

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 Modello Logico	3
3.2 Analisi di alcune operazioni effettuabili	3
3.2.1 Tabella dei volumi	3
3.2.2 Operazione 1	3
3.2.3 Operazione 2	4
3.2.4 Operazione 3	4
3.2.5 Operazione 4	5
3.2.6 Operazione 5	6
3.2.7 Operazione 6	6
3.2.8 Operazione 7	6
3.2.9 Operazione 8	6
3.2.10 Operazione 9	6
3.3 Analisi Ridondanze	6
3.4 Dipendenze funzionali e normalizzazione (BCNF)	6
3.5 Schema logico	?
4 Creazione Database	?

1 Introduzione

2 Progettazione Concettuale

2.1 Glossario degli attributi

2.2 Associazioni

3 Progettazione Logica

3.1 Modello Logico

- Sede(IdSede,via,nCivico,Citta)

- Magazzino(IdMagazzino,IdSede)
- Scaffale(IdScaffale,IdMagazzino)
- Confezione(IdConfezione,Peso,PrezzoAcquisto,DataAcquisto,DataConsegna,DataScadenza,Aspetto,Stato,QuantitaRimanente,Ingrediente,Scaffale)
- Ingrediente(IdIngrediente,Nome,Provenienza,TipoProduzione,Allergene)
- Strumenti(IdStrumento,Tipo,Nome)
- UtilizziStrumento(Utilizzo,Strumento)
- Utilizzi(IdUtilizzo,DescrizioneUtilizzo)
- Ricetta(IdRicetta,TestoRicetta)
- IngredientiRicetta(Ricetta,Ingrediente,Quantità)
- Passo(Ricetta,nPasso,DescrizionePasso,Strumento,TempoUtilizzo,Ingrediente,QuantitaUtilizzata)
- Menu(IdMenu,Sede,DataInizio,DataFine)
- Piatto(IdPiatto,Menu,Ricetta,Novita)
- VariazioniPiatto(IdPiatto,DescrizioneVariazione)
- Comanda(IdComanda,Tavolo,Ora,TakeAway,Account,Stato,Prezzo)
- Ordine(IdOrdine,Comanda,Piatto,Variazione1,Variazione2,Variazione3,OrdineConsegna,Stato)
- Prenotazione(IdPrenotazione,Account,NumeroTelefono,Tavolo,OraPrenotazione,nPersone)
- Pony(IdPony,TipoMezzo,Stato)
- StatoConsegna(IdStato,Comanda,Pony,Stato,Ora,Data)
- Account(Username,Password,Nome,Cognome,Via,nCivico,Comune,Citta,FruibilitàPrenotazioni,Sesso)
- Recensione(IdRecensione,Account,GiudizioGlobale,GiudizioTesto)
- DomandeQuestionario(IdDomanda,Domanda)
- Compilazione(IdDomanda,IdRecensione,IdRisposta)
- Risposte(IdRisposta,Risposta,Scala)
- PossibilitaRisposte(IdDomanda,IdRisposta)
- ValutazioneRecensione(Account,Recensione,Veridicita,Accuratezza,Descrizione)
- PropostaPiatto(IdPropostaPiatto,Account,Nome)
- IngredientiNuovoPiatto(PropostaPiatto,Ingrediente,Quantità)
- ValutazionePropostaPiatto(Account,PropostaPiatto,Valutazione,Descrizione)
- VariantePiatto(IdVariante,Account,Piatto)
- ModificaVariazione(IdModifica,VariantePiatto,Modifica)
- ValutazioneVariazione(Account,VariantePiatto,Valutazione)
- Serata(IdSerata,Account,NomeOrganizzatore,CognomeOrganizzatore,TelefonoOrganizzatoreSala,Allestimento,nPersone)

3.2 Analisi di alcune operazioni effettuabili

3.2.1 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
Possiede	R	10	In media ogni Sede ha 2 magazzini
Magazzino	E	10	Cardinalità (1,1) con Possiede
Scaffallatura	R	400	In media un magazzino ha 40 scaffali (40 * 10)
Scaffale	E	400	Cardinalità(1,1) con Scaffale
Ingrediente	E	100	Si presume che siano presenti 100 ingredienti
Contenimento	R	2000	In media ci sono 20 confezioni per ingrediente (20*100 = 2000)
Confezione	E	2000	Cardinalità (1,1) con Contenimento
Stoccaggio	R	5	In media uno scaffale conterrà 5 confezioni (2000 / 400 = 5)
Ricetta	E	100	Si ipotizza che siano inserite nel sistema 100 diverse ricette
Principale	R	50	Si ipotizza che in media solo 0,5 ricette richiedano un ingrediente principale

3.2.2 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, 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.

3.2.3 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.

3.2.4 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.

3.2.5 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.

3.2.6 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

```

3.2.7 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.

3.2.8 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.

3.2.9 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.

```

CREATE OR REPLACE VIEW ScadenzaConfezioni AS
SELECT idConfe AS "ID_confezione"
      , idScaffale AS "ID_scaffale"
      , aspetto
      , stato
      , ingrediente
      , DATEDIFF(day, dataScadenza, GETDATE()) AS "giorniAllaScadenza"
FROM confezione
WHERE stato = "in uso";

SELECT *
FROM ScadenzaConfezioni
WHERE giorniAllaScadenza <= 4
ORDER BY giorniAllaScadenza ASC;

```

Tavola degli accessi.

3.2.10 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.

3.3 Analisi Ridondanze

3.4 Dipendenze funzionali e normalizzazione (BCNF)

3.5 Schema logico

4 Creazione Database