Corso di Laurea Magistrale in INGEGNERIA CLINICA

UNIVERSITÀ DEGLI STUDI DI TRIESTE

RICHIAMI DI BASI DI DATI

Corso di Informatica Medica

Docente Aleksandar Miladinović

Dipartimento di Ingegneria e Architettura

Corso di Laurea Magistrale in INGEGNERIA CLINICA

UNIVERSITÀ DEGLI STUDI DI TRIESTE

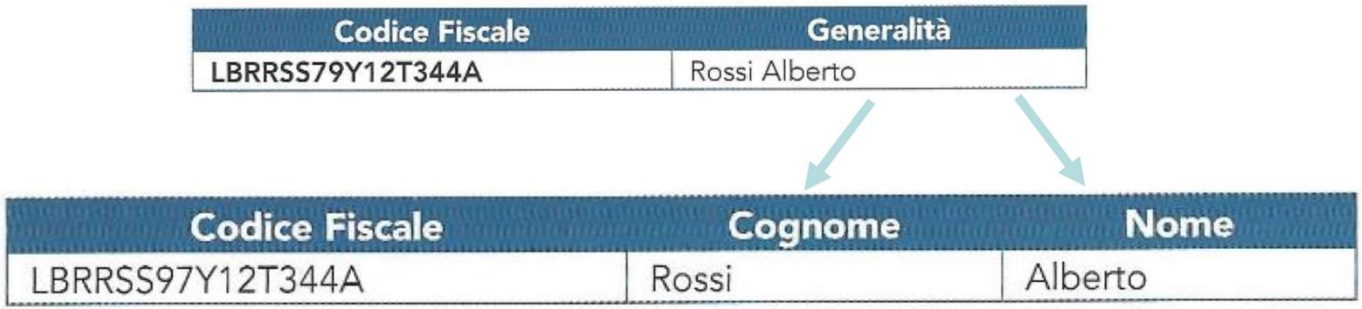
MODELLAZIONE CONCETTUALE E LOGICA DEI DATABASE

Dipartimento di Ingegneria e Architettura

Normalizzazione

UNIVERSITÀ DEGLI STUDI DI TRIESTE

- La normalizzazione è un processo implementato per minimizzare la ridondanza e decomporre le relazioni in relazioni più piccole.
- **Prima Forma Normale (atomicità):**
  - Dominio: insieme di valori consentiti per rappresentare il valore di un attributo.
  - La relazione è in prima forma normale se non contiene attributi composti o multi-valore. Una relazione è in prima forma normale se ogni attributo in essa è a valore singolo.



Codice Fiscale	Generalità
LBRRSS79Y12T344A	Rossi Alberto

Codice Fiscale	Cognome	Nome
LBRRSS97Y12T344A	Rossi	Alberto

Seconda Forma Normale

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Nome	Cognome	CF	Nome_terapia	PrincipioAttivo	CasaProduttrice
Luciana	Nunziatella	NNZLCN55R45F205N	Cumadin	warfarin	Bayer
Roberto	Marzio	MRZRRT71D04F251R	Corvel	carvedilolo	Dompe
Giampiero	Di Nicola	DNCGPR37L07H821Q	Enapren	enelapril	Zambon
Rosa	Russo	RSSRSA55G54F565G	Corvel	carvedilolo	Dompe
Arianna	Lucchini	LCCRNN82B42N127C	Cumadin	warfarin	Bayer
Giampiero	Di Nicola	DNCGPR37L07H821Q	Cumadin	warfarin	Bayer
Rosa	Russo	RSSRSA55G54F565G	Nebilox	Nebivololo	Lobivon

Domanda: Ci sono ridondanze? Qual è la chiave primaria?

Seconda Forma Normale e Chiavi Esterne

Questa ridondanza implica errori di aggiornamento.

- INSERT (per inserire un nuovo farmaco, è necessario inserire una nuova terapia prima).
- UPDATE (per aggiornare il nome di un farmaco, tutte le tuple contenenti quel nome devono essere aggiornate).
- DELETE (se un farmaco viene eliminato, si perde anche tutta l'informazione relativa al paziente e alla terapia).

Qual è la soluzione?

Seconda Forma Normale e Chiavi Esterne

UNIVERSITÀ DEGLI STUDI DI TRIESTE

PRIMARY KEY

TERAPIA

Therapy ID	PatientID	DrugID
1	1	2
2	1	3
3	…	

DrugID	Nome_Farmaco	Principio_Attivo	Casa_Produttrice
4	Nebivox	nebivololo	Lobivon
3	Enapren	enelapril	Zambon
2	Corvel	carvedilolo	Dompe
1	Cumadin	warfarin	Bayer

FOREIGN KEY

PatientID	Nome	Cognome	Sesso	DataNascita	CF
1	Roberto	Marzio	M	1971-04-25	MRZRRT71D04F251R
2	Giampiero	Di Nicola	M	1937-07-03	DNCGPR37L07H821Q
3	Luciana	Nunziatella	F	1955-05-11	NNZLCN55R45F205N
4	Arianna	Lucchini	F	1982-02-02	LCCRNN82B42N127C
5	Rosa	Russo	F	1955-07-14	RSSRSA55G54F565G
6	MariaLuisa	Gichetti	F	1953-08-25	GTCMLS53H65H832F

Definizione

Assunto che una relazione è già in una Prima Forma Normale e ogni attributo non chiave primaria è completamente dipendente funzionalmente dalla chiave primaria è in Seconda Forma Normale (2NF).

Suggerimento:

Tutti gli attributi che inserite in una relazione devono essere strettamente pertinenti all'informazione che state racchiudendo in quella relazione.

Seconda Forma Normale

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Nome	Cognome	CF	Nome_terapia	PrincipioAttivo	CasaProduttrice
Luciana	Nunziatella	NNZLCN55R45F205N	Cumadin	warfarin	Bayer
Roberto	Marzio	MRZRRT71D04F251R	Corvel	carvedilolo	Dompe
Giampiero	Di Nicola	DNCGPR37L07H821Q	Enapren	enelapril	Zambon
Rosa	Russo	RSSRSA55G54F565G	Corvel	carvedilolo	Dompe
Arianna	Lucchini	LCCRNN82B42N127C	Cumadin	warfarin	Bayer
Giampiero	Di Nicola	DNCGPR37L07H821Q	Cumadin	warfarin	Bayer
Rosa	Russo	RSSRSA55G54F565G	Nebilox	Nebivololo	Lobivon

Seconda Forma Normale

- **Prima Forma Normale e atomicità dell'attributo**

La Prima Forma Normale (1NF) richiede che tutti gli attributi di una relazione siano atomici, ovvero che contengano un solo valore indivisibile. Questo significa che non ci devono essere attributi con valori multipli o liste di valori all'interno di un singolo campo.

- **Seconda Forma Normale e atomicità del concetto**

La Seconda Forma Normale (2NF) si applica a una relazione che è già in 1NF e richiede che ogni attributo non chiave sia completamente dipendente dalla chiave primaria, eliminando le dipendenze parziali. In questo contesto, l'atomicità del concetto si riferisce alla necessità che i dati siano suddivisi in modo che ogni attributo dipenda interamente dalla chiave primaria, senza ridondanze o dipendenze parziali.

VINCOLO DI INTEGRITÀ REFERENZIALE: ESEMPIO

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Foreign key of PRESCRIPTIONS (FK)

PRESCRIPTIONS	Patient	Operative Unit	Doctor	Drug name
	1	3	1	Paracetamol
	3	2	1	Antibiotics
	1	3	2	Melatonin

La chiave esterna deve essere una chiave primaria in un'altra tabella

1. The two attributes have the same domain
2. The values occurring in Operative Unit occur in Unit_Number and are Primary Keys

OPERATIVE UNIT	Unit_Number	Name	Specialty	n.beds
	1	Cardiology 1	Cardiology	55
	2	G.Washington	Oncology	37
	3	M. Montessori	Pediatrics	47

Primary key of OPERATIVE UNIT (PK)

VINCOLO DI INTEGRITÀ REFERENZIALE

UNIVERSITÀ DEGLI STUDI DI TRIESTE

- Una chiave esterna può essere nulla?
  - Sì**, una chiave esterna può essere nulla**, a meno che non ci sia una restrizione specifica che lo impedisca (come una regola di integrità referenziale). **Quando una chiave esterna è nulla, significa che non esiste alcuna relazione con una riga nella tabella collegata**, il che è perfettamente accettabile in molti schemi di database, a seconda del contesto e delle necessità del modello di dati.
- Una chiave esterna deve assumere un valore che esiste già in un'altra tabella?
  - Sì, **una chiave esterna deve assumere un valore che esiste già nella chiave primaria della tabella a cui fa riferimento**, per mantenere l'integrità referenziale. Tuttavia, una chiave esterna può anche essere nulla, a meno che non sia esplicitamente vietato dalle regole del database. Quando non è nulla, deve corrispondere a un valore esistente nella tabella collegata.

CHIAVE ESTERNA

UNIVERSITÀ DEGLI STUDI DI TRIESTE

- Una **chiave esterna** (o **foreign key**) è un attributo o un insieme di attributi in una tabella che stabilisce un collegamento tra quella tabella e un'altra, facendo riferimento alla **chiave primaria** della tabella collegata. La chiave esterna impone l'integrità referenziale tra le due tabelle, garantendo che i valori presenti nel campo della chiave esterna corrispondano a valori validi nella chiave primaria della tabella di riferimento. In altre parole, una chiave esterna consente di creare una relazione tra tabelle, mantenendo la coerenza e integrità dei dati.

- La chiave esterna può assumere un valore nullo se la relazione non è obbligatoria, ma quando non è nulla, deve fare riferimento a un valore esistente nella chiave primaria della tabella collegata.

IL VALORE NULL: SIGNIFICATI MULTIPLI

UNIVERSITÀ DEGLI STUDI DI TRIESTE

1. Non valido per l'istanza corrente (es.,Secondo Cognome).
2. Valido ma non ancora esistente (es., Cognome del marito per una donna non sposata).
3. Esistente ma non può essere memorizzato (es., La religione del paziente non può essere memorizzata in alcuni paesi).
4. Esistente ma sconosciuto.
5. Esistente ma non ancora salvato (es., La storia del paziente non ancora raccolta).
6. Memorizzato e poi eliminato (informazioni errate).
7. Disponibile ma in fase di aggiornamento (es., Terapia del paziente in fase di modifica).
8. Disponibile ma non affidabile (una diagnosi non definitiva).
9. Disponibile ma non valida (es., un parametro del sangue al di sopra della soglia).
10. Calcolato da un altro valore NULL (es., BMI se manca il peso).

IL VALORE NULL: SIGNIFICATI MULTIPLI

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Per eliminare la situazione in cui un campo può rimanere nullo senza generare errori e gestire correttamente i dati, puoi utilizzare diverse soluzioni a livello applicativo. Ecco alcune strategie che puoi adottare:

1. **Selezionare valori predefiniti nell'interfaccia dell'applicazione**:

All'interno dell'applicazione, puoi fornire all'utente un elenco di valori predefiniti tra cui scegliere. Ad esempio, puoi usare un menu a tendina o una selezione obbligatoria per guidare l'utente nella scelta di un valore valido prima di inviare i dati al database. In questo modo, l'utente non può lasciare il campo vuoto o nullo.

2. **Impostare un valore di default nel database**:

Puoi configurare il database per riempire automaticamente il campo con un valore predefinito (default) se l'utente non inserisce alcun dato. Questo ti permette di evitare valori nulli, ma assicura che il campo contenga sempre un valore significativo per il contesto, anche se non è stato fornito dall'utente.

IL VALORE NULL: SIGNIFICATI MULTIPLI (2)

UNIVERSITÀ DEGLI STUDI DI TRIESTE

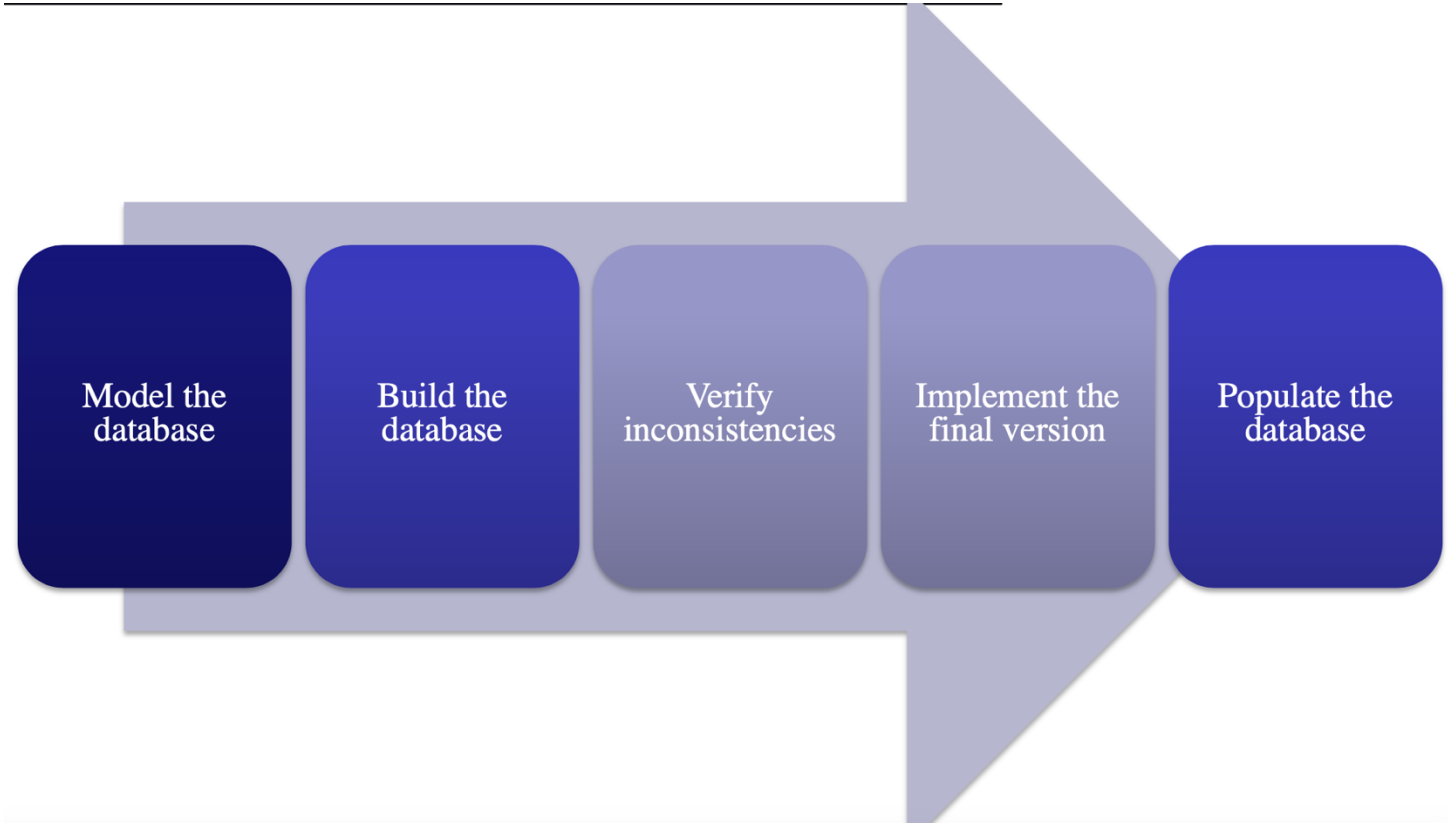
3. **Utilizzare un valore speciale come "non applicabile" (N/A)**:

In alcuni casi, può essere utile avere un valore standard come "N/A" (non applicabile) o "sconosciuto" per quei campi in cui il valore non è rilevante o disponibile. Questo ti permette di distinguere chiaramente i campi che non sono stati compilati volontariamente da quelli che sono effettivamente applicabili.

4. **Vincoli a livello di database**:

Puoi anche aggiungere un vincolo **NOT NULL** a livello di database. Questo vincolo impedisce che vengano inseriti valori nulli nel campo, costringendo l'applicazione o l'utente a inserire un dato valido o un valore predefinito.

Progettazione del Database: Ciclo di Vita (1)



Progettazione del Database: Ciclo di Vita (2)

Il ciclo di vita di un database è un processo complesso, solitamente composto dalle seguenti fasi principali:

1. Raccolta e analisi dei requisiti.
2. **Progettazione concettuale del database.**
3. Scelta di un Sistema di Gestione del Database (DBMS).
4. Progettazione logica del database.
5. Progettazione fisica del database.
6. Implementazione del database.
7. Uso e manutenzione.

Progettazione concettuale

UNIVERSITÀ DEGLI STUDI DI TRIESTE

- La **progettazione concettuale di un database relazionale** è la fase iniziale e fondamentale dello sviluppo di un database, durante la quale si definiscono i concetti di base e le relazioni tra i dati in modo indipendente dalla loro implementazione tecnica.
- L'obiettivo di questa fase è **creare un modello concettuale che rappresenti in maniera chiara e coerente i dati del mondo reale**, le loro proprietà e le relazioni tra di essi.

Progettazione concettuale (2)

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Nella progettazione concettuale, si utilizzano strumenti come i **diagrammi ER (Entity-Relationship)** per descrivere:

1. **Entità**: Oggetti o concetti rilevanti nel contesto del sistema, come “Paziente”, “Terapia”, o “Medico”.
2. **Attributi**: Proprietà o caratteristiche delle entità, come il “Nome” di un paziente o il “prncipio attivo” di un Farmaco.
3. **Relazioni**: Collegamenti logici tra entità, come la relazione tra “Paziente” e “Terapia”, in cui un paziente può avere zero o più terapie.
4. **Vincoli**: Regole che definiscono le proprietà delle entità e delle relazioni, come il vincolo di cardinalità che specifica il numero di entità che possono partecipare a una relazione.

Progettazione concettuale (3)

La progettazione concettuale è orientata a creare una visione ad alto livello dei dati che facilita la comprensione da parte degli stakeholder e getta le basi per la successiva **progettazione logica**, dove il modello concettuale viene tradotto in uno schema relazionale dettagliato.

Progettazione concettuale (3)

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Vista di alto livello e astratta della realtà.

Indipendente dal DBMS che verrà utilizzato.

→ Uno strumento tipico per questa rappresentazione è il **Modello dei Dati Entità-Relazione (E-R)**, che utilizza entità, attributi e relazioni per descrivere i dati e le loro interazioni in modo astratto. Questo modello rappresenta la base per la progettazione concettuale del database.

Modello dei Dati Entità-Relazione (E-R)

UNIVERSITÀ DEGLI STUDI DI TRIESTE

- Un'**entità** è un oggetto o **entità indipendente** del mondo reale.
- Una **relazione è un'associazione tra più entità**.

Attenzione:

Le RELAZIONI nel modello E-R sono DIFFERENTI dalle RELAZIONI nel modello relazionale.

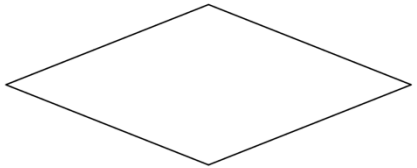
Modello dei Dati Entità-Relazione (E-R)

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Il modello **E-R (Entity-Relationship)** è un **linguaggio grafico standard** che utilizza diagrammi visivi per rappresentare concetti astratti, come entità, relazioni e attributi, in una forma intuitiva e facilmente comprensibile

ENTITY

RELATIONSHIP



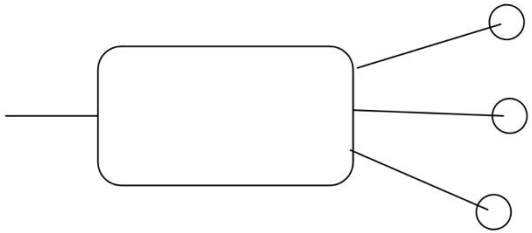
Modello dei Dati Entità-Relazione (E-R)

UNIVERSITÀ DEGLI STUDI DI TRIESTE

ATTRIBUTE

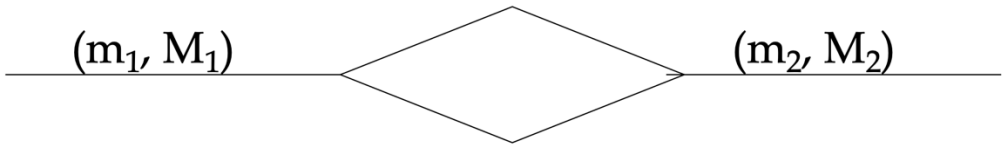
ATTRIBUTE WITH CARDINALITY

COMPOSITE ATTRIBUTE



Cardinalità delle Relazioni

La **Cardinalità delle Relazioni** nel modello E-R (Entity-Relationship) definisce **il numero di entità che possono partecipare a una determinata relazione**. Essa descrive come le entità in una relazione sono correlate tra loro in termini quantitativi.



RELATIONSHIP
CARDINALITY

Cardinalità delle Relazioni

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Le cardinalità delle relazioni si classificano in tre categorie principali:

1. Uno a Uno (1:1):

In una relazione **uno a uno**, a ciascuna occorrenza di un'entità A corrisponde al massimo una occorrenza di un'entità B, e viceversa.

• Esempio: Un cittadino può avere un solo passaporto e un passaporto può essere associato a un solo cittadino.

2. Uno a Molti (1:N)

In una relazione **uno a molti**, a ciascuna occorrenza di un'entità A possono essere associate molteplici occorrenze di un'entità B, ma ciascuna occorrenza di B può essere associata solo a una occorrenza di A.

• Esempio: Un professore può insegnare a molti studenti, ma ogni studente può avere un solo professore responsabile.

3. Molti a Molti (M:N):

In una relazione **molti a molti**, a ciascuna occorrenza di un'entità A possono essere associate molteplici occorrenze di un'entità B, e ciascuna occorrenza di B può essere associata a molteplici occorrenze di A.

• Esempio: Gli studenti possono iscriversi a molti corsi, e ciascun corso può essere frequentato da molti studenti.

Cardinalità delle Relazioni

Cardinalità Minima e Massima

Oltre alla cardinalità massima (1 o N), è possibile definire anche la **cardinalità minima**, che stabilisce il numero minimo di occorrenze necessarie.

Ad esempio, una relazione può richiedere che un'entità A abbia **almeno** un'associazione con un'entità B (cardinalità minima di 1) o nessuna (cardinalità minima di 0).

Modello dei Dati Entità-Relazione (E-R): Chiavi

KEY
ATTRIBUTES

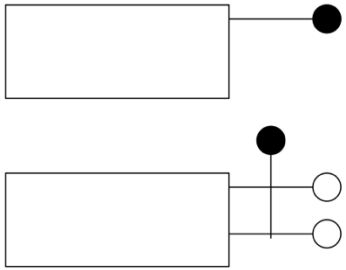
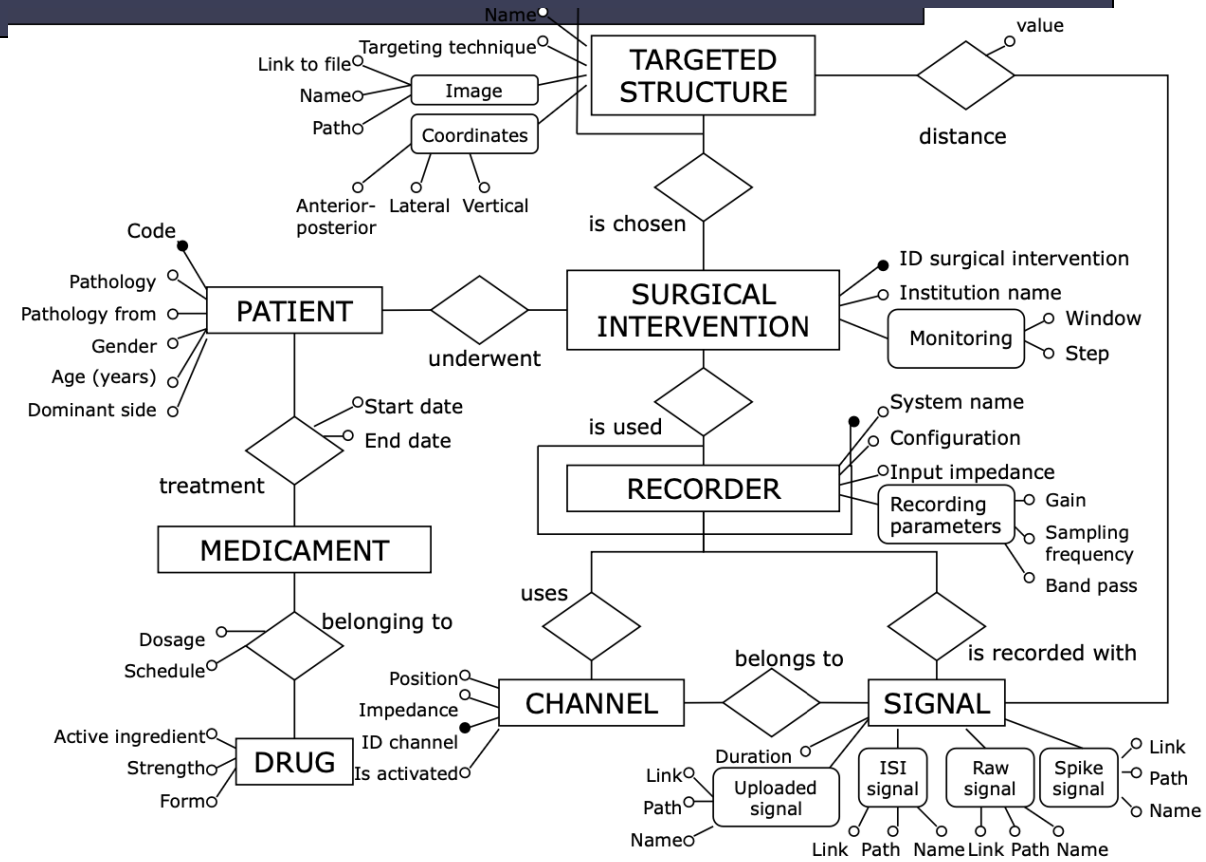


Diagramma E-R come descrittore di dominio informativo

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Il diagramma E-R è uno strumento grafico utilizzato per rappresentare il **dominio informativo** di un sistema. Esso descrive le entità principali, le relazioni tra di esse e le loro proprietà, aiutando a modellare i dati in modo comprensibile.

ESEMPIO


Name
Targeting technique
Link to file
Name
Path
Image
Coordinates
Anterior-posterior
Lateral
Vertical
TARGETED STRUCTURE
value
distance
is chosen
Code
Pathology
Pathology from
Gender
Age (years)
Dominant side
PATIENT
underwent
SURGICAL INTERVENTION
ID surgical intervention
Institution name
Monitoring
Window
Step
Start date
End date
treatment
is used
System name
Configuration
Input impedance
RECORDER
Recording parameters
Gain
Sampling frequency
Band pass
MEDICAMENT
uses
belonging to
Dosage
Schedule
belongs to
is recorded with
Position
Impedance
ID channel
Is activated
CHANNEL
SIGNAL
Active ingredient
Strength
Form
DRUG
Duration
Link
Path
Name
Uploaded signal
ISI signal
Raw signal
Spike signal
Link
Path
Name

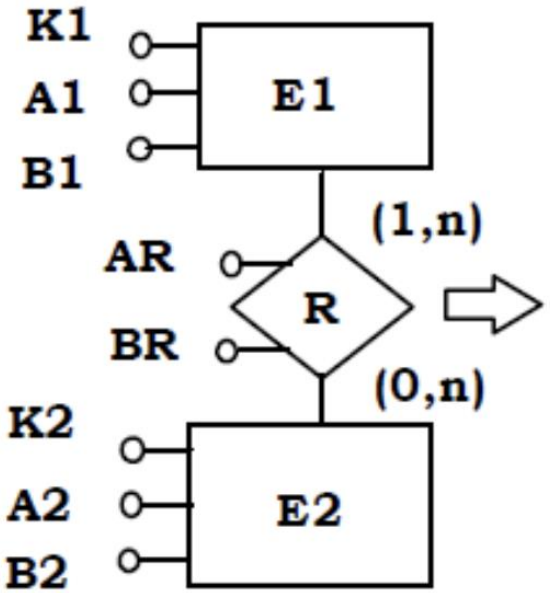
Progettazione Logica del Database

Traduzione della rappresentazione astratta del modello concettuale in specifiche che possono essere implementate tramite un DBMS.

Il risultato è lo **schema logico**.

1. Traduzione dallo schema concettuale allo schema logico utilizzando il modello di dati del DBMS.
2. Adattamento dello schema logico alle caratteristiche specifiche del DBMS.
3. Ottimizzazione dello schema logico → **Normalizzazione**.

Traduzione standard



E1 (<u>K1</u>, A1, B1,...)

E2 (<u>K2</u>, A2, B2,...)

R (<u>K1,K2</u>, AR, BR,...)

Traduzione standard 1:1

UNIVERSITÀ DEGLI STUDI DI TRIESTE

- **Per le relazioni 1:1**, la decisione di unire due entità o mantenerle separate dipende dalla natura del dominio informativo e dalla dipendenza degli attributi. Unire le entità può essere efficiente, ma bisogna considerare se tutti gli attributi rispettano la **Seconda Forma Normale**.
- **Mantenere separate le tabelle** è spesso la scelta migliore quando le entità hanno un certo grado di indipendenza o se ci sono attributi che non dipendono completamente dalla chiave primaria dell'unione.

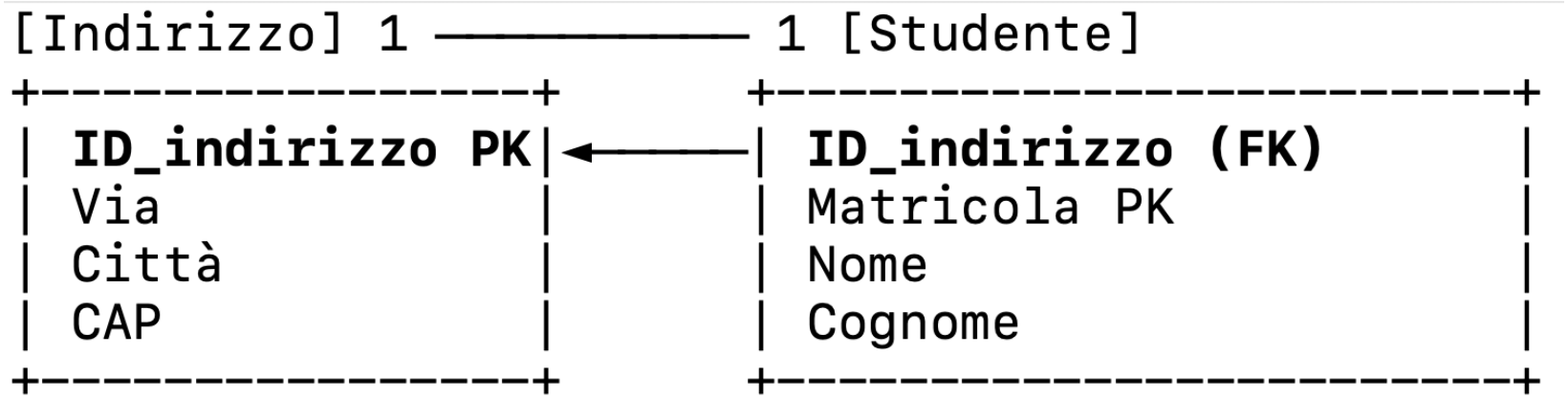
ESEMPIO:
Entità: Studente
Attributi: Matricola (PK), Nome, Cognome
Entità: Indirizzo
Attributi: ID_indirizzo (PK), Via, Città, CAP
Relazione: Ha_Indirizzo (1:1)
Ogni **studente** ha **un solo indirizzo**
Ogni **indirizzo** appartiene a **un solo studente**

Opzione A – Tabelle separate con FK
Studente
| Matricola (PK) | Nome | Cognome | ID_indirizzo **(FK → Indirizzo.ID_indirizzo)** |
Indirizzo
| ID_indirizzo (PK) | Via | Città | CAP |

Opzione B – Unione entità e relazione
Studente_Indirizzo
| Matricola (PK) | Nome | Cognome | Via | Città | CAP |

Traduzione standard 1:1

UNIVERSITÀ DEGLI STUDI DI TRIESTE



Traduzione standard 1:N

UNIVERSITÀ DEGLI STUDI DI TRIESTE

- **La chiave esterna viene sempre inserita dove la cardinalità è "molti" (N)**. Questo permette alla tabella con "N" di avere un riferimento alla tabella con "1", mantenendo il collegamento tra le due.
- Esempio

  Ogni **Cartella Clinica** può avere **molti Referti**, ma ogni **Referto** è associato a **una sola Cartella Clinica**.

  La **chiave esterna** (ID_CartellaClinica) viene quindi posizionata nella tabella **Referti** (dove la partecipazione è "N"), in modo che ogni referto sappia a quale cartella clinica è collegato

Esempio:

Entità: CartellaClinica

Attributi: ID_CartellaClinica (PK), Paziente, Data_apertura

Entità: Referto

Attributi: ID_Referto (PK), Data_referto, Descrizione

Relazione: Contiene (1:N)

Una **Cartella Clinica** può avere molti **Referti**

Ogni **Referto** appartiene a una sola **Cartella Clinica**

Regola: La **chiave esterna** si mette sempre dalla parte "N", cioè nella tabella che può avere più record.

Tabelle:

CartellaClinica

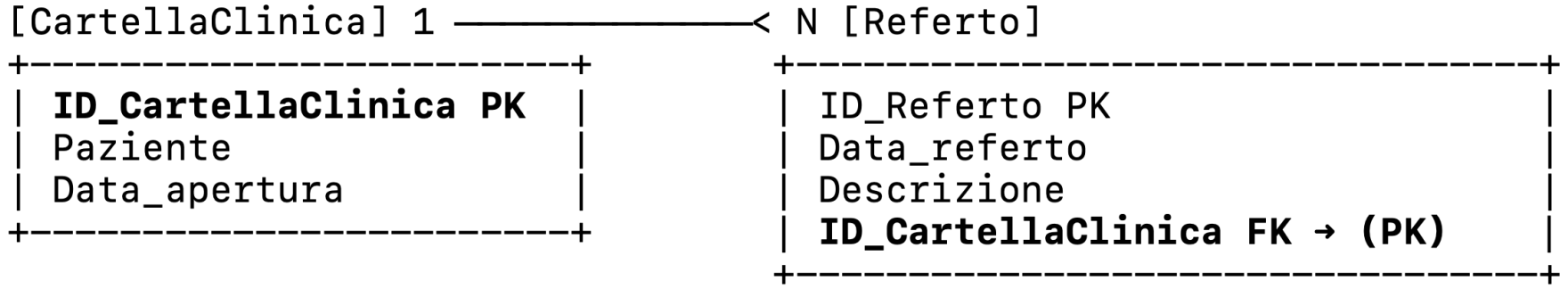
| ID_CartellaClinica (PK) | Paziente | Data_apertura |

Referto

| ID_Referto (PK) | Data_referto | Descrizione | ID_CartellaClinica **(FK→CartellaClinica.ID_CartellaClinica)** |

La FK ID_CartellaClinica permette a ogni Referto di sapere a quale Cartella Clinica appartiene.

Traduzione standard 1:N



Traduzione standard N:N

UNIVERSITÀ DEGLI STUDI DI TRIESTE

In una relazione **N:N (molti a molti)** tra due entità in un database relazionale, è necessario creare una **tabella ausiliaria** (o tabella di associazione) per gestire questa relazione. La tabella ausiliaria permette di collegare le due entità in modo che ogni riga rappresenti una combinazione di elementi collegati tra le due tabelle originali.

In una relazione **N:N**, un'entità può essere collegata a molte altre, e viceversa. Tuttavia, i database relazionali non possono direttamente rappresentare una relazione molti-a-molti. Pertanto, si utilizza una **tabella di associazione** per trasformare questa relazione in due relazioni **1:N**.

Entità: Paziente
Attributi: ID_Paziente (PK), Nome, Cognome, Data_nascita
Entità: Medico
Attributi: ID_Medico (PK), Nome, Cognome, Specializzazione
Relazione: Visita (N:N)
Un **Paziente** può avere molte **Visite con diversi Medici**
Un **Medico** può visitare molti **Pazienti**

Paziente
| ID_Paziente (PK) | Nome | Cognome | Data_nascita |

Medico
| ID_Medico (PK) | Nome | Cognome | Specializzazione|

Visita (tabella di associazione)
| **ID_Paziente (FK → Paziente.ID_Paziente)** | **ID_Medico (FK → Medico.ID_Medico)** | Data_visita | Note |
PK composta da (ID_Paziente, ID_Medico, Data_visita) per identificare univocamente ogni visita

Traduzione standard N:N