



**Università di Pisa**

**Progetto per il corso di Basi di Dati**

*Realizzato da Diego Casu e Luigi Leonardi*

**Anno Accademico 2014/2015**



# Indice

<i>Introduzione</i>	4
<i>Cardinalità e associazioni</i>	5
<i>Tabelle</i>	11
<i>Normalizzazione e dipendenze funzionali</i>	12
<i>Vincoli di integrità referenziale</i>	17
<i>Tavola dei volumi</i>	20
<i>Ridondanze introdotte e operazioni sulla base di dati</i>	26
<i>Operazioni in SQL e sintassi MySQL</i>	32
<i>Analytic Function: Analisi multidimensionale del business</i>	37
<i>Analytic Function: Magazzino intelligente</i>	40
<i>Analytic Function: Qualità del take away</i>	44



## Introduzione

Il progetto presentato in queste pagine si propone come obiettivo la realizzazione di una base di dati relazionale che gestisca il sistema informativo di una catena di ristoranti, fornendo approfondite funzionalità di analisi e organizzazione dei dati stessi. Il database si occupa di mettere in relazione tre diverse aree funzionali della catena: l'area gestione, l'area clienti e l'area analytics. La prima area è incentrata sull'attività di ristorazione nelle diverse sedi e si occupa principalmente: dell'amministrazione dei magazzini, con il relativo stoccaggio degli ingredienti in confezioni caratterizzate da parametri di valutazione quali peso, prezzo, qualità, integrità; dell'amministrazione delle cucine e dei piatti che da esse possono essere prodotti, con conseguente attenzione alla catalogazione di ricette e variazioni predisposte; della gestione dell'intero sistema di ricezione e preparazione degli ordini, costituiti da comande richieste direttamente dai tavoli oppure da comande take away, evase da personale specializzato. L'area clienti ha la funzione fondamentale di interagire con la clientela e di raccogliere informazioni utili al miglioramento dell'attività stessa attraverso recensioni, valutazioni e proposte che arrivano direttamente dai clienti tramite una piattaforma web, di cui il database è sotto-struttura principale. L'ultima area è rappresentata da un insieme di funzionalità volte a supportare le decisioni di business, ad analizzare l'efficienza delle operazioni effettuate nella catena, a realizzare previsioni e statistiche utili nella gestione sistemica. La progettazione del database, a partire dalle specifiche fornite, è stata suddivisa in fasi successive di elaborazione e traduzione dei modelli: nelle sezioni della documentazione e nel diagramma Entità-Relazione allegato sono stati individuati e sviluppati tutti i dettagli necessari al corretto funzionamento del base di dati, affiancati dalle motivazioni che hanno portato all'adozione di particolari soluzioni realizzative.

## Cardinalità e associazioni

Di seguito sono elencate le associazioni che compongono il diagramma Entità-Relazione, insieme alle relative cardinalità. Le scelte di traduzione derivano, nella maggior parte dei casi, direttamente dalle specifiche fornite: le aggiunte o le modifiche apportate sono state state introdotte per sviluppare in profondità le funzionalità essenziali e caratteristiche della base di dati. Insieme alle cardinalità è stata inserita una breve motivazione che ne giustifica il ruolo all'interno della propria associazione.

### Associazione "Dispone di":

- Cardinalità (1,1) con MAGAZZINO, perchè un magazzino appartiene ad una ed una sola sede ;
- Cardinalità (1,N) con SEDE, perchè una sede può avere più di un magazzino.

### Associazione "Stocca":

- Cardinalità (1,1) con CONFEZIONE, perchè una confezione si trova in un solo magazzino ;
- Cardinalità (0,N) con MAGAZZINO, perchè un magazzino può stoccare nessuna, una o più di una confezione.

### Associazione "Possiede":

- Cardinalità (1,1) con TAVOLO, perchè un tavolo si trova in una sola sede ;
- Cardinalità (1,N) con SEDE, perchè una sede possiede almeno un tavolo.

### Associazione "Contiene":

- Cardinalità (1,1) con CONFEZIONE, perchè dentro una confezione si trova un solo ingrediente ;
- Cardinalità (0,N) con INGREDIENTE, perchè di un ingrediente possono esserci nessuna, una o più di una confezione.

### Associazione "Svolge":

- Cardinalità (1,N) con FUNZIONE, perchè una funzione può essere svolta da più di un macchinario/attrezzatura ;
- Cardinalità (1,N) con MACCHINARIO/ATTREZZATURA, perchè un macchinario/attrezzatura può svolgere una o più funzioni.

### Associazione "Organizza":

- Cardinalità (1,1) con MENU, perchè un menu appartiene ad una sola sede ;
- Cardinalità (1,N) con SEDE, perchè una sede stila più menu nel tempo.

### Associazione "Compare nel":

- Cardinalità (0,N) con RICETTA, perchè una ricetta può comparire in nessuno, uno o più di un menu ;
- Cardinalità (1,N) con MENU, perchè un menu può essere formato da una o più ricette.

### **Associazione "Prevede":**

- Cardinalità (0,N) con RICETTA, perchè una ricetta può avere nessuna, una o più di una variazione ;
- Cardinalità (1,1) con VARIAZIONE, perchè una variazione si riferisce ad una ed una sola ricetta.

### **Associazione "Adopera":**

- Cardinalità (1,N) con INGREDIENTE, perchè un ingrediente può appartenere a più di una ricetta ;
- Cardinalità (1,N) con RICETTA, perchè in una ricetta si adopera almeno un ingrediente.

### **Associazione "Utilizza":**

- Cardinalità (0,1) con FASE DI PREPARAZIONE, perchè una fase di preparazione coinvolge nessuno o un solo ingrediente ;
- Cardinalità (0,N) con INGREDIENTE, perchè un ingrediente può essere utilizzato in nessuna, una o più di una fase di preparazione.

### **Associazione "Coinvolge":**

- Cardinalità (0,1) con FASE DI PREPARAZIONE, perchè una fase di preparazione coinvolge nessuno o un solo ingrediente ;
- Cardinalità (0,N) con MACCHINARIO/ATTREZZATURA, perchè un macchinario può essere impiegato in nessuna, una o più di una fase di preparazione.

### **Associazione "Cambia":**

- Cardinalità (0,N) con MACCHINARIO/ATTREZZATURA, perchè un macchinario può essere impiegato per nessuna, una o più di una variazione ;
- Cardinalità (0,1) con VARIAZIONE, perchè una variazione può non prevedere l'utilizzo di una macchinario/attrezzatura.

### **Associazione "Divisa in":**

- Cardinalità (1,1) con FASE DI PREPARAZIONE, perchè una fase di preparazione si riferisce ad una e una sola ricetta ;
- Cardinalità (1,N) con RICETTA, perchè una ricetta può avere più fasi di preparazione.

### **Associazione "Modifica":**

- Cardinalità (0,N) con FASE DI PREPARAZIONE, perchè una fase di preparazione può avere nessuna, una o più di una variazione ;
- Cardinalità (0,1) con VARIAZIONE, perchè una variazione può non modificare una fase di preparazione.

### **Associazione "Interessa":**

- Cardinalità (0,N) con INGREDIENTE, perchè un ingrediente può non essere utilizzato in una variazione, oppure in più di una ;
- Cardinalità (0,1) con VARIAZIONE, perchè una variazione può non impiegare nessun ingrediente.

### **Associazione "Riguarda":**

- Cardinalità (0,N) con VARIAZIONE, perchè una variazione appartiene a nessuno, uno o più di un piatto ordinato ;
- Cardinalità (0,N) con PIATTO ORDINATO, perchè per un piatto ordinato può avere nessuna, una o più di una variazione.

### **Associazione "Produce":**

- Cardinalità (0,1) con PIATTO ORDINATO, perchè un piatto ordinato può non appartenere ad una comanda "normale" ;
- Cardinalità (1,N) con COMANDA, perchè una comanda può comprendere più di un piatto ordinato.

### **Associazione "Impiega":**

- Cardinalità (1,1) con MACCHINARIO/ATTREZZATURA, perchè una macchinario/attrezzatura appartiene ad una ed una sola sede ;
- Cardinalità (1,N) con SEDE, perchè una sede può possedere più di una macchinario/attrezzatura.

### **Associazione "Genera":**

- Cardinalità (0,1) con PIATTO ORDINATO, perchè un piatto ordinato può non appartenere ad una comanda Take Away ;
- Cardinalità (1,N) con COMANDA TAKE AWAY, perchè una comanda Take Away può comprendere più di un piatto ordinato.

### **Associazione "Gestisce":**

- Cardinalità (1,N) con SEDE, perchè una sede può gestire una "flotta" di pony ;
- Cardinalità (1,1) con PONY, perchè un pony è gestito da una ed una sola sede.

### **Associazione "Si Riferisce":**

- Cardinalità (0,N) con TAVOLO, perchè per un tavolo possono esserci nessuna, una o più di una comanda ;
- Cardinalità (1,1) con COMANDA, perchè una comanda si riferisce ad uno ed un solo tavolo.

### **Associazione "Redige":**

- Cardinalità (1,N) con SEDE, perchè una sede può redigere più di un questionario ;
- Cardinalità (1,1) con QUESTIONARIO, perchè un questionario si riferisce ad una ed una sola sede.

### **Associazione "Valuta":**

- Cardinalità (1,1) con RECENSIONE, perchè una recensione si riferisce ad una ed una sola sede ;
- Cardinalità (0,N) con SEDE, perchè una sede può avere nessuna, una o più di una recensione.



### **Associazione " È associata":**

- Cardinalità (0,N) con RECENSIONE, perchè una recensione può ricevere nessuna, una o più di una valutazione ;
- Cardinalità (1,1) con VALUTAZIONE RECENSIONE, perchè una valutazione è associata ad una ed una sola recensione.

### **Associazione "Evade":**

- Cardinalità (0,N) con PONY, perchè un pony prende in carico nessuna, una o più di una comanda Take Away ;
- Cardinalità (1,1) con COMANDA TAKE AWAY, perchè una comanda Take Away è presa in carico da uno ed un solo pony.

### **Associazione "Muta":**

- Cardinalità (0,N) con RICETTA, perchè per una ricetta vi può essere nessuna, una o più di una variante ;
- Cardinalità (1,1) con VARIANTE, perchè una variante si riferisce ad una ed una sola ricetta.

### **Associazione "Stima":**

- Cardinalità (0,N) con VARIANTE, perchè una variante può ricevere nessuna, una o più di una valutazione ;
- Cardinalità (0,1) con VALUTAZIONE, perchè una valutazione può riferirsi a nessuna o una variante.

### **Associazione "Giudica":**

- Cardinalità (0,N) con VARIANTE PIATTO, perchè una variante piatto può ricevere nessuna, una o più di una valutazione ;
- Cardinalità (0,1) con VALUTAZIONE, perchè una valutazione può riferirsi a nessuno o ad una variante piatto.

### **Associazione "Suggerisce":**

- Cardinalità (0,N) con ACCOUNT, perchè un account può suggerire nessuna, una o più di una variante piatto ;
- Cardinalità (1,1) con VARIANTE PIATTO, perchè una variante piatto è suggerita da uno ed un solo account.

### **Associazione "Promuove":**

- Cardinalità (0,N) con ACCOUNT, perchè un account può suggerire nessuna, una o più di una variante ;
- Cardinalità (1,1) con VARIANTE, perchè una variante è suggerita da uno ed un solo account.

### **Associazione "Opera":**

- Cardinalità (0,N) con ACCOUNT, perchè un account può operare nessuna, una o più di una valutazione ;
- Cardinalità (1,1) con VALUTAZIONE, perchè una valutazione è data da uno ed un solo account.

### **Associazione "Decide":**

- Cardinalità (1,1) con VALUTAZIONE RECENSIONE, perchè una valutazione recensione è data da uno ed un solo account ;
- Cardinalità (0,N) con ACCOUNT, perchè un account può operare nessuna, una o più di una valutazione di recensione.

### **Associazione "Scrive":**

- Cardinalità (1,1) con RECENSIONE, perchè una recensione è scritta da uno ed un solo account ;
- Cardinalità (0,N) con ACCOUNT, perchè un account può scrivere nessuna, una o più di una recensione.

### **Associazione "Ordina":**

- Cardinalità (0,N) con ACCOUNT, perchè un account può richiedere nessuna, una o più di una comanda Take Away ;
- Cardinalità (1,1) con COMANDA TAKE AWAY, perchè una comanda Take Away è richiesta da uno ed un solo account.

### **Associazione "Propone":**

- Cardinalità (0,N) con ACCOUNT, perchè un account può proporre nessuna, una o più di una serata a tema proposta ;
- Cardinalità (1,1) con SERATA A TEMA PROPOSTA, perchè una serata a tema è proposta da uno ed un solo account.

### **Associazione "Formata":**

- Cardinalità (1,N) con RECENSIONE, perchè una recensione può avere più risposte ;
- Cardinalità (0,N) con RISPOSTA, perchè una risposta appartiene a nessuna, una o più di una recensione.

### **Associazione "Ottiene":**

- Cardinalità (1,N) con DOMANDA, perchè per una domanda vi sono più risposte ;
- Cardinalità (1,1) con RISPOSTA, perchè una risposta si riferisce ad una ed una sola domanda.

### **Associazione "Effettua":**

- Cardinalità (0,1) con PRENOTAZIONE, perchè una prenotazione può non essere effettuata da un account ;
- Cardinalità (0,N) con ACCOUNT, perchè un account può effettuare nessuna, una o più di una prenotazione.

**Associazione "Sottopone":**

- Cardinalità (1,N) con QUESTIONARIO, perchè per un questionario vi possono essere più domande;
- Cardinalità (1,1) con DOMANDA, perchè una domanda appartiene ad uno ed un solo questionario.

**Associazione " È assegnato":**

- Cardinalità (0,N) con TAVOLO, perchè un tavolo può avere nessuna, una o più di una prenotazione ;
- Cardinalità (1,1) con PRENOTAZIONE, perchè ad una prenotazione è assegnato uno ed un solo tavolo.

**Associazione "Deriva":**

- Cardinalità (0,N) con RICETTA, perchè un ricetta può avere nessuna, una o più di una ordinazione ;
- Cardinalità (1,1) con PIATTO ORDINATO, perchè un piatto ordinato ha una ed una sola ricetta.

**Associazione "È oggetto di":**

- Cardinalità (1,1) con RICETTA, perchè una ricetta ha uno ed un solo questionario ;
- Cardinalità (0,1) con QUESTIONARIO, perchè un questionario può non riferirsi ad una ricetta.

## Tabelle

Di seguito sono elencate le tabelle che compongono la base di dati, ricavate dal diagramma Entità-Relazione. Nel processo di costruzione delle stesse si è posta particolare attenzione nella traduzione delle associazioni e alla eventuale distribuzione degli attributi, in modo tale da conservare e sviluppare le relazioni tra le aree funzionali della struttura dati.

- **Account** ( CodFiscale, Nome, Cognome, Città, Indirizzo, Sesso, Bandito ) ;
- **Adopera** ( IdRicetta, IdIngrediente, FunzioneNellaPietanza ) ;
- **Comanda** ( IdComanda, Stato, Timestamp, IdTavolo ) ;
- **ComandaTakeAway** ( IdComandaTakeAway, Timestamp, Stato, TimestampPartenza, TimestampConsegna, TimestampRientro, IdPony, CodFiscale ) ;
- **CompareNel** ( IdRicetta, IdMenu ) ;
- **Confezione** ( IdConfezione, CodiceLotto, PrezzoAcquisto, DataAcquisto, Peso, Aspetto, IndicatoreCollocazione, Stato, DataScadenza, DataArrivo, IdMagazzino, IdIngrediente ) ;
- **Domanda** ( IdDomanda, Testo, IdQuestionario ) ;
- **FasePreparazione** ( IdFase, Durata, Dose, Manovra, IdMacchina, IdIngrediente, IdRicetta ) ;
- **Formata** ( IdRecensione, IdRisposta ) ;
- **Funzione** ( IdFunzione, Descrizione ) ;
- **Ingrediente** ( IdIngrediente, Nome, Genere, Provenienza, TipoProduzione, Allergene ) ;
- **MacchinarioAttrezzatura** ( IdMacchina, Nome, Tipologia, IdSede ) ;
- **Magazzino** ( IdMagazzino, Indirizzo, Città, CAP, NumeroTelefono, IdSede ) ;
- **Menu** ( IdMenu, DataInizio, DataFine, IdSede ) ;
- **PiattoOrdinato** ( IdPiatto, Stato, IdRicetta, IdComanda, IdComandaTakeAway ) ;
- **Pony** ( IdPony, TipologiaMezzo, IdSede ) ;
- **Prenotazione** ( IdPrenotazione, NumeroPersone, Data, Ora, NumeroTelefono, Allestimenti, Rispettata, IdTavolo, CodFiscale ) ;
- **Questionario** ( IdQuestionario, Autore, DataCreazione, InUso, IdSede, IdRicetta ) ;
- **Recensione** ( IdRecensione, Testo, GiudizioGlobale, IdSede, CodFiscale ) ;
- **Ricetta** ( IdRicetta, Nome, Testo, Descrizione, Prezzo ) ;
- **Riguarda** ( IdVariazione, IdPiatto ) ;
- **Risposta** ( IdRisposta, PunteggioEfficienza, IdDomanda ) ;
- **Sede** ( IdSede, Nome, Indirizzo, Città, CAP, NumeroTelefono ) ;
- **SerataTemaProposta** ( IdSerata, Allestimenti, Data, NumeroPersone, Approvata, CodFiscale ) ;
- **Svolge** ( IdFunzione, IdMacchina ) ;
- **Tavolo** ( IdTavolo, NumeroPosti, Sala, IdSede ) ;
- **Valutazione** ( IdValutazione, Punteggio, CodFiscale, IdVariante, IdVariantePiatto ) ;
- **ValutazioneRecensione** ( IdValutazione, Testo, GradoAccuratezza, GradoVeridicità, IdRecensione, CodFiscale ) ;
- **Variante** ( IdVariante, Nome, Descrizione, CodFiscale, IdRicetta ) ;
- **VariantePiatto** ( IdVariantePiatto, Nome, Ricetta, CodFiscale ) ;
- **Variazione** ( IdVariazione, Nome, Descrizione, Tipologia, Durata, Prezzo, IdMacchina, IdRicetta, IdIngrediente, IdFase ) ;

## Normalizzazione e dipendenze funzionali

La strategia di progettazione adottata e le scelte operate in fase di traduzione delle specifiche hanno portato direttamente alla costruzione di una base di dati in forma normale di Boyce e Codd: di seguito sono elencate, per ciascuna tabella, le dipendenze funzionali ad essa associata e una breve analisi che determina la forma normale. Non è stato necessario applicare nessun processo di normalizzazione successivo alla definizione della composizione delle tabelle.

### Tabella Account :

ACCOUNT ( CodFiscale, Nome, Cognome, Città, Indirizzo, Sesso, Bandito ) ;

- CodFiscale → Nome, Cognome, Città, Indirizzo, Sesso, Bandito

Poichè la parte sinistra è superchiave, ACCOUNT è in BCNF.

### Tabella Adopera :

ADOPERA ( IdRicetta, IdIngrediente, FunzioneNellaPietanza ) ;

- IdRicetta, IdIngrediente → FunzioneNellaPietanza

Poichè la parte sinistra è superchiave, ADOPERA è in BCNF.

### Tabella Comanda :

COMANDA ( IdComanda, Stato, Timestamp, IdTavolo ) ;

- IdComanda → Stato, Timestamp, IdTavolo

Poichè la parte sinistra è superchiave, COMANDA è in BCNF.

### Tabella ComandaTakeAway :

COMANDA TAKE AWAY ( IdComandaTakeAway, Timestamp, Stato, TimestampPartenza, TimestampConsegna, TimestampRientro, IdPony, CodFiscale ) ;

- IdComandaTakeAway → Timestamp, Stato, TimestampPartenza, TimestampConsegna, TimestampRientro, IdPony, CodFiscale

Poichè la parte sinistra è superchiave, COMANDA TAKE AWAY è in BCNF.

### Tabella CompareNel :

COMPARE NEL ( IdRicetta, IdMenu ) ;

Poichè non esistono dipendenze funzionali, COMPARE NEL è in BCNF.

### **Tabella Confezione :**

CONFEZIONE ( IdConfezione, CodiceLotto, PrezzoAcquisto, DataAcquisto, Peso, Aspetto, IndicatoreCollocazione, Stato, DataScadenza, DataArrivo, IdMagazzino, IdIngrediente ) ;

- IdConfezione → CodiceLotto, PrezzoAcquisto, DataAcquisto, Peso, Aspetto, IndicatoreCollocazione, Stato, DataScadenza, DataArrivo, IdMagazzino, IdIngrediente

Poichè la parte sinistra è superchiave, CONFEZIONE è in BCNF.

### **Tabella Domanda :**

DOMANDA ( IdDomanda, Testo, IdQuestionario ) ;

- IdDomanda → Testo, IdQuestionario

Poichè la parte sinistra è superchiave, DOMANDA è in BCNF.

### **Tabella FasePreparazione :**

FASE PREPARAZIONE ( IdFase, Durata, Dose, Manovra, IdMacchina, IdIngrediente, IdRicetta ) ;

- IdFase → Durata, Dose, Manovra, IdMacchina, IdIngrediente, IdRicetta

Poichè la parte sinistra è superchiave, FASE PREPARAZIONE è in BCNF.

### **Tabella Formata :**

FORMATATA ( IdRecensione, IdRisposta ) ;

Poichè non esistono dipendenze funzionali, FORMATA è in BCNF.

### **Tabella Funzione :**

FUNZIONE ( IdFunzione, Descrizione ) ;

- IdFunzione → Descrizione

Poichè la parte sinistra è superchiave, FUNZIONE è in BCNF.

### **Tabella Ingrediente :**

INGREDIENTE ( IdIngrediente, Nome, Genere, Provenienza, TipoProduzione, Allergene ) ;

- IdIngrediente → Nome, Genere, Provenienza, TipoProduzione, Allergene

Poichè la parte sinistra è superchiave, INGREDIENTE è in BCNF.

### **Tabella MacchinarioAttrezzatura :**

MACCHINARIO ATTREZZATURA ( IdMacchina, Nome, Tipologia, IdSede ) ;

- IdMacchina → Nome, Tipologia, IdSede

Poichè la parte sinistra è superchiave, MACCHINARIO ATTREZZATURA è in BCNF.

### **Tabella Magazzino :**

MAGAZZINO ( IdMagazzino, Indirizzo,Città, CAP, NumeroTelefono, IdSede ) ;

- IdMagazzino → Indirizzo,Città, CAP, NumeroTelefono, IdSede

Poichè la parte sinistra è superchiave, MAGAZZINO è in BCNF.

### **Tabella Menu :**

MENU ( IdMenu , DataInizio, DataFine, IdSede ) ;

- IdMenu → DataInizio, DataFine, IdSede

Poichè la parte sinistra è superchiave, MENU è in BCNF.

### **Tabella PiattoOrdinato :**

PIATTO ORDINATO ( IdPiatto, Stato, IdRicetta, IdComanda, IdComandaTakeAway ) ;

- IdPiatto → Stato, IdRicetta, IdComanda, IdComandaTakeAway

Poichè la parte sinistra è superchiave, PIATTO ORDINATO è in BCNF.

### **Tabella Pony :**

PONY ( IdPony, TipologiaMezzo, IdSede ) ;

- IdPony → TipologiaMezzo, IdSede

Poichè la parte sinistra è superchiave, PONY è in BCNF.

### **Tabella Prenotazione :**

PRENOTAZIONE ( IdPrenotazione, NumeroPersone, Data, Ora, NumeroTelefono, Allestimenti, Rispettata, IdTavolo, CodFiscale ) ;

- IdPrenotazione → NumeroPersone, Data, Ora, NumeroTelefono, Allestimenti, Rispettata, IdTavolo, CodFiscale

Poichè la parte sinistra è superchiave, PRENOTAZIONE è in BCNF.

### **Tabella Questionario :**

QUESTIONARIO ( IdQuestionario, Autore, DataCreazione, InUso, IdSede, IdRicetta ) ;

- IdQuestionario → Autore, DataCreazione, InUso, IdSede, IdRicetta

Poichè la parte sinistra è superchiave, QUESTIONARIO è in BCNF.

### **Tabella Recensione :**

RECENSIONE ( IdRecensione, Testo, GiudizioGlobale, IdSede, CodFiscale ) ;

- IdRecensione → Testo, GiudizioGlobale, IdSede, CodFiscale

Poichè la parte sinistra è superchiave, RECENSIONE è in BCNF.

### **Tabella Ricetta :**

RICETTA ( IdRicetta, Nome, Testo, Descrizione, Prezzo ) ;

- IdRicetta → Nome, Testo, Descrizione, Prezzo

Poichè la parte sinistra è superchiave, RICETTA è in BCNF.

### **Tabella Riguarda :**

RIGUARDA ( IdVariazione, IdPiatto ) ;

Poichè non esistono dipendenze funzionali, RIGUARDA è in BCNF.

### **Tabella Risposta :**

RISPOSTA ( IdRisposta, PunteggioEfficienza, IdDomanda ) ;

- IdRisposta → PunteggioEfficienza, IdDomanda

Poichè la parte sinistra è superchiave, RISPOSTA è in BCNF.

### **Tabella Sede :**

SEDE ( IdSede, Nome, Indirizzo, Città, CAP, NumeroTelefono ) ;

- IdSede → Nome, Indirizzo, Città, CAP, NumeroTelefono

Poichè la parte sinistra è superchiave, SEDE è in BCNF.

### **Tabella SerataTemaProposta :**

SERATA TEMA PROPOSTA ( IdSerata, Allestimenti, Data, NumeroPersone, Approvata, CodFiscale ) ;

- IdSerata → Allestimenti, Data, NumeroPersone, Approvata, CodFiscale

Poichè la parte sinistra è superchiave, SERATA TEMA PROPOSTA è in BCNF.

### **Tabella Svolge :**

SVOLGE ( IdFunzione, IdMacchina ) ;

Poichè non esistono dipendenze funzionali, SVOLGE è in BCNF.

### **Tabella Tavolo :**

TAVOLO ( IdTavolo, NumeroPosti, Sala, IdSede ) ;

- IdTavolo → NumeroPosti, Sala, IdSede

Poichè la parte sinistra è superchiave, TAVOLO è in BCNF.

### **Tabella Valutazione :**

VALUTAZIONE ( IdValutazione, Punteggio, CodFiscale, IdVariante, IdVariantePiatto ) ;

- IdValutazione → Punteggio, CodFiscale, IdVariante, IdVariantePiatto

Poichè la parte sinistra è superchiave, VALUTAZIONE è in BCNF.



**Tabella ValutazioneRecensione :**

VALUTAZIONE RECENSIONE ( IdValutazione, Testo, GradoAccuratezza, GradoVeridicità, IdRecensione, CodFiscale ) ;

- IdValutazione → Testo, GradoAccuratezza, GradoVeridicità, IdRecensione, CodFiscale

Poichè la parte sinistra è superchiave, VALUTAZIONE RECENSIONE è in BCNF.

**Tabella Variante :**

VARIANTE ( IdVariante, Nome, Descrizione, CodFiscale, IdRicetta ) ;

- IdVariante → Nome, Descrizione, CodFiscale, IdRicetta

Poichè la parte sinistra è superchiave, VARIANTE è in BCNF.

**Tabella VariantePiatto :**

VARIANTE PIATTO ( IdVariantePiatto, Nome, Ricetta, CodFiscale ) ;

- IdVariantePiatto → Nome, Ricetta, CodFiscale

Poichè la parte sinistra è superchiave, VARIANTE PIATTO è in BCNF.

**Tabella Variazione :**

VARIAZIONE ( IdVariazione, Nome, Descrizione, Tipologia, Durata, Prezzo, IdMacchina, IdRicetta, IdIngrediente, IdFase ) ;

- IdVariazione → Nome, Descrizione, Tipologia, Durata, Prezzo, IdMacchina, IdRicetta, IdIngrediente, IdFase

Poichè la parte sinistra è superchiave, VARIAZIONE è in BCNF.

## Vincoli di integrità referenziale

I vincoli di integrità referenziali proposti di seguito derivano dalla composizione del diagramma Entità-Relazione e riflettono le scelte effettuate in sede di traduzione delle associazioni. Le definizioni dei vincoli sono presenti nello script di generazione della base di dati fornito in allegato.

- Esiste un vincolo di integrità referenziale tra l'attributo IdSede della tabella MAGAZZINO e la tabella SEDE ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdMagazzino della tabella CONFEZIONE e la tabella MAGAZZINO ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdIngrediente della tabella CONFEZIONE e la tabella INGREDIENTE ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdSede della tabella MACCHINARIO ATTREZZATURA e la tabella SEDE ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdFunzione della tabella SVOLGE e la tabella FUNZIONE ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdMacchina della tabella SVOLGE e la tabella MACCHINARIO ATTREZZATURA ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdMacchina della tabella FASE PREPARAZIONE e la tabella MACCHINARIO ATTREZZATURA ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdIngrediente della tabella FASE PREPARAZIONE e la tabella INGREDIENTE ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdSede della tabella MENU e la tabella SEDE ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdRicetta della tabella FASE PREPARAZIONE e la tabella RICETTA ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdRicetta della tabella COMPARE NEL e la tabella RICETTA ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdMenu della tabella COMPARE NEL e la tabella MENU ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdRicetta della tabella ADOPERA e la tabella RICETTA ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdIngrediente della tabella ADOPERA e la tabella INGREDIENTE ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdMacchina della tabella VARIAZIONE e la tabella MACCHINARIO ATTREZZATURA ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdRicetta della tabella VARIAZIONE e la tabella RICETTA ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdIngrediente della tabella VARIAZIONE e la tabella INGREDIENTE ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdFase della tabella VARIAZIONE e la tabella FASE PREPARAZIONE ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdVariazione della tabella RIGUARDA e la tabella VARIAZIONE ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdPiatto della tabella RIGUARDA e la tabella PIATTO ORDINATO ;

- Esiste un vincolo di integrità referenziale tra l'attributo IdRicetta della tabella PIATTO ORDINATO e la tabella RICETTA ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdSede della tabella PONY e la tabella SEDE ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdComanda della tabella PIATTO ORDINATO e la tabella COMANDA ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdComandaTakeAway della tabella PIATTO ORDINATO e la tabella COMANDA TAKE AWAY ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdPony della tabella COMANDA TAKE AWAY e la tabella PONY ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdSede della tabella TAVOLO e la tabella SEDE ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdTavolo della tabella COMANDA e la tabella TAVOLO ;
- Esiste un vincolo di integrità referenziale tra l'attributo CodFiscale della tabella COMANDA TAKE AWAY e la tabella ACCOUNT ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdTavolo della tabella PRENOTAZIONE e la tabella TAVOLO ;
- Esiste un vincolo di integrità referenziale tra l'attributo CodFiscale della tabella PRENOTAZIONE e la tabella ACCOUNT ;
- Esiste un vincolo di integrità referenziale tra l'attributo CodFiscale della tabella SERATA TEMA PROPOSTA e la tabella ACCOUNT ;
- Esiste un vincolo di integrità referenziale tra l'attributo CodFiscale della tabella VARIANTE e la tabella ACCOUNT ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdRicetta della tabella VARIANTE e la tabella RICETTA ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdVariante della tabella VALUTAZIONE e la tabella VARIANTE ;
- Esiste un vincolo di integrità referenziale tra l'attributo CodFiscale della tabella VALUTAZIONE e la tabella ACCOUNT ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdVariantePiatto della tabella VALUTAZIONE e la tabella VARIANTE PIATTO ;
- Esiste un vincolo di integrità referenziale tra l'attributo CodFiscale della tabella VARIANTE PIATTO e la tabella ACCOUNT ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdSede della tabella QUESTIONARIO e la tabella SEDE ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdQuestionario della tabella DOMANDA e la tabella QUESTIONARIO ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdDomanda della tabella RISPOSTA e la tabella DOMANDA ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdSede della tabella RECENSIONE e la tabella SEDE ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdRecensione della tabella FORMATA e la tabella RECENSIONE ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdRisposta della tabella FORMATA e la tabella RISPOSTA ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdRecensione della tabella VALUTAZIONE RECENSIONE e la tabella RECENSIONE ;

- Esiste un vincolo di integrità referenziale tra l'attributo CodFiscale della tabella RECENSIONE e la tabella ACCOUNT ;
- Esiste un vincolo di integrità referenziale tra l'attributo CodFiscale della tabella VALUTAZIONE RECENSIONE e la tabella ACCOUNT ;
- Esiste un vincolo di integrità referenziale tra l'attributo IdRicetta della tabella QUESTIONARIO e la tabella RICETTA ;

## Tavola dei volumi

Nome	E/R	Numero Istanze	Motivazione
Account	E	5000	Ipotesi iniziale
Adopera	R	2400	In media ogni ricetta adopera 16 ingredienti  $2400 = 150 * 16$
Cambia	R	300	In media ogni variazione modifica 0.2 macchinari/attrezzature  $300 = 1500 * 0.2$
Coinvolge	R	490	In media una fase impiega 0.7 macchinari/attrezzature  $490 = 0.7 * 700$
Comanda	E	60000	Cardinalità (1,1) con “Si Riferisce”
Comanda Take Away	E	45000	Cardinalità (1,1) con “Evade”
Compare nel	R	600	In media una ricetta compare in 4 menu  $600 = 150 * 4$
Confezione	E	16000	Cardinalità (1,1) con “Contiene”
Contiene	R	16000	In media per ogni ingrediente vi sono 80 confezioni  $16000 = 200 * 80$
Decide	R	30000	Ogni account scrive in media 6 valutazioni  $30000 = 5000 * 6$
Deriva	R	150000	In media ogni ricetta viene ordinata 1000 volte ( in un mese )  $150000 = 150 * 1000$

Dispone di	R	40	Ogni sede ha in media 4 magazzini  $40 = 4 * 10$
Divisa in	R	12000	In media ogni ricetta è divisa in 80 fasi  $12000 = 80 * 150$
Domanda	E	31500	Cardinalità (1,1) con “Sottopone”
È assegnato	R	15000	In media ogni tavolo è prenotato 0.5 volte al giorno  $15000 = 1000 * 0.5 * 30$ ( al mese )
È associata	R	30000	In media per ogni recensione vi sono 10 valutazioni  $30000 = 3000 * 10$
Effettua	R	7500	Ogni account in media effettua 1.5 prenotazioni al mese  $7500 = 5000 * 1.5$ ( al mese )
Evade	R	45000	In media ogni pony evade 10 comande ( al giorno )  $45000 = 150 * 10 * 30$
Fase di Preparazione	E	12000	Cardinalità (1,1) con “Divisa in”
Formata	R	45000	Per ogni recensione vi sono in media 15 risposte  $45000 = 3000 * 15$
Funzione	E	350	In media la stessa funzione è svolta da 4 macchinari/attrezzature  $350 = 1400/4$

Genera	R	60000	In media ogni piatto appartiene a 0.4 comande Take Away  $60000 = 150000 * 0.4$
Gestisce	R	150	Ogni sede gestisce in media 15 pony  $150 = 15 * 10$
Giudica	R	7500	Per ogni variante piatto vi sono in media 5 valutazioni  $7500 = 1500 * 5$
Impiega	R	700	Ogni sede impiega in media 70 macchinari/attrezzature  $700 = 70 * 10$
Ingrediente	E	200	Ipotesi iniziale
Interessa	R	750	In media ogni variazione interessa 0.5 ingredienti  $750 = 1500 * 0.5$
Macchinario/Attrezzatura	E	700	Cardinalità (1,1) con "Possiede"
Magazzino	E	40	Cardinalità (1,1) con "Dispone di"
Menu	E	200	Cardinalità (1,1) con "Organizza"
Modifica	R	450	In media ogni variazione modifica 0.3 fasi di preparazione  $450 = 1500 * 0.3$
Muta	R	750	In media ogni ricetta ha 5 varianti  $3500 = 150 * 5$

Opera	R	22500	Ogni account in media scrive 4.5 valutazioni  $22500 = 5000 * 4.5$
Ordina	R	45000	Ogni account in media effettua 0.3 ordini al giorno  $5000 * 30 * 0.3$
Organizza	R	200	Ogni sede ha in media 20 menu  $200 = 20 * 10$
Ottiene	R	157500	In media per ogni domanda vi sono 5 risposte  $157500 = 4500 * 5$
Piatto Ordinato	E	150000	Cardinalità (1,1) con “Deriva”
Pony	E	150	Cardinalità (1,1) con “Gestisce”
Possiede	R	1000	Ogni sede ha in media 100 tavoli  $1000 = 100 * 10$
Prenotazione	E	15000	Cardinalità (1,1) con “È assegnato”
Prevede	R	1500	In media ogni ricetta prevede 10 variazioni  $1500 = 150 * 10$
Produce	R	90000	In media ogni piatto appartiene a 0.6 comande  $90000 = 150000 * 0.6$
Promuove	R	750	Ogni account propone in media 0.15 varianti  $750 = 5000 * 0.15$



Propone	R	2000	Ogni account in media propone 0.4 serate a tema  $2000 = 5000 * 0.4$
Questionario	E	2100	Cardinalità (1,1) con “Redige”
Recensione	E	3000	Cardinalità (1,1) con “Valuta”
Redige	R	2100	Ogni sede in media redige 210 questionari  $2100 = 210 * 10$
Ricetta	E	150	Ipotesi iniziale
Riguarda	R	30000	In media ogni variazione riguarda 0.1 piatti ordinati  $30000 = 150000 * 0.2$
Risposta	E	157500	Cardinalità (1,1) con “Ottiene”
Scrive	R	3000	Ogni account scrive in media 0.6 recensioni  $3000 = 5000 * 0.6$
Sede	E	10	Ipotesi iniziale
Serata a tema Proposta	E	2000	Cardinalità (1,1) con “Propone”
Si Riferisce	R	60000	Ogni tavolo in media produce 2 comande al giorno  $60000 = 1000 * 2 * 30$ ( al mese )
Sottopone	R	31500	Ogni questionario in media ha 15 domande  $31500 = 2100 * 15$

Stima	R	15000	Per ogni variante vi sono in media 10 valutazioni  $15000 = 1500 * 10$
Stocca	R	16000	Ogni magazzino stocca 400 confezioni  $16000 = 400 * 40$
Suggerisce	R	1500	Ogni account suggerisce in media 0.3 varianti piatto  $1500 = 5000 * 0.3$
Svolge	R	1400	In media ogni macchinario/attrezzatura svolge 2 funzioni  $1400 = 700 * 2$
Tavolo	E	1000	Cardinalità (1,1) con “Possiede”
Utilizza	R	2400	In media una fase utilizza 0.2 ingredienti  $2400 = 12000 * 0.2$
Valuta	R	3000	Per ogni sede vi sono in media 300 recensioni  $3000 = 300 * 10$
Valutazione	E	22500	Cardinalità (1,1) con “Opera”
Valutazione Recensione	E	30000	Cardinalità (1,1) con “È associata”
Variante	E	750	Cardinalità (1,1) con “Promuove”
Variante Piatto	E	1500	Cardinalità (1,1) con “Suggerisce”
Variazione	E	1500	Cardinalità (1,1) con “Prevede”

## Ridondanze introdotte

- *Attributo "Bandito" nell'entità Account*

In assenza di ridondanza è possibile verificare che un profilo utente sia bandito attraverso il controllo delle prenotazioni rispettate. Un utente viene bandito dal servizio se, effettuata una prenotazione sul sito web, non si presenta.

- *Attributo "Sede" nell'entità Recensione*

In assenza di ridondanza è possibile verificare la sede a cui una recensione si riferisce ricostruendo il questionario da cui essa deriva.

## Operazioni sulla base di dati

1. Visualizzare il numero di recensioni per ogni sede, specificando il nome della sede stessa.  
Frequenza stimata: 1 volta al mese.
2. Visualizzare la città con più utenti banditi.  
Frequenza stimata: 1 volta al giorno.
3. Aggiornare la ridondanza "Bandito" in Account.  
Frequenza stimata: 2 volte al giorno.
4. Visualizzare l'incasso del mese considerando, per ogni piatto ordinato, il costo della ricetta ed il costo delle singole variazioni ( se presenti ).  
Frequenza stimata: 1 volta al mese.
5. Visualizzare il nome della ricetta più ordinata del mese.  
Frequenza stimata: 1 volta al mese.
6. Visualizzare il numero dei pony disponibili per ogni sede.  
Frequenza stimata: 5 volte al giorno.
7. Visualizzare il nome degli ingredienti con meno di un dato numero di confezioni.  
Frequenza stimata: 5 volte al giorno.
8. Visualizzare il nome della variante di un piatto più apprezzata.  
Frequenza stimata: 1 volta al giorno.

1. Visualizzare il numero di recensioni per ogni sede, specificando il nome della sede stessa.

- In assenza di ridondanza

<b>Numero Operazioni Elementari</b>	<b>Tipo Operazioni</b>	<b>Motivazione</b>
45000	Lettura	Leggo tutte le associazioni tra Risposta e Recensione, attraverso la tabella Formata
45000	Lettura	Associo alle risposte le relative domande
45000	Lettura	Associo le domande al relativo questionario
45000	Lettura	Associo i questionari alle relative sedi
10	Lettura	Associo alle tuple trovate il relativo nome
$45000 * 4 + 10 = 180010$	<b>Numero operazioni elementari per interrogazione</b>	
$180010/30 \sim 6000$	<b>Numero operazioni elementari al giorno</b>	

- In presenza di ridondanza

<b>Numero Operazioni Elementari</b>	<b>Tipo Operazioni</b>	<b>Motivazione</b>
3000	Lettura	Leggo tutte le recensioni
10	Lettura	Associo le recensioni alle relative sedi
3010	<b>Numero operazioni elementari per interrogazione</b>	
$3010/30 = 100$	<b>Numero operazioni elementari al giorno</b>	

2. Visualizzare la città con più utenti banditi.

- In assenza di ridondanza

<b>Numero Operazioni Elementari</b>	<b>Tipo Operazioni</b>	<b>Motivazione</b>
7500	Lettura	Leggo tutte le tuple di Effettua
7500	Lettura	Associo Effettua con Prenotazione per trovare le prenotazioni non rispettate
100	Lettura	Associo le tuple trovate al rispettivo account
15100	<b>Numero operazioni elementari per interrogazione</b>	
15100	<b>Numero operazioni elementari al giorno</b>	

- In presenza di ridondanza

<b>Numero Operazioni Elementari</b>	<b>Tipo Operazioni</b>	<b>Motivazione</b>
5000	Lettura	Leggo tutte le tuple di Account
5000	<b>Numero operazioni elementari per interrogazione</b>	
5000	<b>Numero operazioni elementari al giorno</b>	

3. Aggiornare la ridondanza "Bandito" in Account.

<b>Numero Operazioni Elementari</b>	<b>Tipo Operazioni</b>	<b>Motivazione</b>
7500	Lettura	Leggo tutte le associazioni tra Account e Prenotazione, attraverso la tabella Effettua
7500	Lettura	Controllo in Prenotazione quali tuple rispettano il criterio di esclusione
$100 * 2 = 200$	Scrittura	Aggiorno l'attributo bandito in Account
15200	<b>Numero operazioni elementari per interrogazione</b>	
$15200 * 2 = 30400$	<b>Numero operazioni elementari al giorno</b>	

4. Visualizzare l'incasso del mese considerando, per ogni piatto ordinato, il costo della ricetta ed il costo delle singole variazioni ( se presenti ).

<b>Numero Operazioni Elementari</b>	<b>Tipo Operazioni</b>	<b>Motivazione</b>
30000	Lettura	Leggo tutte le tuple da Riguarda
1500	Lettura	Associo le tuple trovate con Variazione per trovare il prezzo
150000	Lettura	Leggo tutte le tuple da Deriva
150	Lettura	Associo tutte le tuple trovate con Ricetta per trovare il prezzo
181650	<b>Numero operazioni elementari per interrogazione</b>	
$181650/30 = 6055$	<b>Numero operazioni elementari al giorno</b>	

5. Visualizzare il nome della ricetta più ordinata del mese.

<b>Numero Operazioni Elementari</b>	<b>Tipo Operazioni</b>	<b>Motivazione</b>
150000	Lettura	Leggo tutte le tuple da Deriva
1	Lettura	Associo la tupla trovata (supposta unica) a Ricetta per trovarne il nome
150001	<b>Numero operazioni elementari per interrogazione</b>	
$150001/30 = 500$	<b>Numero operazioni elementari al giorno</b>	

6. Visualizzare il numero dei pony disponibili per ogni sede.

<b>Numero Operazioni Elementari</b>	<b>Tipo Operazioni</b>	<b>Motivazione</b>
45000	Lettura	Leggo tutte le tuple da Comanda Take Away e trovo quelle il cui pony non è rientrato
150	Lettura	Associo le tuple trovate con i relativi pony, attraverso la tabella Evade
150	Lettura	Associo i pony alle rispettive sedi, attraverso la tabella Gestisce
45300	<b>Numero operazioni elementari per interrogazione</b>	
$45300 * 5 = 226500$	<b>Numero operazioni elementari al giorno</b>	

7. Visualizzare il nome degli ingredienti con meno di un dato numero di confezioni.

<b>Numero Operazioni Elementari</b>	<b>Tipo Operazioni</b>	<b>Motivazione</b>
16000	Lettura	Leggo le associazioni fra Ingrediente e Confezione, attraverso la tabella Contiene
200	Lettura	Associo alle tuple trovate il nome dell'ingrediente
16200	<b>Numero operazioni elementari per interrogazione</b>	
$5 * 16200 = 81000$	<b>Numero operazioni elementari al giorno</b>	

8. Visualizzare il nome della variante di un piatto più apprezzata.

<b>Numero Operazioni Elementari</b>	<b>Tipo Operazioni</b>	<b>Motivazione</b>
7500	Lettura	Leggo tutte le associazioni fra Valutazione e Variante Piatto, attraverso la tabella Giudica
7500	Lettura	Associo ad ogni tupla in Variante Piatto il suo punteggio
1	Lettura	Associo alla tupla con più occorrenze ( supposta unica ) il relativo nome
15001	<b>Numero operazioni elementari per interrogazione</b>	
15001	<b>Numero operazioni elementari al giorno</b>	



## Operazioni in SQL e sintassi MySQL

1. Visualizzare il numero di recensioni per ogni sede, specificando il nome della sede stessa.

- In assenza di ridondanza

```
SELECT D1.IdSede, COUNT(*) AS NumeroRecensioni
FROM ( SELECT Q. IdSede
        FROM Sede S NATURAL JOIN Formato F
        NATURAL JOIN Recensione R
        INNER JOIN Domanda D ON ( R.Domanda = D.IdDomanda )
        INNER JOIN Questionario Q ON ( D.questionario = Q.IdQuestionario )
      ) AS D1
GROUP BY D1.IdSede ;
```

- In presenza di ridondanza

```
SELECT S.Nome, COUNT(*) AS NumeroRecensioni
FROM Recensione R INNER JOIN Sede S ON ( R.Sede = S.IdSede )
GROUP BY S.IdSede ;
```

2. Visualizzare la città con più utenti banditi.

- In assenza di ridondanza

```
CREATE OR REPLACE VIEW UtentiBanditi AS  
SELECT A.CodFiscale  
FROM Account A NATURAL JOIN Prenotazione P  
WHERE P.Rispettata = 0  
GROUP BY A.CodFiscale  
HAVING COUNT(*) >= 2 ;
```

```
CREATE OR REPLACE VIEW UtentiCittaBanditi AS  
SELECT A.Citta, COUNT(*) AS UtentiBanditi  
FROM AccountBanditi A  
GROUP BY A.Citta ;
```

```
SELECT U.Citta  
FROM UtentiCittaBanditi U  
WHERE U.UtentiBanditi = ( SELECT MAX(U1.UtentiBanditi)  
                        FROM UtentiCittaBanditi U1 ) ;
```

- In presenza di ridondanza

```
CREATE OR REPLACE VIEW UtentiCittaBanditi AS  
SELECT A.Citta, COUNT(*) AS UtentiBanditi  
FROM Account A  
WHERE A.Bandito = 1  
GROUP BY A.Citta ;
```

```

SELECT U.Citta
FROM UtentiCittaBanditi U
WHERE U.UtentiBanditi = ( SELECT MAX( U1. UtentiBanditi )
                           FROM UtentiCittaBanditi U1 );

```

3. Aggiornare la ridondanza "Bandito" in Account.

```

UPDATE Account
SET Bandito = 1
WHERE CodFiscale IN ( SELECT *
                      FROM ( SELECT A.CodFiscale
                           FROM Account A NATURAL JOIN Prenotazione P
                           WHERE P.Rispettata = 0
                           GROUP BY A.CodFiscale
                           HAVING COUNT(*) >= 2 ) AS D
                      );

```

4. Visualizzare l'incasso del mese considerando, per ogni piatto ordinato, il costo della ricetta ed il costo delle singole variazioni ( se presenti ).

```

SELECT SUM(R.Prezzo) + SUM( IF( RG.IdVariazione = NULL, 0, V.Prezzo ) )
FROM ( PiattoOrdinato PO INNER JOIN Ricetta R ON ( PO.Piatto = R1.IdRicetta ) )
     NATURAL LEFT OUTER JOIN Riguarda RG
     NATURAL JOIN Variazione V ;

```

5. Visualizzare il nome della ricetta più ordinata del mese.

```
CREATE OR REPLACE VIEW OrdinazioniRicette AS  
SELECT PO.Ricetta, COUNT(*) AS NumeroOrdinazioni  
FROM PiattoOrdinato PO  
GROUP BY PO.Ricetta ;  
  
SELECT R.Nome  
FROM Ricetta R INNER JOIN OrdinazioniRicette OR1 ON ( OR1.Ricetta = R.IdRicetta )  
WHERE OR1. NumeroOrdinazioni = ( SELECT MAX( OR2.NumeroOrdinazioni )  
                                FROM OrdinazioniRicette OR2 ) ;
```

6. Visualizzare il numero dei pony disponibili per ogni sede.

```
SELECT S.Nome, COUNT(*) AS NumeroPony  
FROM Pony P INNER JOIN Sede S ON ( P.IdSede = S.IdSede )  
WHERE P.IdPony NOT IN ( SELECT TA.Pony  
                        FROM ComandaTakeAway TA  
                        WHERE TA.TimestampRientro IS NULL ) ;
```

7. Visualizzare il nome degli ingredienti con meno di un dato numero di confezioni.

```
SELECT I.Nome  
FROM Ingrediente I INNER JOIN Confezione C ON ( C.Ingrediente = I.IdIngrediente )  
GROUP BY I.IdIngrediente  
HAVING COUNT(*) < X ;
```

8. Visualizzare il nome della variante di un piatto più apprezzata.

```
CREATE OR REPLACE VIEW GraduatoriaVariantiPiatto AS  
SELECT    P.IdVariantePiatto, P.Nome, AVG(V.Punteggio) AS ValutazioneMedia  
FROM      VariantePiatto P INNER JOIN Valutazione V ON ( V.Piatto = P.IdVariantePiatto )  
GROUP BY P.IdVariantePiatto ;
```

```
SELECT G1.Nome  
FROM    GraduatoriaVariantiPiatto G1  
WHERE G1.ValutazioneMedia = ( SELECT MAX ( G2. ValutazioneMedia )  
                                FROM    GraduatoriaVariantiPiatto G2 ) ;
```

## Analisi multidimensionale del business

La stored procedure *valutaVariantiPiatto* ha il compito di suggerire alla sede avente le peggiori recensioni, date tutte le varianti piatto, quelle aventi una valutazione media superiore a 8.

```
DROP PROCEDURE IF EXISTS valutaVariantiPiatto ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE valutaVariantiPiatto()
```

```
BEGIN
```

```
    CREATE OR REPLACE VIEW Piatto_Voto AS
```

```
    SELECT VP.IdVariantePiatto
```

```
    FROM VariantePiatto VP INNER JOIN Valutazione V
```

```
        ON ( VP.IdVariantePiatto = V.IdVariantePiatto )
```

```
    GROUP BY VP.IdVariantePiatto
```

```
    HAVING AVG (V.Punteggio) >= 8 ;
```

```
    CREATE OR REPLACE VIEW Sede_Voto AS
```

```
    SELECT S.IdSede AS Sede, AVG(R.GiudizioGlobale) AS Voto
```

```
    FROM Questionario Q INNER JOIN Sede S ON ( S.IdSede = Q.Sede )
```

```
        NATURAL JOIN Recensione R
```

```
    GROUP BY S.IdSede ;
```

```
    SELECT SV1.Sede
```

```
    FROM Sede_Voto SV1 CROSS JOIN Piatto_Voto PV
```

```
    WHERE SV1.Voto = ( SELECT MIN(SV2.Voto)
```

```
        FROM Sede_Voto SV2 ) ;
```

```
END $$
```

```
DELIMITER ;
```

La stored procedure *piattoPiuVenduto* restituisce il piatto più venduto fra tutte le sedi.

```
DROP PROCEDURE IF EXISTS piattoPiuVenduto ;  
DELIMITER $$  
CREATE PROCEDURE piattoPiuVenduto()  
BEGIN  
  
    SELECT PO.IdRicetta, COUNT(*) AS Numero  
  
    FROM    PiattoOrdinato PO  
  
    GROUP BY PO.IdRicetta  
  
    HAVING COUNT(*) >= ALL ( SELECT COUNT(*)  
                                FROM    PiattoOrdinato PO2  
                                GROUP BY PO2.IdRicetta ) ;  
  
END $$  
DELIMITER ;
```

La stored procedure *valutaVarianti* consiglia a ciascuna sede quali varianti inserire in menù. La scelta è effettuata in base alle valutazioni ricevute dagli utenti.

```
DROP PROCEDURE IF EXISTS valutaVarianti ;  
DELIMITER $$  
CREATE PROCEDURE valutaVarianti()  
BEGIN  
  
    CREATE OR REPLACE VIEW Varianti_Voto AS  
  
    SELECT VP.IdVariante, VP.IdRicetta  
  
    FROM    Variante VP NATURAL JOIN Valutazione V
```

```

GROUP BY VP.IdVariante
HAVING AVG(V.Punteggio) >= 8 ;

CREATE OR REPLACE VIEW Sedi_Target AS
SELECT VV.IdVariante, S.IdSede
FROM Sede S NATURAL JOIN Organizza O
      NATURAL JOIN Menu M
      NATURAL JOIN CompareNel CN
      NATURAL JOIN Varianti_Voto VV ;

```

```

END $$

```

```

DELIMITER ;

```

La stored procedure *venditePerSede* restituisce i piatti più ordinati per ciascuna sede.

```

DROP PROCEDURE IF EXISTS venditePerSede;
DELIMITER $$
CREATE PROCEDURE venditePerSede()
BEGIN

      SELECT D.IdSede, D.IdRicetta, COUNT(*) AS N_Ordinazioni
      FROM ( SELECT PO.IdRicetta, T.IdSede
            FROM PiattoOrdinato PO
                  INNER JOIN Comanda C ON ( C.IdComanda = PO.IdComanda )
                  NATURAL JOIN Tavolo T
            UNION ALL
            SELECT PO.IdRicetta, P.IdSede
            FROM PiattoOrdinato PO2
                  INNER JOIN ComandaTakeAway CT

```



```

        ON ( PO2.IdComandaTakeAway = CT.IdComandaTakeAway )
        NATURAL JOIN Pony P
    ) AS D
GROUP BY D.IdSede, D.IdRicetta
ORDER BY D.IdSede ASC, N_ordinazioni DESC ;

END $$
DELIMITER ;

```

La stored procedure *piattoSede* restituisce quante volte un piatto dato è stato ordinato in ciascuna sede.

```

DROP PROCEDURE IF EXISTS piattoSede;

DELIMITER $$

CREATE PROCEDURE piattoSede ( IN _IdPiatto INTEGER )
BEGIN
    SELECT D.IdSede, COUNT(*) AS Numero
    FROM ( SELECT T.IdSede
            FROM PiattoOrdinato PO INNER JOIN Comanda C
            ON ( C.IdComanda = PO.IdComanda )
            NATURAL JOIN Tavolo T
        WHERE PO.IdRicetta = _IdPiatto
        UNION ALL
        SELECT P.idSede
        FROM Piattoordinato PO2 NATURAL JOIN ComandaTakeAway CT
        NATURAL JOIN Pony P
        WHERE PO2.IdRicetta = _IdPiatto ) AS D
    GROUP BY D.IdSede ;

END $$
DELIMITER ;

```

## Magazzino intelligente

La stored procedure suggerisce quali pietanze inserire in menù in base agli ingredienti presenti in magazzino. Al fine di evitare sprechi di ingredienti, si assegna un “peso” a ciascuno di essi in rapporto alle confezioni presenti in magazzino, secondo il seguente criterio:

- peso pari a 3 a tutti gli ingredienti per i quali vi sono delle confezioni aperte ;
- peso pari a 2 a tutti gli ingredienti per i quali le confezioni sono tutte chiuse, ma con scadenza tra meno di 15 giorni ;
- peso pari a 1.5 a tutti gli ingredienti per i quali le confezioni sono tutte chiuse, ma con scadenza tra meno di 30 giorni ;
- peso pari a 1 negli altri casi.

Ad ogni ricetta si associa il voto medio dato dagli utenti, ottenuto mediante una funzione, e tutti gli ingredienti, con relativo peso, che la compongono; infine si calcola la media dei valori ottenuti moltiplicando voti e pesi delle singole ricette.

```
DROP FUNCTION IF EXISTS voto;
```

```
DELIMITER $$
```

```
CREATE FUNCTION voto( _idRicetta INTEGER )
```

```
RETURNS DOUBLE NOT DETERMINISTIC
```

```
BEGIN
```

```
    DECLARE _media DOUBLE;
```

```
    SET _media = ( SELECT ( AVG(RE.GiudizioGlobale)/10 ) + 1
```

```
        FROM Questionario Q NATURAL JOIN Domande D
```

```
        NATURAL JOIN Risposte R
```

```
        NATURAL JOIN Formata F
```

```
        NATURAL JOIN Recensione RE
```

```
    WHERE Q.IdRicetta = _idRicetta ) ;
```

```
    RETURN ( _media ) ;
```

```
END $$
```

```
DELIMITER ;
```

```

DROP PROCEDURE IF EXISTS magazzinoIntelligente ;

DELIMITER $$

CREATE PROCEDURE magazzinoIntelligente()

BEGIN

    DECLARE finito INTEGER DEFAULT 0 ;

    DECLARE id INTEGER ;

    DECLARE peso INTEGER ;


    DECLARE cursore CURSOR FOR

        SELECT C.IdIngrediente, 3 AS Peso                                /* Peso 3 */
        FROM   Confezione C
        WHERE  C.stato = 'parziale'
        GROUP BY C.IdIngrediente
        UNION ALL
        SELECT C.IdIngrediente, 2 AS Peso                                /* Peso 2 */
        FROM   Confezione C
        WHERE  NOT EXISTS( SELECT *
                           FROM   Confezione C3
                           WHERE  C3.IdIngrediente = C.IdIngrediente
                                   AND C3.Stato = 'parziale' )
        AND EXISTS( SELECT *
                   FROM   Confezione C2
                   WHERE  C2.IdIngrediente = C.IdIngrediente AND
                           TIMESTAMPDIFF( DAY, CURRENT_DATE(), C.DataScadenza ) < 15 )
        UNION ALL
        SELECT C.IdIngrediente, 1.5 AS Peso                            /* Peso 1.5 */
        FROM   Confezione C
        WHERE  NOT EXISTS( SELECT *

```

```

        FROM    Confezione C3
    WHERE C3.IdIngrediente = C.IdIngrediente
        AND C3.Stato = 'parziale' OR
TIMESTAMPDIFF( DAY, CURRENT_DATE(), C.DataScadenza ) < 15 )
    AND EXISTS( SELECT *
        FROM    Confezione C2
        WHERE C2.IdIngrediente = C.IdIngrediente AND
TIMESTAMPDIFF( DAY, CURRENT_DATE(), C.DataScadenza ) < 30 )

UNION ALL

SELECT C.IdIngrediente, 1 AS Peso          /* Peso 1 */
FROM    Confezione C
WHERE NOT EXISTS( SELECT *
        FROM    Confezione C3
        WHERE C3.IdIngrediente = C.IdIngrediente
        AND C3.Stato = 'parziale' OR
TIMESTAMPDIFF( DAY, CURRENT_DATE(), C.DataScadenza ) < 30 );

DECLARE CONTINUE HANDLER FOR NOT FOUND

SET finito = 1 ;

DROP TEMPORARY TABLE IF EXISTS piatti_consigliati ;
CREATE TEMPORARY TABLE piatti_consigliati (
    Id INTEGER PRIMARY KEY AUTO_INCREMENT ,
    IdRicetta INTEGER ,
    Voto INTEGER
) ;

OPEN cursore ;

preleva : LOOP

```

**FETCH** cursore **INTO** id, peso ;

**INSERT INTO** piatti\_consigliati ( IdRicetta, Voto )

**SELECT** FP.IdRicetta, peso \* voto(Id)

**FROM** FasePreparazione FP

**WHERE** FP.IdIngrediente = id ;

**END LOOP** preleva ;

**CLOSE** cursore ;

**SELECT** PC.IdRicetta, **AVG**(PC.Voto) **AS** Voto

**FROM** piatti\_consigliati PC

**GROUP BY** PC.IdRicetta

**ORDER BY** Voto **ASC** ;

**END \$\$**

**DELIMITER** ;

## Qualità Take Away

La seguente stored procedure, riguardante la qualità del take away, crea una materialized view contenente, sede per sede, tutti i pony che nell'ultimo mese hanno effettuato consegne in ritardo.

Nello specifico si considera, per ogni sede, la media globale dei tempi di consegna e di rientro: se uno dei due tempi di un dato ordine dovesse sfiorare la media del 20%, esso verrà inserito nella materialized view insieme al codice del pony e al nome della sede per la quale lavora.

```
DROP PROCEDURE IF EXISTS qualitaTA ;
```

```
DROP TABLE IF EXISTS MV_Ritardi ;
```

```
CREATE TABLE MV_Ritardi (
```

```
IdSede INTEGER ,
```

```
IdPony INTEGER ,
```

```
RitardoConsegna INTEGER ,
```

```
RitardoRitorno INTEGER
```

```
) ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE qualitaTA()
```

```
BEGIN
```

```
    DECLARE finito INTEGER DEFAULT 0 ;
```

```
    DECLARE sede INTEGER ;
```

```
    DECLARE pony INTEGER ;
```

```
    DECLARE tempoC INTEGER ;
```

```
    DECLARE tempoR INTEGER ;
```

**DECLARE** cursore **CURSOR FOR**

**SELECT** CTA.IdPony, P.IdSede, CTA.TimestampPartenza, CTA.TimestampConsegna

**FROM** ComandaTakeAway CTA **NATURAL JOIN** Pony P

**WHERE** CTA.TimestampRientro **IS NOT NULL** ;

**DECLARE CONTINUE HANDLER FOR NOT FOUND**

**SET** finito = **1** ;

**CREATE OR REPLACE VIEW** Consegne\_Target **AS**

**SELECT** P.IdSede,

**TIMESTAMPDIFF**(**MINUTE**, CTA.TimestampPartenza, CTA.TimestampConsegna)

**AS** Consegna,

**TIMESTAMPDIFF**(**MINUTE**, CTA.TimestampConsegna, CTA.TimestampRientro)

**AS** Rientro

**FROM** ComandaTakeAway CTA **NATURAL JOIN** Pony P

**WHERE** CTA.TimestampRientro **IS NOT NULL** ;

**CREATE OR REPLACE VIEW** media\_globale **AS**

**SELECT** D.IdSede, **AVG**(D.Consegna) **AS** Consegna, **AVG**(D.Rientro) **AS** Rientro

**FROM** Consegne\_Target D

**GROUP BY** D.IdSede ;

**OPEN** cursore ;

preleva : **LOOP**

**FETCH** cursore **INTO** pony, sede, tempoC, tempoR ;

**IF** ( finito = 1) **THEN**

**LEAVE** preleva ;

**END IF** ;

```

SET @mediaC = ( SELECT MG.Consegna
                FROM   Media_Globale MG
                WHERE  MG.IdSede = sede );

SET @mediaR = ( SELECT MG.rientro
                FROM   Media_Globale MG
                WHERE  MG.IdSede = sede );

IF ( tempoC > @mediaC*1.20 OR tempoR > @mediaR*1.20 ) THEN
    INSERT INTO mv_ritardi
        VALUES( sede, pony, tempoC, tempoR );
END IF ;

END LOOP preleva ;

CLOSE cursore ;

SELECT *
FROM   MV_Ritardi ;

END $$
DELIMITER ;

```