

Descrizione

Una catena di farmacie decide di automatizzare la gestione dei propri clienti e vendite ed è necessario il progetto del database di riferimento. La prima discussione con il gruppo responsabile presso l'azienda consente di derivare le seguenti informazioni parziali:

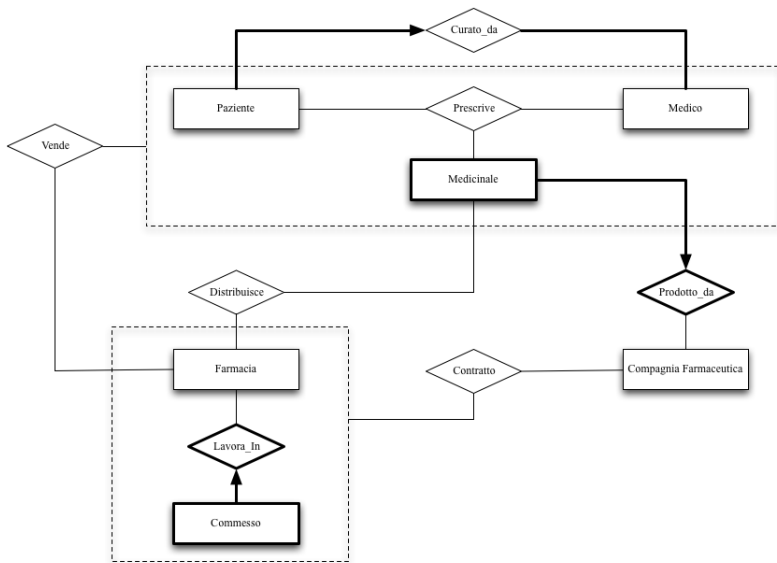
Specifiche:

Nella catena di farmacie sono incluse più di cento farmacie, sparse per il territorio. Esse distribuiscono prodotti di numerose compagnie farmaceutiche con cui hanno contratti a lungo termine, supervisionati da un impiegato per ogni farmacia.

Ogni farmacia dovrebbe documentare/memorizzare i dati delle prescrizioni.

Ogni paziente può, presentando la prescrizione di un medico, accedere ad un medicinale. La vendita di tale medicinale, a fronte della prescrizione medica, va registrata col suo ammontare. I medici che fanno prescrizioni vanno registrati per poter essere contattati al momento o dopo la vendita.

ER



Schema Logico

Entità

- Paziente(CF:string , Nome:String , Età:int , Indirizzo:string, CFMed:string)
- Medico(CF:string , Nome:String , AnzianitàServizio:int , Spec:string)
- Farmacia(idF:int , Nome:string , Telefono:String , Indirizzo:string)
- CompagniaFarmaceutica(idCF:int , Nome:string , Telefono:String)
- Commesso(CF:string , Nome:string, idF:int)
- Medicinale(Nome:string , idCF:int , Formula:string)

Schema Logico

Entità

- Paziente(CF:string , Nome:String , Et :int , Indirizzo:string, CFMed:string)
- Medico(CF:string , Nome:String , Anzianit Servizio:int , Spec:string)
- Farmacia(idF:int , Nome:string , Telefono:String , Indirizzo:string)
- CompagniaFarmaceutica(idCF:int , Nome:string , Telefono:String)
- Commesso(CF:string , Nome:string, idF:int)
- Medicinale(Nome:string , idCF:int , Formula:string)

Relazioni

- Prescrive(idPr:int ,
CFPaz:string , CFMed:string, NomeMedicin:String , idCF:int , Data:data ,
Quantit :int)
- Distribuisce(idF:int , NomeMedicin:String , idCF:int , Prezzo:int)
- Contratto(idF:int , idCF:int , CFImp:string , Inizio:data , Fine:data ,
Testo:string)
- Vende(idF:int , idPr:int)

Creazione Entità Farmacia

```
CREATE TABLE `Farmacia` (  
  `idF` int(11) NOT NULL auto_increment,  
  `Nome` varchar(255) NOT NULL,  
  `Telefono` varchar(20) NOT NULL,  
  `Indirizzo` varchar(255) NOT NULL,  
  PRIMARY KEY (`idF`)  
) ENGINE=InnoDB
```

Creazione Entità Medico

```
CREATE TABLE `Medico` (  
  `CF` varchar(18) NOT NULL,  
  `Nome` varchar(255) NOT NULL,  
  `AnzServizio` int(11) NOT NULL,  
  `Specializzazione` varchar(255) NOT NULL,  
  PRIMARY KEY (`CF`)  
) ENGINE=InnoDB
```

Creazione Entità Paziente

```
CREATE TABLE `Paziente` (  
  `CF` varchar(18) NOT NULL,  
  `Nome` varchar(255) NOT NULL,  
  `Eta` int(11) NOT NULL,  
  `Indirizzo` varchar(255) NOT NULL,  
  `CFMed` varchar(18) NOT NULL,  
  PRIMARY KEY (`CF`),  
  FOREIGN KEY (`CFMed`)  
    REFERENCES `Medico` (`CF`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
) ENGINE=InnoDB
```


Creazione Relazione Prescrive

```
CREATE TABLE `Prescrive` (  
  `idPR` int(11) NOT NULL auto_increment,  
  `CFPaz` varchar(18) NOT NULL,  
  `CFMed` varchar(18) NOT NULL,  
  `NomeMedicinale` varchar(255) NOT NULL,  
  `idCF` int(11) NOT NULL,  
  `data` date NOT NULL,  
  `Quantita` int(11) NOT NULL,  
  PRIMARY KEY (`idPR`),  
  UNIQUE (`CFPaz`, `CFMed`, `NomeMedicinale`, `idCF`),  
  FOREIGN KEY (`CFPaz`) REFERENCES `Paziente` (`CF`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
  FOREIGN KEY (`CFMed`) REFERENCES `Medico` (`CF`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE,  
  FOREIGN KEY (`NomeMedicinale`, `idCF`)  
    REFERENCES `Medicinale` (`Nome`, `idCF` )  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
) ENGINE=InnoDB
```

Queries

- Quale è il numero di telefono della Farmacia *X*?
- Quali medicinali sono stati prescritti al paziente *X*?
- Quale farmacia vende il medicinale *X* della Casa Farmaceutica *Y*?
- Quali medici hanno prescritto il medicinale *X* della Casa Farmaceutica *Y*?
- Quali medicinali sono venduti dalla farmacia *X* ma non dalla *Y*?
- Quale è il prezzo di vendita minimo del Medicinale *X* della Casa Farmaceutica *Y*?
- Quali Farmacie vendono al prezzo più basso il Medicinale *X* della Casa Farmaceutica *Y*?

Dall'algebra relazionale a SQL

Quale è il numero di telefono della Farmacia X?

Dall'algebra relazionale a SQL

Quale è il numero di telefono della Farmacia X?

$$\pi_{\text{telefono}}(\sigma_X(\text{Farmacia}))$$

Dall'algebra relazionale a SQL

Quale è il numero di telefono della Farmacia X?

$$\pi_{\text{telefono}}(\sigma_X(\text{Farmacia}))$$

```
SELECT telefono  
  FROM Farmacia  
 WHERE nome = 'X'
```

Dall'algebra relazionale a SQL

Quale è il numero di telefono della Farmacia X?

$$\pi_{\text{telefono}}(\sigma_X(\text{Farmacia}))$$

Uso dell'operatore LIKE:

```
SELECT telefono  
  FROM Farmacia  
 WHERE nome LIKE 'X'
```

Dall'algebra relazionale a SQL

Quale è il numero di telefono della Farmacia X?

$$\pi_{\text{telefono}}(\sigma_X(\text{Farmacia}))$$

LIKE usato sulle stringhe permette match parziali attraverso l'uso di wildcards % e _ :

% : Corrisponde ad un qualsiasi carattere senza limiti di ripetizione (anche 0 volte)

_ : Corrisponde ad un qualsiasi carattere ma che occorre esattamente una volta

```
SELECT telefono  
FROM Farmacia  
WHERE nome LIKE '%X_'
```

Dall'algebra relazionale a SQL

Quali medicinali sono stati prescritti al paziente X?

$$\pi_{Medicinale.Nome, CF.Nome}(\sigma_X(Paziente) \bowtie Prescrive \bowtie (Medicinale \bowtie CF))$$

Versione senza l'uso del costrutto JOIN:

```
SELECT `Medicinale`.`Nome`, `CompagniaFarmaceutica`.`Nome`  
FROM `Medicinale`, `CompagniaFarmaceutica`,  
     `Paziente`, `Prescrive`  
WHERE `Paziente`.`CF` = 'X' AND  
      `Medicinale`.`idCF` = `CompagniaFarmaceutica`.`idCF` AND  
      `Paziente`.`CF` = `Prescrive`.`CFPaz` AND  
      `Prescrive`.`idCF` = `CompagniaFarmaceutica`.`idCF` AND  
      `Prescrive`.`NomeMedicinale` = `Medicinale`.`Nome`
```


Dall'algebra relazionale a SQL

Quali medicinali sono stati prescritti al paziente X?

$$\pi_{Medicinale.Nome, CF.Nome}(\sigma_X(Paziente) \bowtie Prescrive \bowtie (Medicinale \bowtie CF))$$

Versione senza l'uso del costrutto JOIN

Riscrittura dei Nomi:

```
SELECT 'M'.'.Nome' AS 'Nome_Farmaco',  
       'CF'.'.Nome' AS 'Nome_Compagnia'  
FROM 'Medicinale' AS 'M', 'CompagniaFarmaceutica' AS 'CF',  
     'Paziente' AS 'P', 'Prescrive' AS 'Pr'  
WHERE 'P'.'.CF' = 'X' AND  
      'M'.'.idCF' = 'CF'.'.idCF' AND  
      'P'.'.CF' = 'Pr'.'.CFPaz' AND  
      'Pr'.'.idCF' = 'CF'.'.idCF' AND  
      'Pr'.'.NomeMedicinale' = 'M'.'.Nome' AND  
      'M'.'.idCF' = 'CF'.'.idCF'
```

Dall'algebra relazionale a SQL

Quali medicinali sono stati prescritti al paziente X?

$$\pi_{Medicinale.Nome, CF.Nome}(\sigma_X(Paziente) \bowtie Prescrive \bowtie (Medicinale \bowtie CF))$$

Versione con l'uso del costrutto JOIN:

```
SELECT 'M'.'.Nome' AS 'Nome_Farmaco',
       'CF'.'.Nome' AS 'Nome_Compagnia'
FROM 'Paziente' AS 'P'
JOIN 'Prescrive' AS 'Pr' ON 'P'.'.CF' = 'Pr'.'.CFPaz'
JOIN 'CompagniaFarmaceutica' AS 'CF'
    ON 'Pr'.'.idCF' = 'CF'.'.idCF'
JOIN 'Medicinale' AS 'M'
    ON 'Pr'.'.NomeMedicinale' = 'M'.'.Nome'
    AND 'M'.'.idCF' = 'CF'.'.idCF',
WHERE 'P'.'.CF' = 'X'
```

Dall'algebra relazionale a SQL

Quale farmacia vende il medicinale X della Casa Farmaceutica Y?

$$\pi_{\text{Farmacia.Nome}}(\sigma_X(\text{Medicinale}) \bowtie \sigma_Y(\text{CF}) \bowtie (\text{Farmacia} \bowtie \text{Distribuisce}))$$

Dall'algebra relazionale a SQL

Quale farmacia vende il medicinale X della Casa Farmaceutica Y?

$\pi_{Farmacia.Nome}(\sigma_X(Medicinale) \bowtie \sigma_Y(CF) \bowtie (Farmacia \bowtie Distribuisce))$

```
SELECT 'F'.'.Nome' AS 'Nome_Farmacia'
FROM 'Medicinale' AS 'M'
JOIN 'CompagniaFarmaceutica' AS 'CF'
      ON 'M'.'.idCF' = 'CF'.'.idCF'
JOIN 'Distribuisce' AS 'D'
      ON 'M'.'.Nome' = 'D'.'.NomeMedicinale'
      AND 'M'.'.idCF' = 'D'.'.idCF'
JOIN 'Farmacia' AS 'F'
      ON 'F'.'.idF' = 'D'.'.idF'
WHERE 'M'.'.Nome' LIKE 'X'
      AND 'CF'.'.Nome' LIKE 'Y'
```

Dall'algebra relazionale a SQL

Quali medici hanno prescritto il medicinale X della Casa Farmaceutica Y?

$\pi_{Medico.Nome}((\sigma_X(Medicinale) \bowtie \sigma_Y(CF) \bowtie Prescrive \bowtie Medico))$

Dall'algebra relazionale a SQL

Quali medici hanno prescritto il medicinale X della Casa Farmaceutica Y?

$\pi_{Medico.Nome}((\sigma_X(Medicinale) \bowtie \sigma_Y(CF) \bowtie Prescrive \bowtie Medico)$

```
SELECT 'M'. 'Nome' AS 'Nome_Medico'
FROM 'Medicinale' AS 'M'
JOIN 'CompagniaFarmaceutica' AS 'CF'
      ON 'M'. 'idCF' = 'CF'. 'idCF'
JOIN 'Prescrive' AS 'P'
      ON 'M'. 'Nome' = 'P'. 'NomeMedicinale'
      AND 'M'. 'idCF' = 'P'. 'idCF'
JOIN 'Medico' AS 'Med'
      ON 'Med'. 'CF' = 'P'. 'CFMed'
WHERE 'M'. 'Nome' LIKE 'X'
      AND 'CF'. 'Nome' LIKE 'Y'
```

Dall'algebra relazionale a SQL

Quali medicinali sono venduti dalla farmacia X ma non dalla Y?

$\rho(R1, \pi_{Md.Nome, CF.Nome}(\sigma_X(Farmacia) \bowtie Distribuisce \bowtie Medicinale \bowtie CF))$

$\rho(R2, \pi_{Md.Nome, CF.Nome}(\sigma_Y(Farmacia) \bowtie Distribuisce \bowtie Medicinale \bowtie CF))$

$R1 - R2$

Dall'algebra relazionale a SQL

Quali medicinali sono venduti dalla farmacia X ma non dalla Y?

$\rho(R1, \pi_{Md.Nome, CF.Nome}(\sigma_X(Farmacia) \bowtie Distribuisce \bowtie Medicinale \bowtie CF))$

$\rho(R2, \pi_{Md.Nome, CF.Nome}(\sigma_Y(Farmacia) \bowtie Distribuisce \bowtie Medicinale \bowtie CF))$

$R1 - R2$

```
CREATE TEMPORARY TABLE R1 (
  SELECT 'M'.'.Nome' AS 'Nome_Medicinale',
         'CF'.'.Nome' AS 'Nome_Compagnia'
  FROM 'Medicinale' AS 'M'
  JOIN 'CompagniaFarmaceutica' AS 'CF'
    ON 'M'.'.idCF' = 'CF'.'.idCF'
  JOIN 'Distribuisce' AS 'D'
    ON 'M'.'.Nome' = 'D'.'.NomeMedicinale'
    AND 'M'.'.idCF' = 'D'.'.idCF'
  JOIN 'Farmacia' AS 'F'
    ON 'F'.'.idF' = 'D'.'.idF'
  WHERE 'F'.'.Nome' LIKE 'X'
)
```


Dall'algebra relazionale a SQL

Quali medicinali sono venduti dalla farmacia X ma non dalla Y?

$\rho(R1, \pi_{Md.Nome, CF.Nome}(\sigma_X(Farmacia) \bowtie Distribuisce \bowtie Medicinale \bowtie CF))$

$\rho(R2, \pi_{Md.Nome, CF.Nome}(\sigma_Y(Farmacia) \bowtie Distribuisce \bowtie Medicinale \bowtie CF))$

$R1 - R2$

```
CREATE TEMPORARY TABLE R2 (  
  SELECT 'M'.'.Nome' AS 'Nome_Medicinale',  
         'CF'.'.Nome' AS 'Nome_Compagnia'  
FROM 'Medicinale' AS 'M'  
JOIN 'CompagniaFarmaceutica' AS 'CF'  
  ON 'M'.'.idCF' = 'CF'.'.idCF'  
JOIN 'Distribuisce' AS 'D'  
  ON 'M'.'.Nome' = 'D'.'.NomeMedicinale'  
  AND 'M'.'.idCF' = 'D'.'.idCF'  
JOIN 'Farmacia' AS 'F'  
  ON 'F'.'.idF' = 'D'.'.idF'  
WHERE 'F'.'.Nome' LIKE 'Y'  
)
```

Dall'algebra relazionale a SQL

Quali medicinali sono venduti dalla farmacia X ma non dalla Y?

$\rho(R1, \pi_{Md.Nome, CF.Nome}(\sigma_X(Farmacia) \bowtie Distribuisce \bowtie Medicinale \bowtie CF))$

$\rho(R2, \pi_{Md.Nome, CF.Nome}(\sigma_Y(Farmacia) \bowtie Distribuisce \bowtie Medicinale \bowtie CF))$

$R1 - R2$

```
SELECT *  
FROM 'R1'  
WHERE ('Nome_Medicinale', 'Nome_Compagnia') NOT IN (  
    SELECT *  
    FROM 'R2'  
    )
```

Operatori di Aggregazione

Quale è il prezzo di vendita minimo del Medicinale X della Casa Farmaceutica Y

Operatori di Aggregazione

Quale è il prezzo di vendita minimo del Medicinale X della Casa Farmaceutica Y

```
SELECT MIN(`D`.`Prezzo`)
FROM `Distribuisce` AS `D`
JOIN `CompagniaFarmaceutica` AS `CF` ON `CF`.`idCF` = `D`.`idCF`
WHERE `CF`.`Nome` LIKE 'Y' AND `D`.`NomeMedicinale` LIKE 'X'
GROUP BY `D`.`NomeMedicinale`
```

Operatori di Aggregazione

Quali Farmacie vendono al prezzo più basso il Medicinale X della Casa Farmaceutica Y

Operatori di Aggregazione

Quali Farmacie vendono al prezzo più basso il Medicinale X della Casa Farmaceutica Y

```
SELECT 'F'.'.Nome' AS 'Nome_Farmacacia'
FROM 'Distribuisce' AS 'D'
JOIN 'CompagniaFarmaceutica' AS 'CF' ON 'CF'.'.idCF' = 'D'.'.idCF'
JOIN 'Farmacia' AS 'F' ON 'F'.'.idF' = 'D'.'.idF'
WHERE 'CF'.'.Nome' LIKE 'Y' AND 'D'.'.NomeMedicinale' LIKE 'X' AND
'D'.'.Prezzo' = (SELECT MIN('D'.'.Prezzo') AS Prezzo
                  FROM 'Distribuisce' AS 'D'
                  JOIN 'CompagniaFarmaceutica' AS 'CF'
                    ON 'CF'.'.idCF' = 'D'.'.idCF'
                  WHERE 'CF'.'.Nome' LIKE 'X' AND
                        'D'.'.NomeMedicinale' LIKE 'X'
                  GROUP BY 'D'.'.NomeMedicinale' ) )
```

Mysql Download

- <http://dev.mysql.com/downloads/mysql/>
- <http://dev.mysql.com/downloads/gui-tools/5.0.html>