

SIMULAZIONE DI PROVA SCRITTA

ESAME DI STATO DI ISTITUTO TECNICO 2016/17

Indirizzo: INFORMATICA E TELECOMUNICAZIONI (Nuovo ordinamento)

Articolazione: INFORMATICA

Tema di: INFORMATICA

Tipologia C

Il candidato svolga la prima parte della prova e risponda ad almeno due tra i quesiti proposti nella seconda parte

Prima Parte

L'officina di autoriparazioni di dimensioni medio-grandi "Alfa", situata in una grande città italiana, già dotata di un proprio sito web, intende ampliarne le funzionalità implementando una applicazione che permetta ai propri clienti di poter controllare online lo stato delle operazioni di assistenza effettuate sui propri veicoli.

Ogni operazione di assistenza si apre con l'accettazione di un veicolo in una determinata data e si chiude con la riconsegna – della quale viene memorizzata la data prevista e poi la data effettiva – del veicolo al cliente. Ogni operazione può comprendere interventi di vario tipo; ogni singolo tipo di intervento è chiaramente identificato e prevede un tempo di lavorazione standard espresso in ore. Durante l'operazione di assistenza sono impiegate diverse quantità di prodotti appartenenti a diverse categorie (parti di ricambio, prodotti lubrificanti o altri consumabili). La fatturazione delle operazioni di assistenza avviene pertanto in base agli interventi effettuati (e ai relativi tempi di lavorazione) e ai prodotti impiegati (con le relative quantità).

Ciascun cliente, i cui dati vengono comunque memorizzati nel database per permettere l'emissione della fattura, può richiedere la registrazione al sito mediante username e password. Dei veicoli occorre memorizzare le informazioni significative.

Il cliente registrato potrà:

- visualizzare lo storico delle operazioni di assistenza effettuate sui propri veicoli
- visualizzare lo stato della operazione di assistenza di un veicolo di cui viene fornita la targa

Il candidato, fatte le opportune ipotesi aggiuntive, sviluppi

- (a) un'analisi della realtà di riferimento individuando le possibili soluzioni e scelga quella che a suo motivato giudizio è la più idonea a rispondere alle specifiche indicate
- (b) lo schema concettuale della base di dati
- (c) lo schema logico della base di dati
- (d) la definizione in linguaggio SQL di un sottoinsieme delle relazioni della base di dati in cui siano presenti alcune di quelle che contengono vincoli di integrità referenziale e/o vincoli di dominio, laddove presenti.
- (e) implementi in linguaggio SQL le seguenti interrogazioni:
 1. numero delle auto che sono attualmente in officina per operazioni di assistenza
 2. visualizzare le informazioni delle operazioni di assistenza con il maggior numero di interventi effettuati al suo interno
 3. visualizzare i dati dei clienti che non si sono ancora registrati
 4. le categorie di prodotti con prezzo medio superiore ad un valore dato

- (f) la progettazione dell'interfaccia della pagina web che permetta ad un cliente registrato di svolgere le operazioni previste
- (g) la codifica in un linguaggio a scelta di un segmento significativo dell'applicazione Web che consente l'interazione con la base di dati.

Seconda parte

Il candidato risponda ad almeno due quesiti a scelta tra quelli sotto riportati

1. In relazione al tema proposto nella prima parte, descriva in che modo è possibile integrare lo schema concettuale sopra sviluppato in modo da poter gestire la possibilità da parte di un cliente registrato di fornire un feed-back che esprima il suo livello di soddisfazione al servizio ricevuto esprimendo un commento ed un voto da 1 a 5
2. In relazione al tema proposto nella prima parte esprimere in sql la query che permetta di calcolare il costo totale da fatturare per una operazione di assistenza
3. Illustri le regole di derivazione che permettono di ottenere uno schema relazionale a partire da uno schema concettuale.
4. Illustrare, anche servendosi di esempi, le tipologie di operazioni che si possono effettuare all'interno di un DBMS relazionale

Durata massima della prova: 5 ore.

È consentito soltanto l'uso di manuali tecnici (references riportanti solo la sintassi, non guide) dei linguaggi utilizzati.

SOLUZIONE PRIMA PARTE

(a) Analisi

Il testo fa riferimento ad una autofficina che intende ampliare i servizi offerti dal proprio sito web di cui è già dotata ma che evidentemente ha una funzione puramente informativa. E' richiesta la realizzazione di un sistema informatico che permetta, oltre che di gestire le operazioni di assistenza ad autoveicoli, ai clienti che lo desiderano previa registrazione al sito, di controllare online lo stato della operazione di assistenza in corso oppure di visualizzare lo storico degli interventi effettuati sui propri autoveicoli.

In particolare una operazione di assistenza può comprendere diversi tipi di interventi e, in ciascuno di essi possono essere utilizzati prodotti appartenenti a varie categorie.

Per poter soddisfare queste richieste l'applicazione che si deve realizzare è una Web Application che permetta sia la gestione dei dati dei clienti, delle operazioni di assistenza, degli interventi, dei prodotti da parte del personale dell'officina, sia la possibilità di consultare online le informazioni riguardanti le operazioni relative ai veicoli da parte dei clienti registrati.

L'applicazione deve essere accessibile da vari dispositivi (pc, tablet, smartphone).

Per quanto riguarda gli strumenti software da utilizzare avremo:

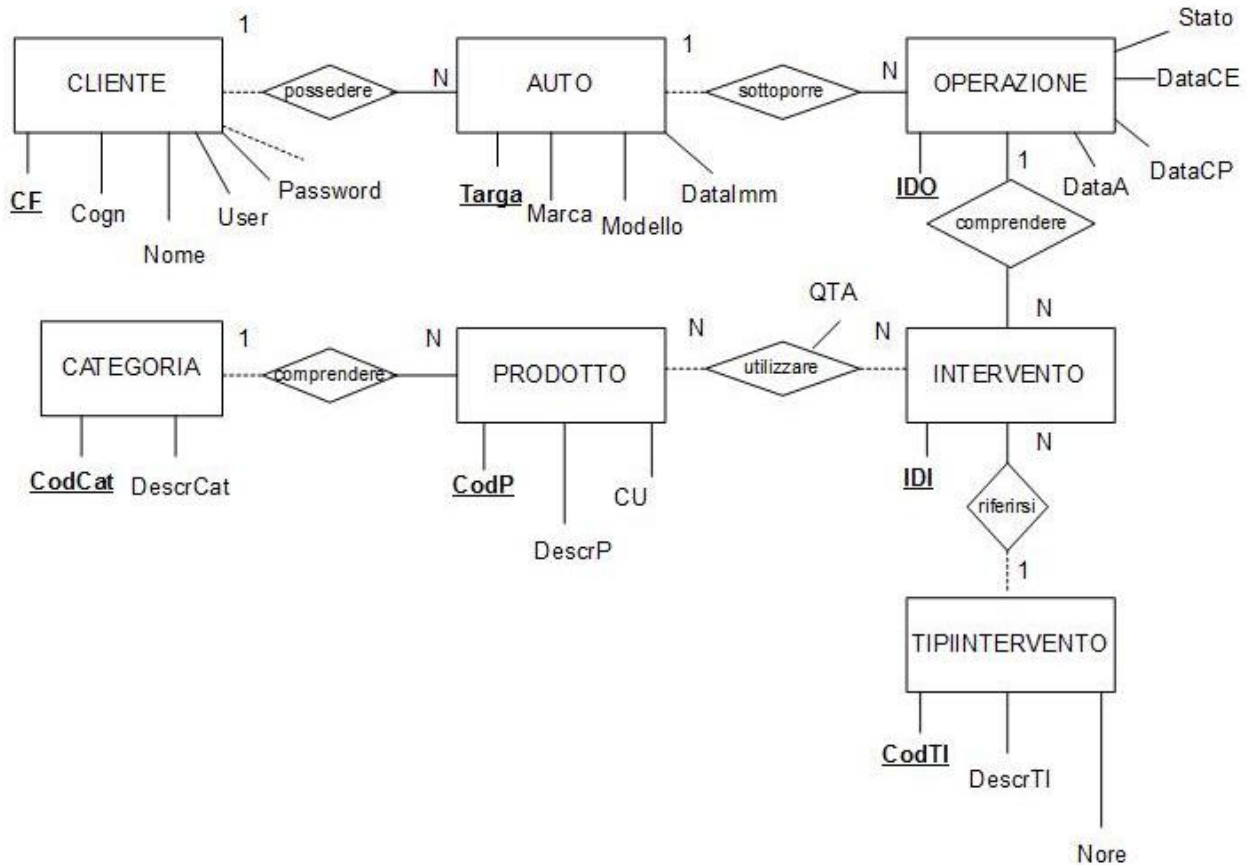
- MySQL come Dbms relazionale per la gestione del database
- Apache come Web server
- Php come linguaggio di scripting lato server per l'interazione con il data base
- HTML, CSS, Javascript per implementare le pagine Web

(b) Schema concettuale della base di dati

Ipotesi di lavoro

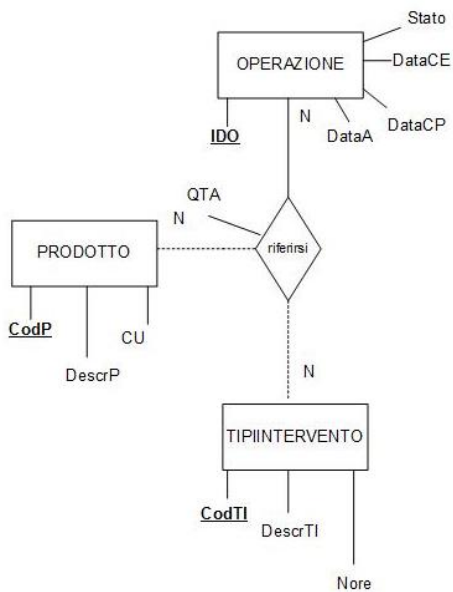
- Per quanto riguarda gli attributi si sono presi in considerazione quelli espressamente indicati dal testo e altri ritenuti significativi soprattutto relativamente al cliente e all'auto.
In particolare anziché considerare le entità Modello e Marca sono stati inseriti altrettanti attributi in Auto per non appesantire eccessivamente le interrogazioni al database anche se sarebbe opportuno per garantire l'integrità dei dati. Si è invece ritenuto opportuno utilizzare l'entità Categoria relativamente ai prodotti come ci suggerisce la n.4 del punto e).
- Il costo all'ora della manodopera non viene fornito dal testo. Per poter essere utilizzato esso si suppone fisso e noto (ad esempio pari a 30€)
- Non si prende in considerazione l'aspetto relativo alla gestione delle fatture emesse ai clienti il cui importo è comunque possibile calcolare in base agli interventi effettuati e ai prodotti utilizzati

Lo schema E/R che si propone è il seguente:



Osservazione:

Tra Operazioni, TipiIntervento e Prodotto esiste una associazione ternaria come mostrato nella seguente porzione di schema E/R



Lo schema è stato trasformato con l'introduzione dell'entità Intervento come mostrato nello schema completo

Entità Cliente: contiene le informazioni relative ai clienti dell'officina. Come chiave primaria viene utilizzato il Codice Fiscale anche se si potrebbe utilizzare un codice interno e considerare CF come attributo comunque univoco.

Del cliente sono stati indicati gli attributi essenziali ma sicuramente dovrebbero essere presi in considerazione l'indirizzo ed i dati di contatto (numero telefonico ed email).

Gli attributi User e Password non sono obbligatori in quanto il cliente potrebbe non essere interessato alla registrazione online. Entrambi, quindi, sono di default nulli e valorizzati solo al momento della registrazione (ovviamente user deve essere univoco).

Entità Auto: contiene le informazioni delle auto sottoposte ad assistenza nell'autofficina. Si sono scelti come attributi significativi la marca, il modello, la data di immatricolazione. Si potrebbero aggiungere altri dati quali ad esempio i km effettuati dalla vettura al momento della accettazione, la cilindrata, il tipo di alimentazione, se è in garanzia oppure no.

Entità Operazione: contiene le informazioni relative alle operazioni di assistenza. Gli attributi sono in pratica indicati dal testo ed in particolare

DataA	data accettazione che rappresenta la data in cui l'auto viene presa in consegna dall'autofficina
DataCP	data consegna presunta che rappresenta la data in cui presumibilmente l'assistenza è conclusa. La data è indicativa in base agli interventi che si pensa debbano essere effettuati
DataCE	data consegna effettiva che potrebbe essere diversa da quella presunta per esempio per ulteriori problemi emersi durante l'assistenza. L'attributo conterrà di default il valore NULL e sarà valorizzato con la data della consegna effettiva (potrebbe quindi coincidere con la data di consegna presunta)
Stato	indicatore dello stato in cui si trova l'operazione di assistenza. Conterrà dei valori predefiniti quali ad esempio NON AVVIATO/IN CORSO/CONCLUSO ecc

Entità TipiIntervento: contiene le informazioni sui tipi di interventi che l'autofficina è in grado di effettuare quali ad esempio Controllo frizione, Sostituzione pastiglie freni ecc.

L'attributo Nore indica il numero di ore standard previste per l'intervento cui fa riferimento. Da osservare che il testo parla di durata standard espressa in ore. Per semplicità si ipotizza che non vi siano interventi di durata inferiore all'ora. Se così non fosse occorrerebbe esprimere la durata in minuti oppure prevedere due attributi, il primo per il numero di ore ed il secondo per il numero di minuti.

Entità Prodotto: contiene le informazioni sui prodotti che l'autofficina può utilizzare negli interventi

Entità Categorie: contiene le informazioni relative alle categorie di prodotti utilizzate (ad esempio: Lubrificanti, Batterie ecc.)

Entità Intervento: questa entità di fatto non contiene altro che l'attributo chiave primaria IDI. Di fatto Interventi è un'entità inserita per spezzare l'associazione N-N che sussiste tra TipiIntervento e Operazione creando due associazioni 1-N. Questo ci consente di realizzare una associazione tra Prodotto e Intervento (di tipo N-N) rendendo più semplice lo schema E/R. In pratica se TipiIntervento classifica le prestazioni che l'autofficina è in grado di effettuare, Intervento rappresenta l'Intervento effettuato, la prestazione effettuata.

Non si ritiene di inserire altri attributi che potrebbero comunque essere presenti quali ad esempio la data dello specifico intervento o il numero di ore extra richieste.

Associazioni e lettura dello Schema E/R

Un cliente **può** possedere una o più auto. Un'auto **deve** appartenere ad un cliente.

Un'auto **può** essere sottoposta ad una o più operazioni (di assistenza). Una operazione (di assistenza) **deve** riferirsi ad una sola auto.

Una operazione (di assistenza) **deve** contenere uno o più interventi. Un intervento **deve** riferirsi ad una sola operazione (di assistenza)

Un intervento **deve** riferirsi ad un tipo di intervento, un tipo di intervento **può** trovarsi in uno o più interventi.

Un prodotto **può** essere utilizzato in uno o più interventi, un intervento **può** impiegare uno o più prodotti. La quantità di prodotto utilizzata è attribuito dell'associazione.

Una categoria **può** comprendere uno o più prodotti. Un prodotto **deve** appartenere ad una categoria.

Vincoli

In Cliente

User → univoco

Password → il valore della password viene criptato con l'algoritmo MD5 per garantire una maggiore sicurezza

In Auto

DataImm (data di immatricolazione) → DataImm < data corrente

In Operazione

DataA (data di accettazione) → DataCP ≥ DataA

DataCP (data di consegna presunta)

DataCE (Data di consegna effettiva) → Valore di default NULL
DataCE ≥ DataA

In Prodotti

Cu → Cu > 0

In Tipi Interventi

Nore → Nore ≥ 1

(c) Schema logico della base di dati

Applicando le opportune regole di derivazione si ottiene il seguente

SCHEMA DEL MODELLO RELAZIONALE

Clienti(**CF**, Cogn, Nome, User, Password,

Auto(**Targa**, Modello, Marca, DataImm, *CF*)

Operazioni(**IDO**, DataA, DataCP, DataCE, Stato, *Targa*)

Interventi(**IDI**, *IDO*, *CodTI*)

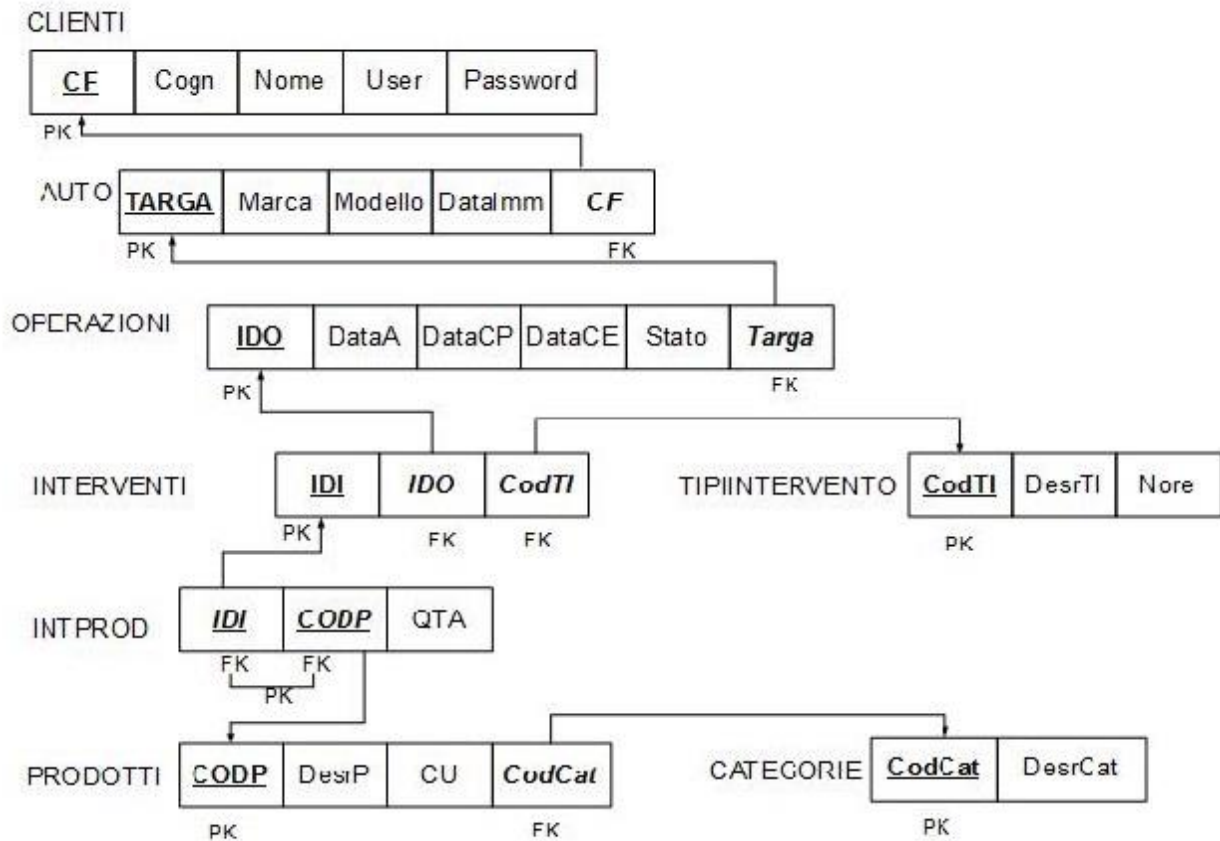
TipiIntervento(**CodTI**, DescrTI, Nore)

Categorie(**CodCat**, DescrCat)

Prodotti(**CodP**, DescrP, Cu, *CodCat*)

Intprod(**IDI**, **CodP**, Qta)

Esplicitazione dei collegamenti



(d) Definizione in linguaggio DDL delle relazioni del Data Base

Nome del database da realizzare in MySql è officina. Tutte le tabelle sono di tipo INNODB.

```

Create database officina;
Use officina;
create table Clienti
( CF Char(16) primary key not null,
  Cogn VarChar(20) not null,
  Nome VarChar(20) not null,
  User VarChar(20) UNIQUE,
  Password Char(32)
);
create table Auto
( Targa varChar(10) primary key not null,
  CF Char(16) not null,
  Modello VarChar(25) not null,
  Marca VarChar(25) not null,
  DataImm Date not null,
  Foreign key(CF) references Clienti(CF)
);

```

```
create table Operazioni
( IDO int(20) primary key not null auto_increment ,
  dateA date not null,
  dateCP date not null,
  dateCE date,
  Stato VarChar(20),
  Targa VarChar(10) not null,
  foreign key(Targa) references Auto(Targa)
);

create table TipiIntervento
( CodTI VarChar(4) primary key not null,
  DescrTI VarChar(500) not null,
  Nore int(20) not null
);

create table Interventi
( IDI int(20) primary key not null auto_increment,
  IDO int(20) not null,
  CodTI VarChar(4) not null,
  foreign key(IDO) references Operazioni(IDO),
  foreign key(CodTI) references TipiIntervento
(CodTI)
);

create table Categorie( CodCat Char(4) primary key not null,
  DescrCat VarChar(500) not null
);

create table Prodotti
( CodP Char(4) primary key not null,
  CodCat Char(4) not null,
  DescrP VarChar(500) not null,
  CU Float not null,
  foreign key(CodCat) references Categorie(CodCat)
);

create table IntProd
( CodP Char(4) not null,
  IDI int(20) not null,
  QTA float not null,
  primary key(IDI,CodP),
  foreign key(CodP) references Prodotti(CodP),
  foreign key(IDI) references Interventi(IDI)
);
```


(e) Realizzazione in linguaggio SQL delle seguenti interrogazioni

1. numero delle auto che sono attualmente in officina per operazioni di assistenza

La richiesta è piuttosto semplice. Si tratta di contare le auto presenti attualmente in officina. Poiché in OPERAZIONI abbiamo sia la data di consegna presunta che quella effettiva e la data effettiva è nulla fino a quando l'auto non viene effettivamente consegnata, conviene utilizzare proprio DataCE (data di consegna effettiva). Di fatto basta verificare se la data di consegna effettiva è nulla.

```
SELECT count(operazioni.targa) as 'Numero auto presenti in officina'  
FROM operazioni  
WHERE DataCE IS NULL;
```

2. visualizzare le informazioni delle operazioni di assistenza con il maggior numero di interventi effettuati al suo interno

Questa query richiede tecnicamente l'utilizzo della VISTA in cui andare a predisporre, per ogni operazione di assistenza, il numero di interventi in essa inclusi. In seguito, sulla vista così creata, si andranno ad estrarre le operazioni con il numero di interventi più alto.

Passo 1) Creazione della vista che chiameremo **totali**

```
CREATE VIEW totali  
AS  
SELECT operazioni.IDO, operazioni.Targa, count(interventi.IDO) as numint  
FROM operazioni, interventi  
WHERE operazioni.IDO=interventi.IDO  
GROUP BY operazioni.IDO, operazioni.Targa;
```

Passo 2) Estrazione dalla vista **totali** contenente le operazioni con il valore di numint uguale al valore massimo

```
SELECT *  
FROM totali  
WHERE numint = (SELECT MAX(numint) FROM totali);
```

In alternativa si potrebbe ovviare alla creazione della vista con la seguente query

```
SELECT operazioni.IDO, operazioni.Targa, count(interventi.IDO) as numint  
FROM operazioni, interventi  
WHERE operazioni.IDO=interventi.IDO  
GROUP BY operazioni.IDO, operazioni.Targa  
ORDER BY count(interventi.IDO) DESC;
```

In tal modo abbiamo una sorta di "classifica" per cui le operazioni con il più alto numero di interventi saranno in testa all'elenco

Differenza tra i due metodi

Se consideriamo per ogni operazione di assistenza, il numero di interventi come nella seguente tabella:

IDO	Targa	numint
1	AA222AA	5
2	CC111CC	4
3	BB111AA	5
4	AA222AA	3
5	BB111BB	5
6	CC222CC	4
7	AA111AA	4
.....		

Con il metodo 1 avremmo come risultato

IDO	Targa	numint
1	AA222AA	5
3	BB111AA	5
5	BB111BB	5

Con il metodo 2 avremmo come risultato

IDO	Targa	numint
1	AA222AA	5
3	BB111AA	5
5	BB111BB	5
2	CC111CC	4
6	CC222CC	4
7	AA111AA	4
4	AA222AA	3
.....		

E' chiaro che i primi tre records si riferiscono alle operazioni con il più alto numero di interventi

3. visualizzare i dati dei clienti che non si sono ancora registrati

```
SELECT *
FROM clienti
WHERE (User IS NULL);
```

Nota: risulta sufficiente la condizione (User IS NULL) in quanto non può esistere una registrazione senza password

4. le categorie di prodotti con prezzo medio superiore ad un valore dato

In pratica di ogni categoria si calcola il prezzo medio dei prodotti ad essa appartenenti visualizzando solo quelle per cui tale prezzo medio risulta superiore ad un valore dato

```
SELECT categorie.DescrCat, avg(prodotti.Cu)
FROM categorie, prodotti
WHERE categorie.CodCat=prodotti.CodCat
GROUP BY categorie.DescrCat
HAVING avg(prodotti.CU) > xxx;           dove xxx è un valore dato a scelta (ad es. 200)
```

Per corrispondere pienamente alla richiesta la query dovrebbe essere la seguente:

```
SELECT categorie.DescrCat
FROM categorie, prodotti
WHERE categorie.CodCat=prodotti.CodCat
GROUP BY categorie.DescrCat
HAVING avg(prodotti.CU) > xxx;
```

senza cioè visualizzare il valore del prezzo medio

(f) la progettazione dell'interfaccia della pagina web che permetta ad un cliente registrato di svolgere le operazioni previste

Il sito presenterà un layout comune a tutte le pagine che potrebbe essere strutturato nel modo seguente:

Layout della Home Page comune a tutte le pagine del sito

Header →	logo					
Nav →	HOME	CHI SIAMO	SERVIZI	REVISIONI	REGISTRATI	LOGIN
Aside	News	SECTION				
	Offerte speciali					
	Ultimi aggiornamenti					
	Contatti Dove siamo					
	Contattaci o richiedi un preventivo gratuito					
Footer →	Footer					

Struttura delle cartelle del sito

Si suppone che all'interno della cartella SITO_OFFICINA siano memorizzate le pagine html e php mentre i fogli di stile e le immagini siano memorizzati rispettivamente nelle cartelle CSS e IMMAGINI all'interno di SITO_OFFICINA.

Pagina web che permette ad un utente registrato di effettuare le operazioni previste:

logo				
HOME	CHI SIAMO	SERVIZI	REVISIONI	LOGOUT
News	Benvenuto <i>Rossi Paolo</i>			
Offerte speciali	Controlla lo stato della Operazione di assistenza sulla tua auto Inserisci il numero di targa <input type="text"/> <input type="button" value="VAI"/>			
Ultimi aggiornamenti				
Contatti	Visualizza tutti gli interventi Scegli l'intervallo di date <input type="text"/> ▼ <input type="text"/> ▼			
Dove siamo				
Contattaci o richiedi un preventivo gratuito				
Footer				

- (g) la codifica in un linguaggio a scelta di un segmento significativo dell'applicazione Web che consente l'interazione con la base di dati.

Viene preso in considerazione il segmento dell'applicazione web che permette di effettuare il login da parte di un utente registrato

In particolare si suppone che dalla home page del sito la voce LOGIN sia un link alla pagina web form_login.html

Progettazione pagina web login.html

logo											
HOME	CHI SIAMO	SERVIZI	REVISIONI	REGISTRATI	LOGIN						
News	<div style="text-align: center;"> <h2 style="color: red;">Accedi alla tua area</h2> <table border="1" style="margin: auto;"> <tr> <td style="padding: 5px;">User (*)</td> <td><input type="text"/></td> </tr> <tr> <td style="padding: 5px;">Password (*)</td> <td><input type="password"/></td> </tr> <tr> <td colspan="2" style="text-align: center;"> <input type="button" value="Accedi"/> <input type="button" value="Azzera"/> </td> </tr> </table> <p>I campi contrassegnati con (*) sono obbligatori</p> </div>					User (*)	<input type="text"/>	Password (*)	<input type="password"/>	<input type="button" value="Accedi"/> <input type="button" value="Azzera"/>	
User (*)						<input type="text"/>					
Password (*)						<input type="password"/>					
<input type="button" value="Accedi"/> <input type="button" value="Azzera"/>											
Offerte speciali											
Ultimi aggiornamenti											
Contatti											
Dove siamo											
Contattaci o richiedi un preventivo gratuito											
Footer											

A seguito dell'inserimento delle credenziali, ci sarà una pagina di risposta che visualizzerà un messaggio "Nome utente e/o password errati" nel caso non esista nel database un utente con user e password inseriti oppure verrà visualizzata una pagina di benvenuto con visualizzazione di nome e cognome dell'utente per poi proseguire nella navigazione all'interno della propria area.

Cliccando sul bottone Accedi viene mandato in esecuzione il file esegui_login.php

Pagina *form_login.html*

Si descrive il codice relativo al contenuto, all'interno della Section

```
<body>
.....
<section>
  <h1> Accedi alla tua area</h1>
  <form name="frm_login" action="esegui_login.php" method="POST">
    <table id="tabella_utente">
      <tr>
        <td>User (*)</td><td><input type="text" name="User" maxlength="20" size="20"
          required></td>
      </tr>
      <tr>
        <td>Password(*)</td><td><input type="password" name="Password" maxlength="32"
          size="20" required></td>
      </tr>
      <tr>
        <td colspan="2"><input type="submit" value="Accedi"><input type="reset"
          value="Azzera"></td>
      </tr>
    </table>
  </form>
</section>
```

```
</tr>
</table>
<br/>I campi contrassegnati con (*) sono obbligatori
</form>
</section>
.....
</body>
.....
```

Pagina *esegui_login.php*

Codice (solo della section) della pagina di risposta

```
<body>
.....
<section>
  <?php
    if (isset($_POST["User"] ))
    {
      /* connessione al database*/
      $conn=mysqli_connect("localhost","root","","5ai_officina");
      if (!$conn)
        echo "connessione al database non riuscita";
      else
      {
        $User=$_POST["User"];
        $Password=md5($_POST["Password"]);
        $testo="select *
              from clienti
              where User='$User' AND Password='$Password'";
        $risultato=mysqli_query($conn,$testo)or die ("errore");
        $nrec=mysqli_num_rows($risultato);
        if (mysqli_num_rows($risultato)!=0)
        {
          $riga=mysqli_fetch_array($risultato);
          echo "<h2>Benvenuto ". $riga['Nome']." ".$riga['Cogn'].</h2>";
          // si passa alla interfaccia con cui l'utente può effettuare consultazioni
        }
        else
          echo "<h1>Nome utente e/o password errato</h1>";
        }
        mysqli_close($conn);
      }
    ?>
  </section>
  .....
</body>
```

SOLUZIONE SECONDA PARTE

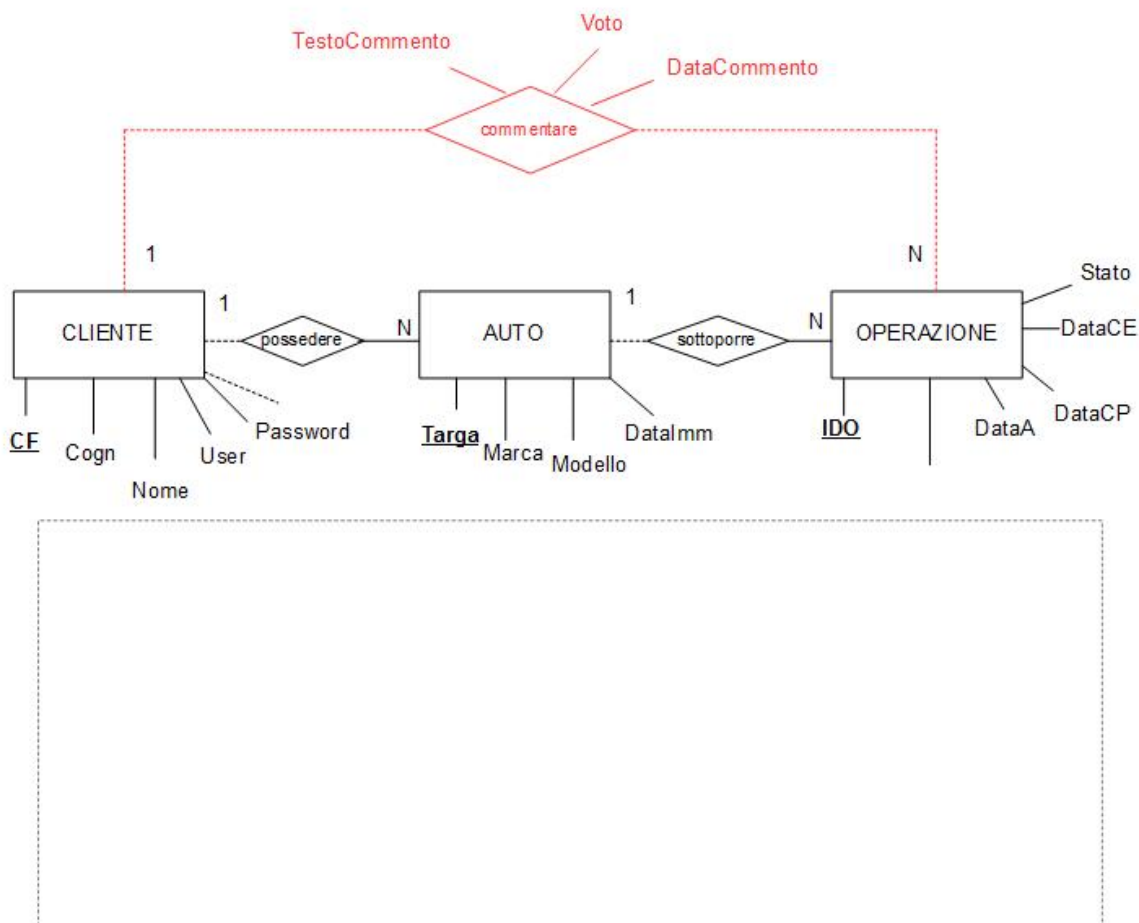
1. In relazione al tema proposto nella prima parte, descriva in che modo è possibile integrare lo schema concettuale sopra sviluppato in modo da poter gestire la possibilità da parte di un cliente registrato di fornire un feed-back che esprima il suo livello di soddisfazione al servizio ricevuto esprimendo un commento ed un voto da 1 a 5

Il quesito richiede la possibilità di esprimere un commento sul servizio ricevuto. Il testo non chiarisce se un cliente possa esprimere un solo giudizio e commento relativo ad una operazione di assistenza specifica, se i commenti saranno pubblicati sul sito né tantomeno se i clienti possano esprimere commenti di risposta ai commenti effettuati da altri.

E' ragionevole supporre che il cliente possa esprimere un solo giudizio (commento e voto) relativamente ad una specifica operazione di assistenza (ovviamente tutte le operazioni di assistenza che si riferiscano alla/e sua/e auto),

Le strade che si possono seguire sono due:

- 1) Inserire una associazione **commentare** tra l'entità CLIENTE e l'entità OPERAZIONE di tipo 1-N con partecipazione facoltativa in entrambi i lati. Il nuovo schema è pertanto (si traslascia la parte che resta invariata)



dove TestoCommento, DataCommento e Voto sono attributi dell'associazione.

Nello schema del Modello Relazionale avremo pertanto la tabella OPERAZIONI che diventa:

Clienti(**CF**,Cogn,Nome,User>Password,.....)

Auto(**Targa**,Modello,Marca,DataImm,**CF**)

Operazioni(**IDO**, DataA,DataCP,DataCE,Stato,TestoCommento,Voto,DataCommento,**CF**,**Targa**)

.....

Le altre tabelle rimangono invariate.

In Operazioni troviamo in aggiunta CF che è la Foreign Key (deriva dall'associazione 1-n tra Clienti e Operazioni) e gli attributi dell'associazione (TestoCommento,Voto, DataCommento) che vanno inseriti nella tabella del lato N.

Nella tabella descrittiva quindi avremo

Tabella	Attributo	Descrizione	Chiave	Formato	Dimensione	Obbligatorio
Clienti	-----	-----	-----	-----	-----	----
Auto	-----	-----	-----	-----	-----	----
Operazioni	IDO	Codice dell'operazione	PK	Intero progressivo		SI
	DataA	Data accettazione		Data		SI
	DataCP	Data di consegna prevista		Data		SI
	DataCE	Data di consegna effettiva Valore di Default NULL		Data		NO
	Stato	Stato di lavorazione della vettura		Stringa	20	SI
	TestoCommento	Testo dell'eventuale commento Valore di Default NULL		Stringa	250	NO
	Voto	Valutazione (valore da 1 a 5)		Intero		NO
	DataCommento	Data in cui è stato espresso il commento/voto		Data		NO
	CF	Codice del cliente che ha effettuato il commento	FK	Stringa	16	NO
	Targa	Targa dell'auto	FK	Stringa	10	SI
Interventi	-----	-----	-----	-----	-----	----

TipiIntervento	-----	-----	-----	-----	-----	-----
IntProd	-----	-----	-----	-----	-----	-----
Prodotti	-----	-----	-----	-----	-----	-----
Categorie	-----	-----	-----	-----	-----	-----

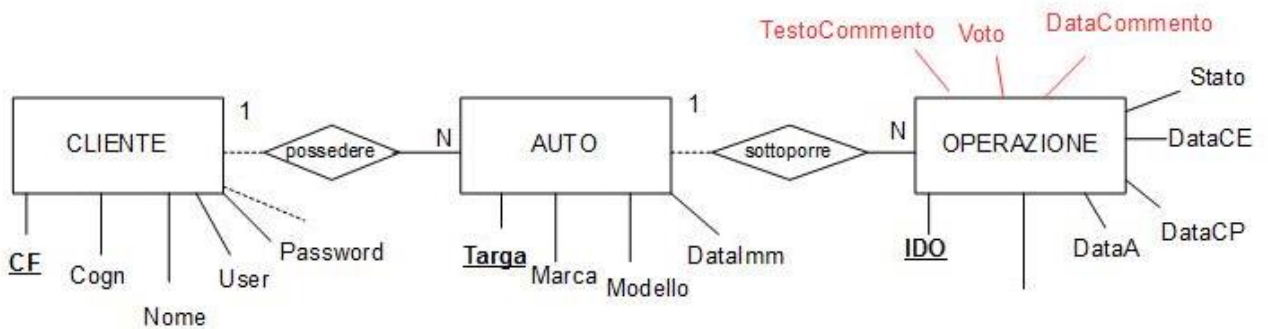
Per modificare la tabella Operazioni all'interno del database dobbiamo utilizzare il comando ALTER TABLE.

ALTER TABLE operazioni

ADD TestoCommento varchar(256) Default NULL,
ADD Voto INT Default NULL,
ADD DataCommento date Default NULL,
ADD CF char(16) Default NULL,
ADD FOREIGN KEY(CF) references clienti(CF);

Si possono ovviamente utilizzare tante ALTER TABLE per quanti ADD dobbiamo applicare!

2) Modificare lo schema E/R inserendo all'interno dell'entità OPERAZIONE gli attributi TestoCommento, Voto, DataCommento.



E' sicuramente la soluzione più semplice da realizzare e che impatta in minima parte sul Modello Relazionale senza perdere in correttezza in quanto si riesce comunque a risalire all'autore del commento grazie alla targa dell'auto interessata dall'operazione.

Nel Modello Relazionale avremo

Clienti(**CF**,Cogn,Nome,User>Password,.....)

Auto(**Targa**,Modello,Marca,DataImm,**CF**)

Operazioni(**IDO**, DataA,DataCP,DataCE,Stato,**TestoCommento**,**Voto**,Data**Commento**,**Targa**)

Con il significato ed i tipi dei dati indicati nella soluzione precedente.

In questo caso la modifica da apportare alla tabella operazioni si limita all'aggiunta dei tre nuovi campi

ALTER TABLE operazioni

ADD TestoCommento varchar(256) Default NULL,

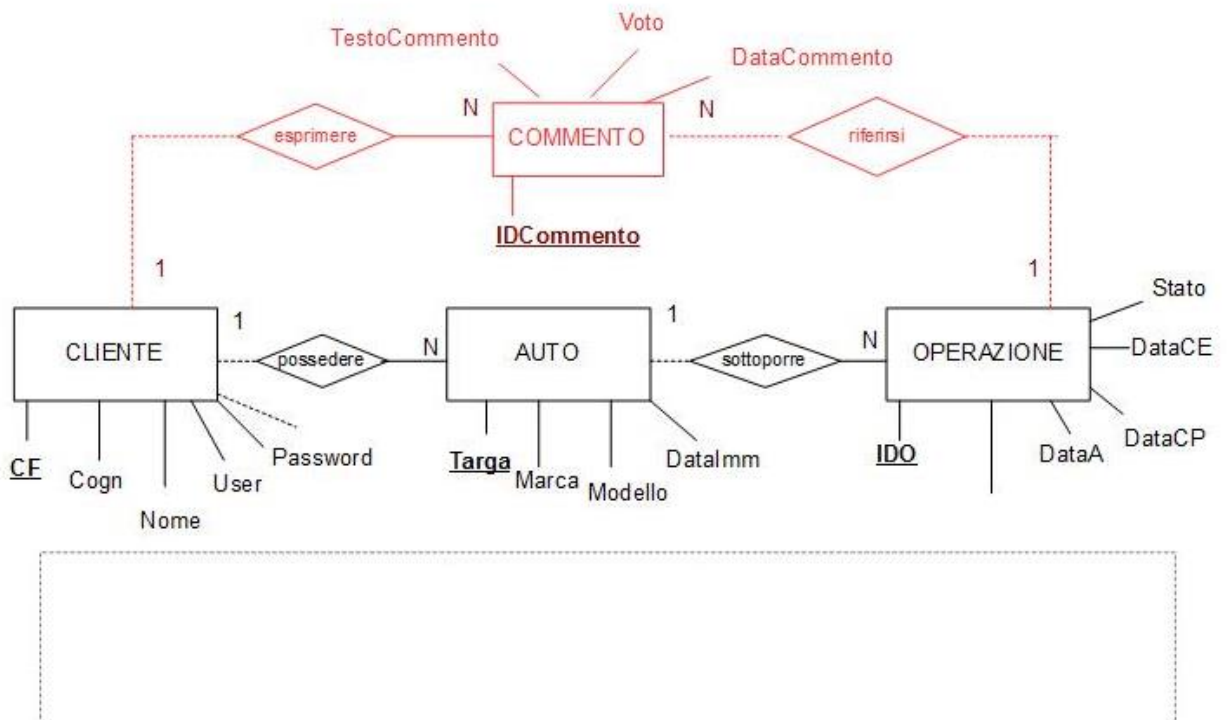
ADD Voto INT Default NULL,

ADD DataCommento date Default NULL;

3) Nota:

Una terza proposta potrebbe essere quella relativa alla possibilità di esprimere commenti da parte di chiunque anche su operazioni effettuate da altri. Non ritengo probabile questa soluzione anche se potrebbe essere accettabile con la limitazione però che solo il cliente direttamente coinvolto nell'operazione di assistenza possa esprimere il proprio voto, mentre gli altri clienti possono eventualmente commentare

In tal caso lo schema E/R risulta



Nel passaggio al Modello Relazionale si avrà la tabella COMMENTI in aggiunta a tutte le altre tabelle

Commenti(IDCommento, TestoCommento, Voto, DataCommento, CF, IDO)

2. In relazione al tema proposto nella prima parte esprimere in sql la query che permetta di calcolare il costo totale da fatturare per una operazione di assistenza

Il costo totale di una operazione di assistenza è dato dal costo dei prodotti utilizzati con le relative quantità e dal costo della manodopera.

Come già evidenziato il costo orario della manodopera si suppone noto e costante ad es. pari a € 30/h.

```
SELECT SUM(intprod.Qta*prodotti.CU)+SUM(TipiIntervento.Nore*30)
FROM prodotti, intprod, interventi, TipiIntervento
WHERE (prodotti.Codp=intprod.Codp AND intprod.IDI=interventi.IDI AND
       Interventi.CoDTI=TipiIntervento.CodTI)
       AND
       (Interventi.IDO=145);
```

Osservazione: nel calcolo della fattura si deve evidenziare l'imponibile e l'IVA. La query indicata restituisce il totale imponibile senza IVA.

L'applicativo realizzato dovrà prevedere una apposita sezione di emissione della fattura che, grazie all'utilizzo del PHP, possa evidenziare le singole voci di imponibile, il totale imponibile, l'iva e quindi l'importo totale della fattura.

Es.:

Restituzione dei prodotti utilizzati in una operazione di assistenza:

```
Select prodotti.CodP,prodotti.DescrP,prodotti.Cu,intprod.Qta,prodotti.Cu*intprod.qta
From prodotti,intprod,interventi
Where (prodotti.Codp=intprod.Codp AND intprod.IDI=interventi.IDI)
       AND
       (Interventi.IDO=145);
```

Restituzione dei tipi di intervento effettuati e relative ore utilizzate in una operazione di assistenza:

```
select TipiIntervento.CodTi,TipiIntervento.DescrTI,TipiIntervento.Nore,
                                             TipiIntervento.Nore*30
From TipiIntervento,Interventi
Where (TipiIntervento.CodTI=Interventi.CodTI)
       AND
       (Interventi.IDO=145);
```

3. Illustri le regole di derivazione che permettono di ottenere uno schema relazionale a partire da uno schema concettuale.

Innanzitutto ogni entità diventa, nel Modello Relazionale, una tabella che avrà come campi (e quindi come colonne) gli attributi dell'entità stessa ed in particolare la chiave primaria dell'entità sarà la chiave primaria della tabella e ciascuna istanza dell'entità sarà una riga (un record) della tabella. In particolare la tabella rispetterà la prima forma normale.

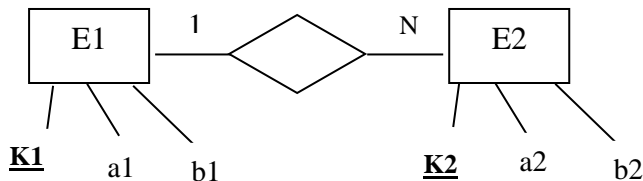
In presenza di associazioni, il passaggio al modello relazionale prevede delle regole a seconda del tipo di associazione presente. Prendiamo in considerazione le associazioni binarie. Poiché i tipi di associazione sono 1-1,1-N,N-N avremo tre regole di derivazione.

Associazione 1-N (semplice)

L'entità del verso 1 dà origine ad una tabella che avrà gli stessi attributi dell'entità da cui deriva.

L'entità del verso N avrà, oltre a tutti gli attributi dell'entità da cui deriva, un attributo aggiuntivo che è la chiave primaria dell'entità del verso 1. Questo campo viene detto Foreign Key (Chiave Esterna) ed esprime in pratica il collegamento tra le due entità.

Es.:



Il modello relazionale sarà

$E1(\underline{K1}, a1, b1)$

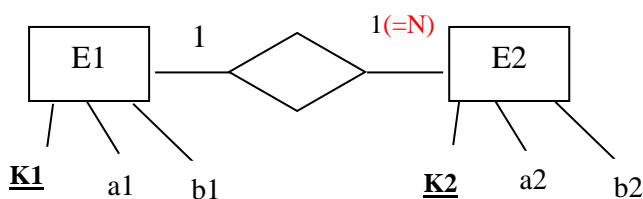
$E2(\underline{K2}, a2, b2, K1)$ K1 Foreign Key

Associazione 1-1 (biunivoca)

Per quanto riguarda l'associazione 1-1 abbiamo due strade.

- 1) Si considera l'associazione 1-1 come un caso particolare di associazione 1-N. Si sceglie quindi quale lato fungerà da "lato N" e si applica pertanto la regola sopra esposta. Per quanto riguarda la scelta del verso che fungerà da lato N in generale si predilige il verso che partecipa all'associazione in maniera obbligatoria (se entrambi i lati lo fossero si sceglie quello ritenuto più opportuno).

Es.:



Supponiamo di scegliere come lato N quello relativo ad E2

Il modello relazionale sarà

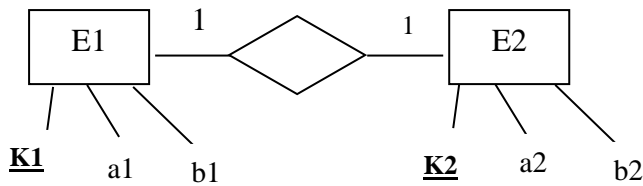
$E1(\underline{K1}, a1, b1)$

$E2(\underline{K2}, a2, b2, K1)$ K1 Foreign Key

La foreign key K1 risulta UNIVOCA (e quindi chiave alternativa)

- 2) Le due entità danno origine ad una unica tabella che conterrà tutti gli attributi delle due tabelle. Si sceglierà come chiave primaria una delle due chiavi primarie delle entità e l'altra sarà da considerare Univoca e quindi una chiave alternativa

Es.:



Il modello relazionale sarà

$E1_E2(\underline{K1}, a1, b1, K2, a2, b2)$

In questo caso ho scelto K1 come Primary Key e quindi K2 dovrà essere dichiarato univoco

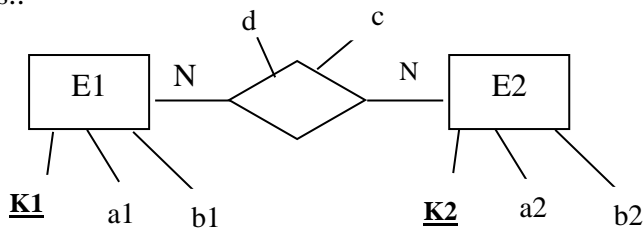
Il nome della tabella E1_E2 è del tutto casuale. Si può prendere, ad esempio, il nome della tabella da cui si deriva la chiave primaria.

Associazione N-N (complessa)

Le due entità danno origine a due tabelle con gli stessi attributi delle entità da cui derivano. Si ha, inoltre, una terza tabella, detta tabella di link, che sarà costituita dalle chiavi primarie delle due entità di partenza più gli eventuali attributo dell'associazione.

Le due chiavi primarie formeranno una chiave primaria composta ma singolarmente sono delle chiavi esterne di collegamento con le due tabelle.

Es.:

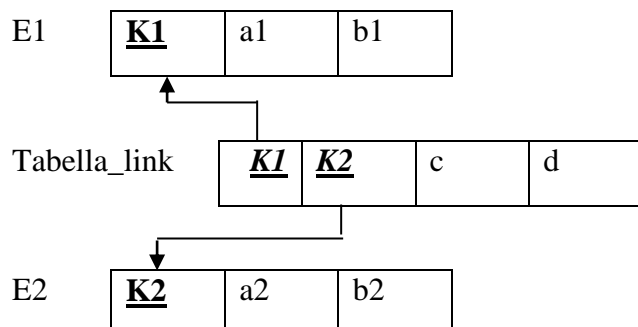


Il modello relazionale sarà

$E1(\underline{K1}, a1, b1)$

$E2(\underline{K2}, a2, b2)$

Tabella_Link($K1, K2$, c, d)



4. Illustrare, anche servendosi di esempi, le tipologie di operazioni che si possono effettuare all'interno di un DBMS relazionale

Diverse sono le operazioni principali che possono essere effettuate sulle relazioni (cioè all'interno di un dbms relazionale). Illustreremo le tre principali:

- Selezione
- Proiezione
- Congiunzione

In SQL il comando che realizza questi operatori è il comando SELECT

Operatore di selezione:

L'operatore di selezione si dice unario in quanto agisce su una sola tabella. Esso agisce sulle righe, ovvero permette, a partire da una tabella, di estrapolare solo i records che soddisfano delle condizioni. La tabella che si ottiene avrà pertanto lo stesso grado della tabella di partenza e cardinalità minore o uguale della tabella di partenza.

In SQL:

```
SELECT tabella.*  
FROM tabella  
WHERE condizione/i
```

ES.: Consideriamo la tabella

Dipendenti

Matr	Cognome	Nome	Dipartimento	Stipendio
100	Rossi	Paolo	A	1850
110	Bianchi	Guido	B	1550
120	Bianchi	Lucia	A	1800
130	Verdi	Paolo	B	1495
140	Ambrosini	Maria	C	1050
150	Bruni	Marco	A	1855

Grado=5 Cardinalità=6

La richiesta "Elenco dei dipendenti del dipartimento A" è un esempio di selezione. Il risultato che si ottiene a partire dalla tabella data risulta

Matr	Cognome	Nome	Dipartimento	Stipendio
100	Rossi	Paolo	A	1850
120	Bianchi	Lucia	A	1800
150	Bruni	Marco	A	1855

Grado=5 Cardinalità =3

Un esempio di selezione con restituzione di una tabella con la stessa cardinalità di quella di partenza potrebbe essere: "Elenco dei dipendenti con stipendio superiore a 1000 €".

In tal caso, infatti, tutti i records di Dipendenti soddisfano alla condizione indicata

Operatore di proiezione:

L'operatore di proiezione si dice unario in quanto agisce su una sola tabella. Esso agisce sulle colonne, ovvero permette, a partire da una tabella, di estrapolare solo alcune colonne. La tabella che si ottiene avrà pertanto grado minore o uguale a quello della tabella di partenza e cardinalità uguale a quello della tabella di partenza (a meno di eliminazione delle righe duplicate)

In SQL:

```
SELECT tabella.attr1,tabella.att2  
FROM tabella
```

Es.:

sempre considerando la tabella Dipendenti la richiesta "Visualizzare nome, cognome e stipendio dei dipendenti" è un esempio di proiezione il cui risultato sarà

Cognome	Nome	Stipendio
Rossi	Paolo	1850
Bianchi	Guido	1550
Bianchi	Lucia	1800
Verdi	Paolo	1495
Ambrosini	Maria	1050
Bruni	Marco	1855

Grado=3 Cardinalità=6

Un esempio di proiezione in cui la cardinalità sia minore a seguito della eliminazione delle righe duplicate potrebbe essere "Elenco dei dipartimenti"

Dipartimento
A
B
C

In SQL:

```
SELECT distinct dipartimento  
FROM dipendenti;
```

Operatore di congiunzione:

L'operatore di congiunzione, a differenza dei precedenti, è un operatore binario in quanto agisce su due tabelle che hanno un campo in comune (generalmente la foreign key).

La tabella che si ottiene avrà grado dato dalla somma dei gradi delle due tabelle meno il numero dei campi in comune (in genere 1). Sulla cardinalità non si può dire nulla a priori ma generalmente se il campo in comune è la foreign key la cardinalità è quella della tabella del lato N.

In SQL:

```
SELECT tabella1.* , tabella2.campo1,tabella2.caampo2, ....
FROM tabella1,tabella2
WHERE tabella1.PK=tabella2.FK
```

Es.:

Dipendenti

Matr	Cognome	Nome	Dipartimento	Stipendio
100	Rossi	Paolo	A	1850
110	Bianchi	Guido	B	1550
120	Bianchi	Lucia	A	1800
130	Verdi	Paolo	B	1495
140	Ambrosini	Maria	C	1050
150	Bruni	Marco	A	1855

Dipartimenti

CodD	NomeD
A	Personale
B	Amministrazione
C	Acquisti
D	R&S

L'attributo in comune è Dipendenti.Dipartimento, Foreign Key collegata a Dipartimenti.CodD

La tabella risultante dalla congiunzione sarà:

Matr	Cognome	Nome	Dipartimento	Stipendio	NomeD
100	Rossi	Paolo	A	1850	Personale
110	Bianchi	Guido	B	1550	Amministrazione
120	Bianchi	Lucia	A	1800	Personale
130	Verdi	Paolo	B	1495	Amministrazione
140	Ambrosini	Maria	C	1050	Acquisti
150	Bruni	Marco	A	1855	Personale

Il grado della nuova tabella risulta essere pari a 6 dato dalla somma del grado della tabella Dipendenti (5) e quello della tabella Dipartimenti (2) – 1 (il campo in comune in questo caso CodD)