

# Base di dati per Magazzino di registratori di cassa

A cura di Federico Gibertoni, Federico Scaltriti e Lorenzo Storchi

<b>Descrizione del caso di studio</b>	<b>2</b>
IPOTESI AGGIUNTIVE	3
GLOSSARIO	4
<b>Progetto concettuale-Schema scheletro e diagramma E/R</b>	<b>5</b>
SCHEMA SCHELETRO	5
SCHEMA E/R	7
GERARCHIA	9
IDENTIFICAZIONI ESTERNE	10
DATO DERIVATO	13
<b>Progetto logico - Schema relazionale</b>	<b>15</b>
ELIMINAZIONE GERARCHIE ISA	15
SELEZIONE CHIAVI PRIMARIE ED ELIMINAZIONE IDENTIFICAZIONI ESTERNE	15
TRASFORMAZIONE ATTRIBUTI COMPOSTI/MULTIPLI	16
TRADUZIONI DELLE ENTITÀ IN SCHEMA DI RELAZIONI	16
VERIFICA DELLA NORMALIZZAZIONE	17
PROGETTO LOGICO COMPLESSIVO	17
<b>Operazioni previste dalla base di dati</b>	<b>19</b>
<b>Codice SQL per la creazione delle tabelle e per l'inserimento dei dati</b>	<b>24</b>
CREAZIONE DATABASE E TABELLE	24
QUERY DI INSERIMENTO DEI DATI	29
QUERY DI MODIFICA DEI DATI	35
QUERY DI ELIMINAZIONE DEI DATI	35
CREAZIONE DI TRIGGER	36

## A. Descrizione del caso di studio

Il progetto si ispira a un problema reale: la gestione di un magazzino adibito alla ricezione e alla vendita di registratori di cassa.

L'obiettivo è quello di aiutare un magazzino nella gestione delle proprie attività, come l'arrivo e la spedizione dei prodotti ma anche la gestione dei suoi dipendenti, con i relativi ruoli.

Si necessita quindi di una base di dati che implementi le seguenti tematiche:

### **DIPENDENTI**

I dipendenti sono una caratteristica fondamentale della base di dati in quanto svolgono il lavoro all'interno del magazzino. Conoscere le mansioni e le attività dei singoli dipendenti è un aspetto importante. È quindi opportuno identificarli efficacemente tramite una matricola univoca.

I dipendenti hanno tutti almeno un ruolo che può essere magazziniere, autista o impiegato.

Il magazzino è diviso in reparti dove i dipendenti compiono il proprio lavoro.

I dipendenti, in quanto tali, devono essere retribuiti. Il compenso è definito dal tipo di contratto che hanno stipulato con l'azienda, che può variare da persona a persona.

Un compito importante riguardante un dipendente è quello delle consegne. A questo riguardo è prevista una gestione delle patenti, per una selezione efficace degli autisti, e uno storico delle guide effettuate per la spedizione di ordini.

### **REGISTRATORI**

I registratori di cassa sono i prodotti specifici di cui si occupa il magazzino preso in considerazione. Il magazzino presenta un'ampia gamma di modelli che possono essere gestiti. È quindi opportuno memorizzarne caratteristiche ulteriori quali il peso e la dimensione occupata.

Di conseguenza si rende necessario memorizzare le ditte che riforniscono il magazzino, attraverso l'entità Fornitore. A livello di magazzino è importante sapere solo alcune delle caratteristiche di queste aziende, come la partita IVA, per identificarle univocamente, o il contatto interno all'azienda, necessario in caso sorgano problematiche nelle consegne dei registratori di cassa.

È importante sapere dove ogni registratore è allocato. A questo scopo è stata creata un'entità Scaffale contenente tale informazione.

### **ORDINI**

Un ordine è il modo che un cliente ha per poter acquistare i registratori di cassa forniti dal magazzino.

Di un determinato ordine ci interessa sapere in particolare la quantità di registratori che sono stati acquistati di un determinato modello, quali clienti hanno acquistato i

registratori e anche a quale prezzo li hanno acquistati; ogni cliente ha una sua P.IVA, un suo nome, un contatto e un ambito in cui lavora, informazione che può tornare utile a livello statistico o nelle scelte future relative ai modelli di registratori da ricevere dai fornitori.

È importante sapere il tipo di registratore acquistato associato ad ogni ordine effettuato (con la relativa quantità). A questo proposito, viene memorizzata anche la fattura emessa relativa a un ordine, che conterrà il suo numero identificativo, la data di emissione, la descrizione dell'ordine e la spesa totale dovuta al nostro magazzino. Ogni ordine sarà poi spedito verso il cliente richiedente.

## **SPEDIZIONE**

Di ogni spedizione è interessante tracciare la data prevista per la consegna, un identificatore univoco per la singola spedizione e lo stato della consegna, che può variare tra tre diversi valori: "In preparazione", che indica che l'ordine è stato ricevuto ed è in fase di preparazione per la partenza; "In transito", che significa che l'ordine è partito e arriverà al cliente che ha fatto l'acquisto entro la data prevista; "Consegnato", indicante che l'ordine è arrivato a destinazione.

Ogni spedizione è effettuata attraverso i veicoli a disposizione del magazzino, adibiti al trasporto degli ordini. In base all'ordine da consegnare, una spedizione potrà essere effettuata più facilmente da un camion piuttosto che da un'auto utilizzata dai dipendenti.

Si suppone che una singola spedizione occupi tutta una giornata lavorativa e quindi un certo veicolo, in una certa giornata sia guidato solamente dal dipendente che è stato scelto per quella giornata.

## **IPOTESI AGGIUNTIVE**

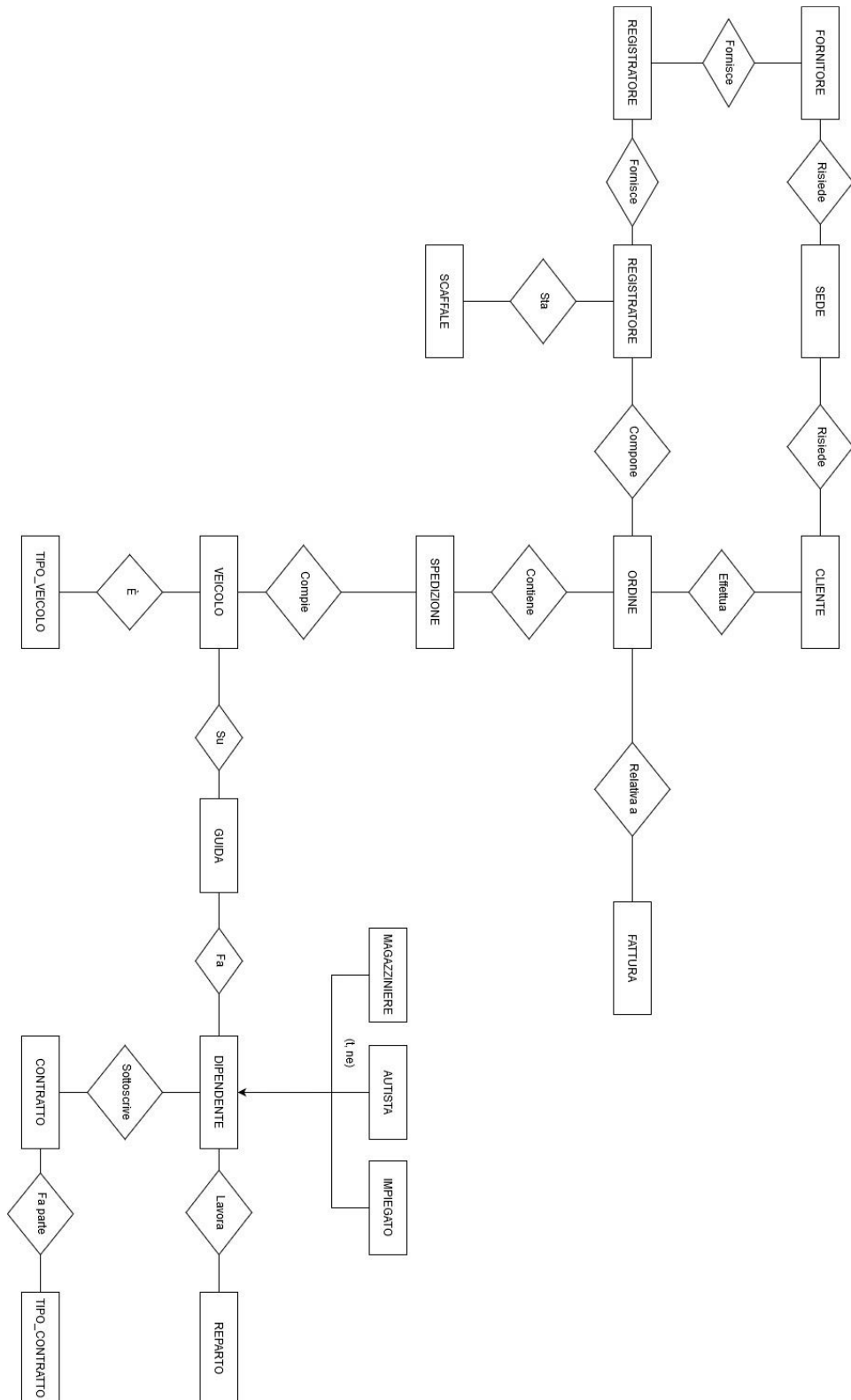
- Si trascura la reale presenza di vie e civici specificati nell'entità SEDE;
- Le aziende citate sono solo a titolo esemplificativo: si escludono relazioni tra questo progetto e le aziende in questione;
- Si trascurano i valori reali dei numeri delle partite IVA di clienti e fornitori, in quanto è stato rispettato solamente il vincolo che prevede la presenza di 11 cifre, e dei loro contatti;
- Si è sorvolato sulle effettive corrispondenze tra le targhe riportate e i tipi di veicoli, come tra i dati dei dipendenti e i codici fiscali associati;
- Si considera un sistema di patenti in cui ogni patente posseduta implichi di avere anche tutte quelle di "grado inferiore" automaticamente. (es. se un autista ha la patente C allora possiederà anche la patente B; se un autista ha l'estensione E allora la avrà per tutte le patenti per cui è disponibile);
- Per quanto riguarda la fornitura di un modello di registratore si è pensato di porre il limite, ad ogni fornitore, di fornire al massimo un modello dal giorno.

## GLOSSARIO

TERMINE	DESCRIZIONE	SINONIMI	LEGAME
Dipendente	<u>Matricola</u> ; Nome; Cognome; Data_assunzione; Data_nascita;	Impiegato, Operaio	Contratto; Reparto; Guida
Magazziniere	Patente_carrello_elevatore		Dipendente
Autista	Patente		Dipendente
Impiegato	CF		Dipendente
Contratto	<u>ID</u> ; Stipendio; Orario		Dipendente; Tipo_contratto
Reparto	<u>Nome</u> ; Descrizione;		Dipendente;
Fornitore	<u>P_IVA</u> ; Nome; Contatto		Registratore; Sede
Cliente	<u>P_IVA</u> ; Nome; Contatto; Ambito Lavorativo		Ordine; Sede
Registratore	<u>NOME_MODELLO</u> ; Peso; Dimensione; Descrizione		Ordine; Scaffale; Fornitore
Sede	<u>Comune</u> ; <u>Provincia</u> ; CAP; Indirizzo		Fornitore, Cliente
Ordine	<u>NUMERO</u> ; Quantità	Richiesta	Registratore; Cliente; Spedizione; Fattura
Spedizione	<u>ID</u> ; Stato; Data_prevista_consegna;		Ordine; Veicolo
Fattura	<u>NUMERO</u> ; Data; Descrizione; Totale;		Ordine;
Veicolo	<u>TARGA</u>	Vettura	Spedizione; Guida; Tipo_veicolo
Tipo_veicolo	<u>NOME</u> ; Capienza, Peso, Dimensione; Descrizione		Veicolo
Scaffale	<u>ID</u> , Capacità, N_Presenti		Registratore
Tipo_contratto	<u>NOME</u>	Categoria	Contratto
Guida	<u>DATA</u>		Dipendente; Veicolo

## B. Progetto concettuale-Schema scheletro e diagramma E/R

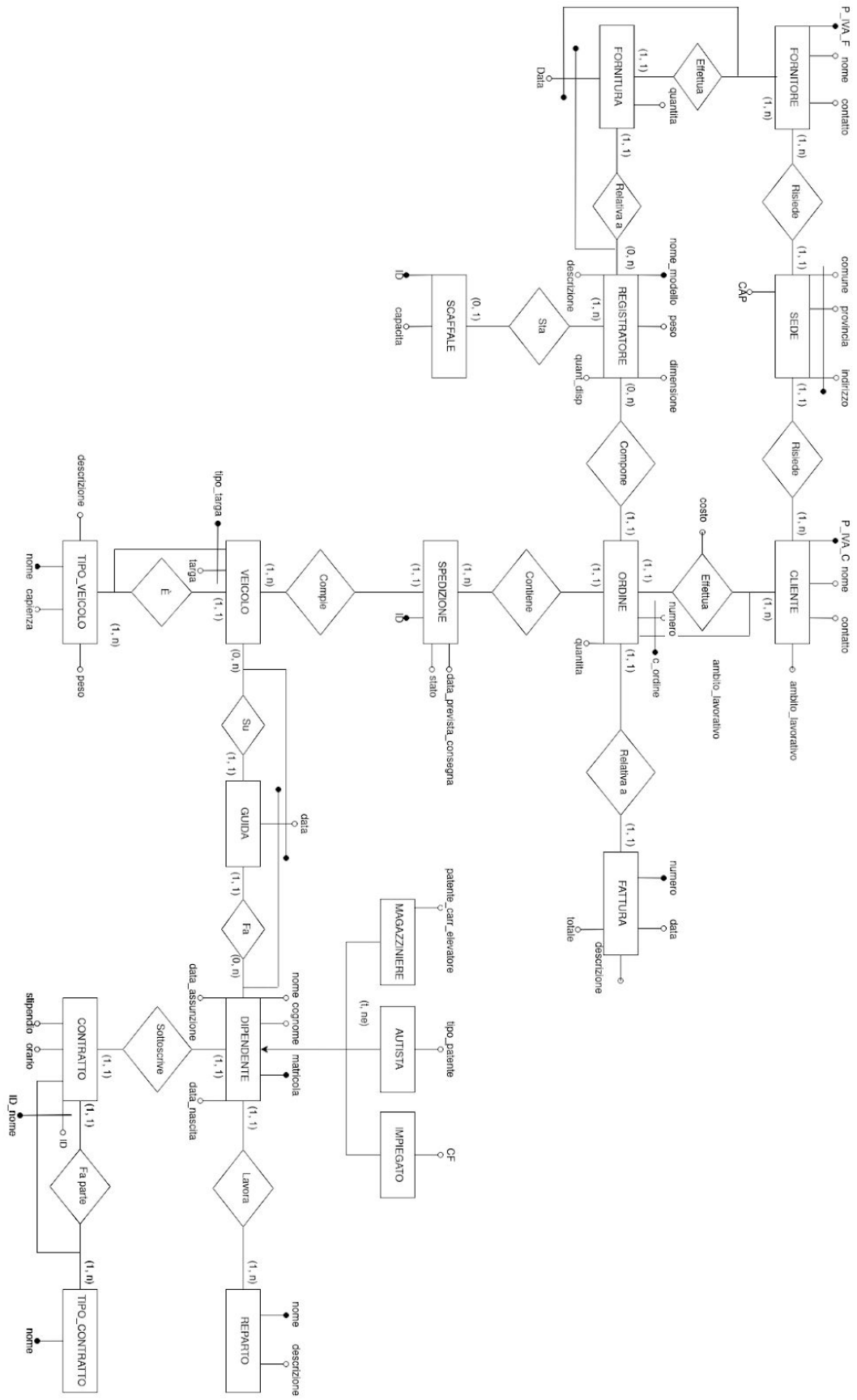
### SCHEMA SCHELETRO



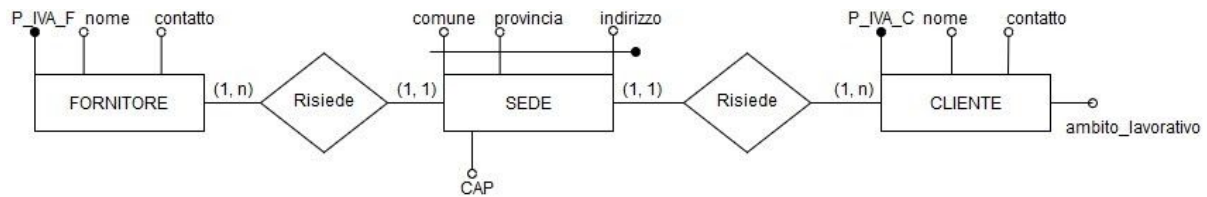
Lo schema scheletro riporta le relazioni logiche che legano i dati da gestire. Viene quindi fatta luce sulla struttura basilare del progetto, in quanto riporta tutte le principali relazioni stabilite dopo aver effettuato una prima raccolta dei dati e dei requisiti del magazzino in questione.

Viene riportato, ad esempio, il legame tra i fornitori e la presenza di registratori all'interno del magazzino, che verranno inseriti in opportuni scaffali, così come la gestione degli ordini che vengono effettuati dai clienti e come vengono consegnati. È data anche una visuale dell'organizzazione aziendale del magazzino nella divisione dei ruoli tra i dipendenti assunti con contratti, e quindi retribuzioni, differenti. È stato successivamente necessario raffinare tali relazioni per rispettare ulteriori vincoli riguardanti, per esempio, richieste più complesse sui dati di spedizioni, auto e ordini, all'interno dello schema E/R.

# SCHEMA E/R

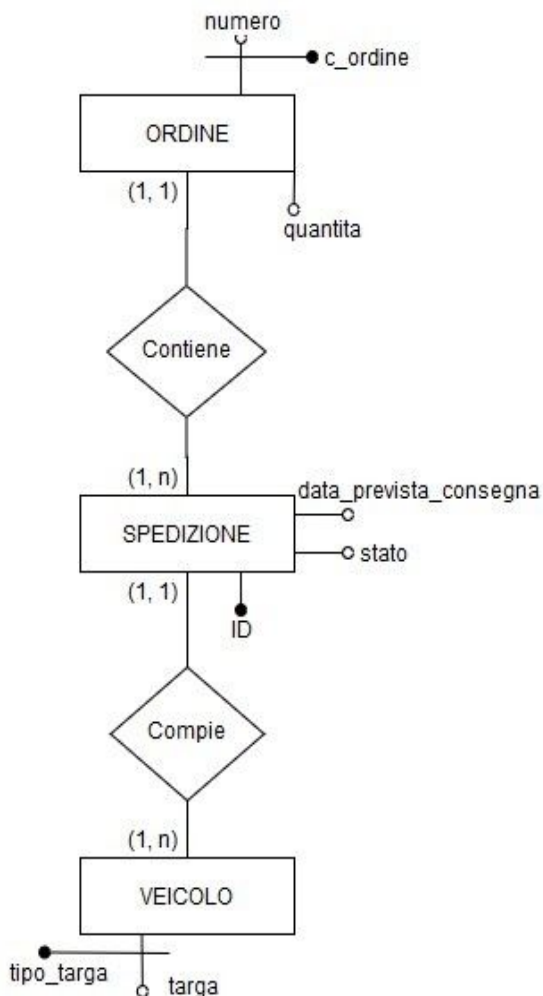


## Entità SEDE



Si è deciso, per una questione di efficienza e praticità, di usare un'unica entità Sede sia per Fornitore che per Cliente. Tale entità è stata pensata in tal maniera perché è impossibile che nella realtà sia un cliente che un fornitore, di un certo tipo di prodotto, abbiano ugual indirizzo fisico della sede. Ogni sede è caratterizzata dalla presenza di un campo nullo, in base al riferimento con Cliente o Fornitore. Se la sede è quella di un fornitore, il campo relativo alla partita IVA del cliente, P\_IVA\_C, avrà valore NULL. Viceversa, in caso di riferimento a un cliente, sarà il campo relativo a P\_IVA\_F a essere nullo.

## Entità SPEDIZIONE



Una volta richiesto un particolare ordine, bisognerà provvedere a spedirlo prontamente. Questo processo è attuato attraverso la presenza dell'entità Spedizione che conterrà al suo interno campi fondamentali come la data\_prevista\_consegna, lo stato attuale della spedizione e un ID identificativo della spedizione. All'interno della singola spedizione possono essere contenuti più ordini, per rendere più efficienti i trasporti, mentre, ovviamente, un singolo ordine potrà essere presente solo all'interno di una spedizione. La composizione della chiave di Ordine è dovuta alla relazione con l'entità Cliente, descritta in uno dei successivi esempi. Lo stesso principio vale per la chiave di Veicolo, legata all'entità Tipo\_Veicolo. Il magazzino gestisce anche un insieme di veicoli, per cui sarà necessario associarli alle spedizioni effettuate. Uno stesso veicolo può essere usato in più momenti differenti, mentre la spedizione specifica verrà effettuata una sola volta. Quindi verrà associata a un solo veicolo.



## GERARCHIA

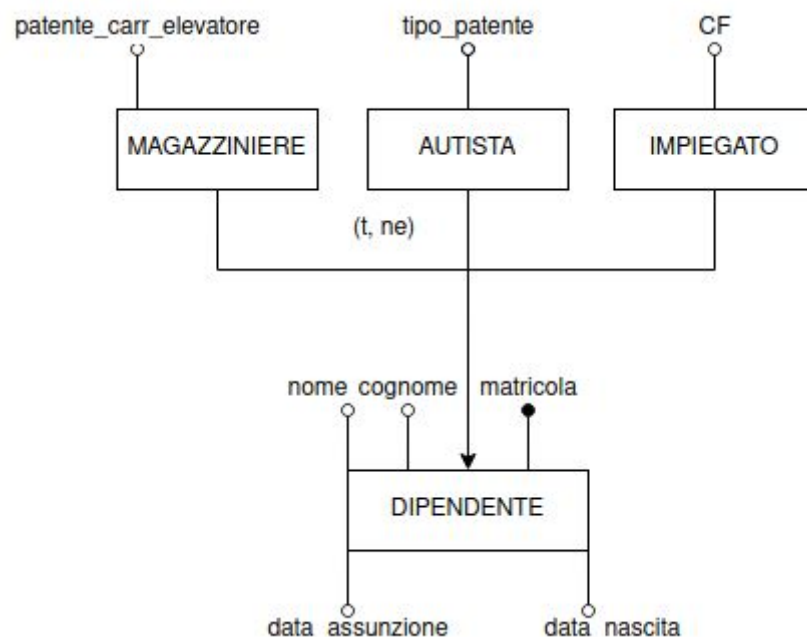
Abbiamo applicato il concetto di gerarchia all'entità DIPENDENTE per distinguere i ruoli che lo interessano a livello informativo all'interno del database.

Il DIPENDENTE può essere:

- un MAGAZZINIERE se si occupa degli spostamenti dei registratori all'interno del magazzino utilizzando il carrello elevatore, si rende infatti necessario memorizzare chi può e chi non può guidarlo;
- un AUTISTA se si occupa delle spedizioni della merce; quindi ci serve sapere quali patenti ha a sua disposizione per capire che cosa può guidare;
- un IMPIEGATO se lavora all'interno degli uffici. In questo caso è sufficiente memorizzare il suo codice fiscale.

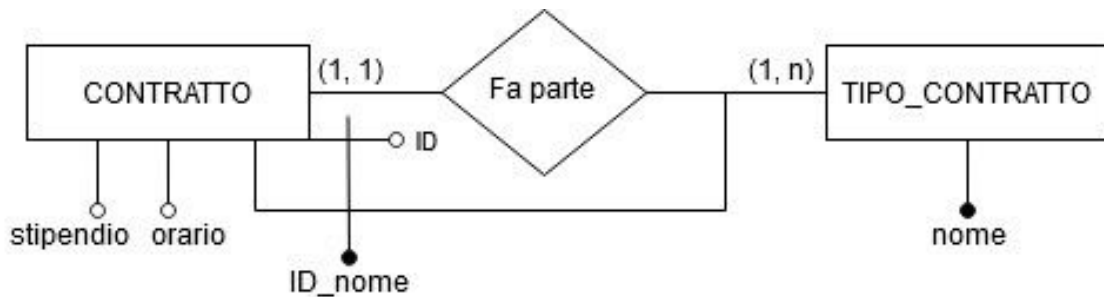
Si tratta di una gerarchia totale (t) perché tutti i dipendenti sono classificati all'interno di queste tre sole categorie e non esistono altre categorie di dipendenti.

La gerarchia inoltre è non esclusiva (ne) in quanto un dipendente può assumere più ruoli nel tempo. Può, ad esempio, essere sia MAGAZZINIERE che AUTISTA.



## IDENTIFICAZIONI ESTERNE

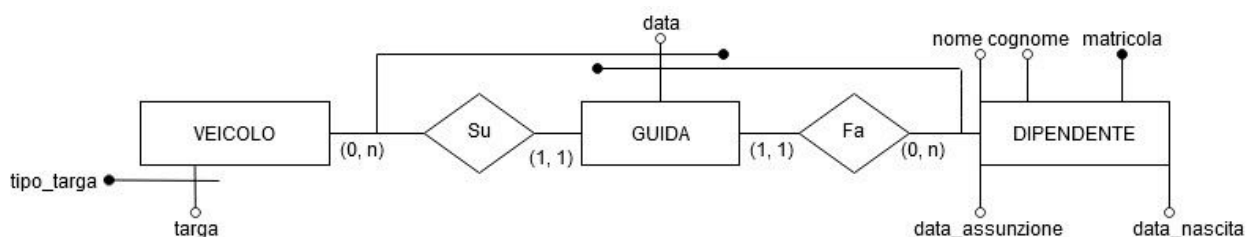
### Esempio 1 - CONTRATTO e TIPO\_CONTRATTO



Abbiamo adottato l'utilizzo di un identificatore esterno vista la dipendenza tra le entità **CONTRATTO** e **TIPO\_CONTRATTO**. Quest'ultima entità contiene valori come "Apprendistato", "Full-time" e "Part-time". Nel caso uno di questi tipi di contratto venisse abolito, o comunque non contemplato, i contratti stipulati fino a quel momento non sarebbero validi e dovrebbero essere eliminati.

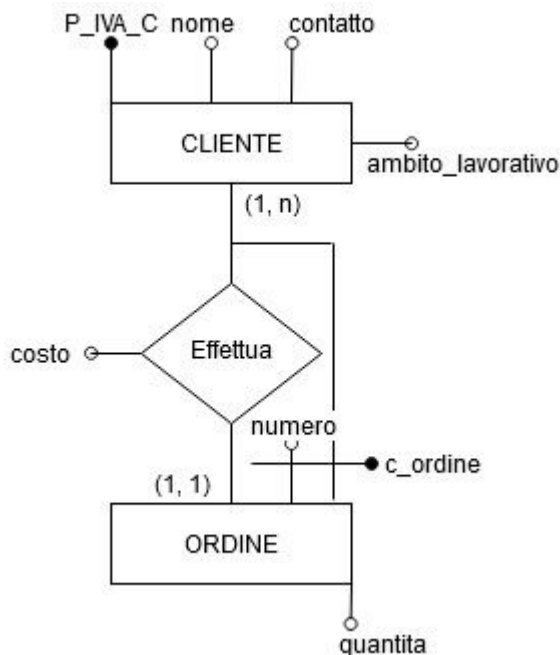
Le cardinalità sono tali per cui un tipo di contratto non è esclusivo a un dipendente. Nulla infatti impedisce di avere più dipendenti con un contratto di tipo "Full-time" o di qualsiasi altro tipo. Un singolo dipendente avrà un solo contratto e quindi una sola categoria.

### Esempio 2 - VEICOLO, GUIDA e DIPENDENTE



Abbiamo reificato l'associazione tra **VEICOLO** e **DIPENDENTE** aggiungendo l'entità **GUIDA** per evitare problemi di incongruenza tra le due entità iniziali. Senza questa specificazione si sarebbe potuto verificare il caso in cui, dato un dipendente e un giorno dell'anno, non sarebbe stato univoco il veicolo da guidare. Inoltre, dato un veicolo e una data, sarebbe stato possibile per più dipendenti guidare uno stesso veicolo durante lo stesso giorno.

### Esempio 3 - CLIENTE e ORDINE



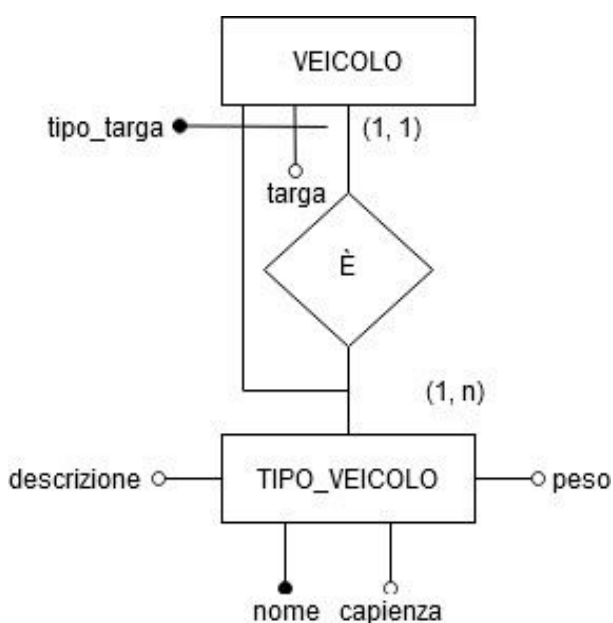
Abbiamo adottato l'utilizzo di un identificatore esterno vista la dipendenza tra le entità CLIENTE e ORDINE.

Quest'ultima entità è vincolata dalla presenza di un cliente perché, nel caso venisse a mancare il cliente, i suoi ordini non dovrebbero essere più presenti.

È stato quindi necessario introdurre un vincolo con chiave esterna per poter gestire questa eventualità.

Le cardinalità sono tali per cui un cliente possa effettuare nell'arco del tempo più ordini ma il singolo ordine è unico per ogni cliente.

### Esempio 4 - VEICOLO e TIPO\_VEICOLO



Abbiamo adottato l'utilizzo di un identificatore esterno vista la dipendenza tra le entità TIPO\_VEICOLO e VEICOLO.

Quest'ultima entità è vincolata dalla presenza della categoria a cui appartiene (per esempio "Ducato", "Transporter"...), perché, nel caso la ditta decidesse di non utilizzare più un determinato tipo di mezzo di trasporto, i veicoli di quel genere non verrebbero più utilizzati.

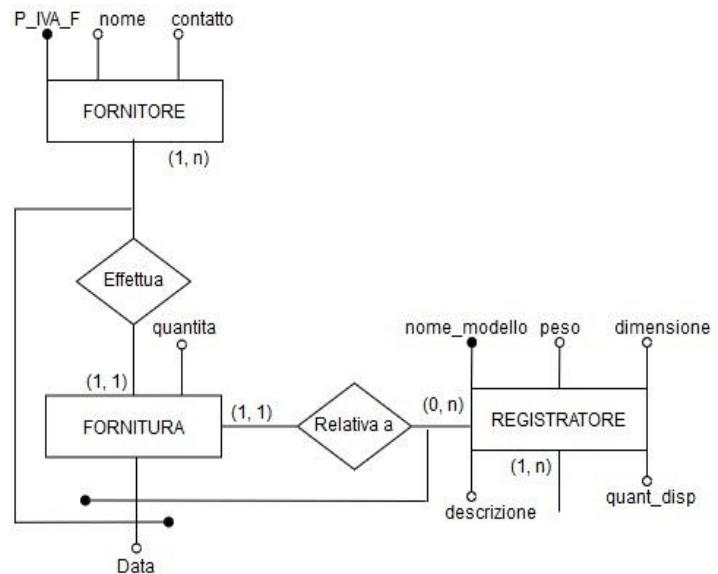
È stato quindi necessario introdurre un vincolo con chiave esterna per poter gestire questa eventualità.

Le cardinalità consentono di avere più veicoli dello stesso tipo ma, ovviamente, un singolo veicolo non può far parte di due categorie.

## Esempio 5 - FORNITORE, FORNITURA e REGISTRATORE

Abbiamo utilizzato una doppia identificazione esterna in Fornitura per assicurarci che un registratore, in una certa data, possa essere consegnato solo da un fornitore, mentre un fornitore in una certa data potrà consegnarci solamente un tipo di registratore.

La scelta di avere una cardinalità (0, n) per un registratore è data dal fatto che il magazzino potrebbe non ricevere nuove unità di un modello di registratore.



## DATO DERIVATO

Abbiamo individuato 3 operazioni frequenti nella base di dati per stabilire se l'aggiunta del dato derivato potessero portare benefici o meno.

Dato derivato: Quantità disponibile come attributo in registratore

Operazione 1: Inserire nuova fornitura

Operazione 2: Visualizzare la quantità disponibile di un registratore

Operazione 3: Inserire un nuovo ordine

	Tabella Volumi				Tabella operazioni		
	Concetto	Tipo	Volume dati		Operazione	Tipo	Frequenza
	Registratore	E	20		Operazione 1	I	5
	Fornitura	E	60		Operazione 2	I	30
	Ordine	E	100		Operazione 3	I	10
	Relativa a	R	60				
	Compone	R	100				
	CON DATO DERIVATO				SENZA DATO DERIVATO		
	Tabella accessi - Op. 1				Tabella accessi - Op.1		
	Concetto	Accessi	Tipo		Concetto	Accessi	Tipo
	Fornitura	1	S		Relativa a	1	S
	Relativa a	1	S		Fornitura	1	S
	Registratore	1	L				
	Registratore	1	S				
Costo accessi:		7* 5 = 35/giorno				4 * 5 = 20/giorno	
	Tabella accessi - Op. 2				Tabella accessi - Op.2		
	Concetto	Accessi	Tipo		Concetto	Accessi	Tipo
	Registratore	1	L		Registratore	1	L
					Relativa a	3	L
					Fornitura	3	L
					Compone	5	L
					Ordine	5	L
Costo accessi:		1 * 30 = 30/giorno				17 * 30 = 510/giorno	

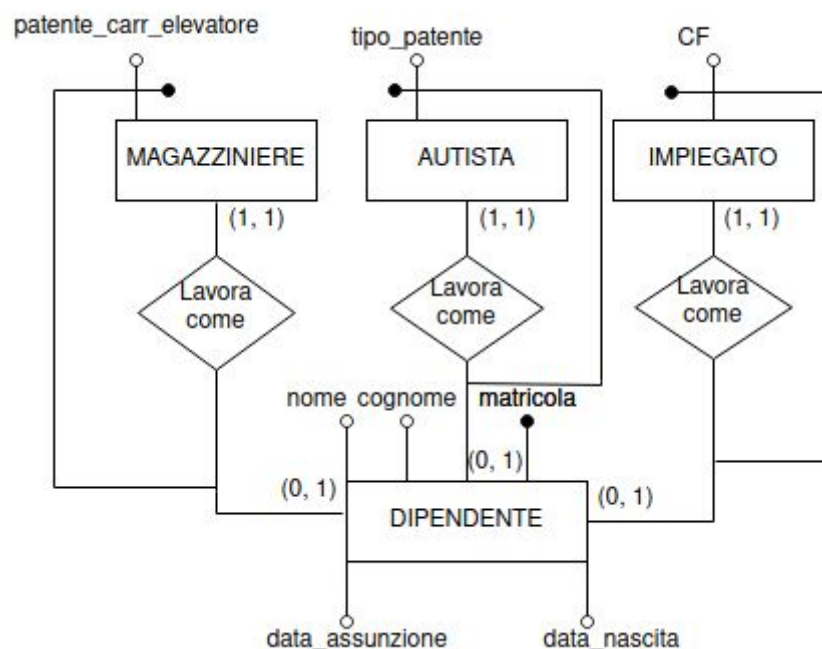
	<b>Tabella accessi - Op. 3</b>				<b>Tabella accessi - Op. 3</b>		
	<b>Concetto</b>	<b>Accessi</b>	<b>Tipo</b>		<b>Concetto</b>	<b>Accessi</b>	<b>Tipo</b>
	Ordine	1	S		Ordine	1	S
	Compone	1	S		Compone	1	S
	Registratore	1	L				
	Registratore	1	S				
Costo accessi:		7*10 = 70/giorno				4*10 = 40/giorno	
	Totale:	135 accessi/gi orno			Totale:	570 accessi/gio rno	
		<b>CONVIENE TENERE IL DATO DERIVATO</b>					

## C. Progetto logico - Schema relazionale

### 1. ELIMINAZIONE GERARCHIE ISA

Abbiamo scelto di applicare la tecnica del mantenimento delle entità in quanto può migliorare la chiarezza della divisione dei ruoli all'interno del magazzino e aiutare a distinguere dove è assegnato ogni dipendente.

Applicando questa tecnica, le entità nella gerarchia ISA vengono identificate esternamente con il loro padre, in modo da essere direttamente associate a lui; inoltre non ci sono relazioni con altre entità collegate alle figlie ma solo col genitore. Si creano così 4 entità separate: dipendente (l'entità genitore), autista, magazziniere e impiegato. Ognuna ha delle caratteristiche che la distingue dalle altre, oltre a quelle del padre identificate esternamente.



### 2. SELEZIONE CHIAVI PRIMARIE ED ELIMINAZIONE IDENTIFICAZIONI ESTERNE

Le identificazioni esterne sono trasformate in chiavi esterne verso le entità deboli, ovvero la chiave di identificazione esterna farà parte di una chiave composta nell'entità su cui ricadrà.

Contratto - Tipo\_Contratto: per effettuare un'identificazione esterna in questo caso è stata aggiunta la chiave primaria dell'entità Tipo\_Contratto, l'elemento debole in questione, alla chiave primaria di Contratto. Ottenendo così la nuova chiave composta da "ID" e "nome".

Cliente - Ordine: l'identificazione esterna avente come entità debole Ordine è stata trasformata aggiungendo la chiave principale di Cliente, P\_IVA\_C, alla chiave primaria di Ordine, ossia il numero dell'ordine; inoltre la chiave trasportata è anche chiave esterna dell'entità.

Veicolo - Tipo\_Veicolo: il caso è analogo ai precedenti, in cui viene portata la chiave primaria dell'entità debole Tipo\_veicolo nell'entità Veicolo, formando in tal modo la chiave composta con i campi "nome" e "tipo\_targa".

Veicolo - Guida - Dipendente: sull'entità Guida è stata applicata una doppia identificazione esterna per identificare i seguenti vincoli:

- dato un veicolo e una data, vi è un solo dipendente associato alla guida effettuata;
- dato un dipendente e una data, verrà associato un solo veicolo.

In questo caso la chiave primaria di Guida verrà composta dalla data in cui viene effettuata e dalle chiavi di Dipendente e Veicolo, rispettivamente la matricola, la targa e il nome del tipo di veicolo usato.

Fornitore - Fornitura - Registratore: all'entità Fornitura è stata applicata una doppia identificazione esterna per poter identificare in che data e quale fornitore ha fornito un certo modello di registratore. I vincoli che vengono applicati sono i seguenti:

- dato un fornitore ed una data c'è solo un modello di registratore fornito;
- specificato un registratore e una data, esso è dispensato da un solo fornitore.

### 3. TRASFORMAZIONE ATTRIBUTI COMPOSTI/MULTIPLI

Non è stato necessario apportare alcuna trasformazione di questo tipo in quanto non era presente alcun tipo di attributo composto o multiplo.

### 4. TRADUZIONI DELLE ENTITÀ IN SCHEMA DI RELAZIONI

Nel nostro schema non sono presenti associazione del tipo M:N quindi non è stata fatta alcuna derivazione di tipo "standard".

Tutte le associazioni del tipo 1:N sono state derivate trasportando la chiave principale di ogni entità del lato a (1:N) nella sua parte con cardinalità a (1:1).



L'associazione 1 a 1 che vede come partecipanti Ordine e Fattura non sono state trasformate applicando l'unione delle due entità in una sola ma tramite il mantenimento di entrambe le entità, in quanto abbiamo giudicato opportuno tenere separata la Fattura del resto per una maggiore chiarezza e atomicità del dato.

## 5. VERIFICA DELLA NORMALIZZAZIONE

Grazie alla costruzione ottenuta dai passaggi precedenti, le relazioni sono già presenti in forma normale.

### PROGETTO LOGICO COMPLESSIVO

DIPENDENTE (matricola, nome, cognome, data\_assunzione, data\_nascita, nome\_reparto)

FK: nome\_reparto REFERENCES REPARTO

MAGAZZINIERE (matricola, patente\_carr\_elevatore)

FK: matricola REFERENCES DIPENDENTE

AUTISTA (matricola, tipo\_patente)

FK: matricola REFERENCES DIPENDENTE

IMPIEGATO (matricola, CF)

FK: matricola REFERENCES DIPENDENTE

CLIENTE (P\_IVA\_C, nome, contatto, ambito\_lavorativo)

ORDINE (P\_IVA\_C, numero, quantita, costo, nome\_modello, ID)

FK: numero REFERENCES FATTURA

FK: nome\_modello REFERENCES REGISTRATORE

FK: P\_IVA\_C REFERENCES CLIENTE

FK: ID REFERENCES SPEDIZIONE

VEICOLO (nome, targa)

FK: nome REFERENCES TIPO\_VEICOLO

TIPO\_VEICOLO (nome, capienza, dimensione, peso, descrizione)

GUIDA (data, nome, targa, matricola)

FK: nome, targa REFERENCES VEICOLO

FK: matricola REFERENCES DIPENDENTE

CONTRATTO (ID, nome, matricola, stipendio, orario)

FK: matricola REFERENCES DIPENDENTE

FK: nome REFERENCES TIPO\_CONTRATTO

TIPO\_CONTRATTO (nome)

REPARTO (nome, descrizione)

SPEDIZIONE (ID, data\_prevista\_consegna, stato, nome, targa)

FK: nome, targa REFERENCES VEICOLO

FATTURA (numero, data, descrizione, totale)

REGISTRATORE (nome\_modello, peso, dimensione, quant\_disp, descrizione, P\_IVA\_F)

FK: P\_IVA\_F REFERENCES FORNITORE

SCAFFALE (ID, capacita, nome\_modello)

FK: nome\_modello REFERENCES REGISTRATORE

FORNITORE (P\_IVA\_F, nome, contatto)

FORNITURA (data, P\_IVA\_F, nome\_modello, quantita)

FK: P\_IVA\_F REFERENCES FORNITORE

FK: nome\_modello REFERENCES REGISTRATORE

SEDE (comune, provincia, indirizzo, CAP, P\_IVA\_F, P\_IVA\_C)

FK: P\_IVA\_F REFERENCES FORNITORE

FK: P\_IVA\_C REFERENCES CLIENTE

## D. Operazioni previste dalla base di dati

- 1) Il registratore meno acquistato
- 2) Visualizzare in ordine decrescente, in base all'utilizzo, i veicoli usati per le consegne
- 3) Il cliente che ha comprato più registratori (di qualsiasi modello)
- 4) Ordinare in senso decrescente il numero di registratori in magazzino per ogni modello
- 5) Numero medio di registratori acquistati da ciascun cliente
- 6) I clienti che non hanno effettuato ordini nell'ultimo anno solare
- 7) Stipendio medio per ogni tipo di contratto
- 8) L'ambito lavorativo più remunerante
- 9) Identificare chi ha effettuato consegne nel 2019 ma non nel 2020
- 10) Elencare chi ha effettuato un ordine ma non l'ha ancora ricevuto
- 11) La provincia con più fornitori
- 12) Il reparto con il maggior numero di dipendenti

- 1) SELECT r.nome\_modello, SUM(quantita) AS quantita  
FROM registratore AS r, ordine AS o  
WHERE r.nome\_modello = o.nome\_modello  
GROUP BY r.nome\_modello  
ORDER BY SUM(quantita)  
LIMIT 1;

	nome_modello [PK] character varying (30)	quantita bigint
1	Nettuna 700	23

- 2) SELECT nome, targa, COUNT(targa)  
FROM GUIDA  
GROUP BY targa, nome  
ORDER BY COUNT(targa) DESC;

	nome character varying (30)	targa character (7)	count bigint
1	Punto	ER528FO	3
2	Daily	ET734MN	3
3	Ducato	GA432AS	3
4	Panda	GA736VL	2
5	Transporter	FD345WE	2
6	Punto	FR319OD	2
7	Ducato	EE777TH	2
8	Transporter	FA135XX	2
9	Panda	EQ671PS	1

- 3) SELECT nome, SUM(quantita)  
 FROM CLIENTE C, ORDINE O  
 WHERE C.P\_IVA\_C = O.P\_IVA\_C  
 GROUP BY nome  
 ORDER BY SUM(quantita) DESC  
 LIMIT 1;

	nome character varying (50)	sum bigint
1	Samsung S.p.A.	210

- 4) SELECT nome\_modello, quant\_disp  
 FROM REGISTRATORE  
 GROUP BY nome\_modello  
 ORDER BY quant\_disp DESC;

	nome_modello [PK] character varying (30)	quant_disp integer
1	Form 100	1500
2	Form 200plus	700
3	Form 200	700
4	Nettuna 700	300
5	Nettuna 300	200

- 5) SELECT p\_iva\_c as cliente, ROUND(AVG(quantita), 2) AS acquisti\_medi  
 FROM ORDINE  
 GROUP BY p\_iva\_c;

	cliente character (11)	acquisti_medi numeric
1	01093853923	63.00
2	79685718274	70.00
3	16457382980	100.00
4	10395832935	30.00
5	08978675600	47.33
6	11122233300	56.67
7	12345678900	12.00
8	90329583855	13.00
9	32903953679	53.33
10	64821910928	20.00
11	09864902245	3.00
12	47384389284	40.00

6) SELECT nome as nome\_cliente  
 FROM CLIENTE  
 WHERE p\_iva\_c NOT IN (SELECT cl.p\_iva\_c  
                           FROM CLIENTE cl, ORDINE o, FATTURA f  
                           WHERE cl.p\_iva\_c = o.p\_iva\_c  
                           AND o.numero = f.numero  
                           AND f.data\_emissione >= '2020-01-01');

	<b>nome_cliente</b> character varying (50)
1	Samsung S.p.A.
2	Tetra Pak Italiana S.p.A.
3	Bar Gilles

7) SELECT nome, AVG(stipendio)  
 FROM contratto  
 GROUP BY nome;

	<b>nome</b> character varying (30)	<b>avg</b> double precision
1	Full-time	2300
2	Tirocinio	1100
3	A contratto	1585
4	Part-time	1700
5	Apprendistato	1500

8) SELECT ambito\_lavorativo, SUM(costo)  
 FROM cliente AS c, ordine AS o  
 WHERE c.P\_IVA\_C = o.P\_IVA\_C  
 GROUP BY ambito\_lavorativo  
 ORDER BY SUM(costo) DESC  
 LIMIT 1;

	<b>ambito_lavorativo</b> character varying (100)	<b>sum</b> bigint
1	Elettrodomestici	135000

```

9) SELECT matricola, nome, cognome
FROM DIPENDENTE
WHERE matricola NOT IN (SELECT d.matricola
                        FROM GUIDA gg, DIPENDENTE d, AUTISTA au
                        WHERE gg.matricola = d.matricola
                        AND d.matricola = au.matricola
                        AND gg.data >= '2020-01-01');

```

	<b>matricola</b> [PK] character varying (10)	<b>nome</b> character varying (15)	<b>cognome</b> character varying (15)
1	139858	Federico	Boccaletti
2	134598	Federico	Boccaletti
3	123456	Lorenzo	Bianchi
4	567890	Mattia	Innocenzi
5	234567	Davide	Beltrami
6	124578	Enea	Giglioli
7	189762	Federico	Ciampă
8	175845	Michele	Facchetti
9	556677	Simone	Miselli
10	112233	Stefano	Ruggieri
11	990088	Andrea	Albano
12	126790	Giacomo	Simoncini
13	197655	Jack	Moscato
14	245786	Chiara	Fornaciari
15	123657	Sofia	Spinelli
16	138906	Giulia	Manzoni

```

10) SELECT DISTINCT c.nome
FROM CLIENTE as c, ORDINE as o, SPEDIZIONE as s
WHERE c.P_IVA_C = o.P_IVA_C
AND o.ID = s.ID
AND o.id NOT IN (
  SELECT ID
  from SPEDIZIONE
  where stato = 'Consegnato'
);

```

	<b>nome</b> character varying (50)	
1	Samsung S.p.A.	
2	Buffetti	
3	Energys S.p.A.	
4	Max Mara S.r.l.	
5	Coop S.r.l.	
6	Walvoil S.p.A.	
7	Smeg S.p.A.	
8	Lombardini S.r.l.	

```

11)SELECT provincia, COUNT(provincia)
FROM sede AS s, fornitore AS f
WHERE s.P_IVA_F = f.P_IVA_F
GROUP BY provincia
ORDER BY COUNT(provincia) DESC
LIMIT 1;

```

	<b>provincia</b> character (2)	<b>count</b> bigint
1	PR	2

```

12)SELECT nome_reparto, COUNT(*) AS numero_dipendenti
FROM DIPENDENTE
GROUP BY (nome_reparto)
ORDER BY COUNT(*) DESC
LIMIT 1;

```

	<b>nome_reparto</b> character varying (30)	<b>numero_dipendenti</b> bigint
1	Picking	5

## E. Codice SQL per la creazione delle tabelle e per l'inserimento dei dati

### CREAZIONE DATABASE E TABELLE

```
CREATE TABLE CLIENTE (  
    P_IVA_C CHAR(11),  
    nome VARCHAR(50) NOT NULL,  
    contatto VARCHAR(100) NOT NULL,  
    ambito_lavorativo VARCHAR(100),  
    PRIMARY KEY (P_IVA_C)  
);  
  
CREATE TABLE TIPO_VEICOLO (  
    nome VARCHAR(30),  
    capienza FLOAT NOT NULL CHECK (capienza >= 0),  
    peso FLOAT CHECK (peso >= 0),  
    descrizione VARCHAR(100),  
    PRIMARY KEY (nome)  
);  
  
CREATE TABLE TIPO_CONTRATTO(  
    nome VARCHAR(30),  
    PRIMARY KEY (nome)  
);  
  
CREATE TABLE FATTURA(  
    numero INT,  
    data_emissione DATE NOT NULL,  
    descrizione VARCHAR(100),  
    totale FLOAT NOT NULL CHECK (totale >= 0),  
    PRIMARY KEY (numero)  
);  
  
CREATE TABLE FORNITORE(  
    P_IVA_F CHAR(11),  
    nome VARCHAR(50) NOT NULL,  
    contatto VARCHAR(100),  
    PRIMARY KEY (P_IVA_F)  
);
```



```

CREATE TABLE REPARTO(
    nome VARCHAR(30),
    descrizione VARCHAR(100),
    PRIMARY KEY (nome)
);

CREATE TABLE VEICOLO(
    nome VARCHAR(30),
    targa CHAR(7),
    PRIMARY KEY (nome, targa),
    FOREIGN KEY (nome) REFERENCES TIPO_VEICOLO
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE SPEDIZIONE(
    ID INT,
    data_prevista_consegna DATE NOT NULL,
    stato VARCHAR(30),
    nome VARCHAR(30) NOT NULL,
    targa CHAR(7) NOT NULL,
    PRIMARY KEY (ID),
    FOREIGN KEY (nome, targa) REFERENCES VEICOLO
        ON DELETE SET NULL
        ON UPDATE CASCADE
);

CREATE TABLE REGISTRATORE (
    nome_modello VARCHAR(30) NOT NULL,
    peso FLOAT NOT NULL CHECK (peso >= 0),
    dimensione FLOAT NOT NULL CHECK (dimensione >= 0),
    quant_disp INT NOT NULL CHECK (quant_disp >= 0),
    descrizione VARCHAR(100),
    PRIMARY KEY (nome_modello)
);

CREATE TABLE FORNITURA (
    quantita INT NOT NULL CHECK (quantita >= 0),
    data DATE NOT NULL,
    P_IVA_F CHAR(11) NOT NULL,
    nome_modello VARCHAR(30) NOT NULL,
    PRIMARY KEY (data, P_IVA_F, nome_modello),

```

```

        FOREIGN KEY (P_IVA_F) REFERENCES FORNITORE
        ON DELETE NO ACTION
        ON UPDATE CASCADE,
        FOREIGN KEY (nome_modello) REFERENCES REGISTRATORE
        ON DELETE NO ACTION
        ON UPDATE CASCADE
    );

```

```

CREATE TABLE SCAFFALE(
    ID INT,
    capacita INT CHECK (capacita >= 0),
    nome_modello VARCHAR(30),
    PRIMARY KEY (ID),
    FOREIGN KEY (nome_modello) REFERENCES REGISTRATORE
    ON DELETE SET NULL
    ON UPDATE CASCADE
);

```

```

CREATE TABLE DIPENDENTE(
    matricola VARCHAR(10),
    nome VARCHAR(15) NOT NULL,
    cognome VARCHAR(15) NOT NULL,
    data_assunzione DATE NOT NULL,
    data_nascita DATE NOT NULL,
    nome_reparto VARCHAR(30) NOT NULL,
    PRIMARY KEY(matricola),
    FOREIGN KEY (nome_reparto) REFERENCES REPARTO
    ON DELETE SET NULL
    ON UPDATE CASCADE
);

```

```

CREATE TABLE SEDE(
    comune VARCHAR(30),
    provincia CHAR(2),
    CAP CHAR(5) NOT NULL,
    indirizzo VARCHAR(50) NOT NULL,
    P_IVA_F CHAR(11),
    P_IVA_C CHAR(11),
    PRIMARY KEY (comune, provincia, indirizzo),
    FOREIGN KEY (P_IVA_F) REFERENCES FORNITORE
    ON DELETE SET NULL
    ON UPDATE CASCADE,

```

```

        FOREIGN KEY (P_IVA_C) REFERENCES CLIENTE
        ON DELETE SET NULL
        ON UPDATE CASCADE
    );

CREATE TABLE MAGAZZINIERE(
    matricola VARCHAR(10),
    patente_carr_elevatore BIT,
    PRIMARY KEY(matricola, patente_carr_elevatore),
    FOREIGN KEY (matricola) REFERENCES DIPENDENTE
    ON DELETE NO ACTION
    ON UPDATE CASCADE
);

CREATE TABLE AUTISTA(
    matricola VARCHAR(10),
    tipo_patente CHAR(3) NOT NULL,
    PRIMARY KEY(matricola, tipo_patente),
    FOREIGN KEY (matricola) REFERENCES DIPENDENTE
    ON DELETE NO ACTION
    ON UPDATE CASCADE
);

CREATE TABLE IMPIEGATO(
    matricola VARCHAR(10),
    CF VARCHAR(16) NOT NULL,
    PRIMARY KEY (matricola, CF),
    FOREIGN KEY (matricola) REFERENCES DIPENDENTE
    ON DELETE NO ACTION
    ON UPDATE CASCADE
);

CREATE TABLE GUIDA(
    data DATE,
    nome VARCHAR(30),
    targa CHAR(7),
    matricola VARCHAR(10),
    PRIMARY KEY (data, nome, targa, matricola),
    FOREIGN KEY (nome, targa) REFERENCES VEICOLO
    ON DELETE CASCADE,
    ON UPDATE CASCADE,
    FOREIGN KEY (matricola) REFERENCES DIPENDENTE

```

```

        ON DELETE CASCADE
        ON UPDATE CASCADE
    );

CREATE TABLE CONTRATTO(
    ID INT NOT NULL,
    nome VARCHAR(30),
    matricola VARCHAR(10),
    stipendio FLOAT NOT NULL CHECK (stipendio >= 0),
    orario INT CHECK (ORARIO >= 0),
    PRIMARY KEY (ID, nome),
    FOREIGN KEY (matricola) REFERENCES DIPENDENTE
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    FOREIGN KEY (nome) REFERENCES TIPO_CONTRATTO
    ON DELETE SET NULL
    ON UPDATE CASCADE
);

CREATE TABLE ORDINE(
    P_IVA_C CHAR(11),
    numero INT NOT NULL CHECK (numero >= 0),
    quantita INT NOT NULL CHECK (quantita >= 0),
    costo INT NOT NULL CHECK (costo >= 0),
    nome_modello VARCHAR(30) NOT NULL,
    ID int,
    PRIMARY KEY (P_IVA_C, numero),
    FOREIGN KEY (numero) REFERENCES FATTURA
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    FOREIGN KEY (nome_modello) REFERENCES REGISTRATORE
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
    FOREIGN KEY (P_IVA_C) REFERENCES CLIENTE
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
    FOREIGN KEY (ID) REFERENCES SPEDIZIONE
    ON DELETE NO ACTION
    ON UPDATE CASCADE
);

```

## QUERY DI INSERIMENTO DEI DATI

```
INSERT INTO CLIENTE (P_IVA_C, nome, contatto, ambito_lavorativo)
VALUES ('11122233300', 'Buffetti', '+391112223344', 'Cancelleria'),
('12345678900', 'Energya S.p.A.', '+391231233445', 'Cancelleria'),
('08978675600', 'Coop S.r.l.', '+391010293948', 'Alimentari'),
('16457382980', 'Smeg S.p.A.', '+391236831286', 'Elettrodomestici'),
('79685718274', 'Samsung S.p.A.', '+393204563459', 'Elettrodomestici'),
('47384389284', 'Max Mara S.r.l.', '+393453451234', 'Abbigliamento'),
('90329583855', 'Tetra Pak Italiana S.p.A.', '+391234349087', 'Riparazione'),
('10395832935', 'Comet S.p.A.', '+393124759843', 'Elettronica'),
('64821910928', 'Conad S.p.A.', '+393180900573', 'Alimentari'),
('01093853923', 'Walvoil S.p.A.', '+393453451234', 'Meccanica'),
('32903953679', 'Lombardini S.r.l.', '+393126789456', 'Meccanica'),
('09864902245', 'Bar Gilles', '+393472764312', 'Ristorazione');
```

```
INSERT INTO TIPO_VEICOLO (nome, capienza, peso, descrizione)
VALUES ('Punto', '300', '1300', 'Veicolo utilitario per spostamenti di lavoro'),
('Giulietta', '330', '1500', 'Veicolo utilitario per spostamenti di lavoro'),
('Panda', '280', '1200', 'Veicolo utilitario per spostamenti di lavoro'),
('Ducato', '17000', '2000', 'Veicolo per trasporto di medie quantità di merce'),
('Transporter', '15000', '1800', 'Veicolo per trasporto di medie quantità di merce'),
('Daily', '13000', '1700', 'Veicolo per trasporto di medie quantità di merce');
```

```
INSERT INTO FATTURA (numero, data_emissione, descrizione, totale)
VALUES (0001234, '2020-07-10', 'Fornitura di lotto di registratori per S.p.A', 60000),
(0001934, '2019-07-10', 'Fornitura di lotto di registratori per S.p.A', 6500),
(0012345, '2018-07-10', 'Fornitura di lotto di registratori per S.p.A', 25000),
(0054321, '2020-07-10', 'Fornitura di lotto di registratori per S.p.A', 8750),
(1236876, '2019-08-05', 'Fornitura di lotto di registratori per S.p.A', 25000),
(0089755, '2020-08-05', 'Fornitura di lotto di registratori per S.p.A', 46000),
(0012434, '2019-11-27', 'Fornitura di lotto di registratori per S.p.A', 40500),
(0201934, '2020-11-27', 'Fornitura di lotto di registratori per S.p.A', 5000),
(1012345, '2019-07-28', 'Fornitura di lotto di registratori per S.p.A', 15000),
(0254321, '2020-07-28', 'Fornitura di lotto di registratori per S.p.A', 3000),
(1996876, '2019-03-08', 'Fornitura di lotto di registratori per S.p.A', 5300),
(0004321, '2020-03-08', 'Fornitura di lotto di registratori per S.r.l.', 13000),
(0066554, '2020-04-15', 'Fornitura di lotto di registratori per S.r.l.', 3000),
(0975344, '2020-04-15', 'Fornitura di lotto di registratori per S.r.l.', 8000),
(0004320, '2019-07-31', 'Fornitura di lotto di registratori per S.r.l.', 13000),
```

```
(1204333, '2020-07-31', 'Fornitura di lotto di registratori per S.r.l.', 4700),
(0204321, '20219-07-31', 'Fornitura di lotto di registratori per S.r.l.', 8000),
(0466554, '2020-05-09', 'Fornitura di lotto di registratori per S.r.l.', 7200),
(0975554, '2020-05-09', 'Fornitura di lotto di registratori per S.r.l.', 6000),
(0004520, '2019-05-09', 'Fornitura di lotto di registratori per S.r.l.', 6000),
(0554333, '2020-07-10', 'Fornitura di lotto di registratori per S.r.l.', 13000),
(1004311, '2019-07-10', 'Fornitura di lotto di registratori per privati.', 3300);
```

```
INSERT INTO FORNITORE (p_iva_f, nome, contatto)
VALUES ('54967315927', 'Amazon', 'assistenzaclienti@amazon.com'),
('18967135246', 'TecnoCassa', 'contatti@tecnocassa.it'),
('86425869468', 'Olivetti S.p.A.', 'vendite@olivetti.it'),
('59761584239', 'Gruppo Pixel', 'pixel@gmail.com'),
('74139851865', 'Gruppo CEI', 'vendite@cei.it');
```

```
INSERT INTO REPARTO (nome, descrizione)
VALUES ('Ricevimento', 'Area di scarico facilitato per la merce sui mezzi'),
('Controllo', 'Area per il controllo qualità del prodotto'),
('Stoccaggio', 'Area stoccaggio merce'),
('Picking', 'Partendo da unità di carico piene si vanno a formare unità di carico miste'),
('Imballaggio', 'Consolidare, imballare e caricare i prodotti sui mezzi'),
('Spedizione', 'Vengono preparate le spedizioni');
```

```
INSERT INTO VEICOLO (nome, targa)
VALUES ('Punto', 'ER528FO'),
('Punto', 'FR319OD'),
('Panda', 'EQ671PS'),
('Panda', 'GA736VL'),
('Ducato', 'GA432AS'),
('Ducato', 'EE777TH'),
('Transporter', 'FA135XX'),
('Transporter', 'FD345WE'),
('Daily', 'ET734MN');
```

```
INSERT INTO REGISTRATORE (nome_modello, peso, dimensione, quant_disp,
descrizione)
VALUES ('Form 100', 0.250, 1000, 1500, 'Registratore di cassa telematico più versatile della gamma'),
('Form 200', 0.300, 1000, 700, 'Nuovo registratore di cassa telematico dal design moderno e curato'),
```

('Form 200plus', 0.450, 1000, 700, 'Registratore di cassa telematico con tastiera ibrida, touch screen e tradizionale'),  
 ('Nettuna 700', 1.000, 1000, 300, 'Registratore progettato per offrire una risposta professionale a qualsiasi esigenza'),  
 ('Nettuna 300', 0.750, 1000, 200, 'Registratore compatto, curato nei dettagli, con prestazioni tecniche avanzate');

```
INSERT INTO FORNITURA (quantita, data, P_IVA_F, nome_modello)
VALUES (100, '2019-08-19', '54967315927', 'Form 100'),
(200, '2020-09-02', '18967135246', 'Form 200'),
(150, '2019-11-29', '86425869468', 'Form 200plus'),
(50, '2020-07-02', '59761584239', 'Nettuna 700'),
(75, '2019-02-01', '74139851865', 'Nettuna 300');
```

```
INSERT INTO SCAFFALE (ID, capacita, nome_modello)
VALUES (0, 300, 'Form 100'),
(1, 500, 'Form 100'),
(2, 500, 'Form 100'),
(3, 500, 'Form 100'),
(4, 300, 'Form 200'),
(5, 300, 'Form 200'),
(6, 300, 'Form 200'),
(7, 500, 'Form 200plus'),
(8, 500, 'Form 200plus'),
(9, 500, 'Nettuna 700'),
(10, 500, 'Nettuna 300');
```

```
INSERT INTO DIPENDENTE (matricola, nome, cognome, data_assunzione,
data_nascita, nome_reparto)
VALUES (139858, 'Federico', 'Boccaletti', '2016-08-10', '1986-05-11', 'Ricevimento'),
(134598, 'Federico', 'Boccaletti', '2018-09-10', '1996-09-19', 'Imballaggio'),
(123456, 'Lorenzo', 'Bianchi', '2013-05-15', '1988-11-17', 'Stoccaggio'),
(567890, 'Mattia', 'Innocenzi', '2019-10-11', '1983-04-15', 'Picking'),
(234567, 'Davide', 'Beltrami', '2016-08-09', '1982-04-14', 'Picking'),
(124578, 'Enea', 'Giglioli', '2016-08-10', '1980-06-11', 'Imballaggio'),
(189762, 'Federico', 'Ciampà', '2017-07-10', '1985-09-12', 'Controllo'),
(175845, 'Michele', 'Facchetti', '2010-02-02', '1986-10-15', 'Picking'),
(556677, 'Simone', 'Miselli', '2016-02-02', '1997-11-19', 'Stoccaggio'),
(112233, 'Stefano', 'Ruggieri', '2011-12-09', '1998-12-20', 'Controllo'),
(445568, 'Andrea', 'Schwarz', '2012-11-16', '1999-07-23', 'Spedizione'),
(990088, 'Andrea', 'Albano', '2013-10-22', '1978-06-30', 'Picking'),
(126790, 'Giacomo', 'Simoncini', '2014-09-23', '1989-01-31', 'Ricevimento'),
```

```
(197655, 'Jack', 'Moscato', '2015-01-15', '1991-02-22', 'Picking'),
(168798, 'William', 'Grillo', '2013-02-25', '1995-03-27', 'Spedizione'),
(245786, 'Chiara', 'Fornaciari', '2016-01-31', '1979-03-12', 'Controllo'),
(123459, 'Alessia', 'Mennillo', '2017-01-30', '1996-05-09', 'Spedizione'),
(123657, 'Sofia', 'Spinelli', '2018-06-23', '1994-09-08', 'Imballaggio'),
(138906, 'Giulia', 'Manzoni', '2019-06-14', '1991-08-04', 'Ricevimento'),
(123455, 'Enrica', 'Lanzafame', '2020-05-14', '2000-07-12', 'Spedizione');
```

```
INSERT INTO TIPO_CONTRATTO (nome)
VALUES ('Apprendistato'), ('Tirocinio'), ('Part-time'), ('Full-time'), ('A contratto');
```

```
INSERT INTO CONTRATTO (id, nome, matricola, stipendio, orario)
VALUES (1, 'Apprendistato', 139858, 2000, 8),
(2, 'Part-time', 134598, 1500, 5),
(3, 'Full-time', 123456, 2500, 8),
(4, 'A contratto', 567890, 1750, 4),
(5, 'Tirocinio', 234567, 1000, 5),
(6, 'A contratto', 124578, 1800, 8),
(7, 'Part-time', 189762, 1400, 5),
(8, 'Apprendistato', 175845, 1600, 8),
(9, 'A contratto', 556677, 1800, 8),
(10, 'A contratto', 112233, 1800, 8),
(11, 'Full-time', 445568, 2100, 8),
(12, 'Part-time', 990088, 2200, 4),
(13, 'A contratto', 197655, 1800, 8),
(14, 'Tirocinio', 168798, 1200, 4),
(15, 'A contratto', 245786, 1300, 8),
(16, 'A contratto', 123459, 1100, 8),
(17, 'A contratto', 123657, 1000, 8),
(18, 'Apprendistato', 124578, 900, 8),
(19, 'A contratto', 138906, 1700, 8),
(20, 'A contratto', 123455, 1800, 8);
```

```
INSERT INTO IMPIEGATO (matricola, cf)
VALUES (138906, 'ZVJMBS46S52I191T'),
(123455, 'CLNLQD54H62L466M'),
(245786, 'SVGZLT63A54E570Y'),
(126790, 'VNDRFJ67C27A953S'),
(556677, 'JLPLPQ60D41A084F');
```

```
INSERT INTO MAGAZZINIERE(matricola, patente_carr_elevatore)
VALUES (134598, '1'),
```



```
(567890, '0'),  
(445568, '0'),  
(189762, '1'),  
(139858, '0'),  
(123456, '0'),  
(234567, '1'),  
(124578, '0'),  
(175845, '0'),  
(112233, '0'),  
(990088, '1'),  
(197655, '0'),  
(123657, '1'),  
(123459, '1');
```

```
INSERT INTO AUTISTA (matricola, tipo_patente)  
VALUES (123459, 'B'),  
(445568, 'B'),  
(168798, 'C'),  
(123455, 'C');
```

```
INSERT INTO GUIDA (data, nome, targa, matricola)  
VALUES ('2019-11-02', 'Punto', 'ER528FO', 123459),  
( '2020-05-02', 'Punto', 'FR319OD', 123459),  
( '2020-06-15', 'Panda', 'EQ671PS', 445568),  
( '2019-03-22', 'Panda', 'GA736VL', 123459),  
( '2019-11-12', 'Ducato', 'GA432AS', 123459),  
( '2019-12-22', 'Ducato', 'EE777TH', 123459),  
( '2019-12-22', 'Transporter', 'FA135XX', 168798),  
( '2019-12-23', 'Transporter', 'FD345WE', 123455),  
( '2019-12-23', 'Daily', 'ET734MN', 168798),  
( '2019-11-11', 'Punto', 'ER528FO', 445568),  
( '2019-12-24', 'Punto', 'ER528FO', 123459),  
( '2019-12-27', 'Punto', 'FR319OD', 445568),  
( '2019-12-28', 'Panda', 'GA736VL', 123459),  
( '2020-01-07', 'Transporter', 'FA135XX', 123455),  
( '2020-01-15', 'Daily', 'ET734MN', 168798),  
( '2019-09-23', 'Ducato', 'EE777TH', 123459),  
( '2020-02-27', 'Ducato', 'GA432AS', 445568),  
( '2020-02-28', 'Ducato', 'GA432AS', 123459),  
( '2020-04-15', 'Transporter', 'FD345WE', 123455),  
( '2020-07-19', 'Daily', 'ET734MN', 168798);
```

```

INSERT INTO SEDE (comune, provincia, CAP, indirizzo, P_IVA_F, P_IVA_C)
VALUES ('Reggio Emilia', 'RE', '42123', 'Via Francia 24', NULL, '11122233300'),
('Reggio Emilia', 'RE', '42124', 'Via Bismantova 9', NULL, '12345678900'),
('Bologna', 'BO', '40121', 'Via Saragozza 15', NULL, '08978675600'),
('Mantova', 'MN', '46100', 'Via Girolamo 13', NULL, '16457382980'),
('Sassuolo', 'MO', '41049', 'Via Cavour 56', NULL, '79685718274'),
('Reggio Emilia', 'RE', '42124', 'Via Galimberti 12', NULL, '47384389284'),
('Modena', 'MO', '41121', 'Via Garibaldi 5', NULL, '90329583855'),
('Reggio Emilia', 'RE', '42122', 'Via Pasteur 12', NULL, '10395832935'),
('Reggio Emilia', 'RE', '42122', 'Via Danimarca 13', NULL, '64821910928'),
('Padova', 'PD', '35121', 'Via Svevo 12', NULL, '01093853923'),
('Bologna', 'BO', '40121', 'Via Saba 10', NULL, '32903953679'),
('Milano', 'MI', '20121', 'Via Barcellona 34', NULL, '09864902245'),
('Piacenza', 'PC', '29121', 'Via Massenet 14', '54967315927', NULL),
('Parma', 'PR', '43121', 'Via Balzac 34', '18967135246', NULL),
('Parma', 'PR', '43121', 'Via Stendhal 120', '86425869468', NULL),
('Piacenza', 'PC', '29121', 'Via Vico 11', '59761584239', NULL),
('Modena', 'MO', '41121', 'Via della Liberazione 111', '74139851865', NULL);

```

```

INSERT INTO SPEDIZIONE (id, data_prevista_consegna, stato, nome, targa)
VALUES (5698746, '2020-07-10', 'Consegnato', 'Transporter', 'FA135XX'),
(7845692, '2020-08-05', 'In preparazione', 'Ducato', 'EE777TH'),
(6532941, '2020-11-27', 'In preparazione', 'Daily', 'ET734MN'),
(1596374, '2020-07-28', 'In transito', 'Transporter', 'FD345WE'),
(7835842, '2020-03-08', 'Consegnato', 'Ducato', 'GA432AS'),
(1674298, '2020-04-15', 'Consegnato', 'Daily', 'ET734MN'),
(4961873, '2020-07-31', 'In preparazione', 'Ducato', 'GA432AS'),
(7931456, '2020-05-09', 'Consegnato', 'Transporter', 'FD345WE'),
(6984235, '2020-07-10', 'In transito', 'Daily', 'ET734MN');

```

```

INSERT INTO ORDINE (p_iva_c, numero, quantita, costo, nome_modello, ID)
VALUES (11122233300, 0004321, 100, 10000, 'Form 100', 5698746),
('16457382980', 0001234, 50, 55000, 'Form 100', 5698746),
('09864902245', 1004311, 3, 3000, 'Nettuna 700', 5698746),
('11122233300', 0066554, 20, 2500, 'Form 100', 7845692),
('08978675600', 0975344, 7, 7000, 'Nettuna 700', 7845692),
('79685718274', 0001934, 10, 5000, 'Nettuna 300', 6532941),
('11122233300', 0004320, 50, 10000, 'Form 200', 6532941),
('47384389284', 1204333, 40, 4000, 'Form 100', 1596374),
('79685718274', 0012345, 100, 20000, 'Form 200', 1596374),
('32903953679', 0204321, 70, 7000, 'Form 100', 1596374),
('32903953679', 0466554, 40, 6500, 'Form 200plus', 7835842),

```

```
( '08978675600', 0975554, 25, 5000, 'Form 200', 7835842),
( '32903953679', 0004520, 50, 5000, 'Form 100', 7835842),
( '10395832935', 0054321, 30, 6000, 'Form 200', 1674298),
( '79685718274', 1236876, 100, 20000, 'Form 200', 1674298),
( '08978675600', 0554333, 110, 11000, 'Form 100', 1674298),
( '01093853923', 0089755, 80, 40000, 'Nettuna 300', 4961873),
( '16457382980', 0012434, 150, 35000, 'Form 200plus', 4961873),
( '64821910928', 0201934, 20, 4000, 'Form 200', 7931456),
( '90329583855', 1012345, 13, 13000, 'Nettuna 700', 7931456),
( '12345678900', 0254321, 12, 2400, 'Form 200', 6984235),
( '01093853923', 1996876, 46, 4600, 'Form 100', 6984235);
```

## QUERY DI MODIFICA DEI DATI

- 1) Modifica stato e data di consegna di una spedizione in transito con codice '6984235' in quanto è arrivata;
- 2) Modifica del contatto del fornitore con partita iva '74139851865' in 'venditeCEI@gmail.com'
- 3) Cambio di tipo contratto per matricola numero 134598 da 'Part-time' in 'Full-time'

- 1) UPDATE SPEDIZIONE  
SET data\_prevista\_consegna=current\_date, stato='Consegnato'  
WHERE id = 6984235;
- 2) UPDATE FORNITORE  
SET contatto = 'venditeCEI@gmail.com'  
WHERE p\_iva\_f = '74139851865';
- 3) UPDATE CONTRATTO  
SET nome = 'Full-time'  
WHERE matricola = 134598;

## QUERY DI ELIMINAZIONE DEI DATI

- 1) Eliminazione di dipendenti con contratto di tipo 'Tirocinio'
- 2) Ipotesi di vendita o rottamazione dell'auto targata 'ER528FO'
- 3) Eliminare tutte le fatture antecedenti il '2020-07-01'(primo semestre 2020)

- 1) DELETE FROM CONTRATTO  
WHERE nome = 'Tirocinio';
- 2) DELETE FROM VEICOLO  
WHERE targa = 'ER528FO';
- 3) DELETE FROM FATTURA  
WHERE data\_emissione < '2020-07-01';

## CREAZIONE DI TRIGGER

- 1) Trigger per il controllo e l'aggiornamento del dato derivato quant\_disp alla richiesta di un ordine
- 2) Trigger per il controllo e l'aggiornamento del dato derivato quant\_disp all'arrivo di una fornitura

```

1) CREATE FUNCTION nuovo_ordine() RETURNS trigger AS $$
DECLARE
    q1 INTEGER;
BEGIN
    SELECT quant_disp INTO q1
    FROM registratore AS r
    WHERE r.nome_modello = NEW.nome_modello;
    IF NEW.quantita > q1 THEN
        RAISE EXCEPTION 'Quantità maggiore della disponibilità in
magazzino';
    ELSE
        UPDATE REGISTRATORE
        SET quant_disp = quant_disp - NEW.quantita
        WHERE NEW.nome_modello = nome_modello;
        RETURN NEW;
    END IF;
END;
$$ LANGUAGE 'plpgsql';

CREATE TRIGGER GESTIONE_ORDINE
AFTER INSERT ON ORDINE
FOR EACH ROW
EXECUTE PROCEDURE nuovo_ordine();

```

```
2) CREATE FUNCTION nuova_fornitura() RETURNS trigger AS $$
BEGIN
    IF NEW.quantita < 0 THEN
        RAISE EXCEPTION 'Quantità non ammissibile';
    END IF;
    UPDATE REGISTRATORE
    SET quant_disp = quant_disp + NEW.quantita
    WHERE NEW.nome_modello = nome_modello;
    RETURN NEW;
END;
$$ LANGUAGE 'plpgsql';

CREATE TRIGGER GESTIONE_FORNITURA
BEFORE INSERT ON FORNITURA
FOR EACH ROW
EXECUTE PROCEDURE nuova_fornitura();
```