

Regole di Derivazione dal Modello Concettuale al Modello Logico

Prof. Fedeli Massimo
ITS Fabbrica 4.0 Tutti i diritti riservati

Introduzione

La trasformazione dal modello concettuale (schema Entità-Relazione) al modello logico (schema relazionale) è un processo fondamentale nella progettazione di basi di dati. Questo documento descrive le regole sistematiche per effettuare questa conversione.

1 Traduzione delle Entità

Regola Base: Ogni entità del modello E-R diventa una relazione (tabella) nel modello logico.

1.1 Componenti della traduzione

- Il **nome dell'entità** diventa il nome della relazione
- Gli **attributi dell'entità** diventano attributi (colonne) della relazione
- Gli **attributi identificatori** (chiave primaria) dell'entità diventano la chiave primaria della relazione

1.2 Esempio

Modello E-R:

STUDENTE (Matricola, Nome, Cognome, DataNascita)

Chiave: Matricola

Modello Logico:

STUDENTE (Matricola, Nome, Cognome, DataNascita)

PK: Matricola

2 Traduzione delle Relazioni (Associazioni)

La traduzione delle relazioni dipende dalla loro cardinalità.

2.1 Relazioni Uno a Molti (1:N)

Regola: Si propaga la chiave primaria dell'entità con cardinalità (1,1) o (0,1) nell'entità con cardinalità (0,N) o (1,N) come chiave esterna.

Procedura:

1. Non si crea una nuova tabella per la relazione
2. Si aggiunge la chiave primaria del lato "uno" come chiave esterna nel lato "molti"
3. Gli eventuali attributi della relazione vengono aggiunti all'entità sul lato "molti"

Esempio:

Modello E-R:

DIPARTIMENTO (CodDip, Nome)

IMPIEGATO (CodImp, Nome, Stipendio)

APPARTIENE: DIPARTIMENTO (1,1) -- (0,N) IMPIEGATO

Modello Logico:

DIPARTIMENTO (CodDip, Nome)

PK: CodDip

IMPIEGATO (CodImp, Nome, Stipendio, CodDip)

PK: CodImp

FK: CodDip REFERENCES DIPARTIMENTO

2.2 Relazioni Molti a Molti (N:M)

Regola: Si crea una nuova relazione (tabella) che rappresenta l'associazione.

Procedura:

1. Si crea una nuova tabella per la relazione
2. La chiave primaria è composta dalle chiavi primarie delle due entità coinvolte
3. Si aggiungono le chiavi esterne verso entrambe le entità
4. Gli eventuali attributi della relazione diventano attributi della nuova tabella

Esempio:

Modello E-R:

STUDENTE (Matricola, Nome)

CORSO (CodCorso, Titolo)

FREQUENTA: STUDENTE (0,N) -- (0,N) CORSO

Attributo: Voto

Modello Logico:

STUDENTE (Matricola, Nome)

PK: Matricola

CORSO (CodCorso, Titolo)

PK: CodCorso

FREQUENTA (Matricola, CodCorso, Voto)

PK: (Matricola, CodCorso)

FK: Matricola REFERENCES STUDENTE

FK: CodCorso REFERENCES CORSO

2.3 Relazioni Uno a Uno (1:1)

Regola: Esistono tre alternative, da scegliere in base al contesto.

Alternativa 1 - Accorpamento: Unire le due entità in un'unica tabella (se logicamente coerente).

Alternativa 2 - Chiave esterna lato opzionale: Propagare la chiave dell'entità con partecipazione obbligatoria (1,1) nell'entità con partecipazione opzionale (0,1).

Alternativa 3 - Tabella separata: Creare una tabella separata per la relazione (raramente usato).

Esempio (Alternativa 2):

Modello E-R:

PERSONA (CF, Nome, Cognome)

PASSAPORTO (NumPassaporto, DataRilascio, Scadenza)

POSSIEDE: PERSONA (0,1) -- (1,1) PASSAPORTO

Modello Logico:

PERSONA (CF, Nome, Cognome)

PK: CF

PASSAPORTO (NumPassaporto, DataRilascio, Scadenza, CF)

PK: NumPassaporto

FK: CF REFERENCES PERSONA

2.4 Relazioni Ternarie (e n-arie)

Regola: Si crea sempre una nuova relazione (tabella) che contiene le chiavi esterne di tutte le entità coinvolte.

Procedura:

1. Si crea una nuova tabella per la relazione ternaria
2. Si aggiungono le chiavi esterne verso tutte e tre (o più) le entità coinvolte
3. La chiave primaria è generalmente composta da tutte le chiavi esterne (salvo vincoli specifici di cardinalità)
4. Gli eventuali attributi della relazione diventano attributi della nuova tabella

Note sulla chiave primaria:

- Se tutte le cardinalità sono (0,N) o (1,N), la chiave primaria è composta da tutte le chiavi esterne

- Se una o più entità hanno cardinalità (0,1) o (1,1), potrebbero non far parte della chiave primaria
- La scelta dipende dall'analisi semantica della relazione

Esempio:

Modello E-R:

FORNITORE (CodFornitore, Nome, Città)

PRODOTTO (CodProdotto, Descrizione, Prezzo)

PROGETTO (CodProgetto, Titolo, Budget)

FORNITURA: FORNITORE (0,N) -- (0,N) PRODOTTO -- (0,N) PROGETTO

Attributi: Quantità, DataFornitura

Nota: Un fornitore può fornire lo stesso prodotto a progetti diversi, e uno stesso prodotto può essere fornito da fornitori diversi allo stesso progetto.

Modello Logico:

FORNITORE (CodFornitore, Nome, Città)

PK: CodFornitore

PRODOTTO (CodProdotto, Descrizione, Prezzo)

PK: CodProdotto

PROGETTO (CodProgetto, Titolo, Budget)

PK: CodProgetto

FORNITURA (CodFornitore, CodProdotto, CodProgetto, Quantità, DataFornitura)

PK: (CodFornitore, CodProdotto, CodProgetto)

FK: CodFornitore REFERENCES FORNITORE

FK: CodProdotto REFERENCES PRODOTTO

FK: CodProgetto REFERENCES PROGETTO

Esempio con cardinalità particolari:

Se nel modello E-R la cardinalità di PROGETTO fosse (0,1), indicando che ogni combinazione fornitore-prodotto può essere associata al massimo a un progetto, la chiave primaria cambierebbe:

FORNITURA (CodFornitore, CodProdotto, CodProgetto, Quantità, DataFornitura)

PK: (CodFornitore, CodProdotto)

FK: CodFornitore REFERENCES FORNITORE

FK: CodProdotto REFERENCES PRODOTTO

FK: CodProgetto REFERENCES PROGETTO

3 Traduzione delle Generalizzazioni (Gerarchie IS-A)

Esistono tre strategie principali per tradurre le gerarchie di generalizzazione.

3.1 Strategia della Relazione Unica

Descrizione: Si crea un'unica tabella che contiene tutti gli attributi dell'entità padre e di tutte le entità figlie.

Vantaggi: Semplice, nessun join necessario per recuperare i dati.

Svantaggi: Molti valori NULL se le sottoclassi hanno molti attributi specifici.

Quando usarla: Quando le sottoclassi hanno pochi attributi specifici.

Esempio:

Modello E-R:

PERSONA (CF, Nome, Cognome)

—STUDENTE (Matricola, CorsoLaurea)

—DOCENTE (Dipartimento, Ruolo)

Modello Logico:

PERSONA (CF, Nome, Cognome, TipoPersona, Matricola, CorsoLaurea, Dipartimento, Ruolo)

PK: CF

TipoPersona: discriminante per identificare il tipo

3.2 Strategia delle Relazioni Separate

Descrizione: Si crea una tabella per l'entità padre e una tabella separata per ogni entità figlia.

Vantaggi: Nessun valore NULL, struttura pulita.

Svantaggi: Richiede join per recuperare informazioni complete.

Quando usarla: Quando le sottoclassi hanno molti attributi specifici.

Esempio:

Modello E-R:

PERSONA (CF, Nome, Cognome)

—STUDENTE (Matricola, CorsoLaurea)

—DOCENTE (Dipartimento, Ruolo)

Modello Logico:

PERSONA (CF, Nome, Cognome)

PK: CF

STUDENTE (CF, Matricola, CorsoLaurea)

PK: CF

FK: CF REFERENCES PERSONA

DOCENTE (CF, Dipartimento, Ruolo)

PK: CF

FK: CF REFERENCES PERSONA

3.3 Strategia delle Sottoclassi Indipendenti

Descrizione: Si crea solo una tabella per ogni sottoclasse, duplicando gli attributi del padre.

Vantaggi: Nessun join necessario, nessun NULL.

Svantaggi: Ridondanza degli attributi del padre, problemi di integrità.

Quando usarla: Quando la generalizzazione è **totale e disgiunta**, e si accede raramente all'entità padre.

Esempio:

Modello E-R:

PERSONA (CF, Nome, Cognome)

 |—STUDENTE (Matricola, CorsoLaurea)

 |—DOCENTE (Dipartimento, Ruolo)

Modello Logico:

STUDENTE (CF, Nome, Cognome, Matricola, CorsoLaurea)

 PK: CF

DOCENTE (CF, Nome, Cognome, Dipartimento, Ruolo)

 PK: CF

4 Traduzione degli Attributi

4.1 Attributi Semplici

Diventano direttamente attributi della relazione.

4.2 Attributi Composti

Si decompongono nei loro componenti atomici.

Esempio:

Indirizzo (Via, Città, CAP) → Via, Città, CAP

4.3 Attributi Multivalore

Si creano tabelle separate con chiave esterna verso l'entità principale.

Esempio:

Modello E-R:

PERSONA (CF, Nome, {Telefono})

Modello Logico:

PERSONA (CF, Nome)

 PK: CF

TELEFONO (CF, NumeroTelefono)

 PK: (CF, NumeroTelefono)

 FK: CF REFERENCES PERSONA

4.4 Attributi Derivati

Generalmente non vengono memorizzati (si calcolano quando necessario), a meno che non ci siano esigenze di prestazioni.

5 Vincoli di Integrità

Durante la traduzione è importante preservare tutti i vincoli:

5.1 Vincoli di Chiave

- Chiave primaria (PRIMARY KEY)
- Chiave candidata (UNIQUE)

5.2 Vincoli di Integrità Referenziale

- Chiavi esterne (FOREIGN KEY)
- Politiche di cancellazione e aggiornamento (CASCADE, SET NULL, RESTRICT)

5.3 Vincoli di Dominio

- NOT NULL per partecipazioni obbligatorie
- CHECK per vincoli sui valori

5.4 Vincoli di Cardinalità

Alcuni vincoli di cardinalità complessi potrebbero richiedere trigger o asserzioni.

Conclusione

La derivazione dal modello concettuale al modello logico è un processo sistematico che, se seguito correttamente, garantisce uno schema relazionale corretto, efficiente e fedele al modello concettuale di partenza. La conoscenza approfondita di queste regole è fondamentale per ogni progettista di basi di dati.