

**UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II**



**FACOLTÀ DI INGEGNERIA INFORMATICA**

**S.A.S.: SALUTEme A Soreta**

Corso di Basi di Dati (9CFU)

A.A. 2025/26

Prof. Vincenzo Moscato

Christian Russo, N46007636

Claudia Spina, N46007588

Luca Maria Incarnato, N46006935



## INDICE DEGLI ARGOMENTI

1. IL CONTESTO E L'APPROCCIO (p. 3)
  - 1.1. ANALISI CONTESTUALE (p. 3)
  - 1.2. GLI OBIETTIVI E L'APPROCCIO METODOLOGICO (p. 3)
2. ANALISI ED INTERPRETAZIONE DEI DATI (p. 4)
  - 2.1. IL PAZIENTE (p. 4)
  - 2.2. LA STORIA CLINICA (p. 4)
  - 2.3. COMPLEMENTI TESTUALI (p. 7)
  - 2.4. OSSERVAZIONI RILEVANTI (p. 7)
3. PROGETTAZIONE CONCETTUALE (p. 8)
  - 3.1. MODELLO E-R PORTANTE (p. 8)
  - 3.2. MODELLO E-R ESTESO (p. 8)
  - 3.3. PANORAMICA GLOBALE (p. 9)
    - 3.3.1. ENTITÀ (p. 9)
    - 3.3.2. ASSOCIAZIONI (p. 10)
4. PROGETTAZIONE LOGICA (p. 12)
  - 4.1. IL PAZIENTE (p. 12)
  - 4.2. LA STORIA CLINICA (p. 12)
  - 4.3. COMPLEMENTI TESTUALI (p. 13)
5. RIORGANIZZAZIONE E FILTRAGGIO DEL DATASET (p. 14)
6. PROGETTAZIONE FISICA (p. 17)
  - 6.1. DIMENSIONAMENTO FISICO (p. 17)
    - 6.1.1. IL PAZIENTE (p. 17)
    - 6.1.2. LA STORIA CLINICA (p. 18)
    - 6.1.3. COMPLEMENTI TESTUALI (p. 27)
  - 6.2. DATA DEFINITION LANGUAGE (p. 28)
7. QUERY, VISTE E TRIGGER (p. 36)
  - 7.1. QUERY (p. 36)
  - 7.2. VISTE (p. 37)
  - 7.3. TRIGGER (p. 38)

## 1 IL CONTESTO E L'APPROCCIO

### 1.1 ANALISI CONTESTUALE

#### “Do Your Part, Care for Your Heart” (American Health Month)

Dai primi anni Venti del secolo scorso, dai primi passi nella comprensione delle malattie cardiovascolari, ad oggi, il progresso tecnologico e la ricerca scientifica hanno contribuito al salvataggio di milioni di vite, di milioni di cuori; tuttavia, ciò non esclude la possibilità di un ulteriore passo in avanti. In un contesto sociale in cui l'obesità, la sedentarietà, il fumo e l'ipertensione sono diventati termini troppo comuni nel vocabolario colloquiale, anche le piccole realtà possono contribuire a quel numero di cuori salvati; per questo motivo, una soluzione come **S.A.S.: SALUTEme A Soreta** (d'ora in poi, solo S.A.S.), seppur non vada a contribuire in maniera diretta alla causa, può comunque essere uno strumento ausiliario sul quale i medici e i chirurghi del settore possono virare per evitare che risorse potenzialmente utilizzabili per task delicate o più importanti vengano utilizzate per cercare un dato di un paziente in un bacino di dati male organizzati, ridondanti o incongrui.

### 1.2 GLI OBIETTIVI E L'APPROCCIO METODOLOGICO

S.A.S. è un sistema che permette di memorizzare, monitorare e analizzare l'intero percorso clinico di un paziente cardiologico (a partire dalle informazioni anamnestiche agli esami diagnostici ed interventistici, con l'inclusione di visite di follow-up e di eventi clinici registrati nel tempo), configurandosi, in questo modo, come un punto di raccolta unico a supporto di medici e chirurghi nel processo di analisi e intervento.

I principali obiettivi, per la realizzazione di S.A.S., sono:

- **Integrazione e Aggregazione dei dati**, strutturando lo schema di base di dati in modo che possa includere tutte le informazioni a disposizione dei medici per ogni singolo paziente;
- **Supporto decisionale**, attraverso l'implementazione di query volte alla comprensione dell'intero contesto clinico del paziente;
- **Coerenza dei dati**, realizzando una base di dati attiva in grado di reagire in maniera automatica a nuovi eventi.

Il ciclo di vita del progetto si compone di diverse fasi, seguendo la metodologia standard per la progettazione di un'applicazione di basi di dati:

1. **Interpretazione analitica dei dati**, finalizzata alla comprensione dello schema di basi di dati a partire dal quale effettuare il lavoro di riprogettazione;
2. **Progettazione Concettuale**, volta alla produzione di modello Entity Relationship adeguato alla descrizione della realtà sotto esame ed efficace in relazione agli obiettivi posti;
3. **Progettazione Logica**, per la traduzione del modello concettuale individuato in uno schema di base di dati agevole all'implementazione;
4. **Riorganizzazione dei dati e Filtraggio**, in modo da eliminare eventuali ridondanze e normalizzare il dataset a disposizione al nuovo schema di base di dati;
5. **Implementazione Fisica**, volta alla produzione di script DDL e DML in ambiente Oracle Database per l'attuale realizzazione della base di dati;

## 2 ANALISI ED INTERPRETAZIONE DEI DATI

Il processo di analisi del patrimonio di dati preesistente, composto da una raccolta di file .csv volti alla collezione di una quantità eterogenea e variegata di dati circa la storia clinica di ogni singolo paziente del reparto di Cardiologia, è finalizzato alla **comprensione dello status quo**, al fine di poter individuare eventuali punti critici nella struttura del sistema o eventuali ridondanze che rendono il lavoro difficile o rallentano l'accesso alle informazioni. Sulla base dell'osservazione della struttura e della documentazione preesistente, si è potuta operare la seguente suddivisione in aree semantiche.

Nota: laddove si è riscontrata una elevata quantità di informazioni, la cui trascrizione in questo documento esula dallo scopo della trattazione, sono stati inseriti punti di omissione atti ad indicare l'esistenza di altre informazioni non ancora rilevati a questo punto del ciclo di sviluppo.

### 2.1 IL PAZIENTE

Entità **ANAGRAFICA**:

- Rappresenta il paziente in cura presso l'azienda ospedaliera e al quale faranno riferimento tutti gli eventi clinici ad esso associati;
- Identificatori: SEZIONE, CODPAZ;
- Altri attributi:
  - Demografia: DATADINASCITA, SESSO, COMUNE\_DI\_NASCITA;
  - Contesto clinico: GATE\_DI\_INGRESSO, MOTIVO\_DECESSO, DATA\_DECESSO

Entità **PREVALENT**:

- Highlight clinico del paziente, informazioni sulle condizioni prevalenti al fine di fornire un quadro delle comorbidità e dello stato di salute del paziente, utile per la gestione del paziente a lungo termine;
- Identificatori: SEZIONE, CODPAZ;
- Altri attributi:
  - Contesto clinico: GATE\_DI\_INGRESSO, CAUSA\_RICOVERO\_EXTRA;
  - Dati antropometrici: PESO, ALTEZZA;
  - Semantica relativa: DIABETE, DISLIPIDEMIA, BMI, OBESITÀ, PREVIOUS\_IMA, CAD, ATRIAL\_FIBRILLATION, ...;
  - Terapie farmacologiche: DIURETICI, ALFABLOCCANTI, BETABLOCCANTI, ACEINIBITORI, ANTIARITMICI, ANTIAGGREGANTI, ...;
  - FUMO, NYHA, STADIO\_SCOMPENSO.

### 2.2 LA STORIA CLINICA

È stato rilevato un meccanismo di Ereditarietà tra le entità che descrivono le informazioni circa gli eventi clinici di ogni singolo paziente.

Entità Padre **LISTA\_EVENTI**:

- Raccorta cronologica di tutti gli eventi clinici di ogni singolo paziente;
- Identificatori: SEZIONE, CODPAZ, NUM\_PROGRESSIVO\_GLOBALE;
- Attributi comuni alle entità figlie: NUM\_PROGRESSIVO\_GLOBALE;
- Altri attributi:

- TIPO\_EVENTO (descrive l'entità figlia che specializza l'evento clinico del record).

#### Entità Figlia **ANAMNESI**:

- Strumento di ricostruzione della storia clinica dei pazienti;
- Identificatori: SEZIONE, CODPAZ, NUM\_PROGRESSIVO;
- Altri attributi:
  - Semantica relativa: DIABETE, DISLIPIDEMIA, STROKE, CAD, ...;
  - FUMO;
  - Contesto clinico: GATE\_DI\_INGRESSO;
  - NUM\_PROGRESSIVO\_GLOBALE.

#### Entità Figlia **ESAMI\_SPECIALISTICI**:

- Focus sugli esami specialistici, in particolare in relazione alla funzione tiroidea;
- Identificatori: SEZIONE, CODPAZ, NUM\_PROGRESSIVO;
- Altri attributi:
  - DATE;
  - Semantica relativa: FT3, FT4, TSH;
  - Contesto clinico: GATE\_DI\_INGRESSO;
  - NUM\_PROGRESSIVO\_GLOBALE.

#### Entità Figlia **ESAMI\_LABORATORIO**:

- Offre un quadro completo dello stato metabolico del paziente riassumendo una vasta gamma di risultati di laboratorio, finalizzato al monitoraggio delle condizioni metaboliche, della funzione renale ed epatica e dell'infiammazione;
- Identificatori: SEZIONE, CODPAZ, NUM\_PROGRESSIVO;
- Altri attributi:
  - DATE;
  - Semantica relativa: GLICEMIA, CREATININEMIA, EMOGLOBINA, URICEMIA, SODIEMIA, MALBUMINURIA, OMOCISTEINEMIA, DIABETE, FILTRATO\_GFR, CKD, CKD\_STADIO, ...;
  - Contesto clinico: GATE\_DI\_INGRESSO;
  - NUM\_PROGRESSIVO\_GLOBALE.

#### Entità Figlia **ESAMI\_STRUMENTALI\_CARDIO**:

- Raccolta e sintesi di esami strumentali cardiaci;
- Identificatori: SEZIONE, CODPAZ, NUM\_PROGRESSIVO;
- Altri attributi:
  - DATE;
  - Semantica relativa: CONCLUSIONI;
  - Contesto clinico: GATE\_DI\_INGRESSO;
  - NUM\_PROGRESSIVO\_GLOBALE.

#### Entità Figlia **CORONAROGRAFIA\_PTCA**:

- Riporta i risultati delle coronarografie e delle Angioplastiche Coronariche Transluminali Percutanee;
- Identificatori: SEZIONE, CODPAZ, NUM\_PROGRESSIVO;
- Altri attributi:
  - DATA;
  - Semantica relativa: LESIONI\_TC, LESIONI\_IVA, LESIONI\_CX, LESIONI\_DX;
  - Contesto clinico: GATE\_DI\_INGRESSO;
  - NUM\_PROGRESSIVO\_GLOBALE.

#### Entità Figlia **ECOCAROTIDI**:

- Riporta dati derivanti dagli ecocardiogrammi, dettagliando la funzione cardiaca e la morfologia, consentendo di identificare la presenza e la gravità dell'aterosclerosi carotidea (fattore di rischio per ictus e altre patologie cardiovascolari);
- Identificatori: SEZIONE, CODPAZ, NUM\_PROGRESSIVO;
- Altri attributi:
  - DATE;
  - Semantica relativa: IMTMAX, C\_COMUNE\_DIAMETRO, ATEROSCLEROSI\_SEVERA, ATEROSCLEROSI\_MODERATA, ATEROSCLEROSI\_LIEVE, ...;
  - Contesto clinico: GATE\_DI\_INGRESSO;
  - NUM\_PROGRESSIVO\_GLOBALE.

#### Entità Figlia **RICOVERO\_OSPEDALIERO**:

- Rileva le informazioni circa i ricoveri ospedalieri dei pazienti;
- Identificatori: SEZIONE, CODPAZ, NUM\_PROGRESSIVO;
- Altri attributi:
  - DATE;
  - Semantica relativa: SINTOMI, SINTOMI\_PS, RICHIESTA\_RICOVERO;
  - Contesto clinico: GATE\_DI\_INGRESSO;
  - NUM\_PROGRESSIVO\_GLOBALE.

#### Entità Figlia **ECOCARDIO\_DATI**:

- Riporta i dati provenienti da ecocardiogrammi per dettagliare la funzione cardiaca e la morfologia;
- Identificatori: SEZIONE, CODPAZ, NUM\_PROGRESSIVO;
- Altri attributi:
  - DATE;
  - Dati antropometrici: PESO, ALTEZZA, PAS, PAD;
  - Semantica relativa: AORTA\_ASCENDENTE, AORTA\_ADDOMINALE, BICUSPIDE, STENOSI\_AORTICA, TROMBO, HEART\_FAILURE, ...;
  - Contesto clinico: GATE\_DI\_INGRESSO;
  - NUM\_PROGRESSIVO\_GLOBALE.

#### Entità Figlia **VISITA\_CONTROLLO\_ECG**:

- Riporta informazioni sulle condizioni prevalenti dei pazienti, in modo da rilevare eventuali comorbidità precedentemente sfuggite e fornendo una visione d'insieme delle condizioni croniche, dei trattamenti e dei fattori di rischio al fine della gestione a lungo termine del paziente;
- Identificatori: SEZIONE, CODPAZ, NUM\_PROGRESSIVO;
- Altri attributi:
  - Contesto clinico: GATE\_DI\_INGRESSO, CAUSA\_RICOVERO\_EXTRA;
  - Dati antropometrici: PESO, ALTEZZA;
  - Semantica relativa: DIABETE, DISLIPIDEMIA, BMI, OBESITÀ, PREVIOUS\_IMA, CAD, ATRIAL\_FIBRILLATION, ...;
  - Terapie farmacologiche: DIURETICI, ALFABLOCCANTI, BETABLOCCANTI, ACEINIBITORI, ANTIARITMICI, ANTIAGGREGANTI, ...;
  - FUMO, NYHA, STADIO\_SCOMPENSO.
  - NUM\_PROGRESSIVO\_GLOBALE.

## 2.3 COMPLEMENTI TESTUALI

Entità **TEXT**:

- Bacino di dati eterogenei dedicato all'archiviazione e all'analisi di dati testuali non strutturati, derivanti da note cliniche e documentazione medica (spesso ricca di valori non interpretabili o non validi, indicando l'assenza di note testuali dettagliate);
- Identificatori: SEZIONE, CODPAZ, NUM\_PROGRESSIVO;
- Altri attributi:
  - DATA, TIPO\_EVENTO (descrive l'entità figlia che specializza l'evento clinico del record);
  - TEXT (Contiene il testo descrittivo libero inserito dai clinici durante le visite);
  - Patologie e fattori di rischio: DIABETE, DISLIPIDEMIA, FUMO, IPERTENSIONE\_ARTERIOSA, OBESITÀ, CAD, BPCO, STROKE, AORTIC\_DISEASE, ...;
  - Contesto clinico: GATE\_DI\_INGRESSO;
  - NUM\_PROGRESSIVO\_GLOBALE.

## 2.4 OSSERVAZIONI RILEVANTI

L'operazione di interpretazione del dataset ha fatto emergere importanti considerazioni e vincoli che intrinseci alla realtà di interesse:

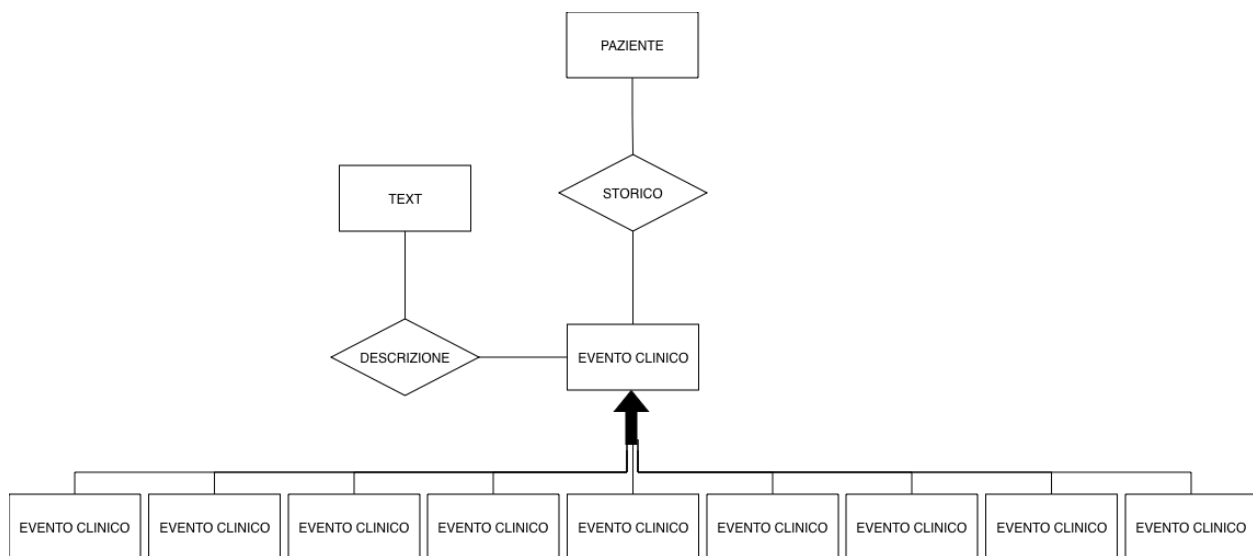
- **Single Source of Truth**, ANAGRAFICA e PREVALENT, in associazione 1:1 tra di loro, raccolgono l'insieme di tutti i pazienti in cura presso l'azienda ospedaliera e li identifica tramite l'attributo CODPAZ, il che significa che non esisterà alcuna informazione contestuale associata ad un paziente che non abbia un relativo record nelle relazioni sopra specificate;
- **Generalizzazione Totale Disgiunta**, LISTA\_EVENTI generalizza gli eventi clinici associati ad un paziente in modo che ogni record individui sempre uno e un solo evento specializzato (un ESAME\_LABORATORIO non può essere anche ESAME\_SPECIALISTICO);
- **Complementarità dei Report Testuali**, TEXT esiste solo in associazione ad un evento clinico e, pertanto, non potrà esistere se non in associazione con un record di LISTA\_EVENTI, seppur un evento clinico possa esistere senza un report testuale.



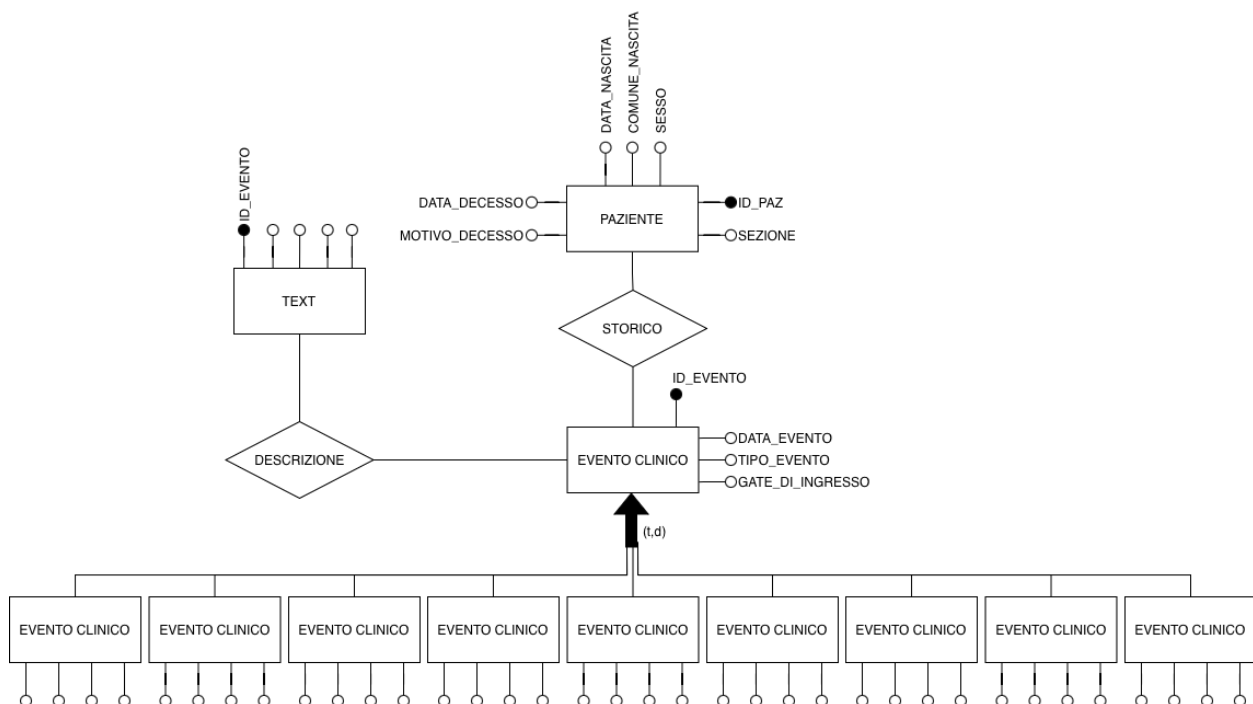
### 3 PROGETTAZIONE CONCETTUALE

La prima vera **Milestone** durante lo sviluppo di S.A.S. è stata la realizzazione dello **schema E-R** descrittivo della realtà di interesse; l'importanza della sua formulazione risiede nella possibilità di tradurre i requisiti e le analisi finora svolte in un modello che, seppur sia ancora di alto livello, permette di descrivere esaustivamente e a pieno la realtà che si intende modellare, prescindendo da alcuna soluzione fisica o relazionale, in modo da potersi focalizzare sulla specifica di tutte le dinamiche senza distrazioni. La progettazione concettuale tramite il modello teorico Entity-Relationship, così pensata, permette la creazione di un'ossatura solida ma allo stesso tempo modulabile, in modo da poter adattare la sua traduzione ad una qualsiasi infrastruttura relazionale e fisica.

#### 3.1 MODELLO E-R PORTANTE



#### 3.2 MODELLO E-R ESTESO



### 3.3 PANORAMICA GLOBALE

#### 3.3.1 ENTITÀ

Di seguito saranno riportate tutte le Entità coinvolte nella descrizione del sistema, con l'elenco dettagliato di tutti gli attributi che, per motivi di sintesi, non sono stati riportati nelle rappresentazioni grafiche del modello E-R.

Entità Forte: **PAZIENTE**

ID\_PAZ, SEZIONE, DATA\_DI\_NASCITA, SESSO, COMUNE\_DI\_NASCITA,  
MOTIVO\_DECESSO, DATA\_DECESSO

Entità: TEXT

ID\_TEXT, TESTO, MATCHED\_CONCEPTS, MATCHED\_CONCEPTS.1,  
MATCHED\_CONCEPT\_TYPES, DIABETE, DISPLIDEMIA,  
DISPLIDEMIA\_IPERCOLESTEROLEMIA, DISPLIDEMIA\_IPERTRIGLICERIDEMIA,  
FUMO, IPERTENSIONE\_ARTERIOSA, OBESITÀ, IRC\_CKD, IRC\_STADIO5, CAD,  
CAD\_FAMILIARITY, BPCO, ATRIAL\_FIBRILLATION, PREVIOUS\_IMA,  
PREVIOUS\_PCI, PREVIOUS\_CABG, STROKE, REVASCULARIZATION\_PERIPHERAL,  
AORTIC\_DISEASE, HEART\_FAILURE, PERIPHERAL\_ARTERY\_DISEASE, SVA\_SVM,  
ICD, PM, CRT-D

Entità Superclasse: **EVENTO CLINICO**

ID\_EVENTO, TIPO\_EVENTO, DATA\_EVENTO, GATE\_INGRESSO

Entità Sottoclasse: **ECOCARDIO**

AORTA\_ASCENDENTE, AORTA\_ADDOMINALE, PESO, ALTEZZA, PAS, PAD,  
BICUSPIDE, RR, AORTA, VSTD, VSTS,  
SIVTD, PPTD, PFVEPFVA, EA, DT, LVVS, LVVD, LVVD\_2CH, LVVS\_2CH, E\_TDI\_SETTAL  
E, E\_TDI, LAV, \_2CH\_LAV, JET\_RIG\_TRIC, TAPSE, ATRIO,  
INSUFFICIENZA\_AORTICA, INSUFFICIENZA\_MITRALICA,  
INSUFFICIENZA\_TRICUSPIDALICA, STENOSI\_AORTICA, STENOSI\_MITRALICA,  
TROMBO, HEART\_FAILURE, AORTIC\_DISEASE\_ASCENDING\_AORTIC\_DILATION, AORT  
IC\_DISEASE\_ASCENDING\_AORTIC\_ANEURYSM,  
AORTIC\_DISEASE ABDOMINAL\_AORTIC\_ANEURYSM

Entità Sottoclasse: **ECOCAROTIDI**

C\_COMUNE\_DIAMETRO, IMT\_MAX, MAXIMTNEARCIDX, MAXIMTNEARCISX,  
MAXIMTFARCIDX, MAXIMTFARCISX, MAXIMTNEARBDX, MAXIMTNEARBSX,  
MAXIMTFARBDX, MAXIMTFARBSX, MAXIMTNEARCCDX, MAXIMTNEARCCSX,  
MAXIMTFARCCDX, MAXIMTFARCCSX, ESAME\_NELLA\_NORMA,  
ATEROSCLEROSI\_LIEVE, ATEROSCLEROSI\_MODERATA, ATEROSCLEROSI\_SEVERA

Entità Sottoclasse: **PROCEDURA INTERVENTISTICA**

LESIONI\_TC, LESIONI\_IVA, LESIONI\_CX, LESIONI\_DX

Entità Sottoclasse: **VISITA CONTROLLO**

DIABETE, DIABETE\_INSULINE\_TREATED, DIABETE\_ORAL\_THERAPY,  
 DISPLIDEMIA\_STATINE, DISPLIDEMIA\_FIBRATI, BMI, OBESITÀ,  
 PREVIOUS\_IMA, CAD, FIBRILLAZIONE\_ATRIALE, PESO, ALTEZZA,  
 PRESSIONE\_ARTERIOSA\_SISTOLICA\_DOMICILIARE,  
 PRESSIONE\_ARTERIOSA\_DIASTOLICA\_DOMICILIARE,  
 PRESSIONE\_ARTERIOSA\_CLINOSTATICA\_MIN,  
 PRESSIONE\_ARTERIOSA\_CLINOSTATICA\_MAX, FREQ\_CLINO,  
 PRESSIONE\_ARTERIOSA\_ORTOSTATICA\_MIN,  
 PRESSIONE\_ARTERIOSA\_ORTOSTATICA\_MAX, FREQ\_QORTO,  
 PRESSIONE\_ARTERIOSA\_SEDUTO\_MIN, PRESSIONE\_ARTERIOSA\_SEDUTO\_MAX,  
 FREQ\_SIT, FUMO, DIURETICI, ALFABLOCCANTI, BETABLOCCANTI,  
 ACEINIBITORI, AT1\_ANTAGONISTI, DIIDROPIRI-DINICI, ANTIARITMICI,  
 ANTIAGGREGANTI, ANTICOAGULANTI, SEGUETERAPIA,  
 TIPO\_DI\_RICOVERO\_CARDIO, TIPO\_DI\_RICOVERO\_EXTRA,  
 CAUSA\_RICOVERO\_EXTRA, NYHA, STADIO\_SCOMPENSO

#### Entità Sottoclasse: **ANAMNESI**

DIABETE, DISPLIDEMIA, DISPLIDEMIA\_IPERCOLESTEROLEMIA,  
 DISPLIDEMIA\_IPERTRIGLICERIDEMIA, DISPLIDEMIA\_MISTA, FUMO,  
 PREVIOUS\_IMA, PREVIOUS\_PCI, PREVIOUS\_CABG, CAD, STROKE,  
 CAD\_FAMILIARITY\_CEREBRAL, CAD\_FAMILIARITY\_CARDIAC

#### Entità Sottoclasse: **RICOVERO OSPEDALIERO**

RICHIESTA\_RICOVERO, SINTOMI, SINTOMI\_PS

#### Entità Sottoclasse: **ESAME STRUMENTALE**

CONCLUSIONI

#### Entità Sottoclasse: **ESAME SPECIALISTICO**

FT3, FT4, TSH

#### Entità Sottoclasse: **ESAME LABORATORIO**

GLICEMIA, CREATININEMIA, EMOGLOBINA, URICEMIA, SODIEMIA,  
 POTASSIEMIA, CALCEMIA, TRIGLICERIDI, COLESTEROLO, HDL, PCR,  
 HBA1C\_PERC, MALBUMINURIA, PROTEINURIA24, OMOCISTEINEMI,  
 PROTEINE\_TOTALI, ALBUMINE, GOT, GPT, DIABETE, FILTRAT\_GFR, CKD,  
 CKD\_STADIO

### 3.3.2 ASSOCIAZIONI

#### Associazione: **STORICO**

- Entità **PAZIENTE**:
  - Un paziente può essere associato ad una molteplicità di esami ed eventi clinici ma, nel momento in cui è inserito o in cui non ha ancora effettuato alcun test o esame, può anche non possederne;
  - Cardinalità: (0,N) ;

- Entità **EVENTO CLINICO**:
  - Uno stesso evento clinico non potrà mai essere fatto su due pazienti diversi e, pertanto, sarà associato ad uno ed un unico paziente;
  - Cardinalità: (1, 1);
- Tipo: (1, N).

Associazione: **DESCRIZIONE**

- Entità **TEXT**:
  - Uno stesso testo descrittivo complementare può riferirsi ad uno ed un solo evento clinico, nel caso in cui sia necessario specificare informazioni aggiuntive su un altro esame sarà necessaria un'ulteriore entry;
  - Cardinalità: (1, 1);
- Entità **EVENTO CLINICO**
  - Un evento clinico può essere descritto così come può non necessitare di un complemento descrittivo ma, se descritto, sarà associato ad uno ed un solo testo descrittivo complementare;
  - Cardinalità: (0, 1);
- Tipo: (1, 1).

Gerarchia di Specializzazione

- **Superclasse**: EVENTO CLINICO;
  - Specializzazione di tipo totale;
- **Sottoclassi**: ECOCARDIO, ECOCAROTIDI, PROCEDURA INTERVENTISTICA, VISITE CONTROLLO, ANAMNESI, RICOVERO OSPEDALIERO, ESAME STRUMENTALE, ESAME SPECIALISTICO, ESAME LABORATORIO;
  - Specializzazione di tipo disgiunta;
- **Tipo di specializzazione**: Totale-Disgiunta.

## 4 PROGETTAZIONE LOGICA

Di seguito è riportato il risultato del processo di traduzione del modello E-R in un modello logico-relazionale più facile da implementare in un modello fisico da importare in una base di dati. La necessità di questa Milestone risiede nell'avvicinamento del progetto al ciclo relativo alla progettazione fisica della base di dati, richiedendo un livello di astrazione più basso e una specificità più elevata.

Si osservi, in particolare, la traduzione delle due associazioni del modello E-R:

- **STORICO**, viene accorpata in EVENTO CLINICO tramite un vincolo di integrità referenziale con PAZIENTE;
- **DESCRIZIONE**, viene accorpata in EVENTO CLINICO tramite un vincolo di integrità referenziale con TEXT.

### 4.1 IL PAZIENTE

**PAZIENTE** (ID\_PAZ, SEZIONE, DATA\_DI\_NASCITA, SESSO, COMUNE\_DI\_NASCITA, MOTIVO\_DECESSO, DATA\_DECESSO)

### 4.2 LA STORIA CLINICA

**ECOCARDIO** (ID\_ECOCARDIO, DATA\_EVENTO, ID\_PAZ:PAZIENTI, GATE\_DI\_INGRESSO, ID\_TEXT:TEXT, EF, AORTA\_ASCENDENTE, AORTA\_ADDOMINALE, PESO, ALTEZZA, PAS, PAD, BICUSPIDE, RR, AORTA\_VSTD, VSTS, SIVTD, PPTD, PFVEPFVA, EA, DT, LVVS, LVVD, LVVD\_2CH, LVVS\_2CH, E\_TDI\_SETTAL E, E\_TDI, LAV, \_2CH\_LAV, JET\_RIG\_TRIC, TAPSE, ATRIO, INSUFFICIENZA\_AORTICA, INSUFFICIENZA\_MITRALICA, INSUFFICIENZA\_TRICUSPIDALICA, STENOSI\_AORTICA, STENOSI\_MITRALICA, TROMBO, HEART\_FAILURE, AORTIC\_DISEASE\_ASCENDING\_AORTIC\_DILATION, AORTIC\_DISEASE\_ASCENDING\_AORTIC\_ANEURYSM, AORTIC\_DISEASE ABDOMINAL\_AORTIC\_ANEURYSM)

**ECOCAROTIDI** (ID\_ECOCAROTIDI, DATA\_EVENTO, ID\_PAZ:PAZIENTI, GATE\_DI\_INGRESSO, ID\_TEXT:TEXT, C\_COMUNE\_DIAMETRO, IMT\_MAX, MAXIMTNEARCIDX, MAXIMTNEARCISX, MAXIMTFARCIDX, MAXIMTFARCISX, MAXIMTNEARBDX, MAXIMTNEARBSX, MAXIMTFARBDX, MAXIMTFARBSX, MAXIMTNEARCCDX, MAXIMTNEARCCSX, MAXIMTFARCCDX, MAXIMTFARCCSX, ESAME\_NELLA\_NORMA, ATROSCLEROSI\_LIEVE, ATROSCLEROSI\_MODERATA, ATROSCLEROSI\_SEVERA)

**PROCEDURA\_INTERVENTISTICA** (ID\_CORONOGRRAFIA\_PTCA, DATA\_EVENTO, ID\_PAZ:PAZIENTI, GATE\_DI\_INGRESSO, ID\_TEXT:TEXT, LESIONI\_TC, LESIONI\_IVA, LESIONI\_CX, LESIONI\_DX)

**VISITA\_CONTROLLO\_ECG** (ID\_VISITA, DATA\_EVENTO, ID\_PAZ:PAZIENTI, GATE\_DI\_INGRESSO, ID\_TEXT:TEXT, DIABETE, DIABETE\_INSULINE\_TREATED, DIABETE\_ORAL\_THERAPY, DISPLIDEMIA\_STATINE, DISPLIDEMIA\_FIBRATI, BMI, OBESITÀ, PREVIOUS\_IMA, CAD, FIBRILLAZIONE\_ATRIALE, PESO, ALTEZZA, PRESSIONE\_ARTERIOSA\_SISTOLICA\_DOMICILIARE, PRESSIONE\_ARTERIOSA\_DIASTOLICA\_DOMICILIARE,

PRESSIONE\_ARTERIOSA\_CLINOSTATICA\_MIN,  
 PRESSIONE\_ARTERIOSA\_CLINOSTATICA\_MAX, FREQ\_CLINO,  
 PRESSIONE\_ARTERIOSA\_ORTOSTATICA\_MIN,  
 PRESSIONE\_ARTERIOSA\_ORTOSTATICA\_MAX, FREQ\_QORTO,  
 PRESSIONE\_ARTERIOSA\_SEDUTO\_MIN, PRESSIONE\_ARTERIOSA\_SEDUTO\_MAX,  
 FREQ\_SIT, FUMO, DIURETICI, ALFABLOCCANTI, BETABLOCCANTI,  
 ACEINIBITORI, AT1\_ANTAGONISTI, DIIDROPIRI-DINICI, ANTIARITMICI,  
 ANTIAGGREGANTI, ANTICOAGULANTI, SEGUETERAPIA,  
 TIPO\_DI\_RICOVERO\_CARDIO, TIPO\_DI\_RICOVERO\_EXTRA,  
 CAUSA\_RICOVERO\_EXTRA, NYHA, STADIO\_SCOMPENSO)

**ANAMNESI** (ID\_ANAMNESI, DATA\_EVENTO, ID\_PAZ:PAZIENTI,  
 GATE\_DI\_INGRESSO, ID\_TEXT:TEXT, DIABETE, DISPLIDEMIA,  
 DISPLIDEMIA\_IPERCOLESTEROLEMIA, DISPLIDEMIA\_IPERTRIGLICERIDEMIA,  
 DISPLIDEMIA\_MISTA, FUMO, PREVIOUS\_IMA, PREVIOUS\_PCI, PREVIOUS\_CABG,  
 CAD, STROKE, CAD\_FAMILIARITY\_CEREBRAL, CAD\_FAMILIARITY\_CARDIAC)

**RICOVERO OSPEDALIERO** (ID\_RICOVERO OSPEDALIERO, DATA\_EVENTO,  
 ID\_PAZ:PAZIENTI, GATE\_DI\_INGRESSO, ID\_TEXT:TEXT,  
 RICHIESTA\_RICOVERO, SINTOMI, SINTOMI\_PS)

**ESAME STRUMENTALE** (ID\_STRUMENTALE, DATA\_EVENTO, ID\_PAZ:PAZIENTI,  
 GATE\_DI\_INGRESSO, ID\_TEXT:TEXT, CONCLUSIONI)

**ESAME SPECIALISTICO** (ID\_ESAME\_SPECIALISTICO, DATA\_EVENTO,  
 ID\_PAZ:PAZIENTI, GATE\_DI\_INGRESSO, ID\_TEXT:TEXT, FT3, FT4, TSH)

**ESAME LABORATORIO** (ID\_LAB, DATA\_EVENTO, ID\_PAZ:PAZIENTI,  
 GATE\_DI\_INGRESSO, ID\_TEXT:TEXT, GLICEMIA, CREATININEMIA,  
 EMOGLOBINA, URICEMIA, SODIEMIA, POTASSIEMIA, CALCEMIA, TRIGLICERIDI,  
 COLESTEROLO, HDL, PCR, HBA1C\_PERC, MALBUMINURIA, PROTEINURIA24,  
 OMOCISTEINEMI, PROTEINE\_TOTALI, ALBUMINE, GOT, GPT, DIABETE,  
 FILTRAT\_GFR, CKD, CKD\_STADIO)

### 4.3 COMPLEMENTI TESTUALI

**TEXT** (ID\_TEXT, TESTO, MATCHED\_CONCEPTS, MATCHED\_CONCEPTS.1,  
 MATCHED\_CONCEPT\_TYPES, DIABETE, DISPLIDEMIA,  
 DISPLIDEMIA\_IPERCOLESTEROLEMIA, DISPLIDEMIA\_IPERTRIGLICERIDEMIA,  
 FUMO, IPERTENSIONE\_ARTERIOSA, OBESITÀ, IRC\_CKD, IRC\_STADIO5, CAD,  
 CAD\_FAMILIARITY, BPCO, ATRIAL\_FIBRILLATION, PREVIOUS\_IMA,  
 PREVIOUS\_PCI, PREVIOUS\_CABG, STROKE, REVASCULARIZATION\_PERIPHERAL,  
 AORTIC\_DISEASE, HEART\_FAILURE, PERIPHERAL\_ARTERY\_DISEASE, SVA\_SVM,  
 ICD, PM, CRT-D)

## 5 RIORGANIZZAZIONE E FILTRAGGIO DEL DATASET

Prima di procedere alla progettazione fisica e alla popolazione del database, è necessario **trasformare il dataset** preesistente e adattarlo al nuovo schema di basi di dati, mantenendo l'integrità semantica e ontologica delle informazioni. Il merito del compimento di questa operazione è del seguente **script Python**, che, ricorrendo alla libreria di analisi dei dati **Pandas**, riorganizza i file `.csv` che gli sono forniti in ingresso seguendo un insieme di regole specificate nel codice.

```
import pandas as pd
import os
import warnings

# IMPOSTAZIONI
cartella_input = 'GREZZI'
cartella_output = './cleaned_for_oracle2'
warnings.filterwarnings("ignore")
if not os.path.exists(cartella_output):
    os.makedirs(cartella_output)

# REGOLE
REGOLE = {
    'ANAGRAFICA.csv': {
        'date': ['DATA_NASCITA', 'DATA_DECESSO'],
        'interi': ['ID_PAZ', 'SEZIONE', 'COD_PAZ']
    },
    'TEXT.csv': {
        'date': [],
        'interi': ['ID_TEXT']
    },
    'ECOCARDIO_DATI.csv': {
        'date': ['DATA_EVENTO'],
        'interi': ['ID_ECOCARDIO', 'ID_PAZ', 'ID_TEXT', 'PAS', 'LVVD',
'LVVD_2CH', 'LVVS_2CH', 'E_TDI', 'LAV_2CH', 'JET_RIG_TRIC']
    },
    'ECOCAROTIDI.csv': {
        'date': ['DATA_EVENTO'],
        'interi': ['ID_ECOCAROTIDI', 'ID_PAZ', 'ID_TEXT', 'IMT_MAX']
    },
    'CORONAROGRAFIA_PTCA.csv': {
        'date': ['DATA_EVENTO'],
        'interi': ['ID_CORONOGRAFIA_PTCA', 'ID_PAZ', 'ID_TEXT']
    },
    'VISITA_CONTROLLO_ECG.csv': {
        'date': ['DATA_EVENTO'],
        'interi': ['ID_VISITA', 'ID_PAZ', 'ID_TEXT', 'PASDOMICILIO',
'PADDOMICILIO', 'FREQ_CLINO', 'FREQ_QORTO']
    },
    'ANAMNESI.csv': {
        'date': ['DATA_EVENTO'],
        'interi': ['ID_ANAMNESI', 'ID_PAZ', 'ID_TEXT']
    },
    'RICOVERO_OSPEDALIERO.csv': {
        'date': ['DATA_EVENTO'],
        'interi': ['ID_RICOVERO_OSPEDALIERO', 'ID_PAZ', 'ID_TEXT']
    },
    'ESAMI_STRUMENTALI_CARDIO.csv': {
        'date': ['DATA_EVENTO'],
        'interi': ['ID_STRUMENTALE', 'ID_PAZ', 'ID_TEXT']
    }
}
```

```

    },
    'ESAMI_SPECIALISTICI.csv': {
        'date': ['DATA_EVENTO'],
        'interi': ['ID_ESAME_SPECIALISTICO', 'ID_PAZ', 'ID_TEXT']
    },
    'ESAMI_LABORATORIO.csv': {
        'date': ['DATA_EVENTO'],
        'interi': ['ID_LAB', 'ID_PAZ', 'ID_TEXT', 'GPT']
    },
    'LISTA_EVENTI.csv': {
        'date': ['DATA_EVENTO'],
        'interi': ['ID_EVENTO', 'ID_PAZ']
    }
}

# FUNZIONI DI PULIZIA
def pulisci_intero(valore):
    """Prende una stringa, toglie i decimali se ci sono, restituisce stringa pulita."""
    s = str(valore).strip()
    if s == '' or s.lower() in ['nan', 'none', 'null', '<na>']:
        return ''
    try:
        # sostituisce virgola con punto
        s = s.replace(',', '.')
        # converte in float per gestire "123.0"
        num = float(s)
        # converte in int per togliere il decimale, poi di nuovo in stringa
        return str(int(num))
    except:
        # se fallisce restituisce stringa vuota
        return ''

def pulisci_generico(valore):
    """Pulisce spazi, a capo e sostituisce virgole con punti (per i decimali)."""
    if not isinstance(valore, str):
        return str(valore) if pd.isna(valore) else ''

    valore = valore.strip()
    valore = valore.replace('\n', ' ').replace('\r', ' ').replace(';', ',')

    # se sembra un numero con la virgola (eg 12,5), metto il punto (12.5)
    return valore.replace(',', '.')

# PROCESSING DEI FILE
def processa_file(nome_file):
    input_path = os.path.join(cartella_input, nome_file)
    output_path = os.path.join(cartella_output, nome_file)

    print(f"[{nome_file}] Sto lavorando...", end="")

    try:
        df = pd.read_csv(input_path, sep=None, engine='python',
                        encoding='utf-8', on_bad_lines='skip', dtype=str)

        # sostituisco i valori nulli di pandas con stringhe vuote
        df = df.fillna('')

        regole = REGOLE.get(nome_file, {})

```



```

cols_date = regole.get('date', [])
cols_interi = regole.get('interi', [])

# itero sulle nuove colonne
for col in df.columns:

    # CASO 1: DATE
    if col in cols_date:
        # converto in data e subito dopo in stringa YYYY-MM-DD
        # i valori errati diventano NaT, che poi riempio con ''
        df[col] = pd.to_datetime(df[col], dayfirst=True,
errors='coerce').dt.strftime(
            '%Y-%m-%d').fillna('')

    # CASO 2: INTERI
    elif col in cols_interi:
        # uso la funzione custom che lavora solo sulle stringhe
        df[col] = df[col].apply(pulisci_intero)

    # CASO C: TUTTO IL RESTO (Decimali e Testo)
    else:
        df[col] = df[col].apply(pulisci_generico)

# SALVATAGGIO
df.to_csv(output_path, sep=';', index=False, encoding='utf-8')
print(" -> OK (Salvato)")

except Exception as e:
    print(f"\n -> ERRORE: {e}")

# ESECUZIONE
files = [f for f in os.listdir(cartella_input) if f.endswith(
    '.csv')]

for f in files:
    if f.startswith("~$"):
        continue
    processa_file(f)

input("Premi INVIO...")

```

## 6 PROGETTAZIONE FISICA

Un primo approccio con l'infrastruttura del DBMS è avvenuto durante la fase di **Progettazione Fisica**, il cui obiettivo principale è stato rendere pronta la base di dati pronta ad ospitare il dataset appena filtrato e riorganizzato. In un primo luogo ci si è occupati di determinare il dimensionamento fisico della base di dati e codificare gli schemi relazionali tramite istruzioni DDL in SQL.

### 6.1 DIMENSIONAMENTO FISICO

Lo scopo primario della fase di dimensionamento fisico è quello di ottenere una stima del volume di memoria secondaria necessario per l'archiviazione persistente dei dati nel database. A partire dalle specifiche ottenute in fase di analisi dei dati, si determina per ogni singola relazione lo spazio occupato da ogni tupla, calcolato come somma dello spazio occupato da ogni suo attributo; il valore finale, corrispondente allo spazio occupato dall'intera relazione nella memoria secondaria, si ottiene moltiplicando il valore appena ottenuto per il numero di tuple stimate a regime. Lo spazio di archiviazione necessario, infine, viene determinato sommando le stime fatte per ogni singola relazione che compone lo schema di basi di dati.

#### 6.1.1 IL PAZIENTE

Tabella **PAZIENTI**

Attributo	Tipo Dato	Dimensione (Byte)
ID_PAZ	INT	4
SEZIONE	INT	4
DATA_NASCITA	DATE	7
SESSO	CHAR(1)	1
COMUNE_DI_NASCITA	VARCHAR(100)	100
MOTIVO_DECESSO	VARCHAR(200)	200
DATA_DECESSO	DATE	7
<b>TOTALE RIGA</b>		<b>323 Byte</b>

$$D_{PAZIENTI} = N_{TUPLE} \cdot D_{TUPLA} \cong 60.000 \cdot 325Byte \cong 19.5MB$$

## 6.1.2 LA STORIA CLINICA

Tabella **ECOCARDIO**

Attributo	Tipo Dato	Dimensione (Byte)
ID_ECOCARDIO	INT	4
DATA_EVENTO	DATE	7
ID_PAZ	INT	4
GATE_IN_INGRESSO	VARCHAR(100)	100
ID_TEXT	INT	4
EF	INT	4
AORTA_ASCENDENTE	INT	4
AORTA_ADDOMINALE	INT	4
PESO	FLOAT	4
ALTEZZA	FLOAT	4
PAS	INT	4
PAD	INT	4
BICUSPIDE	VARCHAR(100)	100
RR	INT	4
AORTA	INT	4
VSTD	INT	4
VSTS	INT	4
SIVTD	INT	4
PPTD	INT	4
PFVEPFVA	INT	4
EA	INT	4
DT	INT	4
LVVS	INT	4

LVVD	INT	4
LVVD_2CH	INT	4
LVVS_2CH	INT	4
E_TDI_SETTALE	INT	4
E_TDI	INT	4
LAV	INT	4
_2CH_LAV	INT	4
JET_RIG_TRIC	INT	4
TAPSE	INT	4
ATRIO	INT	4
INSUFFICIENZA_AORTICA	VARCHAR(100)	100
INSUFFICIENZA_MITRALICA	VARCHAR(100)	100
INSUFFICIENZA_TRICUSPIDALI CA	VARCHAR(100)	100
STENOSI_AORTICA	VARCHAR(100)	100
STENOSI_MITRALICA	VARCHAR(100)	100
TROMBO	VARCHAR(100)	100
HEART_FAILURE	VARCHAR(100)	100
AORTIC_DISEASE_ASCENDING_ ..	VARCHAR2(100)	100
AORTIC_DISEASE_ASCENDING_ ..	VARCHAR2(100)	100
AORTIC_DISEASE ABDOMINAL_ ..	VARCHAR2(100)	100
<b>TOTALE RIGA</b>		<b>1327 Byte</b>

$$D_{\text{ECOCARDIO}} = N_{\text{TUPLE}} \cdot D_{\text{TUPLA}} \cong 98.000 \cdot 1330\text{Byte} \cong 131\text{MB}$$

Tabella **ECOCAROTIDI**

Attributo	Tipo Dato	Dimensione (Byte)
ID_ECOCAROTIDI	INT	4
DATA_EVENTO	DATE	7
ID_PAZ	INT	4
GATE_DI_INGRESSO	VARCHAR(100)	100
ID_TEXT	INT	4
C_COMUNE_DIAMETRO	FLOAT	4
IMT_MAX	INT	4
MAXIMTNEARCIDX	INT	4
MAXIMTNEARCISX	INT	4
MAXIMTFARCIDX	INT	4
MAXIMTFARCISX	INT	4
MAXIMTNEARBDX	INT	4
MAXIMTNEARBSX	INT	4
MAXIMTFARBDX	INT	4
MAXIMTFARBSX	INT	4
MAXIMTNEARCCDX	INT	4
MAXIMTNEARCCSX	INT	4
MAXIMTFARCCDX	INT	4
MAXIMTFARCCSX	INT	4
ESAME_NELLA_NORMA	VARCHAR(100)	100
ATEROSCLEROSI_LIEVE	VARCHAR(100)	100
ATEROSCLEROSI_MODERATA	VARCHAR(100)	100
ATEROSCLEROSI_SEVERA	VARCHAR(100)	100
<b>TOTALE RIGA</b>		<b>575 Byte</b>

$$D_{\text{ECOCAROTIDI}} = N_{\text{TUPLE}} \cdot D_{\text{TUPLA}} \cong 51.000 \cdot 580\text{Byte} \cong 30\text{MB}$$

Tabella **PROCEDURE INTERVENTISTICHE**

Attributo	Tipo Dato	Dimensione (Byte)
ID_CORONOGRAFIA_PTCA	INT	4
DATA_EVENTO	DATE	7
ID_PAZ	INT	4
GATE_DI_INGRESSO	VARCHAR(100)	100
ID_TEXT	INT	4
LESIONI_TC	VARCHAR(100)	100
LESIONI_IVA	VARCHAR(100)	100
LESIONI_CX	VARCHAR(100)	100
LESIONI_DX	VARCHAR(100)	100
<b>TOTALE RIGA</b>		<b>519 Byte</b>

$$D_{P.\text{INTERVENTISTICHE}} = N_{\text{TUPLE}} \cdot D_{\text{TUPLA}} \cong 5.900 \cdot 519\text{Byte} \cong 3.1\text{MB}$$

Tabella **VISITE DI CONTROLLO ECG**

Attributo	Tipo Dato	Dimensione (Byte)
ID_VISITA	INT	4
DATA_EVENTO	DATE	7
ID_PAZ	INT	4
GATE_DI_INGRESSO	VARCHAR(100)	100
ID_TEXT	INT	4
DIABETE	VARCHAR(100)	100
DIABETE_INSULINE_TREATED	VARCHAR(100)	100

DIABETE_ORAL_THERAPY	VARCHAR(100)	100
DISPLIDEMIA_STATINE	VARCHAR(100)	100
DISPLIDEMIA_FIBRATI	VARCHAR(100)	100
BMI	FLOAT	4
OBESITÀ	VARCHAR(100)	100
PREVIOUS_IMA	VARCHAR(100)	100
CAD	VARCHAR(100)	100
FIBRILLAZIONE ATRIALE	VARCHAR(100)	100
PESO	FLOAT	4
ALTEZZA	FLOAT	4
PRESSIONE_ART_SISTOLICA_DOM	INT	4
PRESSIONE_ART_DIASTOLICA_DOM	INT	4
PRESSIONE_ART_CLINO_MIN	INT	4
PRESSIONE_ART_CLINO_MAX	INT	4
FREQ_CLINO	INT	4
PRESSIONE_ART_ORTO_MIN	INT	4
PRESSIONE_ART_ORTO_MAX	INT	4
FREQ_QORTO	INT	4
PRESSIONE_ART_SEDUTO_MIN	INT	4
PRESSIONE_ART_SEDUTO_MAX	INT	4
FREQ_SIT	INT	4
FUMO	VARCHAR(100)	100
DIURETICI	VARCHAR(100)	100
ALFABLOCCANTI	VARCHAR(100)	100
BETABLOCCANTI	VARCHAR(100)	100
ACEINIBITORI	VARCHAR(100)	100

AT1_ANTAGONISTI	VARCHAR(100)	100
DIIDROPIRI-DINICI	VARCHAR(100)	100
ANTIARITMICI	VARCHAR(100)	100
ANTIAGGREGANTI	VARCHAR(100)	100
ANTICOAGULANTI	VARCHAR(100)	100
SEGUETERAPIA	VARCHAR(100)	100
TIPO_DI_RICOVERO_CARDIO	VARCHAR(100)	100
TIPO_DI_RICOVERO_EXTRA	VARCHAR(100)	100
CAUSA_RICOVERO_EXTRA	VARCHAR(100)	100
NYHA	VARCHAR(100)	100
STADIO_SCOMPENSO	VARCHAR(100)	100
<b>TOTALE RIGA</b>		<b>2675 Byte</b>

$$D_{V.C.ECG} = N_{TUPLE} \cdot D_{TUPLA} \cong 163.000 \cdot 2675Byte \cong 437MB$$

Tabella **ANAMNESI**

Attributo	Tipo Dato	Dimensione (Byte)
ID_ANAMNESI	INT	4
DATA_EVENTO	DATE	7
ID_PAZ	INT	4
GATE_DI_INGRESSO	VARCHAR(100)	100
ID_TEXT	INT	4
DIABETE	VARCHAR(100)	100
DISPLIDEMIA	VARCHAR(100)	100
DISPLIDEMIA_IPERCOLEST.	VARCHAR(100)	100
DISPLIDEMIA_IPERTRIG.	VARCHAR(100)	100



DISPLIDEMIA_MISTA	VARCHAR(100)	100
FUMO	VARCHAR(100)	100
PREVIOUS_IMA	VARCHAR(100)	100
PREVIOUS_PCI	VARCHAR(100)	100
PREVIOUS_CABG	VARCHAR(100)	100
CAD	VARCHAR(100)	100
STROKE	VARCHAR(100)	100
CAD_FAMILIARITY_CEREBRAL	VARCHAR(100)	100
CAD_FAMILIARITY_CARDIAC	VARCHAR(100)	100
<b>TOTALE RIGA</b>		<b>1419 Byte</b>

$$D_{ANAMNESI} = N_{TUPLE} \cdot D_{TUPLA} \cong 57.000 \cdot 1419\text{Byte} \cong 81\text{MB}$$

Tabella **RICOVERI OSPEDALIERI**

Attributo	Tipo Dato	Dimensione (Byte)
ID_RICOVERO_OSPEDALIERO	INT	4
DATA_EVENTO	DATE	7
ID_PAZ	INT	4
GATE_DI_INGRESSO	VARCHAR(100)	100
ID_TEXT	INT	4
RICHIESTA_RICOVERO	VARCHAR(100)	100
SINTOMI	VARCHAR(100)	100
SINTOMI_PS	VARCHAR(100)	100
<b>TOTALE RIGA</b>		<b>419 Byte</b>

$$D_{RICOVERI\ OSPEDALIERI} = N_{TUPLE} \cdot D_{TUPLA} \cong 3.500 \cdot 419\text{Byte} \cong 1.5\text{MB}$$

Tabella **ESAMI STRUMENTALI**

Attributo	Tipo Dato	Dimensione (Byte)
ID_STRUMENTALE	INT	4
DATA_EVENTO	DATE	7
ID_PAZ	INT	4
GATE_DI_INGRESSO	VARCHAR(100)	100
ID_TEXT	INT	4
CONCLUSIONI	VARCHAR(100)	100
<b>TOTALE RIGA</b>		<b>219 Byte</b>

$$D_{E.STRUMENTALI} = N_{TUPLE} \cdot D_{TUPLA} \cong 4.000 \cdot 219Byte \cong 1MB$$

Tabella **ESAMI SPECIALISTICI**

Attributo	Tipo Dato	Dimensione (Byte)
ID_ESAME_SPECIALISTICO	INT	4
DATA_EVENTO	DATE	7
ID_PAZ	INT	4
GATE_DI_INGRESSO	VARCHAR(100)	100
ID_TEXT	INT	4
FT3	INT	4
FT4	INT	4
TSH	INT	4
<b>TOTALE RIGA</b>		<b>131 Byte</b>

$$D_{E.SPECIALISTICO} = N_{TUPLE} \cdot D_{TUPLA} \cong 22.000 \cdot 131Byte \cong 3MB$$

Tabella **ESAMI LABORATORIO**

Attributo	Tipo Dato	Dimensione (Byte)
ID_LAB	INT	4
DATA_EVENTO	DATE	7
ID_PAZ	INT	4
GATE_DI_INGRESSO	VARCHAR(100)	100
ID_TEXT	INT	4
GLICEMIA	INT	4
CREATININEMIA	INT	4
EMOGLOBINA	INT	4
URICEMIA	INT	4
SODIEMIA	INT	4
POTASSIEMIA	INT	4
CALCEMIA	INT	4
TRIGLICERIDI	INT	4
COLESTEROLO	INT	4
HDL	INT	4
PCR	INT	4
HBA1C_	INT	4
PERC	INT	4
MALBUMINURIA	INT	4
PROTEINURIA24	INT	4
OMOCISTEINEMI	INT	4
PROTEINE_TOTALI	INT	4
ALBUMINE	INT	4
GOT	VARCHAR(100)	100

GPT	INT	4
DIABETE	VARCHAR(100)	100
FILTRAT_GFR	INT	4
CKD	VARCHAR(100)	100
CKD_STADIO	VARCHAR(100)	100
<b>TOTALE RIGA</b>		<b>599 Byte</b>

$$D_{E.LABORATORIO} = N_{TUPLE} \cdot D_{TUPLA} \cong 123.000 \cdot 599Byte \cong 74MB$$

### 6.1.3 COMPLEMENTI TESTUALI

Tabella **TEXT**

Attributo	Tipo Dato	Dimensione (Byte)
ID_TEXT	INT	4
TESTO	CLOB	400
MATCHED_CONCEPTS	CLOB	400
MATCHED_CONCEPTS.1	CLOB	400
MATCHED_CONCEPT_TYPES	CLOB	400
DIABETE	VARCHAR(100)	100
DISPLIDEMIA	VARCHAR(100)	100
DISPLIDEMIA_IPERCOLESTEROLEMIA	VARCHAR(100)	100
DISPLIDEMIA_IPERTRIGLICERIDEMIA	VARCHAR(100)	100
FUMO	VARCHAR(100)	100
IPERTENSIONE_ARTERIOSA	VARCHAR(100)	100
OBESITA	VARCHAR(100)	100
IRC_CKD	VARCHAR(100)	100
IRC_STADIO5	VARCHAR(100)	100

CAD	VARCHAR(100)	100
CAD_FAMILIARITY	VARCHAR(100)	100
BPCO	VARCHAR(100)	100
ATRIAL_FIBRILLATION	VARCHAR(100)	100
PREVIOUS_IMA	VARCHAR(100)	100
PREVIOUS_PCI	VARCHAR(100)	100
PREVIOUS_CABG	VARCHAR(100)	100
STROKE	VARCHAR(100)	100
REVASCULARIZATION_PERIPHERAL	VARCHAR(100)	100
AORTIC_DISEASE	VARCHAR(100)	100
HEART_FAILURE	VARCHAR(100)	100
PERIPHERAL_ARTERY_DISEASE	VARCHAR(100)	100
SVA_SVM	VARCHAR(100)	100
ICD	VARCHAR(100)	100
PM	VARCHAR(100)	100
CRT-D	VARCHAR(100)	100
<b>TOTALE RIGA</b>		<b>4104 Byte</b>

$$D_{TEXT} = N_{TUPLE} \cdot D_{TUPLA} \cong 260.000 \cdot 4104Byte \cong 1070MB$$

In totale, sapendo che  $D_i$  è la dimensione di ogni tabella:

$$D_{TOT} = \sum_i D_i \cong 1851MB \cong 2000MB$$

## 6.2 DATA DEFINITION LANGUAGE

Il codice seguente traduce in istruzioni **DDL (Data Definition Language)** SQL le tabelle del modello relazionale ottenuto come risultato della fase di Progettazione Logica. Per ognuna delle tabelle in questione vengono definiti i tipi di dato specifici per gli attributi e i loro vincoli di integrità, inter-relazionali e intra-relazionali.

```

CREATE TABLE PAZIENTE (

    id_paz INT GENERATED BY DEFAULT AS IDENTITY,
    sezione INT,
    data_di_nascita DATE,
    sesso CHAR(1) CHECK (sesso IN ('F','M', 'U')),
    comune_di_nascita VARCHAR2(100),
    motivo_decesso VARCHAR2(100),
    data_decesso DATE,
    CONSTRAINT pk_paziente PRIMARY KEY(id_paz)
);

CREATE TABLE TEXT (

    id_text INT GENERATED BY DEFAULT AS IDENTITY,
    testo CLOB,
    matched_concepts CLOB,
    matched_concepts_1 CLOB,
    matched_concepts_type CLOB,
    diabete VARCHAR2(3) CHECK (diabete IN ('YES', 'NO')),
    displidemia VARCHAR2(3) CHECK (displidemia IN ('YES', 'NO')),
    displidemia_ipercolesterolemia VARCHAR2(3) CHECK
(displidemia_ipercolesterolemia IN ('YES', 'NO')),
    displidemia_ipertrigliceridemia VARCHAR2(3) CHECK
(displidemia_ipertrigliceridemia IN ('YES', 'NO')),
    displidemia_mista VARCHAR2(3) CHECK (displidemia_mista IN ('YES', 'NO')),
    fumo VARCHAR2(3) CHECK (fumo IN ('YES', 'NO')),
    ipertensione arteriosa VARCHAR2(3) CHECK (ipertensione arteriosa IN
('YES', 'NO')),
    obesita VARCHAR2(3) CHECK (obesita IN ('YES', 'NO')),
    irc_ckd VARCHAR2(3) CHECK (irc_ckd IN ('YES', 'NO')),
    irc_stadio5 VARCHAR2(3) CHECK (irc_stadio5 IN ('YES', 'NO')),
    cad VARCHAR2(3) CHECK (cad IN ('YES', 'NO')),
    cad_familiarity VARCHAR2(3) CHECK (cad_familiarity IN ('YES', 'NO')),
    bpcos VARCHAR2(3) CHECK (bpcos IN ('YES', 'NO')),
    atrial_fibrillation VARCHAR2(3) CHECK (atrial_fibrillation IN ('YES', 'NO')),
    previous_ima VARCHAR2(3) CHECK (previous_ima IN ('YES', 'NO')),
    previous_pci VARCHAR2(3) CHECK (previous_pci IN ('YES', 'NO')),
    previous_cabg VARCHAR2(3) CHECK (previous_cabg IN ('YES', 'NO')),
    stroke VARCHAR2(3) CHECK (stroke IN ('YES', 'NO')),
    revascularization_peripheral VARCHAR2(3) CHECK
(revascularization_peripheral IN ('YES', 'NO')),
    aortic_disease VARCHAR2(3) CHECK (aortic_disease IN ('YES', 'NO')),
    heart_failure VARCHAR2(3) CHECK (heart_failure IN ('YES', 'NO')),
    peripheral_artery_disease VARCHAR2(3) CHECK (peripheral_artery_disease IN
('YES', 'NO')),
    sva_svm VARCHAR2(3) CHECK (sva_svm IN ('YES', 'NO')),
    icd VARCHAR2(3) CHECK (icd IN ('YES', 'NO')),
    pm VARCHAR2(3) CHECK (pm IN ('YES', 'NO')),
    crt_d VARCHAR2(3) CHECK (crt_d IN ('YES', 'NO')),
    CONSTRAINT pk_text PRIMARY KEY(id_text)
);

CREATE TABLE ECOCARDIO (
id_ecocardio INT GENERATED BY DEFAULT AS IDENTITY,
    data_evento DATE,
    id_paz INT,
    gate_di_ingresso VARCHAR2(50),
    id_text INT,
    ef NUMBER,

```

```

aorta_ascendente NUMBER,
aorta_addominale NUMBER,
peso NUMBER,
altezza NUMBER,
pas NUMBER,
pad NUMBER,
bicuspide VARCHAR2(3) NOT NULL CHECK (bicuspide IN ('YES','NO')),
rr NUMBER,
aorta NUMBER,
vstd NUMBER,
vsts NUMBER,
sivtd NUMBER,
pptd NUMBER,
pfve NUMBER,
pfva NUMBER,
ea NUMBER,
dt NUMBER,
lvvs NUMBER,
lvvd INT,
lvvd_2ch INT,
lvvs_2ch INT,
e_tdi_settale NUMBER,
e_tdi INT,
lav NUMBER,
lav_2ch INT,
jet_rig_tric INT,
tapse NUMBER,
atrio NUMBER,
insufficienza_aortica VARCHAR2(10) NOT NULL CHECK (insufficienza_aortica IN
('NO','LIEVE','MODERATA','SEVERA')),
insufficienza_mitralica VARCHAR2(10) NOT NULL CHECK (insufficienza_mitralica
IN ('NO','LIEVE','MODERATA','SEVERA')),
insufficienza_tricuspidalica VARCHAR2(10) NOT NULL CHECK
(insufficienza_tricuspidalica IN ('NO','LIEVE','MODERATA','SEVERA')),
stenosi_aortica VARCHAR2(10) NOT NULL CHECK (stenosi_aortica IN
('NO','LIEVE','MODERATA','SEVERA')),
stenosi_mitralica VARCHAR2(10) NOT NULL CHECK (stenosi_mitralica IN
('NO','LIEVE','MODERATA','SEVERA')),
trombo VARCHAR2(3) NOT NULL CHECK (trombo IN ('YES','NO')),
heart_failure VARCHAR2(3) CHECK (heart_failure IN ('YES','NO')),
aortic_disease_ascending_dilation VARCHAR2(3) CHECK
(aortic_disease_ascending_dilation IN ('YES','NO')),
aortic_disease_ascending_aneurysm VARCHAR2(3) CHECK
(aortic_disease_ascending_aneurysm IN ('YES','NO')),
aortic_disease_abdominal_aneurysm VARCHAR2(3) CHECK
(aortic_disease_abdominal_aneurysm IN ('YES','NO')),
CONSTRAINT pk_ecocardio PRIMARY KEY(id_ecocardio)
);

CREATE TABLE ECOCAROTIDI (

id_ecocarotidi INT GENERATED BY DEFAULT AS IDENTITY,
data_evento DATE,
id_paz INT,
gate_di_ingresso VARCHAR2(50),
id_text INT,
c_comune diametro INT,
imt_max INT,
maximtnearcidx NUMBER,
maximtnearcix NUMBER,
maximtfarcidx NUMBER,

```

```

    maximtfarcisx    NUMBER,
    maximtnearbdx    NUMBER,
    maximtnearbbsx   NUMBER,
    maximtfarbdx     NUMBER,
    maximtfarbsx     NUMBER,
    maximtnearcddx   NUMBER,
    maximtnearcscsx  NUMBER,
    maximtfarccdx    NUMBER,
    maximtfarccsx    NUMBER,
    esame_nella_norma VARCHAR2(3) CHECK (esame_nella_norma IN ('YES','NO')),
    aterosclosi_lieve VARCHAR2(3) CHECK (aterosclosi_lieve IN ('YES','NO')),
    aterosclosi_moderata VARCHAR2(3) CHECK (aterosclosi_moderata IN
('YES','NO')),
    aterosclosi_severa VARCHAR2(3) CHECK (aterosclosi_severa IN
('YES','NO')),
    CONSTRAINT pk_ecocarotidi PRIMARY KEY(id_ecocarotidi)
);

CREATE TABLE PROCEDURA_INTERVENTISTICA (

    id_coronarografia_ptca INT GENERATED BY DEFAULT AS IDENTITY,
    data_evento DATE,
    id_paz INT,
    gate_di_ingresso VARCHAR2(50),
    id_text INT,
    lesioni_tc VARCHAR2(50),
    lesioni_iva VARCHAR2(50),
    lesioni_cx VARCHAR2(50),
    lesioni_dx VARCHAR2(50),
    CONSTRAINT pk_procedura_interventistica PRIMARY KEY(id_coronarografia_ptca)
);

CREATE TABLE VISITA_DI_CONTROLLO_ECG (
id_visita_di_controllo_ecg INT GENERATED BY DEFAULT AS IDENTITY,
    id_paz INT,
    data_evento DATE,
    gate_di_ingresso VARCHAR2(50),
    id_text INT,
    diabete VARCHAR2(3) CHECK (diabete IN ('YES','NO')),
    diabete_insuline_treated VARCHAR2(3) CHECK (diabete_insuline_treated IN
('YES','NO')),
    diabete_oral_therapy VARCHAR2(3) CHECK (diabete_oral_therapy IN
('YES','NO')),
    displidemia_statine VARCHAR2(3) CHECK (displidemia_statine IN ('YES','NO')),
    displidemia_fibrati VARCHAR2(3) CHECK (displidemia_fibrati IN ('YES','NO')),
    bmi NUMBER,
    obesita VARCHAR2(3) CHECK (obesita IN ('YES','NO')),
    previous_ima VARCHAR2(3) CHECK (previous_ima IN ('YES','NO')),
    cad VARCHAR2(3) CHECK (cad IN ('YES','NO')),
    fibrillazione_atriale VARCHAR2(3) CHECK (fibrillazione_atriale IN
('YES','NO')),
    peso NUMBER,
    altezza NUMBER,
    pasdomicilio INT,
    paddomicilio INT,
    paclinominm NUMBER,
    paclinomaxm NUMBER,
    freq_clino INT,
    paortominm NUMBER,
    paortomaxm NUMBER,

```



```

    freq_gorto INT,
    pressione_art_seduto_min NUMBER,
    pressione_art_seduto_max NUMBER,
    freq_sit NUMBER,
    fumo VARCHAR2(20),
    diuretici VARCHAR2(3) CHECK (diuretici IN ('YES','NO')),
    alfabloccanti VARCHAR2(3) CHECK (alfabloccanti IN ('YES','NO')),
    betabloccanti VARCHAR2(3) CHECK (betabloccanti IN ('YES','NO')),
    aceinibitori VARCHAR2(3) CHECK (aceinibitori IN ('YES','NO')),
    atl_antagonisti VARCHAR2(3) CHECK (atl_antagonisti IN ('YES','NO')),
    diidropiri_dinici VARCHAR2(3) CHECK (diidropiri_dinici IN ('YES','NO')),
    antiaritmici VARCHAR2(3) CHECK (antiaritmici IN ('YES','NO')),
    antiaggreganti VARCHAR2(3) CHECK (antiaggreganti IN ('YES','NO')),
    anticoagulanti VARCHAR2(3) CHECK (anticoagulanti IN ('YES','NO')),
    segueterapia VARCHAR2(3) CHECK (segueterapia IN ('YES','NO')),
    tipo_ricovero_cardio VARCHAR2(50),
    tipo_ricovero_extra VARCHAR2(50),
    causa_ricovero_extra VARCHAR2(50),
    nyha VARCHAR2(255),
    stadio_scompenso VARCHAR2(50),
    CONSTRAINT pk_visita_di_controllo_ecg PRIMARY
KEY(id_visita_di_controllo_ecg)
);

```

```
CREATE TABLE ANAMNESI (
```

```

    id_anamnesi INT GENERATED BY DEFAULT AS IDENTITY,
    data_evento DATE,
    id_paz INT,
    gate_di_ingresso VARCHAR2(50),
    id_text INT,
    diabete VARCHAR2(3) CHECK (diabete IN ('YES','NO')),
    displidemia VARCHAR2(3) CHECK (displidemia IN ('YES','NO')),
    displidemia_ipercolesterolemia VARCHAR2(3) CHECK
(displidemia_ipercolesterolemia IN ('YES','NO')),
    displidemia_ipertrigliceridemia VARCHAR2(3) CHECK
(displidemia_ipertrigliceridemia IN ('YES','NO')),
    displidemia_mista VARCHAR2(3) CHECK (displidemia_mista IN ('YES','NO')),
    fumo VARCHAR2(3) CHECK (fumo IN ('YES','NO')),
    previous_ima VARCHAR2(3) CHECK (previous_ima IN ('YES','NO')),
    previous_pci VARCHAR2(3) CHECK (previous_pci IN ('YES','NO')),
    previous_cabg VARCHAR2(3) CHECK (previous_cabg IN ('YES','NO')),
    cad VARCHAR2(3) CHECK (cad IN ('YES','NO')),
    stroke VARCHAR2(3) CHECK (stroke IN ('YES','NO')),
    cad_familiarity_cerebral VARCHAR2(3) CHECK (cad_familiarity_cerebral IN
('YES','NO')),
    cad_familiarity_cardiac VARCHAR2(3) CHECK (cad_familiarity_cardiac IN
('YES','NO')),
    CONSTRAINT pk_anamnesi PRIMARY KEY(id_anamnesi)
);

```

```
CREATE TABLE RICOVERO_OSPEDALIERO (
```

```

    id_ricovero_ospedaliero INT GENERATED BY DEFAULT AS IDENTITY,
    data_evento DATE,
    id_paz INT,
    gate_di_ingresso VARCHAR2(50),
    id_text INT,
    richiesta_ricovero VARCHAR2(50),
    sintomi VARCHAR2(50),

```

```

    sintomi_ps VARCHAR2(50),
    CONSTRAINT pk_ricovero_ospedaliero PRIMARY KEY(id_ricovero_ospedaliero)
);

```

```

CREATE TABLE ESAME_STRUMENTALE (

    id_esame_strumentale INT GENERATED BY DEFAULT AS IDENTITY,
    data_evento DATE,
    id_paz INT,
    gate_di_ingresso VARCHAR2(50),
    id_text INT,
    conclusioni VARCHAR2(50),
    CONSTRAINT pk_esame_strumentale PRIMARY KEY (id_esame_strumentale)
);

```

```

CREATE TABLE ESAME_SPECIALISTICO (
id_esame_specialistico INT GENERATED BY DEFAULT AS IDENTITY,
    data_evento DATE,
    id_paz INT,
    gate_di_ingresso VARCHAR2(50),
    id_text INT,
    ft3 NUMBER,
    ft4 NUMBER,
    tsh NUMBER,
    CONSTRAINT pk_esame_specialistico PRIMARY KEY(id_esame_specialistico)
);

```

```

CREATE TABLE ESAME_LABORATORIO (

    id_esame_laboratorio INT GENERATED BY DEFAULT AS IDENTITY,
    data_evento DATE,
    id_paz INT,
    gate_di_ingresso VARCHAR2(50),
    id_text INT,
    glicemia NUMBER,
    creatininemia NUMBER,
    emoglobina NUMBER,
    uricemia NUMBER,
    sodiemia NUMBER,
    potassiemia NUMBER,
    calcemia NUMBER,
    trigliceridi NUMBER,
    colesterolo NUMBER,
    hdl NUMBER,
    pcr NUMBER,
    hbalc_perc NUMBER,
    perc NUMBER,
    malbuminuria NUMBER,
    proteinuria24 NUMBER,
    omocisteinemia NUMBER,
    proteine_totali NUMBER,
    albumine NUMBER,
    got VARCHAR(50),
    gpt INT,
    diabete VARCHAR2(3) CHECK (diabete IN ('YES','NO')),
    filtrato_gfr NUMBER,
    ckd VARCHAR2(3) CHECK (ckd IN ('YES','NO')),
    ckd_stadio VARCHAR2(50),
    CONSTRAINT pk_esame_laboratorio PRIMARY KEY(id_esame_laboratorio)
);

```

```
);

--DDL CHIAVI ESTERNE

--DEFINIZIONE CHIAVI ESTERNE ECOCARDIO
ALTER TABLE ECOCARDIO
ADD CONSTRAINT fk_ecocardio FOREIGN KEY(id_paz) REFERENCES PAZIENTE(id_paz);
ALTER TABLE ECOCARDIO
ADD CONSTRAINT fk_ecocardio_text FOREIGN KEY(id_text) REFERENCES TEXT(id_text);

--DEFINIZIONE CHIAVI ESTERNE ECOCAROTIDI
ALTER TABLE ECOCAROTIDI
ADD CONSTRAINT fk_ecocarotidi FOREIGN KEY(id_paz) REFERENCES PAZIENTE(id_paz);
ALTER TABLE ECOCAROTIDI
ADD CONSTRAINT fk_ecocarotidi_text FOREIGN KEY(id_text) REFERENCES TEXT(id_text);

--DEFINIZIONE CHIAVI ESTERNE PROCEDURA INTERVENTISTICA
ALTER TABLE PROCEDURA_INTERVENTISTICA
ADD CONSTRAINT fk_procedura_interventistica FOREIGN KEY(id_paz) REFERENCES
PAZIENTE(id_paz);
ALTER TABLE PROCEDURA_INTERVENTISTICA
ADD CONSTRAINT fk_procedura_interventistica_text FOREIGN KEY(id_text) REFERENCES
TEXT(id_text);

--DEFINIZIONE CHIAVI ESTERNE VISITA DI CONTROLLO ECG
ALTER TABLE VISITA_DI_CONTROLLO_ECG
ADD CONSTRAINT fk_visita_di_controllo_ecg FOREIGN KEY(id_paz) REFERENCES
PAZIENTE(id_paz);
ALTER TABLE VISITA_DI_CONTROLLO_ECG
ADD CONSTRAINT fk_visita_di_controllo_ecg_text FOREIGN KEY(id_text) REFERENCES
TEXT(id_text);

--DEFINIZIONE CHIAVI ESTERNE ANAMNESI
ALTER TABLE ANAMNESI
ADD CONSTRAINT fk_anamnesi FOREIGN KEY(id_paz) REFERENCES PAZIENTE(id_paz);
ALTER TABLE ANAMNESI
ADD CONSTRAINT fk_anamnesi_text FOREIGN KEY(id_text) REFERENCES TEXT(id_text);

--DEFINIZIONE CHIAVI ESTERNE RICOVERO OSPEDALIERO
ALTER TABLE RICOVERO_OSPEDALIERO
ADD CONSTRAINT fk_ricovero_ospedaliero FOREIGN KEY(id_paz) REFERENCES
PAZIENTE(id_paz);
ALTER TABLE RICOVERO_OSPEDALIERO
ADD CONSTRAINT fk_ricovero_ospedaliero_text FOREIGN KEY(id_text) REFERENCES
TEXT(id_text);

--DEFINIZIONE CHIAVI ESTERNE ESAME STRUMENTALE
ALTER TABLE ESAME_STRUMENTALE
ADD CONSTRAINT fk_esame_strumentale FOREIGN KEY(id_paz) REFERENCES
PAZIENTE(id_paz);
ALTER TABLE ESAME_STRUMENTALE
ADD CONSTRAINT fk_esame_strumentale_text FOREIGN KEY(id_text) REFERENCES
TEXT(id_text);
```

```
--DEFINIZIONE CHIAVI ESTERNE ESAME SPECIALISTICO
ALTER TABLE ESAME_SPECIALISTICO
ADD CONSTRAINT fk_esame_specialistico FOREIGN KEY(id_paz) REFERENCES
PAZIENTE(id_paz);
ALTER TABLE ESAME_SPECIALISTICO
ADD CONSTRAINT fk_esame_specialistico_text FOREIGN KEY(id_text) REFERENCES
TEXT(id_text);

--DEFINIZIONE CHIAVI ESTERNE ESAME LABORATORIO
ALTER TABLE ESAME_LABORATORIO
ADD CONSTRAINT fk_esame_laboratorio FOREIGN KEY(id_paz) REFERENCES
PAZIENTE(id_paz);
ALTER TABLE ESAME_LABORATORIO
ADD CONSTRAINT fk_esame_laboratorio_text FOREIGN KEY(id_text) REFERENCES
TEXT(id_text);
```

## 7 QUERY, VISTE E TRIGGER

### 7.1 QUERY

Tramite le query, gli utenti del Sistema (ovvero i clinici del reparto di Cardiologia) possono estrarre, filtrare e analizzare i risultati di una ricerca secondo specifiche esigenze.

--QUERY 1: Trova i pazienti che hanno avuto un infarto pregresso, riporta le date di tutte le visite effettuate dal paziente e controlla se stanno prendendo farmaci corretti come antiaggreganti e betabloccanti.

```
SELECT
    P.id_paz AS id_paziente,
    A.previous_ima AS infarto_pregresso,
    VC.antiaggreganti,
    VC.betabloccanti,
    VC.data_evento AS data_visita
FROM PAZIENTE P
JOIN ANAMNESI A ON P.id_paz = A.id_paz
JOIN VISITA_DI_CONTROLLO_ECG VC ON P.id_paz = VC.id_paz
WHERE A.previous_ima = 'YES'
AND (VC.antiaggreganti = 'NO' OR VC.betabloccanti = 'NO');
```

--QUERY 2: Conta il numero di pazienti ricoverati e il numero totale dei ricoveri (valgono anche gli stessi pazienti che sono stati ricoverati più volte) a seguito di una diagnosi sulla presenza di scompenso.

```
SELECT
    E.heart_failure AS diagnosi_scompenso,
    COUNT(DISTINCT ro.id_paz) as pazienti_ricoverati,
    COUNT(ro.id_ricovero_ospedaliero) as numero_ricoveri
FROM RICOVERO_OSPEDALIERO RO
JOIN ECOCARDIO E ON RO.id_paz = E.id_paz
WHERE E.heart_failure IS NOT NULL
GROUP BY E.heart_failure
ORDER BY COUNT(RO.id_ricovero_ospedaliero) DESC;
```

--QUERY 3: Conta il numero di uomini e donne che assumono diuretici per vedere se c'è una prevalenza di trattamento in uno dei due sessi.

```
SELECT
    P.sesso,
    VC.diuretici AS assume_diuretici,
    COUNT(DISTINCT P.id_paz) AS numero_pazienti
FROM PAZIENTE P
JOIN VISITA_DI_CONTROLLO_ECG VC ON P.id_paz = VC.id_paz
WHERE P.sesso IN ('M', 'F')
AND VC.diuretici IN ('YES', 'NO')
GROUP BY P.sesso, VC.diuretici
ORDER BY P.sesso, VC.diuretici;
```

--QUERY 4: Calcola la dimensione atriale media e massima dei pazienti obesi e non obesi per paragonare i risultati e avere un indice sulla gravità dell'obesità.

```
SELECT
    VC.obesita AS stato_obesita,
    COUNT(DISTINCT P.id_paz) AS numero_pazienti,
    ROUND(AVG(E.atrio), 2) AS dimensione_media_atrio,
    ROUND(MAX(E.atrio), 2) AS dimensione_massima_rilevata
FROM PAZIENTE P
JOIN VISITA_DI_CONTROLLO_ECG VC ON P.id_paz = VC.id_paz
JOIN ECOCARDIO E ON P.id_paz = E.id_paz
```

```
WHERE VC.obesita IN ('YES', 'NO')
      AND E.atrio IS NOT NULL
GROUP BY VC.obesita
ORDER BY dimensione_media_atrio DESC;
```

--QUERY 5: Calcola i valori di filtrato glomerulare medio tra diabetici e non diabetici. Il valore medio più basso indica reni più stanchi e che filtrano peggio il sangue.

```
SELECT
      EL.diabete AS diagnosi_diabete,
      COUNT(DISTINCT EL.id_paz) AS numero_pazienti,
      ROUND(AVG(EL.filtrato_gfr),2) AS media_filtrato_gfr
FROM ESAME_LABORATORIO EL
WHERE EL.diabete IN ('YES', 'NO')
      AND EL.filtrato_gfr IS NOT NULL
GROUP BY EL.diabete
ORDER BY media_filtrato_gfr ASC;
```

## 7.2 VISTE

Le viste semplificano query complesse, migliorandone la sicurezza dei dati e limitando l'accesso a colonne o righe specifiche, così da rendere il codice riutilizzabile e centralizzando la logica delle query in una tabella virtuale.

--VISTA 1: Questa view permette di valutare un rischio preoperatorio per il paziente tenendo conto delle funzioni renali, infarti e ictus, o rischio emorragico per i farmaci assunti dallo stesso

```
CREATE OR REPLACE VIEW VW_RISCHIO_PREOPERAZIONE AS
SELECT
      P.id_paz,
      P.data_di_nascita,
      -- Dati Anamnestici Critici
      A.previous_ima AS infarto_pregresso,
      A.stroke AS ictus_pregresso,
      -- Dati Renali
      EL.ckd_stadio AS stadio_renale,
      EL.filtrato_gfr,
      -- Dati Metabolici ed Ematici
      EL.hbalt_perc AS emoglobina_glicata,
      EL.emoglobina AS anemia_preop,
      -- Rischio Emorragico dovuto all'assunzione dei farmaci
      VC.antiaggreganti,
      VC.anticoagulanti
FROM PAZIENTE P
JOIN ANAMNESI A ON P.id_paz = A.id_paz
LEFT JOIN ESAME_LABORATORIO EL ON P.id_paz = EL.id_paz
LEFT JOIN VISITA_DI_CONTROLLO_ECG VC ON P.id_paz = VC.id_paz;
```

--VISTA 2: Questa view isola i pazienti con patologie strutturali gravi (aneurismi o valvole rotte) così da programmare un intervento tempestivo qualora necessario

```
CREATE OR REPLACE VIEW VW_PLANNING_AORTA_VALVOLE AS
SELECT
      P.id_paz,
      E.data_evento AS data_eco,
      -- Anatomia Aortica
      E.aorta_ascendente,
      E.aorta_addominale,
      E.aortic_disease_ascending_aneurysm AS aneurisma_ascendente,
```

```

-- Funzionalità Valvolare
E.bicuspid AS valvola_bicuspid,
E.insufficienza_aortica,
E.stenosi_aortica,
E.insufficienza_mitralica,
E.stenosi_mitralica,
-- Funzione di Pompa (alto rischio se il valore è troppo basso)
E.ef AS frazione_eiezione
FROM PAZIENTE P
JOIN ECOCARDIO E ON P.id_paz = E.id_paz
WHERE E.aorta_ascendente > 40 OR E.insufficienza_aortica IN ('MODERATA', 'SEVERA')
OR E.stenosi_aortica IN ('MODERATA', 'SEVERA') OR E.insufficienza_mitralica IN
('MODERATA', 'SEVERA'); -- Mostra solo pazienti con patologia rilevante

--VISTA 3: Questa view è molto utile per sapere se le carotidi o le coronarie di
un paziente sono libere oppure otturate, generando rischi altissimi in caso di
intervento qualora fossero otturate
CREATE OR REPLACE VIEW VW_MAPPA_VASCOLARE AS
SELECT
    P.id_paz,
    -- Stato Coronarico
    PI.lesioni_tc AS lesione_tronco_comune, -- CRITICO: richiede operazione
urgente
    PI.lesioni_iva AS lesione_discendente_anteriore,
    PI.lesioni_cx AS lesione_circonflessa,
    PI.lesioni_dx AS lesione_coronaria_dx,
    -- Stato Carotideo
    EC.aterosclerosi_severa AS placca_carotidea_severa,
    EC.imt_max AS spessore_carotideo_max
FROM PAZIENTE P
LEFT JOIN PROCEDURA_INTERVENTISTICA PI ON P.id_paz = PI.id_paz
LEFT JOIN ECOCAROTIDI EC ON P.id_paz = EC.id_paz
WHERE PI.lesioni_tc LIKE '%YES%' OR EC.aterosclerosi_severa = 'YES'; -- Trova solo
chi ha almeno un problema rilevato

```

### 7.3 TRIGGER

I trigger sono eventi che reagiscono automaticamente in seguito ad un evento particolare sulla base di dati (come UPDATE, INSERT o DELETE) garantiscono, tramite la loro immediata reazione agli eventi, l'integrità dei dati.

```

--TRIGGER 1: Questo trigger blocca la modifica all'id di un paziente
CREATE OR REPLACE TRIGGER trg_blocca_modifica_id
BEFORE UPDATE ON PAZIENTE
FOR EACH ROW
BEGIN
    -- Se il nuovo ID è diverso dal vecchio ID, blocca tutto.
    IF :NEW.id_paz != :OLD.id_paz THEN
        RAISE_APPLICATION_ERROR(-20007, 'Errore Critico di Sicurezza: Non è
consentito modificare l''ID identificativo di un paziente. Cancellare e reinserire
l''ID');
    END IF;
END;
/

--TRIGGER 2: Questo trigger controlla che dei valori importanti come la frazione
di eiezione o la dimensione dell'atrio di un paziente non siano errati
CREATE OR REPLACE TRIGGER trg_validazione_ecocardio

```

```

BEFORE INSERT OR UPDATE ON ECOCARDIO
FOR EACH ROW
BEGIN
    -- Controllo Frazione di Eiezione (EF deve essere tra 0 e 100)
    IF :NEW.ef > 100 THEN
        RAISE_APPLICATION_ERROR(-20002, 'ERRORE DATI: La Frazione di Eiezione (EF)
non può essere maggiore di 100');
    END IF;
    --Controllo Dimensione Atrio (Non può essere negativo o maggiore di 150)
    IF :NEW.atrio <= 0 OR :NEW.atrio > 150 THEN
        RAISE_APPLICATION_ERROR(-20003, 'ERRORE DATI: Dimensione atrio non valida
(Range 0-150mm).');
    END IF;
END;
/

--TRIGGER 3: Questo trigger evita che ad un paziente con TROMBO impostata su YES
non gli vengano prescritti degli anticoagulanti
CREATE OR REPLACE TRIGGER trg_check_trombo_terapia
BEFORE INSERT OR UPDATE ON VISITA_DI_CONTROLLO_ECG
FOR EACH ROW
DECLARE
    v_presenza_trombo VARCHAR2(3);
BEGIN
    -- 1. Cerchiamo se esiste un ecocardiogramma recente con Trombo = YES per
questo paziente
    BEGIN
        SELECT trombo INTO v_presenza_trombo
        FROM ECOCARDIO
        WHERE id_paz = :NEW.id_paz
        AND trombo = 'YES'
        -- ROWNUM = 1 serve a prenderne almeno uno se ce ne sono tanti, per
evitare errori
        AND ROWNUM = 1;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            v_presenza_trombo := 'NO'; -- Nessun trombo trovato
    END;

    -- 2. Se c'è un trombo (YES) e si inserisce una visita SENZA anticoagulanti
restituisce errore
    IF v_presenza_trombo = 'YES' AND :NEW.anticoagulanti = 'NO' THEN
        RAISE_APPLICATION_ERROR(-20008, 'ALLERTA CLINICA: Il paziente ha un TROMBO
diagnosticato in Ecocardio. È obbligatorio prescrivere Anticoagulanti.');
```

diagnosticato in Ecocardio. È obbligatorio prescrivere Anticoagulanti.');

```

    END IF;
END;
/

--TRIGGER 4: Questo trigger evita la modifica di patologie pregresse (ictus) che
sono importanti nella cartella clinica di un paziente per avere un quadro generale
ben definito
CREATE OR REPLACE TRIGGER trg_stroke_irreversibile
BEFORE UPDATE ON ANAMNESI
FOR EACH ROW
BEGIN
    -- Se il vecchio valore era YES e il nuovo è NO, blocca tutto.
    IF :OLD.stroke = 'YES' AND :NEW.stroke = 'NO' THEN
        RAISE_APPLICATION_ERROR(-20010, 'Errore Clinico: Impossibile rimuovere lo
storico di Ictus (STROKE). Una volta diagnosticato, l'evento resta
nell'anamnesi.');
```



```

        END IF;
    END;
/

--TRIGGER 5: Questo trigger controlla che la pressione massima sia maggiore di
quella minima evitando errori
CREATE OR REPLACE TRIGGER trg_check_pressione_fisiologica
BEFORE INSERT OR UPDATE ON VISITA_DI_CONTROLLO_ECG
FOR EACH ROW
BEGIN
    -- Controlla se entrambi i valori della pressione domiciliare sono presenti
    IF :NEW.pasdomicilio IS NOT NULL AND :NEW.paddomicilio IS NOT NULL THEN

        -- ERRORE: La Massima (PAS) è minore o uguale alla Minima (PAD)
        IF :NEW.pasdomicilio <= :NEW.paddomicilio THEN
            RAISE_APPLICATION_ERROR(-20011, 'Errore Fisiologico: La pressione
Sistolica (Massima) deve essere superiore alla Diastolica (Minima). Controllare i
valori inseriti.');
```

```

            END IF;
        END IF;
    END;
/

--TRIGGER 6: Questo trigger controlla che il decesso sia completo di informazioni
(sia data che motivo, non possono esistere decessi senza motivo nè viceversa)
CREATE OR REPLACE TRIGGER trg_check_completeness_decesso
BEFORE INSERT OR UPDATE ON PAZIENTE
FOR EACH ROW
BEGIN
    IF :NEW.data_decesso IS NOT NULL AND :NEW.motivo_decesso IS NULL THEN
        RAISE_APPLICATION_ERROR(-20012, 'Errore Amministrativo: È obbligatorio
specificare il MOTIVO DEL DECESSO se viene inserita una Data di Decesso.');
```

```

    END IF;

    IF :NEW.motivo_decesso IS NOT NULL AND :NEW.data_decesso IS NULL THEN
        RAISE_APPLICATION_ERROR(-20013, 'Errore Amministrativo: Specificare la
DATA DEL DECESSO se è presente un motivo.');
```

```

    END IF;
END;
/

--TRIGGER 7: Questo trigger evita che vengano salvati 2 esami uguali per sbaglio
e che si abbia quindi uno spreco di risorse
CREATE OR REPLACE TRIGGER trg_no_doppioni_ecocardio
BEFORE INSERT ON ECOCARDIO
FOR EACH ROW
DECLARE
    v_conteggio INTEGER;
BEGIN
    -- Conta se esiste già un esame per questo paziente in questa data
    SELECT COUNT(*) INTO v_conteggio
    FROM ECOCARDIO
    WHERE id_paz = :NEW.id_paz
        AND data_evento = :NEW.data_evento;

    -- Se ne trova già uno, blocca il secondo inserimento
    IF v_conteggio > 0 THEN

```

```

        RAISE_APPLICATION_ERROR(-20014, 'Errore Duplicazione: Risulta già
inserito un Ecocardiogramma per questo paziente in data odierna.');
```

END IF;

END;

--TRIGGER 6: Questo trigger controlla che il decesso sia completo di informazioni  
(sia data che motivo, non possono esistere decessi senza motivo nè viceversa)

```

CREATE OR REPLACE TRIGGER trg_check_completeness_decesso
BEFORE INSERT OR UPDATE ON PAZIENTE
FOR EACH ROW
BEGIN
    IF :NEW.data_decesso IS NOT NULL AND :NEW.motivo_decesso IS NULL THEN
        RAISE_APPLICATION_ERROR(-20012, 'Errore Amministrativo: È obbligatorio
specificare il MOTIVO DEL DECESSO se viene inserita una Data di Decesso.');
```

END IF;

```

    IF :NEW.motivo_decesso IS NOT NULL AND :NEW.data_decesso IS NULL THEN
        RAISE_APPLICATION_ERROR(-20013, 'Errore Amministrativo: Specificare la
DATA DEL DECESSO se è presente un motivo.');
```

END IF;

END;

/

--TRIGGER 8: Questo trigger evita che vengano salvati 2 esami uguali per sbaglio  
e che si abbia quindi uno spreco di risorse

```

CREATE OR REPLACE TRIGGER trg_no_doppioni_ecocardio
BEFORE INSERT ON ECOCARDIO
FOR EACH ROW
DECLARE
    v_conteggio INTEGER;
BEGIN
    -- Conta se esiste già un esame per questo paziente in questa data
    SELECT COUNT(*) INTO v_conteggio
    FROM ECOCARDIO
    WHERE id_paz = :NEW.id_paz
        AND data_evento = :NEW.data_evento;

    -- Se ne trova già uno, blocca il secondo inserimento
    IF v_conteggio > 0 THEN
        RAISE_APPLICATION_ERROR(-20014, 'Errore Duplicazione: Risulta già
inserito un Ecocardiogramma per questo paziente in data odierna.');
```

END IF;

END;

/