



Università degli Studi di Salerno



Dipartimento di Ingegneria dell'Informazione ed Elettrica e
Matematica Applicata

Corso di Laurea in Ingegneria Informatica

Basi di Dati 2022/2023 Canale A-H

Project Work

Traccia 1 - Digital Marketing: Key Performance Indicator

Gruppo n. **05 – AH**

WP	Cognome e Nome	Matricola	e-mail	Responsabile
1	Garofalo Mariachiara	0612705066	m.garofalo38@studenti.unisa.it	
2	Capaldo Vincenzo	0612705174	v.capaldo7@studenti.unisa.it	X
3	Donnarumma Luca	0612705051	l.donnarumma16@studenti.unisa.it	
4	Esposito Paolo	0612705027	p.esposito57@studenti.unisa.it	

Anno accademico 2022-2023

Sommario

1. Descrizione della realtà di interesse	3
1.1. Analisi della realtà di interesse	3
2. Analisi delle specifiche	5
2.1. Glossario dei termini	5
2.2. Strutturazione dei requisiti in frasi.....	5
2.2.1. Frasi di carattere generale	5
2.2.2. Frasi relative a azienda	5
2.2.3. Frasi relative a campagna	5
2.2.4. Frasi relative a video	6
2.2.5. Frasi relative a utente	6
2.2.6. Frasi relative a interazione.....	6
2.2.7. Frasi relative a KPI.....	6
2.3. Identificazione delle operazioni principali	6
3. Progettazione Concettuale.....	7
3.1. Schema Concettuale.....	7
3.1.1. Note sullo schema E-R	8
3.2. Design Pattern	8
3.2.1. Pattern Reificazione di relazione binaria	8
3.2.2. Pattern Part-of	8
3.2.3. Pattern Storizzazione di un'entità	9
3.3. Dizionario dei Dati	10
3.4. Regole Aziendali	12
4. Progettazione Logica	13
Ristrutturazione Schema Concettuale	13
4.1.1. Analisi delle Prestazioni	13
4.2. Analisi delle ridondanze	14
4.2.1. Analisi della ridondanza 1: Numero campagne	14
4.2.2. Analisi della ridondanza 2: Numero visualizzazioni	15
4.3. Eliminazione delle generalizzazioni	17
4.3.1. Generalizzazione <i>Campagna</i>	17
4.3.2. Generalizzazione <i>Video</i>	17
4.3.1. Generalizzazione <i>KPI</i>	18
4.4. Partizionamento/Accorpamento Entità e Associazioni	18
4.5. Scelta degli identificatori principali.....	18
4.6. Schema ristrutturato finale	20
4.7. Schema logico	21
4.8. Documentazione dello schema logico	21
5. Normalizzazione	22
6. Script Creazione e Popolamento Database	25
7. Query SQL.....	34

7.1.	Query con operatore di aggregazione e join: Azienda-visualizzazioni_totali-tempo_medio_di_visualizzazione	34
7.2.	Query nidificata complessa: aziende associate ai tre utenti con più interazioni	34
7.3.	Query insiemistica: Video musicali con almeno 3 visualizzazioni	35
7.4.	Altre query	35
7.4.1.	Operazione 7: tre aziende con maggior successo	35
8.	Viste	36
8.1.	Vista riepilogo_video	36
8.1.1.	Query con Vista: video con numero di visualizzazioni maggiore della media	37
8.2.	Vista monitoraggio_campagne	37
8.2.1.	Query con Vista: Costo per reach	38
9.	Trigger	39
9.1.	Trigger inizializzazione: <i>v_i e proteggi_numero_studenti</i>	39
9.2.	Trigger per vincoli aziendali	40
9.2.1.	Trigger1: verifica_v_i e verifica_update_interazione	40
9.2.2.	Trigger2: cardinalita_tag	43
9.2.3.	Trigger3: cardinalita_azienza	44
9.2.4.	Trigger4: cardinalita_utente	45

1. Descrizione della realtà di interesse

Titolo: Digital Marketing: Key Performance Indicator

Si progetta la base di dati per la gestione di campagne pubblicitarie online sulla piattaforma YouTube, attraverso risorse video: si vuole porre l'attenzione, in particolare, sull'interazione tra video e utente.

- La campagna viene commissionata da un'azienda caratterizzata da partita IVA, nome, sede legale, recapito telefonico, indirizzo e-mail ed una breve descrizione.
- Si vogliono rappresentare le diverse campagne pubblicitarie online memorizzando il nome, la data di inizio, il budget totale, il target di riferimento definito dall'azienda ed una breve descrizione.
- Per ogni campagna si vogliono memorizzare le informazioni relative ai video sponsorizzati sulla piattaforma, come l'URL, il titolo, la descrizione, la durata, i tag, l'estensione, la risoluzione massima del file e il numero di visualizzazioni.
- Ogni video può essere visualizzato da zero o più utenti, di cui si vogliono memorizzare: nickname, indirizzo e-mail, nome, cognome, data di nascita, numero di telefono e genere. Tali valori sono relativi al corrispettivo account YouTube.
- Per ogni visualizzazione sono di particolare interesse le informazioni relative all'interazione tra video e utente, come l'istante della visualizzazione (data e orario), il tempo di visualizzazione, se ha cliccato il tasto "skip", se ha cliccato sul link che riporta al sito allegato, se ha messo like e se ha condiviso il video.
- Per ogni video si memorizzano gli indici chiave di performance (KPI), i quali permettono di analizzare l'interesse generato da un video. Per ognuno di essi si vuole specificare il nome, la formula per calcolarlo ed una breve descrizione.

1.1. Analisi della realtà di interesse

L'obiettivo del progetto è di realizzare un database per memorizzare le informazioni relative alla gestione di una campagna di advertising online, attraverso risorse video.

Ogni azienda può commissionare più campagne e per ogni azienda si vuole tener traccia del numero di campagne commissionate. In media un'azienda commissiona due campagne. Saranno trattate solo campagne commissionate in Italia. Ogni campagna viene commissionata da una e una sola azienda e può promuovere un solo video.

L'utente, registrato sulla piattaforma YouTube con account nominale, può interagire con più video e anche più volte con lo stesso video. Non sono trattati utenti anonimi ed account aziendali.

Per ogni campagna bisogna distinguere se essa è in svolgimento oppure se è terminata. Le campagne prese in esame hanno una durata media di 15 giorni e durante il loro svolgimento bisogna aggiornare i valori dei KPI ogni 12 ore. Poiché ogni 15 giorni si hanno circa 10 nuove commissioni, si stima di avere circa 10 campagne correnti e per ogni video avvengono circa 100 interazioni al giorno. Dunque, sulla base delle informazioni date, si stima di avere, in un anno, in media: 240 campagne (di cui 10 correnti), 120 aziende, 360.000 interazioni totali e 250.000 nuovi utenti (poiché alcuni utenti potrebbero essere già stati inseriti precedentemente).

Quando una campagna termina si vuole memorizzare la sua data di fine e il suo grado di raggiungimento dell'obiettivo, cioè un numero compreso tra 0 e 1, il quale è calcolato da un'applicazione esterna sulla base dei nostri dati.

Oltre alle campagne, in tale progetto si vuole focalizzare l'attenzione sulla rappresentazione delle informazioni relative ai video, con dettagli relativi all'URL, al titolo, alla descrizione, alla durata, ai tag, all'estensione, alla risoluzione massima del file e al numero di visualizzazioni. Inoltre, è richiesta la classificazione dei video basata sulla presenza, o assenza, del tasto "skip". In media un video su due ha la funzione di skip. Se più campagne utilizzano uno stesso video, essi avranno comunque un URL diverso. Il video deve avere durata minima di 5 secondi e massima di 3 minuti.

Inoltre, si vuole analizzare l'interazione tra utenti e video al fine di valutare la capacità di un particolare video di catturare l'interessamento dell'utente. Nello specifico, per ogni interazione, si vuole riportare l'istante della visualizzazione (data e orario), il tempo di visualizzazione, se ha cliccato il tasto "skip", se ha cliccato sul link che riporta al sito allegato, se ha messo like e se ha condiviso il video. La raccolta dei dati avviene tramite un applicativo software che si interfaccia col database. Dai dati raccolti saranno poi calcolati i KPI.

L'obiettivo principale della campagna è monitorare le interazioni tra il video e gli utenti. Si vuole ottenere visibilità e interessamento, tenendo conto anche dell'aspetto economico. Per questi motivi, i KPI relativi al monitoraggio dei video saranno alcuni di natura tecnica e altri di natura economica. In particolare, i KPI presi in considerazione sono:

- Reach (tecnico): indica il numero di utenti che hanno visto il video almeno una volta durante il ciclo di vita della campagna;
- Frequency (tecnico): indica il numero medio di volte che un singolo utente ha visualizzato quel video;
- Visualizzazione completa (tecnico): indica la percentuale di utenti che hanno guardato il video per intero;
- Skip (tecnico): indica il numero di volte che il video è stato saltato dagli utenti;
- Costo per click (economico): indica il costo medio per ogni click generato dal video;
- Costo per like (economico): indica il costo medio per ottenere un like sul tuo video;
- Costo per condivisione (economico): indica il costo medio per ottenere una condivisione del tuo video.

La base di dati garantisce comunque il corretto funzionamento anche nel caso in cui manca qualcuno di questi KPI o nel caso in cui se ne vogliono aggiungere altri.

2. Analisi delle specifiche

Workpackage	Task	Responsabile
WP0	Analisi delle specifiche	Intero Gruppo

2.1. Glossario dei termini

	Termine	Descrizione	Sinonimi	Collegamenti
1	Azienda	Azienda che commissiona la campagna pubblicitaria e fornisce i video da utilizzare.	Società, Compagnia	Campagna
2	Campagna	Campagna pubblicitaria online, effettuata sulla piattaforma YouTube.	Promozione	Azienda, Video
3	Video	Video promozionale utilizzato dalla campagna. È la risorsa utilizzata per diffondere la pubblicità tra gli utenti di YouTube. Appartiene ad una sola campagna.	Clip, Pubblicità	Campagna, Interazione, KPI
4	Utente	Utente registrato sulla piattaforma YouTube e che visualizza i video pubblicitari, interagendo con essi.	Persona	Interazione
5	Interazione	Interazione tra utente e video che descrive una serie di azioni effettuate sul video da parte dell'utente.	Contatto	Utente, Video
6	KPI	KPI (Key Performance Indicator) utili per valutare il successo della campagna.	Indice	Video

Tabella 1. Glossario dei Termini

2.2. Strutturazione dei requisiti in frasi

2.2.1. Frasi di carattere generale

Si vuole realizzare una base di dati per la gestione di campagne pubblicitarie online sulla piattaforma YouTube, attraverso risorse video, in particolare si vuole porre l'attenzione sull'interazione tra il video e l'utente.

2.2.2. Frasi relative a azienda

Per l'azienda è utile definire la partita IVA, il nome, la sede legale, un recapito telefonico, un indirizzo e-mail ed una breve descrizione. Inoltre, si vuole tenere traccia del numero di campagne commissionate.

2.2.3. Frasi relative a campagna

Per le campagne è di interesse il nome, la data di inizio, il budget totale, il target di riferimento definito dall'azienda ed una breve descrizione. Inoltre, bisogna distinguere se la campagna è in svolgimento oppure se è già terminata, in quanto, per ogni campagna terminata, si vuole memorizzare la data di fine e il suo grado di raggiungimento dell'obiettivo.

2.2.4. Frasi relative a video

Per i video si vogliono rappresentare l'URL, il titolo, la descrizione, la durata, i tag, l'estensione, la risoluzione massima del file e il numero di visualizzazioni. Inoltre, è richiesta la classificazione dei video basata sulla presenza, o assenza, del tasto "skip".

2.2.5. Frasi relative a utente

Per gli utenti si vuole indicare il nickname, l'indirizzo e-mail, il nome, il cognome, la data di nascita, il numero di telefono e il genere. Tali valori sono relativi al corrispettivo account YouTube. Non sono trattati utenti anonimi ed account aziendali.

2.2.6. Frasi relative a interazione

Per l'interazione tra video e utente viene monitorato l'istante della visualizzazione (data e orario), il tempo di visualizzazione, se ha cliccato il tasto "skip", se ha cliccato sul link che riporta al sito allegato, se ha messo like e se ha condiviso il video.

2.2.7. Frasi relative a KPI

Per i KPI (Key Performance Indicator) si vuole rappresentare il nome, la formula per calcolarlo ed una breve descrizione. I KPI relativi al monitoraggio dei video saranno alcuni di natura tecnica e altri di natura economica.

2.3. Identificazione delle operazioni principali

Operazione 1: inserire una nuova interazione (operazione da effettuare in media 100 volte al giorno)

Operazione 2: aggiornare i valori dei KPI per ogni video (in media 2 volte al giorno)

Operazione 3: inserire una nuova campagna (in media 10 volte ogni 15 giorni)

Operazione 4: trovare il numero di visualizzazioni di un video (in media 10 volte ogni 15 giorni)

Operazione 5: trovare l'azienda che ha commissionato più campagne (in media 1 volta ogni 6 mesi)

Operazione 6: trovare il numero totale di interazioni sui video di un'azienda (in media 1 volta ogni 6 mesi)

Operazione 7: trovare le 3 aziende le cui campagne, in media, hanno avuto maggior successo (in media 1 volta all'anno)

3. Progettazione Concettuale

Workpackage	Task	Responsabile
WP1	Progettazione Concettuale	Garofalo Mariachiara

3.1. Schema Concettuale

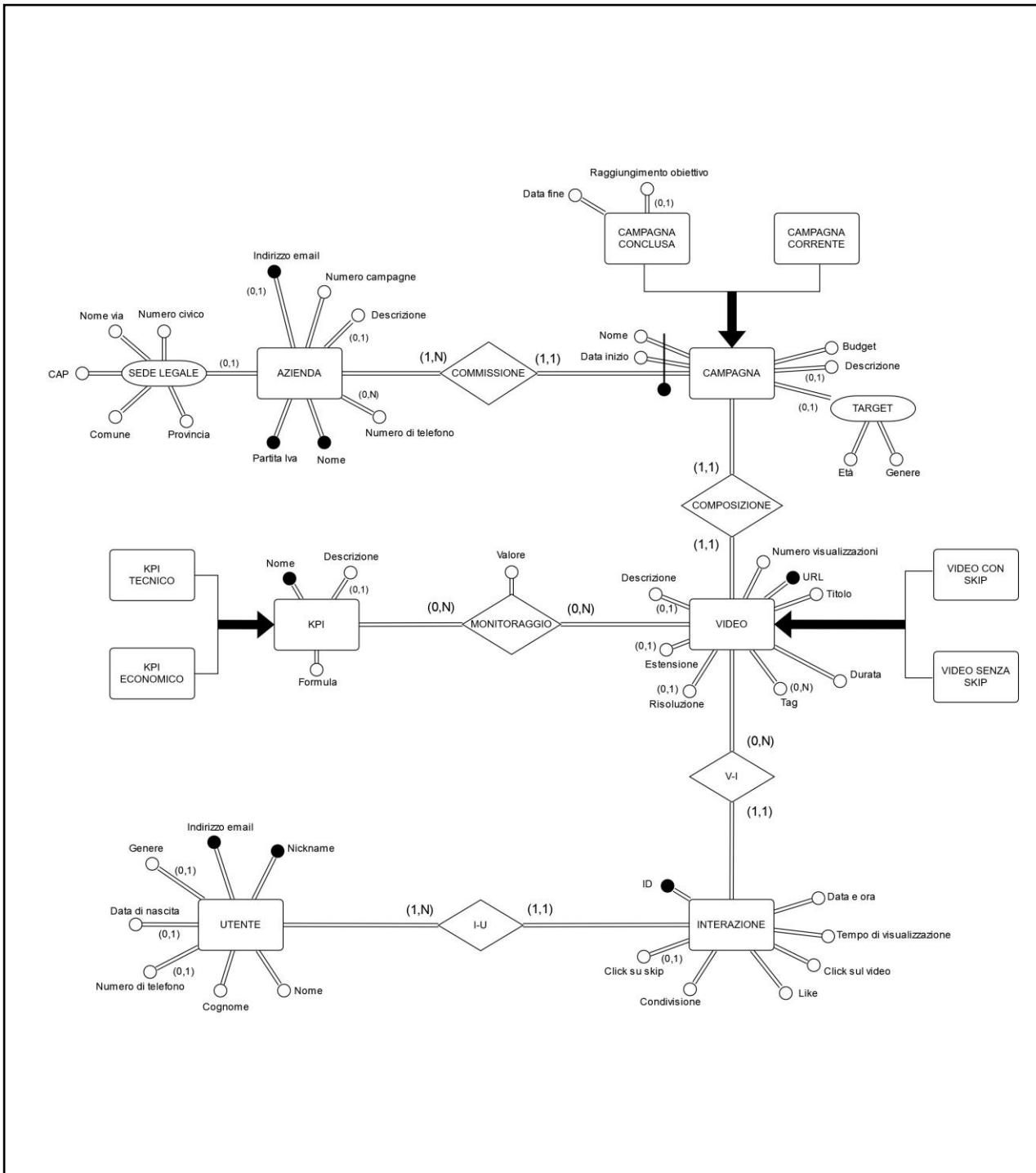


Figura 1. Schema E-R

3.1.1. Note sullo schema E-R

La metodologia utilizzata per lo sviluppo dello schema concettuale prevede l'analisi dei requisiti, la definizione di uno schema scheletro e il successivo affinamento. In particolare, la strategia utilizzata per la progettazione è la strategia mista, poiché combina gli aspetti positivi della top-down e della bottom-up. Infine, è stata svolta l'analisi di qualità, verificando correttezza, completezza, leggibilità e minimalità.

3.2. Design Pattern

3.2.1. Pattern Reificazione di relazione binaria

Il motivo per cui è stato applicato questo pattern è che un utente può trovarsi ad interagire più volte con lo stesso video e si vogliono memorizzare tutte le sue interazioni. In seguito all'applicazione del pattern è stato aggiunto l'ID per identificare un'interazione.

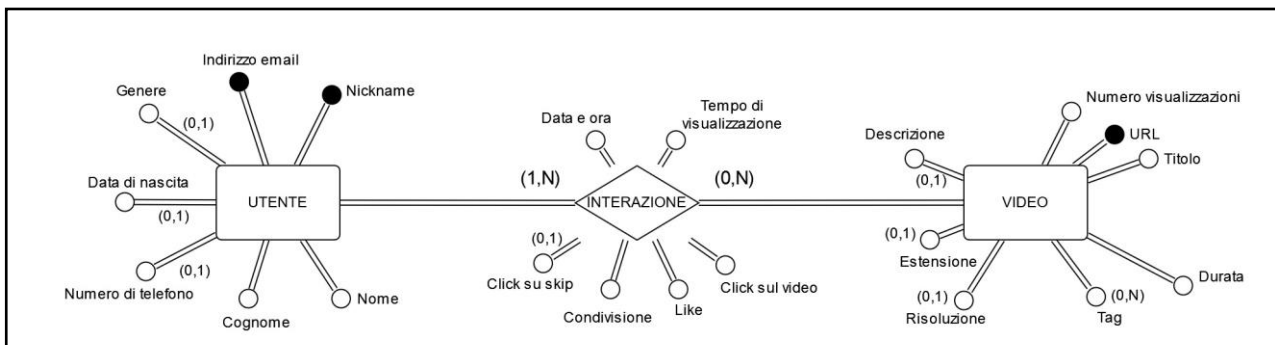


Figura 2. Schema precedente all'applicazione del Pattern Reificazione di relazione binaria

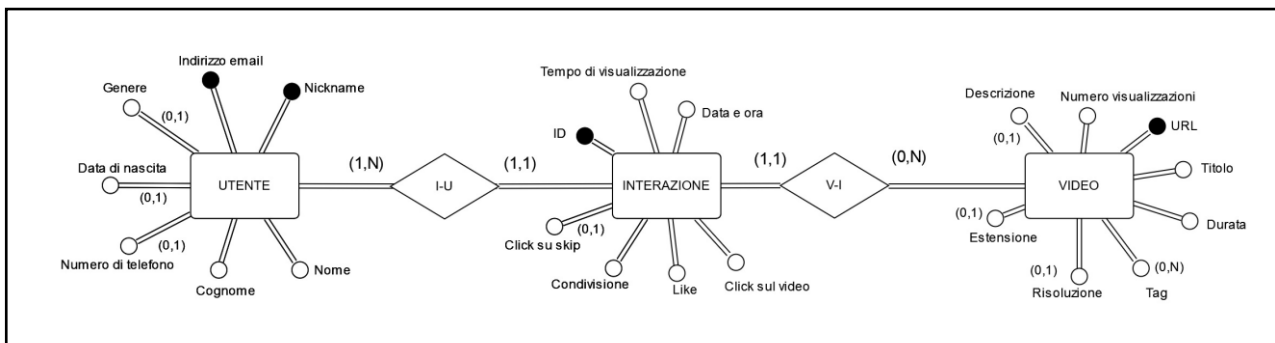


Figura 3. Schema successivo all'applicazione del Pattern Reificazione di relazione binaria

3.2.2. Pattern Part-of

Il motivo per cui è stato applicato questo pattern è che l'entità Campagna è di fatto parte dell'entità Azienda. La campagna è commissionata da un'unica azienda e l'azienda può commissionare più campagne.

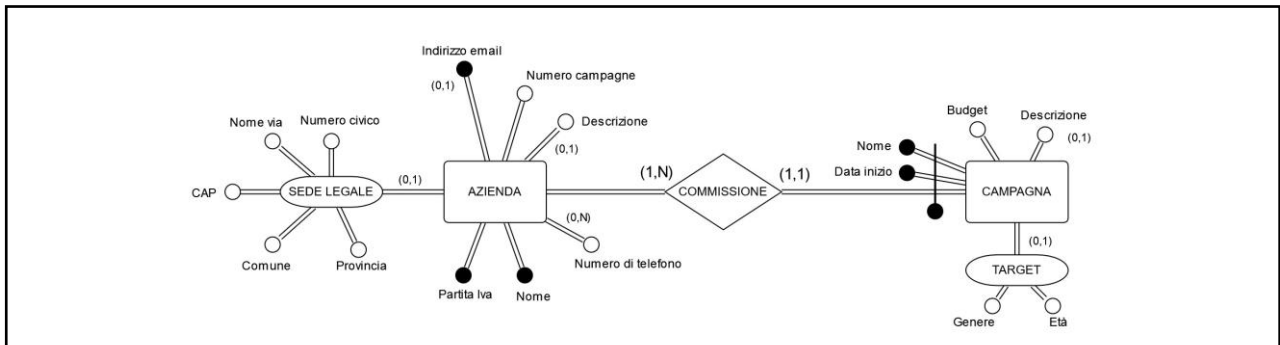


Figura 4. Schema successivo all'applicazione del Pattern Part-of

3.2.3. Pattern Storizzazione di un'entità

Il motivo per cui è stato applicato questo pattern è che si vogliono memorizzare le informazioni relative alle campagne correnti, ma si vuole anche tenere traccia delle informazioni delle campagne passate.

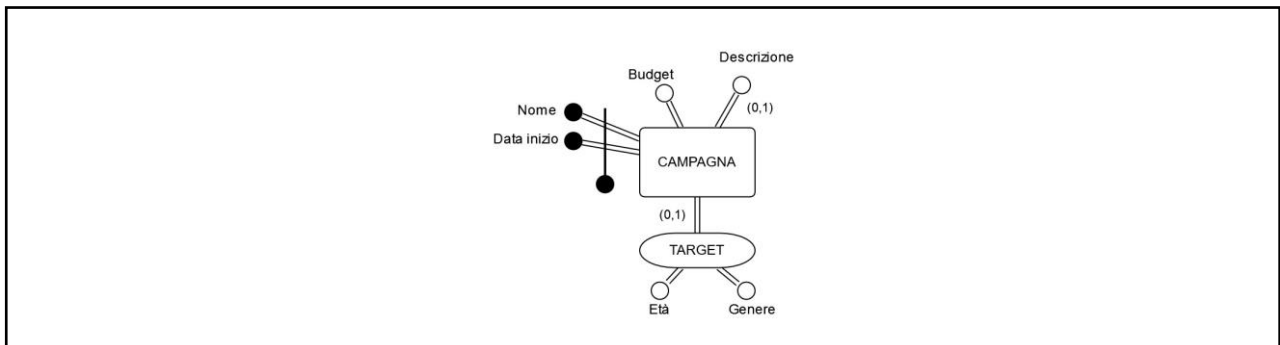


Figura 5. Schema precedente all'applicazione del Pattern Storizzazione di un'entità

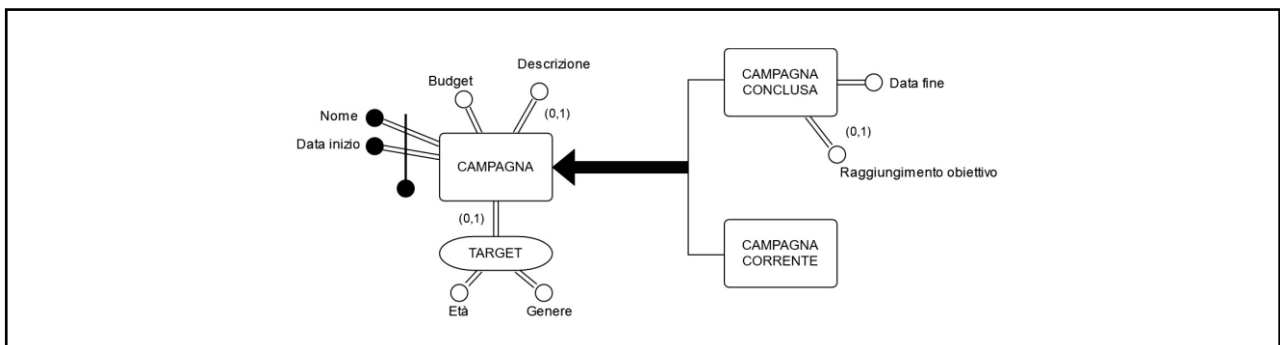


Figura 6. Schema successivo all'applicazione del Pattern Storizzazione di un'entità

3.3. Dizionario dei Dati

Entità	Descrizione	Attributi	Identificatore
Azienda	Azienda che commissiona la campagna pubblicitaria e fornisce i video da utilizzare.	Partita IVA, Nome, Sede legale (composto da Provincia, Comune, CAP, Nome via, Numero civico), Numero di telefono, Indirizzo e-mail, Descrizione, Numero di campagne.	Partita IVA oppure Nome.
Campagna	Campagna pubblicitaria online, effettuata sulla piattaforma YouTube.	Nome, Data inizio, Budget, Target (composto da Età e Genere), Descrizione.	Nome, Data inizio e Partita IVA dell'azienda oppure Nome, Data inizio e Nome dell'azienda.
Campagna Conclusa	Campagna pubblicitaria terminata.	Nome, Data inizio, Budget, Target (composto da Età e Genere), Descrizione, Data fine, Raggiungimento obiettivo.	Nome, Data inizio e Partita IVA dell'azienda oppure Nome, Data inizio e Nome dell'azienda.
Campagna Corrente	Campagna pubblicitaria in corso.	Nome, Data inizio, Budget, Target (composto da Età e Genere), Descrizione.	Nome, Data inizio e Partita IVA dell'azienda oppure Nome, Data inizio e Nome dell'azienda.
Video	Video promozionale utilizzato dalla campagna. È la risorsa utilizzata per diffondere la pubblicità tra gli utenti di YouTube.	URL, Titolo, Descrizione, Durata, Tag, Estensione, Risoluzione, Numero visualizzazioni.	URL.
Video con skip	Video in cui è presente il tasto di skip per saltare la pubblicità.	URL, Titolo, Descrizione, Durata, Tag, Estensione, Risoluzione, Numero visualizzazioni.	URL.
Video senza skip	Video in cui non è presente il tasto di skip per saltare la pubblicità.	URL, Titolo, Descrizione, Durata, Tag, Estensione, Risoluzione, Numero visualizzazioni.	URL.

Utente	Utente registrato sulla piattaforma YouTube e che visualizza i video pubblicitari, interagendo con essi.	Nickname, Indirizzo e-mail, Nome, Cognome, Data di nascita, Numero di telefono, Genere.	Nickname oppure Indirizzo e-mail.
Interazione	Interazione tra utente e video che descrive una serie di azioni effettuate sul video da parte dell'utente.	ID, Data e ora, Tempo di visualizzazione, Click sul video, Like, Condivisione, Click su skip.	ID.
KPI	KPI (Key Performance Indicator) utili per valutare il successo della campagna.	Nome, Descrizione, Formula.	Nome.
KPI Tecnico	KPI di tipo tecnico: tengono conto solo di parametri tecnici.	Nome, Descrizione, Formula.	Nome.
KPI Economico	KPI di tipo economico: rapportano i parametri economici a quelli tecnici.	Nome, Descrizione, Formula.	Nome.

Tabella 2. Dizionario dei dati – Entità

Relazioni	Descrizione	Entità Coinvolte	Attributi
Commissione	Un'azienda può commissionare più campagne e una campagna è commissionata da un'unica azienda.	Azienda, Campagna.	
Composizione	Una campagna è composta da un video e un video è relativo ad una campagna.	Campagna, Video.	
Monitoraggio	Un KPI può essere utilizzato per monitorare più video e un video può essere monitorato da più KPI.	KPI, Video.	Valore.
V-I	Un video può avere più interazioni e un'interazione è associata ad un video.	Video, Interazione.	
I-U	Un utente può avere più interazioni e un'interazione è associata a un utente.	Interazione, Utente.	

Tabella 3. Dizionario dei dati – Relazioni

Workpackage	Task	Responsabile
WP4	Regole Aziendali	Esposito Paolo

3.4. Regole Aziendali

Regole di Vincolo
<p>(RV1) Partita Iva in azienda deve essere di 11 caratteri.</p> <p>(RV2) CAP in azienda deve essere di 5 caratteri.</p> <p>(RV3) Numero di campagne in azienda è un numero positivo ≥ 1.</p> <p>(RV4) Budget in campagna deve essere un numero positivo.</p> <p>(RV5) Età del Target in campagna è un numero positivo.</p> <p>(RV6) Genere del Target in campagna deve essere M o F.</p> <p>(RV7) Raggiungimento dell'obiettivo in campagna conclusa è un numero compreso tra 0 e 1.</p> <p>(RV8) Data fine in campagna conclusa deve essere maggiore o uguale alla Data inizio in campagna.</p> <p>(RV9) URL in video ha una lunghezza fissata di 43 caratteri.</p> <p>(RV10) Durata in video deve essere compresa tra 5 secondi e 3 minuti.</p> <p>(RV11) Numero di visualizzazioni in video è un numero positivo ≥ 0.</p> <p>(RV12) Data e ora in interazione deve essere \geq Data inizio in campagna.</p> <p>(RV13) Si possono aggiungere solo interazioni relative a video di campagne correnti.</p> <p>(RV14) Tempo di visualizzazione in interazione deve essere minore o uguale della durata del video associato all'interazione.</p> <p>(RV15) Click su skip in interazione deve esserci se il video è con skip e non deve esserci se il video è senza skip.</p> <p>(RV16) Genere in utente deve essere M o F.</p>

Tabella 4. Regole di vincolo

Regole di derivazione
<p>(RD1) Numero di campagne in azienda è ridondante, in quanto può essere ricavato dal numero di commissioni dell'azienda.</p> <p>(RD2) Numero di visualizzazioni in video è ridondante, in quanto può essere ricavato contando il numero di occorrenze dell'associazione V-I relative a quel video.</p> <p>(RD3) Valore in monitoraggio si calcola con la formula del KPI a cui è associato, la quale viene applicata sui dati raccolti dalle interazioni relative al video che si sta monitorando.</p>

Tabella 5. Regole di derivazione

4. Progettazione Logica

Workpackage	Task	Responsabile
WP2	Progettazione Logica	Capaldo Vincenzo

Ristrutturazione Schema Concettuale

4.1.1. Analisi delle Prestazioni

4.1.1.1. Tavola dei volumi

Concetto	Tipo	Volume
Azienda	E	120
Commissione	R	240
Campagna	E	240
Campagna corrente	E	10
Campagna terminata	E	230
Composizione	R	240
Video	E	240
Video con skip	E	120
Video senza skip	E	120
Monitoraggio	R	1680
KPI	E	7
KPI tecnico	E	4
KPI economico	E	3
V-I	R	360.000
Interazione	E	360.000
I-U	R	360.000
Utenti	E	250.000

Tabella 6. Tavola dei volumi

Ipotesi: i volumi considerati sono calcolati supponendo che sia trascorso un anno dalla creazione della base dati.

4.1.1.2. Tavola delle operazioni

Operazione	Tipo	Frequenza
Operazione 1: <i>inserimento interazione</i>	B	36.500/anno
Operazione 2: <i>aggiornamento KPI</i>	B	730/anno
Operazione 3: <i>inserimento campagna</i>	I	240/anno
Operazione 4: <i>stampa il numero di visualizzazioni di un video</i>	I	240/anno
Operazione 5: <i>stampa l'azienda che ha commissionato più campagne</i>	I	2/anno
Operazione 6: <i>stampa il numero interazioni di un'azienda</i>	I	2/anno
Operazione 7: <i>stampa le 3 aziende, in media, con maggior successo</i>	I	1/anno

Tabella 7. Tavola delle operazioni

4.2. Analisi delle ridondanze

- Ridondanza 1: Numero campagne (AZIENDA).** Il numero di campagne commissionate da un'azienda si ottiene contando il numero di occorrenze dell'associazione COMMISSIONE relative a quell'azienda.
 VINCOLO DI DERIVAZIONE: RD1
 TIPO: Attributo Derivabile da conteggio di occorrenze
- Ridondanza 2: Numero visualizzazioni (VIDEO).** Il numero di visualizzazioni di un video si ottiene contando il numero di occorrenze dell'associazione V-I relative a quel video.
 VINCOLO DI DERIVAZIONE: RD2
 TIPO: Attributo Derivabile da conteggio di occorrenze

4.2.1. Analisi della ridondanza 1: Numero campagne

- Operazione 3: inserimento campagna**

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Campagna	E	1	S
Campagna corrente	E	1	S
Commissione	R	1	S
Azienda	E	1	L
Azienda	E	1	S
Composizione	R	1	S
Video	E	1	S
Video con skip / Video senza skip	E	1	S

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Campagna	E	1	S
Campagna corrente	E	1	S
Commissione	R	1	S
Composizione	R	1	S
Video	E	1	S
Video con skip / Video senza skip	E	1	S

- Operazione 5: stampa l'azienda che ha commissionato più campagne**

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Azienda	E	120	L

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Azienda	E	120	L
Commissione	R	240	L

4.2.1.1. Valutazione della ridondanza 1

Calcolo del costo delle operazioni con ridondanza: $15 \cdot 240 + 120 \cdot 2 = 3.840$

Calcolo del costo delle operazioni senza ridondanza: $12 \cdot 240 + 360 \cdot 2 = 3.600$

Calcolo occupazione di memoria: $4 \text{ byte} \cdot 120 = 480 \text{ byte}$

Dopo aver analizzato le operazioni che coinvolgono la ridondanza si osserva che, con il carico considerato:

- In presenza di ridondanza il costo delle operazioni è di 3840 accessi giornalieri
- L'occupazione di memoria è di circa 480 byte
- In assenza di ridondanza il costo delle operazioni è di 3600 accessi giornalieri

Pertanto, si decide di non mantenere la ridondanza in quanto aumenta, seppur di poco, il numero di accessi e l'occupazione di memoria.

4.2.2. Analisi della ridondanza 2: Numero visualizzazioni

- **Operazione 1: inserimento interazione**

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Interazione	E	1	S
I-U	E	1	S
V-I	R	1	S
Video	E	1	L
Video	E	1	S
Video con skip / Video senza skip	E	1	L
Video con skip / Video senza skip	E	1	S

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Interazione	E	1	S
I-U	R	1	S
V-I	R	1	S

- **Operazione 4: stampa il numero visualizzazioni di un video**

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Video	E	1	L
Video con skip / Video senza skip	E	1	L

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Video	E	1	L
Video con skip / Video senza skip	E	1	L
V-I	R	1.500	L

- **Operazione 6: stampa il numero interazioni di un'azienda**

Con Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Azienda	E	1	L
Commissione	R	2	L
Campagna	E	2	L
Composizione	R	2	L
Video	E	2	L

Senza Ridondanza

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Azienda	E	1	L
Commissione	R	2	L
Campagna	E	2	L
Composizione	R	2	L
Video	E	2	L
Video con skip / Video senza skip	E	2	L
V-I	R	3.000	L

4.2.2.1. Valutazione della ridondanza 2

Calcolo del costo delle operazioni con ridondanza: $12 \cdot 36.500 + 2 \cdot 240 + 9 \cdot 2 = 438.498$

Calcolo del costo delle operazioni senza ridondanza: $6 \cdot 36.500 + 1502 \cdot 240 + 3.011 \cdot 2 = 585.502$

Calcolo occupazione di memoria: $4 \text{ byte} \cdot 240 = 960 \text{ byte}$

Dopo aver analizzato le operazioni che coinvolgono la ridondanza si osserva che, con il carico considerato:

- In presenza di ridondanza il costo delle operazioni è di 438.498 accessi giornalieri
- L'occupazione di memoria è di circa 960 byte
- In assenza di ridondanza il costo delle operazioni è di 585.502 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 *Campagna*

Per eliminare la gerarchia relativa alle campagne viene utilizzata la prima strategia, ovvero l'accorpamento delle figlie della generalizzazione nel genitore. Questo perché nelle operazioni principali gli accessi al padre e alle figlie sono contestuali. Dunque, vengono aggiunti gli attributi Data fine e Raggiungimento obiettivo alla campagna.

Se Data fine non c'è (la campagna è corrente) allora Raggiungimento obiettivo non deve esserci.

Se Data fine c'è (la campagna è conclusa) allora Raggiungimento obiettivo può esserci.

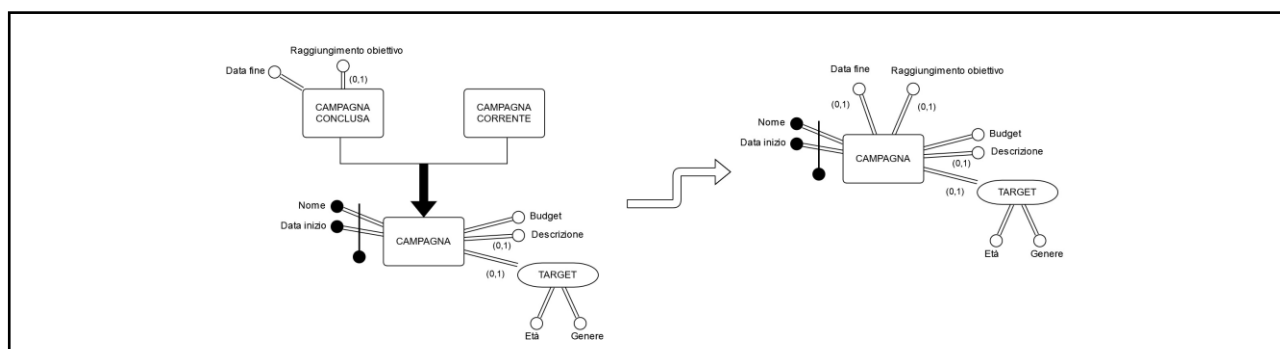


Figura 7. Schema dell'eliminazione della generalizzazione Campagna

4.3.2. Generalizzazione *Video*

Per eliminare la gerarchia relativa ai video viene utilizzata la prima strategia, ovvero l'accorpamento delle figlie della generalizzazione nel genitore. Questo perché nelle operazioni principali gli accessi al padre e alle figlie sono contestuali. Dunque, viene aggiunto l'attributo booleano Skip all'entità video, il quale sarà true nel caso di video con skip e false nel caso di video senza skip.

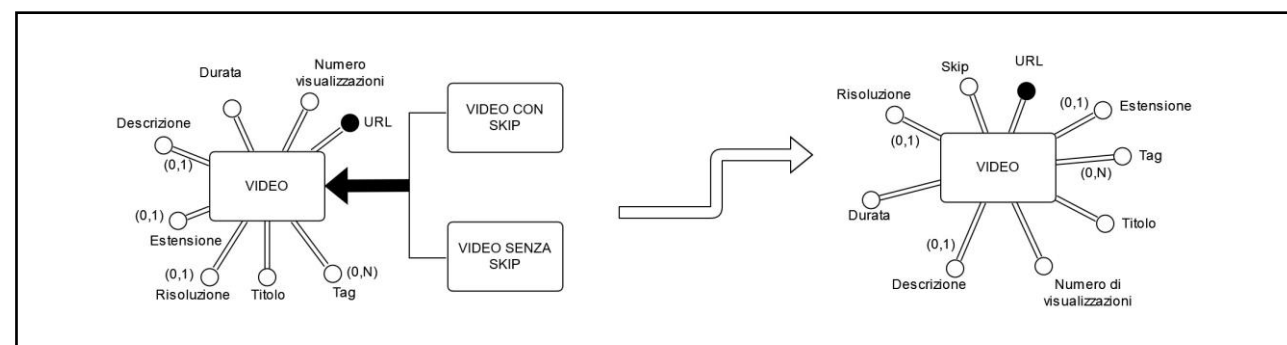


Figura 8. Schema dell'eliminazione della generalizzazione Video

4.3.1. Generalizzazione KPI

Per eliminare la gerarchia relativa ai KPI viene utilizzata la prima strategia, ovvero l'accorpamento delle figlie della generalizzazione nel genitore. Questo perché nelle operazioni principali gli accessi al padre e alle figlie sono contestuali. Dunque, viene aggiunto l'attributo Tipo all'entità KPI, il quale potrà assumere due valori prestabiliti: "tecnico" o "economico".

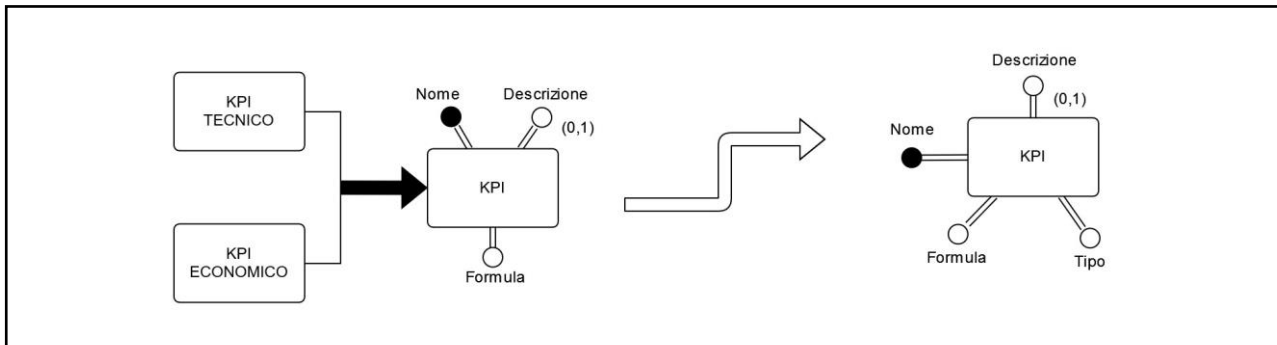


Figura 9. Schema dell'eliminazione della generalizzazione KPI

4.4. Partizionamento/Accorpamento Entità e Associazioni

- Non sono stati effettuati partizionamenti/accorpamenti di entità o di associazioni.
- Eliminazione attributi composti:
 1. Sede legale in Azienda
Vincolo aggiuntivo: Provincia, Comune, CAP, Nome via, Numero civico o mancano tutti o sono presenti tutti.
 2. Target in Campagna
Vincolo aggiuntivo: Età e Genere o mancano tutti o sono presenti tutti.
- Eliminazione attributi multivalore:
 1. Numero di telefono in Azienda
Numero di telefono diventa un'entità in associazione ad Azienda. Un'azienda può avere da 0 a N numeri di telefono, mentre un numero di telefono appartiene ad una e una sola azienda.
 2. Tag in Video
Tag diventa un'entità in associazione a Video. Un video può avere da 0 a N tag, mentre un tag può appartenere ad 1 o più video.

4.5. Scelta degli identificatori principali

Per la scelta degli identificatori principali sono stati adottati i seguenti criteri:

- I. assenza di opzionalità
- II. semplicità
- III. utilizzo nelle operazioni più frequenti o importanti

Di seguito sono riportate le scelte degli identificatori principali, con le rispettive motivazioni, per ogni entità:

Azienda: l'identificatore principale scelto è Partita IVA perché tipicamente è più breve rispetto a Nome ed è anche quello che poi viene utilizzato nella ricerca di un'azienda.

Numero di telefono: l'identificatore principale scelto è quello derivante dall'eliminazione dell'attributo multivalore; dunque, sarà composto dal Numero e dalla Partita IVA dell'azienda.

Campagna: un possibile identificatore principale è quello composto da Nome, Data inizio e Partita IVA, ma è troppo complesso e poco pratico da utilizzare; per questo motivo viene introdotto un identificatore principale più semplice chiamato ID.

Video: un possibile identificatore principale è URL, ma è una stringa lunga e per questo motivo viene introdotto un identificatore principale più semplice chiamato ID.

KPI: l'identificatore principale scelto è Nome, poiché rispecchia i criteri.

Tag: l'identificatore principale scelto è quello derivante dall'eliminazione dell'attributo multivalore; dunque, sarà composto dal Nome del tag.

Interazione: l'identificatore principale scelto è ID, derivante dal pattern utilizzato.

Utente: l'identificatore principale scelto è Nickname perché è più breve rispetto a Indirizzo email ed è anche quello che poi viene utilizzato nella ricerca di un utente.

4.6. Schema ristrutturato finale

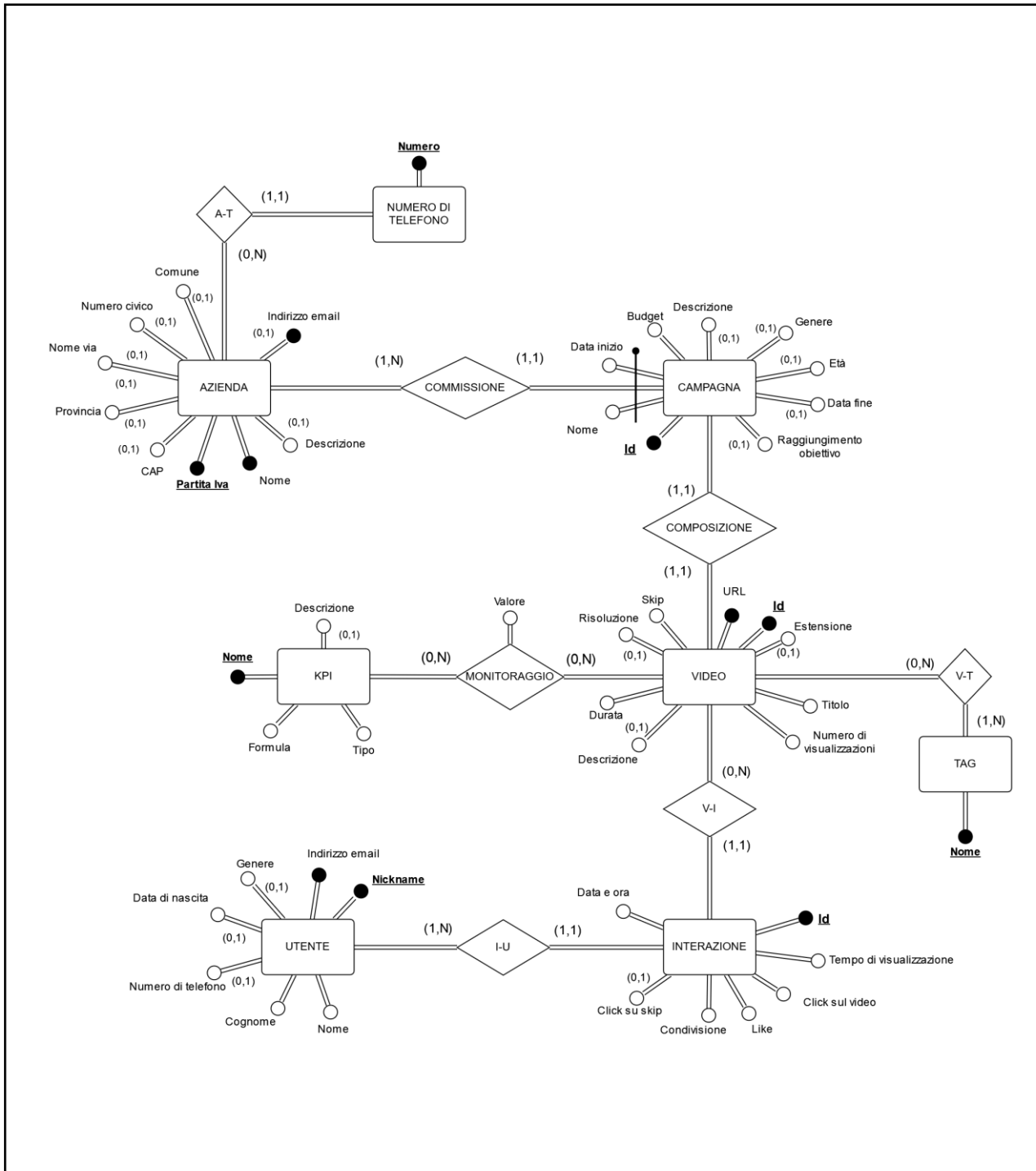


Figura 10. Schema ER Ristrutturato

4.7. Schema logico

NUMERO_DI_TELEFONO (numero, partita_iva)

AZIENDA (partita_iva, nome, provincia, comune, nome_via, numero_civico, cap, indirizzo_email, descrizione)

CAMPAGNA (id, nome, data_inizio, data_fine, budget, eta, genere, descrizione, raggiungimento_obiettivo, partita_iva, id_video)

VIDEO (id, url, titolo, descrizione, durata, estensione, risoluzione, numero_visualizzazioni, skip, id_campagna)

MONITORAGGIO (nome_kpi, id_video, valore)

KPI (nome, formula, descrizione, tipo)

V_T (id_video, nome_tag)

TAG (nome)

V_I (id_interazione, id_video)

INTERAZIONE (id, data_e_ora, tempo_di_visualizzazione, click_su_skip, click_su_video, like, condivisione, nickname_utente)

UTENTE (nickname, indirizzo_email, nome, cognome, data_di_nascita, numero_di_telefono, genere)

4.8. Documentazione dello schema logico

Vincoli non rappresentabili nella traduzione:

- la terna (nome, data_inizio e partita_iva) in campagna deve essere univoca;
- un tag deve avere almeno un video;
- un'azienda deve aver commissionato almeno una campagna;
- un utente deve avere almeno un'interazione;
- i seguenti vincoli di integrità referenziale:

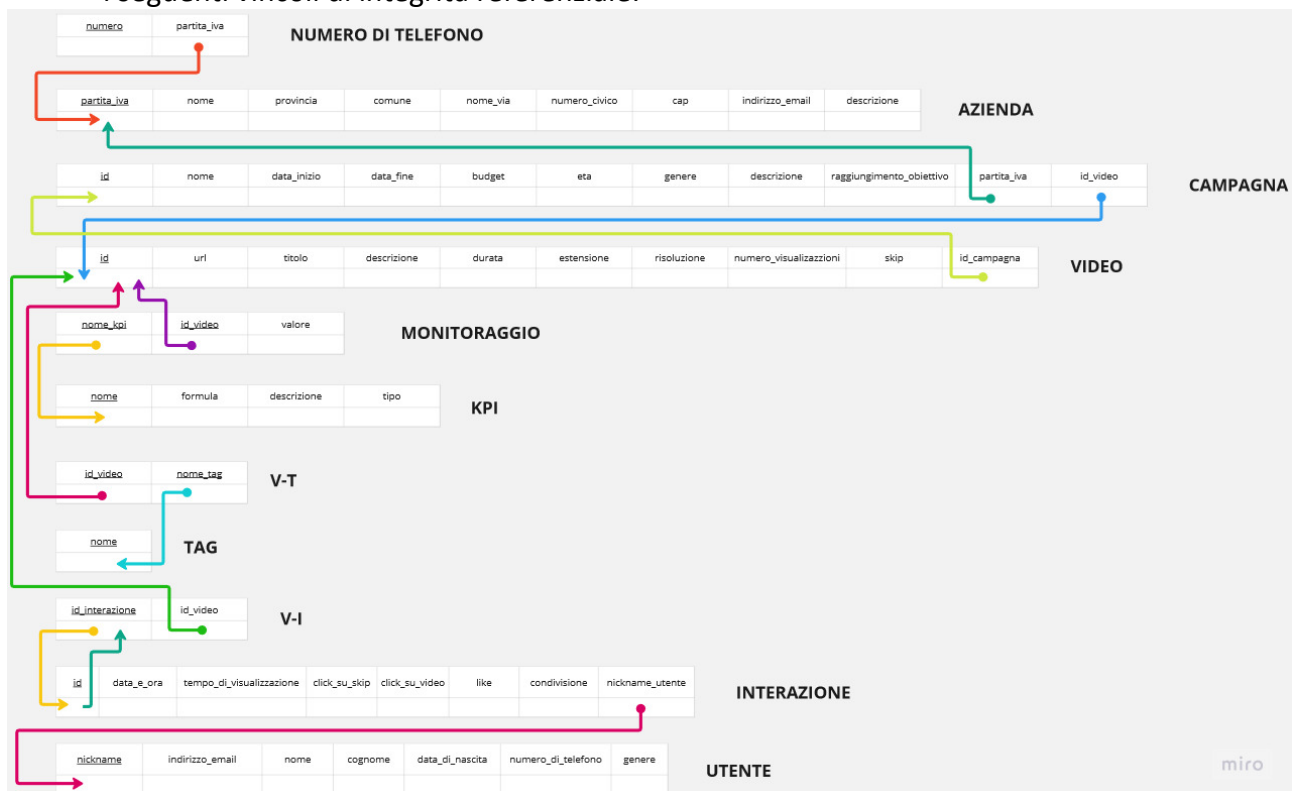


Figura 11. Schema dei vincoli di integrità referenziale

5. Normalizzazione

Workpackage	Task	Responsabile
WP3	Normalizzazione	Donnarumma Luca

Dipendenze funzionali Numero di Telefono:

- numero -> partita_iva

Dipendenze funzionali Azienda:

- partita_iva -> nome
- partita_iva -> provincia
- partita_iva -> comune
- partita_iva -> nome_via
- partita_iva -> numero_civico
- partita_iva -> cap
- partita_iva -> indirizzo_email
- partita_iva -> descrizione
- nome -> partita_iva
- nome -> provincia
- nome -> comune
- nome -> nome_via
- nome -> numero_civico
- nome -> cap
- nome -> indirizzo_email
- nome -> descrizione

Dipendenze funzionali Campagna:

- id -> nome
- id -> data_inizio
- id -> data_fine
- id -> budget
- id -> eta
- id -> genere
- id -> descrizione
- id -> raggiungimento_obiettivo
- id -> partita_iva
- id -> id_video
- nome, data_inizio, partita_iva -> id
- nome, data_inizio, partita_iva -> budget
- nome, data_inizio, partita_iva -> eta
- nome, data_inizio, partita_iva -> genere
- nome, data_inizio, partita_iva -> descrizione
- nome, data_inizio, partita_iva -> raggiungimento_obiettivo
- nome, data_inizio, partita_iva -> partita_iva
- nome, data_inizio, partita_iva -> id_video
- id_video -> id
- id_video -> nome

- id_video -> data_inizio
- id_video -> data_fine
- id_video -> budget
- id_video -> eta
- id_video -> genere
- id_video -> descrizione
- id_video -> raggiungimento_obiettivo
- id_video -> partita_iva

Dipendenze funzionali Video:

- id -> url
- id -> titolo
- id -> descrizione
- id -> durata
- id -> estensione
- id -> risoluzione
- id -> numero_visualizzazioni
- id -> skip
- id -> id_campagna
- url -> id
- url -> titolo
- url -> descrizione
- url -> durata
- url -> estensione
- url -> risoluzione
- url -> numero_visualizzazioni
- url -> skip
- url -> id_campagna
- id_campagna -> id
- id_campagna -> url
- id_campagna -> titolo
- id_campagna -> descrizione
- id_campagna -> durata
- id_campagna -> estensione
- id_campagna -> risoluzione
- id_campagna -> numero_visualizzazioni
- id_campagna -> skip

Dipendenze funzionali Monitoraggio:

- nome_kpi, id_video -> valore

Dipendenze funzionali KPI:

- nome -> formula
- nome -> descrizione
- nome -> tipo

Dipendenze funzionali V-I:

- id_interazione -> id_video

Dipendenze funzionali Interazione:

- id -> data_e_ora
- id -> tempo_di_visualizzazione
- id -> click_su_skip
- id -> click_su_video
- id -> like
- id -> condivisione
- id -> nickname_utente

Dipendenze funzionali Utente:

- nickname -> indirizzo_email
- nickname -> nome
- nickname -> cognome
- nickname -> data_di_nascita
- nickname -> numero_di_telefono
- nickname -> genere
- indirizzo_email -> nickname
- indirizzo_email -> nome
- indirizzo_email -> cognome
- indirizzo_email -> data_di_nascita
- indirizzo_email -> numero_di_telefono
- indirizzo_email -> genere

Verifiche forme normali:

1. Prima forma normale:
Tutte le relazioni sono in prima forma normale perché ognuna di esse non presenta attributi multivalori e attributi composti.
2. Seconda forma normale:
Tutte le relazioni sono in seconda forma normale perché ognuna di esse è in prima forma normale e ogni attributo non primo dipende in maniera piena dalla chiave della relazione.
3. Terza forma normale:
Tutte le relazioni sono in terza forma normale perché ognuna di esse è in seconda forma normale e per ogni dipendenza funzionale non banale $Y \rightarrow Z$, definita sulla relazione, Y è chiave della relazione oppure Z è un attributo primo.
4. Forma normale di Boyce and Codd:
Tutte le relazioni sono in forma normale di Boyce and Codd perché ognuna di esse è in prima forma normale e per ogni dipendenza funzionale non banale $Y \rightarrow Z$, definita sulla relazione, Y contiene la chiave della relazione.

6. Script Creazione e Popolamento Database

Workpackage	Task	Responsabile
WP2	SQL: Script creazione e popolamento	Capaldo Vincenzo

SCRIPT DI CREAZIONE:

```

/*SCRIPT DI CREAZIONE*/

/*creazione database*/
DROP DATABASE IF EXISTS basi_di_dati_gruppo_05;
CREATE DATABASE basi_di_dati_gruppo_05;

/*DROP delle tabelle*/
DROP TABLE IF EXISTS azienda CASCADE;
DROP TABLE IF EXISTS numero_di_telefono CASCADE;
DROP TABLE IF EXISTS campagna CASCADE;
DROP TABLE IF EXISTS video CASCADE;
DROP TABLE IF EXISTS kpi CASCADE;
DROP TABLE IF EXISTS monitoraggio CASCADE;
DROP TABLE IF EXISTS tag CASCADE;
DROP TABLE IF EXISTS v_t CASCADE;
DROP TABLE IF EXISTS utente CASCADE;
DROP TABLE IF EXISTS interazione CASCADE;
DROP TABLE IF EXISTS v_i CASCADE;

/*creazione tabella azienda*/
CREATE TABLE azienda (
    partita_iva CHAR(11) PRIMARY KEY,
    nome VARCHAR(50) NOT NULL UNIQUE,
    provincia VARCHAR(30),
    comune VARCHAR(30),
    nome_via VARCHAR(30),
    numero_civico INTEGER,
    cap CHAR(5),
    indirizzo_email VARCHAR(70) UNIQUE,
    descrizione VARCHAR(300),
    CONSTRAINT check_partita_iva_azienza CHECK (CHAR_LENGTH(partita_iva)=11),
    CONSTRAINT check_cap_azienza CHECK(CHAR_LENGTH(cap)=5),
    CONSTRAINT check_sede_legale_azienza CHECK ( /*gli attributi della sede legale
o ci sono tutti o nessuno*/
        (provincia IS NULL AND comune IS NULL AND cap IS NULL AND nome_via IS NULL
AND numero_civico IS NULL)
        OR
        (provincia IS NOT NULL AND comune IS NOT NULL AND cap IS NOT NULL AND
nome_via IS NOT NULL AND numero_civico IS NOT NULL)
    )
);

```

```

/*creazione tabella numero_di_telefono*/
CREATE TABLE numero_di_telefono(
    numero VARCHAR(30) PRIMARY KEY,
    partita_iva CHAR(11) NOT NULL REFERENCES azienda(partita_iva) ON DELETE CASCADE
ON UPDATE CASCADE,
    CONSTRAINT check_partita_iva_numero_di_telefono
CHECK(CHAR_LENGTH(partita_iva)=11)
);

/*creazione tabella campagna*/
CREATE TABLE campagna(
    id INTEGER PRIMARY KEY,
    nome VARCHAR(50) NOT NULL,
    data_inizio DATE NOT NULL,
    data_fine DATE,
    budget DECIMAL(8,2) NOT NULL,
    eta INTEGER,
    genere CHAR(1),
    descrizione VARCHAR(300),
    raggiungimento_obiettivo DECIMAL(3,2),
    partita_iva CHAR(11) NOT NULL REFERENCES azienda(partita_iva) ON DELETE
RESTRICT ON UPDATE CASCADE
    DEFERRABLE INITIALLY DEFERRED, /*deferred per poter implementare il trigger
di cardinalità minima dell'azienda*/
    id_video INTEGER NOT NULL UNIQUE,
    CONSTRAINT attributi_univoci UNIQUE (nome,data_inizio,partita_iva),
    CONSTRAINT check_budget_campagna CHECK(budget >= 0.00),
    CONSTRAINT check_eta_campagna CHECK(eta>=0),
    CONSTRAINT check_genere_campagna CHECK (genere='M' OR genere='F'),
    CONSTRAINT check_data_fine_campagna CHECK(data_fine >= data_inizio),
    CONSTRAINT check_raggiungimento_obiettivo_campagna CHECK (
        (data_fine IS NULL AND raggiungimento_obiettivo IS NULL)
        /*se data_fine è NULL allora raggiungimento_obiettivo deve essere NULL */
        OR (data_fine IS NOT NULL AND ((raggiungimento_obiettivo >= 0.00 AND
raggiungimento_obiettivo <= 1.00) OR raggiungimento_obiettivo IS NULL))
        /*se data_fine non è NULL allora raggiungimento_obiettivo deve essere
compreso tra 0 e 1 oppure deve essere NULL*/
    ),
    CONSTRAINT check_partita_iva_campagna CHECK(CHAR_LENGTH(partita_iva)=11),
    CONSTRAINT check_target_campagna CHECK ( /*gli attributi del target o ci sono
tutti o nessuno*/
        (eta IS NULL AND genere IS NULL)
        OR
        (eta IS NOT NULL AND genere IS NOT NULL)
    )
);

```

```

/*creazione tabella video*/
CREATE TABLE video(
    id INTEGER PRIMARY KEY,
    url CHAR(43) NOT NULL UNIQUE,
    titolo VARCHAR(30) NOT NULL,
    descrizione VARCHAR(300),
    durata TIME NOT NULL,
    estensione VARCHAR(10),
    risoluzione VARCHAR(10),
    numero_visualizzazioni INTEGER NOT NULL,
    skip BOOLEAN NOT NULL,
    id_campagna INTEGER NOT NULL UNIQUE REFERENCES campagna(id) ON DELETE CASCADE
ON UPDATE CASCADE,
    CONSTRAINT check_url_video CHECK(CHAR_LENGTH(url)=43),
    CONSTRAINT check_durata_video CHECK(durata >= '00:00:05' AND durata <=
'00:03:00'),
    CONSTRAINT check_numero_visualizzazioni_video CHECK(numero_visualizzazioni>=0)
);

/*aggiunta del vincolo di integrità referenziale alla tabella campagna verso
video*/
ALTER TABLE campagna ADD CONSTRAINT fk_campagna_video
    FOREIGN KEY (id_video) REFERENCES video(id) ON DELETE RESTRICT ON UPDATE
CASCADE
    DEFERRABLE INITIALLY DEFERRED;

/*creazione tabella kpi*/
CREATE TABLE kpi(
    nome VARCHAR(30) PRIMARY KEY,
    formula VARCHAR(100) NOT NULL,
    descrizione VARCHAR(300),
    tipo VARCHAR(10) NOT NULL,
    CONSTRAINT check_tipo_kpi CHECK (tipo='tecnico' OR tipo='economico')
);

/*creazione tabella monitoraggio*/
CREATE TABLE monitoraggio(
    nome_kpi VARCHAR(30) REFERENCES kpi(nome) ON DELETE CASCADE ON UPDATE CASCADE,
    id_video INTEGER REFERENCES video(id) ON DELETE CASCADE ON UPDATE CASCADE,
    valore DECIMAL(7,3) NOT NULL,
    CONSTRAINT pk_monitoraggio PRIMARY KEY(nome_kpi, id_video)
);

/*creazione tabella tag*/
CREATE TABLE tag(
    nome VARCHAR(30) PRIMARY KEY
);

/*creazione tabella v_t*/

```

```

CREATE TABLE v_t(
    id_video INTEGER REFERENCES video(id) ON DELETE CASCADE ON UPDATE CASCADE,
    nome_tag VARCHAR(30) REFERENCES tag(nome) ON DELETE CASCADE ON UPDATE CASCADE
    DEFERRABLE INITIALLY DEFERRED, /*deferred per poter implementare il trigger
di cardinalità minima del tag*/
    CONSTRAINT pk_v_t PRIMARY KEY(id_video, nome_tag)
);

/*creazione tabella utente*/
CREATE TABLE utente(
    nickname VARCHAR(30) PRIMARY KEY,
    indirizzo_email VARCHAR(70) NOT NULL UNIQUE,
    nome VARCHAR(30) NOT NULL,
    cognome VARCHAR(30) NOT NULL,
    data_di_nascita DATE,
    numero_di_telefono VARCHAR(30),
    genere CHAR(1),
    CONSTRAINT check_genere_utente CHECK (genere='M' OR genere='F')
);

/*creazione tabella interazione*/
CREATE TABLE interazione(
    id INTEGER PRIMARY KEY,
    data_e_ora TIMESTAMP NOT NULL,
    tempo_di_visualizzazione TIME NOT NULL,
    click_su_skip BOOLEAN,
    click_su_video BOOLEAN NOT NULL,
    "like" BOOLEAN NOT NULL,
    condivisione BOOLEAN NOT NULL,
    nickname_utente VARCHAR(30) NOT NULL REFERENCES utente(nickname) ON DELETE
RESTRICT ON UPDATE CASCADE
    DEFERRABLE INITIALLY DEFERRED /*deferred per poter implementare il trigger
di cardinalità minima dell'utente*/
);

/*creazione tabella v_i*/
CREATE TABLE v_i(
    id_interazione INTEGER PRIMARY KEY REFERENCES interazione(id) ON DELETE CASCADE
ON UPDATE CASCADE,
    id_video INTEGER NOT NULL REFERENCES video(id) ON DELETE CASCADE ON UPDATE
CASCADE
);

/*aggiunta del vincolo di integrità referenziale alla tabella interazione verso
v_i*/
ALTER TABLE interazione ADD CONSTRAINT fk_interazione_v_i
    FOREIGN KEY (id) REFERENCES v_i(id_interazione) ON DELETE CASCADE ON UPDATE
RESTRICT
    DEFERRABLE INITIALLY DEFERRED;

```

SCRIPT DI POPOLAMENTO:

```

/*SCRIPT DI POPOLAMENTO*/

BEGIN TRANSACTION;
/*popolamento campagna*/
INSERT INTO campagna(id, nome, data_inizio, data_fine, budget, eta, genere,
descrizione, raggiungimento_obiettivo, partita_iva, id_video) VALUES
(1, 'Presentazione Singolo', '2023-06-09', NULL, 10000.00, 17, 'M', NULL, NULL,
'12345678901', 1),
(2, 'Lancio nuovo modello Piaggio', '2023-05-20', NULL, 150000.00, 18, 'F', NULL,
NULL, '11145672345', 2),
(3, 'Lancio nuovi Spot, robotica per il lavoro', '2023-05-30', NULL, 5000.00, 37,
'M', NULL, NULL, '25678274560', 3),
(4, 'La capacità della nostra AI', '2023-05-15', NULL, 120000.00, 15, 'M', NULL,
NULL, '45623456789', 4),
(5, 'Costruzioni 3d a basso costo', '2023-04-10', NULL, 13000.00, NULL, NULL,
'materiale per la stampa 3d', NULL, '56734590122', 5),
(6, 'Costruzioni 3d ad alto costo', '2023-07-10', '2023-07-20', 13000.00, NULL,
NULL, 'materiale per la stampa 3d', 0.9, '56734590122', 6);

/*popolamento azienda*/
INSERT INTO azienda(partita_iva, nome, provincia, comune, nome_via, numero_civico,
cap, indirizzo_email, descrizione) VALUES
('12345678901', 'PincoPal', 'SA', 'Nocera Inferiore', 'Vittorio Emanuele III', 33,
'84014', 'business@pincopal.it', NULL),
('25678274560', 'FrancoSRL', 'SA', 'Pagani', 'Corso', 82, '84015',
'produzione@francosrl.it', NULL),
('11145672345', 'Piaggio', NULL, NULL, NULL, NULL, NULL, 'commercial@piaggio.it',
'Storica azienda produttrice di motocicli'),
('45623456789', 'Google Italia', NULL, NULL, NULL, NULL, NULL,
'googleitalia@google.it', NULL),
('56734590122', 'LibriINC', 'MI', 'Milano', 'StradaMilanese', 123, '20019',
'annunci@libriinc.it', 'Grande azienda di produzione cartacea');

/*popolamento video*/
INSERT INTO video(id, url, titolo, descrizione, durata, estensione, risoluzione,
numero_visualizzazioni, skip, id_campagna) VALUES
(1, 'https://www.youtube.com/watch?v=dQw4w9WgXcQ', 'Never Gonna Give You Up', NULL,
'00:00:30', '.mp4', '1920x1080', 4, TRUE, 1),
(2, 'https://www.youtube.com/watch?v=OCMnrsxTbEs', 'PiaggioSpot', 'Spot piaggio',
'00:01:45', '.mp4', '240p', 4, TRUE, 2),
(3, 'https://www.youtube.com/watch?v=qgHeCfMa39E', 'Boston Dynamics', 'The industry
standard for dangerous and routine autonomous inspections just got better',
'00:01:45', '.mp4', '4k', 5, FALSE, 3),
(4, 'https://www.youtube.com/watch?v=VKEA5cJluc0', 'NVIDIA AI', 'Voyager: An Open-
Ended Embodied Agent with Large Language Models', '00:02:30', '.mp4', '1920x1080',
3, TRUE, 4),

```

```
(5, 'https://www.youtube.com/watch?v=qe5mv07vF4Y', 'Material for 3d printing',
'composite materials and electroplating', '00:02:59', '.mp4', '1920x1080', 2,
FALSE, 5),
(6, 'https://www.youtube.com/watch?v=qe5mv07vT9Y', 'Material for 3d printing',
'composite materials and electroplating', '00:02:59', '.mp4', '1920x1080', 2,
FALSE, 6);
COMMIT;

/*popolamento numero di telefono*/
INSERT INTO numero_di_telefono(numero, partita_iva) VALUES
('3403344516', '12345678901'),
('3385613368', '25678274560'),
('3346783367', '25678274560'),
('3409906654', '25678274560'),
('3472233345', '11145672345'),
('3392468456', '11145672345'),
('3346667897', '56734590122'),
('3394363748', '56734590122');

BEGIN TRANSACTION;
/*popolamento v_t*/
INSERT INTO v_t(id_video, nome_tag) VALUES
(1, 'musica'),
(3, 'tecnologia'),
(3, 'engineering'),
(3, 'ai'),
(3, 'robot'),
(2, 'meccanica'),
(2, 'engineering'),
(2, 'italia'),
(5, '3d'),
(5, 'engineering'),
(5, 'tecnologia'),
(4, 'papers'),
(4, 'engineering'),
(4, 'tecnologia');

/*popolamento tag*/
INSERT INTO tag(nome) VALUES
('musica'),
('tecnologia'),
('3d'),
('engineering'),
('meccanica'),
('italia'),
('robot'),
('ai'),
('papers');
COMMIT;
```

```

/*popolamento kpi*/
INSERT INTO kpi(nome, formula, descrizione, tipo) VALUES
('Reach', 'numero di utenti unici che hanno visualizzato il video', 'indica il
numero di utenti che hanno visto il video almeno una volta durante il ciclo di vita
della campagna', 'tecnico'),
('Frequency', 'visualizzazioni/reach', 'indica il numero medio di volte che un
singolo utente ha visualizzato quel video', 'tecnico'),
('Visualizzazione Completa', '(visualizzazioni complete / visualizzazioni) * 100',
'indica la percentuale di utenti che hanno guardato il video per intero',
'tecnico'),
('Skip', 'numero di volte che il video è stato saltato', 'indica il numero di volte
che il video è stato saltato dagli utenti', 'tecnico'),
('Costo per Click', 'budget/numero click', 'indica il costo medio per ogni click
generato dal video', 'economico'),
('Costo per like', 'budget/numero like', 'indica il costo medio per ottenere un
like sul tuo video', 'economico'),
('Costo per condivisione', 'budget/numero condivisioni', 'indica il costo medio per
ottenere una condivisione del tuo video', 'economico');

/*popolamento monitoraggio*/
INSERT INTO monitoraggio(nome_kpi, id_video, valore) values
('Reach', 1, 500.00),
('Reach', 2, 347.00),
('Reach', 3, 142.00),
('Reach', 4, 243.00),
('Reach', 5, 111.00),
('Frequency', 1, 12.34),
('Frequency', 2, 55.34),
('Skip', 1, 12.00),
('Skip', 2, 235.00),
('Skip', 4, 143.00),
('Costo per Click', 3, 0.12),
('Costo per Click', 5, 1.00),
('Costo per Click', 1, 0.05);

BEGIN TRANSACTION;
/*popolamento interazione*/
INSERT INTO interazione(id, data_e_ora, tempo_di_visualizzazione, click_su_skip,
click_su_video, "like", condivisione, nickname_utente) VALUES
(1, '2023-04-15;10:43:57', '00:02:59', NULL, FALSE, FALSE, FALSE, 'enzosong'),
(2, '2023-05-01;18:22:12', '00:02:59', NULL, TRUE, TRUE, FALSE, 'enzosong'),
(3, '2023-06-09;09:40:12', '00:00:07', TRUE, FALSE, FALSE, FALSE, 'lucapro'),
(4, '2023-06-10;11:25:45', '00:00:05', TRUE, FALSE, FALSE, FALSE, 'lucapro'),
(5, '2023-06-10;23:12:33', '00:00:12', TRUE, FALSE, FALSE, FALSE, 'paolofootball'),
(6, '2023-06-10;16:34:11', '00:00:30', FALSE, TRUE, TRUE, TRUE,
'mariachiarissima'),
(7, '2023-05-22;20:13:31', '00:00:30', TRUE, FALSE, TRUE, FALSE, 'alzu991'),
(8, '2023-05-26;11:15:15', '00:01:45', FALSE, TRUE, TRUE, TRUE, 'giornogiovanna'),

```



```

(9, '2023-05-27;16:46:11', '00:00:40', TRUE, FALSE, TRUE, TRUE,
'xDarkAngelcraftXx'),
(10, '2023-06-01;22:22:22', '00:01:45', FALSE, TRUE, TRUE, FALSE,
'michelepizza99'),
(11, '2023-05-30;05:13:33', '00:01:45', NULL, TRUE, TRUE, FALSE, 'enzosong'),
(12, '2023-05-31;10:22:35', '00:01:45', NULL, TRUE, FALSE, FALSE, 'alzu991'),
(13, '2023-06-02;14:42:23', '00:01:45', NULL, FALSE, FALSE, FALSE,
'michelepizza99'),
(14, '2023-06-05;08:46:29', '00:01:45', NULL, FALSE, TRUE, FALSE,
'mariachiarissima'),
(15, '2023-06-10;13:37:00', '00:01:45', NULL, TRUE, TRUE, TRUE, 'lucapro'),
(16, '2023-05-15;17:32:12', '00:00:05', TRUE, FALSE, FALSE, FALSE,
'paolofootball'),
(17, '2023-05-18;22:56:57', '00:02:30', FALSE, FALSE, TRUE, FALSE,
'mariachiarissima'),
(18, '2023-06-07;18:59:31', '00:02:30', FALSE, TRUE, TRUE, TRUE, 'lucapro');

/*popolamento utente*/
INSERT INTO utente(nickname, indirizzo_email, nome, cognome, data_di_nascita,
numero_di_telefono, genere) VALUES
('enzosong', 'es@gmail.com', 'vincenzo', 'capaldo', '2001-07-28', '3334445567',
'M'),
('lucapro', 'lp@gmail.com', 'luca', 'donnarumma', '2001-06-12', '3345678907', 'M'),
('paolofootball', 'pf@gmail.com', 'paolo', 'esposito', '2001-06-23', '3457563213',
'M'),
('mariachiarissima', 'mc@gmail.com', 'mariachiara', 'garofalo', '2001-02-28',
'3425366677', 'F'),
('xDarkAngelcraftXx', 'darkcraft@libero.it', 'piero', 'valentino', '1995-05-05',
'4445556677', 'M'),
('giornogiovanna', 'gg@gmail.com', 'giovanna', 'giorno', '2002-12-12',
'3334410987', 'F'),
('michelepizza99', 'mp@libero.it', 'michele', 'ostuni', NULL, NULL, 'M'),
('alzu991', 'alzu@virgilio.it', 'alessandra', 'zuola', NULL, NULL, 'F');

/*popolamento v_i*/
INSERT INTO v_i(id_interazione, id_video) VALUES
(1, 5),
(2, 5),
(3, 1),
(4, 1),
(5, 1),
(6, 1),
(7, 2),
(8, 2),
(9, 2),
(10, 2),
(11, 3),
(12, 3),
(13, 3),

```

```
(14, 3),
(15, 3),
(16, 4),
(17, 4),
(18, 4);
COMMIT;

/*terminazione campagne*/
UPDATE campagna set data_fine = '2023-06-10', raggiungimento_obiettivo = 0.80 WHERE
id = 2;
UPDATE campagna set data_fine = '2023-06-09', raggiungimento_obiettivo = 0.57 WHERE
id = 4;
UPDATE campagna set data_fine = '2023-05-10', raggiungimento_obiettivo = 0.69 WHERE
id = 5;
```

7. Query SQL

Workpackage	Task	Responsabile
WP3	SQL: Query	Donnarumma Luca

7.1. Query con operatore di aggregazione e join: Azienda-visualizzazioni_totali-tempo_medio_di_visualizzazione

Lo scopo di questa query è trovare, per ogni azienda, il numero di visualizzazioni totali e il tempo medio di visualizzazione dei video relativi alle sue campagne. Per farlo si mettono in join 5 tabelle (azienda, campagna, video, v_i e interazione), si seleziona il nome dell'azienda, si conta il numero di interazioni e si fa la media dei tempi di visualizzazione delle varie interazioni. Il risultato è raggruppato per azienda e posto in ordine decrescente in base al numero di visualizzazioni totali.

```
SELECT A.nome, COUNT(I.id) AS visualizzazioni_totali, AVG (I.tempo_di_visualizzazione) AS
tempo_medio_di_visualizzazione
FROM azienda AS A
JOIN campagna AS C ON A.partita_iva = C.partita_iva
JOIN video AS V ON C.id_video = V.id
JOIN v_i ON v_i.id_video = V.id
JOIN interazione AS I ON I.id = v_i.id_interazione
GROUP BY A.nome
ORDER BY visualizzazioni_totali DESC
```

7.2. Query nidificata complessa: aziende associate ai tre utenti con più interazioni

Lo scopo di questa query è capire gli utenti con più interazioni con quali aziende hanno interagito; dunque, si vuole ottenere il numero di interazioni tra le aziende e i 3 utenti che hanno registrato il maggior numero di interazioni complessive. La query interna calcola il numero di interazioni per ciascun utente, ordinandoli in modo decrescente e prelevando soltanto i primi 3. La query esterna unisce questa selezione con le tabelle interazione, v_i, video, campagna e azienda mostrando, nel risultato finale, il nome dell'azienda, il nickname dell'utente e il conteggio delle interazioni, ordinato in modo decrescente per numero di interazioni e, in caso di parità, in ordine alfabetico per il nickname dell'utente.

```
SELECT azienda.nome AS nome_azienza, utenti_interazioni.utente,
COUNT(interazione.nickname_utente) AS numero_interazioni
FROM (
  SELECT interazione.nickname_utente as utente, COUNT(*) AS num_interazioni
  FROM interazione
  GROUP BY interazione.nickname_utente
  ORDER BY num_interazioni DESC
  LIMIT 3
) AS utenti_interazioni
JOIN interazione ON interazione.nickname_utente = utenti_interazioni.utente
```

```
JOIN v_i ON interazione.id = v_i.id_interazione
JOIN video ON video.id = v_i.id_video
JOIN campagna ON campagna.id = video.id_campagna
JOIN azienda ON azienda.partita_iva = campagna.partita_iva
GROUP BY azienda.nome, utenti_interazioni.utente
ORDER BY numero_interazioni DESC, utenti_interazioni.utente
```

7.3. Query insiemistica: Video musicali con almeno 3 visualizzazioni

Lo scopo di questa query è trovare il titolo e il numero di visualizzazioni dei video che hanno un tag 'musica' e un numero di visualizzazioni maggiore o uguale a 3. Utilizzando l'operatore insiemistico intersect, facciamo in modo che solo i video che soddisfano entrambe le condizioni saranno inclusi nel risultato finale.

```
SELECT V.titolo, V.numero_visualizzazioni
FROM video AS V
JOIN v_t ON V.id = v_t.id_video
WHERE v_t.nome_tag = 'musica'
INTERSECT
SELECT V.titolo, V.numero_visualizzazioni
FROM video AS V
WHERE v.numero_visualizzazioni >= 3
```

7.4. Altre query

7.4.1. Operazione 7: tre aziende con maggior successo

Lo scopo di questa query è trovare le tre aziende che hanno raggiunto i risultati più elevati in termini di obiettivi raggiunti. Per farlo si calcola la media di tutti i raggiungimenti obiettivo delle campagne concluse delle aziende. Il risultato è ordinato in modo decrescente per raggiungimento obiettivo medio.

```
SELECT A.nome AS nome_azienza, AVG(C.raggiungimento_obiettivo) AS
raggiungimento_obiettivo_medio
FROM azienda AS A
JOIN campagna AS C ON A.partita_iva = C.partita_iva
WHERE C.raggiungimento_obiettivo IS NOT NULL
GROUP BY nome_azienza
ORDER BY raggiungimento_obiettivo_medio DESC
LIMIT 3
```

8. Viste

Workpackage	Task	Responsabile
WP4	Viste	Esposito Paolo

8.1. Vista riepilogo_video

La vista "riepilogo_video" è stata creata utilizzando una query SQL che combina diverse tabelle per fornire una visione dettagliata dei video con le relative informazioni di riepilogo ricavate dalle loro interazioni.

La vista comprende le colonne id, url, titolo e numero_visualizzazioni della tabella "video".

Utilizzando LEFT JOIN, a quest'ultimi si uniscono gli attributi generati dalle sottoquery, le quali calcolano le informazioni di riepilogo sul video, a partire dalle sue interazioni.

Le sottoquery utilizzate sono:

- La sottoquery likes che calcola il numero totale di like per ogni video.
- La sottoquery condivisioni che calcola il numero totale di condivisioni per ogni video.
- La sottoquery skip che calcola il numero totale di skip per ogni video.
- La sottoquery click che calcola il numero totale di click sui video per ogni video.
- La sottoquery tempo_medio_visualizzazione che calcola il tempo medio di visualizzazione per ogni video.

Utilizzando la funzione COALESCE, le informazioni di riepilogo del video vengono restituite come 0 se sono NULL (per la colonna tempo_medio_visualizzazione viene restituito '00:00:00').

Infine, i risultati vengono ordinati in base all'ID del video (V.id).

```
CREATE VIEW riepilogo_video AS
```

```
SELECT V.id, V.url, V.titolo, V.numero_visualizzazioni,
       COALESCE(likes.numero_like, 0) AS numero_like,
       COALESCE(condivisioni.numero_condivisioni, 0) AS numero_condivisioni,
       COALESCE(skip.numero_skip, 0) AS numero_skip,
       COALESCE(click.numero_click, 0) AS numero_click,
       COALESCE(tempo_medio_visualizzazione.media_tempo_visualizzazione, '00:00:00') AS
       tempo_medio_visualizzazione
```

```
FROM video V
```

```
LEFT JOIN (SELECT count(*) AS numero_like, v_i.id_video AS id_video
           FROM interazione I
           JOIN v_i ON v_i.id_interazione = I.id
           WHERE I.like = true
           GROUP BY v_i.id_video
           ) AS likes ON likes.id_video = V.id
```

```
LEFT JOIN (SELECT count(*) AS numero_condivisioni, v_i.id_video AS id_video
           FROM interazione I
           JOIN v_i ON v_i.id_interazione = I.id
           WHERE I.condivisione = true
           GROUP BY v_i.id_video
           ) AS condivisioni ON condivisioni.id_video = V.id
```

```
LEFT JOIN (SELECT count(*) AS numero_skip, v_i.id_video AS id_video
           FROM interazione I
           JOIN v_i ON v_i.id_interazione = I.id
           WHERE I.click_su_skip = true
           GROUP BY v_i.id_video
           ) AS skip ON skip.id_video = V.id
```

```
LEFT JOIN (SELECT count(*) AS numero_click, v_i.id_video AS id_video
           FROM interazione I
           JOIN v_i ON v_i.id_interazione = I.id
           WHERE I.click_su_video = true
           GROUP BY v_i.id_video
           ) AS click ON click.id_video = V.id
```

```
LEFT JOIN (SELECT AVG(I.tempo_di_visualizzazione) AS media_tempo_visualizzazione, v_i.id_video
           AS id_video
           FROM interazione I
           JOIN v_i ON v_i.id_interazione = I.id
           GROUP BY v_i.id_video
           ) AS tempo_medio_visualizzazione ON tempo_medio_visualizzazione.id_video = V.id
```

```
ORDER BY V.id;
```

8.1.1. Query con Vista: video con numero di visualizzazioni maggiore della media

Questa query seleziona tutte le righe dalla vista `riepilogo_video` in cui il numero di visualizzazioni è maggiore della media del numero di visualizzazioni di tutti i video.

I risultati sono ordinati in base al tempo medio di visualizzazione in ordine decrescente.

```
SELECT *
FROM riepilogo_video
WHERE numero_visualizzazioni > (
    SELECT AVG(numero_visualizzazioni)
    FROM riepilogo_video
)
ORDER BY tempo_medio_visualizzazione DESC
```

8.2. Vista monitoraggio_campagne

La vista "monitoraggio_campagne" è stata creata utilizzando una query SQL che combina diverse tabelle per fornire una visione dettagliata del monitoraggio delle campagne pubblicitarie.

La vista include le seguenti colonne:

- `id_campagna`: Questa colonna rappresenta l'ID univoco di ogni campagna pubblicitaria.
- `nome_campagna`: Questa colonna contiene il nome della campagna pubblicitaria.
- `data_inizio`: Rappresenta la data di inizio della campagna.

- data_fine: Rappresenta la data di fine della campagna.
- budget: Questa colonna contiene il budget assegnato per la campagna.
- id_video: Rappresenta l'ID univoco del video associato alla campagna.
- url_video: Rappresenta l'URL del video associato alla campagna.
- titolo_video: Questa colonna contiene il titolo del video associato alla campagna.
- nome_kpi: Rappresenta il nome del KPI (Indicatore chiave di prestazione) utilizzato per il monitoraggio.
- valore_kpi: Questa colonna contiene il valore del monitoraggio associato al KPI.

La vista viene creata combinando le seguenti tabelle:

- La tabella "campagna" che contiene informazioni generali sulle campagne pubblicitarie, come l'ID della campagna, il nome, la data di inizio, la data di fine e il budget.
- La tabella "video" che contiene informazioni sui video associati alle campagne pubblicitarie, come l'ID, l'URL e il titolo.
- La tabella "monitoraggio" che contiene i dati di monitoraggio, dunque i valori dei KPI per ogni video.
- La tabella "kpi" che contiene informazioni sui KPI utilizzati per il monitoraggio, come il nome del KPI.

La vista è ordinata, in modo crescente, in base all'ID della campagna.

```
CREATE VIEW monitoraggio_campagne AS
SELECT C.id AS id_campagna, C.nome AS nome_campagna, C.data_inizio, C.data_fine, C.budget,
       V.id AS id_video, V.url AS url_video, V.titolo AS titolo_video,
       K.nome AS nome_kpi, M.valore AS valore_kpi
FROM campagna AS C
JOIN video AS V ON C.id_video = V.id
JOIN monitoraggio AS M ON V.id = M.id_video
JOIN kpi K ON M.nome_kpi = K.nome
ORDER BY C.id
```

8.2.1. Query con Vista: Costo per reach

Questa vista può essere utilizzata per calcolare il valore di altri KPI sulla base di quelli già presenti. Ad esempio, si può calcolare il “costo per reach”, cioè il costo per far raggiungere il proprio video ad una singola persona.

La seguente query restituisce il “costo per reach” per ogni campagna conclusa, insieme alle informazioni aggiuntive come il titolo del video e l'URL. Inoltre, il risultato è ordinato, in modo crescente, in base al “costo per reach”.

```
SELECT id_campagna, nome_campagna, titolo_video, url_video, (budget/valore_kpi) AS
costo_per_reach
FROM monitoraggio_campagne
WHERE nome_kpi = 'Reach' AND data_fine IS NOT NULL
ORDER BY costo_per_reach
```

9. Trigger

9.1. Trigger inizializzazione: *v_i* e *proteggi_numero_studenti*

Workpackage	Task	Responsabile
WP1	Trigger inizializzazione/popoloamento database	Garofalo Mariachiara

I seguenti trigger, *v_i* e *proteggi_numero_visualizzazioni*, sono utilizzati per gestire la ridondanza dell'attributo "numero_visualizzazioni" nella tabella video. Infatti, per preservare l'integrità dei dati, questo numero deve sempre coincidere con il numero di interazioni di quel video. Il trigger *v_i* scatta ogni volta che inseriamo, modifichiamo o cancelliamo una riga della tabella *v_i*, la quale contiene le corrispondenze tra le interazioni e i video. La funzione del trigger *v_i* farà l'update del numero di visualizzazioni sul video coinvolto (viene messo un valore simbolico, 1 in questo caso). Dunque, il trigger *v_i* farà scattare l'altro trigger (*proteggi_numero_visualizzazioni*), in quanto esso scatta in caso di inserimento o modifica dell'attributo *numero_visualizzazioni* della tabella video. La funzione *proteggi_numero_visualizzazioni()* conta, dalla tabella *v_i*, il numero di interazioni del video coinvolto e lo mette nella nuova riga che sarà inserita nella tabella video dopo il trigger.

```
create or replace function v_i() returns trigger as $$
begin
    if(new is not null) then
        update video set numero_visualizzazioni=1 where id = new.id_video;
    end if;
    if(old is not null) then
        update video set numero_visualizzazioni=1 where id=old.id_video;
    end if;
    return null;
end
$$ language plpgsql;
```

```
create trigger v_i
after insert or delete or update on v_i
for each row
execute function v_i();
```

```
create or replace function proteggi_numero_visualizzazioni() returns trigger as $$
begin
    select count(*) into new.numero_visualizzazioni
    from v_i
    where id_video = new.id;
    return new;
end
$$ language plpgsql;
```

```
create trigger proteggi_numero_visualizzazioni
before insert or update of numero_visualizzazioni on video
for each row
execute function proteggi_numero_visualizzazioni();
```


9.2. Trigger per vincoli aziendali

Workpackage	Task	Responsabile
WP4	Trigger per vincoli aziendali	Esposito Paolo

9.2.1. Trigger1: verifica_v_i e verifica_update_interazione

Il trigger verifica_v_i, insieme al trigger verifica_update_interazione, implementa i seguenti vincoli aziendali:

- **(RV12)** Data e ora in interazione deve essere \geq Data inizio in campagna.
- **(RV13)** Si possono aggiungere solo interazioni relative a video di campagne correnti.
- **(RV14)** Tempo di visualizzazione in interazione deve essere minore o uguale della durata del video associato all'interazione.
- **(RV15)** Click su skip in interazione deve esserci se il video è con skip e non deve esserci se il video è senza skip.

L'inserimento di un'interazione prevede una transazione in cui viene inserita prima l'interazione e poi la relativa riga in v_i, la quale associa l'interazione al video. Dunque, questo trigger scatta dopo l'inserimento in v_i e controlla che:

- la campagna relativa all'interazione inserita sia una campagna corrente;
- la data dell'interazione inserita sia successiva alla data di inizio della campagna relativa all'interazione;
- il tempo di visualizzazione dell'interazione non superi la durata totale del video;
- se il video è con skip l'interazione deve avere click su skip, altrimenti non deve averlo.

Il trigger verifica_v_i scatta anche nel caso di update della tabella v_i, è diplo after row-level.

Invece, il trigger verifica_update_interazione scatta nel caso di update della tabella interazione. Questo trigger esegue gli stessi controlli del trigger verifica_v_i. Non è possibile implementare un unico trigger perché le variabili per effettuare i controlli vengono prelevate diversamente.

```
CREATE OR REPLACE FUNCTION verifica_v_i() RETURNS TRIGGER AS $$
DECLARE
```

```
    controllo_data_inizio DATE;
    controllo_data_fine DATE;
    data_interazione DATE;
```

```
    durata_interazione TIME;
    durata_video TIME;
```

```
    skip_interazione BOOL;
    skip_video BOOL;
```

```
BEGIN
```

```
    SELECT data_inizio INTO controllo_data_inizio
    FROM campagna
    WHERE id_video = NEW.id_video;
```

```
SELECT data_fine INTO controllo_data_fine
FROM campagna
WHERE id_video = NEW.id_video;

SELECT data_e_ora INTO data_interazione
FROM interazione
WHERE id = NEW.id_interazione;

IF (controllo_data_fine IS NOT NULL) THEN
    RAISE EXCEPTION 'Il video associato all''interazione non appartiene ad una campagna
    corrente.';
END IF;

IF (data_interazione < controllo_data_inizio) THEN
    RAISE EXCEPTION 'La data dell''interazione deve essere maggiore o uguale alla data
    di inizio della campagna.';
END IF;

SELECT tempo_di_visualizzazione INTO durata_interazione
FROM interazione
WHERE interazione.id = NEW.id_interazione;

SELECT durata INTO durata_video
FROM video
WHERE video.id = NEW.id_video;

IF (durata_interazione > durata_video) THEN
    RAISE EXCEPTION 'Il tempo di visualizzazione deve essere minore o uguale alla durata
    del video.';
END IF;

SELECT skip INTO skip_video
FROM video
WHERE video.id = NEW.id_video;

SELECT click_su_skip INTO skip_interazione
FROM interazione
WHERE interazione.id = NEW.id_interazione;

IF (skip_interazione IS NULL AND skip_video = TRUE) THEN
    RAISE EXCEPTION 'Questa interazione deve avere click su skip ';
END IF;

IF (skip_interazione IS NOT NULL AND skip_video = FALSE) THEN
    RAISE EXCEPTION 'Questa interazione non può avere click su skip ';
END IF;

RETURN NULL;
```

```
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER verifica_v_i
AFTER INSERT OR UPDATE ON v_i
FOR EACH ROW
EXECUTE FUNCTION verifica_v_i();
```

```
CREATE OR REPLACE FUNCTION verifica_update_interazione() RETURNS TRIGGER AS $$
```

```
DECLARE
```

```
    video_associato INTEGER;
```

```
    controllo_data_inizio DATE;
```

```
    controllo_data_fine DATE;
```

```
    durata_video TIME;
```

```
    skip_video BOOL;
```

```
BEGIN
```

```
    SELECT id_video INTO video_associato
```

```
    FROM v_i
```

```
    WHERE v_i.id_interazione = NEW.id;
```

```
    SELECT data_inizio INTO controllo_data_inizio
```

```
    FROM campagna
```

```
    WHERE campagna.id_video = video_associato;
```

```
    SELECT data_fine INTO controllo_data_fine
```

```
    FROM campagna
```

```
    WHERE campagna.id_video = video_associato;
```

```
    IF (controllo_data_fine IS NOT NULL) THEN
```

```
        RAISE EXCEPTION 'Il video associato all''interazione non appartiene ad una campagna
        corrente.';
```

```
    END IF;
```

```
    IF (NEW.data_e_ora < controllo_data_inizio) THEN
```

```
        RAISE EXCEPTION 'La data dell''interazione deve essere maggiore o uguale alla data
        di inizio della campagna.';
```

```
    END IF;
```

```
    SELECT durata INTO durata_video
```

```
    FROM video
```

```
    WHERE video.id = video_associato;
```

```
IF (NEW.tempo_di_visualizzazione > durata_video) THEN
    RAISE EXCEPTION 'Il tempo di visualizzazione deve essere minore o uguale alla durata
    del video.';
END IF;

SELECT skip INTO skip_video
FROM video
WHERE video.id = video_associato;

IF (NEW.click_su_skip IS NULL AND skip_video = TRUE) THEN
    RAISE EXCEPTION 'Questa interazione deve avere click su skip';
END IF;

IF (NEW.click_su_skip IS NOT NULL AND skip_video = FALSE) THEN
    RAISE EXCEPTION 'Questa interazione non può avere click su skip';
END IF;

RETURN NULL;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER verifica_update_interazione
AFTER UPDATE ON interazione
FOR EACH ROW
EXECUTE FUNCTION verifica_update_interazione();
```

9.2.2. Trigger2: cardinalita_tag

Il seguente trigger (cardinalita_tag) viene implementato per gestire la cardinalità minima della tabella tag (1,N). Il trigger scatta ogni volta che inseriamo una riga nella tabella tag e ogni volta che cancelliamo una riga o modifichiamo l'attributo nome_tag della tabella v_t, la quale contiene le corrispondenze tra le i video e i tag. La funzione del trigger cardinalita_tag() controlla se esistono dei tag nella tabella tag che non ci sono anche nella tabella v_t. In caso affermativo, la funzione lancia un'eccezione con un messaggio di errore.

```
CREATE OR REPLACE FUNCTION cardinalita_tag() RETURNS TRIGGER AS $$
BEGIN
    IF (EXISTS (SELECT nome
                FROM tag
                WHERE nome NOT IN (SELECT nome_tag FROM v_t))
    ) THEN RAISE EXCEPTION 'Non è possibile avere un tag che non è associato a nessun video';
    END IF;

    RETURN NULL;
END
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER cardinalita_tag
AFTER INSERT ON tag
FOR EACH ROW
EXECUTE FUNCTION cardinalita_tag();
```

```
CREATE TRIGGER cardinalita_tag
AFTER DELETE OR UPDATE OF nome_tag ON v_t
FOR EACH ROW
EXECUTE FUNCTION cardinalita_tag();
```

9.2.3. Trigger3: cardinalita_azienda

Il seguente trigger (cardinalita_azienda) viene implementato per gestire la cardinalità minima della tabella azienda (1,N). Il trigger scatta ogni volta che inseriamo una riga nella tabella azienda e ogni volta che cancelliamo una riga o modifichiamo l'attributo partita_iva della tabella campagna, la quale contiene le campagne associate alle aziende. La funzione del trigger cardinalita_azienda() controlla se esistono delle aziende nella tabella azienda che non ci sono anche nella tabella campagna. In caso affermativo, la funzione lancia un'eccezione con un messaggio di errore.

```
CREATE OR REPLACE FUNCTION cardinalita_azienda() RETURNS TRIGGER AS $$
BEGIN
    IF (EXISTS (SELECT partita_iva
                FROM azienda
                WHERE partita_iva NOT IN (SELECT partita_iva FROM campagna))
    ) THEN RAISE EXCEPTION 'Non è possibile avere un'azienda che non è associata a nessuna
        campagna';
    END IF;

    RETURN NULL;
END
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER cardinalita_azienda
AFTER INSERT ON azienda
FOR EACH ROW
EXECUTE FUNCTION cardinalita_azienda();
```

```
CREATE TRIGGER cardinalita_azienda
AFTER DELETE OR UPDATE OF partita_iva ON campagna
FOR EACH ROW
EXECUTE FUNCTION cardinalita_azienda();
```

9.2.4. Trigger4: cardinalita_utente

Il seguente trigger (cardinalita_utente) viene implementato per gestire la cardinalità minima della tabella utente (1,N). Il trigger scatta ogni volta che inseriamo una riga nella tabella utente e ogni volta che cancelliamo una riga o modifichiamo l'attributo nickname_utente della tabella interazione, la quale contiene le interazioni associate agli utenti. La funzione del trigger cardinalita_utente() controlla se esistono degli utenti nella tabella utente che non ci sono anche nella tabella interazione. In caso affermativo, la funzione lancia un'eccezione con un messaggio di errore.

```
CREATE OR REPLACE FUNCTION cardinalita_utente() RETURNS TRIGGER AS $$
BEGIN
    IF (EXISTS (SELECT nickname
                FROM utente
                WHERE nickname NOT IN (SELECT nickname_utente FROM interazione))
    ) THEN RAISE EXCEPTION 'Non è possibile avere un utente che non è associato a nessuna
        interazione';
    END IF;

    RETURN NULL;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER cardinalita_utente
AFTER INSERT ON utente
FOR EACH ROW
EXECUTE FUNCTION cardinalita_utente();

CREATE TRIGGER cardinalita_utente
AFTER DELETE OR UPDATE OF nickname_utente ON interazione
FOR EACH ROW
EXECUTE FUNCTION cardinalita_utente();
```