Linguaggio SQL: costrutti avanzati

Gestione delle transazioni

DBMG

Gestione delle transazioni

- Introduzione
- Transazioni in SQL
- Proprietà delle transazioni

Gestione delle transazioni

Introduzione

Esempio applicativo

- Operazioni bancarie
  - operazione di prelievo dal proprio conto corrente mediante bancomat
  - operazione di versamento di denaro contante sul proprio conto corrente

Prelievo

- Operazioni svolte
  - specificare l'importo richiesto
  - verificare la disponibilità
  - memorizzare il movimento
  - aggiornare il saldo
  - abilitare l'erogazione della somma richiesta

Prelievo

- Operazioni svolte
  - specificare l'importo richiesto
  - verificare la disponibilità
  - memorizzare il movimento
  - aggiornare il saldo
  - abilitare l'erogazione della somma richiesta
- Tutte le operazioni devono essere eseguite correttamente, altrimenti il prelievo non va a buon fine

Prelievo

- *Cosa succede se un cointestatario diverso del conto fa un altro prelievo?*
- *Cosa succede in caso di malfunzionamento?*

Versamento

- Operazioni svolte
  - verificare l'importo versato
  - memorizzare il movimento
  - aggiornare il saldo

- Tutte le operazioni devono essere eseguite correttamente, altrimenti il versamento non va a buon fine

Versamento

- *Cosa succede se un'altra persona fa un versamento sullo stesso conto?*
- *Cosa succede in caso di malfunzionamento?*

Esempio: operazioni bancarie

- La base di dati bancaria è un ambiente multiutente
  - diversi operatori possono operare contemporaneamente sulla stessa porzione di dati
- La gestione corretta delle informazioni richiede
  - meccanismi per la gestione *dell'accesso concorrente* alla base di dati
  - meccanismi per il *ripristino* (recovery) dello stato corretto della base di dati in caso di guasti

Gestione delle transazioni

- Necessaria quando più utenti possono accedere contemporaneamente ai dati
- Offre meccanismi efficienti per
  - gestire l'accesso concorrente ai dati
  - effettuare il recovery a seguito di un malfunzionamento

Transazione

- Una transazione è una sequenza di operazioni che
  - rappresenta un'unità elementare di lavoro
  - può concludersi con un successo o un insuccesso
    - in caso di successo, il risultato delle operazioni eseguite deve essere memorizzato in modo permanente nella base di dati
    - in caso di insuccesso, la base di dati deve ritornare allo stato precedente l'inizio della transazione

Sistema transazionale

- Un sistema che mette a disposizione un meccanismo per la definizione e l'esecuzione di transazioni viene detto *sistema transazionale*
- I DBMS contengono blocchi architetturali che offrono servizi di gestione delle transazioni

Gestione delle transazioni

Transazioni in SQL

Transazione

- Una transazione è
  - un'unità logica di lavoro, non ulteriormente scomponibile
  - una sequenza di operazioni (istruzioni SQL) di modifica dei dati, che porta la base di dati da uno stato consistente a un altro stato consistente
    - non è necessario conservare la consistenza negli stati intermedi

Inizio di una transazione

- Per definire l'inizio di una transazione, il linguaggio SQL prevede l'istruzione
  - START TRANSACTION
- Di solito l'istruzione di inizio della transazione è omessa
  - l'inizio è implicito
    - prima istruzione SQL del programma che accede alla base di dati
    - prima istruzione SQL successiva all'istruzione di termine della transazione precedente

Fine di una transazione

- Il linguaggio SQL prevede istruzioni per definire la fine di una transazione
  - Transazione terminata con successo
    - COMMIT [WORK]
    - l'azione associata all'istruzione si chiama *commit*
  - Transazione terminata con insuccesso
    - ROLLBACK [WORK]
    - l'azione associata all'istruzione si chiama *abort*

Commit

- Azione eseguita quando una transazione termina con successo
- La base di dati è in un nuovo stato (finale) corretto
- Le modifiche dei dati effettuate dalla transazione divengono
  - permanenti
  - visibili agli altri utenti

Rollback

- Azione eseguita quando una transazione termina a causa di un errore
  - per esempio, di un errore applicativo
- Tutte le operazioni di modifica dei dati eseguite durante la transazione sono "annullate"
- La base di dati ritorna nello stato precedente l'inizio della transazione
  - i dati sono nuovamente visibili agli altri utenti

Esempio

- Trasferire la somma 100
  - dal conto corrente bancario IT92X0108201004300000322229
  - al conto corrente bancario IT32L0201601002410000278976

Esempio

- Trasferire la somma 100
  - dal conto corrente bancario IT92X0108201004300000322229
  - al conto corrente bancario IT32L0201601002410000278976

START TRANSACTION;

Esempio

- Trasferire la somma 100
  - dal conto corrente bancario IT92X0108201004300000322229
  - al conto corrente bancario IT32L0201601002410000278976

```
START TRANSACTION;
UPDATE Conto-Corrente
    SET Saldo = Saldo - 100
    WHERE IBAN='IT92X0108201004300000322229';
```

Esempio

- Trasferire la somma 100
  - dal conto corrente bancario IT92X0108201004300000322229
  - al conto corrente bancario IT32L0201601002410000278976

```
START TRANSACTION;
UPDATE Conto-Corrente
    SET Saldo = Saldo - 100
    WHERE IBAN='IT92X0108201004300000322229';
UPDATE Conto-Corrente
    SET Saldo = Saldo + 100
    WHERE IBAN= 'IT32L0201601002410000278976';
```

Esempio

- Trasferire la somma 100
  - dal conto corrente bancario IT92X0108201004300000322229
  - al conto corrente bancario IT32L0201601002410000278976

```
START TRANSACTION;
UPDATE Conto-Corrente
    SET Saldo = Saldo - 100
    WHERE IBAN='IT92X0108201004300000322229';
UPDATE Conto-Corrente
    SET Saldo = Saldo + 100
    WHERE IBAN= 'IT32L0201601002410000278976';
COMMIT;
```

Aggiornamento di più tabelle

- Modificare con il valore F9 il codice del fornitore F2

```
UPDATE F
SET CodF='F9'
WHERE CodF='F2';
```

- Se in FP esistono forniture che fanno riferimento ai codici dei fornitori aggiornati, è violato il vincolo di integrità referenziale
  - occorre aggiornare anche tali forniture in FP

Aggiornamento di più tabelle

- Modificare con il valore F9 il codice del fornitore F2

```
UPDATE F
SET CodF='F9'
WHERE CodF='F2';

UPDATE FP
SET CodF='F9'
WHERE CodF='F2';
```

Gestione delle transazioni

Proprietà delle transazioni

Proprietà delle transazioni

- Le proprietà principali delle transazioni sono
  - Atomicity – atomicità
  - Consistency – consistenza
  - Isolation – isolamento
  - Durability – persistenza (o durabilità)
- Sono riassunte dall'acronimo (inglese) *ACID*

Atomicità

- Una transazione è un'unità indivisibile (atomo) di lavoro
  - devono essere eseguite tutte le operazioni contenute nella transazione
  - oppure nessuna delle operazioni contenute nella transazione deve essere eseguita
    - la transazione non ha nessun effetto sulla base di dati
- La base di dati non può rimanere in uno stato intermedio assunto durante l'esecuzione di una transazione

Consistenza

- L'esecuzione di una transazione deve portare la base di dati
  - da uno stato iniziale consistente (corretto)
  - a uno stato finale consistente
- La correttezza è verificata dai vincoli di integrità definiti sulla base di dati
- Quando si verifica la violazione di un vincolo di integrità il sistema interviene
  - per annullare la transazione
  - oppure, per modificare lo stato della base di dati eliminando la violazione del vincolo

Isolamento

- L'esecuzione di una transazione è indipendente dalla contemporanea esecuzione di altre transazioni
- Gli effetti di una transazione non sono visibili dalle altre transazioni fino a quando la transazione non è terminata
  - si evita la visibilità di stati intermedi non stabili
    - uno stato intermedio può essere annullato da un rollback successivo
    - in caso di rollback, è necessario effettuare rollback delle altre transazioni che hanno osservato lo stato intermedio (effetto domino)

Persistenza

- L'effetto di una transazione che ha effettuato il commit è memorizzato in modo permanente
  - le modifiche dei dati eseguite da una transazione terminata con successo sono permanenti dopo il commit
- Garantisce l'affidabilità delle operazioni di modifica dei dati
  - i DBMS offrono meccanismi di ripristino dello stato corretto della base di dati dopo che si è verificato un guasto

Linguaggio SQL: costrutti avanzati

Controllo dell'accesso

Controllo dell'accesso

- Sicurezza dei dati
- Risorse e privilegi
- Gestione dei privilegi in SQL
- Gestione dei ruoli in SQL

Sicurezza dei dati

- Protezione dei dati da
  - letture non autorizzate
  - alterazione o distruzione
- Il DBMS fornisce strumenti per realizzare le protezioni, che sono definite dall'amministratore della base dati (DBA)

Sicurezza dei dati

- Il controllo della sicurezza verifica che gli utenti siano autorizzati a eseguire le operazioni che richiedono di eseguire
- La sicurezza è garantita attraverso un insieme di vincoli
  - specificati dal DBA in un opportuno linguaggio
  - memorizzati nel dizionario dei dati del sistema

Controllo dell'accesso

Risorse e privilegi

Risorse

- Qualsiasi componente dello schema di una base di dati è una risorsa
  - tabella
  - vista
  - attributo all'interno di una tabella o di una vista
  - dominio
  - procedura
  - ...
- Le risorse sono protette mediante la definizione di *privilegi di accesso*

Concetto di vista

- La vista è una *tabella "virtuale"*
  - il contenuto (tuple) è definito mediante un'interrogazione SQL sulla base di dati
    - il contenuto della vista dipende dal contenuto delle altre tabelle presenti nella base di dati
  - il contenuto *non* è memorizzato fisicamente nella basi di dati
    - è ricalcolato tutte le volte che si usa la vista eseguendo l'interrogazione che la definisce
- La vista è un oggetto della base di dati
  - è utilizzabile nelle interrogazioni come se fosse una tabella

DB forniture prodotti

P

CodP	NomeP	Colore	Taglia	Magazzino
P1	Maglia	Rosso	40	Torino
P2	Jeans	Verde	48	Milano
P3	Camicia	Blu	48	Roma
P4	Camicia	Blu	44	Torino
P5	Gonna	Blu	40	Milano
P6	Bermuda	Rosso	42	Torino

FP

CodF	CodP	Qta
F1	P1	300
F1	P2	200
F1	P3	400
F1	P4	200
F1	P5	100
F1	P6	100
F2	P1	300
F2	P2	400
F3	P2	200
F4	P3	200
F4	P4	300
F4	P5	400

F

CodF	NomeF	NSoci	Sede
F1	Andrea	2	Torino
F2	Luca	1	Milano
F3	Antonio	3	Milano
F4	Gabriele	2	Torino
F5	Matteo	3	Venezia

Esempio n.1

- Definizione della vista *piccoli fornitori*
  - i fornitori che hanno meno di 3 soci sono considerati “piccoli fornitori”

Esempio n.1

- Definizione della vista *piccoli fornitori*
  - i fornitori che hanno meno di 3 soci sono considerati "piccoli fornitori"
- La vista piccoli fornitori
  - contiene il codice, il nome, il numero di soci e la sede dei fornitori che hanno meno di 3 soci

Esempio n.1: definizione della vista

- Definizione della vista piccoli fornitori
  - contiene il codice, il nome, il numero di soci e la sede dei fornitori che hanno meno di 3 soci

```
SELECT CodF, NomeF, NSoci, Sede
FROM   F
WHERE Nsoci<3
```

Esempio n.1: definizione della vista

- Definizione della vista piccoli fornitori
  - contiene il codice, il nome, il numero di soci e la sede dei fornitori che hanno meno di 3 soci

```
SELECT CodF, NomeF, NSoci, Sede
FROM   F
WHERE Nsoci<3
```

Interrogazione associata alla vista

Esempio n.1: definizione della vista

- Definizione della vista piccoli fornitori
  - contiene il codice, il nome, il numero di soci e la sede dei fornitori che hanno meno di 3 soci

```
CREATE VIEW PICCOLI_FORNITORI AS
SELECT CodF, NomeF, NSoci, Sede
FROM   F
WHERE Nsoci<3;
```

Esempio n.1: definizione della vista

- Definizione della vista piccoli fornitori
  - contiene il codice, il nome, il numero di soci e la sede dei fornitori che hanno meno di 3 soci

Nome della vista

```
CREATE VIEW PICCOLI_FORNITORI AS
SELECT CodF, NomeF, NSoci, Sede
FROM   F
WHERE Nsoci<3;
```

Esempio n.1: interrogazione

- Visualizzare il codice, il nome, la sede e il numero di soci dei piccoli fornitori di Torino

Esempio n.1: interrogazione

- Visualizzare il codice, il nome, la sede e il numero di soci dei piccoli fornitori di Torino
- L'interrogazione può essere risolta senza l'uso di viste

```
SELECT *
FROM   F
WHERE NSoci<3 AND
          Sede='Torino';
```

Esempio n.1: interrogazione

- Visualizzare il codice, il nome, la sede e il numero di soci dei piccoli fornitori di Torino
- L'interrogazione può essere risolta usando la vista definita in precedenza

```
SELECT *
FROM   PICCOLI_FORNITORI
WHERE Sede='Torino';
```

Esempio n.1: interrogazione

- Visualizzare il codice, il nome, la sede e il numero di soci dei piccoli fornitori di Torino
- L'interrogazione può essere risolta usando la vista definita in precedenza

```
SELECT *
FROM   PICCOLI_FORNITORI
WHERE Sede='Torino';
```

- La vista PICCOLI_FORNITORI è usata come se fosse una tabella

Riscrittura delle interrogazioni

- Se l'interrogazione fa riferimento a una vista, deve essere riscritta dal DBMS prima dell'esecuzione
- La riscrittura è svolta automaticamente
  - si sostituiscono i riferimenti alla vista con la sua definizione

Esempio n.1: riscrittura dell'interrogazione

- Visualizzare il codice, il nome, la sede e il numero di soci dei piccoli fornitori di Torino

```
SELECT *
FROM   PICCOLI_FORNITORI
WHERE Sede='Torino';
```

Esempio n.1: riscrittura dell'interrogazione

- Visualizzare il codice, il nome, la sede e il numero di soci dei piccoli fornitori di Torino

```
SELECT CodF, NomeF, Sede, NSoci
FROM   PICCOLI_FORNITORI
WHERE Sede='Torino';
```

Esempio n.1: riscrittura dell'interrogazione

- Visualizzare il codice, il nome, la sede e il numero di soci dei piccoli fornitori di Torino

```
SELECT CodF, NomeF, Sede, NSoci
FROM   PICCOLI_FORNITORI
WHERE Sede='Torino';
```

- Riscrittura della clausola SELECT
  - si rendono espliciti gli attributi presenti nella definizione della vista

Esempio n.1: riscrittura dell'interrogazione

- Visualizzare il codice, il nome, la sede e il numero di soci dei piccoli fornitori di Torino

```
SELECT CodF, NomeF, Sede, NSoci
FROM   F
WHERE NSoci<3  AND
          Sede='Torino';
```

Esempio n.1: riscrittura dell'interrogazione

- Visualizzare il codice, il nome, la sede e il numero di soci dei piccoli fornitori di Torino

```
SELECT CodF, NomeF, Sede, NSoci
FROM   F
WHERE NSoci<3  AND
          Sede='Torino';
```

- Introduzione della definizione della vista
  - nella clausola FROM
  - nella clausola WHERE

Privilegi di accesso

- Descrivono i diritti di accesso alle risorse del sistema
- SQL offre meccanismi di controllo dell'accesso molto flessibili mediante i quali è possibile specificare
  - le risorse a cui possono accedere gli utenti
  - le risorse che devono essere mantenute private

Privilegi: caratteristiche

- Ogni privilegio è caratterizzato dalle seguenti informazioni
  - la risorsa a cui si riferisce
  - il tipo di privilegio
    - descrive l'azione permessa sulla risorsa

Privilegi: caratteristiche

- Ogni privilegio è caratterizzato dalle seguenti informazioni
  - la risorsa a cui si riferisce
  - il tipo di privilegio
    - descrive l'azione permessa sulla risorsa
  - l'utente che concede il privilegio
  - l'utente che riceve il privilegio

Privilegi: caratteristiche

- Ogni privilegio è caratterizzato dalle seguenti informazioni
  - la risorsa a cui si riferisce
  - il tipo di privilegio
    - descrive l'azione permessa sulla risorsa
  - l'utente che concede il privilegio
  - l'utente che riceve il privilegio
  - la facoltà di trasmettere il privilegio ad altri utenti

Tipi di privilegi (1/2)

- INSERT
  - permette di inserire un nuovo oggetto nella risorsa
  - vale per le tabelle e le viste

Tipi di privilegi (1/2)

- INSERT
  - permette di inserire un nuovo oggetto nella risorsa
  - vale per le tabelle e le viste
- UPDATE
  - permette di aggiornare il valore di un oggetto
  - vale per le tabelle, le viste e gli attributi

Tipi di privilegi (1/2)

- INSERT
  - permette di inserire un nuovo oggetto nella risorsa
  - vale per le tabelle e le viste
- UPDATE
  - permette di aggiornare il valore di un oggetto
  - vale per le tabelle, le viste e gli attributi
- DELETE
  - permette di rimuovere oggetti dalla risorsa
  - vale per le tabelle e le viste

Tipi di privilegi (2/2)

- SELECT
  - permette di utilizzare la risorsa all'interno di un'interrogazione
  - vale per le tabelle e le viste

Tipi di privilegi (2/2)

- SELECT
  - permette di utilizzare la risorsa all'interno di un'interrogazione
  - vale per le tabelle e le viste
- REFERENCES
  - permette di far riferimento a una risorsa nella definizione dello schema di una tabella
  - può essere associato a tabelle e attributi

Tipi di privilegi (2/2)

- SELECT
  - permette di utilizzare la risorsa all'interno di un'interrogazione
  - vale per le tabelle e le viste
- REFERENCES
  - permette di far riferimento a una risorsa nella definizione dello schema di una tabella
  - può essere associato a tabelle e attributi
- USAGE
  - permette di utilizzare la risorsa (per esempio, un nuovo tipo di dato) nella definizione di nuovi schemi

Privilegi del creatore della risorsa

- Alla creazione di una risorsa, il sistema concede tutti i privilegi su tale risorsa all'utente che ha creato la risorsa
- Solo il creatore della risorsa ha il privilegio di eliminare una risorsa (DROP) e modificarne lo schema (ALTER)
  - il privilegio di eliminare e modificare una risorsa non può essere concesso a nessun altro utente

Privilegi dell'amministratore del sistema

- L'amministratore del sistema (utente system) possiede tutti i privilegi su tutte le risorse

Controllo dell'accesso

Gestione dei privilegi in SQL

Gestione dei privilegi in SQL

- I privilegi sono concessi o revocati mediante le istruzioni SQL
  - GRANT
    - concede privilegi su una risorsa a uno o più utenti
  - REVOKE
    - toglie a uno o più utenti i privilegi che erano stati loro concessi

GRANT

GRANT *ElencoPrivilegi* ON *NomeRisorsa* TO *ElencoUtenti*
[WITH GRANT OPTION]

GRANT

GRANT *ElencoPrivilegi* ON *NomeRisorsa* TO *ElencoUtenti*
[WITH GRANT OPTION]

- *ElencoPrivilegi*
  - specifica l'elenco dei privilegi
  - ALL PRIVILEGES
    - parola chiave per identificare tutti i privilegi

GRANT

GRANT *ElencoPrivilegi* ON *NomeRisorsa* TO *ElencoUtenti*
[WITH GRANT OPTION]

- *ElencoPrivilegi*
  - specifica l'elenco dei privilegi
  - ALL PRIVILEGES
    - parola chiave per identificare tutti i privilegi
- *NomeRisorsa*
  - specifica la risorsa sulla quale si vuole concedere il privilegio

GRANT

GRANT *ElencoPrivilegi* ON *NomeRisorsa* TO *ElencoUtenti*
[WITH GRANT OPTION]

- *ElencoPrivilegi*
  - specifica l'elenco dei privilegi
  - ALL PRIVILEGES
    - parola chiave per identificare tutti i privilegi
- *NomeRisorsa*
  - specifica la risorsa sulla quale si vuole concedere il privilegio
- *ElencoUtenti*
  - specifica gli utenti a cui viene concesso il privilegio

Esempio n. 1

```
GRANT ALL PRIVILEGES ON P
TO Neri, Bianchi
```

- Agli utenti Neri e Bianchi sono concessi tutti i privilegi sulla tabella P

Esempio n. 1

```
GRANT ALL PRIVILEGES ON P
TO Neri, Bianchi
```

- Agli utenti Neri e Bianchi sono concessi tutti i privilegi sulla tabella P

```
GRANT UPDATE(CodP) ON P
TO PUBLIC
```

- A tutti gli utenti è concesso di modificare la colonna CodP della tabella P

GRANT

GRANT *ElencoPrivilegi* ON *NomeRisorsa* TO *ElencoUtenti*
[WITH GRANT OPTION]

- WITH GRANT OPTION
  - facoltà di trasferire il privilegio ad altri utenti

Esempio n. 2

```
GRANT SELECT ON F TO Rossi
                    WITH GRANT OPTION
```

- All'utente Rossi è concesso il privilegio di SELECT sulla tabella F
- L'utente Rossi ha facoltà di trasferire il privilegio ad altri utenti

REVOKE

REVOKE *ElencoPrivilegi* ON *NomeRisorsa* FROM *ElencoUtenti*
[RESTRICT|CASCADE]

REVOKE

REVOKE *ElencoPrivilegi* ON *NomeRisorsa* FROM *ElencoUtenti*
[RESTRICT|CASCADE]

- Il comando REVOKE può togliere
  - tutti i privilegi che erano stati concessi
  - un sottoinsieme dei privilegi concessi

Esempio n. 1

```
REVOKE UPDATE ON P FROM Bianchi
```

- All'utente Bianchi è revocato il privilegio di UPDATE sulla tabella P

REVOKE

REVOKE *ElencoPrivilegi* ON *NomeRisorsa* FROM *ElencoUtenti* [RESTRICT|CASCADE]

- RESTRICT
  - il comando non deve essere eseguito qualora la revoca dei privilegi all'utente comporti qualche altra revoca di privilegi
    - Esempio: l'utente ha ricevuto i privilegi con GRANT OPTION e ha propagato i privilegi ad altri utenti
  - valore di default

Esempio n. 1

```
REVOKE UPDATE ON P FROM Bianchi
```

- All'utente Bianchi è revocato il privilegio di UPDATE sulla tabella P
  - il comando non è eseguito se comporta la revoca del privilegio ad altri utenti

REVOKE

REVOKE *ElencoPrivilegi* ON *NomeRisorsa* FROM *ElencoUtenti* [RESTRICT|CASCADE]

- CASCADE
  - revoca anche tutti i privilegi che erano stati propagati
    - genera una reazione a catena
  - per ogni privilegio revocato sono
    - revocati in cascata tutti i privilegi concessi
    - rimossi tutti gli elementi della base di dati che erano stati creati sfruttando questi privilegi

Esempio n. 2

REVOKE SELECT ON F FROM Rossi CASCADE

- All'utente Rossi è revocato il privilegio di SELECT sulla tabella F
- L'utente Rossi aveva ricevuto il privilegio con GRANT OPTION
  - se Rossi ha propagato il privilegio ad altri utenti, il privilegio è revocato in cascata
  - se Rossi ha creato una vista utilizzando il privilegio di SELECT, la vista è rimossa

Controllo dell'accesso

Gestione dei ruoli in SQL

Concetto di ruolo (1/2)

- Il ruolo è un profilo di accesso
  - definito dall'insieme di privilegi che lo caratterizzano
- Ogni utente ricopre un ruolo predefinito
  - gode dei privilegi associati al ruolo

Concetto di ruolo (2/2)

- Vantaggi
  - controllo dell'accesso più flessibile
    - possibilità che un utente ricopra ruoli diversi in momenti diversi
  - semplificazione dell'attività di amministrazione
    - possibilità di definire un profilo di accesso in un momento diverso dalla sua attivazione
    - facilità nella definizione del profilo di nuovi utenti

Ruoli in SQL-3

- Definizione di un ruolo

  CREATE ROLE *NomeRuolo*

Ruoli in SQL-3

- Definizione di un ruolo

  CREATE ROLE *NomeRuolo*

- Definizione dei privilegi di un ruolo e del ruolo di un utente
  - istruzione GRANT

Ruoli in SQL-3

- Definizione di un ruolo

  CREATE ROLE *NomeRuolo*

- Definizione dei privilegi di un ruolo e del ruolo di un utente
  - istruzione GRANT
- Un utente in momenti diversi può ricoprire ruoli diversi
  - associazione dinamica di un ruolo a un utente

    SET ROLE *NomeRuolo*