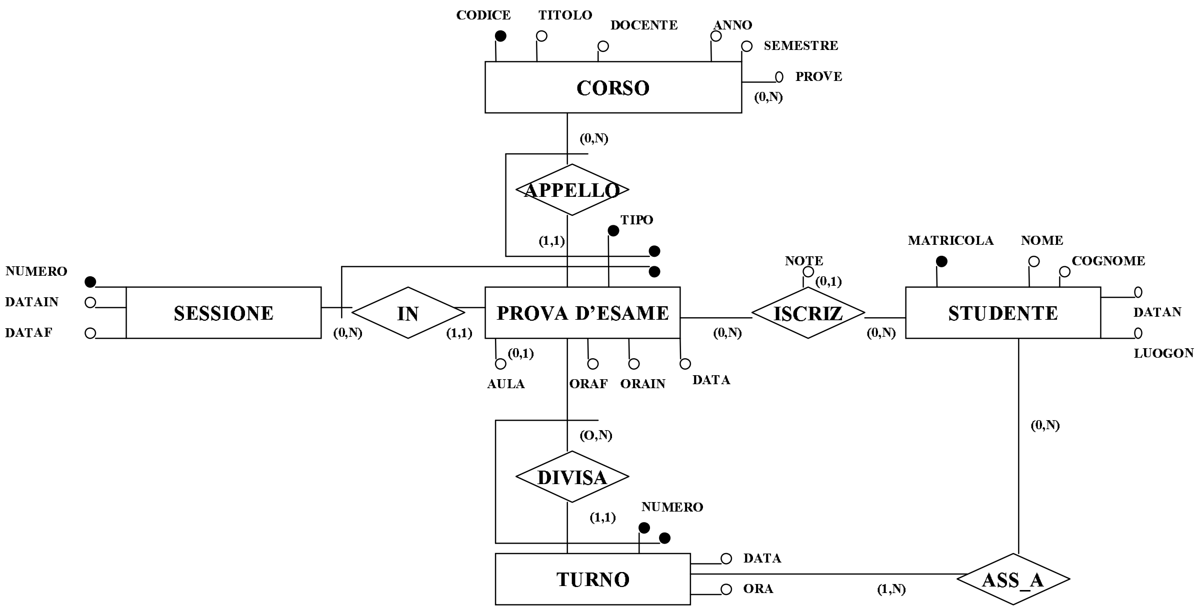
Corso di Basi di dati

26 Maggio 2004 - Soluzioni



(b)

- dom(PROVE) = dom (TIPO) = {s,o,l}
- in SESSIONE: DATAF > DATAIN + 7 (supponendo che una sessione d'esame duri almeno una settimana), se NUMERO di una sessione é maggiore di quello di un'altra, le date devono essere successive, non vi devono essere sovrapposizioni tra i periodi corrispondenti a sessioni diverse (DATAIN e DATAF sono chiavi alternative per SESSIONE)
- in PROVA D'ESAME: ORAF > ORAIN, DATA compresa tra DATAIN e DATAF della sessione corrispondente, TIPO appartentente alle PROVE previste per il corso corrispondente, AULA presente se DATA < CURRENT_DATE + 7
- una chiave alternativa per PROVA D'ESAME é APPELLO con DATA e ORAIN
- in TURNO: la prova d'esame corrispondente deve avere TIPO = l, la DATA deve essere >= della DATA della prova d'esame corrispondente
- se uno studente é iscritto a una prova d'esame che prevede piú turni la cui data e' < CURRENT_DATE + 7, allora deve essere assegnato ad un turno di quella prova – viceversa, uno studente appartenente ad un turno, deve essere iscritto alla prova d'esame corrsispondente
- se ad una prova d'esame di tipo laboratoratorio e data < CURRENT_DATE + 7 sono iscritti piú di 15 studenti, allora devono esistere dei turni corrispondenti a tale prova, ad ognuno dei quali devono essere assegnati non piú di 15 studenti
- non ci possono essere prove di corsi dello stesso anno con stessa data e orari sovrapposti
- non ci possono essere nella stessa aula o laboratorio prove né turni con stessa data e orari sovrapposti
- uno studente non si puø' iscrivere né puó essere assegnato a piú prove o turni con stessa data e orari sovrapposti.

(c) $CORSO(\underline{CodiceC}, Titolo, Docente, Anno, Semestre)$
$PROVE(\underline{CodiceC^*, TipoP})$
$SESSIONE(\underline{NumS}, DataIn, DataF)$
$STUDENTE(\underline{Matricola}, Nome, Cognome, DataN, LuogoN)$
$PROVAD'ESAME(\underline{CodiceC^*, NumS^*, TipoP}, Data, OraIn, OraF, Aula)$
$TURNO(\underline{CodiceC^*, NumS^*, TipoP^*, NumT}, Data, Ora)$
$ISCRIZ(\underline{Matricola^*, CodiceC^*, NumS^*, TipoP^*}, Note)$
$ASS(\underline{Matricola^*, CodiceC^*, NumS^*, TipoP^*, NumT^*})$

2. $PAGWEB(\underline{IDPag}, Indirizzo, Titolo, Testo)$
$CONTIENE(\underline{IDPag^*, Keyword})$
$VISITA(\underline{IDUt, IDPag^*, DataOra})$

(a) i. $\pi_{IdUt}(\sigma_{DataOra>='01/03/04:00:00'\ AND\ DataOra<='31/03/04:23:59'}VISITA \bowtie \sigma_{Indirizzo\ LIKE\ 'www.disi.unige.it/\%'}PAGWEB \bowtie \sigma_{Keyword='basi\ di\ dati'}CONTIENE)$

ii. $\pi_{IdUt}VISITA \setminus \pi_{IdUt}(VISITA \bowtie \sigma_{Keyword='basi\ di\ dati'}CONTIENE)$

iii. $\pi_{IdUt,IdPag}VISITA :$
$\pi_{IdPag}(\sigma_{Indirizzo\ LIKE\ 'www.disi.unige.it/\%'}PAGWEB \bowtie \sigma_{Keyword='basi\ di\ dati'}CONTIENE)$

(b) $\{t : \{IdUt\} \mid \exists p(\exists c(\exists v(p \in PAGWEB \wedge c \in CONTIENE \wedge v \in VISITA \wedge v.DataOra >=' 01/03/04 : 00 : 00' \wedge v.DataOra <=' 31/03/04 : 23 : 59' \wedge p.IDPag = v.IdPag \wedge p.Indirizzo\ LIKE\ 'www.disi.unige.it/\%' \wedge p.IDPag = c.IdPag \wedge c.Keyword = 'basi\ di\ dati' \wedge t.IDUt = v.IdUt)))\}$

(c) i.
```
SELECT Indirizzo, IDUt
FROM VISITA V NATURAL JOIN PAGINA
WHERE DataOra >= ALL (SELECT DataOra
                      FROM VISITA
                      WHERE IDUt = V.IDUt)
```

ii.
```
SELECT IDPag, COUNT(DISTINCT Keyword) AS NK, COUNT(DISTINCT IDUt,DataOra) AS NV,
       COUNT(DISTINCT IDUt) AS NU
FROM VISITA NATURAL JOIN CONTIENE
GROUP BY IDPag
```

iii. BDDISI: per le pagine del dominio `www.disi.unige.it` contenenti la keyword basi di dati il numero di visite e data e ora dell'ultima visita
```
BDDISI = SELECT IDPag, COUNT(DISTINCT IDUt,DataOra) AS NV, MAX(DataOra) AS DOU
FROM VISITA NATURAL JOIN CONTIENE
WHERE Indirizzo LIKE 'www.disi.unige.it/\%' AND Keyword = 'basi di dati'}
GROUP BY IDPag
```
```
SELECT X.IDPag AS Preferita, Y.IDPag AS Ultima
FROM BDDISI X, BDDISI Y
WHERE X.NV >= ALL (SELECT NV FROM BDDISI) AND
      Y.DOU >= ALL (SELECT DOU FROM BDDISI)
```

3. R(A,B,C,D,E), dipendenze funzionali:
$A \rightarrow BC, CD \rightarrow E, AC \rightarrow E, B \rightarrow D, E \rightarrow AB$

(a) che verifica:
$a_1\ b_1\ c_1\ d_1\ e_1$
$a_2\ b_2\ c_2\ d_2\ e_2$
che viola:
$a_1\ b_1\ c_1\ d_1\ e_1$
$a_1\ b_1\ c_1\ d_1\ e_2$

(b) in forma minimale:
$A \to B, A \to C, CD \to E, B \to D, E \to A$

(c) chiavi: A, E, CD, BC

(d) Lo schema non é in BCNF perché in $B \to D$ B non é superchiave, é in 3NF perché tutti gli attributi sono primi

(e) Lo schema stesso é in 3NF, preserva le dipenenze, non é in BCNF. Oppure, applicando l'algoritmo, ABC, CDE, BD, AE é in 3NF e preserva le dipendenze perché ottenuto dall'algoritmo ed é anche in BCNF.

(f) Applicando l'algoritmo: BD, con $B \to D$, chiave B e $ABCE$ con $A \to B, A \to C, CB \to E, E \to A$, chiavi A, E, BC. É ovviamente in 3NF e non preserva la dipendenza $CD \to E$.

4. (a) Il clustering di R ed S, su Z e K, rispettivamente, consiste nel memorizzare fisicamente vicine le tuple di R e S che hanno gli stessi valori per tali attributi. Vantaggioso per

```
SELECT * FROM R, S WHERE Z = K
```

svantaggioso per

```
SELECT * FROM S
```

(b) Si avrá quindi:

- indice denso non clusterizzato su $R.X$, cardinalitá $T(R)$
- indice denso non clusterizzato su $R.Y$, cardinalitá $V(Y, R)$
- indice sparso clusterizzato su $R.Z$, cardinalitá $B(R)$
- indice sparso clusterizzato su $S.K$, cardinalitá $B(S)$
- indice denso non clusterizzato su $S.L$, cardinalitá $V(L, S)$
- indice denso non clusterizzato su $S.W$, cardinalitá $V(W, S)$

Considero l'indice su $R.Y$

(c) operazione di aggiornamento velocizzata:

```
UPDATE R SET Z = k WHERE Y = y
```

operazione di aggiornamento rallentata:

```
UPDATE R SET Y = y WHERE Z = k
```

(d) operazione per cui l'indice é utile solo se ad albero:

```
SELECT * FROM R WHERE y1 < Y < y2
```

(e) operazione che potrebbe essere eseguita accedendo unicamente l'indice e non i dati:

```
SELECT DISTINCT Y FROM R
```