

A2 - 3.Progettazione e normalizzazione di un DB relazionale - 2NF

pag. A47 - A55 (parte terza)

seconda forma normale

Il test di **seconda forma normale (2NF)** comporta l'analisi della chiave primaria (e chiavi candidate) quando queste sono di tipo *composto* (quindi formata da più attributi).

Diciamo allora che uno schema di relazione è in 2NF se oltre ad essere in 1NF **non vi sono attributi non primi** (cioè non facenti parte della chiave primaria) **che dipendono in modo parziale dalla chiave primaria**, (cioè solo da qualcuno degli attributi primi facenti parte della chiave primaria) o in modo parziale da qualche altra chiave candidata.

Questa affermazione (**non escludendo nello schema la presenza di chiavi primarie o candidate composte**) equivale a dire che uno schema di relazione $Rel(A_1, A_2, \dots, A_n)$ è in 2NF se ogni attributo (o insieme di attributi) X dipende funzionalmente in modo completo da ogni chiave (primaria o candidata che sia) di Rel .

seconda forma normale

Consideriamo il seguente schema di relazione:

Imp_Prog (CodFisc, CodProg, *NominativoImp*, *OreLavorative*, *DenominazProg*, *SedeProg*);

per la registrazione dei dati degli impiegati che lavorano su vari progetti portati avanti in diverse sedi operative del dipartimento di ricerca.

<u>CodFisc</u>	<u>CodProg</u>	NominativoImp	OreLavorative	DenominazProg	SedeProg
PSRFRN11C22D333E	A-01	Franco Pisori	125	Progetto Alfa	Roma
PSRFRN11C22D333E	B-02	Franco Pisori	38	Progetto Beta	Torino
BPPRGI11C22D333E	B-02	Beppe Regi	82	Progetto Beta	Torino
PTRSMN11C22D333E	D-03	Simona Petri	110	Progetto Delta	Napoli
PTRSMN11C22D333E	A-01	Simona Petri	45	Progetto Alfa	Roma
Imp_Prog					

seconda forma normale

Ogni riga della tabella registra le ore di lavoro (*OreLavorative*) svolte da ogni impiegato (*CodFisc*, *NominativoImp*) su un certo progetto (*CodProg*, *DenominazProg*, *SedeProg*).

La chiave primaria è formata dal codice fiscale dell'impiegato insieme con il codice del progetto su cui sta lavorando, visto che **su ciascun progetto normalmente lavorano più impiegati, così come ogni impiegato può collaborare alla realizzazione di più progetti.**


<u>CodFisc</u>	<u>CodProg</u>	NominativoImp	OreLavorative	DenominazProg	SedeProg
PSRFRN11C22D333E	A-01	Franco Pisori	125	Progetto Alfa	Roma
PSRFRN11C22D333E	B-02	Franco Pisori	38	Progetto Beta	Torino
BPPRGI11C22D333E	B-02	Beppe Regi	82	Progetto Beta	Torino
PTRSMN11C22D333E	D-03	Simona Petri	110	Progetto Delta	Napoli
PTRSMN11C22D333E	A-01	Simona Petri	45	Progetto Alfa	Roma
Imp_Prog					

seconda forma normale

Esistono almeno le seguenti dipendenze funzionali:

(CodFisc, CodProg) → OreLavorative:

dal momento l'attributo *OreLavorative* indica le ore lavorative effettivamente svolte da ciascun impiegato su un determinato progetto;




<u>CodFisc</u>	<u>CodProg</u>	NominativoImp	OreLavorative	DenominazProg	SedeProg
PSRFRN11C22D333E	A-01	Franco Pisori	125	Progetto Alfa	Roma
PSRFRN11C22D333E	B-02	Franco Pisori	38	Progetto Beta	Torino
BPPRGI11C22D333E	B-02	Beppe Regi	82	Progetto Beta	Torino
PTRSMN11C22D333E	D-03	Simona Petri	110	Progetto Delta	Napoli
PTRSMN11C22D333E	A-01	Simona Petri	45	Progetto Alfa	Roma
Imp_Prog					

seconda forma normale

Esistono almeno le seguenti dipendenze funzionali:

CodFisc → NominativoImp:

cioè il codice fiscale dell'impiegato determina funzionalmente il suo nominativo, come è ovvio;




<u>CodFisc</u>	<u>CodProg</u>	NominativoImp	OreLavorative	DenominazProg	SedeProg
PSRFRN11C22D333E	A-01	Franco Pisori	125	Progetto Alfa	Roma
PSRFRN11C22D333E	B-02	Franco Pisori	38	Progetto Beta	Torino
BPPRGI11C22D333E	B-02	Beppe Regi	82	Progetto Beta	Torino
PTRSMN11C22D333E	D-03	Simona Petri	110	Progetto Delta	Napoli
PTRSMN11C22D333E	A-01	Simona Petri	45	Progetto Alfa	Roma
Imp_Prog					

seconda forma normale

Esistono almeno le seguenti dipendenze funzionali:

**CodProg → DenominazProg,
CodProg → SedeProg:**

dato che ogni progetto ha una sua denominazione e una sola sede di svolgimento



<u>CodFisc</u>	<u>CodProg</u>	NominativoImp	OreLavorative	DenominazProg	SedeProg
PSRFRN11C22D333E	A-01	Franco Pisori	125	Progetto Alfa	Roma
PSRFRN11C22D333E	B-02	Franco Pisori	38	Progetto Beta	Torino
BPPRGI11C22D333E	B-02	Beppe Regi	82	Progetto Beta	Torino
PTRSMN11C22D333E	D-03	Simona Petri	110	Progetto Delta	Napoli
PTRSMN11C22D333E	A-01	Simona Petri	45	Progetto Alfa	Roma
Imp_Prog					

seconda forma normale

Come possiamo vedere allora, gli attributi (non primi) *NominativoImp*, *DenominazProg* e *SedeProg* dipendono solo parzialmente dalla chiave primaria, cioè solo da una parte di essa; questa situazione può provocare alcuni problemi.

I dati di ciascun progetto (*CodProg*, *DenominazProg* e *SedeProg*) sono ripetuti in tutte le tuple relative a ogni impiegato che lavora su quel progetto, e questa ridondanza di dati, come sappiamo, può provocare *inconsistenza potenziale* (cioè **anomalie di aggiornamento**) e **anomalie di cancellazione**.

<u>CodFisc</u>	<u>CodProg</u>	NominativoImp	OreLavorative	DenominazProg	SedeProg
PSRFRN11C22D333E	A-01	Franco Pisori	125	Progetto Alfa	Roma
PSRFRN11C22D333E	B-02	Franco Pisori	38	Progetto Beta	Torino
BPPRGI11C22D333E	B-02	Beppe Regi	82	Progetto Beta	Torino
PTRSMN11C22D333E	D-03	Simona Petri	110	Progetto Delta	Napoli
PTRSMN11C22D333E	A-01	Simona Petri	45	Progetto Alfa	Roma
Imp_Prog					

seconda forma normale

Come possiamo vedere allora, gli attributi (non primi) *NominativoImp*, *DenominazProg* e *SedeProg* dipendono solo parzialmente dalla chiave primaria, cioè solo da una parte di essa; questa situazione può provocare alcuni problemi.

Inoltre non potremmo registrare i dati di un determinato progetto se non vi sono impiegati che ci lavorano sopra, dal momento che il codice fiscale di ciascun impiegato fa parte della chiave primaria, e in virtù del principio di *integrità sull'entità* non può essere omissso (***anomalie di inserimento***).

<u>CodFisc</u>	<u>CodProg</u>	NominativoImp	OreLavorative	DenominazProg	SedeProg
PSRFRN11C22D333E	A-01	Franco Pisori	125	Progetto Alfa	Roma
PSRFRN11C22D333E	B-02	Franco Pisori	38	Progetto Beta	Torino
BPPRGI11C22D333E	B-02	Beppe Regi	82	Progetto Beta	Torino
PTRSMN11C22D333E	D-03	Simona Petri	110	Progetto Delta	Napoli
PTRSMN11C22D333E	A-01	Simona Petri	45	Progetto Alfa	Roma
NULL	S-04	NULL	NULL	Progetto Sigma	Cosenza
Imp_Prog					

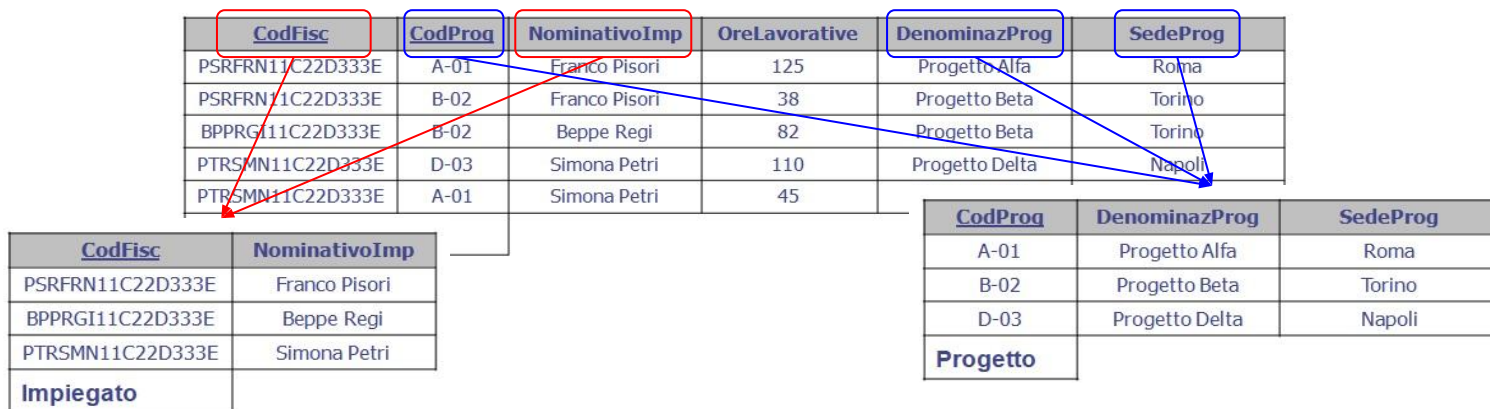
seconda forma normale

In sostanza, il problema deriva dal fatto che abbiamo utilizzato un unico schema di relazione per rappresentare informazioni di natura eterogenea, cioè i dati relativi agli impiegati (*CodFisc*, *NominativoImp*) e quelli che riguardano invece i progetti (*CodProg*, *DenominazProg*, *SedeProg*).

CodFisc	CodProg	NominativoImp	OreLavorative	DenominazProg	SedeProg
PSRFRN11C22D333E	A-01	Franco Pisori	125	Progetto Alfa	Roma
PSRFRN11C22D333E	B-02	Franco Pisori	38	Progetto Beta	Torino
BPPRGI11C22D333E	B-02	Beppe Regi	82	Progetto Beta	Torino
PTRSMN11C22D333E	D-03	Simona Petri	110	Progetto Delta	Napoli
PTRSMN11C22D333E	A-01	Simona Petri	45	Progetto Alfa	Roma
Imp_Prog					

seconda forma normale

La soluzione consiste allora nel **togliere** dallo schema di relazione **dato tutti gli attributi non chiave che dipendono solo parzialmente dalla chiave primaria** (nel nostro caso togliamo quindi: *NominativoImp*, *DenominazProg* e *SedeProg*), **aggiungendo così nuovi schemi di relazione nei quali questi attributi siano associati solo alla parte di chiave primaria da cui sono funzionalmente dipendenti**.



seconda forma normale

Quindi lo schema di relazione originario diventa:

Imp_Prog (*CodFisc*, *CodProg*, *OreLavorative*);

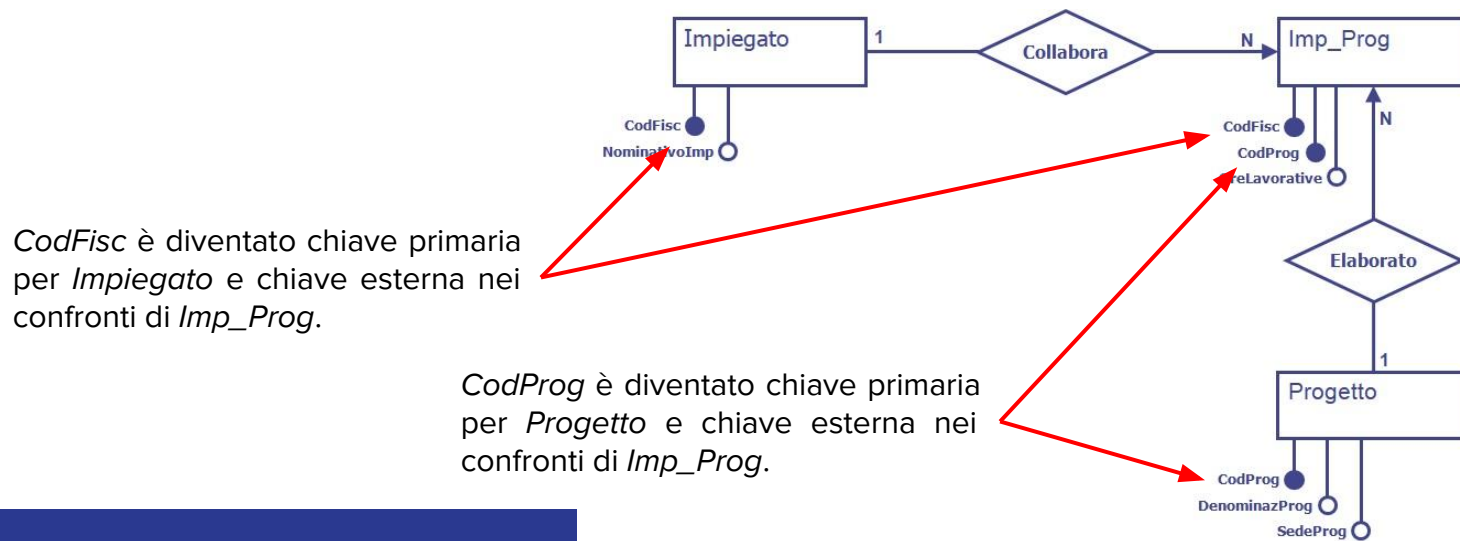
mentre vengono aggiunti i seguenti due schemi (tutti come si vede in 2NF):

Impiegato (*CodFisc*, *NominativoImp*) – per la dipendenza: **CodFisc** → **NominativoImp**;

Progetto (*CodProg*, *DenominazProg*, *SedeProg*) – per le dipendenze: **CodProg** → **DenominazProg** e
CodProg → **SedeProg**;

seconda forma normale

I nuovi schemi di relazione aggiunti non sono indipendenti dallo schema di partenza, ma in associazione *uno-a-molti* con questo; notiamo inoltre come la parte di chiave da cui dipendevano gli attributi tolti, è diventata chiave primaria per ciascun nuovo schema aggiunto, e dunque chiave esterna nello schema originario.



seconda forma normale

Consideriamo adesso il seguente schema di relazione:

Esame (*NumRegistr*, *Matricola*, *Nominativo*, *CodMateria*, *DataEsame*, *Esito*, *CodDocente*);

per la registrazione degli esami sostenuti dagli studenti di un ateneo universitario.

Se dovessimo considerare solo la chiave primaria, potremmo dire che questo schema rispetta il test di 2NF, visto che la chiave primaria è formata da un solo attributo (*NumRegistr*) e quindi non possono esserci dipendenze parziali dalla chiave primaria.

Oltre alla chiave primaria però è possibile definire anche una chiave candidata formata dagli attributi *Matricola*, *CodMateria* e *DataEsame*, e visto che sussiste la dipendenza funzionale *Matricola* \rightarrow *Nominativo*, possiamo dire che in effetti lo schema dato non è in 2NF, come invece sembrava a prima vista.

seconda forma normale

Anche in questo caso la soluzione consiste nel separare dallo schema originale gli attributi con la chiave parziale che provocano la non conformità alla 2NF:

Esame (*NumRegistr*, *CodMateria*, *DataEsame*, *Esito*, *CodDocente*, *Matricola*);

Studente (*Matricola*, *Nominativo*);

Di nuovo possiamo notare come lo schema di relazione aggiunto (*Studente*) sia legato allo schema di partenza (*Esame*) con un'associazione *uno-a-molti*; inoltre, la parte di chiave candidata (*Matricola*) da cui dipendevano gli attributi tolti, è diventata chiave primaria per il nuovo schema aggiunto, e dunque chiave esterna nello schema originario.