

Progetto di Basi di dati

Negozio di Riparazioni apparecchiature per lo spettacolo.

Studente: Carnevale Luigi.

Matricola: 0512119029.

Progetto di basi di dati

- Negozio di riparazioni per apparecchiature dedicate allo spettacolo -

Si vuole progettare una base di dati per la gestione di un negozio che si occupa dell'assistenza per tutte le tipologie di apparecchiature dedicate al mondo dello spettacolo ed anche di elettronica più nel generico. Nel negozio lavora il titolare, da solo, senza dipendenti; quindi, si occupa della gestione dell'attività e fornisce anche l'assistenza di cui i clienti necessitano.

Il negozio preso in esame effettua riparazioni per apparecchiature dedicate allo spettacolo ed elettronica varia; La tipologia di attrezzatura è dedicata all'audio (casse, subwoofer, amplificatori, ecc...) oppure attrezzatura da illuminazione (faretti, teste mobili, ecc...) ma anche apparecchi di altro tipo sempre dedicate all'intrattenimento (macchine per il fumo, macchine per le bolle, fontane fredde, ho considerato anche cellulari o elettronica più in generale).

Il cliente entra in negozio con l'oggetto per cui necessita assistenza o comunque riparazioni; egli espone al titolare il malfunzionamento/difetto e richiede un preventivo in base all'entità del problema, il quale contiene tutte le informazioni riguardanti il lavoro da svolgere, il tipo di riparazione/sostituzione (per tenere traccia e controllo del lavoro da svolgere), successivamente può decidere di effettuare la prestazione necessaria per risolvere il problema oppure ritirare il prodotto così come lo aveva consegnato.

Il titolare, in caso di riscontro positivo al preventivo presentato, procede a contattare i fornitori che dovranno inviare i componenti da sostituire per effettuare la prestazione, se questi non sono al momento disponibili in negozio. Avendo in negozio i componenti necessari, si procede ad effettuare il lavoro accordato, salvo imprevisti o aggiunte, viene calcolato il totale che il cliente spenderà per la riparazione che ha richiesto, il quale è presente nel preventivo iniziale ma può variare durante il lavoro.

Per ogni cliente è possibile che si abbiano più apparecchi in attesa di assistenza, così come potrebbe esserci una "lista di attesa", questa si crea quando molti clienti consegnano attrezzatura e in ordine di urgenza e disponibilità dei componenti necessari si accumulano le riparazioni da effettuare, le quali verranno svolte però comunque in ordine di consegna (il primo cliente che ha consegnato sarà il primo ad essere assistito).

//Ricavo adesso le entità interessate dal progetto:

1. Cliente (Codice Fiscale Cliente, Nome, Cognome, TelefonoCellulare, E-Mail, Indirizzo Residenza)
2. Apparecchio (ID, Apparecchio, Tipo, Marca, Modello, Codice Fiscale Cliente)
3. Difetto (ID Difetto, ID Apparecchio, Descrizione del Difetto)
4. Preventivo (ID Preventivo, IDs Difetto, IDs Componenti, Data Creazione, DescrizioneDifetto, Accettato(boolean))
5. ListaDiAttesa(ID Lista, ID Preventivo, Data Inserimento)
6. Riparazione (ID Riparazione, Data Inserimento, Data Fine Lavori, ID Lista)
7. Pagamento (ID Pagamento, ID Riparazione, Data Pagamento, Importo, Pagamento Elettronico(boolean), TotaleIncasso)
8. Componente (ID Componente, MarcaComponente, ModelloComponente, Costo Acquisto, QtaDeposito)
9. Fornitore (Azienda, Sede)
10. OrdineComponenti (ID Ordine, ID Componente, Data Ordine, Data Ricezione)

//Adesso elenco tutte le varie relazioni fra le entità:

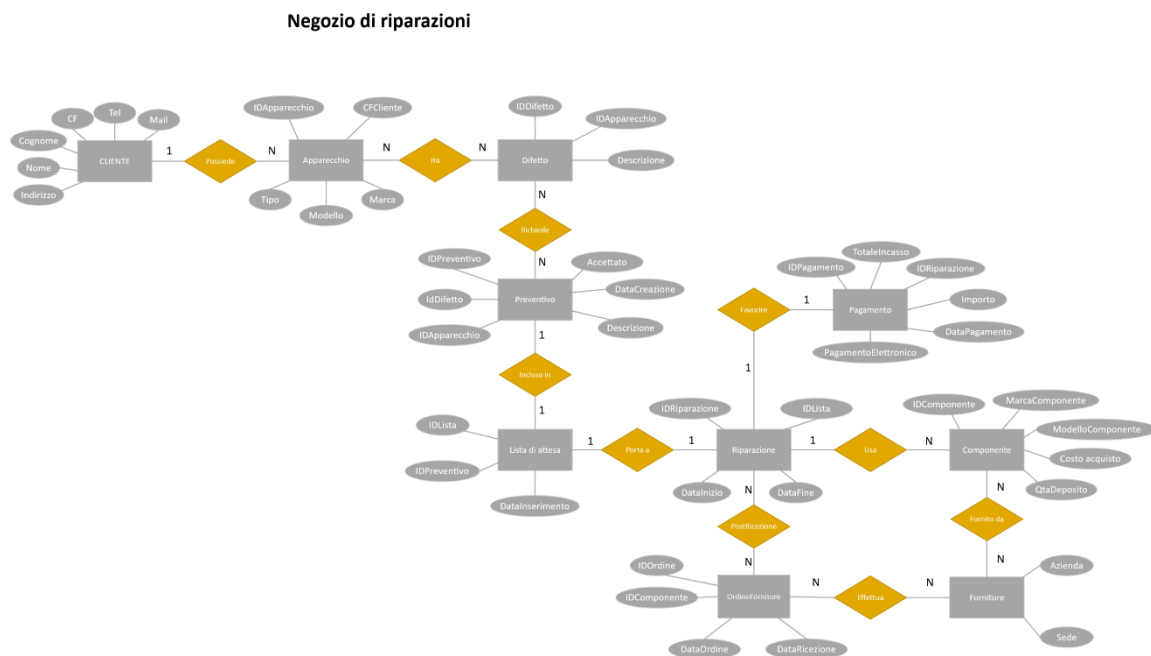
1. Un CLIENTE possiede uno o più APPARECCHI;
2. Un APPARECCHIO ha un DIFETTO;
3. Ogni DIFETTO richiede un PREVENTIVO;
4. Un PREVENTIVO se accettato viene incluso in LISTA DI ATTESA;
5. La LISTA DI ATTESA porta alla RIPARAZIONE;
6. La RIPARAZIONE usa COMPONENTI;
7. I COMPONENTI se non disponibili vanno ordinati dal FORNITORE;
8. Tramite il FORNITORE si effettua l'ORDINE COMPONENTI per i componenti non disponibili in negozio;
9. Post ricezione dell'ORDINE COMPONENTI si effettua la RIPARAZIONE.

GLOSSARIO:

1. Id: Codice identificativo univoco
2. Cliente: Ogni persona che entra in negozio con un apparecchio danneggiato o difettoso;
3. Apparecchio: Gli oggetti che vengono consegnati in negozio per poter ricevere assistenza;

4. Lista di attesa: Lista che si viene a creare quando sono presenti più preventivi accettati, questi vengono riposti in lista e tutte le riparazioni vengono effettuate in ordine cronologico di arrivo.
5. Preventivo: Prospetto di costi riguardo il prodotto su cui si necessita assistenza, comprensivo anche di eventuali ordini per componenti non disponibili e tipo di intervento effettuato;
6. Componenti: Le parti difettose o danneggiate di un apparecchio;
7. Riparazione: L'intervento effettuato sugli apparecchi consegnati
8. Pagamento: Somma di denaro necessaria per l'intervento richiesto.

Questa è la rappresentazione dello schema ER che ho descritto precedentemente:



//Vincoli non esprimibili nello schema:

1. Un cliente non può consegnare lo stesso apparecchio due volte senza prima averlo ritirato;
2. Le riparazioni partono dalla lista di attesa e devono essere svolte in ordine cronologico;
3. Le date devono avere un formato valido (gg-mm-aaaa);
4. I pagamenti non possono avere valore negativo;
5. Un pagamento deve essere associato ad una riparazione completata;
6. La quantità dei componenti disponibili in deposito non può avere valore negativo

//Progettazione logica:

Procediamo verso lo studio del carico applicativo e la traduzione verso il modello logico, mediante un'eventuale ristrutturazione dello schema ER che abbiamo appena costruito e rappresentato.

Le entità non hanno bisogno di modifiche, dato che non presentano valori multipli.

Per quanto riguarda gli attributi multi-valore invece, adottato questa soluzione:

In Cliente – Indirizzo (Città-Via-Cap), lo scompongo nei singoli attributi;

In Preventivo – DataCreazione (ggCreazione-mmCreazione-aaaaCreazione), lo scompongo nei singoli attributi;

In Riparazione – DataInizio (ggInizio-mmInizio-aaaInizio), lo scompongo nei singoli attributi;

In Riparazione – DataFine (ggFine-mmFine-aaaaFine), lo scompongo nei singoli attributi;

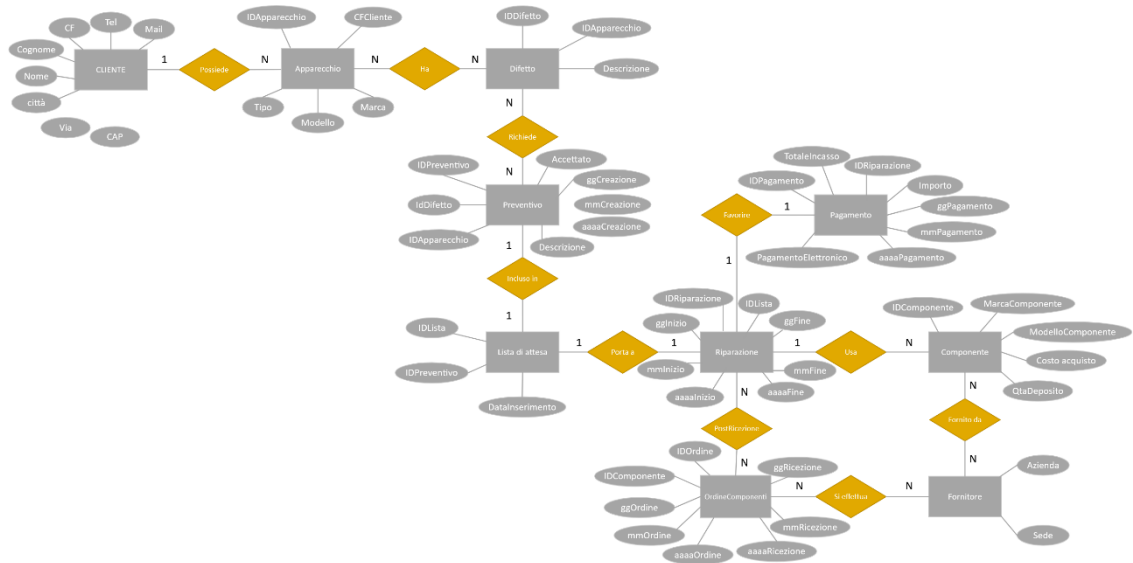
In OrdineComponenti – DataOrdine (ggOrdine-mmOrdine-aaaaOrdine), lo scompongo nei singoli attributi;

In OrdineComponenti – DataRicezione (ggRicezione-mmRicezione-aaaaRicezione), lo scompongo nei singoli attributi;

In Pagamento – DataPagamento (ggPagamento-mmPagamento-aaaaPagamento), lo scompongo nei singoli attributi.

Lo schema risultante è il seguente:

Negozi di riparazioni



//Le operazioni che verranno effettuate maggiormente nel DataBase sono:

- OP1: Inserimento nuovo Cliente (4G) (I);
- OP2: Inserimento nuovo Apparecchio (5G) (I);
- OP3: Inserimento nuovo Preventivo (3G) (I);
- OP4: Inserimento nuova Riparazione (3G) (I);
- OP5: Inserimento nuovo Componente (5M) (I);
- OP6: Effettuare nuovo OrdineComponenti(10M) (B);
- OP6: Inserimento nuovo Pagamento (2G) (I);
- OP7: Calcolo finale incassi (1A);

//Adesso analizziamo la Tavola dei Volumi:

Concetto	Tipo	Volume
Cliente	E	1.000
Apparecchio	E	5.000
Difetto	E	10.000
Preventivo	E	5.000
Lista Di Attesa	E	1.000
Riparazione	E	4.000
Pagamento	E	4.000

Componente	E	50.000	
Fornitore	E	10.000	
Ordine Componenti	E	2.000	
Possiede	R	5.000	Ogni cliente ha 1-5 apparecchi
Ha	R	10.000	Ogni apparecchio può avere più difetti
Richiede	R	5.000	Non tutti i difetti hanno un preventivo
Incluso in	R	1.000	Solo i preventivi accettati
Porta a	R	4.000	Alcuni preventivi vengono annullati
Usa	R	8.000	Una riparazione può richiedere più componenti
Fornito da	R	50.000	Ogni componente ha più fornitori
Si effettua	R	2.000	Ogni ordine coinvolge 1-3 fornitori
Post ricezione	R	2.000	Solo alcune riparazioni necessitano di ordini
Favorire	R	4.000	Ogni riparazione ha un pagamento

Abbiamo un attributo derivabile:

- Incasso Totale → (Pagamento)

Nelle operazioni elencate precedentemente abbiamo OP7 che sfrutta questo attributo.

Operazione 7 CON DATO DERIVABILE			
Concetto	Tipo	Accessi	Tipo
Pagamento	E	4000	L
Riparazione->Pagamento	R	4000	L

Operazione 7 SENZA DATO DERIVABILE			
Concetto	Tipo	Accessi	Tipo
Pagamento	E	4000	L

In questo caso possiamo notare come passeremo in medi da 8000 letture a circa 1000, ma se calcoliamo il costo in termini di memoria dove se contiamo 4byte per ogni campo “Totale” di ogni record di “Pagamento”, in 4000 record accumuleremo 16000 byte di dati e poi se calcoliamo che l` OP4 andrebbe modificata per far si che sia lei a memorizzare di volta in volta il dato “Totale” che con una media di 5 prodotti da riparare per ordine porterebbe ad un aumento di 10 letture ed una scrittura al giorno, che moltiplicate per 365 giorni porterebbero a 3650 letture extra e 365 scritture, pur mantenendo le 4000

letture già, precedentemente considerate, per ricavare il dato sfruttando OP4. Tutto ciò graverebbe sicuramente di più sulla Base di Dati rispetto alle 8000 letture da eseguire. Ne conveniamo che in questo caso conviene mantenere l'attributo derivabile.

//Ricaviamo adesso lo schema relazionale:

- Cliente (**Codice Fiscale Cliente**, Nome, Cognome, TelefonoCellulare, E-Mail, Città, Via, Cap)
- Apparecchio (**ID Apparecchio**, Tipo, Marca, Modello, Codice Fiscale Cliente↑)
- Difetto (**ID Difetto**, ID Apparecchio, DescrizioneDifetto)
- Preventivo (**ID Preventivo**, ID Difetto, ID Apparecchio, ggCreazione, mmCreazione, aaaaCreazione, Accettato, DescrizioneDifetto)
- Lista di attesa (**ID Lista**, ID Preventivo↑, DataInserimento)
- Riparazione (**ID Riparazione**, ID Lista↑, ggInizio, mmInizio, aaaaInizio, ggFine, mmFine, aaaaFine)
- Componente (**ID Componente**, MarcaComponente, ModelloComponente, Costo Acquisto, QtaDeposito)
- Fornitore (**Azienda**, Sede)
- OrdineComponenti (**ID Ordine**, ID Componente↑, ggOrdine, mmOrdine, aaaaOrdine, ggRicezione, mmRicezione, aaaaRicezione)
- Pagamento (**ID Pagamento**, ID Riparazione↑, ggPagamento, mmPagamento, aaaaPagamento, Importo, Pagamento Elettronico(boolean), Incasso Totale)

//Normalizzazione

Il database si presenta già normalizzato.

È infatti già in prima forma normale (1NF) in quanto tutti gli attributi sono atomici dopo la ristrutturazione (sono stati infatti eliminati gli attributi multi-valore dove presenti). È in seconda forma normale (2NF) perché oltre ad essere in prima forma, tutti gli attributi non chiave dipendono interamente dalla chiave primaria in ciascuna tabella e inoltre non ci sono dipendenze parziali nelle tabelle. È in terza forma normale (3NF) perché oltre ad essere in seconda forma normale, ogni attributo non chiave dipende solo dalla chiave primaria della tabella alla quale appartiene. È presente un singolo attributo derivabile ma questo è

giustificato dalla analisi sulle operazioni svolta precedentemente; quindi è documentata anche la motivazione dietro alla scelta di mantenere tale attributo.

Procedo Adesso alla costruzione del DataBase e la creazione delle Query:

//Di seguito il codice SQL:

```
-- Disabilitare temporaneamente le foreign key
```

```
SET FOREIGN_KEY_CHECKS = 0;
```

```
-- Eliminare le tabelle
```

```
DROP TABLE IF EXISTS Pagamento;
```

```
DROP TABLE IF EXISTS Ordine_Riparazione;
```

```
DROP TABLE IF EXISTS Fornisce;
```

```
DROP TABLE IF EXISTS OrdineComponenti;
```

```
DROP TABLE IF EXISTS Fornitore;
```

```
DROP TABLE IF EXISTS Componente;
```

```
DROP TABLE IF EXISTS Riparazione;
```

```
DROP TABLE IF EXISTS ListaDiAttesa;
```

```
DROP TABLE IF EXISTS Preventivo;
```

```
DROP TABLE IF EXISTS Difetto;
```

```
DROP TABLE IF EXISTS Apparecchio;
```

```
DROP TABLE IF EXISTS Cliente;
```

```
-- Riabilitare le foreign key
```

```
SET FOREIGN_KEY_CHECKS = 1;
```

```
-- Creazione tabelle
```

```
CREATE TABLE Cliente (
```

```
    CodiceFiscale CHAR(16) PRIMARY key,
```

```
    Nome VARCHAR(50) not null,
```

```
    Cognome VARCHAR(50) not null,
```

```
    TelefonoCellulare VARCHAR(15),
```

```
    Email VARCHAR(50),
```

```
Città VARCHAR(50),  
Via VARCHAR(100),  
CAP CHAR(5)
```

```
);
```

```
CREATE TABLE Apparecchio (  
    ID INT auto_increment PRIMARY key,  
    Tipo VARCHAR(50) not null,  
    Marca VARCHAR(50),  
    Modello VARCHAR(50),  
    CodiceFiscaleCliente CHAR(16),  
    FOREIGN key (CodiceFiscaleCliente) references Cliente(CodiceFiscale) on delete cascade  
);
```

```
CREATE TABLE Difetto (  
    ID INT auto_increment Primary key,  
    IDApparecchio INT,  
    DescrizioneDifetto VARCHAR(255),  
    FOREIGN key (IDApparecchio) references Apparecchio(ID) on delete cascade  
);
```

```
CREATE TABLE Preventivo (  
    ID INT auto_increment PRIMARY key,  
    IDDifetto INT,  
    IDApparecchio INT,  
    DataCreazione DATE not null,  
    Accettato BOOLEAN not null default FALSE,  
    DescrizioneDifetto VARCHAR(255),  
    FOREIGN key (IDDifetto) references Difetto(ID) on delete cascade,  
    FOREIGN key (IDApparecchio) references Apparecchio(ID) on delete cascade  
);
```

```
CREATE TABLE ListaDiAttesa (  
    ID INT auto_increment PRIMARY key,  
    IDPreventivo INT unique,  
    DataInserimento DATE not null,  
    FOREIGN key (IDPreventivo) references Preventivo(ID) on delete cascade  
);
```

```
CREATE TABLE Riparazione (  
    ID INT auto_increment PRIMARY key,  
    IDLista INT unique,  
    DataInizio DATE not null,  
    DataFine DATE,  
    FOREIGN key (IDLista) references ListaDiAttesa(ID) on delete cascade  
);
```

```
CREATE TABLE Componente (  
    ID INT auto_increment PRIMARY key,  
    MarcaComponente VARCHAR(50),  
    ModelloComponente VARCHAR(50),  
    CostoAcquisto DECIMAL(10,2) not null,  
    QtaDeposito INT default 0 check (QtaDeposito >= 0)  
);
```

```
CREATE TABLE Fornitore (  
    ID INT auto_increment PRIMARY key,  
    Azienda VARCHAR(100) not null,  
    Sede VARCHAR(100)  
);
```

```
CREATE TABLE OrdineComponenti (  
    ID INT auto_increment Primary key,  
    DataOrdine DATE not null,  
    DataRicezione DATE  
);
```

```
CREATE TABLE Pagamento (  
    ID INT auto_increment PRIMARY key,  
    IDRiparazione INT unique,  
    DataPagamento DATE not null,  
    Importo DECIMAL(10,2) not null check (Importo >= 0),  
    PagamentoElettronico BOOLEAN not null default FALSE,  
    IncassoTotale DECIMAL(10,2),  
    FOREIGN key (IDRiparazione) references Riparazione(ID) on delete cascade  
);
```

-- Tabella relazione tra "Fornitore" e "OrdineComponenti" (N:N)

```

CREATE TABLE Fornisce (
    IDOrdine INT,
    IDFornitore INT,
    PRIMARY key (IDOrdine, IDFornitore),
    FOREIGN key (IDOrdine) references OrdineComponenti(ID) on delete cascade,
    FOREIGN key (IDFornitore) references Fornitore(ID) on delete cascade
);

```

-- Tabella relazione tra "OrdineComponenti" e "Riparazione" (N:N)

```

CREATE TABLE Ordine_Riparazione (
    IDOrdine INT,
    IDRiparazione INT,
    PRIMARY key (IDOrdine, IDRiparazione),
    FOREIGN key (IDOrdine) references OrdineComponenti(ID) on delete cascade,
    FOREIGN key (IDRiparazione) references Riparazione(ID) on delete cascade
);

```

-- Impedire che un "Apparecchio" venga consegnato più volte senza essere ritirato

DELIMITER //

CREATE TRIGGER Prima_Di_Inserire_Preventivo

Before insert on Preventivo

For each ROW

BEGIN

 DECLARE ApparecchiInRiparazione INT;

 Select COUNT(*) into ApparecchiInRiparazione

 From Riparazione R

 Join ListaDiAttesa L on R.IDLista = L.ID

 Join Preventivo P on L.IDPreventivo = P.ID

 Where P.IDApparecchio = NEW.IDApparecchio and R.DataFine is null;

 IF ApparecchiInRiparazione > 0 then

 Signal SQLSTATE '45000'

 Set Message_text = 'Impossibile accettare un nuovo preventivo: 1\' apparecchio è ancora in riparazione ';

 End If;

End;

//

DELIMITER ;

-- Impedire il pagamento di una riparazione non completata

DELIMITER //

CREATE TRIGGER Prima_Di_Inserire_Pagamento

Before insert on Pagamento

For each ROW

BEGIN

 DECLARE RiparazioneCompletata DATE;

 Select DataFine into RiparazioneCompletata

 From Riparazione

 Where ID = NEW.IDRiparazione;

 If RiparazioneCompletata is null then

 Signal SQLSTATE '45000'

 Set Message_Text = 'Impossibile registrare il pagamento: la riparazione non è ancora stata completata';

 End if;

End;

//

DELIMITER ;

-- Popolamento tabelle con 10 voci

-- Cliente

INSERT INTO Cliente (CodiceFiscale, Nome, Cognome, TelefonoCellulare, Email, Città, Via, CAP) VALUES

('RSSMRA85M01H501Z', 'Mario', 'Rossi', '3331112222', 'mario.rossi@example.com', 'Milano', 'Via Roma 1', '20100'),

('BNCLRD90D20F205X', 'Luca', 'Bianchi', '3332223333', 'luca.bianchi@example.com', 'Torino', 'Corso Italia 15', '10100'),

('VRDLGI85M01C351Z', 'Luigi', 'Verdi', '3333334444', 'luigi.verdi@example.com', 'Roma', 'Piazza Venezia 10', '00100'),

('BNTGRL92A01H501X', 'Giorgia', 'Bianchi', '3334445555', 'giorgia.bianchi@example.com', 'Napoli', 'Via Toledo 25', '80100'),

('BLUSRA98C01H621Z', 'Sara', 'Blu', '3335556666', 'sara.blu@example.com', 'Firenze', 'Via della Scala 5', '50100'),

('MLNLCU80D12H501Y', 'Luca', 'Milano', '3336667777', 'luca.milano@example.com', 'Genova', 'Via XX Settembre 12', '16100'),

('GNCRNC75F21A851Y', 'Francesco', 'Genchi', '3337778888', 'francesco.genchi@example.com', 'Bari', 'Via Sparano 8', '70100'),

('MRNGLA70L12C129Q', 'Giulia', 'Marino', '3338889999', 'giulia.marino@example.com', 'Palermo', 'Corso Vittorio Emanuele 7', '90100'),

('PLLMRC68E10A001B', 'Marco', 'Pallino', '3339990000', 'marco.pallino@example.com', 'Cagliari', 'Via Roma 30', '09100'),

('TRVRS85H15C234P', 'Sara', 'Trevisan', '3330001111', 'sara.trevisan@example.com', 'Venezia', 'Piazza San Marco 2', '30100');

-- Apparecchio

INSERT INTO Apparecchio (Tipo, Marca, Modello, CodiceFiscaleCliente) VALUES

('Laptop', 'Dell', 'XPS 13', 'RSSMRA85M01H501Z'),

```
('Smartphone', 'Samsung', 'Galaxy S21', 'BNCLRD90D20F205X'),
('Tablet', 'Apple', 'iPad Pro', 'VRDLGI85M01C351Z'),
('PC', 'HP', 'EliteBook', 'BNTGRL92A01H501X'),
('Smartwatch', 'Garmin', 'Fenix 6', 'BLUSRA98C01H621Z'),
('Laptop', 'Asus', 'ZenBook 14', 'MLNLCU80D12H501Y'),
('Smartphone', 'Xiaomi', 'Redmi Note 10', 'GNCRNC75F21A851Y'),
('PC', 'Lenovo', 'ThinkPad T14', 'MRNGLA70L12C129Q'),
('Smartwatch', 'Apple', 'Watch Series 6', 'PLLMRC68E10A001B'),
('Tablet', 'Microsoft', 'Surface Pro', 'TRVRS A85H15C234P');
```

-- Difetto

```
INSERT INTO Difetto (IDApparecchio, DescrizioneDifetto) VALUES
```

```
(1, 'Batteria non si carica'),
(2, 'Schermo rotto'),
(3, 'Tasto home non funziona'),
(4, 'Problemi di surriscaldamento'),
(5, 'Sensore battito cardiaco non rileva dati'),
(6, 'WiFi instabile'),
(7, 'Microfono non funziona'),
(8, 'Problema con il touchpad'),
(9, 'Bluetooth non si connette'),
(10, 'Sistema operativo corrotto');
```

-- Preventivo

```
INSERT INTO Preventivo (IDDifetto, IDApparecchio, DataCreazione, Accettato, DescrizioneDifetto) VALUES
```

```
(1, 1, '2024-01-10', TRUE, 'Batteria non si carica'),
(2, 2, '2024-01-12', TRUE, 'Schermo rotto'),
(3, 3, '2024-01-15', FALSE, 'Tasto home non funziona'),
(4, 4, '2024-01-18', TRUE, 'Surriscaldamento'),
(5, 5, '2024-01-20', TRUE, 'Sensore non rileva dati'),
(6, 6, '2024-01-22', TRUE, 'WiFi instabile'),
(7, 7, '2024-01-25', FALSE, 'Microfono non funziona'),
(8, 8, '2024-01-28', TRUE, 'Touchpad non risponde'),
(9, 9, '2024-02-01', TRUE, 'Bluetooth non si connette'),
(10, 10, '2024-02-03', TRUE, 'Sistema operativo corrotto');
```

-- Lista di attesa

```
INSERT INTO ListaDiAttesa (IDPreventivo, DataInserimento) VALUES
```

```
(1, '2024-01-11'),
```

(2, '2024-01-13'),
(4, '2024-01-19'),
(5, '2024-01-21'),
(6, '2024-01-23'),
(8, '2024-01-29'),
(9, '2024-02-02'),
(10, '2024-02-04');

-- Riparazione

INSERT INTO Riparazione (IDLista, DataInizio, DataFine) VALUES

(1, '2024-01-12', '2024-01-14'),
(2, '2024-01-14', '2024-01-16'),
(3, '2024-01-20', '2024-01-22'),
(4, '2024-01-22', '2024-01-25'),
(5, '2024-01-24', NULL),
(6, '2024-01-30', NULL),
(7, '2024-02-03', NULL),
(8, '2024-02-05', NULL);

-- Componente

INSERT INTO Componente (MarcaComponente, ModelloComponente, CostoAcquisto, QtaDeposito) VALUES

('Samsung', 'Batteria Li-ion', 30.00, 5),
('Apple', 'Schermo OLED', 150.00, 2),
('HP', 'Touchpad Precision', 40.00, 3),
('Xiaomi', 'Microfono', 20.00, 6),
('Garmin', 'Sensore Cardiaco', 50.00, 4),
('Lenovo', 'WiFi Adapter', 35.00, 8),
('Dell', 'Altoparlante', 25.00, 5),
('Microsoft', 'Modulo Bluetooth', 45.00, 7),
('Asus', 'Scheda Madre', 120.00, 1),
('Acer', 'Batteria 4000mAh', 60.00, 9);

-- Fornitore

INSERT INTO Fornitore (ID, Azienda, Sede) VALUES

(1, 'TechParts', 'Milano'),
(2, 'PowerUp', 'Torino'),
(3, 'SoundFix', 'Roma'),
(4, 'NavParts', 'Napoli'),
(5, 'MemStore', 'Firenze'),

```
(6, 'CoolTech', 'Genova'),  
(7, 'AudioPlus', 'Bari'),  
(8, 'ImagePro', 'Palermo'),  
(9, 'ConnectX', 'Cagliari'),  
(10, 'BlueWave', 'Venezia');
```

-- OrdineComponenti

INSERT INTO OrdineComponenti (ID, DataOrdine, DataRicezione) VALUES

```
(1, '2024-01-05', '2024-01-10'),  
(2, '2024-01-06', '2024-01-11'),  
(3, '2024-01-07', '2024-01-12'),  
(4, '2024-01-08', '2024-01-13'),  
(5, '2024-01-09', '2024-01-14'),  
(6, '2024-01-10', '2024-01-15'),  
(7, '2024-01-11', '2024-01-16'),  
(8, '2024-01-12', '2024-01-17'),  
(9, '2024-01-13', '2024-01-18'),  
(10, '2024-01-14', '2024-01-19');
```

-- Pagamento

INSERT INTO Pagamento (IDRiparazione, DataPagamento, Importo, PagamentoElettronico, IncassoTotale) VALUES

```
(1, '2024-01-14', 120.00, TRUE, 120.00),  
(2, '2024-01-16', 250.00, FALSE, 250.00),  
(3, '2024-01-22', 80.00, TRUE, 80.00),  
(4, '2024-01-25', 180.00, FALSE, 180.00);
```

-- Fornisce (Relazione N:N fra Fornitore e OrdineComponenti)

INSERT INTO Fornisce (IDOrdine, IDFornitore) VALUES

```
(1, 1),  
(1, 2),  
(2, 3),  
(3, 1),  
(3, 4),  
(4, 2),  
(5, 3),  
(6, 4),  
(7, 5),  
(8, 1);
```


-- Ordine_Riparazione (Relazione N:N tra OrdineComponenti e Riparazione)

INSERT INTO Ordine_Riparazione (IDOrdine, IDRiparazione) VALUES

(1, 1),

(2, 2),

(3, 3),

(3, 4),

(4, 5),

(5, 6),

(6, 7),

(7, 8),

(8, 2),

(9, 4);

-- Query per interrogare il DataBase

-- Inserimento di un nuovo cliente

INSERT INTO Cliente (CodiceFiscale, Nome, Cognome, TelefonoCellulare, Email, Città, Via, CAP)

VALUES ('4GNEWCUST75L12X1', 'Andrea', 'Ferrari', '3338889999', 'andrea.ferrari@example.com', 'Torino', 'Via Garibaldi 88', '10100');

-- Inserimento di un nuovo apparecchio

INSERT INTO Apparecchio (Tipo, Marca, Modello, CodiceFiscaleCliente)

VALUES ('Laptop', 'HP', 'Pavilion x360', '4GNEWCUST75L12X1');

-- Inserimento di un nuovo preventivo

INSERT INTO Preventivo (IDDifetto, IDApparecchio, DataCreazione, Accettato, DescrizioneDifetto)

VALUES (11, LAST_INSERT_ID(), '2024-02-20', TRUE, 'Problema alla scheda madre');

-- Inserimento di una nuova riparazione passando prima per la lista di attesa

INSERT INTO ListaDiAttesa (IDPreventivo, DataInserimento)

VALUES (LAST_INSERT_ID(), '2024-02-21');

INSERT INTO Riparazione (IDLista, DataInizio, DataFine)

VALUES (LAST_INSERT_ID(), '2024-02-22', NULL);

-- Inserimento di un nuovo componente

INSERT INTO Componente (MarcaComponente, ModelloComponente, CostoAcquisto, QtaDeposito)

VALUES ('Intel', 'Processore i7 11th Gen', 320.00, 5);

-- Effettuare un nuovo OrdineComponenti poi aggiungo il collegamento al fornitore

INSERT INTO OrdineComponenti (DataOrdine, DataRicezione)

VALUES ('2024-02-22', NULL);

INSERT INTO Fornisce (IDOrdine, IDFornitore)

VALUES (LAST_INSERT_ID(), 3);

-- Inserire un nuovo pagamento

INSERT INTO Pagamento (IDRiparazione, DataPagamento, Importo, PagamentoElettronico, IncassoTotale)

VALUES (LAST_INSERT_ID(), '2024-02-25', 400.00, TRUE, 400.00);

-- Calcolo incassi annuali

SELECT DATE_FORMAT(DataPagamento, '%Y-%m') AS Mese, SUM(Importo) AS TotaleIncassato

FROM Pagamento

GROUP BY Mese

ORDER BY Mese DESC;

-- Elencare tutti i clienti con i loro apparecchi registrati

Select C.CodiceFiscale, C.Nome, C.Cognome, A.Tipo, A.Marca, A.Modello

From Cliente C

Join Apparecchio A on C.CodiceFiscale = A.CodiceFiscaleCliente;

-- Trovare il numero di apparecchi registrati per cliente

Select C.Nome, C.Cognome, COUNT(A.ID) as NumeroApparecchi

From Cliente C

Left Join Apparecchio A on C.CodiceFiscale = A.CodiceFiscaleCliente

Group by C.CodiceFiscale;

-- Mostrare i preventivi accettati con i dettagli degli apparecchi e dei clienti

Select P.ID, C.Nome, C.Cognome, A.Tipo, A.Marca, A.Modello, P.DataCreazione, P.Accettato

From Preventivo P

Join Apparecchio A on P.IDApparecchio = A.ID

Join Cliente C on A.CodiceFiscaleCliente = C.CodiceFiscale

Where P.Accettato = TRUE;

-- Trovare gli apparecchi attualmente in riparazione

Select A.Tipo, A.Marca, A.Modello, C.Nome, C.Cognome, R.DataInizio

From Riparazione R

Join ListaDiAttesa L on R.IDLista = L.ID

Join Preventivo P on L.IDPreventivo = P.ID

Join Apparecchio A on P.IDApparecchio = A.ID

Join Cliente C on A.CodiceFiscaleCliente = C.CodiceFiscale

Where R.DataFine is not null;

-- Mostrare le riparazioni completate in un determinato intervallo di tempo

SELECT A.Tipo, A.Marca, A.Modello, C.Nome, C.Cognome, R.DataInizio, R.DataFine

FROM Riparazione R

JOIN ListaDiAttesa L ON R.IDLista = L.ID

JOIN Preventivo P ON L.IDPreventivo = P.ID

JOIN Apparecchio A ON P.IDApparecchio = A.ID

JOIN Cliente C ON A.CodiceFiscaleCliente = C.CodiceFiscale

WHERE R.DataFine BETWEEN '2024-01-01' AND '2024-01-31'

ORDER BY R.DataFine ASC;

-- Mostrare gli ordini di componenti non ancora ricevuti

Select OC.ID, C.MarcaComponente, C.ModelloComponente, OC.DataOrdine

From OrdineComponenti OC

Join Fornisce F on OC.ID = F.IDOrdine

Join Componente C on F.IDOrdine = C.ID

Where OC.DataRicezione is not null;

-- Elenco dei componenti disponibili in magazzino con quantità inferiore a 5

Select ID, MarcaComponente, ModelloComponente, QtaDeposito

From Componente

Where QtaDeposito < 5;

-- Calcolo incasso totale mensile

Select DATE_FORMAT(DataPagamento, '%Y-%m') as Mese, SUM(Importo) as TotaleIncassato

From Pagamento

Group by Mese

Order by Mese Desc;

-- Mostrare i pagamenti effettuati in un determinato intervallo di tempo

Select P.ID, R.ID as IDRiparazione, C.Nome, C.Cognome, P.Importo, P.DataPagamento

From Pagamento P

Join Riparazione R on P.IDRiparazione = R.ID

Join ListaDiAttesa L on R.IDLista = L.ID

Join Preventivo PR on L.IDPreventivo = PR.ID

Join Apparecchio A on PR.IDApparecchio = A.ID

Join Cliente C on A.CodiceFiscaleCliente = C.CodiceFiscale

Where P.DataPagamento between '2024-01-01' and '2024-12-31';

-- Visualizzare la lista di attesa in ordine di arrivo

Select L.ID, P.ID as IDPreventivo, C.Nome, C.Cognome, A.Tipo, A.Marca, A.Modello, L.DataInserimento

From ListaDiAttesa L

Join Preventivo P on L.IDPreventivo = P.ID

Join Apparecchio A on P.IDApparecchio = A.ID

Join Cliente C on A.CodiceFiscaleCliente = C.CodiceFiscale

Order by L.DataInserimento ASC;

-- Mostrare gli apparecchi che hanno avuto più di una riparazione

Select A.Tipo, A.Marca, A.Modello, C.Nome, C.Cognome, COUNT(R.ID) as NumeroRiparazioni

From Riparazione R

Join ListaDiAttesa L on R.IDLista = L.ID

Join Preventivo P on L.IDPreventivo = P.ID

Join Apparecchio A on P.IDApparecchio = A.ID

Join Cliente C on A.CodiceFiscaleCliente = C.CodiceFiscale

Group by A.ID

Having COUNT(R.ID) > 1;

-- Test per "Prima_Di_Inserire_Preventivo"

INSERT INTO Preventivo (IDDifetto, IDApparecchio, DataCreazione, Accettato, DescrizioneDifetto)

VALUES (6, 6, '2024-01-22', TRUE, 'WiFi instabile di nuovo');

-- Test per "Prima_Di_Inserire_Pagamento"

INSERT INTO Pagamento (IDRiparazione, DataPagamento, Importo, PagamentoElettronico, IncassoTotale)

VALUES (5, '2024-02-12', 150.00, TRUE, 150.00);