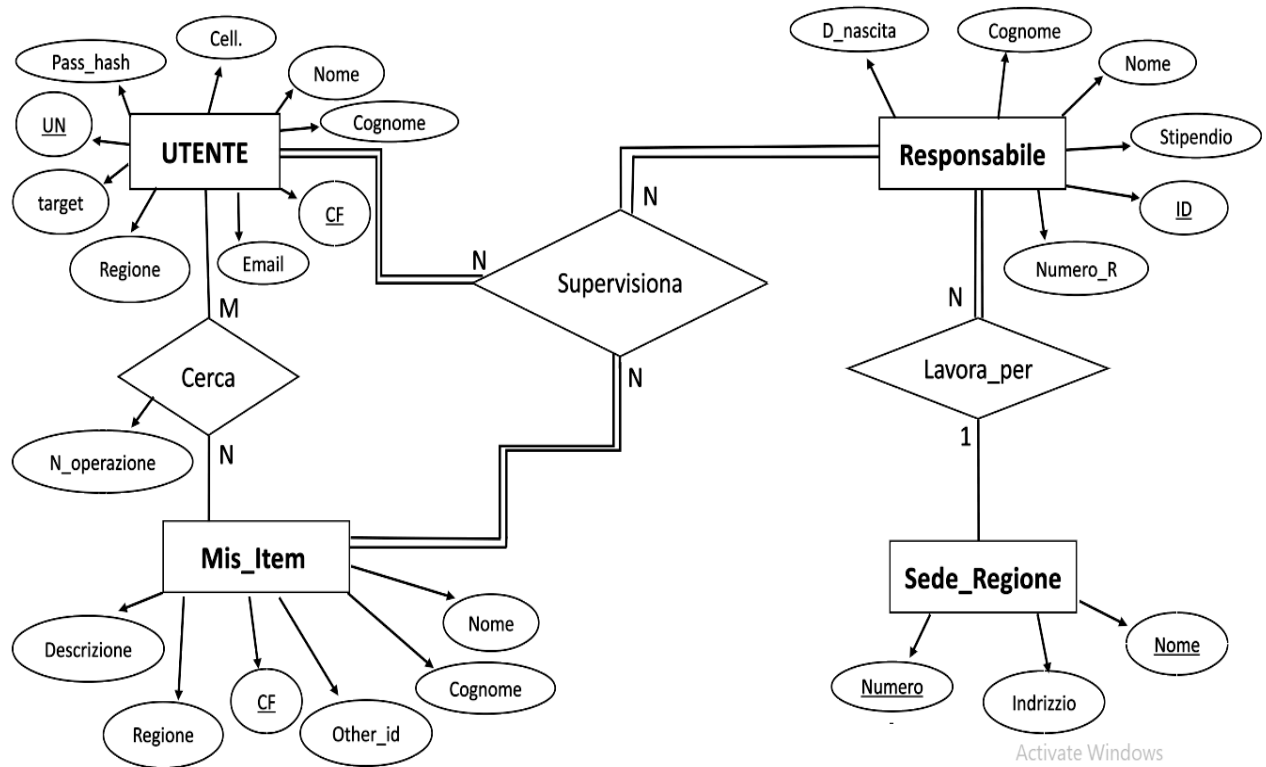


TrovaLO

1) Analisi dei requisiti

In Editing fase

2)Progettazione dello schema ER



Activate Windows
Go to Settings to activate Windows.

3) Schema Relazionale

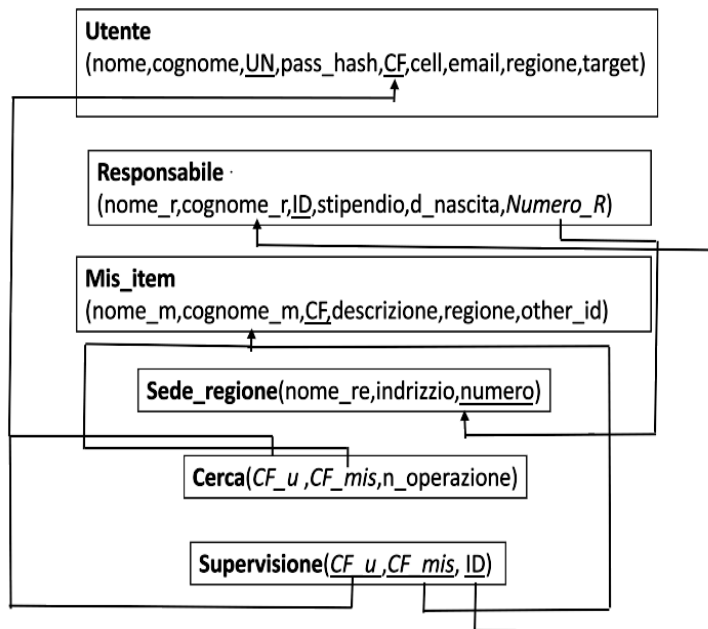
A)

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_region**(nome,indirizzo,numero)
- **Cerca**(*CF_u* ,*CF_mis*,n_operazione)
- **Supervisione**(CF_user,CF_misitem, ID_R)

vincoli di integrità referenziale

- Responsabile.Numero_R ---> Sede_region.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



Activate Windows
Go to Settings to activate Windows.

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 supervisiono 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'indirizzo della zona di consegna e il numero dei responsabili

SOL:

```
select sede_regione.nome AS 'Region_name',indirizzo,count(*) AS  
'num_responsabile'
```

```
from responsabile join sede_regione on numero_r = numero
where stipendio between 1000 and 2000
group by sede_regione.nome,indirizzo
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;
```