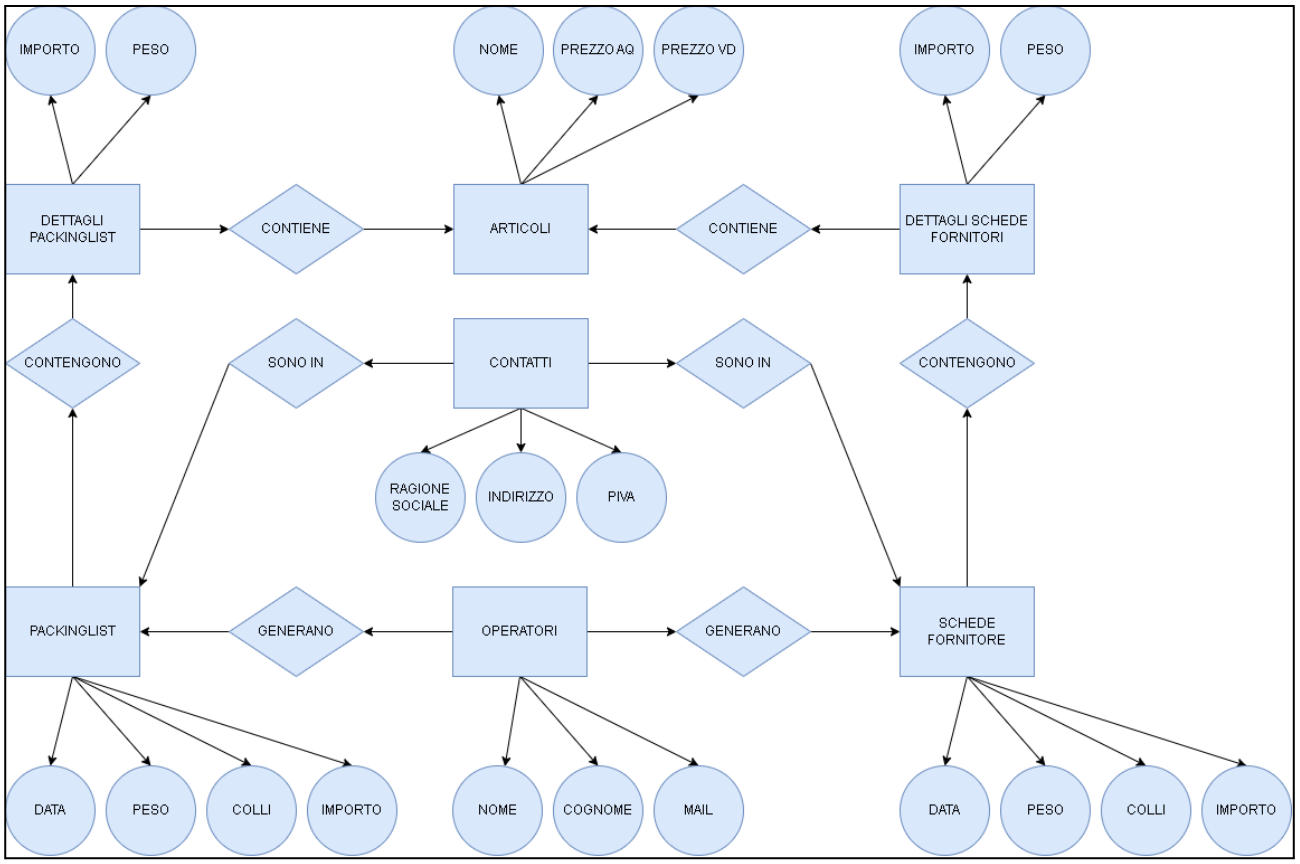
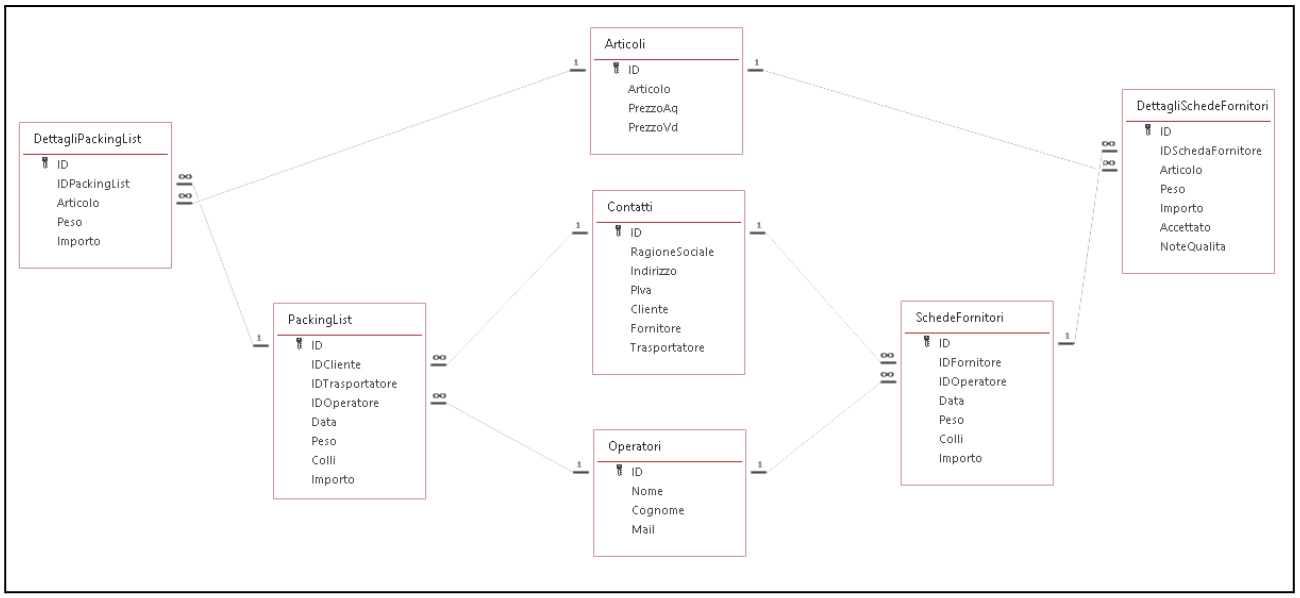
DESCRIZIONE CASO STUDIO:
Con il presente documento si illustra una bozza di un gestionale che permetta ad un'azienda che acquista e rivende prodotti a peso, di gestire tramite schede fornitori per riscontrare il peso e la qualità della merce acquistata dai fornitori e creare packing lists per i clienti.

MODELLO ER:



MODELLO LOGICO:



PROGETTAZIONE FISICA:

CREAZIONE TABELLA ARTICOLI	CREATE TABLE `gestionale`.`articoli` (`ID` INT NOT NULL AUTO_INCREMENT, `Articolo` VARCHAR(45) NULL, `PrezzoAq` DECIMAL(2) NULL, `PrezzoVd` DECIMAL(2) NULL, PRIMARY KEY (`ID`)) COMMENT = '';
CREAZIONE TABELLA CONTATTI	CREATE TABLE `gestionale`.`contatti` (`ID` INT NOT NULL AUTO_INCREMENT, `RagioneSociale` VARCHAR(45) NULL, `Indirizzo` VARCHAR(45) NULL, `Piva` INT NULL, `Cliente` TINYINT(1) NOT NULL DEFAULT 0, `Fornitore` TINYINT(1) NOT NULL DEFAULT 0, `Trasportatore` TINYINT(1) NOT NULL DEFAULT 0, PRIMARY KEY (`ID`), UNIQUE INDEX `Piva_UNIQUE` (`Piva` ASC) VISIBLE);
CREAZIONE TABELLA OPERATORI	CREATE TABLE `gestionale`.`operatori` (`ID` INT NOT NULL AUTO_INCREMENT, `Nome` VARCHAR(45) NOT NULL, `Cognome` VARCHAR(45) NOT NULL, `Mail` VARCHAR(45) NOT NULL, PRIMARY KEY (`ID`));
CREAZIONE TABELLA PACKINGLIST	CREATE TABLE `gestionale`.`packinglist` (`ID` INT NOT NULL AUTO_INCREMENT, `IDCliente` INT NOT NULL, `IDTrasportatore` INT NOT NULL, `IDOperatore` INT NOT NULL, `Data` DATETIME NOT NULL, `Peso` DECIMAL(2) NULL, `Colli` INT NULL, `Importo` DECIMAL(2) NULL, PRIMARY KEY (`ID`), INDEX `ID_idx` (`IDCliente` ASC) VISIBLE, INDEX `ID_idx1` (`IDTrasportatore` ASC) VISIBLE, INDEX `IDOperatore_idx` (`IDOperatore` ASC) VISIBLE, CONSTRAINT `IDCliente` FOREIGN KEY (`IDCliente`) REFERENCES `gestionale`.`contatti` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION, CONSTRAINT `IDTrasportatore` FOREIGN KEY (`IDTrasportatore`) REFERENCES `gestionale`.`contatti` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION, CONSTRAINT `IDOperatore` FOREIGN KEY (`IDOperatore`) REFERENCES `gestionale`.`operatori` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION);
CREAZIONE TABELLA	CREATE TABLE `gestionale`.`dettaglipackinglist` (

DETTAGLIOPACKINGLIST	`ID` INT NOT NULL AUTO_INCREMENT, `IDPackingList` INT NOT NULL, `Articolo` INT NOT NULL, `Peso` DECIMAL(2) NULL DEFAULT 0, `Importo` DECIMAL(2) NULL, PRIMARY KEY (`ID`), INDEX `IDP_idx` (`IDPackingList` ASC) VISIBLE, CONSTRAINT `IDP` FOREIGN KEY (`IDPackingList`) REFERENCES `gestionale`.`packinglist` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION);
CREAZIONE TABELLA SCHEDEFORNITORI	CREATE TABLE `gestionale`.`schedefornitori` (`ID` INT NOT NULL AUTO_INCREMENT, `IDFornitore` INT NOT NULL, `IDOperatore` INT NOT NULL, `Data` DATETIME NOT NULL, `Peso` DECIMAL(2) NULL, `Colli` INT NULL, `Importo` DECIMAL(2) NULL, PRIMARY KEY (`ID`), INDEX `ID_idx` (`IDFornitore` ASC) VISIBLE, INDEX `IDoperatore_idx` (`IDOperatore` ASC) VISIBLE, CONSTRAINT `IDFornitore` FOREIGN KEY (`IDFornitore`) REFERENCES `gestionale`.`contatti` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION, CONSTRAINT `IDO` FOREIGN KEY (`IDOperatore`) REFERENCES `gestionale`.`operatori` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION);
CREAZIONE TABELLA DETTAGLIOSCHEDEFORNITORI	CREATE TABLE `gestionale`.`dettaglischedefornitori` (`ID` INT NOT NULL AUTO_INCREMENT, `IDSchedaFornitore` INT NOT NULL, `Articolo` INT NOT NULL, `Peso` DECIMAL(2) NULL DEFAULT 0, `Importo` DECIMAL(2) NULL, `Accettato` TINYINT(1) NULL DEFAULT 0, `NoteQualita` LONGTEXT NULL, PRIMARY KEY (`ID`), INDEX `IDSF_idx` (`IDSchedaFornitore` ASC) VISIBLE, INDEX `IDA_idx` (`Articolo` ASC) VISIBLE, CONSTRAINT `IDSF` FOREIGN KEY (`IDSchedaFornitore`) REFERENCES `gestionale`.`contatti` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION, CONSTRAINT `IDA` FOREIGN KEY (`Articolo`) REFERENCES `gestionale`.`articoli` (`ID`) ON DELETE NO ACTION ON UPDATE NO ACTION);

10 INTERROGAZIONI:

questa query aggiorna l''importo nella tabella dettaglipackinglist moltiplicando il peso dell''articolo per il prezzo di vendita nella tabella articoli	UPDATE articoli INNER JOIN dettaglipackinglist ON articoli.ID = dettaglipackinglist.Articolo SET dettaglipackinglist.Importo = dettaglipackingList.Peso*articoli.PrezzoVd #'1 - questa query aggiorna l''importo nella tabella dettaglipackinglist moltiplicando il peso dell''articolo per il prezzo di vendita nella tabella articoli';
questa query aggiorna l'importo dell'articolo presente nella tabella dettaglischedefornitori moltiplicando il peso della tabella dettaglischedefornitori per il prezzo di acquisto presente nella tabella articoli	UPDATE articoli INNER JOIN dettaglischedefornitori ON articoli.ID = dettaglischedefornitori.Articolo SET dettaglischedefornitori.Importo = dettaglischedefornitori.Peso*articoli.PrezzoAq; #'2 - questa query aggiorna l'importo dell'articolo presente nella tabella dettaglischedefornitori moltiplicando il peso della tabella dettaglischedefornitori per il prezzo di acquisto presente nella tabella articoli;
questa query crea una tabella schedefornitoritemp che conterrà i totali di dettaglischedefornitori relativi a schedefornitori, che successivamente utilizzeremo per aggiornare schede fornitori	SELECT dettaglischedefornitori.IDSchedaFornitore, Count(dettaglischedefornitori.Articolo) AS ConteggioDiArticolo, Sum(dettaglischedefornitori.Peso) AS SommaDiPeso, Sum(dettaglischedefornitori.Importo) AS SommaDiImporto INTO schedefornitoritemp FROM dettaglischedefornitori GROUP BY dettaglischedefornitori.IDSchedaFornitore #'3 - questa query crea una tabella schedefornitoritemp che conterrà i totali di dettaglischedefornitori relativi a schedefornitori, che successivamente utilizzeremo per aggiornare schede fornitori';
questa query aggiorna i totali all'interno della tabella schede fornitori	UPDATE schedefornitoritemp INNER JOIN schedefornitori ON schedefornitoritemp.IDSchedaFornitore = schedefornitori.ID SET schedefornitori.Peso = schedefornitoritemp.SommaDiPeso, schedefornitori.Colli = schedefornitoritemp.ConteggioDiArticolo, schedefornitori.Importo = schedefornitoritemp.SommaDiImporto; #'4 - questa query aggiorna i totali all'interno della tabella schede fornitori';
questa query crea una tabella packinglisttemp che conterrà i totali di dettaglipackinglist relativi a packinglist, che successivamente utilizzeremo per aggiornare packinlist	SELECT dettaglipackingList.IDPackingList, Count(dettaglipackingList.Articolo) AS ConteggioDiArticolo, Sum(dettaglipackingList.Peso) AS SommaDiPeso, Sum(dettaglipackingList.Importo) AS SommaDiImporto INTO packinglisttemp FROM dettaglipackingList GROUP BY dettaglipackingList.IDPackingList #'5 - questa query crea una tabella packinglisttemp che conterrà i totali di dettaglipackinglist relativi a packinglist, che successivamente utilizzeremo per aggiornare packinlist';
questa query aggiorna i totali all'interno di packinglist	UPDATE packinglisttemp INNER JOIN packinglist ON packinglisttemp.IDPackingList = packinglist.ID SET packinglist.Peso = packinglisttemp.SommaDiPeso, packinglist.Colli = packinglisttemp.ConteggioDiArticolo, packinglist.Importo = packinglisttemp.SommaDiImporto; #'6 - questa query aggiorna i totali all'interno di packinglist';

questa query visualizza il totale degli importi delle schede fornitori per fornitore	SELECT contatti.RagioneSociale, Sum(schedefornitori.Importo) AS SommaDiImporto FROM contatti INNER JOIN schedefornitori ON contatti.ID = schedefornitori.IDFornitore GROUP BY contatti.RagioneSociale; #'7 - questa query visualizza il totale degli importi delle schede fornitori per fornitore';
questa query visualizza il totale degli importi e dei pesi di packinglist per cliente	SELECT contatti.RagioneSociale, Sum(packinglist.Importo) AS SommaDiImporto, Sum(packinglist.Peso) AS SommaDiPeso FROM contatti INNER JOIN packinglist ON (contatti.ID = packinglist.IDTrasportatore) AND (contatti.ID = packinglist.IDCliente) GROUP BY contatti.RagioneSociale #'8 - questa query visualizza il totale degli importi e dei pesi di packinglist per cliente';
questa query visualizza il numero di schede fornitore ed il totale dell'importo per operatore	SELECT operatori.ID, Count(schedefornitori.ID) AS ConteggioDiID, Sum(schedefornitori.Importo) AS SommaDiImporto FROM operatori INNER JOIN schedefornitori ON operatori.ID = schedefornitori.IDOperatore GROUP BY operatori.ID #'9 - questa query visualizza il numero di schede fornitore ed il totale dell'importo per operatore';
questa query visualizza il totale del peso degli articoli venduti per articolo	SELECT Sum(dettaglipackingList.Peso) AS SommaDiPeso, articoli.Articolo FROM dettaglipackingList LEFT JOIN articoli ON dettaglipackingList.Articolo = articoli.ID GROUP BY articoli.Articolo #'10 - questa query visualizza il totale del peso degli articoli venduti per articolo';