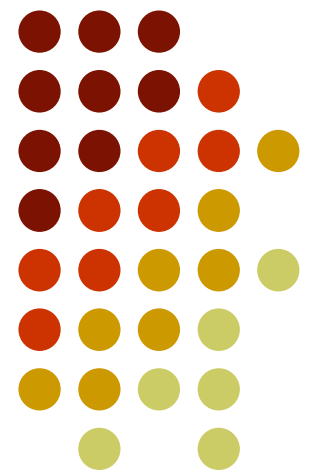


SQL: DML – istruzioni di aggiornamento

Elena Ferrari
Basi di Dati
A.A. 2018/2019





Operazioni di aggiornamento

- SQL fornisce tre comandi per la manipolazione del contenuto di relazioni:
 - **INSERT:** per l'inserimento di tuple
 - **UPDATE:** per la modifica di tuple
 - **DELETE:** per la cancellazione di tuple
- Le tuple oggetto del comando sono determinate tramite condizioni che possono coinvolgere anche altre relazioni

Operazioni di aggiornamento



- Le operazioni di aggiornamento modificano lo stato della base di dati:
 - è necessario verificare che siano rispettati i vincoli di integrità
- Ogni istruzione può aggiornare il contenuto di una sola tabella



Operazioni di inserimento

- Inserimento di una sola tupla:
 - assegnazione di un valore costante ad ogni attributo
- Inserimento di più tuple:
 - lette da altre tabelle mediante un'istruzione SELECT



Inserimento

● **INSERT INTO S [(C₁,C₂,...,C_n)]**
{VALUES (v₁,v₂,...,v_n) | sq};

- S è il nome della relazione su cui si esegue l'inserimento
- C₁,C₂,...,C_n è la lista delle colonne della nuova tupla (o delle nuove tuple) a cui si assegnano valori
- Tutte le colonne non esplicitamente elencate ricevono il valore nullo o il valore di default (se specificato nel comando di creazione di S)
- La mancata specifica di una lista di colonne equivale ad una lista che include tutte le colonne di S nell'ordine dato dal comando CREATE TABLE



Inserimento

- I valori da assegnare alla nuova tupla possono:
 - Essere specificati esplicitamente (clausola **VALUES**)
 - v_1, v_2, \dots, v_n è la lista di valori da assegnare alla nuova tupla
 - I valori sono assegnati in base ad una corrispondenza posizionale: il valore v_i ($i=1, \dots, n$) è assegnato alla colonna C_i
 - Tale lista può anche contenere il valore NULL o la parola chiave DEFAULT, per assegnare alla colonna il valore di default
 - Essere determinati tramite una sotto-interrogazione (**sq**)
 - Le tuple generate come risposta alla sotto-interrogazione sq vengono inserite nella relazione S (quindi più tuple possono essere inserite da un singolo comando)
 - Il dominio della colonna C_i ($i=1, \dots, n$) deve essere compatibile con il dominio della colonna (o espressione) i-esima contenuta nella clausola di proiezione di sq

Esempio



Film

titolo	regista	anno	genere	valutaz
underground	emir kusturica	1995	drammatico	3.20
edward mani di forbice	tim burton	1990	fantastico	3.60
nightmare before christmas	tim burton	1993	animazione	4.00
ed wood	tim burton	1994	drammatico	4.00
mars attacks	tim burton	1996	fantascienza	3.00
il mistero di sleepy hollow	tim burton	1999	horror	3.50
big fish	tim burton	2003	fantastico	3.10
la sposa cadavere	tim burton	2005	animazione	3.50
la fabbrica di cioccolato	tim burton	2005	fantastico	4.00
io non ho paura	gabriele salvatores	2003	drammatico	3.50
nirvana	gabriele salvatores	1997	fantascienza	3.00
mediterraneo	gabriele salvatores	1991	commedia	3.80
pulp fiction	quentin tarantino	1994	thriller	3.50
le iene	quentin tarantino	1992	thriller	4.00

Video

colloc	titolo	regista	tipo	Noleggio	colloc	dataNol	codCli	dataRest
1111	underground	emir kusturica	v		1111	01-Mar-2006	6635	02-Mar-2006
1112	underground	emir kusturica	d		1115	01-Mar-2006	6635	02-Mar-2006
1113	big fish	tim burton	v		1117	02-Mar-2006	6635	06-Mar-2006
1114	big fish	tim burton	d		1118	02-Mar-2006	6635	06-Mar-2006
1115	edward mani di forbice	tim burton	d		1111	04-Mar-2006	6642	05-Mar-2006
1116	nightmare before christmas	tim burton	v		1119	08-Mar-2006	6635	10-Mar-2006
1117	nightmare before christmas	tim burton	d		1120	08-Mar-2006	6635	10-Mar-2006
1118	ed wood	tim burton	d		1116	08-Mar-2006	6642	09-Mar-2006
1119	mars attacks	tim burton	d		1118	10-Mar-2006	6642	11-Mar-2006
1120	il mistero di sleepy hollow	tim burton	d		1121	15-Mar-2006	6635	18-Mar-2006
1121	la sposa cadavere	tim burton	d		1122	15-Mar-2006	6635	18-Mar-2006
1122	la fabbrica di cioccolato	tim burton	d		1113	15-Mar-2006	6635	18-Mar-2006
1123	la fabbrica di cioccolato	tim burton	d		1129	15-Mar-2006	6635	20-Mar-2006
1124	io non ho paura	gabriele salvatores	d		1119	15-Mar-2006	6642	16-Mar-2006
1125	nirvana	gabriele salvatores	d		1126	15-Mar-2006	6610	16-Mar-2006
1126	mediterraneo	gabriele salvatores	d		1112	16-Mar-2006	6610	18-Mar-2006
1127	pulp fiction	quentin tarantino	v		1114	16-Mar-2006	6610	17-Mar-2006
1128	pulp fiction	quentin tarantino	d		1128	18-Mar-2006	6642	20-Mar-2006
1129	le iene	quentin tarantino	d		1124	20-Mar-2006	6610	21-Mar-2006
					1115	20-Mar-2006	6610	21-Mar-2006
					1124	21-Mar-2006	6642	22-Mar-2006
					1116	21-Mar-2006	6610	?
					1117	21-Mar-2006	6610	?
					1127	22-Mar-2006	6635	?
					1125	22-Mar-2006	6635	?
					1122	22-Mar-2006	6642	?
					1113	22-Mar-2006	6642	?

Linguaggio SQL

Esempio



- Con valori espliciti:

- Vogliamo inserire un nuovo film, il **titolo** è ``La tigre e la neve'', il **regista** è Roberto Benigni, **l'anno di produzione** è 2005 e il **genere** è commedia:

```
INSERT INTO Film(titolo,regista,anno,genere)
VALUES ('la tigre e la neve',
       'roberto benigni',2005,'commedia');
```

- Non essendo specificata una valutazione, questa assume valore nullo



Esempio

```
INSERT INTO Video
```

```
VALUES (1130,'le iene','quentin tarantino','d');
```

- Vincolo di integrità referenziale:
 - è necessario che il film le iene di quentin tarantino sia già presente in Film
 - Altrimenti l'inserimento non viene consentito

Esempio



- Con tuple i cui valori sono ottenuti tramite una sotto-interrogazione:

```
INSERT INTO ProdottiMilanesi  
SELECT codice, descrizione FROM Prodotti  
WHERE LuogoProd = 'Milano';
```

- L'interrogazione è un'istruzione SELECT arbitraria
 - non può contenere la clausola ORDER BY



Cancellazione

- **DELETE FROM S [<alias>] [WHERE F];**

- S è il nome della relazione su cui si esegue la cancellazione
- Il nome della relazione può avere associato un alias se è necessario riferire tuple di tale relazione in una qualche sotto-interrogazione presente in F
- F è la clausola di qualificazione che specifica le tuple da cancellare
- Se non è specificata alcuna clausola di qualificazione, vengono cancellate tutte le tuple

Esempio



- Vogliamo cancellare il film “La tigre e la neve” appena inserito:

```
DELETE FROM Film  
WHERE titolo = 'la tigre e la neve'  
AND regista = 'roberto benigni';
```

- Vogliamo cancellare i clienti che non hanno effettuato noleggi nell'ultimo anno:

```
DELETE FROM Cliente  
WHERE codCli NOT IN (SELECT codCli FROM Noleggio  
WHERE dataNoI > (CURRENT_DATE - '1' YEAR));
```

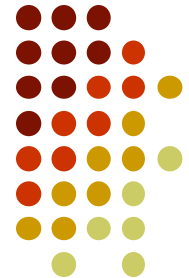


Esempio

Vogliamo cancellare il video con codice di collocazione 1111:

```
DELETE FROM Video  
WHERE colloc = 1111;
```

Esempio



Film

titolo	regista	anno	genere	valutaz
underground	emir kusturica	1995	drammatico	3.20
edward mani di forbice	tim burton	1990	fantastico	3.60
nightmare before christmas	tim burton	1993	animazione	4.00
ed wood	tim burton	1994	drammatico	4.00
mars attacks	tim burton	1996	fantascienza	3.00
il mistero di sleepy hollow	tim burton	1999	horror	3.50
big fish	tim burton	2003	fantastico	3.10
la sposa cadavere	tim burton	2005	animazione	3.50
la fabbrica di cioccolato	tim burton	2005	fantastico	4.00
io non ho paura	gabriele salvatores	2003	drammatico	3.50
nirvana	gabriele salvatores	1997	fantascienza	3.00
mediterraneo	gabriele salvatores	1991	commedia	3.80
pulp fiction	quentin tarantino	1994	thriller	3.50
le iene	quentin tarantino	1992	thriller	4.00

Video

colloc	titolo	regista	tipo	Noleggio	colloc	dataNol	codCli	dataRest
1111	underground	emir kusturica	v		1111	01-Mar-2006	6635	02-Mar-2006
1112	underground	emir kusturica	d		1115	01-Mar-2006	6635	02-Mar-2006
1113	big fish	tim burton	v		1117	02-Mar-2006	6635	06-Mar-2006
1114	big fish	tim burton	d		1118	02-Mar-2006	6635	06-Mar-2006
1115	edward mani di forbice	tim burton	d		1111	04-Mar-2006	6642	05-Mar-2006
1116	nightmare before christmas	tim burton	v		1119	08-Mar-2006	6635	10-Mar-2006
1117	nightmare before christmas	tim burton	d		1120	08-Mar-2006	6635	10-Mar-2006
1118	ed wood	tim burton	d		1116	08-Mar-2006	6642	09-Mar-2006
1119	mars attacks	tim burton	d		1118	10-Mar-2006	6642	11-Mar-2006
1120	il mistero di sleepy hollow	tim burton	d		1121	15-Mar-2006	6635	18-Mar-2006
1121	la sposa cadavere	tim burton	d		1122	15-Mar-2006	6635	18-Mar-2006
1122	la fabbrica di cioccolato	tim burton	d		1113	15-Mar-2006	6635	18-Mar-2006
1123	la fabbrica di cioccolato	tim burton	d		1129	15-Mar-2006	6635	20-Mar-2006
1124	io non ho paura	gabriele salvatores	d		1119	15-Mar-2006	6642	16-Mar-2006
1125	nirvana	gabriele salvatores	d		1126	15-Mar-2006	6610	16-Mar-2006
1126	mediterraneo	gabriele salvatores	d		1112	16-Mar-2006	6610	18-Mar-2006
1127	pulp fiction	quentin tarantino	v		1114	16-Mar-2006	6610	17-Mar-2006
1128	pulp fiction	quentin tarantino	d		1128	18-Mar-2006	6642	20-Mar-2006
1129	le iene	quentin tarantino	d		1124	20-Mar-2006	6610	21-Mar-2006
					1115	20-Mar-2006	6610	21-Mar-2006
					1124	21-Mar-2006	6642	22-Mar-2006
					1116	21-Mar-2006	6610	?
					1117	21-Mar-2006	6610	?
					1127	22-Mar-2006	6635	?
					1125	22-Mar-2006	6635	?
					1122	22-Mar-2006	6642	?
					1113	22-Mar-2006	6642	?

Linguaggio SQL

Esempio



```
CREATE TABLE Noleggio
(colloc DECIMAL(4) REFERENCES Video
ON DELETE CASCADE
ON UPDATE CASCADE,
dataNoI DATE DEFAULT CURRENT_DATE,
codCli DECIMAL(4) NOT NULL REFERENCES Cliente
ON DELETE CASCADE
ON UPDATE CASCADE,
dataRest DATE,
PRIMARY KEY (colloc,dataNoI),
UNIQUE (colloc,dataRest));
```



Esempio

- Il DBMS accetta la cancellazione e cancella anche tutte le tuple in Noleggio che fanno riferimento al video cancellato
- Cosa succede se invece di CASCADE ci fosse l'opzione NO ACTION?



Esempio

- Vogliamo cancellare il video con codice di collocazione 1111:

```
DELETE FROM Video
```

```
WHERE colloc = 1111;
```

- Se in Noleggio esistono noleggi che fanno riferimento al video cancellato, è violato il vincolo di integrità referenziale tra Noleggio e Video:
 - Il DBMS rifiuta la cancellazione



Esempio

- Vogliamo cancellare il video con codice di collocazione 1111:

DELETE FROM Video

WHERE colloc = 1111;

- Se in Noleggio esistono noleggi che fanno riferimento al video cancellato, è violato il vincolo di integrità referenziale tra Noleggio e Video:
 - Il DBMS rifiuta la cancellazione

Esempio



- Vogliamo cancellare il video con codice di collocazione 1111:
 - Occorre cancellare prima i Noleggi del video 1111

Esempio



```
DELETE FROM Noleggio  
WHERE colloc = 1111;
```

```
DELETE FROM Video  
WHERE colloc = 1111;
```



Modifica

- `UPDATE S [<alias>]`
`SET C1={e1 | NULL},, Cn={en | NULL}`
`[WHERE F];`
- S è il nome della relazione su cui si esegue la modifica
- S può avere associato un alias se è necessario riferire tuple di S in una qualche sotto-interrogazione presente in F

Modifica



- $C_i = \{e_i \mid \text{NULL}\}$ ($i=1, \dots, n$) è un'espressione di assegnamento che specifica che alla colonna C_i deve essere assegnato il valore dell'espressione e_i :
 - e_i può essere una costante, oppure un'espressione aritmetica o di stringa, spesso funzione dei valori correnti delle tuple da modificare, oppure una sotto-interrogazione
- Alternativamente si può specificare che alla colonna sia assegnato il valore nullo
- F è la clausola di qualificazione che specifica le tuple da modificare
- Se non è specificata alcuna clausola di qualificazione, vengono modificate tutte le tuple



Esempio

- Vogliamo raddoppiare le valutazioni dei film (in corrispondenza ad un cambio di scala da 0-5 a 0-10):

```
UPDATE Film
```

```
SET valutaz = valutaz * 2;
```

- Il cliente 6635 restituisce tutti i video che ha attualmente in noleggio:

```
UPDATE Noleggio
```

```
SET dataRest = CURRENT_DATE,
```

```
WHERE codCli = 6635 AND dataRest IS NULL;
```



Modifica

- Nel comando UPDATE le sotto-interrogazioni possono essere usate per:
 - Determinare le tuple da modificare (nella clausola di qualificazione)
 - Determinare i nuovi valori da assegnare alle tuple (nella clausola di assegnamento)

Esempio



Film

titolo	regista	anno	genere	valutaz
underground	emir kusturica	1995	drammatico	3.20
edward mani di forbice	tim burton	1990	fantastico	3.60
nightmare before christmas	tim burton	1993	animazione	4.00
ed wood	tim burton	1994	drammatico	4.00
mars attacks	tim burton	1996	fantascienza	3.00
il mistero di sleepy hollow	tim burton	1999	horror	3.50
big fish	tim burton	2003	fantastico	3.10
la sposa cadavere	tim burton	2005	animazione	3.50
la fabbrica di cioccolato	tim burton	2005	fantastico	4.00
io non ho paura	gabriele salvatores	2003	drammatico	3.50
nirvana	gabriele salvatores	1997	fantascienza	3.00
mediterraneo	gabriele salvatores	1991	commedia	3.80
pulp fiction	quentin tarantino	1994	thriller	3.50
le iene	quentin tarantino	1992	thriller	4.00

Video

colloc	titolo	regista	tipo	Noleggio	colloc	dataNol	codCli	dataRest
1111	underground	emir kusturica	v		1111	01-Mar-2006	6635	02-Mar-2006
1112	underground	emir kusturica	d		1115	01-Mar-2006	6635	02-Mar-2006
1113	big fish	tim burton	v		1117	02-Mar-2006	6635	06-Mar-2006
1114	big fish	tim burton	d		1118	02-Mar-2006	6635	06-Mar-2006
1115	edward mani di forbice	tim burton	d		1111	04-Mar-2006	6642	05-Mar-2006
1116	nightmare before christmas	tim burton	v		1119	08-Mar-2006	6635	10-Mar-2006
1117	nightmare before christmas	tim burton	d		1120	08-Mar-2006	6635	10-Mar-2006
1118	ed wood	tim burton	d		1116	08-Mar-2006	6642	09-Mar-2006
1119	mars attacks	tim burton	d		1118	10-Mar-2006	6642	11-Mar-2006
1120	il mistero di sleepy hollow	tim burton	d		1121	15-Mar-2006	6635	18-Mar-2006
1121	la sposa cadavere	tim burton	d		1122	15-Mar-2006	6635	18-Mar-2006
1122	la fabbrica di cioccolato	tim burton	d		1113	15-Mar-2006	6635	18-Mar-2006
1123	la fabbrica di cioccolato	tim burton	d		1129	15-Mar-2006	6635	20-Mar-2006
1124	io non ho paura	gabriele salvatores	d		1119	15-Mar-2006	6642	16-Mar-2006
1125	nirvana	gabriele salvatores	d		1126	15-Mar-2006	6610	16-Mar-2006
1126	mediterraneo	gabriele salvatores	d		1112	16-Mar-2006	6610	18-Mar-2006
1127	pulp fiction	quentin tarantino	v		1114	16-Mar-2006	6610	17-Mar-2006
1128	pulp fiction	quentin tarantino	d		1128	18-Mar-2006	6642	20-Mar-2006
1129	le iene	quentin tarantino	d		1124	20-Mar-2006	6610	21-Mar-2006
					1115	20-Mar-2006	6610	21-Mar-2006
					1124	21-Mar-2006	6642	22-Mar-2006
					1116	21-Mar-2006	6610	?
					1117	21-Mar-2006	6610	?
					1127	22-Mar-2006	6635	?
					1125	22-Mar-2006	6635	?
					1122	22-Mar-2006	6642	?
					1113	22-Mar-2006	6642	?

Esempio



- Supponiamo che la relazione Film contenga una colonna costo che indichi il costo del noleggio del Film. Vogliamo **diminuire del 10%** il costo dei noleggi dei **film usciti prima del 2011** e che sono stati **noleggiati da meno di 10 clienti**

Esempio



Film noleggiati da meno di 10 clienti:

```
SELECT titolo, regista  
FROM Noleggio NATURAL JOIN Video  
GROUP BY titolo, regista  
HAVING COUNT(DISTINCT codCli) < 10);
```

Esempio



Supponiamo che la relazione Film contenga una colonna costo che indichi il costo del noleggio del Film. Vogliamo diminuire del 10% il costo dei noleggi dei film usciti prima del 2011 e che sono stati noleggiati da meno di 10 clienti:

```
UPDATE Film
SET costo = costo * 0.9
WHERE anno < 2011
  AND (titolo, regista)
  IN (SELECT titolo, regista
      FROM Noleggio NATURAL JOIN Video
      GROUP BY titolo, regista
      HAVING COUNT(DISTINCT codCli) < 10);
```

Esempio



Impiegato(Nome, CF, Indirizzo, Stipendio, Resp^{Impiegato}, Dip^{Dipartimento})

Dipartimento(Nome, IdD, Manager^{Impiegato}, ManagerData_In)

Esempio



Vogliamo assegnare ai manager di ogni dipartimento, uno stipendio pari al 110% della media degli stipendi del dipartimento che dirigono

Esempio



Impiegati che dirigono un dipartimento:

```
SELECT Manager  
FROM Dipartimento;
```

Esempio



Vogliamo assegnare ai manager di ogni dipartimento, uno stipendio pari al 110% della media degli stipendi del dipartimento che dirigono:

```
UPDATE Impiegato I
SET Stipendio = (110% della media degli stipendi del dipartimento
                che dirigono)
WHERE CF IN (SELECT Manager
            FROM Dipartimento);
```


Esempio

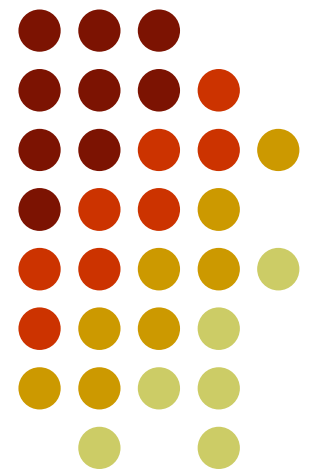


Vogliamo assegnare ai manager di ogni dipartimento, uno stipendio pari al 110% della media degli stipendi del dipartimento che dirigono:

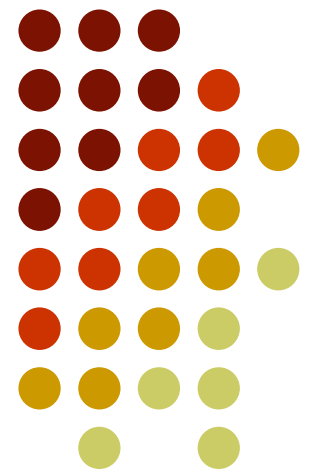
```
UPDATE Impiegato I
SET Stipendio = (SELECT 1.1 * AVG(Stipendio)
                  FROM Impiegato
                  WHERE Dip = I.Dip)
WHERE CF IN (SELECT Manager
              FROM Dipartimento);
```

SQL: Vincoli di integrità & Viste

Elena Ferrari
Basi di Dati
A.A. 2018/2019



SQL: Vincoli di integrità





Vincoli di integrità

- In SQL, abbiamo già discusso la specifica di:
 - vincoli di obbligatorietà di colonne (NOT NULL)
 - vincoli di chiave (UNIQUE e PRIMARY KEY)
 - vincoli di integrità referenziale (chiavi esterne, FOREIGN KEY)



Vincoli di integrità

- SQL mette a disposizione anche altri costrutti per la specifica di generici vincoli di integrità
 - Nel comando CREATE TABLE è possibile definire:
 - **Vincoli CHECK su colonna**
 - **Vincoli CHECK su relazione**
 - Possibilità di definire **asserzioni**
- Vincoli check ed asserzioni sfruttano il linguaggio di query per definire le condizioni che le tuple devono soddisfare per verificare il vincolo

Vincoli CHECK su colonna



- Alla specifica della colonna viene affiancata la parola chiave CHECK seguita da una condizione, cioè un predicato o una combinazione booleana di predicati
- Tale condizione può anche contenere sotto-interrogazioni che fanno riferimento ad altre relazioni



Vincoli CHECK su colonna

- Esempi:

```
CREATE TABLE Film ( ...  
    valutaz DECIMAL(3,2) CHECK (valutaz BETWEEN 0.00 AND 5.00),  
    ...);
```

```
CREATE TABLE Video ( ...  
    tipo CHAR NOT NULL CHECK (tipo IN ('d','v')),  
    ...);
```



Vincoli CHECK su relazione

- Alla definizione di una relazione viene aggiunta la parola chiave CHECK seguita da un predicato o una combinazione booleana di predicati
 - la condizione può contenere sotto-interrogazioni che fanno riferimento ad altre tabelle

- Esempio:

```
CREATE TABLE Noleggio
```

```
( ...
```

```
    CHECK (dataRest >= dataNoI));
```


Assegnare un nome ai vincoli



- È possibile assegnare un nome ai vincoli associati alle definizioni di relazioni facendo seguire la specifica del vincolo dalla parola chiave **CONSTRAINT** e dal nome
 - questa possibilità è prevista anche per i vincoli predefiniti
- Specificare un nome per i vincoli è utile per potervisi poi riferire (ad esempio per eliminarli, mediante `ALTER TABLE DROP CONSTRAINT`)

Esempio



```
CREATE TABLE Video
```

```
(colloc    DECIMAL(4) CONSTRAINT PKey PRIMARY KEY,  
titolo     VARCHAR(30) CONSTRAINT Tnn NOT NULL,  
regista    VARCHAR(20) CONSTRAINT Rnn NOT NULL,  
tipo       CHAR CONSTRAINT Snn NOT NULL DEFAULT 'd'  
           CONSTRAINT Tok CHECK (tipo IN ('d','v')),  
           CONSTRAINT FK FOREIGN KEY (titolo,regista));
```



Esempio

```
ALTER TABLE Video DROP CONSTRAINT Tok;
```

```
ALTER TABLE Video ADD CONSTRAINT Tok  
CHECK (tipo IN ('d','v','x'));
```

```
ALTER TABLE Video DROP CONSTRAINT Rnn;
```



Esempio

```
ALTER TABLE Video DROP CONSTRAINT Tok;
```

```
ALTER TABLE Video ADD CONSTRAINT Tok  
CHECK (tipo IN ('d','v','x'));
```

```
ALTER TABLE Video DROP CONSTRAINT Rnn;
```

Nota: l'aggiunta di un vincolo ad una relazione mediante ALTER TABLE ADD CONSTRAINT è possibile solo se tutte le tuple correntemente presenti nella relazione soddisfano il vincolo



Vincoli CHECK

- Mediante l'uso di sotto-interrogazioni è possibile esprimere vincoli CHECK che verificano condizioni arbitrarie
- E' però consigliabile esprimere tramite vincoli CHECK solo condizioni che fanno riferimento a singole tuple della relazione cui associamo il vincolo:
 - migliore comprensibilità dello schema
 - maggiore efficienza nella verifica dei vincoli
- Condizioni che richiedano di esaminare più tuple della relazione o tuple di relazioni diverse andrebbero invece espresse tramite **asserzioni** (se il DBMS le supporta)



Asserzioni

- Sono elementi dello schema, manipolate da appositi comandi del DDL
- Servono per esprimere vincoli di integrità che coinvolgono più tuple o più relazioni
- Sintassi:

```
CREATE ASSERTION <nome asserzione>  
CHECK (<condizione>);
```



Noleggio

colloc	dataNol	codCli	dataRest
1111	01-Mar-2006	6635	02-Mar-2006
1115	01-Mar-2006	6635	02-Mar-2006
1117	02-Mar-2006	6635	06-Mar-2006
1118	02-Mar-2006	6635	06-Mar-2006
1111	04-Mar-2006	6642	05-Mar-2006
1119	08-Mar-2006	6635	10-Mar-2006
1120	08-Mar-2006	6635	10-Mar-2006
1116	08-Mar-2006	6642	09-Mar-2006
1118	10-Mar-2006	6642	11-Mar-2006
1121	15-Mar-2006	6635	18-Mar-2006
1122	15-Mar-2006	6635	18-Mar-2006
1113	15-Mar-2006	6635	18-Mar-2006
1129	15-Mar-2006	6635	20-Mar-2006
1119	15-Mar-2006	6642	16-Mar-2006
1126	15-Mar-2006	6610	16-Mar-2006
1112	16-Mar-2006	6610	18-Mar-2006
1114	16-Mar-2006	6610	17-Mar-2006
1128	18-Mar-2006	6642	20-Mar-2006
1124	20-Mar-2006	6610	21-Mar-2006
1115	20-Mar-2006	6610	21-Mar-2006
1124	21-Mar-2006	6642	22-Mar-2006
1116	21-Mar-2006	6610	?
1117	21-Mar-2006	6610	?
1127	22-Mar-2006	6635	?
1125	22-Mar-2006	6635	?
1122	22-Mar-2006	6642	?
1113	22-Mar-2006	6642	?



Esempio

- Uno stesso video non può essere noleggiato contemporaneamente da due clienti:

```
CREATE ASSERTION SoloUno  
CHECK (NOT EXISTS  
      (SELECT colloc FROM Noleggio  
        WHERE dataRest IS NULL  
        GROUP BY colloc  
        HAVING COUNT(*) > 1));
```




Esempio

Un cliente non può avere più di tre video in noleggio contemporaneamente:

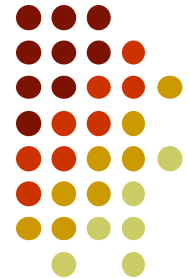
```
CREATE ASSERTION Max3  
CHECK (NOT EXISTS  
      (SELECT CodCli FROM Noleggio  
       WHERE dataRest IS NULL  
       GROUP BY codCli  
       HAVING COUNT(*) > 3));
```

Esempio



Un video non può essere noleggiato prima dell'uscita del film che lo contiene:

Esempio



Film

titolo	regista	anno	genere	valutaz
underground	emir kusturica	1995	drammatico	3.20
edward mani di forbice	tim burton	1990	fantastico	3.60
nightmare before christmas	tim burton	1993	animazione	4.00
ed wood	tim burton	1994	drammatico	4.00
mars attacks	tim burton	1996	fantascienza	3.00
il mistero di sleepy hollow	tim burton	1999	horror	3.50
big fish	tim burton	2003	fantastico	3.10
la sposa cadavere	tim burton	2005	animazione	3.50
la fabbrica di cioccolato	tim burton	2005	fantastico	4.00
io non ho paura	gabriele salvatores	2003	drammatico	3.50
nirvana	gabriele salvatores	1997	fantascienza	3.00
mediterraneo	gabriele salvatores	1991	commedia	3.80
pulp fiction	quentin tarantino	1994	thriller	3.50
le iene	quentin tarantino	1992	thriller	4.00

Video

colloc	titolo	regista	tipo	Noleggio	colloc	dataNol	codCli	dataRest
1111	underground	emir kusturica	v		1111	01-Mar-2006	6635	02-Mar-2006
1112	underground	emir kusturica	d		1115	01-Mar-2006	6635	02-Mar-2006
1113	big fish	tim burton	v		1117	02-Mar-2006	6635	06-Mar-2006
1114	big fish	tim burton	d		1118	02-Mar-2006	6635	06-Mar-2006
1115	edward mani di forbice	tim burton	d		1111	04-Mar-2006	6642	05-Mar-2006
1116	nightmare before christmas	tim burton	v		1119	08-Mar-2006	6635	10-Mar-2006
1117	nightmare before christmas	tim burton	d		1120	08-Mar-2006	6635	10-Mar-2006
1118	ed wood	tim burton	d		1116	08-Mar-2006	6642	09-Mar-2006
1119	mars attacks	tim burton	d		1118	10-Mar-2006	6642	11-Mar-2006
1120	il mistero di sleepy hollow	tim burton	d		1121	15-Mar-2006	6635	18-Mar-2006
1121	la sposa cadavere	tim burton	d		1122	15-Mar-2006	6635	18-Mar-2006
1122	la fabbrica di cioccolato	tim burton	d		1113	15-Mar-2006	6635	18-Mar-2006
1123	la fabbrica di cioccolato	tim burton	d		1129	15-Mar-2006	6635	20-Mar-2006
1124	io non ho paura	gabriele salvatores	d		1119	15-Mar-2006	6642	16-Mar-2006
1125	nirvana	gabriele salvatores	d		1126	15-Mar-2006	6610	16-Mar-2006
1126	mediterraneo	gabriele salvatores	d		1112	16-Mar-2006	6610	18-Mar-2006
1127	pulp fiction	quentin tarantino	v		1114	16-Mar-2006	6610	17-Mar-2006
1128	pulp fiction	quentin tarantino	d		1128	18-Mar-2006	6642	20-Mar-2006
1129	le iene	quentin tarantino	d		1124	20-Mar-2006	6610	21-Mar-2006
					1115	20-Mar-2006	6610	21-Mar-2006
					1124	21-Mar-2006	6642	22-Mar-2006
					1116	21-Mar-2006	6610	?
					1117	21-Mar-2006	6610	?
					1127	22-Mar-2006	6635	?
					1125	22-Mar-2006	6635	?
					1122	22-Mar-2006	6642	?
					1113	22-Mar-2006	6642	?



Esempio

Un video non può essere noleggiato prima dell'uscita del film che lo contiene:

```
CREATE ASSERTION DateOk
CHECK (NOT EXISTS
      (SELECT *
        FROM Noleggio NATURAL JOIN
          Video NATURAL JOIN Film
       WHERE EXTRACT(YEAR FROM dataNoI) < anno));
```



Controllo dei vincoli CHECK

- Un vincolo CHECK è violato se la valutazione della condizione di controllo restituisce FALSE come valore booleano di verità
- Cosa succede in presenza di valori nulli?
 - esempio: il vincolo `CHECK(dataRest >= dataNoI)` per noleggi in corso?
- Questo problema connesso alla presenza di valori nulli si ha anche per decidere quali sono le tuple risultato di una interrogazione



Valori nulli

- SQL usa una logica a tre valori per valutare il valore di verità di una condizione di ricerca:
 - TRUE (T), FALSE (F), UNKNOWN (?)
- UNKNOWN (?) indica che il valore di verità di una condizione di ricerca applicata ad una data tupla non è determinabile
- Un predicato semplice valutato su un attributo a valore nullo dà come risultato della valutazione ?
- Il valore di verità di un predicato complesso viene calcolato in base alle tabelle di verità nel lucido successivo

Valori nulli



AND			
	T	F	?
T	T	F	?
F	F	F	F
?	?	F	?

OR			
	T	F	?
T	T	T	T
F	T	F	?
?	T	?	?

NOT	
T	F
F	T
?	?



Esempio

- R

A	B	C	
a	?	c ₁	t ₁
a ₁	b	c ₂	t ₂
a ₂	?	?	t ₃

- **SELECT * FROM R WHERE A=a OR B=b;**

Il valore di verità della condizione per ogni tupla
è il seguente:

t₁ T OR ? → T

t₂ F OR T → T

t₃ F OR ? → ?

le tuple t₁ e t₂ verificano l'interrogazione



Valori nulli

- **SELECT * FROM R WHERE A=a AND B=b;**

il valore di verità della condizione per ogni tupla

è il seguente: t_1 T AND ? \rightarrow ?

t_2 F AND T \rightarrow F

t_3 F AND ? \rightarrow F

nessuna tupla verifica l'interrogazione

- **SELECT * FROM R WHERE NOT C=c₁;**

il valore di verità della condizione per ogni tupla

è il seguente: t_1 NOT T \rightarrow F

t_2 NOT F \rightarrow T

t_3 NOT ? \rightarrow ?

la tupla t_2 verifica l'interrogazione



Valori nulli

- Nelle espressioni (ad es. aritmetiche) se un argomento è NULL allora il valore dell'intera espressione è NULL
- Esempio: le tuple relative a noleggi correnti hanno durata (dataRest – dataNol) DAY indeterminata
- Nel calcolo delle funzioni di gruppo vengono escluse le tuple che hanno valore nullo per la colonna su cui la funzione è calcolata
 - $SUM(e1 + e2)$ può dare risultato diverso da $SUM(e1) + SUM(e2)$

Valori nulli



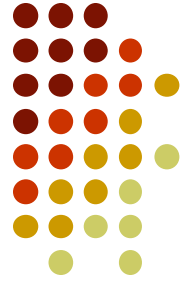
- Una funzione di gruppo può comunque restituire NULL se applicata ad un insieme vuoto di valori o contenente il solo valore NULL
- Se $e1$ ed $e2$ sono NULL, $e1=e2$ non è vero (è ?)



Valori nulli

- Una tupla per cui il valore di verità è ? non viene restituita dall'interrogazione
- Viceversa, in un vincolo di integrità se la valutazione della condizione di controllo restituisce ? il vincolo non è violato
 - il vincolo `CHECK(dataRest >= dataNol)` non è violato dai noleggi in corso

IS NULL



- Il predicato **IS NULL** applicato ad un attributo restituisce TRUE se la tupla ha valore nullo per l'attributo



Esempio

- R

A	B	C	
a	?	c ₁	t ₁
a ₁	b	c ₂	t ₂
a ₂	?	?	t ₃

- `SELECT * FROM R WHERE B IS NULL;`
restituisce le tuple t₁ e t₃
- `SELECT * FROM R WHERE B IS NULL AND C IS NULL;`
restituisce la tupla t₃
- `SELECT * FROM R WHERE B IS NULL OR C IS NULL;`
restituisce le tuple t₁ e t₃



IS NOT NULL

- Il predicato **IS NOT NULL** applicato ad un dato attributo di una tupla restituisce TRUE se la tupla ha valore non nullo per l'attributo
- Esempio:
 - **SELECT * FROM R WHERE B IS NOT NULL;**
restituisce la tupla t_2



Valori nulli

- `SELECT colloc FROM Noleggio`
`WHERE dataRest > CURRENT_DATE;`
non restituisce i noleggi in corso
- `SELECT colloc FROM Noleggio`
`WHERE dataNoI > DATE '1-Nov-2018' OR dataRest > DATE '1-Nov-2018';`
restituisce anche noleggi in corso, purché siano iniziati dopo il 1 di Novembre 2018
- `SELECT colloc FROM Noleggio`
`WHERE NOT dataRest < CURRENT_DATE;`
non restituisce i noleggi in corso



Valori nulli

- Le interrogazioni:

- SELECT colloc FROM Noleggio

- WHERE dataRest = CURRENT_DATE OR

- NOT dataRest = CURRENT_DATE;

- SELECT colloc FROM Noleggio

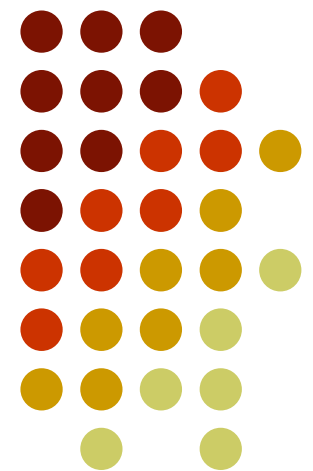
- WHERE dataRest = dataRest;

non restituiscono tutti i noleggi, ma solo i noleggi terminati, cioè:

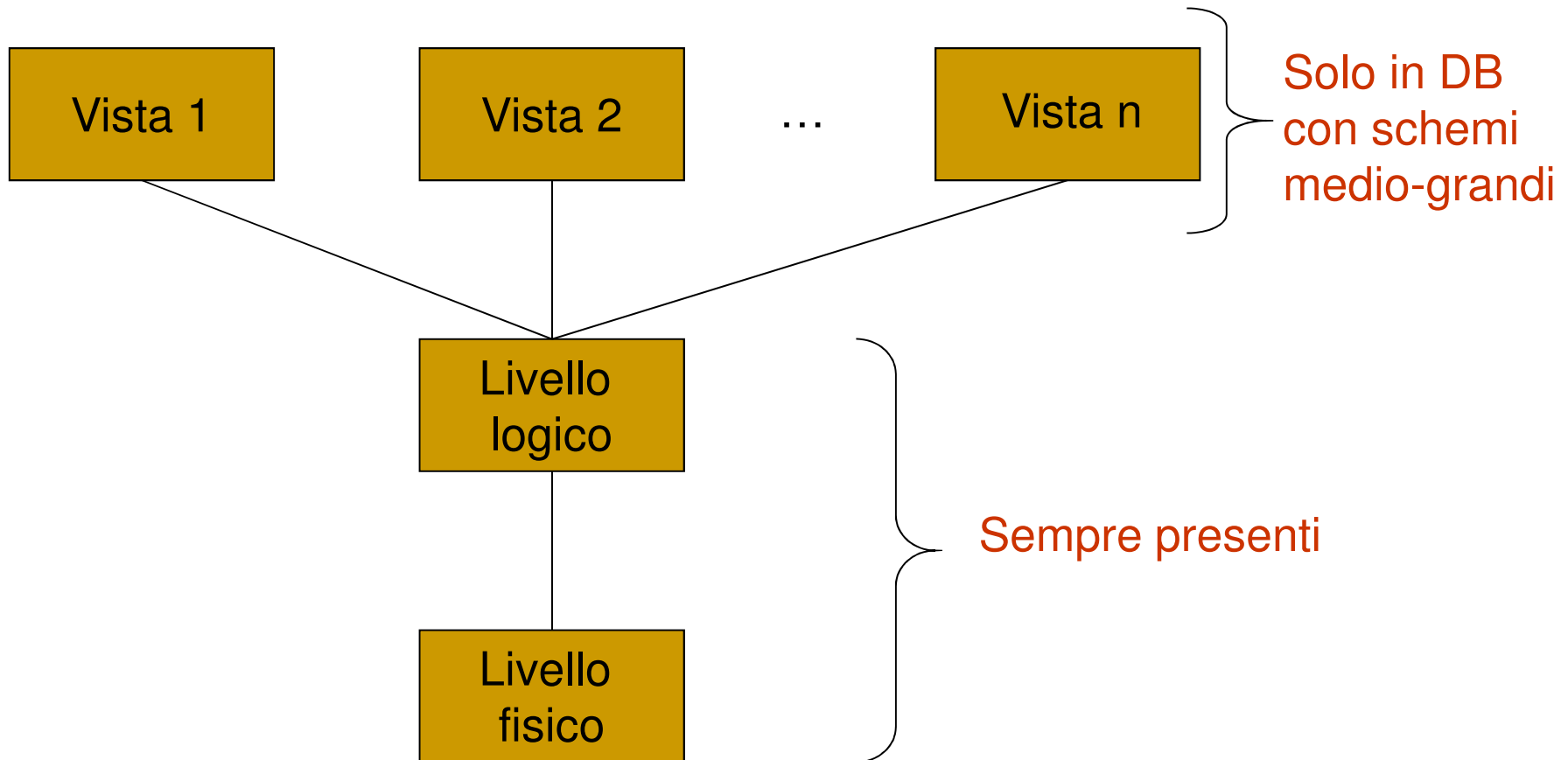
SELECT colloc FROM Noleggio

WHERE dataRest IS NOT NULL;

SQL: Viste



Livelli nella rappresentazione dei dati





Viste

- Una vista è una **relazione virtuale**
 - il contenuto (tuple) è definito mediante un'interrogazione SQL sulla base di dati
 - il contenuto della vista dipende dal contenuto delle altre relazioni (di base) presenti nella base di dati
- **il contenuto non è memorizzato fisicamente nella base di dati**
 - **è ricalcolato tutte le volte che si usa la vista eseguendo l'interrogazione che la definisce**



Viste

- Una vista è un oggetto della base di dati
 - può essere usata a quasi tutti gli effetti come una relazione di base
- Il meccanismo delle viste è utile per:
 - semplificare l'accesso ai dati
 - fornire indipendenza logica
 - garantire la protezione dei dati

Viste



Comando creazione di viste:

```
CREATE VIEW <nome vista> [( <lista nomi colonne> )]  
AS <interrogazione>  
[WITH [{LOCAL| CASCADED}] CHECK OPTION];
```

Viste



- **<nome vista>** è il nome della vista
- **<interrogazione>** è l'**interrogazione di definizione** della vista
le colonne della vista corrispondono in numero e dominio alle colonne specificate nella clausola di proiezione di tale interrogazione
- **<lista nomi colonne>** è una lista di nomi da assegnare alle colonne della vista:
 - la specifica non è obbligatoria, tranne nel caso in cui l'interrogazione contenga nella clausola di proiezione colonne virtuali cui non è assegnato un nome

Viste



Cancellazione di una vista:

```
DROP VIEW <nome vista>;
```


Esempio



Vista contenente il **codice cliente**, la **data di inizio noleggio** e la **collocazione** dei video **in noleggio** da più di tre giorni:

```
CREATE VIEW Nol3gg AS  
SELECT codCli, dataNol, colloc  
FROM Noleggio  
WHERE dataRest IS NULL AND  
      (CURRENT_DATE - dataNol) DAY > INTERVAL '3' DAY;
```

i nomi delle colonne della vista sono **codCli**, **dataNol** e **colloc**



Viste

- Nella definizione di viste è possibile usare tutte le funzionalità del linguaggio di interrogazione
- L'interrogazione di definizione di una vista può ad esempio contenere operazioni di join e fare uso di funzioni di gruppo ed espressioni:
 - **join**: può essere facile per alcuni utenti lavorare con una sola relazione piuttosto che eseguire join tra relazioni diverse (semplificazione dell'accesso ai dati)
 - **funzioni di gruppo**: può essere opportuno per alcuni utenti lavorare con dati aggregati piuttosto che con dati di dettaglio (garanzia di riservatezza)

Esempio



Vista che, per ogni cliente, contiene il codice, il numero di noleggi effettuati e la durata massima in giorni di tali noleggi:

```
CREATE VIEW InfoCli (codCli, numNol, durataM) AS  
  SELECT codCli, COUNT(*), MAX((dataRest - dataNol) DAY)  
  FROM Noleggio  
 GROUP BY codCli;
```

Interrogazioni e aggiornamenti su viste



- Una volta definita, una vista è parte dello schema della base di dati e dovrebbe poter essere manipolata dall'utente “come” una relazione di base
- Su una vista si possono eseguire:
 - interrogazioni
 - aggiornamenti, sotto opportune condizioni



Interrogazioni su viste

- Su una vista ad esempio è possibile:
 - effettuare proiezioni
 - specificare condizioni di ricerca
 - effettuare dei join con altre relazioni o viste
 - effettuare raggruppamenti e calcolare funzioni di gruppo
 - definire altre viste



Esempio

Vogliamo determinare dalla vista **InfoCli** le informazioni relative al cliente 1119:

```
SELECT *  
FROM InfoCli  
WHERE codCli= 1119;
```

Esempio



Vista creata su altra vista:

```
CREATE VIEW InfoCli2 AS  
SELECT codCli  
FROM InfoCli;
```



Aggiornamenti su viste

- L'esecuzione di un'operazione di aggiornamento su una vista deve poter essere propagata alle relazioni su cui la vista è definita
- Problemi:
 - in alcuni casi non è possibile realizzare l'operazione richiesta attraverso operazioni sulle relazioni di base oppure tale realizzazione non è univoca

Aggiornamenti su viste



Problema (esempio modifica):

- Una modifica su una colonna di una vista viene realizzata tramite una modifica sulla colonna corrispondente nella relazione di base
- Se la colonna della vista è virtuale (cioè definita tramite un'espressione) non è possibile stabilire quale valore assegnare alla colonna (o alle colonne) corrispondente nella relazione di base per ottenere il valore specificato nella modifica sulla vista

Esempio



```
CREATE VIEW InfoNoleggio AS  
SELECT colloc, (dataRest - dataNoI) DAY AS durata  
FROM Noleggio;
```



Aggiornamenti su viste

Problema (esempio inserimento):

- Un inserimento su una vista viene realizzato tramite un inserimento sulla relazione di base
- Se la vista non contiene una colonna della relazione di base su cui è specificato un vincolo NOT NULL e per il quale non è specificato nello schema un valore di default:
 - Il comando di inserimento sulla vista non specifica un valore per tale colonna
 - La colonna è obbligatoria e senza valore di default



Aggiornamenti su viste

Problema 2 (esempio cancellazione):

- Una cancellazione su una vista viene realizzata tramite una cancellazione sulla relazione di base
- La cancellazione di una tupla da una vista definita come join di più relazioni di base può essere ottenuta:
 - mediante cancellazione da una delle relazioni di base
 - mediante cancellazione da tutte le relazioni di base
 - ponendo a NULL il valore dell'attributo di join in una o più di tali relazioni



Aggiornamenti su viste

- Sono permesse solo quelle operazioni di aggiornamento che sono mappabili in modo univoco in operazioni sulle relazioni di base su cui la vista è definita
 - Molti DBMS consentono operazioni di aggiornamento solo su viste definite su singola relazione e ponendo delle restrizioni sulla query di formulazione (ad esempio non deve contenere GROUP BY...)



Aggiornamenti su viste

- Standard SQL-92

- sono aggiornabili le viste in cui una sola riga di ciascuna tabella di base corrisponde a una sola riga della vista
- corrispondenza univoca tra le tuple della vista e le tuple della tabella/e su cui è definita
- è possibile propagare senza ambiguità le modifiche apportate sulla vista verso ogni tabella su cui è definita



Aggiornamenti su viste

- Ad esempio, **non è aggiornabile** una vista che, nel blocco più esterno dell'interrogazione che la definisce:
 - non contiene la chiave primaria della tabella su cui è definita
 - contiene join
 - contiene funzioni aggregate
 - contiene DISTINCT



Check option

- Una vista essendo definita da un'interrogazione può contenere una condizione sul contenuto delle tuple delle relazioni su cui la vista è definita
- Solo le tuple che verificano tale condizione appartengono alla vista
- Un problema riguarda gli inserimenti nella vista di tuple che non verificano la condizione specificata dall'interrogazione di definizione della vista



Check option

- Esempio: supponiamo di voler inserire nella vista Nol3gg la tupla:
(1128, CURRENT_DATE, 6635)
 - la data di noleggio specificata è oggi
 - la condizione nell'interrogazione (noleggio iniziato almeno tre giorni fa) non è verificata dalla nuova tupla
 - la tupla viene inserita in Noleggio ma non è poi ritrovata da interrogazioni sulla vista Nol3gg
- Per assicurare che le tuple inserite tramite una vista (o modificate tramite una vista) siano accettate solo se verificano la condizione nell'interrogazione di definizione della vista, si usa la clausola **CHECK OPTION** del comando CREATE VIEW



Esempio

- Se Nol3gg è definita come:

```
CREATE VIEW Nol3gg AS  
SELECT codCli, dataNol, colloc  
FROM Noleggio  
WHERE dataRest IS NULL AND  
  (CURRENT_DATE - dataNol) DAY > INTERVAL '3' DAY;  
WITH CHECK OPTION;
```

L'inserimento di tuple che non soddisfano l'interrogazione di definizione della vista, come (1128, CURRENT_DATE, 6635), non è permesso



Check option

- La situazione si complica nel caso di viste definite in termini di altre viste, perché ognuna di tali viste potrebbe essere definita con CHECK OPTION
- Per tale motivo, la CHECK OPTION può essere specificata con due possibili alternative: LOCAL e CASCADED (default):
 - LOCAL : verifica solo la query di definizione della vista in oggetto
 - CASCADED: verifica ricorsivamente tutte le query di definizione delle viste coinvolte

Esempio



P

<u>CodP</u>	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



Esempio

```
CREATE VIEW PRODOTTI_TAGLIA_MEDIA_O_GRADE(CodP,  
NomeP, Taglia) AS  
SELECT CodP, NomeP, Taglia  
FROM P  
WHERE Taglia>=42 WITH CHECK OPTION;
```

- La vista è aggiornabile:
 - non si possono aggiornare le tuple presenti nella vista con valori di taglia minori di 42
 - non si possono inserire nella vista tuple con valori di taglia minori di 42



Esempio

```
CREATE VIEW PRODOTTI_TAGLIA_MEDIA (CodP, NomeP, Taglia)  
AS  
SELECT CodP, NomeP, Taglia  
FROM PRODOTTI_TAGLIA_MEDIA_O_GRADE  
WHERE Taglia<= 46 WITH CASCADED CHECK OPTION;
```

- La vista è aggiornabile:
 - si può aggiornare il contenuto della vista PRODOTTI_TAGLIA_MEDIA solo usando taglie comprese tra 42 e 46

Esempio



P

<u>CodP</u>	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



Esempio

```
UPDATE PRODOTTI_TAGLIA_MEDIA  
SET Taglia=Taglia-2;
```

- Con **CASCADED CHECK OPTION**:
 - aggiornamento vietato a causa di
PRODOTTI_TAGLIA_MEDIA_O_GRADE



Esempio

```
CREATE VIEW PRODOTTI_TAGLIA_MEDIA (CodP, NomeP, Taglia)  
AS  
SELECT CodP, NomeP, Taglia  
FROM PRODOTTI_TAGLIA_MEDIA_O_GRADE  
WHERE Taglia<= 46 WITH LOCAL CHECK OPTION;
```

- La vista è aggiornabile:
 - il controllo è effettuato **solo** sulla vista PRODOTTI_TAGLIA_MEDIA
 - si può aggiornare con taglie ≤ 46



Esempio

```
UPDATE PRODOTTI_TAGLIA_MEDIA  
SET Taglia=Taglia-2;
```

- Con LOCAL CHECK OPTION:
 - aggiornamento consentito

Caratteristiche di SQL non viste



- Alcune funzionalità non discusse:
 - SEQUENCE, per generare codici progressivi
 - colonne/relazioni derivate
 - nel linguaggio di interrogazione:
 - sotto-interrogazioni nella clausola FROM (danno luogo ad interrogazioni difficili da capire e da verificare)

Caratteristiche di SQL non viste



- Alcuni aspetti del modello dei dati di SQL che non sono stati trattati (vedi corso di modelli innovativi per la gestione dati):
 - caratteristiche object-relational:
 - tipi user-defined, tipi riga, tipi riferimento, tipi collezione
 - ereditarietà
 - trigger