

Università degli Studi di Trieste
Intelligenza Artificiale & Data Analytics

Progetto di Basi di Dati
“Futura S.p.A. - Azienda informatica”

Tommaso Mingrone

Indice

1. Presentazione progetto	2
2. Schema Entity-Relationship	3
3. Dizionario dei dati	4
4. Vincoli non esprimibili	6
5. Tabella dei volumi	7
6. Schema Entity-Relationship ristrutturato	8
7. Schema Logico	10
8. Query SQL	12

Capitolo 1

Presentazione progetto

Un'azienda informatica denominata *Futura S.p.A.* si occupa della produzione di prodotti informatici da diversi anni. Nel corso del tempo il business si è sviluppato e l'azienda è stata ampliata di conseguenza.

Nasce così l'esigenza di creare un database per una più consona ed efficiente gestione aziendale.

I punti di maggior interesse da sviluppare sono diversi, tra cui la gestione delle risorse primarie e del personale. Questi pilastri della catena produttiva sono alla base del sistema aziendale ed è dunque necessaria una corretta direzione.

Un richiamo esclusivo è riservato ai clienti, che acquistando prodotti online o in loco permettono all'azienda di proseguire nella propria attività.

In particolare, per quanto riguarda la logistica delle materie prime si è interessati ad etichettare la tipologia, utile nei vari settori produttivi, a riconoscerla attraverso un codice identificativo. Risulta importante anche saperne la disponibilità e il costo legato a quest'ultime; mediante i fornitori, infatti, l'azienda riesce a rifornirsi degli approvvigionamenti necessari.

Per accertarsi delle materie a disposizione è fondamentale essere a conoscenza del tasso di occupazione dei magazzini; utile anche per determinare la quantità di prodotti finiti, pronti per essere imballati e spediti dai corrieri ai diversi clienti.

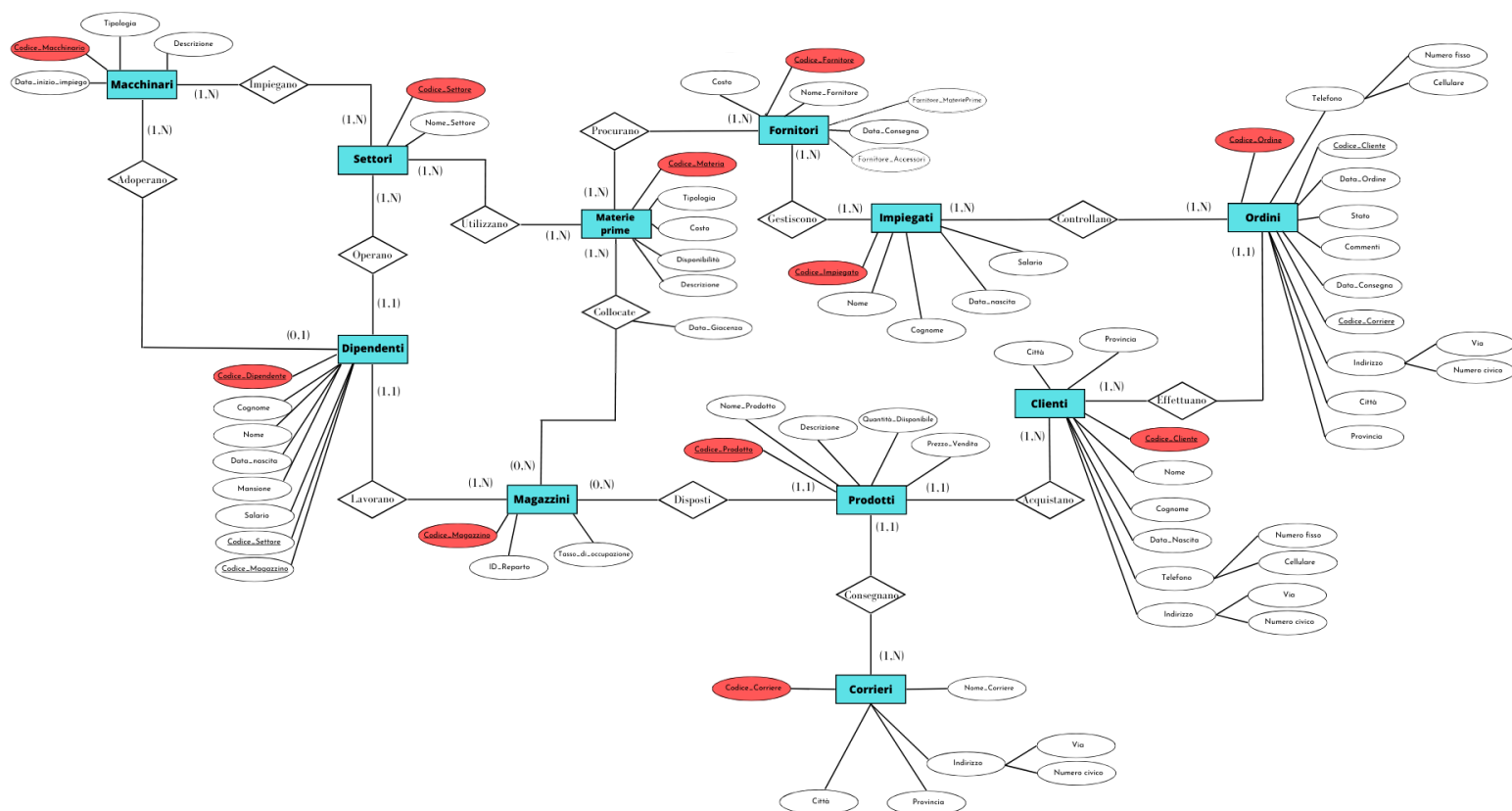
Il personale è suddiviso tra dipendenti e impiegati. I dipendenti operano all'interno dei settori e sono coloro che adoperano i macchinari utili all'elaborazione del materiale. Gli impiegati lavorano in ufficio e si occupano, invece, dell'amministrazione aziendale, tra cui anche la gestione dei contatti coi fornitori e il controllo degli ordini.

I clienti sono riconosciuti attraverso un numero univoco progressivo, i loro nominativi, la data di nascita, un recapito telefonico oltre che al proprio indirizzo, la città e la provincia di provenienza.

Capitolo 2

Schema Entity-Relationship

Di seguito il modello E/R ipotizzato per la Futura S.p.A.; le entità sono stati evidenziati in azzurro, mentre gli identificatori in rosso.



Capitolo 3

Dizionario dei dati

Entità	Descrizione	Attributi	Identificatore
Macchinari	Macchinari aziendali	Codice_Macchinario, data_inizio_impiego, tipologia, descrizione	Codice_Macchinario
Settori	Settori produttivi e di controllo	Codice_Settoe, Nome_Settoe	Codice_Settoe
Dipendenti	Operai, magazzinieri, lavoratori dipendenti	Codice_Dipendente, Cognome, Nome, Data_nascita, Mansione, Salario, Codice_Settoe, Codice_Magazzino	Codice_Dipendente
Materie prime	Materie prime per la produzione	Codice_Materia, Tipologia, Costo, Disponibilità, Descrizione	Codice_Materia
Magazzini	Magazzini aziendali	Codice_Magazzino, ID_Reparto, Tasso_di_occupazione	Codice_Magazzino
Fornitori	Fornitori di materie prime	Codice_Fornitore, Nome_Fornitore, Data_Consegna, Costo, Fornitore_MateriePrime, Fornitore_Accessori	Codice_Fornitore
Impiegati	Impiegati d'ufficio	Codice_Impiegato, Nome, Cognome, Data_nascita, Salario	Codice_Impiegato
Ordini	Ordini effettuati dai clienti	Codice_Ordine, Codice_Cliente, Data_Ordine, Stato, Commenti, Telefono, Data_Consegna, Indirizzo, Città, Provincia, Codice_Corriere	Codice_Ordine
Clienti	Persona che acquista almeno un prodotto	Codice_Cliente, Nome, Cognome, Data_Nascita, Telefono, Indirizzo	Codice_Cliente
Prodotti	Prodotti finiti, pronti per la vendita	Codice_Prodotto, Nome_Prodotto, Prezzo_Vendita, Quantità_Disponibile, Descrizione	Codice_Prodotto
Corrieri	Spedizionieri convenzionati con l'azienda	Codice_Corriere, Nome_Corriere, Città, Provincia, Indirizzo	Codice_Corriere

Associazione	Descrizione	Composizione	Attributi
Adoperano	Un dipendente può adoperare un macchinario	Codice_Macchinario, Codice_Dipendente	-
Impiegano	I settori impiegano i macchinari per la produzione	Codice_Macchinario, Codice_Setto	-
Operano	I dipendenti operano nei settori	Codice_Setto, Codice_Dipendente	-
Lavorano	I dipendenti lavorano nei magazzini	Codice_Dipendente, Codice_Magazzino	-
Utilizzano	Un settore utilizza più materie prime	Codice_Setto, Codice_Materia	-
Procurano	I fornitori procurano più materie prime	Codice_Materia, Codice_Fornitore	-
Disposti	Un prodotto è disposto in un magazzino	Codice_Magazzino, Codice_Prodotto	-
Consegnano	Un corriere consegna più prodotti	Codice_Prodotto, Codice_Corriere	-
Gestiscono	Un impiegato gestisce più fornitori	Codice_Fornitore, Codice_Impiegato	-
Controllano	Un impiegato controlla più ordini	Codice_Impiegato, Codice_Ordine	-
Effettuano	Un cliente effettua più ordini	Codice_Cliente, Codice_Ordine	-
Collocate	Una materia è collocata in un magazzino	Codice_Materia, Codice_Magazzino	Data_Giacenza
Acquistano	Un cliente acquista più prodotti	Codice_Corriere, Codice_Prodotto	-

Capitolo 4

Vincoli non esprimibili

Dall'analisi del testo del problema emergono i seguenti vincoli non esprimibili:

- Quando il tasso di occupazione del magazzino è minore del 50%, è necessario rifornirsi di materie prime;
- La data di consegna di un ordine non può essere antecedente alla data in cui quest'ultimo è stato effettuato;
- Per regolamento aziendale, se un dipendente lavora in un settore di produzione non può lavorare anche nei magazzini e viceversa.

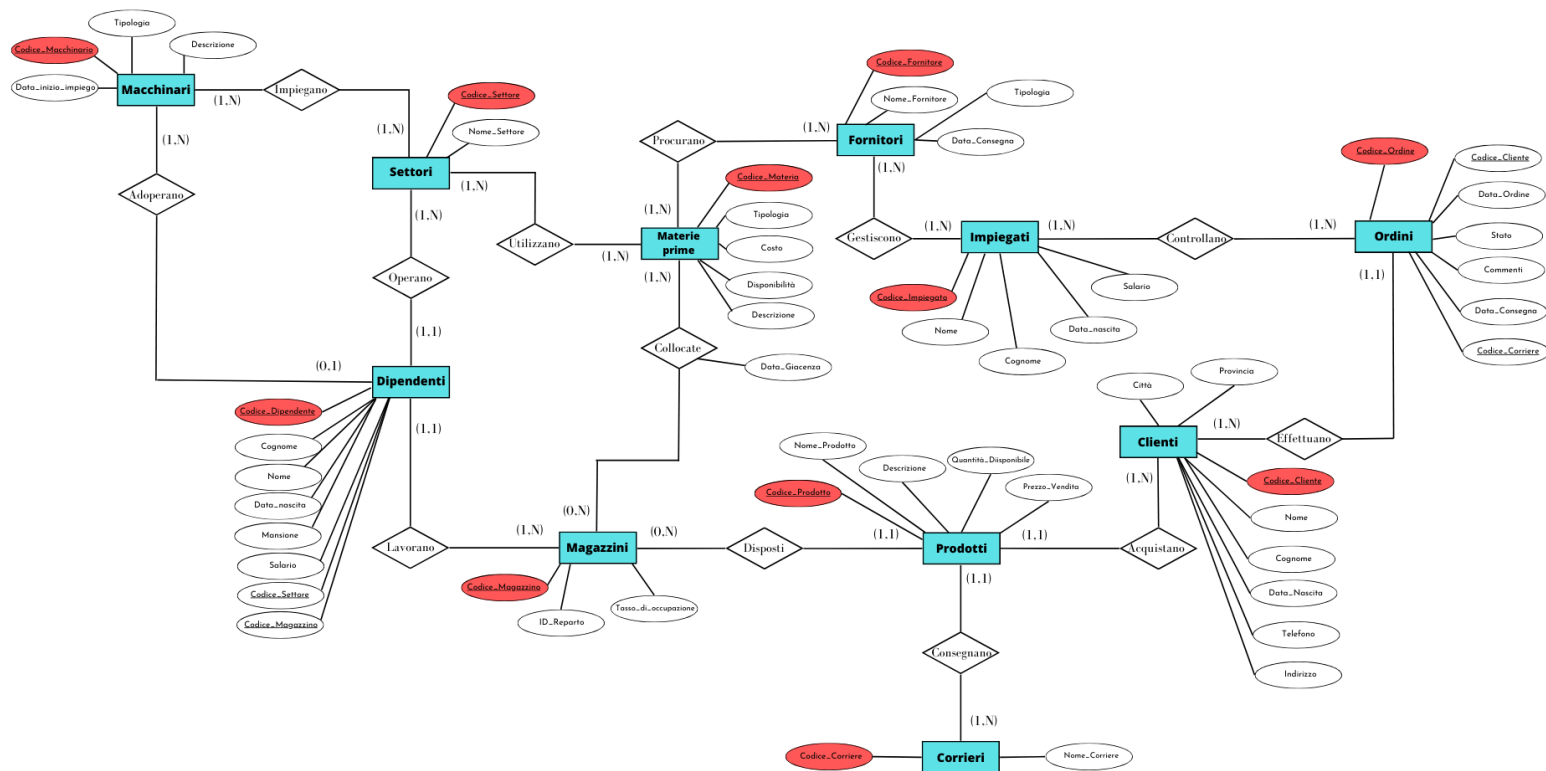
Capitolo 5

Tabella dei volumi

Concetto	Tipo	Volume
Adoperano	R	1000
Impiegano	R	200
Operano	R	600
Lavorano	R	400
Utilizzano	R	10000
Procurano	R	900
Disposti	R	5000
Consegnano	R	5000
Gestiscono	R	800
Controllano	R	5000
Effettuano	R	20000
Collocate	R	15000
Acquistano	R	10000
Macchinari	E	1000
Settori	E	10
Dipendenti	E	400
Materie prime	E	20000
Magazzini	E	15
Fornitori	E	300
Impiegati	E	100
Ordini	E	30000
Clienti	E	10000
Prodotti	E	20000
Corrieri	E	10

Capitolo 6

Schema Entity-Relationship Ristrutturato



Eliminazione delle generalizzazioni

Si ha una sola generalizzazione con entità padre Fornitori ed entità figlie rispettivamente Fornitore_MateriePrime e Fornitore_Accessori. Possiamo osservare che nessuna delle entità figlie presenta attributi diversi o relazioni separate. Inoltre, si ritiene che non si avrà interesse ad accedere a tali entità in maniera distinta e specifica.

Si decide, quindi, di eseguire un accorpamento nell'entità padre, andando ad aggiungere un attributo "Tipologia" nell'entità Fornitori.

Analisi ridondanze

Si è ritenuto necessario rimuovere dalle entità Ordini e Corrieri alcuni attributi che possono essere recuperati nell'entità Clienti. In particolare, all'entità Ordini sono stati rimossi gli attributi: telefono, indirizzo, città e provincia. All'entità Corrieri sono stati tolti gli attributi: indirizzo, città e provincia.

Eliminazione attributi composti

Sono stati rimossi gli attributi composti e si è deciso di accorpare in un unico attributo "indirizzo", tutte le informazioni necessarie collegate ad esso, come "Via" e "Numero civico".

Lo stesso è stato fatto per l'attributo Telefono, rimuovendo la distinzione "Numero fisso" e "Cellulare", che sono informazioni facilmente intuibili.

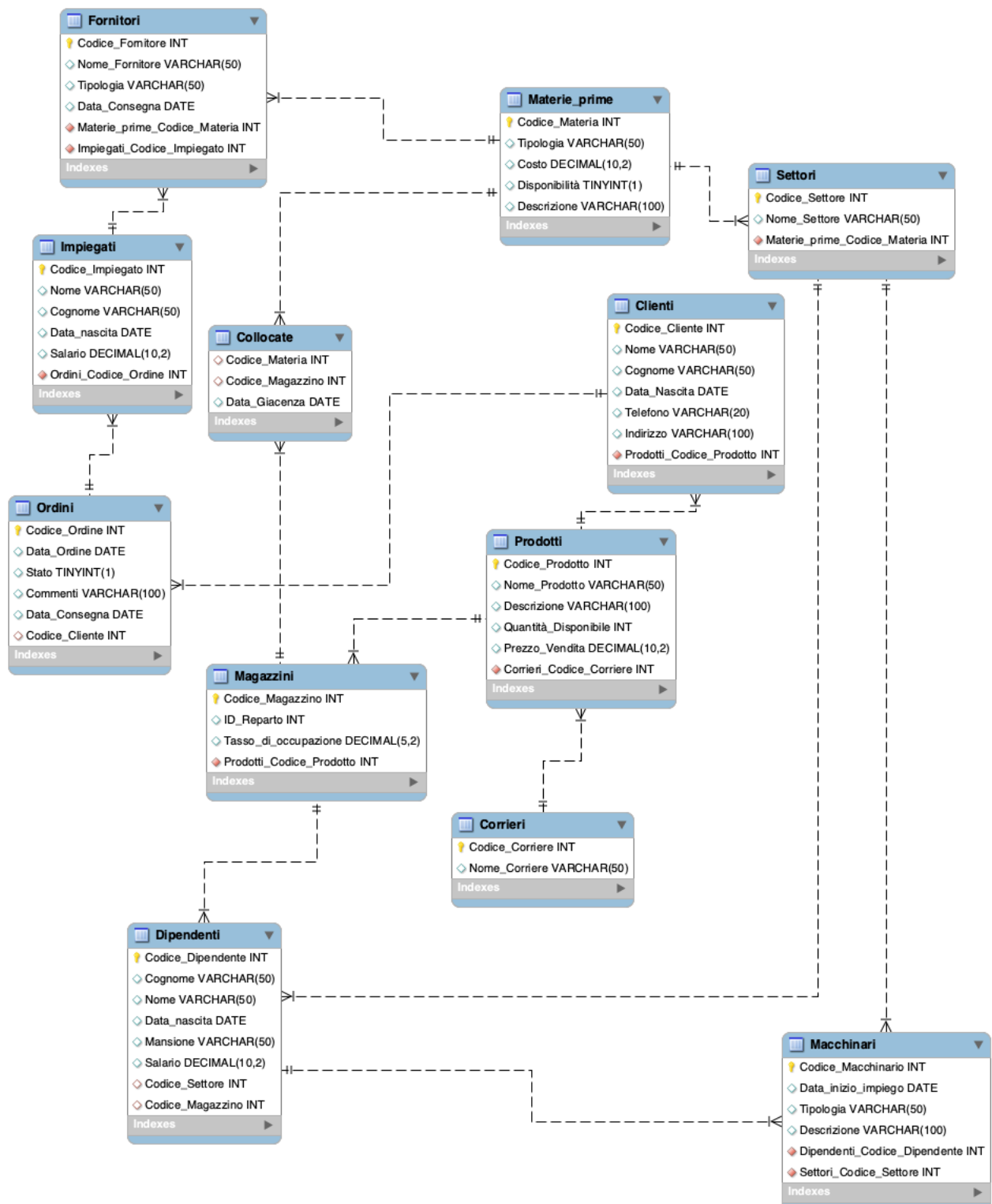
Scelta Identificatori Primari

Per ogni entità è stato evidenziato in rosso un identificatore primario. Rispettivamente, sono stati scelti:

- Macchinari: Codice_Macchinario
- Settori: Codice_Setto
- Dipendenti: Codice_Dipendente
- Materie prime: Codice_Materia
- Magazzini: Codice_Magazzino
- Fornitori: Codice_Fornitore
- Impiegati: Codice_Impiegato
- Ordini: Codice_Ordine
- Clienti: Codice_Cliente
- Prodotti: Codice_Prodotto
- Corrieri: Codice_Corriere

Capitolo 7

Schema Logico



Macchinari: (Codice_Macchinario, Data_inizio_impiego, Tipologia, Descrizione)

Settori: (Codice_Settore, Nome_Settore)

Materie prime: (Codice_Materia, Tipologia, Costo, Disponibilità, Descrizione)

Magazzini: (Codice_Magazzino, ID_Reparto, Tasso_di_occupazione)

Fornitori: (Codice_Fornitore, Nome_Fornitore, Tipologia, Data_Consegna)

Impiegati: (Codice_Impiegato, Nome, Cognome, Data_nascita, Salario)

Clienti: (Codice_Cliente, Nome, Cognome, Data_Nascita, Telefono, Indirizzo)

Prodotti: (Codice_Prodotto, Nome_Prodotto, Descrizione, Quantità_Disponibile, Prezzo_Vendita)

Dipendenti: (Codice_Dipendente, Cognome, Nome, Data_nascita, Mansione, Salario, Codice_Settore, Codice_Magazzino)

Corrieri: (Codice_Corriere, Nome_Corriere)

Ordini: (Codice_Ordine, Data_Ordine, Stato, Commenti, Data_Consegna, Codice_Cliente, Codice_Corriere)

Collocate: (Codice_Materia, Codice_Magazzino, Data_Giacenza)

Normalizzazione

La base di dati è già in prima forma normale, tutte le colonne sono atomiche. La base di dati è già in seconda forma normale, ciascuna colonna dipende dalla chiave primaria.

La base di dati è già in terza forma normale, ogni attributo dipende solo dalla chiave primaria.

Capitolo 8

Query SQL

Questo trigger è un esempio di come aggiornare automaticamente la quantità disponibile dei prodotti quando viene inserito un nuovo ordine nella tabella "Ordini". Si presuppone che la tabella "Prodotti" abbia una colonna "Quantità_Disponibile" che rappresenta la quantità attualmente disponibile del prodotto. Il trigger sottrae la quantità dell'ordine dalla quantità disponibile del prodotto corrispondente.

```
DELIMITER //
```

```
CREATE TRIGGER aggiorna_quantita_disponibile  
AFTER INSERT ON Ordini  
FOR EACH ROW  
BEGIN
```

```
    DECLARE prodotto_id INT;  
    DECLARE quantita_ordine INT;
```

```
#Ottieni l'ID del prodotto e la quantità dell'ordine appena inserito  
    SELECT Codice_Prodotto, Quantità_Disponibile  
    INTO prodotto_id, quantita_ordine  
    FROM Prodotti  
    WHERE Codice_Prodotto = NEW.Codice_Prodotto;
```

```
#Aggiorna la quantità disponibile del prodotto  
    UPDATE Prodotti  
    SET Quantità_Disponibile = quantita_ordine - NEW.Quantità  
    WHERE Codice_Prodotto = prodotto_id;  
END;  
//  
DELIMITER ;
```

Questo trigger chiamato TrgRifornimentoMagazzino viene attivato dopo ogni aggiornamento della tabella Magazzini. Il trigger verifica se il nuovo valore del Tasso_di_occupazione è inferiore al 50%. Se la condizione è vera, viene eseguita l'azione desiderata, che in questo caso è l'inserimento di una notifica nella tabella Notifiche indicando che è necessario un rifornimento di materie prime per il magazzino corrispondente.

```
DELIMITER //

CREATE TRIGGER TrgRifornimentoMagazzino
AFTER UPDATE ON Magazzini
FOR EACH ROW
BEGIN
    DECLARE tasso_occupazione DECIMAL(5, 2);

    #Calcola il nuovo tasso di occupazione
    SET tasso_occupazione = (NEW.Tasso_di_occupazione / 100) * 100;

    IF tasso_occupazione < 50 THEN
        #Esegui l'azione necessaria per il rifornimento di materie prime
        #In questo esempio, si può inviare una notifica o eseguire altre azioni specifiche

        INSERT INTO Notifiche (Messaggio)
        VALUES ('Rifornimento di materie prime necessario per il magazzino con codice '
    || NEW.Codice_Magazzino);
    END IF;
END;
//

DELIMITER ;
```

Questo trigger verifica la disponibilità di una materia prima prima dell'inserimento di una nuova riga nella tabella "Collocate". Si presuppone che la tabella "Materie_prime" abbia una colonna "Disponibilità" di tipo booleano che indica se la materia prima è disponibile o meno. Il trigger effettua una selezione per verificare la disponibilità della materia prima corrispondente al codice inserito nella collocata. Se la materia prima non è disponibile, il trigger genera un'eccezione e annulla l'inserimento della riga nella tabella "Collocate".

```
DELIMITER //
```

```
CREATE TRIGGER verifica_disponibilita_materia
BEFORE INSERT ON Collocate
FOR EACH ROW
BEGIN
    DECLARE disponibilita BOOLEAN;

    #Verifica la disponibilità della materia prima
    SELECT Disponibilita
    INTO disponibilita
    FROM Materie_prime
    WHERE Codice_Materia = NEW.Codice_Materia;

    #Se la materia prima non è disponibile, annulla l'inserimento della collocata
    IF disponibilita = FALSE THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La materia prima non è disponibile.';
    END IF;
END;
//
DELIMITER ;
```

Stored procedure per ottenere informazioni sui macchinari in base alla tipologia:

```
DELIMITER //
```

```
CREATE PROCEDURE GetMacchinariByTipologia(IN tipologia VARCHAR(50))
BEGIN
    SELECT *
    FROM Macchinari
    WHERE Tipologia = tipologia;
END;
//

DELIMITER ;
```

Stored procedure per aggiornare il prezzo di vendita di un prodotto:

DELIMITER //

```
CREATE PROCEDURE UpdatePrezzoVendita(IN prodotto_id INT, IN nuovo_prezzo
DECIMAL(10, 2))
BEGIN
    UPDATE Prodotti
    SET Prezzo_Vendita = nuovo_prezzo
    WHERE Codice_Prodotto = prodotto_id;
END;
//
```

DELIMITER ;

Stored procedure per inserire un nuovo ordine con i relativi dettagli:

DELIMITER //

```
CREATE PROCEDURE InserisciNuovoOrdine(
    IN data_ordine DATE,
    IN stato BOOLEAN,
    IN commenti VARCHAR(100),
    IN data_consegna DATE,
    IN codice_cliente INT,
    IN codice_corriere INT,
    IN ProdottiTableType
)
BEGIN
    DECLARE nuovo_ordine_id INT;

    #Inserimento dell'ordine
    INSERT INTO Ordini (Data_Ordine, Stato, Commenti, Data_Consegna, Codice_Cliente,
Codice_Corriere)
    VALUES (data_ordine, stato, commenti, data_consegna, codice_cliente,
codice_corriere);

    #Ottieni l'ID dell'ordine appena inserito
    SET nuovo_ordine_id = LAST_INSERT_ID();

    #Inserimento dei dettagli dell'ordine
    INSERT INTO DettagliOrdine (Codice_Ordine, Codice_Prodotto, Quantità)
    SELECT nuovo_ordine_id, prodotto_id, quantità
    FROM prodotti_table_type;
END;
//
```

DELIMITER ;

Stored procedure per calcolare il salario medio dei dipendenti in un determinato settore:

DELIMITER //

```
CREATE PROCEDURE CalcolaSalarioMedioSettore(IN settore_id INT, OUT salario_medio
DECIMAL(10, 2))
BEGIN
    SELECT AVG(Salario) INTO salario_medio
    FROM Dipendenti
    WHERE Codice_Settore = settore_id;
END;
//
```

DELIMITER ;

Stored procedure degli ordini effettuati di un cliente nel mese di maggio:

DELIMITER //

```
CREATE PROCEDURE GetOrdiniClienteMaggio(IN codice_cliente INT)
BEGIN
    SELECT *
    FROM Ordini
    WHERE Codice_Cliente = codice_cliente
    AND MONTH(Data_Ordine) = 5;
END;
//
```

DELIMITER ;