TrovaLO

**1) Analisi dei requisiti**

realizzare una base di dati per un’applicazione chiamato(TROVALO) che permette agli utenti registrati di cercare le proprie documenti persi(o qualsiasi articolo che contiene un identificativo),tiene traccia anche delle impiegati che supervisiono sia gli utenti sia gli articoli persi presso ciascun regione,sono raggruppati per regione che ha diversi posti per la consegna degli articoli e verifica del possesso.

-per ciascun UTENTE venga memorizzato il nome , cognome,username che deve essere unico ,codice fiscale,hashing del password ,email , cellulare ,regione e il numero delle ricerca con hanno terminato con successo (target).

-e per ogni MIS-ITEM si memorizza nome del proprietario,cognome,suo codice fiscale ,descrizione dell’articolo ,altri id(se esiste un altro identificativo)e regione .

-ed ogni RESPONSABILE ha il suo nome ,cognome,id(unico),stipendio,data di nascita,e il numero della regione in cui monitora sia gli suoi utenti sia mis\_item.

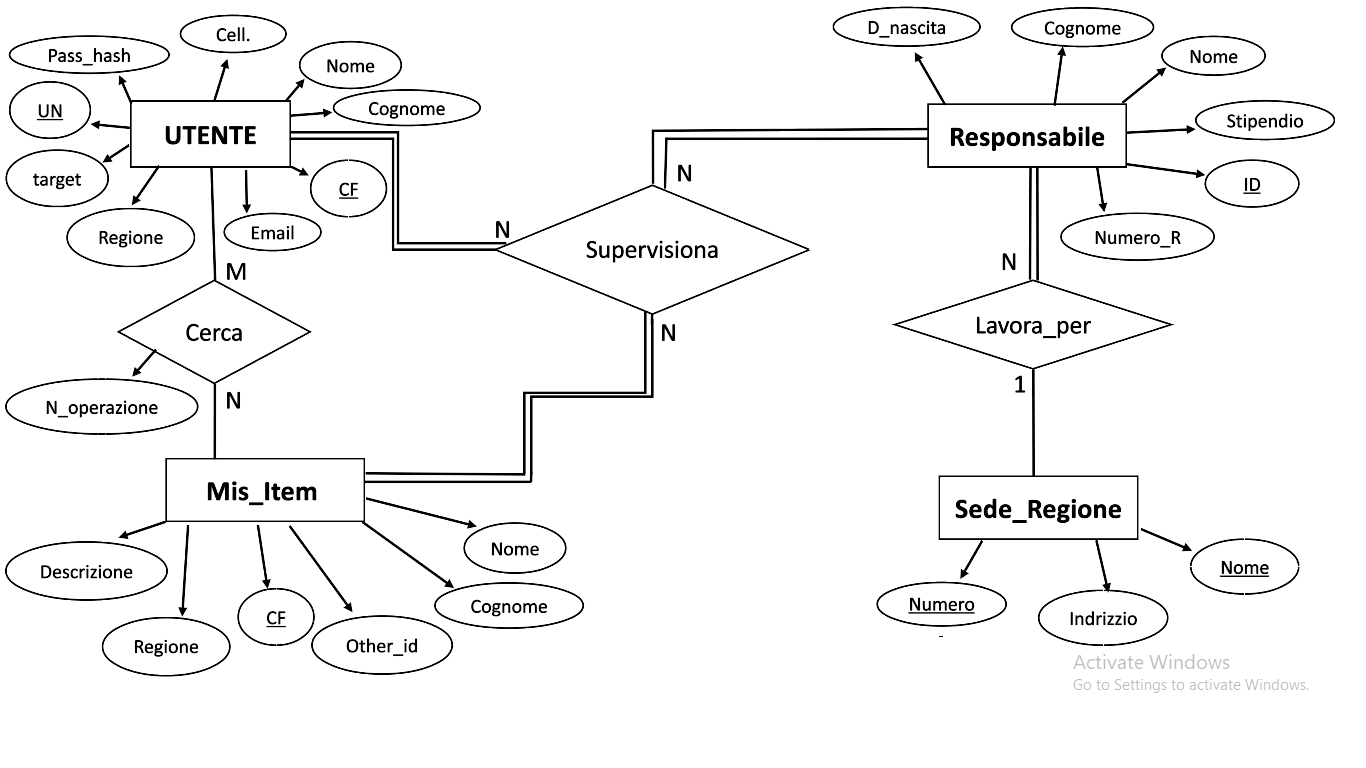
-SEDE\_ REGIONE memorizza il nome di ciascun regione,l’indrizzio della sede per la consegna ed ogni regione viene identificata da un numero es esempi Calabria = ‘1’,Emiglia romgna = ’2’ ecc.

-gli UTENTI possono cercare o meno qualsiasi numero MIS-ITEM ---> le MIS-ITEM si possono essere cercati o meno ,quando un utente trova un articolo, verra’ generato un numero alla ricerca(rapporto di cardinalita M:N e con vincolo di participazione parziali entrambi).

-E I RESPONSABILI si devono fare parte di un SEDE\_REGIONE ---> ancora ci sono sede che stanno cercando delle persone (rapporto di cardinalita 1:N e con vincolo di participazione parziali lato SEDE\_REGIONE ed totale lato SEDE\_REGIONE).

- RESPONSABILI di ciascun regione devono supervisionare sia un no. di UTENTI e no. di MIS-ITEM (rapporto di cardinalita N-arie e con vincolo di participazione totale entrambi) .

**2)Progettazione dello schema ER**



**3) Schema Relazionale**

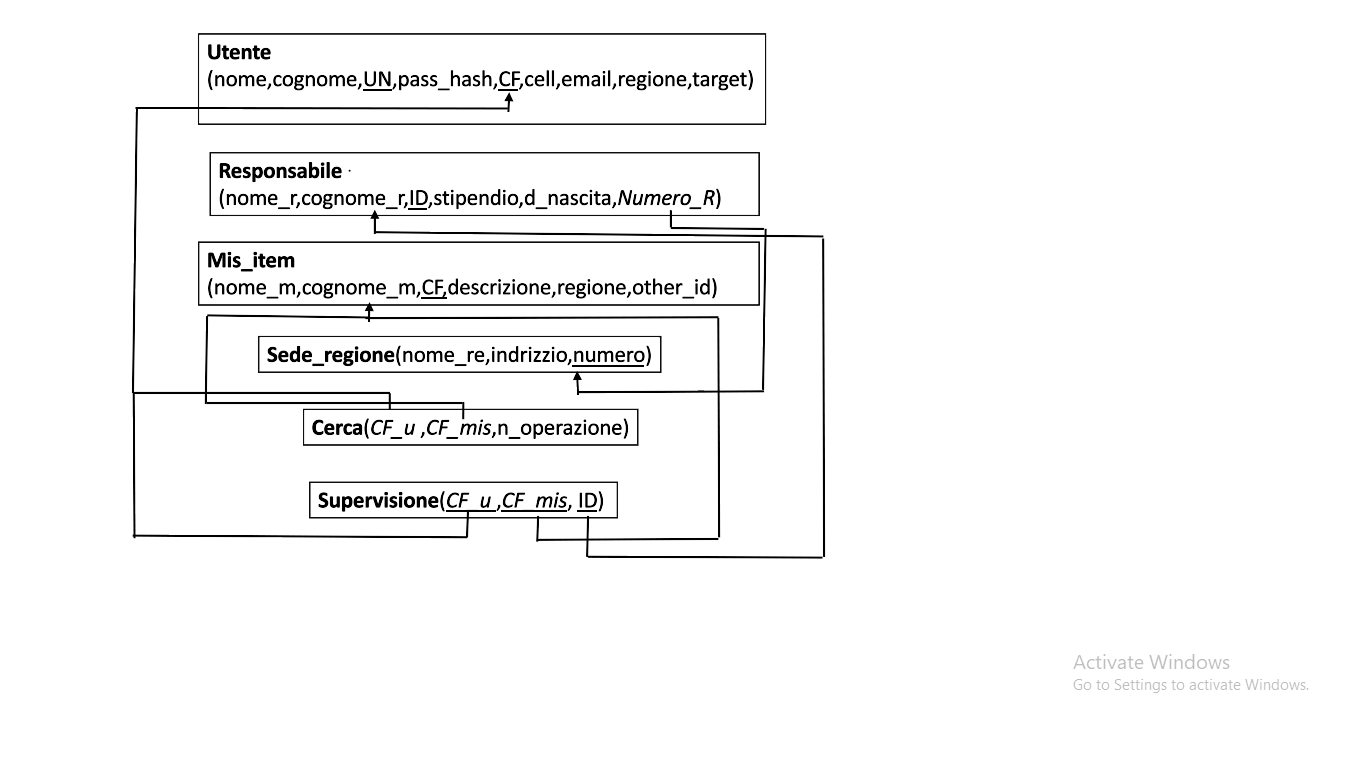
A)

Schema relazionale finale

* **Utente** (nome,cognome,UN,hash\_pass,CF,cell,email,regione,target)
* **Responsabile**(nome,cognome,ID,stipendio,d\_nascita,*Numero\_R*)
* **Mis\_item**(nome,cognome,CF,descrizione,regione,other\_id)
* **Sede\_regione**(nome,indrizzio,numero)
* **Cerca**(*CF\_u* ,*CF\_mis*,n\_operazione)
* **Supervisione**(*CF\_user* ,*CF\_misitem*, *ID\_R*)

vincoli di integrità referenziale

* Responsabile.Numero\_R ---> Sede\_regione.numero
* Cerca.CF\_u ---> Utente.CF
* Cerca.CF\_mis ---> Mis\_item.CF
* Supervisione.CF\_User ---> Utente.CF
* Supervisione.CF\_misitem ---> Mis\_item.CF
* Supervisione.ID\_R ---> Responsabile.ID



B) Normalizzazione fino alla 3NF

* 1NF: lo schema è in 1NF in quanto non sussistono attributi multivalore, composti o loro combinazione.
* 2NF: Per le tabelle con una chiave primaria composta da un solo attributo il test è inutile **(Utente,Responsabile,Mis\_item,Sede\_regione),**Per **Cerca** l’attributo non-primo N\_operazione dipende funzionalmente in modo completo dalla chiave primaria .
* 3NF: Tutte le relazioni sono in 3NF in quanto non sussistono DF transitive.

C)la schema normalizzata e la stessa visto che non sono modifiche

**4) Creazione del database**

il file trovalo.sql

**5)Interrogazione SQL**

1)Per ciascuno dei responsabili di ogni regione visualizza il nome degli utente sue cognomi chi li supervisono e che hanno target piu’ di 2 e la descrizione degli articoli persi .il risultato ha come intestazioni(ID,client\_name, ,client\_surname,description,numero\_r) e si deve ordinare secondo regione in modo crescente .

SOL:

**use** trovalo;

**SELECT**

ID, utente.nome AS ‘client\_name’, utente.cognome AS ‘client\_surname’ , descrizione AS ‘description’, numero\_r

**FROM**

mis\_item NATURAL JOIN utente JOIN

responsabile ON Numero\_R = regione

**WHERE**

target >= 2

**GROUP BY** id , numero\_r

**ORDER BY** numero\_r;

2)

Visualizza la regione che ha il numero di articoli persi massimo e la regione che ha numero di articoli persi minimo ,si deve visualizzare nome della regione e numero .

Sol:

**use** trovalo;

**SELECT** distinct

sede\_regione.nome AS regione\_max,

numero

**FROM**

sede\_regione JOIN mis\_item ON numero = regione

**WHERE**

numero IN

(**SELECT** MAX(regione)

**FROM** mis\_item)

**union**

**select** distinct

sede\_regione.nome AS regione\_min,

numero

**FROM**

sede\_regione JOIN mis\_item ON numero = regione

**WHERE**

numero IN

(SELECT min(regione)

FROM mis\_item);

Commento : la prima soluzione in nero quello che soddisfa la richiesta invece la seconda soluzione in rosso mi da un errore(Error Code: 1054. Unknown column 'n\_regione\_max' in 'IN/ALL/ANY subquery),secondo me dovrebbe andare bene !

**SELECT**

sede\_regione.nome AS regione\_max,

numero AS 'n\_regione\_max',

sede\_regione.nome AS regione\_min,

numero AS 'n\_regione\_min'

**FROM**

sede\_regione JOIN mis\_item ON numero = regione

**WHERE**

(n\_regione\_max,n\_regione\_min) IN

(**SELECT**

MAX(regione), MIN(regione)

FROM mis\_item);

3)

Si trovino nome e cognome dei proprietari che hanno degli articoli persi le descrizione degli articoli e che non sono registrati come utente per l’applicazione.

SOL:

**select** nome,cognome

**from** mis\_item

**where** not exists(**select** \*

**from** utente

**where** mis\_item.CF = utente.CF);

4)

per ciascun regione in cui hanno piu’ di 2 responsabile che hanno stipendio tra 1000 e 2000 visualizza nome di regione come (“Region\_name”)e l’indrizzio della zona di consegna e il numero dei responsabili

SOL:

**select** sede\_regione.nome AS 'Region\_name',indrizzio,count(\*) AS 'num\_responsabile'

**from** responsabile join sede\_regione on numero\_r = numero

**where** stipendio between 1000 and 2000

**group by** sede\_regione.nome,indrizzio

**having** count(\*) > 2;

\*\*Aggiuntivo\*\*

Creare un view INFO\_REGIONE che visualizza per ciascun regione il nome ed il numero ed il numero totale dei responsabile ed la somma dei suoi stipendi

SOL:

**Create** view info\_regione(Region\_name,Regione\_num,num\_responsabile,somma\_stipendio) AS

**select** sede\_regione.nome,numero,count(\*),sum(stipendio)

**from** sede\_regione join responsabile on numero\_r=numero

**group by** sede\_regione.nome,numero;