai capi progetto.

Costo delle operazioni

AM1 – Inserimento impiegato

- Impiegato: 1/Scrittura
- Amministratore/Dipendente Base/Capo Progetto: 1/Scrittura

Costo totale: 4

Accessi/anno: 120

AM2 – Assegnazione capo progetto a progetto

- Capo Progetto: 1/Lettura
- Impiegato: 1/Lettura
- Gestione: 1/Scrittura
- Progetto: 1/Lettura

Costo totale: 5

Accessi/anno: 1.740

AM3 – Creazione Progetto

- Progetto: 1/Scrittura

Costo totale: 2

Accessi/anno: 696

AM4 – Visualizza Impiegati

- Impiegato: 1/Lettura

Costo totale: 1

Accessi/anno: 678

CP1 – Creazione canale pubblico

- Canale: 1/Scrittura
- Canale Pubblico: 1/Scrittura

Costo totale: 4

Accessi/anno: 2.880

CP2 – Inserimento dipendente base in un canale

- Dipendente Base: 1/Lettura
- Impiegato: 1/Lettura
- Appartiene: 1/Scrittura
- Canale: 1/Lettura
- Canale Pubblico: 1/Lettura

Costo totale: 6

Accessi/anno: 25.920

CP3 – Visualizzazione dei messaggi in un canale privato

- Impiegato: 1/Lettura
- Capo Progetto: 1/Lettura
- Gestione: 1/Lettura
- Progetto: 1/Lettura
- Associato: 1/Lettura
- Canale: 1/Lettura
- Canale Privato: 1/Lettura
- Destinazione: 200/Lettura
- Messaggio: 200/Lettura

Costo totale: 407

Accessi/anno: 12.210

CP4 – Inserimento dipendente base in un progetto

- Dipendente Base: 1/Lettura
- Impiegato: 1/Lettura
- Partecipazione: 1/Scrittura

- Progetto: 1/Lettura

Costo totale: 5

Accessi/anno: 8.700

CP5 – Visualizza lista progetti

- Capo Progetto: 1/Lettura
- Impiegato: 1/Lettura
- Partecipazione: 2/Lettura
- Progetto: 2/Lettura

Costo totale: 6

Accessi/anno: 20.880

CP6 – Visualizza membri di un progetto

- Capo Progetto: 1/Lettura
- Impiegato: 1/Lettura
- Partecipazione: 12/Lettura
- Progetto: 12/Lettura

Costo totale: 26

Accessi/anno: 22.620

CP7 – Visualizza assenti nel progetto

- Capo Progetto: 1/Lettura
- Impiegato: 1/Lettura
- Partecipazione: 12/Lettura
- Progetto: 12/Lettura

Costo totale: 26

Accessi/anno: 45.240

CP8 – Visualizza assenti nel canale

- Capo Progetto: 1/Lettura
- Impiegato: 1/Lettura
- Appartiene: 12/Lettura
- Progetto: 12/Lettura

Costo totale: 26

Accessi/anno: 112.320

IM1 – Invio messaggio senza riferimento

- Impiegato: 1/Lettura
- Invio: 1/Scrittura
- Messaggio: 1/Scrittura
- Destinazione: 1/Scrittura
- Canale: 1/Lettura
- Canale Pubblico / Canale Privato: 1/Lettura

Costo totale: 9

Accessi/giorno: 93.960

IM2 – Invio messaggio con riferimento

- Impiegato: 1/Lettura
- Invio: 1/Scrittura
- Messaggio: 1/Scrittura
- Destinazione: 1/Scrittura
- Canale: 1/Lettura
- Canale Pubblico / Canale Privato: 1/Lettura
- Riferimento: 1/Scrittura
- Messaggio: 1/Lettura

Costo totale: 12

Accessi/giorno: 83.520

IM3 – Visualizzazione dei canali di cui un impiegato fa parte

- Impiegato: 1/Lettura
- Appartiene: 12/Lettura
- Canale: 12/Lettura
- Canale Pubblico: 3/Lettura
- Canale Privato: 9/Lettura

Costo totale: 37

Accessi/giorno: 214.600

IM4 – Visualizzazione dei membri di un canale

- Canale: 1/Lettura
- Canale Pubblico: 1/Lettura
- Appartiene: 12/Lettura
- Impiegato: 12/Lettura

Costo totale: 26

Accessi/anno: 36.400

IM5 – Visualizzazione dei messaggi di un canale di cui un impiegato fa parte

- Impiegato: 1/Lettura
- Appartiene: 1/Lettura
- Canale: 1/Lettura
- Canale Pubblico/ Canale Privato: 1/Lettura
- Destinazione: 1.000/Lettura
- Messaggio: 1.000/Lettura

Costo totale: 2.004

Accessi/giorno: 34.869.600

L1 – Login

- Impiegato: 1/Lettura
- Amministratore/Dipendente Base/Capo Progetto: 1/Lettura

Costo totale: 2

Accessi/giorno: 8.400

Ristrutturazione dello schema E-R

Esaminando lo schema E-R illustrato in figura 4, non si riscontrano ridondanze.

Per la generalizzazione dell'entità IMPIEGATO, si è scelto di unire le entità AMMINISTRATORE, DIPENDENTE BASE e CAPO PROGETTO nell'entità padre IMPIEGATO. Di conseguenza, è stato necessario aggiungere l'attributo *Ruolo* a questa entità per differenziare i vari tipi di impiegati che interagiscono con il sistema.

Per la generalizzazione dell'entità CANALE, si è scelto di unire le entità CANALE PUBBLICO e CANALE PRIVATO nell'entità padre CANALE. Di conseguenza, è stato necessario aggiungere l'attributo *Tipo* a questa entità per differenziare tra i canali pubblici e quelli privati.

Queste scelte consentono di ridurre il numero di accessi per le operazioni AM1, AM2, CP1, CP2, CP3, CP4, CP5, CP6, CP7, CP8, IM1, IM2, IM3, IM4, IM5 e L1, con un costo rispettivo di 60/anno, 1.392/anno, 1.440/anno, 17.280/anno, 12.150/anno, 6.960/anno, 17.400/anno, 21.750/anno, 43.500/anno, 109.000/anno, 83.520/giorno, 76.560/anno, 145.000/giorno, 35.000/anno, 34.852.200/giorno, 4200/giorno.

Trasformazione di attributi e identificatori

Per quanto riguarda gli identificatori esterni, essi sono presenti nelle entità MESSAGGIO e CANALE. Per l'entità MESSAGGIO, l'identificazione avviene attraverso gli attributi *Data Invio* e *Ora Invio* e l'entità Impiegato. Questa identificazione risulta essere pesante dato che questi attributi devono essere utilizzati per rappresentare le associazioni Invio e Destinazione, le quali presentano un numero elevato di occorrenze. Di conseguenza, si è deciso di introdurre l'identificatore *CodiceMSG* per ridurre l'uso di memoria.

Per l'entità CANALE, l'identificazione avviene attraverso l'attributo *Nome* e l'entità Progetto. Anche in questo caso, l'identificatore risulta pesante, poiché deve rappresentare associazioni con molte occorrenze. È quindi opportuno introdurre un nuovo identificatore denominato *CodiceCh*.

Lo schema E-R che ne consegue è rappresentato in figura 5.

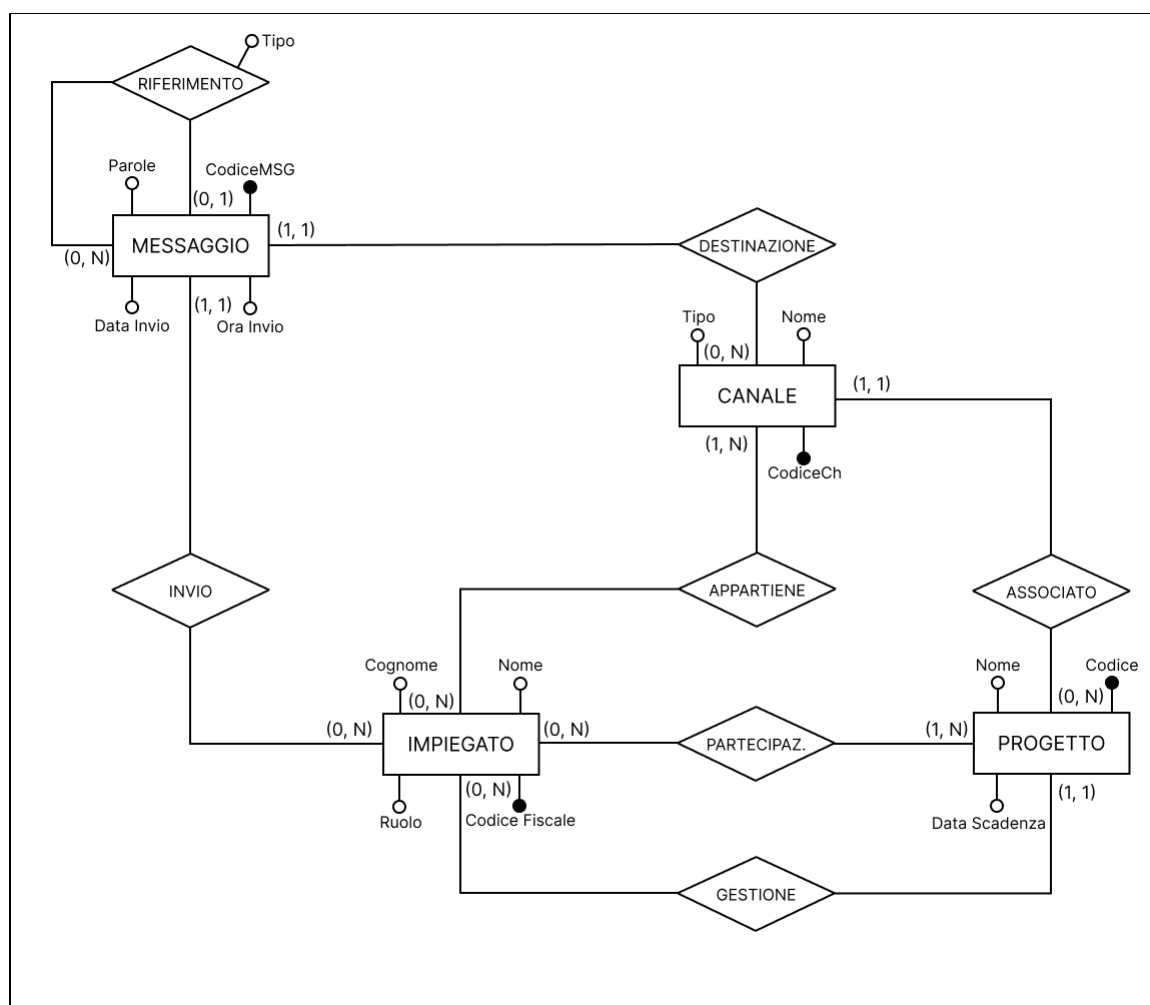


Figura 5: Schema E-R ristrutturato

Traduzione di entità e associazioni

Impiegato (CodiceFiscale, Nome, Cognome, Ruolo)

Progetto (Codice, Nome, DataScadenza, CapoProgetto)

con i seguenti vincoli di integrazione:

Progetto (CapoProgetto) \subseteq Impiegato (CodiceFiscale)

Canale (CodiceCh, Nome, Tipo, Progetto)

con i seguenti vincoli di integrazione:

Canale (Progetto) \subseteq Progetto (Codice)

Messaggio (CodiceMSG, Parole, DataInvio, OraInvio, Mittente, Destinatario)

con i seguenti vincoli di integrazione:

Messaggio (Mittente) \subseteq Impiegato (CodiceFiscale)

Messaggio (Destinatario) \subseteq Canale (CodiceCh)

Riferimento (CodiceMSGMittente, CodiceMSGRiferito, Tipo)

con i seguenti vincoli di integrazione:

Riferimento (CodiceMSGMittente) \subseteq Messaggio (CodiceMSG)

Riferimento (CodiceMSGRiferito) \subseteq Messaggio (CodiceMSG)

Appartiene (Impiegato, Canale)

con i seguenti vincoli di integrazione:

Appartiene (Impiegato) \subseteq Impiegato (CodiceFiscale)

Appartiene (Canale) \subseteq Canale (CodiceCh)

Partecipazione (Impiegato, Progetto)

con i seguenti vincoli di integrazione:

Partecipazione (Impiegato) \subseteq Impiegato (CodiceFiscale)

Partecipazione (Progetto) \subseteq Progetto (Codice)

Lo schema logico che ne consegue è rappresentato in figura 6.

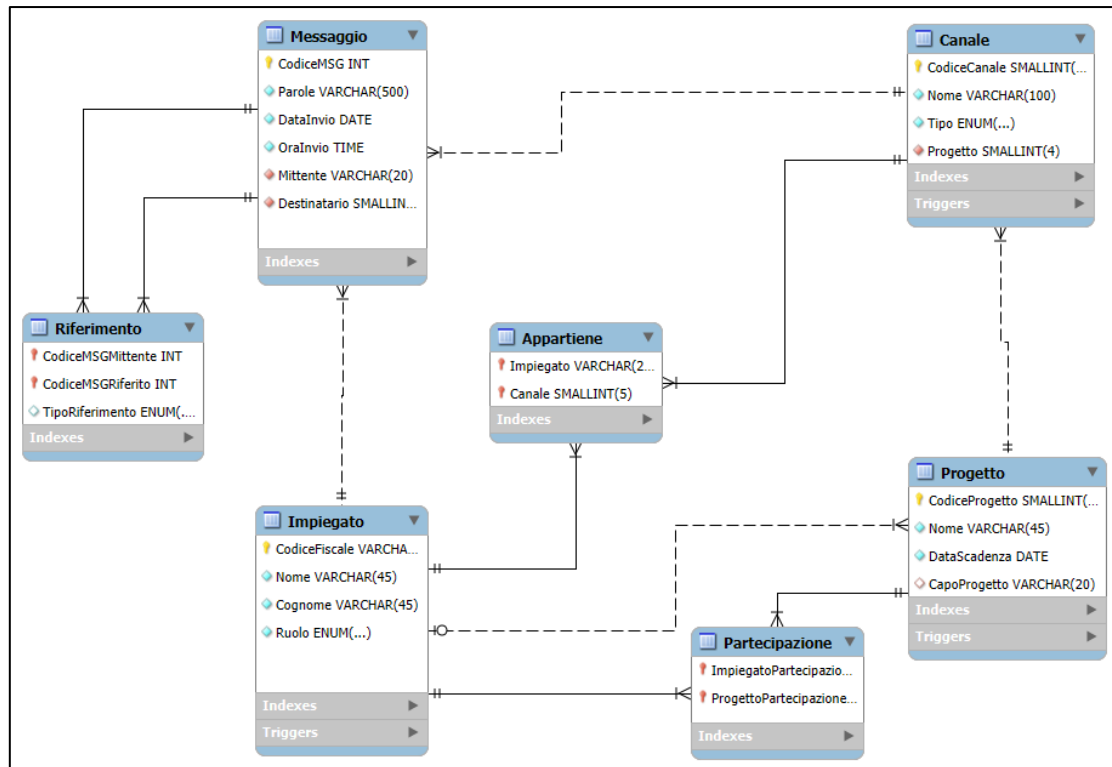


Figura 6: Schema logico

Normalizzazione del modello relazionale

Il modello relazionale è in terza forma normale dato che, per ogni relazione, ogni dipendenza funzionale (non banale) soddisfa almeno una delle seguenti condizioni:

- X contiene una chiave K della relazione
- A appartiene ad almeno una chiave della relazione

5. Progettazione fisica

Utenti e privilegi

Si prevedono quattro ruoli, per implementare il Principle of Least Privilege:

- Login
 - Grant in esecuzione sull'operazione L1
- DipendenteBase
 - Grant in esecuzione sulle operazioni IM1, IM2, IM3, IM4, IM5
- CapoProgetto
 - Grant in esecuzione sulle operazioni CP1, CP2, CP3, CP4, CP5, CP6, CP7, CP8, IM1, IM2, IM3, IM4, IM5, AM4
- Amministratore
 - Grant in esecuzione sulle operazioni AM1, AM2, AM3, AM4

Per identificare gli utenti si introduce una tabella *Utenti* per mantenerne le credenziali. Questa è costituita dagli attributi *Username*, che svolge la funzione di identificatore, *Password* e *Ruolo* per differenziare i diversi tipi di lavoratori presenti nell'azienda.

Lo schema che ne consegue è rappresentato in Figura 7.

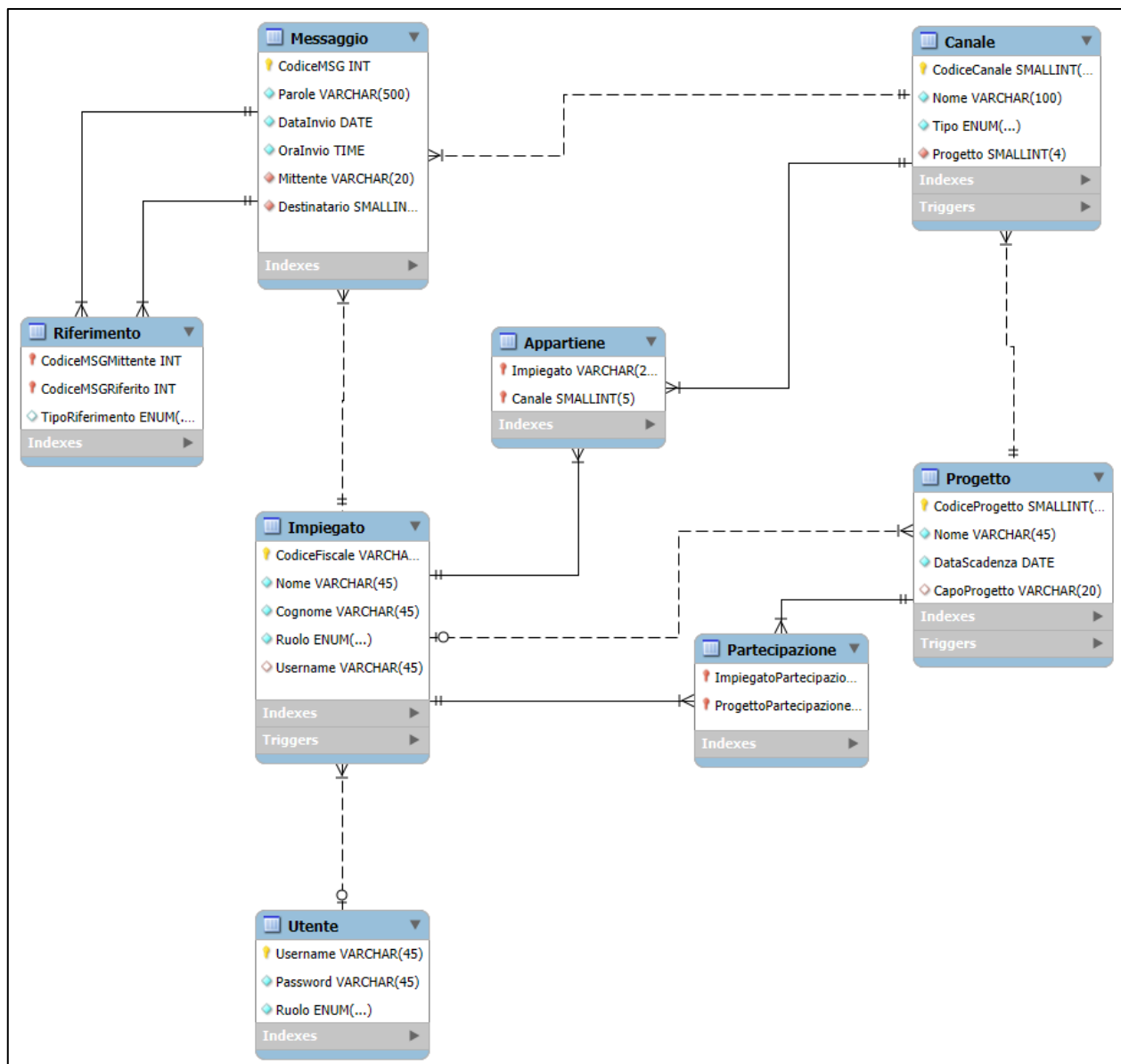


Figura 7: Schema logico finale

Strutture di memorizzazione

Tabella Messaggio		
Colonna	Tipo di dato	Attributi ²
CodiceMSG	INT	PK, NN, AI
Parole	VARCHAR(500)	NN
DataInvio	DATE	NN
OraInvio	TIME	NN
Mittente	VARCHAR(20)	NN
Destinatario	SMALLINT(4)	NN

² PK = primary key, NN = not null, UQ = unique, UN = unsigned, AI = auto increment. È ovviamente possibile specificare più di un attributo per ciascuna colonna.

Tabella Riferimento		
Colonna	Tipo di dato	Attributi
CodiceMSGMittente	INT	PK, NN
CodiceMSGRiferito	INT	PK, NN
TipoRiferimento	ENUM('Pubblico', 'Privato')	NN

Tabella Impiegato		
Colonna	Tipo di dato	Attributi
CodiceFiscale	VARCHAR(20)	PK, NN
Nome	VARCHAR(45)	NN
Cognome	VARCHAR(45)	NN
Ruolo	ENUM('Amministratore', 'Capo progetto', 'Dipendente base')	NN
Username	VARCHAR(45)	

Tabella Canale		
Colonna	Tipo di dato	Attributi
CodiceCanale	SMALLINT(5)	PK, NN, AI
Nome	VARCHAR(45)	NN
Tipo	ENUM('Pubblico', 'Privato')	NN
Progetto	SMALLINT(4)	

Tabella Progetto		
Colonna	Tipo di dato	Attributi
CodiceProgetto	SMALLINT(4)	PK, NN, AI
Nome	VARCHAR(45)	NN
Data	DATE	NN
CapoProgetto	VARCHAR(20)	

Tabella Appartiene		
Colonna	Tipo di dato	Attributi
Impiegato	VARCHAR(20)	PK, NN
Canale	SMALLINT(5)	PK, NN

Tabella Partecipazione		
Colonna	Tipo di dato	Attributi
ImpiegatoPartecipazione	VARCHAR(20)	PK, NN
ProgettoPartecipazione	SMALLINT(4)	PK, NN

Tabella Utente		
Colonna	Tipo di dato	Attributi
Username	VARCHAR(45)	PK, NN
Password	VARCHAR(45)	NN
Ruolo	ENUM('Amministratore', 'Capo progetto', 'Dipendente base')	NN

Indici

Per evitare che vengano assegnati più capi progetto ad uno stesso progetto si è introdotto l'indice `unique_capo`.

Tabella Progetto	
Indice <code>unique_capo</code>	Tipo ³ :
CodiceProgetto, CapoProgetto	UQ

Per evitare che più impiegati usino lo stesso username si è introdotto l'indice `unique_user`.

Tabella Impiegato	
Indice <code>unique_user</code>	Tipo ³ :
CodiceFiscale, UsernameImpiegato	UQ

Trigger

Il seguente trigger è stato implementato per controllare che il nome e il cognome degli impiegati non contengano numeri e/o caratteri speciali. Inoltre, verifica la correttezza del codice fiscale.

```
CREATE DEFINER = CURRENT_USER TRIGGER
`ChatMultiCanale`.`Impiegato_BEFORE_INSERT` BEFORE INSERT ON `Impiegato` FOR
EACH ROW
BEGIN
    IF NOT NEW.NomeImpiegato REGEXP '^[a-zA-Z]+$' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Il nome contiene numeri e/o caratteri speciali.';
    ELSEIF NEW.CognomeImpiegato REGEXP '^[a-zA-Z]+$' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Il cognome contiene numeri e/o caratteri speciali.';
    ELSEIF NOT NEW.CF REGEXP '^[A-Z]{6}[0-9]{2}[A-Z][0-9]{2}[A-Z][0-9]{3}[A-Z] $' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Il codice fiscale non è valido.';
    END IF;
END
```

³ IDX = index, UQ = unique, FT = full text, PR = primary.

Il seguente trigger è stato implementato per controllare che non vi siano progetti con nomi identici. Inoltre, controlla anche che la data di scadenza di un progetto sia valida.

```
CREATE DEFINER = CURRENT_USER TRIGGER
`ChatMultiCanale`.`Progetto_BEFORE_INSERT` BEFORE INSERT ON `Progetto` FOR
EACH ROW
BEGIN
    IF EXISTS (SELECT 1 FROM Progetto WHERE NomeProgetto =
NEW.NomeProgetto) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Nome progetto già esistente';
    ELSEIF NEW.DataScadenza <= CURDATE() THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La data di scadenza deve essere successiva alla data
odierna';
    END IF;
END
```

Il seguente trigger è stato implementato per controllare che non vi siano canali con nomi identici.

```
CREATE DEFINER = CURRENT_USER TRIGGER
`ChatMultiCanale`.`Canale_BEFORE_INSERT` BEFORE INSERT ON `Canale` FOR
EACH ROW
BEGIN
    IF EXISTS (SELECT 1 FROM Canale WHERE NomeCanale = NEW.NomeCanale)
    THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Nome canale già esistente';
    END IF;
END
```


Il seguente trigger è stato implementato per controllare la lunghezza di un messaggio e verificare che il destinatario sia diverso dal mittente.

```
CREATE DEFINER = CURRENT_USER TRIGGER
`ChatMultiCanale`.`Messaggio_BEFORE_INSERT` BEFORE INSERT ON `Messaggio`
FOR EACH ROW
BEGIN
    IF LENGTH(NEW.Parole) > 500 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Messaggio troppo
        lungo';
    END IF;
END
```

Eventi

I seguenti due eventi sono stati implementati per mantenere quanto più pulito possibile il database da messaggi e canali obsoleti in quanto relativi a progetti scaduti da oltre 12 mesi. Questo approccio favorisce una gestione dei dati efficiente e ottimizzata nel tempo, evitando l'accumulo di informazioni obsolete. Gli eventi vengono eseguiti due volte al mese, a partire dai 12 mesi successivi all'inserimento dell'evento nel database, in quanto nei primi 12 mesi non vi saranno canali e messaggi relativi a progetti scaduti da oltre un anno.

```
SET GLOBAL event_scheduler = ON;
```

```
DELIMITER //
```

```
CREATE EVENT IF NOT EXISTS rimozione_messaggi_vecchi
ON SCHEDULE EVERY 2 WEEK
STARTS CURRENT_TIMESTAMP + INTERVAL 12 MONTH
ON COMPLETION PRESERVE
COMMENT 'Rimozione dei messaggi relativi a canali associati a progetti scaduti da più di 12 mesi'
DO
BEGIN
    DELETE FROM Messaggio
    WHERE Destinatario IN (
        SELECT CodiceCanale
```

```
FROM Canale
WHERE Progetto IN (
    SELECT CodiceProgetto
    FROM Progetto
    WHERE DataScadenza < DATE_SUB(CURDATE(), INTERVAL 12
    MONTH)
)
);
END
//
```

```
CREATE EVENT IF NOT EXISTS rimozione_canali_vecchi
ON SCHEDULE EVERY 2 WEEK
STARTS CURRENT_TIMESTAMP
ON COMPLETION PRESERVE
COMMENT 'Rimozione dei canali relativi a progetti scaduti da più di 12 mesi'
DO
BEGIN
    DELETE FROM Canale
    WHERE Progetto IN (
        SELECT CodiceProgetto
        FROM Progetto
        WHERE DataScadenza < DATE_SUB(CURDATE(), INTERVAL 12 MONTH)
    );
END
//
```

Viste

Non è stato fatto uso di viste.

Stored Procedures e transazioni

Operazione AM1 – Inserimento impiegato

```
CREATE PROCEDURE `nuovo_impiegato` (in var_impiegato VARCHAR(20), in var_nome
VARCHAR(45), in var_cognome VARCHAR(45), in var_ruolo ENUM('Amministratore', 'Capo
progetto', 'Dipendente base'), in var_username VARCHAR(45), in var_password VARCHAR(45))
BEGIN
    declare exit handler for sqlexception
    begin
        rollback;
        resignal;
    end;

    set transaction isolation level read uncommitted;
    start transaction;

    insert into Utente(Username, Password, Ruolo) values (var_username, md5(var_password),
var_ruolo);
    insert into Impiegato(CodiceFiscale, Nome, Cognome, Ruolo, Username) values
(var_impiegato, var_nome, var_cognome, var_ruolo, var_username);

    commit;
END
```

Operazione AM2 – Assegnazione capo progetto a progetto

```
CREATE PROCEDURE `assegna_capo` (in var_capo_progetto VARCHAR(20), in
var_codice_progetto SMALLINT(5))
BEGIN
    declare exit handler for sqlexception
    begin
        rollback;
        resignal;
    end;

    set transaction isolation level read committed;
    start transaction;

    if exists(
        select 1
        from Impiegato
        where CodiceFiscale = var_capo_progetto and Ruolo = 'Capo progetto'
    ) then
        update Progetto
```

```
        set CapoProgetto = var_capo_progetto
        where CodiceProgetto = var_codice_progetto;
        insert into Partecipazione (ImpiegatoPartecipazione, ProgettoPartecipazione) values
        (var_capo_progetto, var_codice_progetto);
    else
        signal sqlstate '45000'
        set message_text = 'L\'impiegato non esiste o non ha il ruolo di capo progetto';
    end if;

    commit;
END
```

Operazione AM3 – Creazione Progetto

```
CREATE PROCEDURE `crea_progetto` (in var_nome_progetto VARCHAR(45), in
var_data_scadenza DATE)
BEGIN
    declare exit handler for sqlexception
    begin
        rollback;
        resignal;
    end;

    set transaction isolation level read committed;
    start transaction;

    insert into Progetto(Nome, DataScadenza, CapoProgetto) values (var_nome_progetto,
var_data_scadenza, NULL);
    commit;
END
```

Operazione AM4 – Visualizza Impiegati

```
CREATE PROCEDURE `visualizza_impiegati` ()
BEGIN

    declare exit handler for sqlexception
    begin
        rollback;
        resignal;
    end;
```

```
set transaction isolation level read committed;  
set transaction read only;  
start transaction;
```

```
select CodiceFiscale, Nome, Cognome, Ruolo  
from Impiegato  
order by Ruolo ASC;
```

```
commit;
```

```
END
```

Operazione CP1 – Creazione canale pubblico

```
CREATE PROCEDURE `crea_canale_pubblico` (in var_nome VARCHAR(45), in  
var_codice_progetto SMALLINT(4))  
BEGIN  
    declare var_progetto INT UNSIGNED;  
    declare var_capo_progetto VARCHAR(20);  
    declare exit handler for sqlexception  
    begin  
        rollback;  
        resignal;  
    end;  
  
    set transaction isolation level read committed;  
    start transaction;  
  
    select count(*) into var_progetto  
    from Progetto  
    where CodiceProgetto = var_codice_progetto;  
  
    if var_progetto = 0 then  
        signal sqlstate '45000'  
        set message_text = 'Impossibile creare il canale poichè il progetto non esiste';  
    else  
        select CapoProgetto into var_capo_progetto  
        from Progetto  
        where CodiceProgetto = var_codice_progetto;  
  
        insert into Canale(Nome, Tipo, Progetto) values (var_nome, 'Pubblico',  
var_codice_progetto);
```

```
        insert into Appartiene(Impiegato, Canale) values (var_capo_progetto,
        last_insert_id());

        commit;
    end if;
END
```

Operazione CP2 – Inserimento dipendente base in un canale

```
CREATE PROCEDURE `aggiungi_al_canale` (in var_codice_fiscale VARCHAR(20), in
var_codice_canale SMALLINT(5))
BEGIN
    declare var_controllo INT UNSIGNED;
    declare exit handler for sqlexception
    begin
        rollback;
        resignal;
    end;

    set transaction isolation level read committed;
    start transaction;

    select count(*) into var_controllo
    from Impiegato
    join Partecipazione on Impiegato.CodiceFiscale = Partecipazione.ImpiegatoPartecipazione
    join Canale on Partecipazione.ProgettoPartecipazione = Canale.Progetto
    where Impiegato.CodiceFiscale = var_codice_fiscale
    and Canale.CodiceCanale = var_codice_canale;

    if var_controllo = 0 then
        signal sqlstate '45000'
        set message_text = 'Errore: Il dipendente non esiste, o non appartiene al progetto
        associato al canale';
    end if;

    insert into Appartiene (Impiegato, Canale) values (var_codice_fiscale, var_codice_canale);

    commit;
END
```

Operazione CP3 – Visualizzazione dei messaggi in un canale privato

```
CREATE PROCEDURE `leggi_msg` (in var_codice_canale SMALLINT(5), in var_CF_impiegato
VARCHAR(20))
BEGIN
    declare var_membro INT;
    declare var_capo_progetto INT;

    declare exit handler for sqlexception
    begin
        rollback;
        resignal;
    end;

    set transaction isolation level read committed;
    start transaction;

    select count(*) into var_membro
    from Appartiene
    where Canale = var_codice_canale and Impiegato= var_CF_impiegato;

    select count(*) into var_capo_progetto
    from Canale join Progetto on Canale.Progetto = Progetto.CodiceProgetto
    where Canale.CodiceCanale = var_codice_canale and Progetto.CapoProgetto =
    var_CF_impiegato;

    if var_membro = 0 and var_capo_progetto = 0 then
        signal sqlstate '45000'
        set message_text = 'Non è possibile visualizzare i messaggi di un canale di cui non si
        fa parte';
    end if;

    select Messaggio.DataInvio, Messaggio.OraInvio, Messaggio.CodiceMSG as
    CodiceMessaggio, concat(Impiegato.Nome, ' ', Impiegato.Cognome) AS Mittente,
    Messaggio.Parole as Testo,
        (select Parole
        from Messaggio
        where Messaggio.CodiceMSG = Riferimento.CodiceMSGRiferito
        ) as RispostaA
    from Messaggio
    left join Riferimento on Messaggio.CodiceMSG = Riferimento.CodiceMSGMittente
    join Impiegato on Messaggio.Mittente = Impiegato.CodiceFiscale
    where Messaggio.Destinatario = var_codice_canale
    order by Messaggio.DataInvio ASC, Messaggio.OraInvio ASC;

    commit;
```

END

Operazione CP4 – Inserimento dipendente base in un progetto

```
CREATE PROCEDURE `aggiungi_al_progetto` (in var_codice_fiscale VARCHAR(20), in  
var_codice_progetto SMALLINT(5))
```

```
BEGIN
```

```
    declare exit handler for sqlexception
```

```
    begin
```

```
        rollback;
```

```
        resignal;
```

```
    end;
```

```
    set transaction isolation level read committed;
```

```
    start transaction;
```

```
    if not exists(
```

```
        select 1
```

```
        from Impiegato
```

```
        where CodiceFiscale = var_codice_fiscale
```

```
    ) then
```

```
        signal sqlstate '45000'
```

```
        set message_text = 'Dipendente non trovato';
```

```
    end if;
```

```
    if not exists(
```

```
        select 1
```

```
        from Progetto
```

```
        where CodiceProgetto = var_codice_progetto
```

```
    ) then
```

```
        signal sqlstate '45000'
```

```
        set message_text = 'Progetto non trovato';
```

```
    end if;
```

```
    if exists (
```

```
        select 1
```

```
        from Partecipazione
```

```
        where ImpiegatoPartecipazione = var_codice_fiscale
```

```
        and ProgettoPartecipazione = var_codice_progetto
```

```
    ) then
```

```
        signal sqlstate '45000'
```

```
        set message_text = 'Dipendente già assegnato al progetto';
```

```
    end if;
```



```
insert into Partecipazione (ImpiegatoPartecipazione, ProgettoPartecipazione) values  
(var_codice_fiscale, var_codice_progetto);
```

```
commit;
```

```
END
```

Operazione CP5 – Visualizza progetto di cui si è responsabile

```
CREATE PROCEDURE `visualizza_progetti` (in var_codice_fiscale VARCHAR(20))
```

```
BEGIN
```

```
    declare var_ruolo VARCHAR(20);  
    declare exit handler for sqlexception  
    begin  
        rollback;  
        resignal;  
    end;
```

```
    set transaction isolation level read committed;  
    set transaction read only;  
    start transaction;
```

```
    select Ruolo into var_ruolo  
    from Impiegato  
    where CodiceFiscale = var_codice_fiscale;
```

```
    if var_ruolo = 'Amministratore' then  
        select CodiceProgetto, Nome as NomeProgetto, DataScadenza, CapoProgetto  
        from Progetto  
        order by CodiceProgetto;  
    else  
        select CodiceProgetto, Nome as NomeProgetto, DataScadenza, CapoProgetto  
        from Progetto  
        where CapoProgetto = var_codice_fiscale  
        order by CodiceProgetto;  
    end if;
```

```
commit;
```

```
END
```

Operazione CP6 – Visualizza membri di un progetto

```
CREATE PROCEDURE `visualizza_membri_progetto` (in var_codice_progetto SMALLINT(5))
BEGIN
    declare exit handler for sqlexception
    begin
        rollback;
        resignal;
    end;

    set transaction isolation level read committed;
    set transaction read only;
    start transaction;

    select Progetto.Nome as NomeProgetto, Impiegato.CodiceFiscale, Impiegato.Nome as
    NomeImpiegato, Impiegato.Cognome as CognomeImpiegato, Impiegato.Ruolo
    from Impiegato
    join Partecipazione on Impiegato.CodiceFiscale = Partecipazione.ImpiegatoPartecipazione
    join Progetto on Partecipazione.ProgettoPartecipazione = Progetto.CodiceProgetto
    where Partecipazione.ProgettoPartecipazione = var_codice_progetto;

    commit;
END
```

Operazione CP7 – Visualizza assenti nel canale

```
CREATE PROCEDURE `visualizza_assenti_canale` (in var_codice_canale SMALLINT(5))
BEGIN
    declare exit handler for sqlexception
    begin
        rollback;
        resignal;
    end;

    set transaction isolation level read committed;
    set transaction read only;
    start transaction;

    select Progetto.Nome as NomeProgetto, Impiegato.CodiceFiscale, Impiegato.Nome,
    Impiegato.Cognome, Impiegato.Ruolo
    from Impiegato
    join Partecipazione on Impiegato.CodiceFiscale = Partecipazione.ImpiegatoPartecipazione
    join Canale on Canale.Progetto = Partecipazione.ProgettoPartecipazione
```

```
join Progetto on Progetto.CodiceProgetto = Canale.Progetto
where Canale.CodiceCanale = var_codice_canale
and Impiegato.CodiceFiscale not in
    (select Appartiene.Impiegato
     from Appartiene
     where Appartiene.Canale = var_codice_canale);

commit;

END
```

Operazione CP8 – Visualizza assenti nel progetto

```
CREATE PROCEDURE `visualizza_assenti_progetto` (in var_codice_progetto SMALLINT(5))
BEGIN
    declare exit handler for sqlexception
    begin
        rollback;
        resignal;
    end;

    set transaction isolation level read committed;
    set transaction read only;
    start transaction;

    select Impiegato.CodiceFiscale, Impiegato.Nome, Impiegato.Cognome, Impiegato.Ruolo
    from Impiegato
    where Impiegato.Ruolo = 'Dipendente base'
    and Impiegato.CodiceFiscale not in
        (select Partecipazione.ImpiegatoPartecipazione
         from Partecipazione
         where Partecipazione.ProgettoPartecipazione = var_codice_progetto);

    commit;

END
```

Operazione IM1 – Invio messaggio senza riferimento

```
CREATE PROCEDURE `invia_msg` (in var_mittente VARCHAR(20), in var_destinatario
SMALLINT(5), in var_parole VARCHAR(500))
BEGIN
    declare var_membro INT;
    declare exit handler for sqlexception
```

```
begin
    rollback;
    resignal;
end;

set transaction isolation level read committed;
start transaction;

select count(*) into var_membro
from Appartiene
where Canale = var_destinatario and Impiegato = var_mittente;

if var_membro = 0 then
    signal sqlstate '45000'
end if;

insert into Messaggio(Parole, DataInvio, OraInvio, Mittente, Destinatario) values
(var_parole, curdate(), curtime(), var_mittente, var_destinatario);

commit;
END
```

Operazione IM2 – Invio messaggio con riferimento

```
CREATE PROCEDURE `rispondi_a_msg` (in var_codice_msg_riferito INT, in var_CF_mittente
VARCHAR(20), in var_testo VARCHAR(500), in var_tipo ENUM('Pubblico','Privato'))
BEGIN
    declare var_destinatario_originale SMALLINT(5);
    declare var_mittente_originale VARCHAR(20);
    declare var_codice_canale_privato SMALLINT(5);
    declare var_codice_nuovo_canale SMALLINT(5);
    declare var_nome_canale_privato VARCHAR(45);
    declare var_codice_progetto SMALLINT(5);
    declare var_nome_mittente VARCHAR(45);
    declare var_cognome_mittente VARCHAR(45);
    declare var_nome_destinatario VARCHAR(45);
    declare var_cognome_destinatario VARCHAR(45);

    declare exit handler for sqlexception
    begin
        rollback;
        resignal;
    end;
```

```
set transaction isolation level read committed;
start transaction;

select Mittente, Destinatario into var_mittente_originale, var_destinatario_originale
from Messaggio
where CodiceMSG = var_codice_msg_riferito;

select Progetto into var_codice_progetto
from Canale
where CodiceCanale = var_destinatario_originale;

if var_codice_progetto is NULL then
    signal sqlstate '45000'
    set message_text = 'Impossibile trovare il progetto per il canale.';
end if;

if var_tipo = 'Pubblico' then
    insert into Messaggio(Parole, DataInvio, OraInvio, Mittente, Destinatario) values
    (var_testo, curdate(), curtime(), var_CF_mittente, var_destinatario_originale);
    insert into Riferimento(CodiceMSGMittente, CodiceMSGRiferito, TipoRiferimento)
    values (last_insert_id(), var_codice_msg_riferito, 'Pubblico');
elseif var_tipo = 'Privato' then
    select Canale.CodiceCanale into var_codice_canale_privato
    from Canale
    join Appartiene as A1 on Canale.CodiceCanale= A1.Canale
    join Appartiene as A2 on Canale.CodiceCanale = A2.Canale
    where Canale.Tipo = 'Privato' and ((A1.Impiegato = var_CF_mittente and
    A2.Impiegato = var_mittente_originale) or (A1.Impiegato = var_mittente_originale
    and A2.Impiegato = var_CF_mittente));

if var_codice_canale_privato is not null then
    insert into Messaggio(Parole, DataInvio, OraInvio, Mittente, Destinatario) values
    (var_testo, curdate(), curtime(), var_CF_mittente, var_codice_canale_privato);
    insert into Riferimento(CodiceMSGMittente, CodiceMSGRiferito, TipoRiferimento)
    values (last_insert_id(), var_codice_msg_riferito, 'Privato');
else
    select Nome, Cognome into var_nome_mittente, var_cognome_mittente
    from Impiegato
    where CodiceFiscale = var_CF_mittente;

    select Nome, Cognome into var_nome_destinatario, var_cognome_destinatario
    from Impiegato
    where CodiceFiscale = var_mittente_originale;
```

```
set var_nome_canale_privato = concat_ws('_', var_nome_mittente,
var_cognome_mittente, var_nome_destinatario, var_cognome_destinatario, 'privato');

insert into Canale(Nome, Tipo, Progetto) values (var_nome_canale_privato, 'Privato',
var_codice_progetto);

set var_codice_nuovo_canale = last_insert_id();

insert into Appartiene(Impiegato, Canale) values (var_CF_mittente,
var_codice_nuovo_canale), (var_mittente_originale, var_codice_nuovo_canale);

insert into Messaggio(Parole, DataInvio, OraInvio, Mittente, Destinatario) values
(var_testo, curdate(), curtime(), var_CF_mittente, var_codice_nuovo_canale);

insert into Riferimento(CodiceMSGMittente, CodiceMSGRiferito, TipoRiferimento)
values (last_insert_id(), var_codice_msg_riferito, 'Privato');
end if;
end if;

commit;
END
```

Operazione IM3 – Visualizzazione dei canali di cui un impiegato fa parte

```
CREATE PROCEDURE `visualizza_canali` (in var_CF_impiegato VARCHAR(20))
BEGIN
    declare exit handler for sqlexception
    begin
        rollback;
        resignal;
    end;

    set transaction isolation level read committed;
    set transaction read only;
    start transaction;

    if exists(
        select 1
        from Progetto
        where CapoProgetto = var_CF_impiegato
    )then
```

```
select Canale.CodiceCanale, Canale.Nome as NomeCanale, Canale.Tipo,
Progetto.CodiceProgetto, Progetto.Nome as NomeProgetto
from Canale join Progetto on Canale.Progetto = Progetto.CodiceProgetto
where Progetto.CapoProgetto = var_CF_impiegato or
(Canale.Tipo = 'Privato' and Canale.Progetto in
(select Progetto2.CodiceProgetto
from Progetto as Progetto2
where Progetto2.CapoProgetto = var_CF_impiegato));
else
select Canale.CodiceCanale, Canale.Nome as NomeCanale, Canale.Tipo,
Progetto.CodiceProgetto, Progetto.Nome as NomeProgetto
from Canale
join Progetto on Canale.Progetto = Progetto.CodiceProgetto
join Appartiene on Canale.CodiceCanale = Appartiene.Canale
where Appartiene.Impiegato = var_CF_impiegato;
end if;

commit;
END
```

Operazione IM4 – Visualizzazione dei membri di un canale

```
CREATE PROCEDURE `visualizza_membri_canale` (in var_codice_canale SMALLINT(5))
BEGIN
    declare exit handler for sqlexception
    begin
        rollback;
        resignal;
    end;

    set transaction isolation level read committed;
    set transaction read only;
    start transaction;

    select Canale.Nome as NomeCanale, Impiegato.CodiceFiscale, Impiegato.Nome as
NomeImpiegato, Impiegato.Cognome as CognomeImpiegato, Impiegato.Ruolo
from Impiegato
join Appartiene on Impiegato.CodiceFiscale = Appartiene.Impiegato
join Canale on Appartiene.Canale = Canale.CodiceCanale
where Appartiene.Canale = var_codice_canale;

    commit;
END
```

Operazione IM5 – Visualizzazione dei messaggi di un canale di cui un impiegato fa parte

```
CREATE PROCEDURE `leggi_msg` (in var_codice_canale SMALLINT(5), in var_CF_impiegato
VARCHAR(20))
BEGIN
    declare var_membro INT;
    declare var_capo_progetto INT;

    declare exit handler for sqlexception
    begin
        rollback;
        resignal;
    end;

    set transaction isolation level read committed;
    start transaction;

    select count(*) into var_membro
    from Appartiene
    where Canale = var_codice_canale and Impiegato= var_CF_impiegato;

    select count(*) into var_capo_progetto
    from Canale join Progetto on Canale.Progetto = Progetto.CodiceProgetto
    where Canale.CodiceCanale = var_codice_canale and Progetto.CapoProgetto =
    var_CF_impiegato;

    if var_membro = 0 and var_capo_progetto = 0 then
        signal sqlstate '45000'
        set message_text = 'Non è possibile visualizzare i messaggi di un canale di cui non si
        fa parte';
    end if;

    select Messaggio.DataInvio, Messaggio.OraInvio, Messaggio.CodiceMSG as
    CodiceMessaggio, concat(Impiegato.Nome, ' ', Impiegato.Cognome) AS Mittente,
    Messaggio.Parole as Testo,
        (select Parole
        from Messaggio
        where Messaggio.CodiceMSG = Riferimento.CodiceMSGRiferito
        ) as RispostaA
    from Messaggio
    left join Riferimento on Messaggio.CodiceMSG = Riferimento.CodiceMSGMittente
    join Impiegato on Messaggio.Mittente = Impiegato.CodiceFiscale
```



```
where Messaggio.Destinatario = var_codice_canale  
order by Messaggio.DataInvio ASC, Messaggio.OraInvio ASC;
```

```
commit;
```

```
END
```

Operazione L1 – Login

```
CREATE PROCEDURE `login` (in var_username varchar(45), in var_pass varchar(45), out var_role  
INT, out var_CodiceFiscale varchar(20))
```

```
BEGIN
```

```
    declare var_user_role ENUM('Amministratore','Capo progetto', 'Dipendente base');
```

```
    declare exit handler for sqlexception
```

```
    begin
```

```
        rollback;
```

```
        resignal;
```

```
    end;
```

```
    set transaction isolation level read committed;
```

```
    set transaction read only;
```

```
    start transaction;
```

```
    select Ruolo into var_user_role
```

```
    from Utente
```

```
    where Username = var_username and Password = md5(var_pass);
```

```
    if var_user_role = 'Amministratore' then
```

```
        set var_role = 1;
```

```
        select CodiceFiscale into var_CodiceFiscale
```

```
        from Impiegato
```

```
        where Username = var_username;
```

```
    elseif var_user_role = 'Capo progetto' then
```

```
        set var_role = 2;
```

```
        select CodiceFiscale into var_CodiceFiscale
```

```
        from Impiegato
```

```
        where Username = var_username;
```

```
    elseif var_user_role = 'Dipendente base' then
```

```
        set var_role = 3;
```

```
        select CodiceFiscale into var_CodiceFiscale
```

```
        from Impiegato
```

```
        where Username = var_username;  
    else  
        set var_role = 0;  
    end if;  
    commit;  
END
```