

Università di Pisa

Corso di laurea in Ingegneria Informatica



Progetto per il corso Basi di dati

Anno Accademico 2014 – 2015

Prof. Gigliola Vaglini, Ing. Francesco Pistolesi

OGGETTO: Realizzazione di un database relazionale in grado di gestire i dati relativi al sistema informativo di una catena di ristoranti.

Studenti:

Rossi Federico

Suffredini Matteo

INDICE

1	<i>Cardinalità</i>
5	<i>Tavola dei volumi</i>
7	<i>Ridondanze e tavole degli accessi</i>
11	<i>Progettazione logica e normalizzazione</i>
13	<i>Schema ER</i>
14	<i>Specifica vincoli di integrità referenziale</i>
16	<i>Query</i>
18	<i>Area analytics</i>

CARDINALITA'

Associazione Stoccaggio

- Cardinalità (1,N) con Sede, in quanto ogni sede ha almeno un magazzino ma può averne anche di più
- Cardinalità (1,1) con Magazzino, in quanto un magazzino rifornisce una sola sede

Associazione Collocazione

- Cardinalità (0,N) con Magazzino, in quanto un magazzino contiene diverse confezioni (o nessuna confezione)
- Cardinalità (1,1) con Confezione, in quanto una confezione è presente in un solo magazzino

Associazione Contenuto

- Cardinalità (1,1) con Confezione, in quanto una confezione contiene un solo ingrediente
- Cardinalità (0,N) con Ingrediente, in quanto un ingrediente è contenuto in diverse confezioni (o nessuna confezione)

Associazione Allergia

- Cardinalità (1,N) Allergene, un'allergia può essere scatenata da diversi ingredienti
- Cardinalità (0,N) con Ingrediente, in quanto un ingrediente può provocare più allergie (o nessuna allergia)

Associazione Utilizzo

- Cardinalità (0,N) con Ingrediente, in quanto un ingrediente può partecipare a diversi passaggi (o nessun passaggio)
- Cardinalità (1,1) con Passaggio, in quanto ogni passo è associato ad una solo ingrediente

Associazione Procedimento

- Cardinalità (1,N) con Ricetta, in quanto una ricetta è composta da più passaggi
- Cardinalità (1,1) con Passaggio, in quanto ogni passo è associato ad una sola ricetta

Associazione Svolgimento

- Cardinalità (1,1) con Passaggio, in quanto ogni passo è associato ad una sola funzione
- Cardinalità (0,N) con Funzione, in quanto una funzione può essere usata in più passaggi (o nessun passaggio)

Associazione Necessità

- Cardinalità (1,N) con Funzione, in quanto ogni funzione può essere eseguita in sedi diversi da macchinari diversi
- Cardinalità (1,N) con Macchinario, in quanto un macchinario adempie a diverse funzioni

Associazione Proprietà

- Cardinalità (1,1) con Macchinario, in quanto un macchinario appartiene solo ad una sede
- Cardinalità (1,N) con Sede, in quanto una sede necessita di almeno un macchinario o un'attrezzatura per preparare le pietanze

Associazione Modifica

- Cardinalità (0,N) con Ingrediente, in quanto un ingrediente può essere in più variazioni (o nessuna variazione)
- Cardinalità (0,1) con VariazionePossibile, in quanto una variazione può modificare un ingrediente (o nessun ingrediente)

Associazione Cambia

- Cardinalità (0,N) con Funzione, in quanto una funzione può subire più variazioni (o nessuna variazione)
- Cardinalità (0,1) con VariazionePossibile, in quanto una variazione può modificare una funzione (o nessuna funzione)

Associazione Offerta

- Cardinalità (1,1) con Menu, in quanto ciascun Menu è specifico di una sola sede
- Cardinalità (1,N) con Sede, in quanto ciascuna sede può avere diversi menu in periodi differenti

Associazione Piatto

- Cardinalità (1,N) con Menu, in quanto il Menu è composto da più piatti
- Cardinalità (0,N) con Ricetta, in quanto una ricetta può essere inserita in più menu (o in nessun)

menu)

Associazione Richiesta

- Cardinalità (0,N) con Ricetta, in quanto una ricetta può essere presente in più ordini (o in nessun ordine)
- Cardinalità (1,1) con Ordine, in quanto un ordine è associato ad una sola ricetta

Associazione ValutaPiatto

- Cardinalità (0,N) con Ricetta, in quanto una ricetta può avere più recensioni (o nessuna recensione)
- Cardinalità (1,1) con RecensionePiatto, in quanto una recensione di un piatto è associata ad una ricetta

Associazione Coinvolge

- Cardinalità (1,1) con VariazionePossibile, in quanto una variazione modifica un determinato passaggio
- Cardinalità (0,N) con Passaggio, in quanto ogni passo è associato a più variazioni(o a nessuna variazione)

Associazione Servizio

- Cardinalità (0,N) con Tavolo, in quanto un tavolo è associato a più conti (o nessun conto)
- Cardinalità (1,1) con Comanda, in quanto un conto è relativo ad un solo tavolo

Associazione Posto

- Cardinalità (1,N) con Sede, in quanto una sede necessita di almeno un tavolo per svolgere il servizio
- Cardinalità (1,1) con Tavolo, in quanto un tavolo è posizionato in una determinata sede

Associazione Ordinazione

- Cardinalità (1,N) con Comanda, in quanto ad una comanda sono associati diversi ordini
- Cardinalità (0,1) con Ordine, in quanto un ordine può essere associato a una Comanda o a ComandaTakeAway

Associazione OrdinazioneTA

- Cardinalità (0,1) con Ordine in quanto un ordine può essere associato a una Comanda o a ComandaTakeAway
- Cardinalità (1,N) con ComandaTakeAway, in quanto una comanda prevede più piatti ordinati

Associazione EmissioneAsporto

- Cardinalità (1,1) con ComandaTakeAway, in quanto una comanda ha una sola fattura
- Cardinalità (0,1) con Fattura, in quanto una fattura può essere anche relativa a Comanda

Associazione EmissioneCoperto

- Cardinalità (1,1) con Comanda, in quanto una comanda ha una sola fattura
- Cardinalità (0,1) con Fattura, in quanto una fattura può essere anche relativa a ComandaTakeAway

Associazione Coperto

- Cardinalità (1,N) con Prenotazione, in quanto una prenotazione può coinvolgere più tavoli
- Cardinalità (0,N) con Tavolo, in quanto un tavolo è associato a più prenotazioni (o nessuna prenotazione)

Associazione Saldo

- Cardinalità (1,N) con Prenotazione, in quanto una prenotazione può coinvolgere più Comande
- Cardinalità (1,1) con Comanda, in quanto una Comanda è associata ad una Prenotazione

Associazione Trasporto

- Cardinalità (1,1) con ComandaTakeAway, in quanto un conto viene trasportato da un solo pony
- Cardinalità (0,N) con Pony, in quanto un fattorino trasporta diverse comande (o nessuna comanda)

Associazione Impiego

- Cardinalità (1,N) con Sede in quanto una sede ha più fattorini
- Cardinalità (1,1) con Pony, in quanto un fattorino appartiene ad una sola sede

Associazione TakeAway

- Cardinalità (1,N) con Sede in quanto una sede ha più ComandeTakeAway

- Cardinalità (1,1) con ComandaTakeAway, in quanto una Comanda è relativa ad una sola sede

Associazione PrenotazioneOnline

- Cardinalità (0,1) con Prenotazione, in quanto è possibile prenotare mediante un account (o telefonicamente)
- Cardinalità (0,N) con Account, in quanto un account può fare più prenotazioni (o nessuna prenotazione)

Associazione OrdineWEB

- Cardinalità (0,N) con Account, in quanto un account può prenotare più comande (o nessuna comanda)
- Cardinalità (1,1) con ComandaTakeAway, in quanto si può prenotare solo mediante account

Associazione Variazione

- Cardinalità (0,N) con VariazionePossibile, in quanto una variazione può essere associata a diversi ordini (o nessun ordine)
- Cardinalità (0,N) con Ordine, in quanto un ordine può avere fino a 3 variazioni (o nessuna variazione)

Associazione Specifica

- Cardinalità (0,1) con Prenotazione, in quanto una prenotazione può prevedere una serata a tema (o nessuna serata a tema)
- Cardinalità (0,N) con SerataTema, in quanto una serata a tema può essere richiesta in più prenotazioni (o nessuna prenotazione)

Associazione PropostaSerata

- Cardinalità (1,1) con SerataTema, in quanto una serata a tema può essere proposta solamente da un account
- Cardinalità (0,N) con Account, in quanto un account può proporre più serate a tema (o nessuna serata a tema)

Associazione ValutazionePiatto

- Cardinalità (0,N) con Recensione, in quanto una Recensione può recensire più piatti (o nessun piatto)
- Cardinalità (1,1) con RecensionePiatto, in quanto una RecensionePiatto appartiene ad una Recensione

Associazione PropostaPiatto

- Cardinalità (0,N) con Account, in quanto un account può proporre più piatti (o nessun piatto)
- Cardinalità (1,1) con PiattoProposto, in quanto un piatto può essere proposto solamente da un account

Associazione Valutazione

- Cardinalità (0,N) con Account, in quanto un account può recensire più recensioni (o nessuna recensione)
- Cardinalità (1,1) con ValutazioneRecensione, in quanto ogni valutazione è associata ad un account

Associazione Veridicità

- Cardinalità (0,N) con RecensionePiatto, in quanto una recensione del piatto può essere recensita più volte (o nessuna)
- Cardinalità (0,1) con ValutazioneRecensione, in quanto può essere recensita anche una RecSede

Associazione Opinione

- Cardinalità (0,N) con Account, in quanto un account può recensire più sedi (o nessuna sede)
- Cardinalità (1,1) con Recensione, in quanto una Recensione è fatta solo da un account

Associazione Contiene

- Cardinalità (0,N) con Recensione, in quanto ogni recensione è associata a più risposte (o nessuna risposta)
- Cardinalità (1,1) con RispostaUtente, in quanto ogni risposta data è associata ad una sola Recensione

Associazione Espressione

- Cardinalità (1,1) con RispostaUtente, in quanto ogni risposta data è associata ad un'unica

risposta possibile

- Cardinalità (0,N) con Risposte, in quanto ogni risposta è presente più volte in RisposteUtente (o nessuna volta)

Associazione Relativa

- Cardinalità (1,1) con Domande, in quanto ogni domanda è riferita ad un questionario
- Cardinalità (1,N) con Questionario, in quanto un questionario presenta più domande

Associazione Possibilità

- Cardinalità (1,1) con Risposte, in quanto ogni risposta è associata ad un'unica domanda
- Cardinalità (1,N) con Domande, in quanto ogni domanda ha più risposte

Associazione Sondaggio

- Cardinalità (1,1) con Questionario, in quanto un questionario è associato ad una sola sede
- Cardinalità (0,N) con Sede, in quanto un Direttore può scegliere se aggiungere il questionario, possono esserci più questionari

Associazione RecensioneSede

- Cardinalità (0,N) con Sede, in quanto una sede può ricevere più recensioni (o nessuna recensione)
- Cardinalità (1,1) con Recensione, in quanto la recensione si riferisce ad una sola sede

Associazione GradimentoPiatto

- Cardinalità (0,N) con PiattoProposto, in quanto un' account può esprimere più gradimento (o nessuno)
- Cardinalità (0,N) con Account, in quanto più account possono esprimere gradimento relativamente ad un piatto (o nessuno)

Associazione GradimentoVariazione

- Cardinalità (0,N) con VariazionePossibile, in quanto un' account può esprimere più gradimento (o nessuno)
- Cardinalità (0,N) con Account, in quanto più account possono esprimere gradimento relativamente ad un piatto (o nessuno)

Associazione Proposta

- Cardinalità (0,1) con VariazionePossibile, in quanto una variazione può essere proposta da un account (o dallo chef)
- Cardinalità (0,N) con Account, in quanto più account propongono variazioni (o nessuno)

TAVOLA DEI VOLUMI

Nome	Tipo E/R	Volume	Motivazione
Allergia	R	120	2 Allergeni per ogni Ingrediente
Account	E	50	Ipotesi Iniziale
Allergene	E	40	Ipotesi Iniziale
Cambia	R	3000	3/5 di Variazione Possibile (0,1)
Coinvolge	R	5000	(1,1) Con Variazione Possibile
Collocazione	R	3000	(1,1) Con Confezione
Comanda	E	2000	2 Comande per Prenotazione
ComandaTA	E	1000	20 Conti per Account
Confezione	E	3000	50 Confezioni per Ingrediente
Contenuto	R	3000	(1,1) Con Confezione
Contiene	R	6000	(1,1) con RispostaUtente
Coperto	R	2000	2 Tavolo in media per Prenotazione
Domanda	E	1200	20 Domande per Questionario
EmissioneAsporto	R	1000	(1,1) con ComandaTakeAway
EmissioneCoperto	R	2000	(1,1) Con Comanda
Espressione	R	6000	(1,1) Con RispostaUtente
Fattura	E	2000	1 Fattura per ComandaTA + 1 Fattura ogni 2 Comande
Funzione	E	30	Ipotesi Iniziale
Gradimento P.	R	500	10 Gradimenti espressi per Account
Gradimento V.	R	500	10 Gradimenti espressi per Account
Impiego	R	400	(1,1) Con Pony
Ingrediente	E	60	Ipotesi Iniziale
Macchinario	E	600	30 Macchinari per Sede
Magazzino	E	100	5 magazzini per sede
Menu	E	60	3 Menu per Sede
Modifica	R	4000	4/5 di Variazione Possibile (0,1)
Necessita	R	20	1 Macchinario per la stessa Funzione in ogni Sede
Offerta	R	60	(1,1) Con Menu
Opinione	R	300	(1,1) Con Recensione
Ordinazione	R	12000	6 Ordine per Comanda
OrdinazioneTA	R	12000	12 Ordine per ComandaTakeAway
Ordine	E	24000	24*1000 Ordini = (6*2 Ordini per Comanda + 12 Ordini ComandaTA)*1000
OrdineWEB	R	1000	(1,1) ComandaTakeAway
Passaggio	E	5000	5 passaggi per ricetta
Piatto	R	1000	(1,1) Con Ricetta
PiattoProposto	E	200	4 Proposte per Account
Pony	E	400	20 Pony per Sede
Possibilità	R	6000	(1,1) Con Risposta
Posto	R	400	(1,1) Con Tavolo
Prenot. Online	R	500	½ prenotazioni è online
Prenotazione	E	1000	50 prenotazioni per sede
Procedimento	R	5000	(1,1) Con Passaggio
Proposta	R	1000	1/5 VariazionePossibile proposta da Utenti

PropostaPiatto	R	200	(1,1) Con PiattoProposto
PropostaSerata	R	50	(1,1) SerataTema
Proprietà	R	60	(1,1) Con Macchinario
Questionario	E	60	3 Questionari per Sede
Recensione	E	300	6 Recensioni per Account
RecensionePiat.	E	30000	30 RecensionePiatto per Ricetta
RecensioneSede	R	300	(1,1) Con Recensione
Relatività	R	1200	(1,1) Con Domanda
Ricetta	E	1000	Ipotesi Iniziale
Richiesta	R	24000	(1,1) Con Ordine
Risposta	E	6000	5 Risposte per Domanda
RispostaUtente	E	6000	20 Risposte per Recensione
Saldo	R	1000	(1,1) con ComandaTakeAway
Sede	E	20	Ipotesi Iniziale
SerataTema	E	50	1 Proposta per Account
Servizio	R	2000	(1,1) Con Comanda
Sondaggio	R	60	(1,1) Con Questionario
Specifica	R	100	1/10 Prenotazioni sono SerataTema
Stoccaggio	R	100	(1,1) Con Magazzino
Svolgimento	R	5000	(1,1) Con Passaggio
TakeAway	R	1000	(1,1) ComandaTakeAway
Tavolo	E	400	20 Tavoli per sede
Utilizzo	R	5000	(1,1) Con Passaggio
Valutaz.Piatto	R	30000	(1,1) Con RecensionePiatto
Valutazione	R	600	(1,1) Con ValutazioneRecensione
ValutazioneRec.	E	600	2 Valutazioni per Recensione
Variazione	R	48000	2 Variazione per Ordine
VariazionePoss.	E	5000	5 variazioni per ricetta
Veridicità	R	600	(1,1) Con ValutazioneRecensione

RIDONDANZE E TAVOLE DEGLI ACCESSI

Operazioni proposte

1. Trovare la ricetta con il più alto punteggio medio nelle valutazioni.
 - Frequenza: 20 volte al giorno.
2. Inserimento di una recensione su una ricetta.
 - Frequenza: 10 volte al giorno.
3. Trovare l'importo totale delle fatture emesse, relativamente alle sole prenotazioni online
 - Frequenza: 1 volta al giorno.
4. Inserimento di un nuovo ordine in una comanda (NON takeaway)
 - Frequenza: 1000 volte al giorno.
5. Trovare per ogni ricetta la variazione più frequentemente ordinata.
 - Frequenza: 2 volte al mese.
6. Per ogni magazzino, trovare le confezioni contenenti materie prime in scadenza. Ossia con data di scadenza entro tre giorni dalla data odierna.
 - Frequenza: 5 volte al giorno
7. Per ogni ricetta trovare il rapporto guadagno/spesa
 - Frequenza: 1 volta al mese
8. Eliminare, per ogni menu, la ricetta (supposta unica) che, relativamente all'ultimo mese, è stata ordinata per meno del 5% degli ordini totali.
 - Frequenza: 1 volta al mese

Ridondanze proposte

1. Attributo "Importo" dell'entità "Fattura" che mantiene l'informazione dell'importo totale di una fattura emessa costituisce una ridondanza.
2. Attributi "NumeroValutazioni", "SommaValutazioni" sull'entità Ricetta che mantengono il numero di valutazioni sul piatto e la loro somma al fine di stimare una media delle valutazioni è una ridondanza. Infatti l'informazione è reperibile contando, attraverso l'associazione "ValutaPiatto" le relative recensioni e sommandone l'attributo "Punteggio".

Tavole degli accessi

Operazione 1 (Tra parentesi per singola ricetta)

Con Ridondanza:

1000 (1)	L	Su Ricetta per recuperare gli attributi NumeroRecensioni e SommaRecensioni e trovarne il massimo
1000 (1)	Operazione Elementare	*
20	Operazioni al giorno	=
20000 (20)	Operazioni Elementari al giorno	

Senza Ridondanza:

30000 (30)	L	Su ValutazionePiatto per recuperare i codici delle recensioni di ogni ricetta
30000 (30)	L	Su RecensionePiatto per recuperare la media delle recensioni di ogni ricetta e trovarne il massimo
60000 (60)	Operazioni Elementari	*
20	Operazioni al giorno	=
1200000 (1200)	Opreazioni elementari al giorno	

Operazione 2

Con Ridondanza:

1*2	S	Su RecensionePiatto per inserire una recensione
1*2	S	Su ValutaPiatto per associare la ricetta alla recensione
1*2	S	Su Ricetta per aggiornare gli attributi ridondanti
6	Operazioni Elementari	*
10	Operazioni al giorno	=
60	Operazioni elementari al giorno	

Senza Ridondanza:

1*2	S	Su RecensionePiatto per inserire una recensione
1*2	S	Su ValutaPiatto per associare la ricetta alla recensione
4	Operazioni Elementari	*
10	Operazioni al giorno	=
40	Operazioni Elementari al giorno	

Operazione 3 (tra parentesi per singolo account)

Con Ridondanza:

500 (10)	L	Su PrenotazioneOnline per recuperare il codice delle prenotazioni
1000 (20)	L	Da Saldo per recuperare il codice delle comande effettuate
1000 (20)	L	Da Emissione Coperto per recuperare il codice delle fatture
1000 (20)	L	Da Fattura per recuperarne l'importo
3500 (70)	Operazioni Elementari	*
1	Operazione al giorno	=
3500 (70)	Operazioni Elementari al giorno	

Senza Ridondanza:

500 (10)	L	Su PrenotazioneOnline per recuperare il codice delle prenotazioni
1000 (20)	L	Su Saldo per recuperare il codice delle

		comande effettuate
6000 (120)	L	Su Ordinazione per recuperare il codice degli ordini
6000 (120)	L	Su Richiesta per recuperare il codice delle ricette
6000 (120)	L	Da Ricetta per sommare il costo di ogni ordinazione
19500 (390)	Operazioni Elementari	*
1	Operazione al giorno	=
19500 (390)	Operazioni Elementari al giorno	

Operazione 4

Con Ridondanza

1*2	S	Su Ordine per inserire la nuova ordinazione
1*2	S	Su Richiesta per associare il codice della Ricetta all'Ordine
1	L	Su Ricetta per recuperare il prezzo del piatto
1*2	S	Su Ordinazione per associare il codice della Comanda all'ordine
1	L	Su EmissioneCoperto per recuperare il codice della Fattura
1*2	S	Su Fattura per aggiornare l'attributo Importo
10	Operazioni Elementari	*
1000	Operazioni al giorno	=
10000	Operazioni Elementari al giorno	

Senza Ridondanza:

1*2	S	Su Ordine per inserire l' ordinazione
1*2	S	Su Richiesta per associare il codice della Ricetta all'Ordine
1*2	S	Su Ordinazione per associare il codice della Comanda all'ordine
6	Operazioni Elementari	*
1000	Operazioni al giorno	=
6000	Operazioni Elementari al giorno	

Operazione 5 (Non coinvolge ridondanze) (Tra parentesi per singola ricetta)

24000 (24)	L	Su Richiesta per trovare gli ordini di ogni ricetta
48000 (48)	L	Su Variazione per ricavare, per ogni ricetta, il numero di ordinazioni di ciascuna variazione
72000 (72)	Operazioni Elementari	*
2	Operazioni al mese	=
144000 (144)	Operazioni Elementari al mese	

Operazione 6 (Non coinvolge ridondanze) (Tra parentesi per singolo magazzino)

3000 (30)	L	Su Collocazione per ricavare i codici delle confezioni
3000 (30)	L	Su Confezione per recuperare la data di scadenza
6000 (60)	Operazioni Elementari	*
20	Operazioni al mese	=
120000 (1200)	Operazioni Elementari al mese	

Operazione 7 (Non coinvolge ridondanze) (Tra parentesi per singola ricetta)

5000 (5)	L	Su Procedimento per recuperare i passaggi di ogni ricetta
5000 (5)	L	Su Utilizzo per ricavare l'ingrediente di ciascun passaggio
5000 (5)	L	Su Contenuto per ricavare il costo di ogni confezione, e quindi, proporzionalmente, di ogni ricetta
24000 (24)	L	Su ordinazione per trovare il numero di ordinazioni di ogni ricetta e di conseguenza, l'incasso relativo.
39000 (39)	Operazioni Elementari	*
1	Operazione al mese	=
39000 (39)	Operazioni elementari al mese	

Operazione 8 (Non coinvolge ridondanze) (Tra parentesi per singolo menù)

1000 (~17)	L	Su Piatto per recuperare le ricette in ogni menu
24000 (400)	L	Su Richiesta per trovare il numero di ordinazioni per ciascuna ricetta
17*2 (1)	S	Su Piatto per eliminare dal menu la ricetta ordinata per meno del 5% degli ordini totali
25034 (~418)	Operazioni Elementari	*
1	Operazione al mese	=
25034 (~ 418)	Operazioni Elementari al mese	

Ridondanza 1:

- Con ridondanza: 13500 Operazioni Elementari al giorno
- Senza ridondanza:24500 Operazioni elementari al giorno

E' conveniente mantenere la ridondanza per diminuire il numero di operazioni elementari giornaliere.

Ridondanza 2:

- Con ridondanza: 20060 Operazioni elementari al giorno
- Senza ridondanza:1200040 Operazioni elementari al giorno

E' conveniente mantenere la ridondanza per diminuire il numero di operazioni elementari giornaliere.

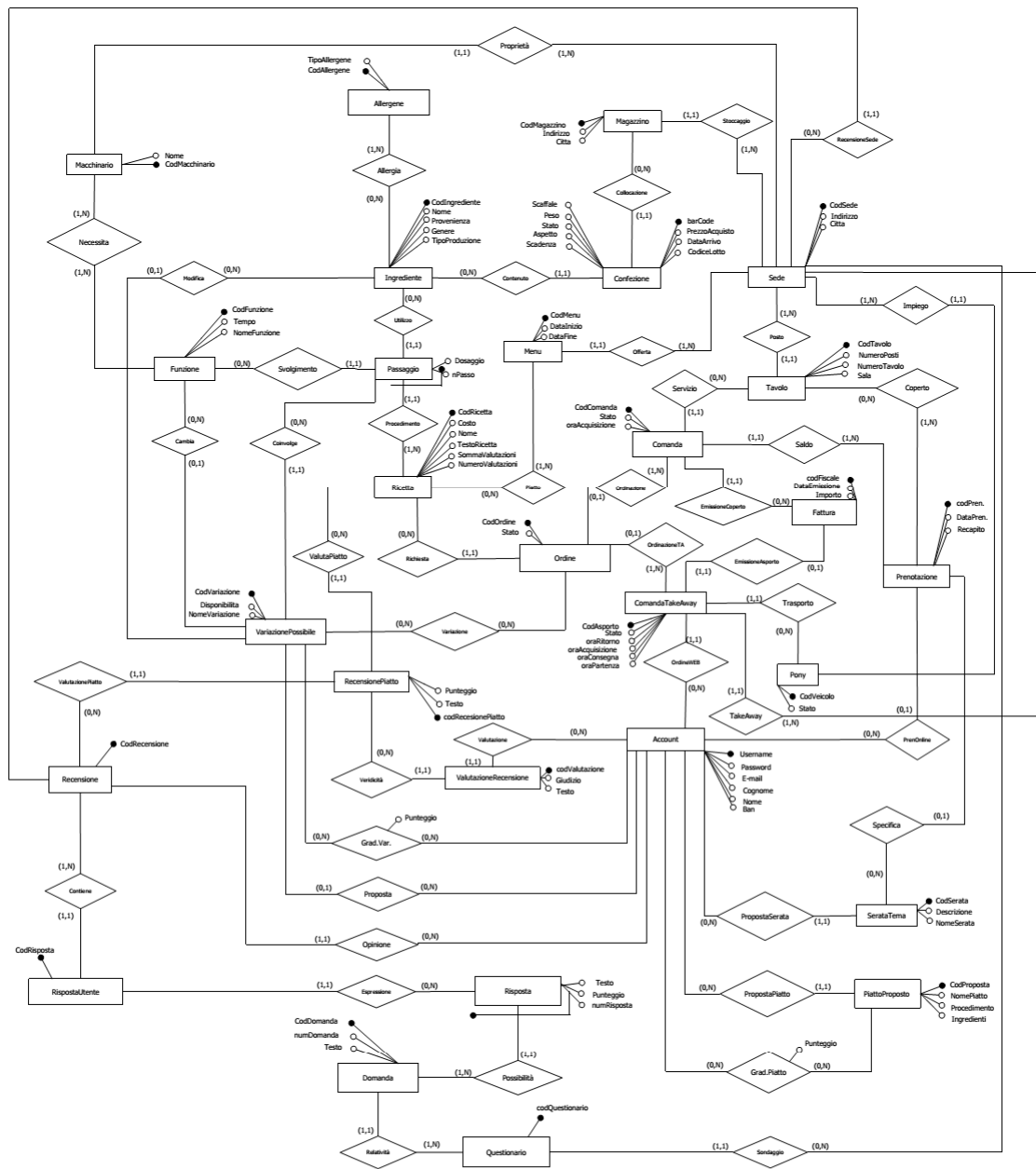
PROGETTAZIONE LOGICA E NORMALIZZAZIONE

- **Sede** (CodSede, Citta, Indirizzo)
CodSede->NomeSede, Citta, Indirizzo, SitoWeb, eMail, Telefono;
- **Menu** (CodMenu, CodSede, DataInizio, DataFine)
CodMenu->CodSede,DataInizio,DataFine;
- **Tavolo** (CodTavolo, NumTavolo, CodSede, Sala, NumPosti)
CodTavolo->NumTavolo, CodSede, Sala, NumPosti
- **Magazzino** (CodMagazzino, Citta, Indirizzo, CodSede)
CodMagazzino->Citta, Indirizzo, CodSede
- **Confezione** (CodConfezione,CodIngrediente,CodMagazzino, Scaffale, CodLotto, Peso, Stato, Danneggiato, Scadenza, PrezzoAcquisto, DataArrivo)
CodConfezione->CodIngrediente,CodMagazzino, Scaffale, CodLotto, Peso, Stato, Danneggiato, Scadenza, PrezzoAcquisto, DataArrivo
- **Ingrediente** (CodIngrediente, NomeIngrediente, Provenienza, Genere, TipoProduzione)
CodIngrediente->NomeIngrediente, Provenienza, Genere, TipoProduzione
- **Ricetta**(CodRicetta, NomeRicetta, Costo, TestoRicetta, NumeroRecensioni, SommaRecensioni)
CodRicetta->NomeRicetta, Costo, TestoRicetta, NumeroRecensioni, SommaRecensioni
- **Piatto** (CodRicetta, CodMenu)
Nessuna dipendenza, (CodRicetta,CodMenu) è superchiave minimale della relazione.
- **Macchinario** (CodMacchinario,NomeMacchinario, CodSede)
CodMacchinario->NomeMacchinario, CodSede
- **Funzione** (CodFunzione,NomeFunzione, Tempo)
CodFunzione->NomeFunzione, Tempo, CodMacchinario
- **Necessita** (CodFunzione,CodMacchinario)
Nessuna dipendenza, (CodFunzione,CodMacchinario) è superchiave minimale della relazione.
- **Passaggio** (CodRicetta, NumeroPassaggio, CodIngrediente, Funzione,Dose)
CodRicetta,NumeroPassaggio -> CodIngrediente,CodFunzione,Dose.
- **Allergene** (CodAllergene, TipoAllergene)
CodAllergene->TipoAllergene
- **Allergia** (CodIngrediente, CodAllergene)
Nessuna dipendenza, (CodIngrediente,CodAllergene) è superchiave minimale della relazione
- **Account** (Username, Nome, Cognome, eMail, Password,Ban)
Username->Nome, Cognome, eMail, Password,Ban
- **Prenotazione** (CodPrenotazione,CodSerata,Recapito, Username, DataPrenotazione)
CodPrenotazione->CodSerata,Recapito, Username, DataPrenotazione
- **Comanda**(CodComanda, Stato, OraAcquisizione, CodTavolo, CodPrenotazione)
CodComanda->Stato, OraAcquisizione, CodTavolo, CodPrenotazione
- **ComandaTakeAway** (CodAsporto, Stato, OraRitorno, OraAcquisizione, OraConsegna, OraPartenza, CodVeicolo, Username,CodSede)
CodAsporto->Stato, OraRitorno, OraAcquisizione, OraConsegna, OraPartenza, CodVeicolo, Username
- **Pony** (CodVeicolo, Stato, CodSede)

CodVeicolo->Stato, CodSede

- **Fattura** (CodFattura, DataEmissione, CodComanda, CodAsporto, Importo)
CodFattura->DataEmissione, CodComanda, CodAsporto, Importo
- **Coperto** (CodPrenotazione, CodTavolo, NumeroCoperti)
CodPrenotazione->CodTavolo, NumeroCoperti
- **Ordine** (CodOrdine, CodRicetta, CodAsporto, CodComanda, Stato)
CodOrdine->CodRicetta, CodAsporto, CodComanda, Stato
- **VariazionePossibile** (CodVariazione, NomeVariazione, Funzione, CodIngrediente, NumeroPassaggio, CodRicetta, Disponibile, Username)
CodVariazione->NomeVariazione, Funzione, CodIngrediente, NumeroPassaggio, CodRicetta, Disponibile, Username
- **GradimentoVariazione**(CodVariazione, Username, Punteggio)
CodVariazione, Username->Punteggio
- **Variazione** (CodVariazione, CodOrdine)
Nessuna dipendenza, (CodVariazione, CodOrdine) è superchiave minimale della relazione.
- **RecensionePiatto**(CodRecensionePiatto, CodRecensione, CodRicetta, Testo, Punteggio)
CodRecensionePiatto->CodRecensione, CodRicetta, Testo, Punteggio
- **Questionario** (CodQuestionario, CodSede)
CodQuestionario->CodSede
- **Domanda** (CodDomanda, CodQuestionario, Testo)
CodDomanda, CodQuestionario->Testo
- **Risposta** (NumRisposta, CodDomanda, Punteggio, Testo)
NumRisposta, CodDomanda ->Punteggio, Testo
- **Recensione**(CodRecensione, Username, CodSede, ValutazioneSede, PunteggioSede)
CodRecensione->Username, CodSede, ValutazioneSede, PunteggioSede
- **ValutazioneRecensione** (CodValutazione, Testo, Punteggio, CodRecensionePiatto)
CodValutazione->Testo, Punteggio, CodRecensionePiatto
- **RispostaUtente** (CodRecensione, CodDomanda, NumRisposta)
Nessuna dipendenza funzionale, (CodRecensione, CodDomanda, NumRisposta) è superchiave minimale della relazione
- **SerataTema** (CodSerata, NomeSerata, Descrizione, Username, CodPrenotazione)
CodSerata->NomeSerata, Descrizione, Username, CodPrenotazione
- **PropostaPiatto** (CodProposta, NomePiatto, Procedimento, Ingredienti, Username)
CodProposta->NomePiatto, Procedimento, Ingredienti, Username
- **GradimentoProposta**(CodProposta, Username, Punteggio)
CodProposta->Username, Punteggio

SCHEMA ER



(In fondo al documento è presente una stampa formato A3 dello stesso)

SPECIFICA VINCOLI DI INTEGRITA' REFERENZIALE

- Esiste un vincolo di integrità referenziale tra l'attributo CodSede della tabella MENU e l'attributo CodSede della tabella SEDE
- Esiste un vincolo di integrità referenziale tra l'attributo CodSede della tabella TAVOLO e l'attributo CodSede della tabella SEDE
- Esiste un vincolo di integrità referenziale tra l'attributo CodSede della tabella MAGAZZINO e l'attributo CodSede della tabella SEDE
- Esiste un vincolo di integrità referenziale tra l'attributo CodMagazzino della tabella CONFEZIONE e l'attributo CodMagazzino della tabella MAGAZZINO
- Esiste un vincolo di integrità referenziale tra l'attributo CodIngrediente della tabella INGREDIENTE e l'attributo CodIngrediente della tabella INGREDIENTE
- Esiste un vincolo di integrità referenziale tra l'attributo CodSede della tabella MACCHINARIO e l'attributo CodSede della tabella SEDE
- Esiste un vincolo di integrità referenziale tra l'attributo CodMacchinario della tabella FUNZIONE e l'attributo CodMacchinario della tabella MACCHINARIO
- Esiste un vincolo di integrità referenziale tra l'attributo CodRicetta della tabella PASSAGGIO e l'attributo CodRicetta della tabella RICETTA
- Esiste un vincolo di integrità referenziale tra l'attributo CodIngrediente della tabella PASSAGGIO e l'attributo CodIngrediente della tabella INGREDIENTE
- Esiste un vincolo di integrità referenziale tra l'attributo CodIngrediente della tabella ALLERGIA e l'attributo CodIngrediente della tabella INGREDIENTE
- Esiste un vincolo di integrità referenziale tra l'attributo CodAllergene della tabella ALLERGIA e l'attributo CodAllergene della tabella ALLERGENE
- Esiste un vincolo di integrità referenziale tra l'attributo Username della tabella PRENOTAZIONE e l'attributo Username della tabella ACCOUNT
- Esiste un vincolo di integrità referenziale tra l'attributo CodSerata della tabella PRENOTAZIONE e l'attributo CodSerata della tabella SERATATEMA
- Esiste un vincolo di integrità referenziale tra l'attributo CodTavolo della tabella COMANDA e l'attributo CodTavolo della tabella TAVOLO
- Esiste un vincolo di integrità referenziale tra l'attributo CodPrenotazione della tabella COMANDA e l'attributo CodPrenotazione della tabella PRENOTAZIONE
- Esiste un vincolo di integrità referenziale tra l'attributo CodFattura della tabella COMANDA e l'attributo CodFattura della tabella FATTURA
- Esiste un vincolo di integrità referenziale tra l'attributo Username della tabella COMANDATAKEAWAY e l'attributo Username della tabella ACCOUNT
- Esiste un vincolo di integrità referenziale tra l'attributo Sede della tabella COMANDATAKEAWAY e l'attributo Sede della tabella SEDE
- Esiste un vincolo di integrità referenziale tra l'attributo CodVeicolo della tabella COMANDATAKEAWAY e l'attributo CodVeicolo della tabella PONY
- Esiste un vincolo di integrità referenziale tra l'attributo CodFattura della tabella COMANDATAKEAWAY e l'attributo CodFattura della tabella FATTURA
- Esiste un vincolo di integrità referenziale tra l'attributo della tabella COPERTO e l'attributo CodPrenotazione della tabella PRENOTAZIONE
- Esiste un vincolo di integrità referenziale tra l'attributo CodTavolo della tabella COPERTO e l'attributo CodTavolo della tabella TAVOLO
- Esiste un vincolo di integrità referenziale tra l'attributo CodRicetta della tabella ORDINE e l'attributo CodRicetta della tabella RICETTA
- Esiste un vincolo di integrità referenziale tra l'attributo CodComanda della tabella ORDINE e l'attributo CodConto della tabella COMANDA
- Esiste un vincolo di integrità referenziale tra l'attributo CodAsporto della tabella ORDINE e l'attributo CodAsporto della tabella COMANDATAKEAWAY
- Esiste un vincolo di integrità referenziale tra l'attributo CodMenu della tabella PIATTO e l'attributo CodMenu della tabella MENU

- Esiste un vincolo di integrità referenziale tra l'attributo CodRicetta della tabella PIATTO e l'attributo CodRicetta della tabella RICETTA
- Esiste un vincolo di integrità referenziale tra l'attributo Funzione della tabella VARIAZIONEPOSSIBILE e l'attributo Funzione della tabella FUNZIONE
- Esiste un vincolo di integrità referenziale tra l'attributo CodIngrediente della tabella VARIAZIONEPOSSIBILE e l'attributo CodIngrediente della tabella INGREDIENTE
- Esiste un vincolo di integrità referenziale tra l'attributo CodRicetta della tabella VARIAZIONEPOSSIBILE e l'attributo CodRicetta della tabella RICETTA
- Esiste un vincolo di integrità referenziale tra l'attributo CodVariazione della tabella VARIAZIONE e l'attributo CodVariazione della tabella VARIAZIONEPOSSIBILE
- Esiste un vincolo di integrità referenziale tra l'attributo CodOrdine della tabella VARIAZIONE e l'attributo CodOrdine della tabella ORDINE
- Esiste un vincolo di integrità referenziale tra l'attributo CodRicetta della tabella RECENSIONEPIATTO e l'attributo CodRicetta della tabella RICETTA
- Esiste un vincolo di integrità referenziale tra l'attributo CodRecensione della tabella RECENSIONEPIATTO e l'attributo CodRecensione della tabella RECENSIONE
- Esiste un vincolo di integrità referenziale tra l'attributo CodSede della tabella QUESTIONARIO e l'attributo CodSede della tabella SEDE
- Esiste un vincolo di integrità referenziale tra l'attributo CodQuestionario della tabella DOMANDA e l'attributo CodQuestionario della tabella QUESTIONARIO
- Esiste un vincolo di integrità referenziale tra l'attributo CodDomanda della tabella RISPOSTA e l'attributo CodDomanda della tabella DOMANDA
- Esiste un vincolo di integrità referenziale tra l'attributo Username della tabella RECENSIONE e l'attributo Username della tabella ACCOUNT
- Esiste un vincolo di integrità referenziale tra l'attributo CodSede della tabella RECENSIONE e l'attributo CodSede della tabella SEDE
- Esiste un vincolo di integrità referenziale tra l'attributo Username della tabella VALUTAZIONERECENSIONE e l'attributo Username della tabella ACCOUNT
- Esiste un vincolo di integrità referenziale tra l'attributo CodRecensionePiatto della tabella VALUTAZIONERECENSIONE e l'attributo CodRecensionePiatto della tabella RECENSIONEPIATTO
- Esiste un vincolo di integrità referenziale tra l'attributo CodDomanda della tabella RISPOSTEUTENTE e l'attributo CodDomanda della tabella DOMANDA
- Esiste un vincolo di integrità referenziale tra gli attributi (CodDomanda, NumRisposta) della tabella RISPOSTEUTENTE e attributi (CodDomanda, NumRisposta) della tabella RISPOSTA
- Esiste un vincolo di integrità referenziale tra l'attributo CodRecensione della tabella RISPOSTAUTENTE e l'attributo CodRecensione della tabella RECENSIONE
- Esiste un vincolo di integrità referenziale tra l'attributo Esiste un vincolo di integrità referenziale tra l'attributo CodPrenotazione della tabella SERATATEMA e l'attributo CodPrenotazione della tabella PRENOTAZIONE
- Esiste un vincolo di integrità referenziale tra l'attributo Username della tabella SERATATEMA e l'attributo Username della tabella ACCOUNT
- Esiste un vincolo di integrità referenziale tra l'attributo Username della tabella PROPOSTAPIATTO e l'attributo Username della tabella ACCOUNT
- Esiste un vincolo di integrità referenziale tra l'attributo CodProposta della tabella GRADIMENTOPROPOSTA e l'attributo CodProposta della tabella PROPOSTAPIATTO
- Esiste un vincolo di integrità referenziale tra l'attributo Username della tabella GRADIMENTOPROPOSTA e l'attributo Username della tabella ACCOUNT

QUERY

Trovare la ricetta con il più alto punteggio medio nelle valutazioni

```
SELECT CodRicetta, NomeRicetta
FROM Ricetta
WHERE NumeroRecensioni > 0
      AND SommaRecensioni/NumeroRecensioni = (
          SELECT MAX(R.SommaRecensioni/R.NumeroRecensioni)
          FROM Ricetta R
          WHERE R.NumeroRecensioni > 0);
```

Inserimento di una recensione su una ricetta

```
INSERT INTO RecensionePiatto (Testo, Punteggio, CodRicetta, CodRecensione)
VALUES ("TestoRecensione", 5, 1, 1);
```

Trovare l'importo totale delle fatture emesse, relativamente alle sole prenotazioni online

```
SELECT SUM(F.Importo)
FROM Fattura F NATURAL JOIN Comanda C
      NATURAL JOIN Prenotazione P
      NATURAL JOIN Account A;
```

Inserimento di un nuovo ordine in una comanda (NON takeaway)

```
INSERT INTO Ordine (CodRicetta, CodComanda) VALUES (1, 1);
```

Trovare per ogni ricetta la variazione più frequentemente ordinata.

```
CREATE OR REPLACE VIEW OrdinazioniVariazioni AS
SELECT R.CodRicetta, V.CodVariazione, COUNT(*) AS Numero
FROM Ricetta R NATURAL JOIN Ordine O
      NATURAL JOIN Variazione V
GROUP BY R.CodRicetta, V.CodVariazione;
```

```
CREATE OR REPLACE VIEW MassimiOrdinazioni AS
SELECT CodRicetta, MAX(Numero) AS Numero
FROM OrdinazioniVariazioni
GROUP BY CodRicetta;
```

```
SELECT CodRicetta, CodVariazione
FROM OrdinazioniVariazioni NATURAL JOIN MassimiOrdinazioni;
```

Per ogni magazzino, trovare le confezioni contenenti materie prime in scadenza. Ossia con data di scadenza entro tre giorni dalla data odierna.

```
SELECT M.CodMagazzino, C.CodConfezione
FROM Magazzino M NATURAL JOIN Confezione C
WHERE DATEDIFF(C.Scadenza, CURRENT_DATE()) <= 3;
```

Per ogni ricetta trovare il rapporto guadagno/spesa

```
CREATE OR REPLACE VIEW GuadagnoRicetta AS
SELECT R.CodRicetta, SUM(R.Costo) AS Guadagno
```

```

FROM Ricetta R NATURAL JOIN Ordine O
GROUP BY R.CodRicetta;

CREATE OR REPLACE VIEW IngredientiRicetta AS
SELECT DISTINCT R.CodRicetta,I.CodIngrediente,P.Dose,C.PrezzoAcquisto
FROM Ricetta R INNER JOIN Passaggio P USING(CodRicetta)
INNER JOIN Ingrediente I USING(CodIngrediente)
INNER JOIN Confezione C USING(CodIngrediente);

CREATE OR REPLACE VIEW SpesaRicetta AS
SELECT R.CodRicetta,SUM(R.Dose*R.PrezzoAcquisto) AS Spesa
FROM IngredientiRicetta R INNER JOIN Ordine O USING(CodRicetta)
GROUP BY R.CodRicetta;

SELECT G.CodRicetta,G.Guadagno/S.Spesa
FROM GuadagnoRicetta G NATURAL JOIN SpesaRicetta S;

```

Eliminare, per ogni menu, la ricetta (supposta unica) che, relativamente all'ultimo mese, è stata ordinata per meno del 5% degli ordini totali

```

CREATE OR REPLACE VIEW OrdinazioniPiattoMenu AS
SELECT R.CodRicetta,P.CodMenu,COUNT(*) AS NumeroOrdinazioni
FROM Ricetta R NATURAL JOIN Ordine O
NATURAL JOIN Piatto P
LEFT OUTER JOIN Comanda C USING(CodComanda)
LEFT OUTER JOIN ComandaTakeAway CT USING(CodAsporto)
WHERE (C.CodComanda IS NOT NULL
AND MONTH(C.OraAcquisizione) >MONTH(CURRENT_DATE() -1)
OR (CT.CodAsporto IS NOT NULL
AND MONTH(CT.OraAcquisizione) >MONTH(CURRENT_DATE() -1)
GROUP BY R.CodRicetta,P.CodMenu;

CREATE OR REPLACE VIEW TotaleOrdinazioniMenu AS
SELECT CodMenu,SUM(NumeroOrdinazioni) as TotaleMenu
FROM OrdinazioniPiattoMenu
GROUP BY CodMenu;

CREATE OR REPLACE VIEW Target AS
SELECT CodRicetta,CodMenu,IF(NumeroOrdinazioni IS NULL, 0,
NumeroOrdinazioni),IF(TotaleMenu IS NULL, 0, TotaleMenu)
FROM Piatto LEFT OUTER JOIN (OrdinazioniPiattoMenu
INNER JOIN TotaleOrdinazioniMenu
USING(CodMenu)) USING(CodRicetta,CodMenu)
WHERE NumeroOrdinazioni <0.05*TotaleMenu
ORDER BY CodRicetta,CodMenu;

DELETE P.* FROM Piatto P
WHERE (P.CodRicetta,P.CodMenu) IN (
SELECT * FROM (SELECT * FROM Target) AS D);

```

AREA ANALYTICS

Magazzino intelligente

L'area analytics deve permettere di gestire in maniera intelligente il magazzino.

In particolare, deve essere realizzata una funzionalità di back-end che suggerisce le ricette da inserire in menu, relativamente allo stato attuale del magazzino. Tali ricette devono ottimizzare, cioè, gestire al meglio, i tempi di scadenza, la freschezza e lo stato degli ingredienti (specialmente lo stato parziale, poiché le confezioni aperte hanno tempi di degrado più veloci) e, al contempo, realizzare i desideri dei clienti, stimati in base all'analisi dei piatti più graditi e meglio recensiti di una determinata sede della catena di ristorazione.

Posto:

- n il numero di ingredienti per una determinata ricetta
- K_i il numero di confezioni aperte relative all'elemento i -esimo di una ricetta
- T_i il numero di giorni alla scadenza relativi alla confezione con scadenza inferiore dell'ingrediente i -esimo
- R il punteggio medio della ricetta ottenuto dalla media delle recensioni
- $w1 = 0,2$ costante atta a sfavorire il punteggio della ricetta
- $w2 = 0,8$ costante atta a favorire la vendita di ingredienti prossimi alla scadenza

Si definisce una funzione F :

$$F = w1 * R + w2 * \sum_{i=1}^n \frac{(1 + K_i)}{T_i}$$

Tramite la funzione viene calcolato il punteggio per ogni piatto e proposti i dieci con il punteggio più alto mediante una stringa formattata.

```
DELIMITER $$
CREATE FUNCTION Punteggio (Ricetta INT(11), Sede INT(11))
RETURNS DOUBLE NOT DETERMINISTIC
BEGIN
DECLARE Finito INT DEFAULT 0;
DECLARE Ingrediente INT;
DECLARE GiorniScadenza INT;
DECLARE ConfezioniPresenti INT;
DECLARE Punteggio DOUBLE DEFAULT 0;
DECLARE MediaRecensioni DOUBLE;
DECLARE ConfezioniAperte INT;
DECLARE ScorriIngredienti CURSOR FOR (
    SELECT DISTINCT(C.CodIngrediente), DATEDIFF(C.Scadenza, CURRENT_DATE)
    AS
        GiorniAllaScadenza
    FROM Confezione C INNER JOIN Magazzino M USING(CodMagazzino)
        INNER JOIN Passaggio P USING(CodIngrediente)
    WHERE C.Scadenza = ( SELECT MIN(C1.Scadenza)
        FROM Confezione C1 INNER JOIN Magazzino M1 USING(CodMagazzino)
        WHERE C.CodIngrediente = C1.CodIngrediente
            AND M1.CodSede=Sede)
        AND M.CodSede = Sede
        AND P.CodRicetta = Ricetta);
DECLARE CONTINUE HANDLER FOR NOT FOUND SET Finito = 1;

OPEN ScorriIngredienti;
scan: LOOP
    FETCH ScorriIngredienti INTO Ingrediente, GiorniScadenza;
    IF Finito = 1 THEN
```

```

        LEAVE scan;
    END IF;
    /* verifico se non ci sono confezioni relative all'ingrediente,
       in tal caso sarebbe impossibile preparare la ricetta */
    SELECT COUNT(*) INTO ConfezioniPresenti
    FROM Confezione C INNER JOIN Magazzino M USING(CodMagazzino)
    WHERE C.CodIngrediente = Ingrediente
           AND M.CodSede = Sede;

    IF ConfezioniPresenti = 0 THEN
        SET Punteggio = -1;
        LEAVE scan;
    END IF;

    /* conto le confezioni aperte */
    SELECT COUNT(*) INTO ConfezioniAperte
    FROM Confezione C INNER JOIN Magazzino M USING(CodMagazzino)
    WHERE M.CodSede = Sede
           AND C.CodIngrediente = Ingrediente
           AND C.Stato = 'parziale';

    /* effettuo 1+scatole aperte / giorni alla scadenza
       e la sommo al punteggio ottenuto dagli altri ingredienti */
    SET Punteggio = Punteggio + ((1 + ConfezioniAperte) / GiorniScadenza);
END LOOP scan;
CLOSE ScorriIngredienti;

IF Punteggio >-1 THEN
    /* recupero il punteggio medio delle recensioni*/
    SELECT SommaRecensioni / NumeroRecensioni INTO MediaRecensioni
    FROM Ricetta
    WHERE CodRicetta = Ricetta;

    /* se non esistono recensioni viene assegnato un punteggio pari alla
       media delle recensioni sugli altri piatti per non penalizzare il
       piatto con un punteggio nullo */
    IF MediaRecensioni IS NULL THEN
        SELECT AVG(SommaRecensioni / NumeroRecensioni) INTO
        MediaRecensioni
        FROM Ricetta;
    END IF;

    /* calcolo il punteggio finale pesando i punti delle recensione */
    SET Punteggio = (0.8 * Punteggio) + (0.2 * MediaRecensioni);
END IF;

RETURN Punteggio;
/* se il valore di ritorno =-1, la ricetta non viene proposta */
END;

/* Creo la procedura per richiamare le ricette */

CREATE PROCEDURE MagazzinoIntelligente (IN Sede INT, INOUT Stringa
VARCHAR(255))
BEGIN
    DECLARE Finito INT DEFAULT 0;
    DECLARE Ricetta INT;
    DECLARE NomeRic VARCHAR(255) DEFAULT "";
    DECLARE PuntiRicetta DOUBLE;
    DECLARE ScorriRicette CURSOR FOR ( SELECT CodRicetta
                                         FROM Ricetta);

    DECLARE Stampa CURSOR FOR (
        SELECT NomeRicetta

```

```

        FROM PunteggioRicetta INNER JOIN Ricetta USING (CodRicetta)
        ORDER BY PunteggioR DESC
        LIMIT 10);
DECLARE CONTINUE HANDLER FOR NOT FOUND SET Finito = 1;

/* creo la tabella dove salvare le ricette con i pnteggi */
CREATE TEMPORARY TABLE IF NOT EXISTS PunteggioRicetta (
    CodRicetta INT(11) PRIMARY KEY,
    PunteggioR DOUBLE
)ENGINE = InnoDB DEFAULT CHARSET = UTF8;
TRUNCATE PunteggioRicetta;

OPEN ScorriRicette;
scan: LOOP
    FETCH ScorriRicette INTO Ricetta;

    IF Finito = 1 THEN
        LEAVE scan;
    END IF;

    SET PuntiRicetta = Punteggio(Ricetta, Sede);

    /* Se i punti sono inferiori a zero vuol dire che non ci sono gli
    ingredienti necessari e quindi non può essere proposto il
    piatto */
    IF PuntiRicetta >-1 THEN
        INSERT INTO PunteggioRicetta VALUES (Ricetta, PuntiRicetta);
    END IF;

END LOOP scan;
CLOSE ScorriRicette;

SET Finito = 0;

OPEN Stampa;
scan2: LOOP
    FETCH Stampa INTO NomeRic;

    IF Finito = 1 THEN
        LEAVE scan2;
    END IF;
    /* aggiungo gli elementi in ordine di punteggio decrescente */
    SET Stringa = CONCAT(Stringa, NomeRic, ";");
END LOOP scan2;
CLOSE Stampa;

DROP TEMPORARY TABLE PunteggioRicetta;
END; $$
DELIMITER ;

```

Qualità del take-away

L'area analytics gestisce anche il take-away in maniera ottimizzata. In particolare, nel back-end deve essere presente una funzionalità di analisi dell'efficienza del take-away, la quale valuta, con una certa cadenza, i tempi di consegna e i tempi di rientro dei pony. Tale valutazione è effettuata comparando i tempi di ciascuna

consegna con i tempi medi ottenuti dall'analisi globale delle consegne, alla ricerca di potenziali problematiche da risolvere, per migliorare costantemente la qualità del servizio fornito.

Per rendere possibile individuare falle nel sistema di trasporto viene realizzata una materialized-view aggiornata con cadenza mensile contenente:

- **codiceVeicolo:** il codice del veicolo che ha consegnato la comanda;

- **codiceAsporto:** il codice della comanda oggetto della consegna;
- **ritardoConsegna:** differenza tra il tempo di consegna e il tempo medio delle consegne di tutti i fattorini relativamente al mese in corso;
- **ritardoRientro:** differenza tra il tempo di rientro e il tempo medio di rientro di tutti i fattorini relativamente al mese in corso;

Tenendo presente che i luoghi di consegna hanno distanza variabile, vengono inserite le consegne che hanno un tempo di consegna o un tempo di rientro superiore al 20% rispetto alla rispettiva media di tutti i fattorini per il mese corrente.

```
CREATE TABLE IF NOT EXISTS mv_QualitaTakeAway (
    CodAsporto INT(11) PRIMARY KEY,
    Pony INT(11) NOT NULL,
    RitardoConsegna INT(11) NOT NULL,
    RitardoRientro INT(11) NOT NULL,
    FOREIGN KEY (CodAsporto) REFERENCES ComandaTakeAway(CodAsporto)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (Pony) REFERENCES Pony(CodVeicolo)
        ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET= UTF8;

DELIMITER $$
CREATE EVENT QualitaTakeAway
ON SCHEDULE EVERY 1 MONTH ON COMPLETION PRESERVE
DO
BEGIN
    DECLARE MediaConsegne INT DEFAULT 0;
    DECLARE MediaRientri INT DEFAULT 0;
    DECLARE Finito INT DEFAULT 0;

    /* dichiaro i limiti sopra i quali la consegna viene inserita */
    DECLARE LimiteConsegne INT DEFAULT 0;
    DECLARE LimiteRientri INT DEFAULT 0;
    DECLARE Pony INT;
    DECLARE Partenza DATETIME;
    DECLARE Consegna DATETIME;
    DECLARE Rientro DATETIME;
    DECLARE Asporto INT;
    DECLARE ConsegnaAttuale INT;
    DECLARE RientroAttuale INT;
    DECLARE Inserimento CURSOR FOR (
        SELECT CodAsporto, CodVeicolo, oraPartenza, oraConsegna, oraRitorno
        FROM ComandaTakeAway INNER JOIN Pony USING (CodVeicolo)
        WHERE OraAcquisizione BETWEEN CURRENT_DATE - INTERVAL 1 MONTH
            AND CURRENT_DATE - INTERVAL 1 DAY);
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET Finito = 1;

    /* calcolo la media dei tempi per il mese in oggetto */
    SELECT AVG(D.TempiConsegne) INTO MediaConsegne
    FROM ( SELECT TIMESTAMPDIFF(MINUTE, oraPartenza, oraConsegna) AS
    TempiConsegne
        FROM ComandaTakeAway INNER JOIN Pony USING (CodVeicolo)
        WHERE OraAcquisizione BETWEEN CURRENT_DATE - INTERVAL 1 MONTH AND
            CURRENT_DATE - INTERVAL 1 DAY) AS D;

    SELECT AVG(D.TempiRientri) INTO MediaRientri
    FROM ( SELECT TIMESTAMPDIFF(MINUTE, oraConsegna, oraRitorno) AS TempiRientri
        FROM ComandaTakeAway INNER JOIN Pony USING (CodVeicolo)
```

```

WHERE OraAcquisizione BETWEEN CURRENT_DATE - INTERVAL 1 MONTH AND
      CURRENT_DATE - INTERVAL 1 DAY) AS D;

/* pongo i margini per la segnalazione */
SET LimiteConsegne = MediaConsegne * 1.2;
SET LimiteRientri = MediaRientri * 1.2;

OPEN Inserimento;
scan: LOOP
  FETCH Inserimento INTO Asporto, Pony, Partenza, Consegna, Rientro;
  IF Finito = 1 THEN
    LEAVE scan;
  END IF;
  SET ConsegnaAttuale = TIMESTAMPDIFF(MINUTE, Partenza, Consegna);
  SET RientroAttuale = TIMESTAMPDIFF(MINUTE, Consegna, Rientro);

  /* verifico se inserire la consegna */
  IF ConsegnaAttuale >= LimiteConsegne OR RientroAttuale >= LimiteRientri
  THEN

    /* preparo i ritardi da inserire nella tabella */
    SET ConsegnaAttuale = ConsegnaAttuale - MediaConsegne;
    SET RientroAttuale = RientroAttuale - MediaRientri;

    INSERT INTO _QualitaTakeAway VALUES (Asporto, Pony,
ConsegnaAttuale,
      RientroAttuale);

  END IF;
END LOOP scan;
CLOSE Inserimento;

END; $$
DELIMITER ;

```

Fornitura automatizzata del magazzino

Si desidera una funzionalità per la gestione automatica degli ordini di fornitura del magazzino. In particolare, il sistema deve contenere stored routine che, giornalmente o settimanalmente, confezionano ordinativi da trasmettere ai fornitori.

Tali ordinativi sono relativi agli ingredienti consumati con regolarità, di cui il magazzino non può essere privo. Gli ordinativi contengono gli ingredienti da acquistare con le rispettive quantità. La tipologia di ingredienti e le rispettive quantità sono stimate in base a previsioni basate sull'analisi degli andamenti di frequentazione delle sedi della catena di ristorazione, e degli acquisti realizzati nel take-away.

E' costruita una materialized view contenente, per ogni sede, la quantità q di ogni ingrediente di cui i magazzini della sede devono essere riforniti. La **mv_rifornimento** viene aggiornata **ogni settimana** mediante un **event**.

Si consideri un ingrediente, una sede e l'insieme $R_1 \dots R_N$ delle ricette che lo coinvolgono.

Presa la ricetta R_i sia:

- d_i la dose dell'ingrediente coinvolto nella ricetta i -esima
- nc_i il numero di ordinazioni (nel ristorante) della ricetta i -esima
- nt_i il numero di ordinazioni (nel takeaway) della medesima ricetta
- p_{pp} il numero di prenotazioni nella sede per il periodo precedente
- p_{ps} il numero di prenotazioni per il periodo successivo

La quantità da rifornire per la sede dell'ingrediente è data come:

$$(p_{pp} + p_{ps}) * \left[\sum_{i=0}^n d_i * nc_i \right] + \sum_{i=0}^n d_i * nt_i$$

```

CREATE TABLE IF NOT EXISTS MV_Rifornimento (
    Ingrediente INT NOT NULL,
    Sede INT NOT NULL,
    Quantita INT NOT NULL, /*Espressa in gr*/
    FOREIGN KEY(Ingrediente) REFERENCES Ingrediente(CodIngrediente),
    FOREIGN KEY(Sede) REFERENCES Sede(CodSede)
);

CREATE OR REPLACE VIEW IngredientiRicetta AS
SELECT DISTINCT CodRicetta,CodIngrediente,Dose
FROM Passaggio;

CREATE OR REPLACE VIEW OrdinazioniRistoranteRicetta AS
SELECT T.CodSede,O.CodRicetta,COUNT(*) NumeroOrdinazioni
FROM Tavolo T INNER JOIN Comanda C USING(CodTavolo)
    INNER JOIN Ordine O USING(CodComanda)
WHERE C.OraAcquisizione >NOW() - INTERVAL 1 YEAR
GROUP BY T.CodSede,O.CodRicetta;

CREATE OR REPLACE VIEW OrdinazioniTakeRicetta AS
SELECT CodSede,CodRicetta,COUNT(*) NumeroOrdinazioni
FROM comandaTakeAway INNER JOIN Ordine O USING(CodAsporto)
GROUP BY CodSede,CodRicetta;

CREATE OR REPLACE VIEW ConsumoIngredientiSedeCoperto AS
SELECT CodIngrediente,CodSede,Dose*NumeroOrdinazioni AS Consumo
FROM IngredientiRicetta NATURAL JOIN OrdinazioniRistoranteRicetta
WHERE CodIngrediente != 23
GROUP BY CodIngrediente,CodSede
ORDER BY CodIngrediente,CodSede;

CREATE OR REPLACE VIEW ConsumoIngredientiSedeTake AS
SELECT CodIngrediente,CodSede,Dose*NumeroOrdinazioni AS Consumo
FROM IngredientiRicetta NATURAL JOIN OrdinazioniTakeRicetta
WHERE CodIngrediente !=23
GROUP BY CodIngrediente,CodSede
ORDER BY CodIngrediente,CodSede;

CREATE OR REPLACE VIEW PrenotazioniProxSett AS
SELECT CodSede,SUM(NumeroCoperti) AS Persone
FROM Coperto INNER JOIN Tavolo USING(CodTavolo)
    INNER JOIN Prenotazione USING(CodPrenotazione)
WHERE DataPrenotazione BETWEEN NOW() AND NOW() + INTERVAL 1 WEEK
    AND DataPrenotazione >NOW()
GROUP BY CodSede;

CREATE OR REPLACE VIEW PrenotazioniPrecSett AS
SELECT CodSede,SUM(NumeroCoperti) AS Persone
FROM Coperto INNER JOIN Tavolo USING(CodTavolo)
    INNER JOIN Prenotazione USING(CodPrenotazione)
WHERE DataPrenotazione BETWEEN NOW() AND NOW() - INTERVAL 1 WEEK
GROUP BY CodSede;

CREATE OR REPLACE VIEW SpecchioUtilizzoIngrediente AS
SELECT CodIngrediente,CodSede,IF(Consumo IS NULL , 0, Consumo) *
    (S.Persone+P.Persone) AS Consumo
FROM Ingrediente I LEFT OUTER JOIN ConsumoIngredientiSedeCoperto
    USING(CodIngrediente)
    INNER JOIN PrenotazioniProxSett S USING(CodSede)
    INNER JOIN PrenotazioniPrecSett P USING(CodSede)

UNION ALL

```

```

SELECT CodIngrediente,CodSede,IF(Consumo IS NULL,0,Consumo) AS Consumo
FROM Ingrediente I LEFT OUTER JOIN ConsumoIngredientiSedeTake
      USING(CodIngrediente);

REPLACE INTO MV_Rifornimento
SELECT CodIngrediente,CodSede,SUM(Consumo) AS Quantità
FROM SpecchioUtilizzoIngrediente
WHERE CodSede IS NOT NULL
GROUP BY CodIngrediente,CodSede
ORDER BY Consumo DESC;

/* Stored procedure per il refresh della materialized view */
DELIMITER $$
CREATE PROCEDURE refresh_MV_Rifornimento(OUT esito INT)
BEGIN
DECLARE esito INT DEFAULT 0;
DECLARE EXIT HANDLER FOR SQLEXCEPTION
BEGIN
    ROLLBACK;
    SET esito = 1;
    SELECT "Errore, impossibile aggiornare la MV_Rifornimento";
END;

TRUNCATE TABLE MV_Rifornimento;

REPLACE INTO MV_Rifornimento
SELECT CodIngrediente,CodSede,SUM(Consumo)*Persone AS Quantità
FROM SpecchioUtilizzoIngrediente INNER JOIN PrenotazioniProxSett
      USING(CodSede)
WHERE CodSede IS NOT NULL
GROUP BY CodIngrediente,CodSede
ORDER BY Consumo;

END $$
DELIMITER ;

/* Event per il deferred refresh, schedule every 1 week */
DELIMITER $$
CREATE EVENT deferred_refresh_Rifornimento
ON SCHEDULE EVERY 1 WEEK
ON COMPLETION PRESERVE
DO
BEGIN
SET @esito = 0;
CALL refresh_MV_Rifornimento(@esito);
IF (@esito = 1) THEN
    SIGNAL SQLSTATE '45000';
    SET MESSAGE_TEXT = 'Errore refresh temporal trigger';
END IF;
END $$
DELIMITER ;

```