

Progetto Basi di Dati 2014-2015

di Alessandro Sieni e Gianluca Mondini

Indice

1 Introduzione	3
2 Progettazione Concettuale	3
2.1 Glossario degli attributi	3
2.2 Associazioni	3
3 Progettazione Logica	3
3.1 Analisi di alcune operazioni effettuabili	3
3.1.1 Tabella dei volumi	3
3.1.2 Operazione 1	3
3.1.3 Operazione 2	3
3.1.4 Operazione 3	4
3.1.5 Operazione 4	4
3.1.6 Operazione 5	5
3.1.7 Operazione 6	6
3.1.8 Operazione 7	6
3.1.9 Operazione 8	6
3.1.10 Operazione 9	6
3.2 Analisi Ridondanze	6
3.3 Dipendenze funzionali e normalizzazione (BCNF)	6
3.4 Schema logico	6
4 Creazione Database	?

1 Introduzione

2 Progettazione Concettuale

2.1 Glossario degli attributi

2.2 Associazioni

sede_magazzino

- Cardinalità (1,N) con Magazzino in quanto ogni Sede può avere uno o più magazzini
- Cardinalità (1,1) con Sede in quanto ogni magazzino deve necessariamente appartenere ad una sede soltanto

magazzino_scaffale

- Cardinalità (1,N) con Scaffale in quanto ogni Magazzino possiede uno o più scaffali

- Cardinalità (1,1) con Magazzino in quanto ogni Scaffale deve necessariamente appartenere ad un Magazzino soltanto

scaffale_confezione

- Cardinalità (0,N) con Confezione, in quanto ogni Scaffale può contenere più Confezioni (oppure nessuna Confezione)
- Cardinalità (1,1) con Scaffale, in quanto ogni confezione deve necessariamente appartenere ad uno Scaffale soltanto

confezione_ingredientente

- Cardinalità (1,1) con Ingrediente, in quanto ogni Confezione deve necessariamente contenere un Ingrediente soltanto
- Cardinalità (0,N) con Confezione, in quanto possono essere presenti più Confezioni di un singolo Ingrediente (oppure nessuna)

sede_menu

- Cardinalità (1,N) con Menù, in quanto ogni Sede può avere più di un Menù disponibile
- Cardinalità (0,1) con Sede, in quanto un singolo Menù può essere disponibile presso una Sede (oppure presso nessuna Sede)

ingrediente_piatto

- Cardinalità (1,1) con Piatto, in quanto ogni Piatto deve necessariamente avere un Ingrediente principale soltanto
- Cardinalità (0,1) con Ingrediente...non dovrebbe essere 0 a molti? Ogni ingrediente può essere ingrediente principale di più piatti

sede_sala

- Cardinalità (1,N) con Sala, in quanto ogni Sede può avere più di una Sala
- Cardinalità (1,1) con Sede, in quanto ogni Sala deve necessariamente appartenere ad una Sede soltanto

sala_serata

- Cardinalità (0,N) con Serata, in quanto ogni Sala può essere occupata per diverse Serate (ma anche nessuna)
- Cardinalità (1,1) con Sala, in quanto ogni Serata deve essere necessariamente effettuata in una Sala soltanto

sala_tavolo

- Cardinalità (1,N) con Tavolo, in quanto ogni Sala deve essere formata da almeno un Tavolo
- Cardinalità (1,1) con Sale, in quanto ogni Tavolo deve necessariamente appartenere ad ogni Sala soltanto

serata_account

- Cardinalità (1,1) con Account, in quanto ogni Serata deve necessariamente essere organizzata da un Account soltanto
- Cardinalità (0,N) con Serata, in quanto ogni Account può organizzare più Serate (ma anche nessuna)

tavolo_account

- Cardinalità (1,1) con Account, in quanto ogni Tavolo deve necessariamente essere prenotato da un Account soltanto
- Cardinalità (0,N) con Tavolo, in quanto ogni Account può prenotare diversi Tavoli (ma anche nessuno)

account_proposta (effettua)

- Cardinalità (0,N) con Proposta, in quanto ogni Account può effettuare più Proposte (ma anche nessuna)
- Cardinalità (1,1) con Account, in quanto ogni Proposta deve necessariamente essere effettuata da un Account soltanto

account_proposta (valuta)

- Cardinalità (0,N) con Proposta, in quanto ogni Account può valutare più Proposte (ma anche nessuna)
- Cardinalità (1,N) con Account, in quanto ogni Proposta può essere valutata da più Account (ma anche nessuno)

account_comanda

- Cardinalità (0,N) con Comanda, in quanto ogni Account può effettuare più Comande (ma anche nessuna)
- Cardinalità (0,1) con Account, in quanto ogni Comanda può essere effettuata da un solo Account (ma anche da nessuno nel caso in cui la prenotazione non provenga da un Account)

comanda_pony

- Cardinalità (0,1) con Pony, in quanto ogni Comanda può interessare al massimo un Pony
- Cardinalità (1,N) con Comanda, in quanto ogni Pony ...

pony_consegna

- Cardinalità (0,N) con Consegna, in quanto ogni Pony può effettuare più Consegne (ma anche nessuna)
- Cardinalità (1,N) con Pony, in quanto ...

piatto_comanda

- Cardinalità (1,N) con Comanda, in quanto ogni Piatto può essere richiesto in una o più Comande

- Cardinalità (1,N) con Piatto, in quanto ogni Comanda può richiere uno o più piatti

piatto_passo

- Cardinalità (1,N) con Passo, in quanto ogni Piatto deve necessariamente essere prepatato mediante uno o più Passi
- Cardinalità (1,1) con Piatto, in quanto ogni Passo può far parte della preparazione di un Piatto soltanto

passo_strumento

- Cardinalità (1,1) con Strumento, in quanto ogni Passo ..

sede_strumento

- Cardinalità (1,1) con Sede in quanto ogni Strumento può appartenere ad una Sede soltanto
- Cardinalità (1,N) con Strumento in quanto ogni Sede deve necessariamente possedere uno o più Strumenti

confezione_passi

- Cardinalità (1,1) con Confezione in quanto ogni Passo richiede una Confezione soltanto

account_recensione (scrive)

- Cardinalità (0,N) con Recensione in quanto ogni Account può scrivere più di una Recensione (ma anche nessuna)
- Cardinalità (1,1) con Account, in quanto ogni Recensione deve essere scritta da un Account soltanto

account_recensione (valuta)

- Cardinalità (0,N) con Account, in quanto ogni Recensione può essere valutata da diversi Account (ma anche nessuno)
- Cardinalità (0,N) con Account, in quanto ogni Account può valutare diverse Recensioni (ma anche nessuna)

3 Progettazione Logica

3.1 Analisi di alcune operazioni effettuabili

3.1.1 Tabella dei volumi

Nella tabella dei volumi sono indicate approssimativamente la quantità di istanze di ogni singola associazione ed entità che saranno memorizzate nel sistema che andremo a realizzare (ovviamente la tabella non indica il peso effettivo in termini di Byte di ogni istanza ma solo il numero delle istanze di esse che si presume siano presenti all'interno del nostro sistema). Con la colonna " Tipo " verrà indicato se le istanze di tale riga saranno riferite ad una associazione(R) o a delle entità(E).

Nome	Tipo	Volume	Motivazione
Sede	E	5	Si presume che il sistema centralizzato gestisca 5 sedi differenti
Magazzino_Sede	R	10	In media ogni Sede ha 2 magazzini
Magazzino	E	10	Cardinalità (1,1) con Magazzino_Sede

3.1.2 Operazione 1

Testo. Identificare tutti i piatti preparabili nell'attuale menu delle sedi presenti a Roma.

Frequenza. 70 volte al giorno

Con ridondanza

Query.

```
Create or replace view PiattiRoma as
select * from Piatto P, Menu_Piatto MP, M, Sede S where
P.Id = MP.Id_Piatto and MP.Id_Menu = M.id and M.Sede = S.id
and S.Citta = "Roma" and M.DataFine is NULL;

Select * from PiattiRoma as PR ,Ingredienti_Piatto as IP where
PR.id = IP.ID_Piatto and IP.qta >
(Select sum(qta) from Magazzino M,Sede S,Scaffale SC, Confezione C where
M.Sede = S.Id and S.citta = "Roma" and SC.Id_Magazzino = S.Id and C.scaffale =
SC.id
and C.ingrediente = IP.ingrediente group by C.ingrediente);
```

Tavola degli accessi.

Senza ridondanza

Query.

Tavola degli accessi.

3.1.3 Operazione 2

Testo. Per ogni sede, identificare il cliente che ha effettuato il maggior numero di recensioni nel mese attuale

Frequenza. 1 volta al mese

Senza Ridondanza

Query.

```
Create or replace view Piatti_Sedi as
select * from Piatto P, Menu_Piatto MP, M, Sede S where
P.Id = MP.Id_Piatto and MP.Id_Menu = M.id and M.Sede = S.id
and M.DataFine is NULL;
```

```

Select * from Piatti_Sede PS, Passi P, Ingredienti I where
Ps.Id = P.Id_Piatto and P.ID_Ingrediente = I.id
group by PS.Id
Having (count(*) = (SELECT count(*) from Passi P2, Ingredienti I2
where P2.ID_Ingrediente = I2.ID and P2.Id_Piatto = P.Id_Piatto
and I2.Allergene IS NOT NULL)) order by PS.NomeSede;

```

Tavola degli accessi.

Con Ridondanza

Query.

```

Create or replace view Piatti_Sedi as
select * from Piatto P, Menu_Piatto MP, M, Sede S where
P.Id = MP.Id_Piatto and MP.Id_Menu = M.id and M.Sede = S.id
and M.DataFine is NULL;

Select * from Piatti_Sedi as PS group by PS.IdPiatto
having(count(*) = (select count(*) from PiattiSedi as PS2
where PS2.IdPiatto = PS.IdPiatto and PS2.Allergene IS NOT NULL));

```

Tavola degli accessi.

3.1.4 Operazione 3

Testo: Visualizzare l'attuale menu esclusi i piatti che contengono almeno un allergene nella sede di Firenze.

Frequenza : 200 volte al giorno.

Senza Ridondanza

Query.

```

Create or replace view PiattiFirenze as
select * from Piatto P, Menu_Piatto MP, M, Sede S where
P.Id = MP.Id_Piatto and MP.Id_Menu = M.id and M.Sede = S.id
and S.Citta = "Firenze" and M.DataFine is NULL;

Select * from PiattiFirenze as PR ,Ingredienti_Piatto as IP,Ingredienti AS I
where
PR.id = IP.ID_Piatto and I.id = IP.Id_Ingrediente
and IP.qta >
(Select sum(qta) from Magazzino M,Sede S,Scaffale SC, Confezione C where
M.Sede = S.Id and S.citta = "Firenze" and SC.Id_Magazzino = S.Id and
C.scaffale = SC.id
and C.ingrediente = IP.ingrediente group by C.ingrediente) and NOT EXISTS
(select * from I where I.allergene is NULL);

```

Tavola degli accessi.

Con Ridondanza

Query.

```

Create or replace view PiattiFirenze as

```

```
select * from Piatto P, Menu_Piatto MP, M, Sede S where
P.Id = MP.Id_Piatto and MP.Id_Menu = M.id and M.Sede = S.id
and S.Citta = "Firenze" and M.DataFine is NULL;
```

```
Select * from PiattiFirenze as PR ,Ingredienti_Piatto as IP,Ingredienti AS I
where
PR.id = IP.ID_Piatto and I.id = IP.Id_Ingrediente
and IP.qta >
(Select sum(qta) from Magazzino M,Sede S,Scaffale SC, Confezione C where
M.Sede = S.Id and S.citta = "Firenze" and SC.Id_Magazzino = S.Id and
C.scaffale = SC.id
and C.ingrediente = IP.ingrediente group by C.ingrediente) and PR.allergene is
NULL;
```

Tavola degli accessi.

3.1.5 Operazione 4

Testo. Visualizzare tutti i comuni (in ordine decrescente) che hanno effettuato richieste take-away

Frequenza. 10 volte al giorno

Qui le ridondanze trovate non incidono in alcun modo sul comportamento della query

Query.

```
SELECT Nome,count(*) AS N_Consegne FROM Consegna C
GROUP BY C.Comune ORDER BY N_Consegne asc
```

Tavola degli accessi.

3.1.6 Operazione 5

Testo. Visualizzare il piatto più richiesto per ogni sede

Frequenza. 2 volte al giorno

Query.

Questo dovrebbe visualizzare i piatti con il numero di ordinazioni totali
indipendentemente dalla sede, come posso usare GROUP BY per suddividerli
in sedi?

```
SELECT
    p.nome,
    COUNT(p.nome) AS "Numero di ordinazioni"
FROM
    piatto p
    INNER JOIN
    piatto_menu pm
    ON
    pm.id_piatto = p.nome
```

3.1.7 Operazione 6

Testo. Mostrare quale è stato il piatto, per ogni sede, ad aver avuto il maggior numero di recensioni positive

Frequenza. 2 volte al giorno

Query.

Stesso problema della query precedente

Tavola degli accessi.

3.1.8 Operazione 7

Testo. Inserire un piatto in un menu presente in una sede a scelta

Frequenza. 1 volta al giorno

Con Ridondanza

Query: .

Query (senza ridondanza).

```
CREATE OR REPLACE VIEW MenuAttuali AS
SELECT
    sede,
    idmenu
FROM
    menu
WHERE
    data_fine IS NULL;

INSERT INTO piatto_menu
    (SELECT idmenu, "lasagne" FROM MenuAttuali WHERE sede = "roma")
```

Tavola degli accessi.

3.1.9 Operazione 8

Testo. Per ogni magazzino indicare quali sono le confezioni che sono in scadenza o che sono state aperte(quindi che hanno un degrado più veloce).

Frequenza. 100 volte al giorno

Query.

```
SELECT
    C.id_confe
FROM
    confezione C
WHERE
    # lo stato in uso non viene considerato perché la confezione è
    # già in uso, giusto?
    # come era stato implementato il tipo Stato?
    C.stato = "parziale"
```


dove ottengo la data di scadenza?

Tavola degli accessi.

3.1.10 Operazione 9

Test. Indicare il guadagno (al netto delle spese degli ingredienti) al termine di ogni giornata per ciascuna sede

Frequenza. 1 volta al giorno

Senza ridondanza

Query.

Tavola degli accessi.

Con ridondanza

Query.

Tavola degli accessi.

3.2 Analisi Ridondanze

3.3 Dipendenze funzionali e normalizzazione (BCNF)

3.4 Schema logico

4 Creazione Database