**Progetto Basi Dati**

**1° parte**

Questo progetto si basa sulla gestione ed organizzazione dei *servizi di ristorazione* tramite un’ applicazione che coinvolge: **ristoranti** registrati, i **servizi di delivery** (consegna) disponibili, le **persone** che richiedono questo tipo di servizio e gli **ordini** effettuati, dei quali bisognerà memorizzarne l’effettiva consegna.

Tutti i *ristoranti*, a quali possono essere affidati degli *ordini*possono rendere disponibili uno o più *servizi di delivery*. Quando questi ultimi sono affidati a **società esterne**, viene permessa da parte del *cliente*la valutazione al **rider** che ha consegnato l’ordine; se la *consegna* degli ordini è affidata al ristorante stesso esso dovrà mettere a disposizione i suoi **dipendenti** per l’effettiva consegna.

Data la situazione corrente questa applicazione permette anche di poter tracciare le persone che hanno avuto contatti, usufruendo di questo servizio.

**Analisi dei requisiti (Prima Parte)**

***Sostantivi****:* Ristoranti, Servizi Delivery, Clienti, Ordini, Società, Rider, Dipendenti

***Verbi***: Prenotare, Ordinazione, Assumere, Impiegare, Valutare, Scegliere, Contattare

Per ogni **cliente** vengono memorizzati i *dati anagrafici* e di *contatto*, *data di registrazione* e il *numero di ordini*  effettuati; Ogni cliente può effettuare ordini.

Inoltre, per ogni **ordine** occorre memorizzare

* Il *tipo di ordine* (primo, secondo, menù completo, ecc.) e una *descrizione*;
* Lo *stato* dell’ordine che potrà essere: ordinato, espletato o consegnato;
* Un *numero giornaliero* (si azzera all’inizio di ogni giorno), la *data* e il *ristorante* a cui è affidato l’ordine(indentificano l’ordine stesso);

I **ristoranti** definiscono: *nome*, *indirizzo*, *numero di telefono*, il *numero massimo di prenotazioni* che possono avere in coda e i *servizi di delivery* di cui dispone.

Per i **servizi di delivery** occorre memorizzare un *codice* che lo identifica, una *descrizione*, la *data* in cui lo si inizia a utilizzare, la *cadenza settimanale* (per esempio se è disponibile soltanto nel weekend), ecc. Un servizio di delivery può essere **interno** o **esterno**.

* Per i servizi di delivery **"interni"** vengono assunti dei dipendenti (assegnati alla consegna degli ordini).

Per ogni **dipendente** di questo tipo, tra gli altri, occorre memorizzare gli *anni di esperienza* e uno *short curriculum* dello stesso; l’assunzione effettiva è caratterizzata dal *tipo di contratto* e la *data* di presa di servizio.

* I servizi di delivery **"esterni"** vengono affidati a società di delivery esterne.

Per ogni **società** di questo tipo occorre memorizzare i dati societari, tra cui la *partita IVA*, il *nome della società* e il *nominativo dell’amministratore delegato.*

Inoltre, ogni società di delivery impiega a sua volta diversi Ridera partire da una certa *data* e con una *specifica quota oraria*.

Per ogni **rider**, tra gli altri, occorre memorizzare la *data del suo primo impiego* in assoluto, il *numero di società* per cui lavora attualmente, lo *score medio* ottenuto nelle valutazioni da parte dei clienti e se il raider *è disponibile*. Se un raider è automunito quindi occorre memorizzare il *tipo di veicolo* e la eventuale *targa*.

Ogni **cliente** può valutare uno o più raider, e per essa vengono memorizzati la *data di valutazione* e lo *score* assegnato.

Una ***consegna*** viene effettuata da una persona a cui è affidata la consegna, che può essere un dipendente del ristorante o un rider. Per registrare l’effettiva consegna c’è bisogno dell’*orario di consegna presunto*, *l’orario di consegna effettivo* (quando la consegna è stata conclusa) e il *nominativo di chi ha ritirato l’ordine* (quando la consegna è stata conclusa).

**Analisi Dei Requisiti (Seconda Parte)**

**Sostantivi:**

* Ristoranti [Contatto, Nome, Indirizzo, Servizi]
* Servizi Delivery[Codice, Descrizione, DataUtilizzo, Disponibilità, Consegna]
  + Generalizzazione totalmente esclusiva: Esterno e Interno
* Clienti[CodiceFiscale, Nome, Cognome, Indirizzo, Contatto, #Ordini, DataRegistrazione,Password]
* Ordini[Data, #Giornaliero, Ristorante, Tipo, Descrizione, Stato]
* Dipendenti[CodiceFiscale, AnniEsperienza, Curriculum]
* Società[PartitaIVA, Nome, Amministratore]
* Rider[IDrider, Nome, Cognome, DataPrimoImpiego, #Società, ScoreMedio, Automobile, Disponibilità]

**Attributi composti:**

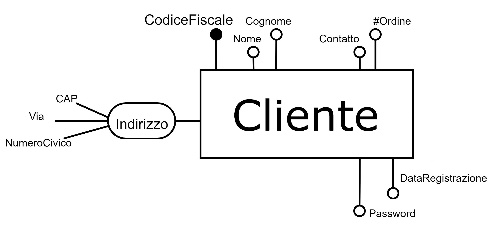
* Indirizzo: Via, CAP, NumeroCivico;
* Automobile: Targa, Tipo;
* Consegna: OrarioPresunto, OrarioEffettivo, NomeClienteRitiro, NominativoFattorino

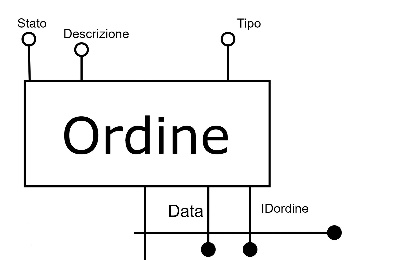
**Verbi:**

* Prenotare [MAXprenotazioni]
* Ordinare
* Assumere [Data, QuotaOraria]
* Impiegare[TipoContratto, DataAssunzione]
* Contattare
* Valutare [Data, ScoreValutazione]
* Scegliere

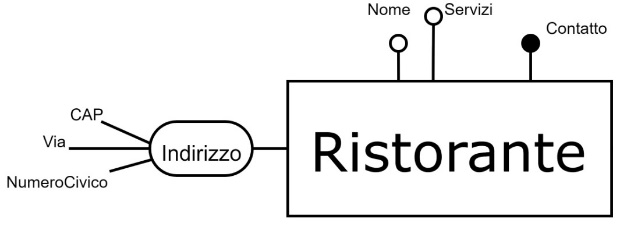
Abbiamo deciso di non specificare il tracciamento relativo ai contatti avuti tra il personale e i clienti poiché sono dati che possono essere facilmente reperibili dall’applicazione; per esempio con il **NominativoFattorino** sappiamo chi ha consegnato l’ordine e con **NomeClienteRitiro** chi l’ha ritirato, quindi si può facilmente contattare il Cliente.

**Descrizione schema entità-relazioni**

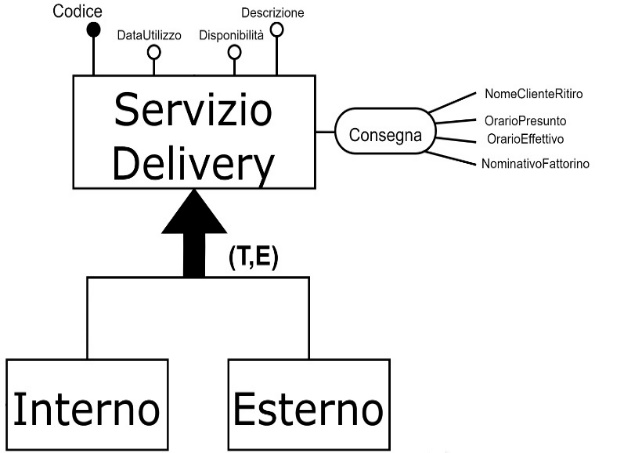
L’entità **Cliente** nello schema rappresenta il cliente dell’applicazione per la gestione del ristorante ed è identificato dal **CodiceFiscale** e come attributi ha anche *Nome*, *Cognome*, *Contatto*, *#Ordini* (numero ordini che ha effettuato), *DataRegistrazione e Password* (aggiunta da noi per effettuare la registrazione alla piattaforma e poter accedere in sicurezza).

L’entità **Ordine** rappresenta l’ordine effettuato dal Cliente che, essendo un’entità, ha i suoi specifici attributi tra i quali:

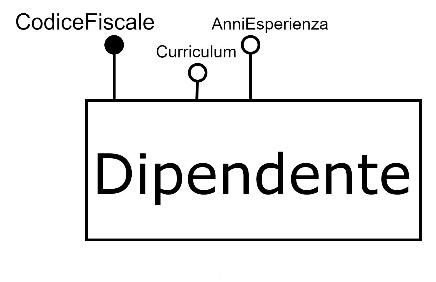
*Tipo* (tipo di ordine), *Stato* (es. consegnato), *Descrizione* e come identificatori **Data**, **#Giornaliero** e il **Ristorante** (a cui viene affidato l’ordine) come attributo esterno.

L’entità **Ristorante** memorizza gli ordini richiesti dai clienti, ed è identificato da **Contatto** ed ha come altri attributi *Nome*, *Servizi* (servizi di delivery di cui dispone).

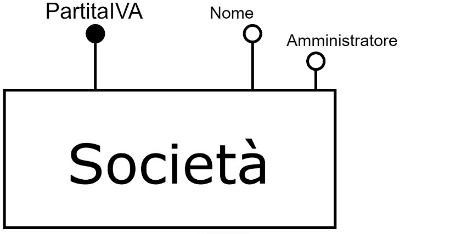
*MAXprenotazioni* ( le prenotazioni che possono avere in coda) abbiamo deciso di inserirlo nell’associazione Prenotazione per gestire la coda degli ordini.

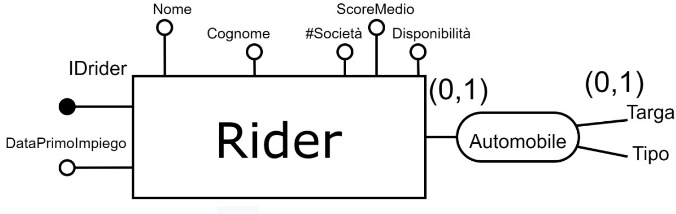
L’entità **Servizio Delivery** è una generalizzazione totalmente esclusiva, poiché si distingue come Interno o esterno. Il servizio Delivery ha come identificatore il **Codice** (è univoco ed identifica un servizio di delivery specifico) e come attributi la *DataUtilizzo* (data in cui si inizia ad utilizzare), *Disponibilità* (cadenza settimanale), *Descrizione* (del servizio) e *Consegna.*

**Consegna** è un attributo composto; ci siamo orientati verso un’ implementazione di questo tipo per ottimizzare la memorizzazione dei suoi attributi che sono: *NomeClienteRitiro* (nominativo di chi ha ritirato l’ordine), *Nominativo Fattorino* (chi consegna l’ordine tra Dipendente e Rider) *OrarioPresunto* e *OrarioEffettivo*.

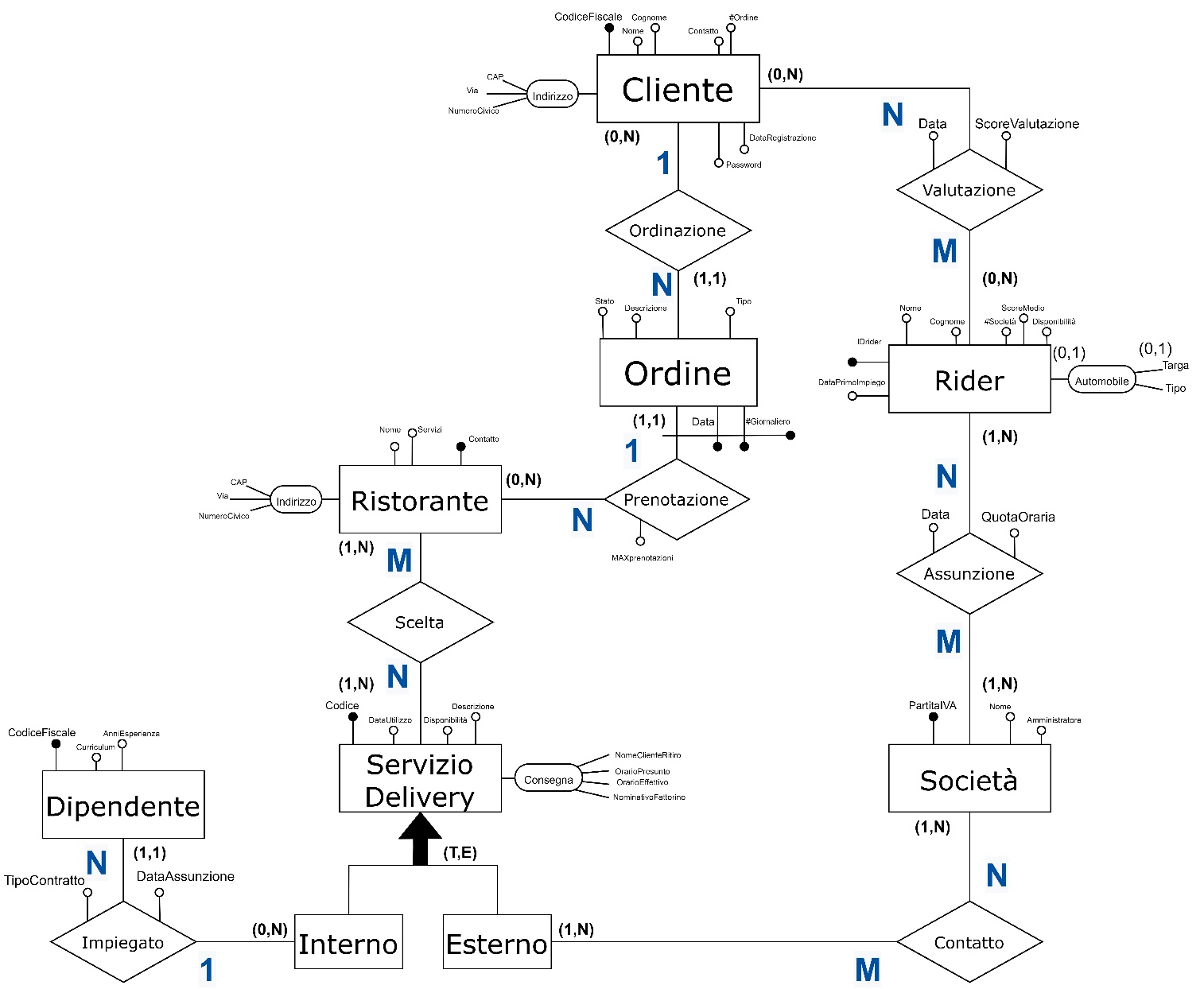


L’entità **Dipendente** descrive il dipendente del Ristorante, il quale si occupa del servizio di delivery interno; Esso viene impiegato, sapendo il *TipoContratto* e la *DataAssunzione* per la consegna degli ordini ed ha come identificatore il **CodiceFiscale** (che abbiamo preferito aggiungere, evitando di inserire nome e cognome) e come attributi semplici il *Curriculum* e *AnniEsperienza*.

L’entità **Società** rappresenta le Società che, in accordo con i ristoranti, offrono il loro contributo per la consegna degli ordini, infatti sono contattate dai Ristoranti e sono dei Servizi di Delivery interni. Hanno come identificatore la **PartitaIVA** e come attributi il *Nome* della società e l’*amministratore* delegato.

**L’entità **Rider** rappresenta le persone assunte dalle società per la consegna in una certa *Data* ed hanno una *QuotaOraria*.

Essi sono caratterizzati dall’ **IDrider** (che è un attributo, aggiunto da noi, che identifica univocamente un rider), *Nome*, *Cognome*, *DataPrimoImpiego*, *#società* (numero società per cui lavora attualmente), *Disponibilità*, *Automobile* (che è un attributo opzionale avente come attributi un *Tipo* ed una *Targa opzionale*) ed ha uno *ScoreMedio* che è il punteggio relativo alla valutazione scritta dagli utenti a consegna effettuata, che ha una *Data* e uno *Score di Valutazione*.

**Schema Entità-Relazioni**

Le cardinalità descritte in questo schema sono:

* **Cliente** (0,N) poiché può ordinare da 0 (minimo) a N (massimo) ordini. Quando un cliente si registra nell’applicazione ipotizziamo che non abbia ancora effettuato ordini;
* **Ordine** (1,1) poiché può avere 1(minimo) a 1 (massimo) cliente;
* La cardinalità tra **Cliente** ed **Ordine** è quindi uno a molti (**1,N**);
* **Ordine** (1,1) poiché può essere associato ad 1 ed 1 solo ristorante;
* **Ristorante** (0,N) poiché può avere da 0 a N ordini. Per esempio, in un giorno un Ristorante potrebbe non ricevere ordinazioni;
* La cardinalità tra **Ordine** e **Ristorante** è quindi uno a molti (**1,N**);
* **Ristorante** (1,N) può avere minimo 1 e massimo N servizi di delivery;
* **Servizio Delivery** (1,N) può avere da 1 a N ristoranti;
* La cardinalità tra **Ristorante** e **Servizio Delivery** è molti a molti (**M,N**);
* **Interno** (0,N) può avere da 0 a N dipendenti, poiché un ristorante potrebbe non avere Dipendenti addetti alla consegna;
* **Dipendente** (1,1) può essere associato 1 ed 1 solo servizio di delivery interno.
* La cardinalità tra **Interno** e **Dipendente** è uno a molti (**1,N**);
* **Esterno** (1,N) poiché può avere da 1 a N società;
* **Società** (1,N) poiché può essere associata da 1 a N servizi di delivery esterni;
* La cardinalità tra **Esterno** e **Società** è molti a molti (**M,N**);
* **Società** (1,N) poiché può avere da 1 a N Rider;
* **Rider** (1,N) poiché può essere assunto da 1 ad N società. Abbiamo l’attributo #Società che rappresenta il numero della società per cui lavora attualmente il Rider;
* La cardinalità tra **Società** e **Rider** è molti a molti (**M,N**);
* **Rider** (0,N) perché potrebbe essere valutato da 0 a N clienti;
* **Cliente** (0,N) può valutare da 0 a N Rider;
* La cardinalità tra **Rider** e **Cliente** è molti a molti (**M,N**).

**Attributi opzionali**: **Automobile** (0,1) , **Targa** (0,1).

|  |  |  |  |
| --- | --- | --- | --- |
| Termine | Descrizione | Attributi | Collegamenti |
| Cliente | Cliente che richiede un servizio | CodiceFiscale, Nome, Cognome, Indirizzo, Contatto, #Ordini, DataRegistrazione, Password | Ordine, Rider |
| Ordine | Ordine effettuato da un Cliente | Data, #Giornaliero , Ristorante, Stato, Tipo, Descrizione | Ristorante, Cliente |
| Ristorante | Luogo in cui vengono accolte le ordinazioni | Contatto, Nome, Indirizzo, Servizi | Ordine, Servizio Delivery |
| Servizio Delivery | Servizi di consegna forniti dal Ristorante, che si può suddividere in Interno ed Esterno | Codice, Descrizione, DataUtilizzo, Disponibilità, Consegna | Interno, Esterno |
| Interno | Servizio di Delivery interno al Ristorante | (Nessun Attributo) | Dipendente |
| Esterno | Servizio di Delivery esterno al Ristorante | (Nessun Attributo) | Società |
| Dipendente | Dipendente interno al Ristorante | CodiceFiscale, AnniEsperienza, Curriculum | Interno |
| Società | Società esterna al Ristorante alla quale affidare l'ordine | PartitaIVA, Nome, Amministratore | Rider |
| Rider | Dipendente esterno al Ristorante | IDRider, Nome, Cognome, DataPrimoImpiego, #Società, ScoreMedio, Disponibilità, Veicolo | Cliente, Società |
| Ordinazione | Inserisce l’ordine del cliente nella coda degli ordini del Ristorante | (Nessun Attributo) | Ordine, Cliente |
| Prenotazione | Ricezione ordine da parte del cliente | MAXprenotazioni | Cliente, Ordine |
| Scelta | Selezione servizio delivery scelto dal ristorante | (Nessun Attributo) | Ristorante, Servizio Delivery |
| Impiegato | Il ristorante sceglie il dipendente per la consegna | TipoContratto, DataAssunzione | Interno, Dipendente |
| Contatto | Il ristorante sceglie la società esterna alla quale affidare la consegna | (Nessun Attributo) | Esterno, Società |
| Assunzione | La società sceglie un rider a cui far consegnare l’ordine | Data, QuotaOraria | Società, Rider |
| Valutazione | Il cliente valuta il Rider per il servizio ricevuto da esso | DataValutazione, Score | Rider, Cliente |

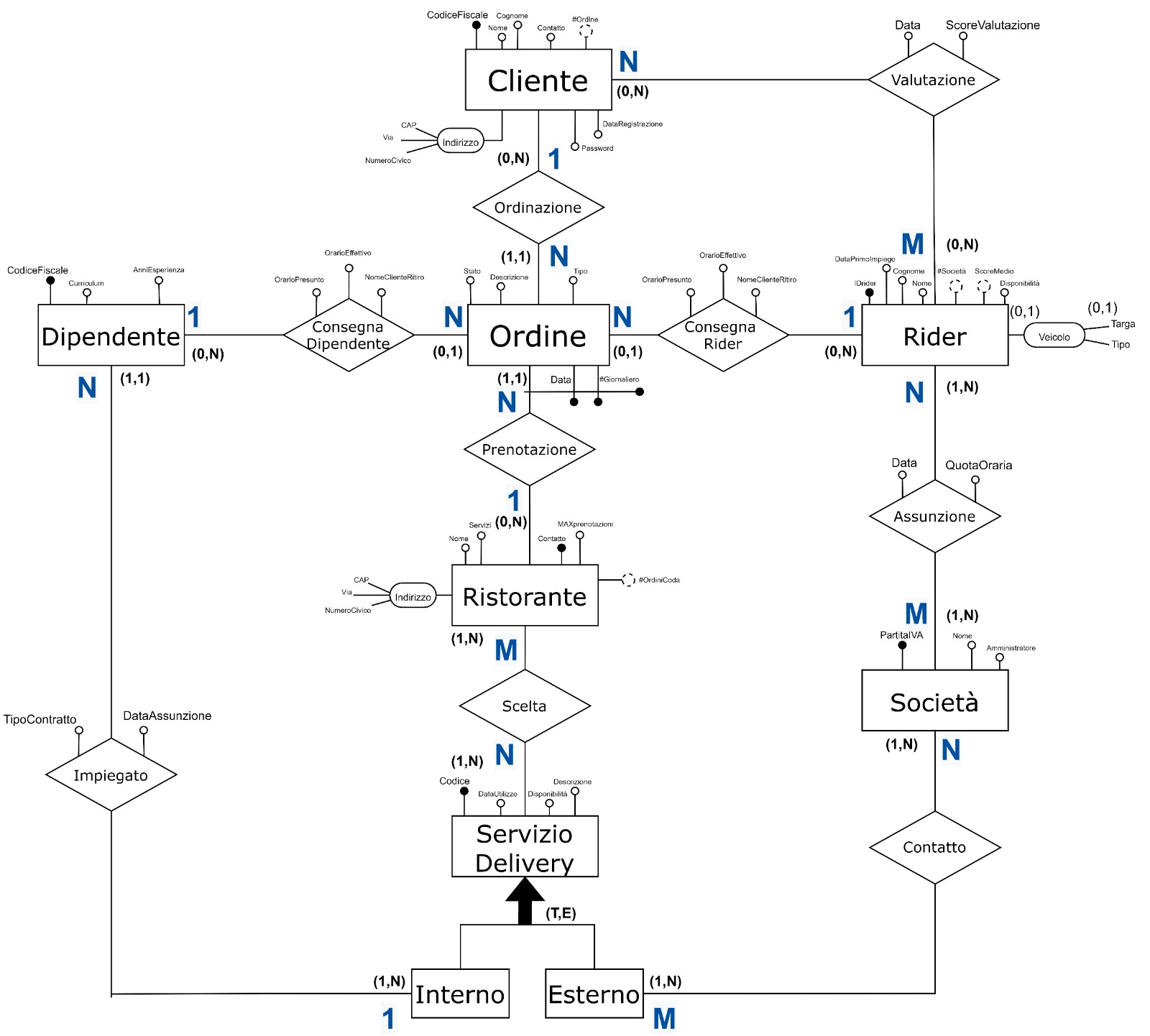
**Glossario dei termini**

**2° parte**

Avendo preso visione della seconda traccia del progetto, lo schema ha subito delle variazioni; in particolare, l’attributo composto **Consegna** è diventato associazione, che è stata divisa in due parti: **Consegna Dipendente** e **Consegna Rider**. Esse sono collegate ad ordine poiché abbiamo bisogno di aggiornare lo stato dell’ordine quando è consegnato.

Sono stati rilevati degli attributi ridondanti ovvero: **#Ordine**, **#Società**, **ScoreMedio** e **#OrdiniCoda**. Quest’ultimo è stato aggiunto per gestire la coda degli ordini del ristorante. Inoltre l’attributo **MAXPrenotazioni**, che avevamo deciso di spostare nell’associazione Prenotazione, è stato rimesso nell’ entità Ristorante.

Infine l’attributo composto Automobile, è stato rinominato in **Veicolo**, poiché ci siamo resi conto che, l’opzionalità dell’attributo **targa** indica che il Rider può avere mezzi senza targa, quindi non solo automobili.

****

|  |  |  |
| --- | --- | --- |
| Concetto | Tipo | Volume |
| Cliente | E | 8.000 |
| Ordine | E | (30\*100\*365) 1.095.000 |
| Ristorante | E | 100 |
| Servizi Delivery | E | 300 |
| Interno | SE | 100 |
| Esterno | SE | 200 |
| Consegna dipendente | R | 328.500 **\*** |
| Consegna rider | R | 766.500 **\*** |
| Dipendente | E | 100 |
| Società | E | 20 |
| Rider | E | 500 |
| Rider Automunito | SE | 250 |
| Valutazione | R | (5\*8.000) 40.000 |
| Assunzione | R | 10.000 |
| Contatto | R | (20\*200) 4.000 |
| Impiegato | R | (500\*3) 15.000 |
| Scelta | R | 30.000 |
| Prenotazione | R | 1.095.000 |
| Ordinazione | R | 1.095.000 |

**Tavola dei Volumi**

**\* N.B:** Supponiamo che le consegne vengano effettuate nel 30% dei casi da un dipendente dei Ristoranti e nel 70% da un Rider delle Società affiliate.

**Principio di Pareto:** Regola empirica secondo la quale un sistema dedica l’80% delle sue risorse per elaborare il 20% delle operazioni più frequenti.

Grazie a questo principio su 25 operazioni più frequenti ne consideriamo 5.

**Operazioni**

**OP1)** Registrazione di un ordine (frequenza deducibile);

**OP2)** Consegna di un ordine (frequenza deducibile);

**OP3)** Valutazione di un rider (frequenza deducibile);

**OP4)** Stampa di un report che mostri i dati dei ristoranti, inclusa la coda di ordini attuale (100/giorno);

**OP5)** Stampa settimanale di un report che mostri i dati dei rider, incluso lo score medio ottenuto nelle valutazioni da parte dei clienti;

**Tavola Delle Operazioni**

|  |  |  |
| --- | --- | --- |
| Operazione | Tipo | Frequenza |
| OP1 | I | 21.000/settimana |
| OP2 | I | 21.000/settimana |
| OP3 | I | 767/settimana |
| OP4 | B | 700/settimana |
| OP5 | B | 1/settimana |

**Op1:**100\*30\*7= **21.000**; **Op2:** 100\*30\*7= **21.000**;

**Op3:** (5/365) \*8000) \*7= **767**; **Op4:** 100\*7= **700**;

**Attributi ridondanti**: **#Ordini, ScoreMedio, #OrdiniCoda**,#Società

Numero società non lo consideriamo poiché nelle operazioni più frequenti non è richiesto.

**Tavola degli accessi**

**#Ordini , ScoreMedio e #OrdiniCoda ridondanti**

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Accesso | Tipo |
| Cliente | E | 1 | L |
| Cliente | E | 1 | S |
| Ordinazione | R | 1 | S |
| Ordine | E | 1 | S |
| Prenotazione | R | 1 | S |
| Ristorante | E | 1 | L |
| Ristorante | E | 1 | S |

**OP1**

Calcolo=2+5(\*2)=12\*21.000=252.000

Ps. Gli accesi in scrittura vengono moltiplicati \*2

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Accesso | Tipo |
| Dipendente | E | 0,3 | L |
| Rider | E | 0,7 | L |
| Consegna Rider | R | 0,7 | L |
| Consegna Dipendente | R | 0,3 | L |
| Ordine | E | 1 | L |
| Ordine | E | 1 | S |

**OP2**

Calcolo=3+1(\*2)= 5\*21.000=105.000

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Accesso | Tipo |
| Valutazione | R | 1 | S |
| Rider | E | 1 | L |
| Rider | E | 1 | S |

**OP3**

Calcolo= (4+1)\*767= 3.835

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Accesso | Tipo |
| Ristorante | E | 100 | L |

**OP4**

Calcolo=100\*700=70.000

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Accesso | Tipo |
| Rider | E | 500 | L |

**OP5**

Calcolo=500

**#Ordini e ScoreMedio ridondanti, #OrdiniCoda non ridondante**

**OP1:** 252.000

**OP2:** 105.000

**OP3:** 3.835

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Accesso | Tipo |
| Prenotazione | R | 3.000 | L |
| Ristorante | E | 100 | L |

**OP4**

Calcolo=(100+3.000)\*700= 2.170.000

**OP5:** 500

**#Ordini e #OrdiniCoda ridondanti, ScoreMedio non ridondante**

**OP1:** 252.000

**OP2:** 105.000

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Accesso | Tipo |
| Valutazione | R | 1 | S |

**OP3**

Calcolo= 2\*767=1.534

**OP4:** 70.000

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Accesso | Tipo |
| Rider | E | 500 | L |
| Valutazione | R | 40.000 | L |
|  |  |  |  |

**OP5**

**Calcolo:** (500+40.000)=40.500

**#Ordini ridondante, ScoreMedio e #OrdiniCoda non ridondanti**

**OP1:** 252.000

**OP2:** 105.000

**OP3:** 1.534

**OP4:** 2.170.000

**OP5**: 40.500

**Senza ridondanza #Ordini, ScoreMedio e #OrdiniCoda ridondanti**

|  |  |  |  |
| --- | --- | --- | --- |
| Concetto | Costrutto | Accesso | Tipo |
| Ordinazione | R | 1 | S |
| Ordine | E | 1 | S |
| Prenotazione | R | 1 | S |
| Ristorante | E | 1 | L |
| Ristorante | E | 1 | S |

**OP1**

**Calcolo:** 9\*21.000= 189.000

**OP2:** 105.000

**OP3:** 3.835

**OP4:** 70.000

**OP5:** 500

**Senza ridondanza #Ordini e #OrdiniCoda, ScoreMedio ridondante**

**OP1:** 189.000

**OP2:** 105.000

**OP3:** 3.835

**OP4:** 2.170.000

**OP5:** 500

**Senza ridondanza #Ordini e ScoreMedio, #OrdiniCoda ridondante**

**OP1:** 189.000

**OP2:** 105.000

**OP3:** 1.534

**OP4:** 70.000

**OP5:** 40.500

**Senza ridondanza in #Ordini, ScoreMedio e #OrdiniCoda**

**OP1:** 189.000

**OP2:** 105.000

**OP3:** 1.534

**OP4:** 2.170.000

**OP5:** 40.500

**Calcolo Carico Applicativo**

**Caso 1:** 252.000 + 105.000 + 3.835 + 70.000 + 500 = 431.335 + 33.400 byte

**Byte:** 2 (\*500)+ 4(\*8000) + 4(\*100)= 33.400 byte

**Caso 2:** 252.000 + 105.000 + 3.835 + 2.170.000 + 500= 2.531.335 + 33.000 byte

**Byte:** 2 (\*500)+ 4(\*8000)= 33.000 byte

**Caso 3:** 252.000 + 105.000 + 1.534 +70.000 + 40.500= 469.034 + 32.400 byte

**Byte:**  4(\*8000) + 4(\*100)= 32.400 byte

**Caso 4:** 252.000 + 105.000 + 1.534 + 2.170.000 + 40.500= 2.569.034 + 32.400 byte

**Byte:**  4(\*8000)= 32.000 byte

**Caso 5:** 189.000+105.000 + 3.835 +70.000 + 500= **368.335 +** 1.400 byte

**Byte:**  2 (\*500) + 4(\*100) = 1.400 byte

**Caso 6:** 189.000+105.000 + 3.835 + 2.170.000 + 500= 2.468.335 + 1.000 byte

**Byte:**  2 (\*500) = 1.000 byte

**Caso 7:** 189.000+105.000 + 1.534 +70.000 +40.500= 406.034 + 400 byte

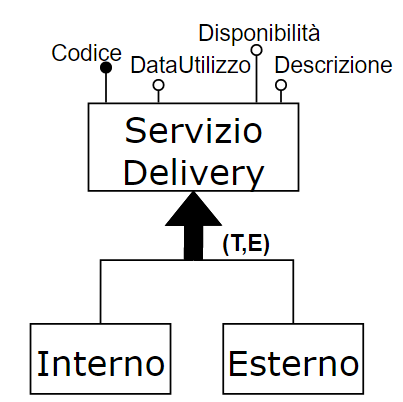
**Byte:**  4(\*100) = 400 byte

**Caso 8:** 189.000+105.000 + 1.534 + 2.170.000 + 40.500= 2.912.068

Ipotizziamo che l’attributo **Score Medio** occupi 2 byte; **#Ordine** e **#OrdiniCoda** 4 byte

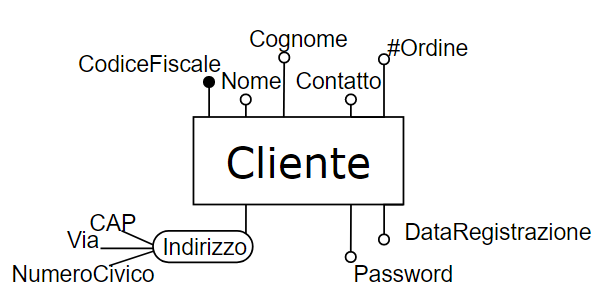
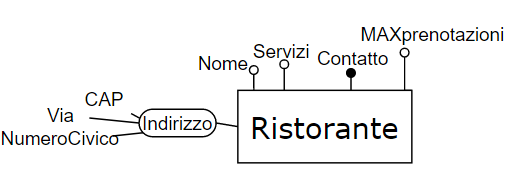
Scegliamo di mantenere ridondante **Score Medio** e **#OrdiniCoda.**

**Se un servizio delivery non è disponibile in “disponibilità” andrebbe, oltre che “festivo” e “feriale” anche “nessuno” e/o “tutti”.**

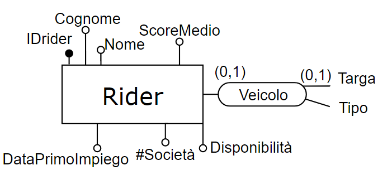
**Ristrutturazione schema**

Analisi gerarchia:

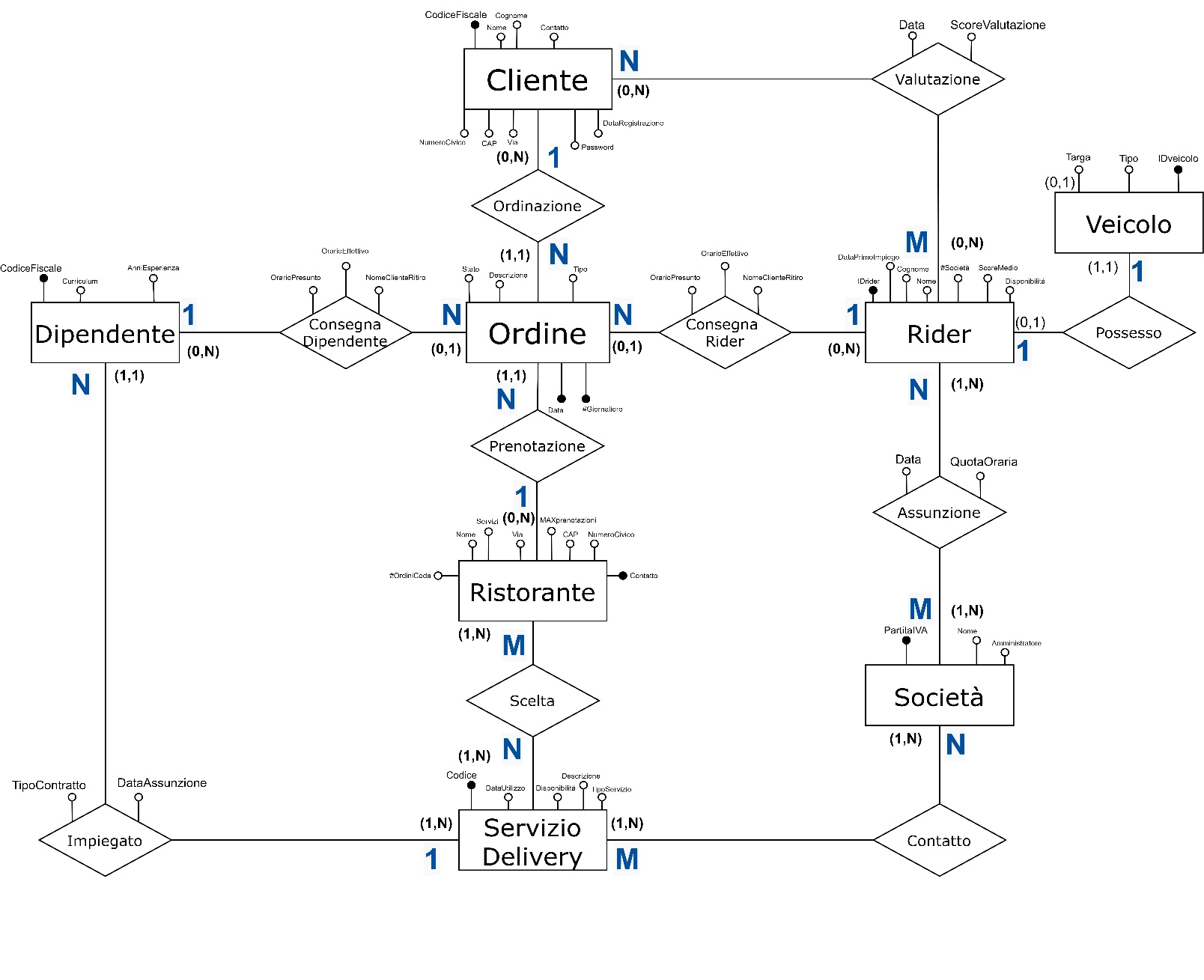
La gerarchia è di tipo **Totalmente Esclusiva**; siccome le entità figlie **NON** hanno attributi e sono collegate singolarmente con un’associazione è preferibile **accorparle** nel padre.

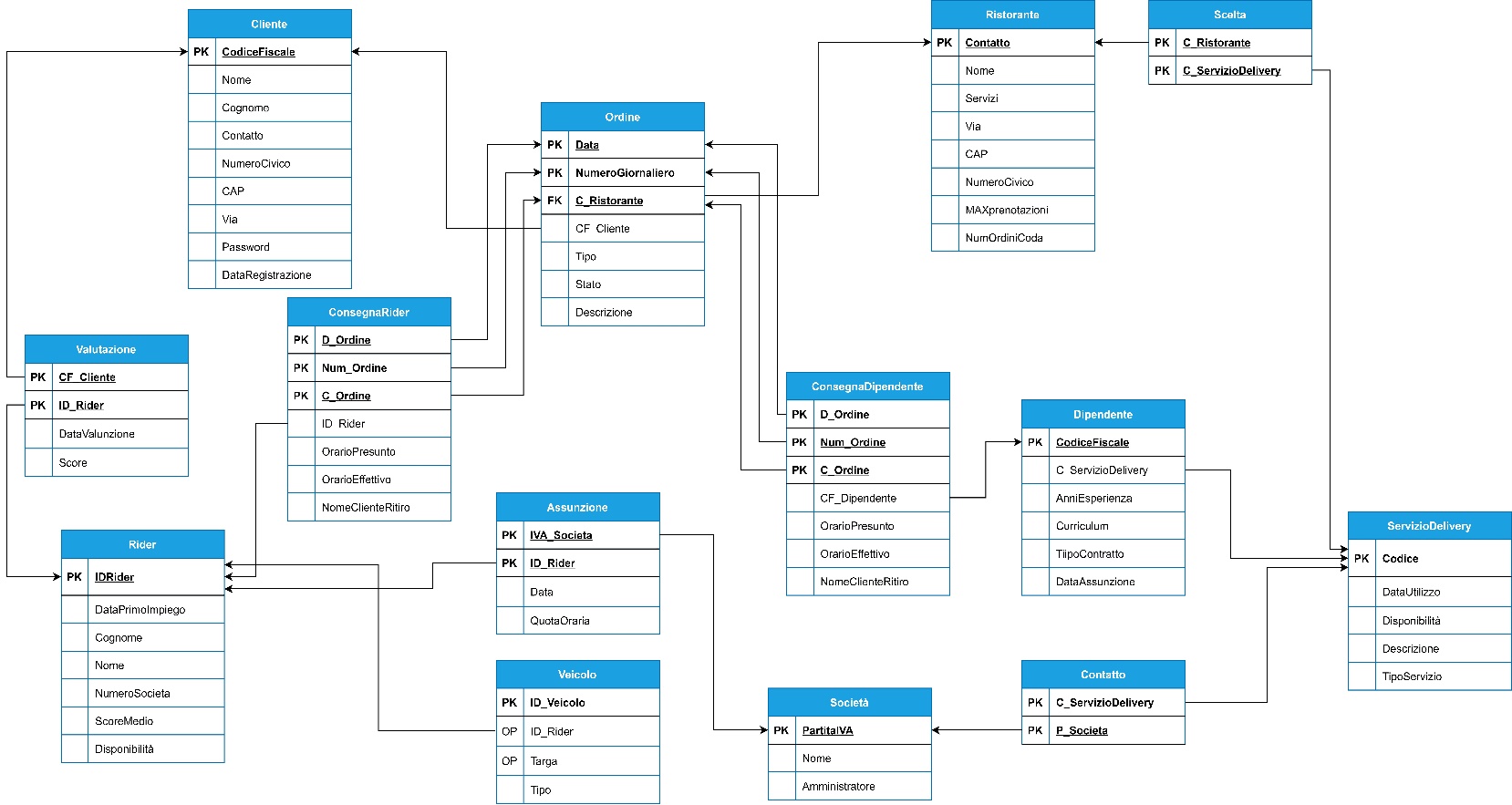
****Analisi attributi ridondanti:

* Per **Ristorante e Cliente**, per evitare di aggiungere rispettivamente un’altra entità e un’associazione, preferiamo **collegare** direttamente gli attributi su di esse;

****

* Per **Rider** abbiamo deciso di creare un’entità separata per memorizzare l’attributo composto **veicolo**.

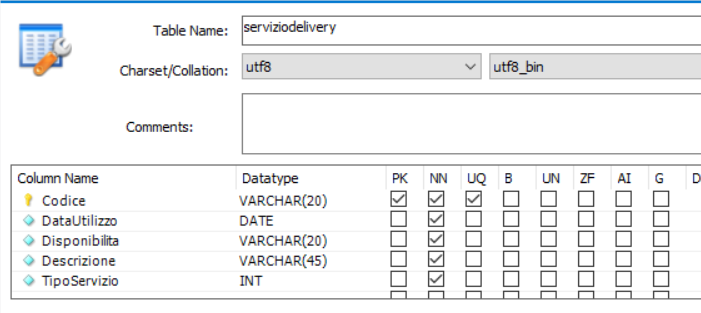
**Schema ristrutturato**

**2Mapping**

**Visto che, dalla ristrutturazione dello schema, togliamo le entità figlie “interno” ed “esterno”, si è impossibilitati a sapere la tipologia di un dato servizio, quindi bisogna aggiungere un attributo “TipoServizio” per identificare la tipologia scelta. Così da semplificare anche l’uso delle query in SQL.**

**3° parte**

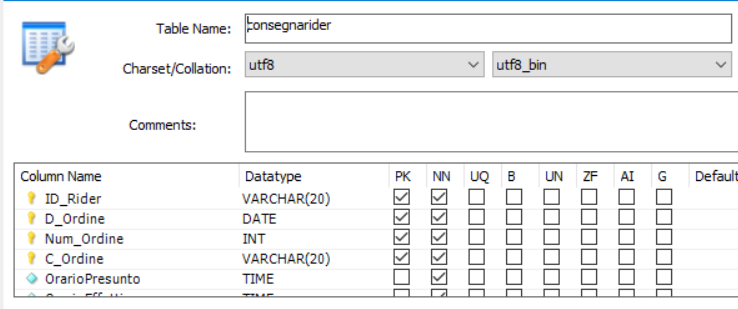
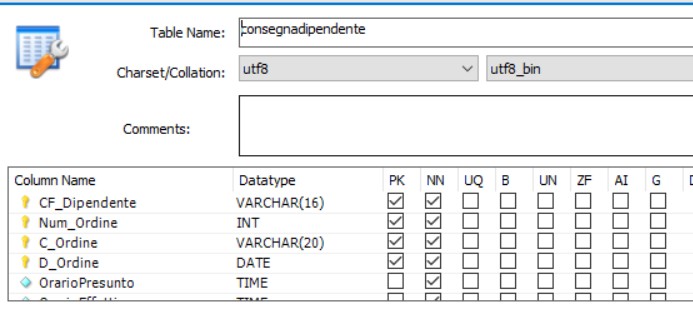
La terza parte del progetto ci avvia verso l’implementazione in SQL. Durante l’inserimento dei dati nel Workbench , abbiamo riscontrato degli errori nell’inserimento di caratteri come “#”. A tal proposito , abbiamo modificato alcuni nomi di attributi nel mapping affinché siano totalmente consoni a quelle del database.



Un'altra modifica che abbiamo effettuato all’interno del database e che abbiamo voluto chiarire è nella tabella ‘**ServizioDelivery’.**

**TipoServizio** è un parametro di tipo INT ed usiamo per identificare di che servizio si tratta(1 se il servizio è esterno e 0 se interno).

n.b. Non siamo riusciti a inserire un tipo booleano o un tipo ‘bit’ siccome con l’utilizzo di questi ultimi si sono riscontrati problemi durante l’implementazione in java.



Un’ultima modifica molto importante che abbiamo attuato, si tratta della modifica di tabelle presenti in **SQL** che corrispondevano ad associazioni nella progettazione **concettuale**(schema E/R).

Nonostante alcuni attributi (principalmente le primary key) siano concettualmente **UNIQUE** , per alcune operazioni, come la **consegna**(rider-dipendente), impostarle in tal modo precluderebbe la **consegna**(in questo caso) ad uno stesso **cliente** da parte di uno stesso **rider** o **dipendente**. Quindi nel pratico risulterebbe impossibile effettuare la maggior parte delle operazioni presenti nella terza parte del progetto.