

# APPUNTI DI INFORMATICA MEDICA – DISPERAT\* NON FREQUENTANTE

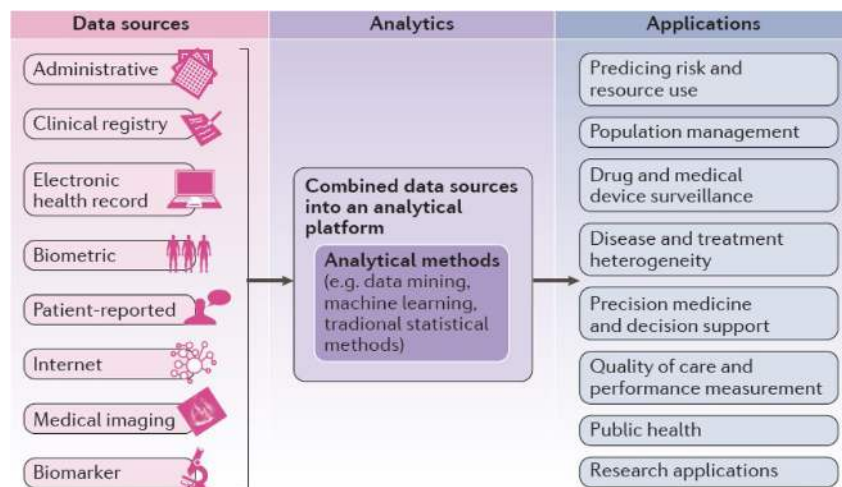
## DOMANDE DI TEORIA

### PARTI 0-4

#### 1. Dare una definizione di Informatica Medica

- Applicazioni dell'**informatica** alla **medicina**.
- La scienza che si occupa della **gestione in sanità dell'informazione** e dei programmi basati su calcolatore.
- Applicazione di **metodologie dell'Ingegneria/Scienza dell'Informazione alla conoscenza medica**, con l'obiettivo di fornire un supporto alla risoluzione di problematiche attinenti a diagnosi, terapia e prevenzione.

#### 2. Spiegare con uno schema cosa si intende per ausilio alla decisione in diagnosi



#### 3. Spiegare perché il vecchio sistema di finanziamento degli ospedali è stato sostituito

- Il nuovo sistema è stato implementato al fine di creare **separazione delle funzioni di acquisto** (enti che pagano i servizi sanitari per i cittadini e quindi, indirettamente, i cittadini) **e di produzione** (strutture sanitarie che curano i cittadini). Il vecchio sistema è stato sostituito per due motivi:
  - Introdurre meccanismi concorrenziali all'interno di strutture sanitarie pubbliche.
  - Potenziare il ruolo di controllo (sia pur indiretto) da parte dei pazienti fruitori del servizio.

#### 4. Spiegare l'idea alla base del nuovo sistema di finanziamento degli ospedali in vigore in Italia da metà degli anni '90

- a. L'**acquirente** Stato (o la Regione, in un sistema non centralizzato) **paga servizi sanitari** con l'obiettivo di migliorare il benessere dei propri assistiti. Le **strutture produttrici** di servizi sanitari (es. le aziende ospedaliere, i servizi sanitari territoriali, cliniche private convenzionate, ...) **competono tra loro** per acquisire pacchetti di prestazioni assistenziali.
- b. Ogni anno, **lo Stato stabilisce in anticipo la spesa sanitaria nazionale complessiva e trasferisce fondi alle varie Regioni**, principalmente sulla base della popolazione residente, in modo da assicurare livelli uniformi di assistenza sanitaria su tutto il territorio nazionale. Ogni Regione **ripartisce**, con criteri simili, il finanziamento **tra le varie Aziende Sanitarie Locali (ASL)**. Le ASL rimborsano ad Aziende Ospedaliere, Servizi a Gestione Diretta ASL e Servizi Privati Convenzionati, **una quantità di denaro che dipende dal numero e dalla complessità degli interventi assistenziali effettuati, indipendentemente dai costi effettivamente sostenuti**. Gli introiti dei produttori obbediscono quindi ad un sistema "a prestazione" (ogni prestazione pagata "a forfait").

#### 5. Illustrare i vantaggi teorici del sistema di rimborso a prestazione

- a. I vantaggi sono principalmente due:
  - i. L'**acquirente** (es. Regione) viene **responsabilizzato a promuovere gli interessi del cittadino**, assicurandosi che ottenga il miglior servizio erogabile con le limitate risorse stanziare.
  - ii. **I produttori** (es. ospedali, cliniche private convenzionate, ...) sono spinti a **concentrarsi sul servizio e a competere tra di loro per assicurarsi quante più commesse possibili** (se non ottengono commesse, non acquisiscono "budget" ...).

#### 6. Illustrare cosa si intende per DRG

- a. Con DRG (diagnosis-related groups) si fa riferimento a un **sistema** che permette di **classificare tutti i pazienti dimessi da un ospedale** (ricoverati in regime ordinario o day hospital) in **gruppi omogenei per assorbimento di risorse impegnate (isorisorse)**.

#### 7. Illustrare come viene determinato in pratica il DRG di un ricovero

- a. Esso viene determinato in base a vari fattori quali:
  - i. i **tipi di intervento** (nel caso C) o di **procedure** (nel caso M);
  - ii. l'**età** del/la paziente (cfr. con lista dei 492 DRG, che ogni tanto distingue per l'età...);
  - iii. le **patologie secondarie** (cfr. comorbidità e complicanze);
  - iv. lo **stato alla dimissione** (cfr. al medico inviante, trasferito, deceduto, ...).

## 8. Illustrare quali sono gli input fondamentali del grouper

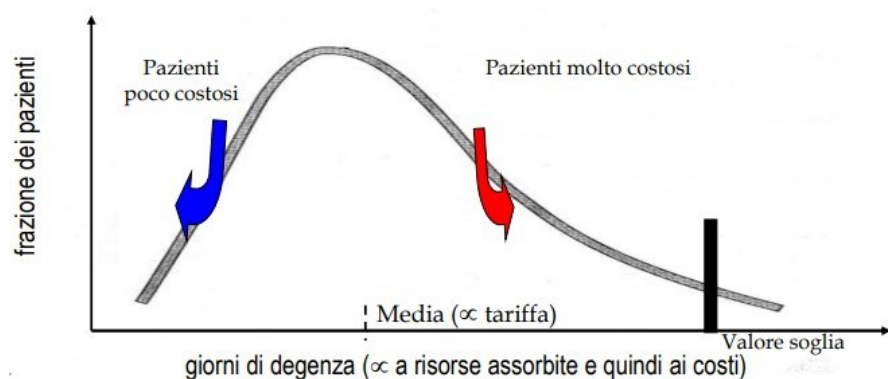
- a. Essi sono i seguenti:
  - i. diagnosi **principale**;
  - ii. diagnosi **secondarie**;
  - iii. **interventi e procedure adottate**;
  - iv. **età e sesso** del/la paziente;
  - v. **stato alla dimissione** (dimess\* a domicilio, trasferit\* ad altro ospedale per acuti, dimess\* contro il parere dei sanitari, decedut\*);
  - vi. **durata** degenza.

## 9. Perché il sistema dei DRG è detto iso-risorse?

- a. Perché esso va a classificare l'insieme di tutti i possibili casi clinici in categorie riguardanti **patologie simili** e il cui **decorso richiede mediamente la stessa intensità di risorse per il trattamento** (si dicono patologie iso-risorse).

## 10. Spiegare, con l'ausilio di un grafico, come il ministero fissa la tariffa di rimborso di uno specifico DRG

- a. Il peso economico di un DRG è riferito ad un ipotetico "costo medio per quel tipo di DRG", ottenuto dal Ministero studiando in modo retrospettivo la distribuzione dei costi dei ricoveri per quel DRG. La tariffa **T1** corrisponde all'ipotetico costo di un **ricovero di un singolo giorno**, la **T2** al **costo medio per giornata moltiplicato per la durata media dei ricoveri del DRG**, la **T3** al costo rimborsato in più per **ricoveri di durata anomala** (la cui durata supera la soglia del 97 percentile).



**11. Spiegare, con l'ausilio di un grafico, perché certi pazienti risultano più costosi della media**

- a. Il trattamento di **pazienti con condizioni cliniche particolari che portano a degenze eccezionalmente lunghe** (outliers) può comportare uno scostamento significativo nel consumo di risorse rispetto alla media della categoria di appartenenza. Per ricoveri anomali per durata di degenza, si rimborsa quindi la tariffa standard (indicata nel seguito con T2) incrementata di una cifra data dal prodotto fra il numero di giornate di degenza oltre il "valore soglia" e una tariffa giornaliera (indicata nel seguito con T3) specifica del DRG di appartenenza.

**12. Spiegare, eventualmente con l'ausilio di un grafico, come la legge tratta i ricoveri anomali**

- a. Spiegato sopra.

**13. Spiegare quanti e quali sono i tipi di tariffa di rimborso previsti dal sistema di finanziamento degli ospedali in relazione ai ricoveri**

- a. Ci sono 3 tariffe di rimborso per i ricoveri:
  - i. **T1**: per i ricoveri di un solo giorno (la ASL rimborsa all'Ospedale la cifra T1);
  - ii. **T2**: per i ricoveri standard;
  - iii. **T3**: per i ricoveri oltre la soglia del 97%.

**14. Illustrare almeno tre pro e almeno tre contro del sistema dei DRG**

- a. Il sistema dei DRG presenta i seguenti pro:
  - i. **Aumento dell'efficienza;**
  - ii. **Qualificazione dell'attività;**
  - iii. **Riduzione della durata media della degenza;**
  - iv. **Calo dei costi sociali.**
- b. Ci sono tuttavia anche i seguenti contro:
  - i. Dai codici riportati sulla SDO si arriva, per ogni paziente dimesso, ad un unico DRG, anche se il paziente è stato etichettato con diagnosi multiple, di diversa complessità -> **semplificazione eccessiva.**
  - ii. Mancando la logica retrospettiva e andando considerata la sola logica prospettiva, costi fissi ed investimenti sono **ammortizzabili solo se viene prodotto un adeguato volume di prestazioni.**
  - iii. La **durata della degenza** rappresenta, a torto o a ragione, un costo da **abbattere.**

- iv. Il sistema rimborsa (a forfait) la cura, **non necessariamente la qualità della cura**. Ospedali con attrezzature/procedure all'avanguardia vengono rimborsati allo stesso modo di quelli meno avanzati.

**15. Illustrare con un esempio almeno due dei possibili abusi cui può condurre un sistema di finanziamento basato sui DRG ma privo di controlli efficaci**

- a. Ci sono i seguenti possibili abusi:
  - i. La durata della degenza rappresenta, a torto o a ragione, un costo da abbattere.
  - ii. **Riduzione artificiosa** della durata della degenza, con dimissione intempestiva del paziente.
  - iii. Aumento del numero di ricoveri per pazienti le cui tariffe risultano, nello specifico ospedale, superiori rispetto al costo di produzione, con **aumento dei ricoveri inappropriati**.
  - iv. **Selezione dei pazienti**, discriminazioni nei confronti dei pazienti più complessi.
  - v. **Spezzettamento della cura** con più ricoveri separati per accaparrarsi più volte la tariffa T2.
  - vi. **Manipolazione della SDO** per spingere l'attribuzione al ricovero di un DRG a peso/tariffa superiore.

**16. Illustrare il significato dell'indice di case-mix**

- a. L'**indice di Case-Mix (ICM)**, o grado di complessità dei casi trattati, del reparto i-esimo è definito come  $ICM_i = (\sum_j p_{ij} D_j) / D_{std}$ , dove  $D_{std} = \sum_j P_j D_j$ , dove  $j \in \{1, 2, \dots, n_{DRG}\}$ .
- b. Ovvero, l'**ICM<sub>i</sub>** indica **il rapporto fra la durata attesa dei ricoveri nello standard se la frequenza relativa dei vari DRG fosse quella effettivamente riscontrata al reparto i-esimo e la durata attesa dei ricoveri nello standard ( $D_{std}$ )**. Se  $ICM_i > 1$  la casistica del reparto i-esimo è  $D_{std}$  più complessa dello standard, se  $ICM_i < 1$  la casistica è meno complessa dello standard (nel reparto i-esimo avrei  $ICM_i = 1$  se ad esempio avessi la stessa frequenza dei casi dello standard per ogni DRG). L'indice di Case-Mix permette di valutare la complessità media dei ricoveri effettuati in un determinato reparto (o Az. Osp). Un ospedale che ha una alta specializzazione (es. un Policlinico) giustifica un ICM elevato.

**17. Illustrare il significato dell'indice comparativo di performance**

- a. L'**indice comparativo di performance (ICP)** del reparto i-esimo è definito come  $ICP_i = (\sum_j P_j d_{ij}) / D_{std}$ , dove  $D_{std} = \sum_j P_j D_j$  è la durata attesa dei ricoveri nello standard e dove  $j \in \{1, 2, \dots, n_{DRG}\}$ .

- b. L'ICP<sub>i</sub> considera il **rapporto fra la durata attesa dei ricoveri nello standard se la durata dei ricoveri per i vari DRG** nello standard fosse quella effettivamente riscontrata nel reparto i-esimo e la durata attesa dei ricoveri nello standard. L'indice comparativo di performance (ICP) valuta quindi l'efficienza del reparto i-esimo rispetto allo standard. Un ICP<sub>i</sub> <1 indica efficienza del reparto i-esimo migliore dello standard, mentre un ICP<sub>i</sub> >1 indica una efficienza peggiore. Avrei ICP<sub>i</sub>=1, ad esempio, se per ogni DRG registrassi nel reparto la stessa durata del ricovero standard, cioè se  $d_{ij}=D_j$ .

**18. Un reparto fornisce prestazioni su 2 DRG, di cui sono note le caratteristiche (D1: 12 gg, D2: 4 gg, P1: 0.3, P2: 0.7, d11: 11 gg, d12: 5 gg, p11: 0.6, p12: 0.4). Si calcolino gli indici ICM e ICP per il reparto e se ne faccia un commento**

- a.  $D_{std} = 12 \cdot 0.3 + 4 \cdot 0.7 = 6.4$
- b.  $ICM = (0.6 \cdot 12 + 0.4 \cdot 4) / 6.4 = 1.375$
- c.  $ICP = (0.3 \cdot 11 + 0.7 \cdot 5) / 6.4 = 1.0625$
- d. Il reparto ha una casistica significativamente peggiore dello standard, ma un'efficienza sostanzialmente nello standard (lievemente peggiore).

## PARTE 5

### 1. Spiegare la necessità di un sistema informativo in sanità

- a. La necessità di un sistema informativo in sanità deriva dai **vari flussi informativi che operano all'interno di una struttura ospedaliera**, in particolare:
- i. Molte attività e molti dati da gestire: gestione quotidiana del paziente, memorizzazione ed emissione di cartelle-richieste-referti, gestione del personale sanitario, gestione strumentazione-farmaci-materiali, gestione globale del sistema sanitario.
  - ii. Molte decisioni da prendere: decisioni cliniche, decisioni amministrative.
  - iii. Molti cicli di gestione delle informazioni: distinti, interconnessi, disordinati/eterogenei/imprevedibili.

### 2. Illustrare almeno 3 possibili funzioni di un sistema informativo sanitario

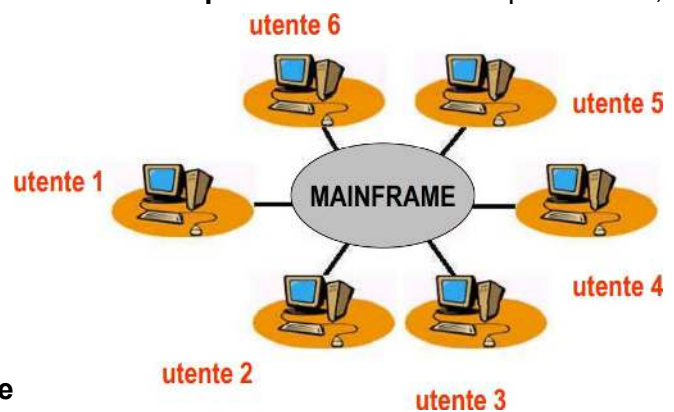
- a. Queste sono le seguenti necessità ed usi dei sistemi informativi in sanità (SIS):
- i. Gestire efficacemente i **dati** sanitari.
  - ii. Gestire efficacemente l'**accesso ai dati** all'interno del sistema (es. dalle varie componenti nei vari poli del sistema).

- iii. Gestire la **connettività verso l'esterno** (es. condivisione dati entro la stessa ASL, refertazione accessibile da rete ad esempio per teleconsulto o visione dal medico di base...).
- iv. Gestire la **connettività verso banche dati** (es. linee guida, protocolli, letteratura scientifica, ...).
- v. Fornire **supporto alle decisioni cliniche** (es. fornitura di metodologie di ausilio alla decisione clinica).
- vi. Gestione **prestazioni sanitarie** (accettazione, prenotazione esami, ...).
- vii. **Organizzazione delle attività amministrative** (es. gestione del personale, fatturazioni, relazioni con ASL e Regione...).
- viii. **Calcolo di dati statistici ed epidemiologici** (es. calcolo distribuzione di ricoveri e/o degenze, dati da usare da ASL e Ministero Salute per statistiche e/o controllo della correttezza dei rimborsi, ...).
- ix. **Supporto alle decisioni organizzative** (es. calcolo indicatori di produttività reparto per reparto, ...).

### 3. Confrontare le architetture mainframe e distribuite di un SIS

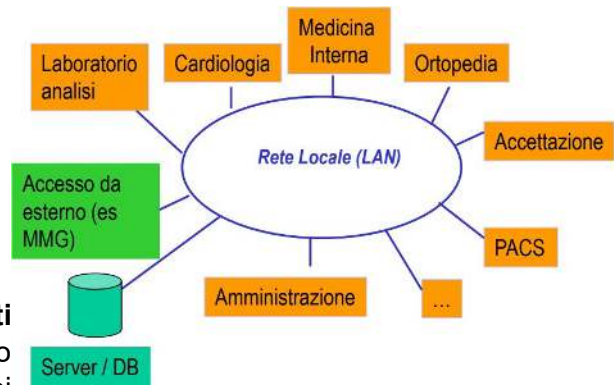
- a. Una soluzione mainframe prevede un'unica macchina che serve tutte le necessità e funziona da **applicazione multiutente multiscopo**. Essa è facile da implementare, ha però i seguenti svantaggi:

- i. **Complesso.**
- ii. **Costoso.**
- iii. **Rigido.**
- iv. **Poco robusto.**
- v. **Difficilmente espandibile e aggiornabile.**



- b. La soluzione **client-server (distribuita)** prevede un **unico server/data base centrale e una rete locale LAN**, e diversi poli dipartimentali aventi ognuno i propri software e le proprie esigenze. Ognuno dei reparti può operare all'interno di una rete attraverso questa struttura che consentirà anche l'accesso dall'esterno per esempio al medico di medicina generale e consentirà anche la gestione dei dati di tipo amministrativo anche per l'interfacciamento con la regione. In questo caso **le procedure e le applicazioni rimangono specifiche dei vari servizi, ad essere messe in comune sono solo le informazioni**. I vantaggi sono i seguenti:

- i. **No ridondanza.**
- ii. **Meno lavoro di codifica.**
- iii. **Meno rischio di avere informazioni errate o inconsistenti.**
- iv. **Maggiore sicurezza per i dati**  
(in quanto vi è solo lo spostamento non fisico dei dati, e non dei pazienti).
- v. **Maggior facilità nella gestione degli accessi.**



#### 4. Cosa si intende per sistema informativo formale?

- a. In un sistema informativo informale, i dati obbediscono solo ad una struttura minima; mentre in un **sistema informativo formale**, i dati sono **strutturati in base ad un certo modello ben definito**.

#### 5. Quali sono i pro e i contro di un sistema informativo formale in ambito sanitario?

- a. I vantaggi possono essere notevoli e sono i seguenti:
  - i. **Ordine.**
  - ii. **Affidabilità.**
  - iii. **Coerenza.**
  - iv. **Efficienza.**
  - v. **Gestibilità dei dati.**
- b. Tuttavia ci sono anche notevoli svantaggi:
  - i. **Alti costi di progettazione.**
  - ii. **Maggiore burocratizzazione.**
  - iii. **Perdita di flessibilità nella descrizione clinica della casistica** (spesso inaccettabile vista la grande variabilità della casistica clinica).

#### 6. Quando nella pratica può essere tollerabile una soluzione informale?

- a. Essa è tollerabile o addirittura preferibile quando si tratti di **processi infrequenti** o ad **alta variabilità** o che coinvolgono solo un **numero limitato di persone** o quando **il contenuto informativo non è prevedibile in anticipo** (e conseguentemente non è neanche prevedibile il tipo e numero di campi).

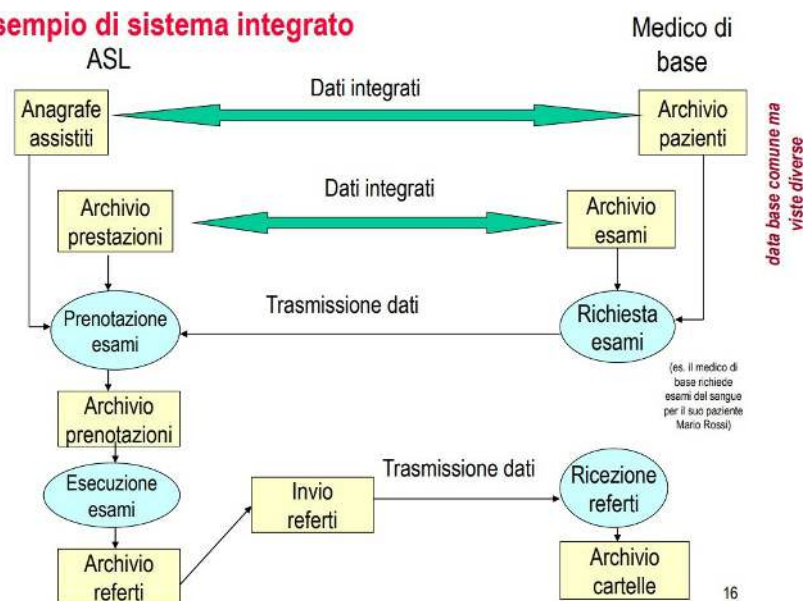


## 7. Illustrare con un esempio un SIS completamente integrato ed uno non integrato

### a. Esempio di SIS integrato:

- i. Composto da almeno tre tipi di componenti. Una è la componente sul territorio del **medico di base**; poi abbiamo un sistema gestionale cioè il **centro unico di prenotazione**, e poi abbiamo un sistema dipartimentale che è quello del **laboratorio di analisi**. Abbiamo un sistema completamente integrato, in cui il medico di base e l'ASL accedono allo **stesso database** cioè dati sono perfettamente integrati con viste diverse cioè il medico di base vedrà solo dati dei suoi pazienti. Nel momento in cui il medico di base richiede un esame, non c'è problema di codifica. Al CUP arriva la richiesta di prenotazione dell'esame, e l'esito dell'esame arriva direttamente al medico di base, il quale ripeto fa accesso allo stesso database.

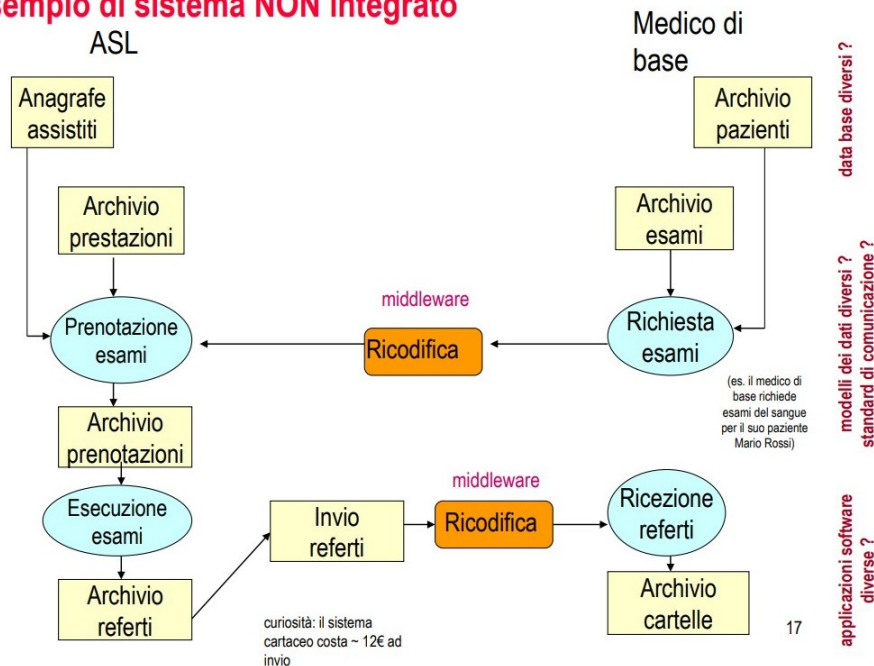
#### Esempio di sistema integrato



### b. Esempio di SIS non integrato:

- i. In questo caso, in cui il medico di base, ha la **propria cartella clinica elettronica**, può **interagire col SIS**, può raggiungere il CUP e fare una richiesta di esami per un paziente. È chiaro che quel paziente ha una codifica nella cartella clinica del medico di base che non è la stessa della codifica nel database dell'ASL quindi i **database sono separati**. Quindi è necessario un **software di inframezzo**, un **middleware**, che sia in grado di far capire che si sta chiedendo una prestazione per un certo tipo di paziente, identificato in un certo modo a livello dell'ASL, e in un altro modo a livello del medico di base. Una volta erogata la prestazione, anche la visione del referto da parte del software di cartella clinica del medico di base, non può essere immediata, serve una **rimodifica delle informazioni** sul referto, per renderle leggibili dal software di cartella clinica elettronica del medico di base. Questo è un sistema per cui l'investimento è centrato nel middleware, il quale assicura l'interoperabilità tra sistemi diversi, e però è il sistema che più facilmente si può realizzare.

## Esempio di sistema NON integrato



## 8. Quali sono le difficoltà principali in cui si incorre nell'utilizzo di realizzazioni di SIS non completamente integrate?

- È impensabile realizzare SIS completamente integrato, cioè è complicato **pensare ad un modello di dati che possa andare bene per tutte le specifiche delle varie realtà dell'ambito clinico**. Quindi bisogna usare sistemi che sono **solo parzialmente integrati**, e per farli funzionare bene, bisogna riuscire a far interoperare le diverse componenti del SIS che sono sviluppate in maniera indipendente, hanno applicativi indipendenti, a volte anche sviluppati da ditte concorrenti tra loro. Quindi la sfida odierna riguarda la **interoperabilità, cioè la creazione di sistemi che siano in grado di scambiare informazioni e cooperare**, perciò vogliamo far interoperare sistemi che sono stati sviluppati per risolvere le specifiche problematiche delle varie realtà sanitarie.

## 9. Illustrare i concetti di standardizzazione semantica, terminologica, sintattica

- Standardizzazione semantica:** le **informazioni corrispondenti** devono essere **strutturate nello stesso modo**.
- Standardizzazione terminologica:** implementazione di un **sistema di nomenclatura e codifica riconosciuto** (a livello nazionale o, meglio, internazionale), in cui lo stesso codice corrisponde a tutti i possibili sinonimi di un certo termine.
- Standardizzazione sintattica:** pratica di definire regole e convenzioni precise per la struttura e la formattazione dei dati nei sistemi informativi sanitari. Questa standardizzazione è fondamentale per consentire l'interscambio affidabile e l'interpretazione coerente delle informazioni tra diversi sistemi informatici e applicazioni nel campo della sanità. Bisogna infatti che gli applicativi siano in grado di comunicare tra di loro. Per poter **scambiare informazioni tra componenti diverse di un SIS** (o tra applicativi diversi o tra software diversi di cartella clinica).

elettronica o tra strumentazione e calcolatori), ovvero per interoperare tra sistemi, bisogna utilizzare un **linguaggio standard**. Lo standard più diffuso si chiama **HL7**.

#### 10. Illustrare l'importanza della standardizzazione terminologica in medicina

- a. Per **trattare in maniera automatica le informazioni**, è fondamentale che ogni concetto clinico abbia una codifica. Volendo abbiamo già visto un sistema di codifica dei concetti medici, riguardo il raggruppamento omogeneo di diagnosi che ammette una codifica con un numero intero da 1 a 492. Indubbiamente la standardizzazione terminologica è importante per garantire l'**interoperabilità** ma è anche fondamentale per l'**analisi di dati clinici** (per ricerca, per analisi economiche, ...). È importante anche perché **se i dati sono tutti codificati, ci sarà una gestione intelligente dei dati**.

#### 11. Perché si fa corrispondere la nascita dell'epidemiologia all'introduzione di un sistema di standardizzazione terminologica?

- a. L'introduzione di un sistema di standardizzazione terminologica ha permesso di **definire in modo più preciso e uniforme le malattie, i loro sintomi e i metodi di trasmissione**. Questo ha **facilitato la comunicazione e la collaborazione** tra i ricercatori, permettendo un **approccio più sistematico e scientifico** allo studio delle malattie a livello di popolazione.

#### 12. Illustrare il significato di concetto, di termine, di codice, e di gruppo

- a. Un **termine** esprime un **concetto medico**. **Diversi termini possono essere utilizzati per lo stesso concetto** (esempio i termini "respiro corto", "difficoltà di respiro", "dispnea" corrispondono al concetto di "difficoltà di respiro").
- b. Un **codice alfanumerico** viene **associato ad ogni distinto concetto**.
- c. Un **gruppo** raccoglie in ogni **singola categoria** un certo numero di codici relativi a **concetti differenti, ma considerati simili per un certo fine**.

#### 13. Descrivere in breve le caratteristiche generali di ICD-9-CM

- a. L'**ICD (International Classification of Diseases)** rappresenta lo **standard terminologico** organizzato dall'organizzazione mondiale della sanità. È un sistema enumerativo (elenca in anticipo tutti i termini possibili che potrebbero venire usati), e come tale:
  - i. Richiede **continua revisione**.
  - ii. È **ridondante** (codici leggermente diversi per concetti molto simili).
  - iii. È relativamente **semplice da fare**.
  - iv. **Costoso** da mantenere/revisionare/aggiornare/arricchire.
- b. Rappresenta il **riferimento de facto** per molte terminologie e sistemi di rimborso adottati dal Ministero della Salute per la SDO (da cui si traggono informazioni statistiche e il DRG dei ricoveri). Il ICD-9-CM **traduce in codici alfa-numerici i termini** con cui sono espressi diagnosi, procedure e interventi. In ICD-9-CM le

informazioni, relative a **13000 diagnosi** e **oltre 3000 interventi e procedure**, e sono organizzate gerarchicamente nel seguente modo:

- i. 17 **capitoli**: ogni capitolo comprende vari codici relativi alle malattie attinenti ad una **stessa tipologia clinica**.
- ii. **Blocchi**: insiemi di condizioni tra loro **strettamente correlate**.
- iii. **Categorie: codici a tre caratteri**, alcuni dei quali già molto specifici e non ulteriormente suddivisibili.
- iv. **Sottocategorie**: codici a quattro caratteri; il quarto carattere fornisce **ulteriore specificità o informazione** relativamente ad eziologia, localizzazione o manifestazione clinica.
- v. **Sotto-classificazioni**: codici a cinque caratteri (livello atomico, non suddivisibile).

**14. Descrivere in breve come sono organizzate le malattie in ICD-9-CM 15.**

**15. Descrivere il sistema di codifica numerica delle malattie in ICD-9-CM 16.**

**16. Illustrare un esempio di problematica nell'uso pratico dei sistemi terminologici standard come ICD - 9 -CM e come si può affrontare**

a. Come sopra +:

- i. I codici per procedure e interventi sono costituiti da **2 a 4 caratteri** numerici. Quando sono necessari più di due caratteri, **un punto decimale è interposto tra il secondo e il terzo**.
- ii. Le rubriche comprese fra **01 e 86** comprendono **interventi chirurgici maggiori**, endoscopie e biopsie.
- iii. Le rubriche comprese fra **87 e 99** comprendono **altre procedure** diagnostiche e terapeutiche.
- iv. I **primi due** identificano generalmente un **organo**.
- v. Il **terzo e il quarto** specificano generalmente la **sede** e il **tipo** dell'intervento.

## PARTE 6

### **1. Definire la cartella clinica nel caso generale**

- a. La cartella clinica è il documento o l'**insieme dei documenti che raccolgono le informazioni di tipo medico ed infermieristico** necessarie a rilevare il percorso diagnostico-terapeutico di un paziente, al fine di determinare le cure da somministrare. Rappresenta quindi il **nucleo fondamentale dei dati sanitari di un paziente**.

## 2. Illustrare almeno tre funzioni della cartella clinica

a. Esse sono:

- i. **Descrizione cronologica del processo di cura:** narra lo stato del paziente dal punto di vista di chi lo cura.
- ii. **Mezzo di comunicazione asincrono** tra il personale sanitario.
- iii. **Facilita la cura** del paziente perché offre una visione a 360 gradi del suo stato.
- iv. È la **base per il calcolo del DRG** quindi è legato al rimborso, perché la cartella clinica contiene anche una sezione legata alla dimissione, dove nella scheda di dimissione ospedaliera sono contenuti i dati per il calcolo del DRG.
- v. È un **contenitore permanente di dati per fare ricerca statistica**, ed è anche una raccolta ufficiale che vale legalmente.

## 3. Elencare le 7 sezioni principali della cartella clinica, e commentarne in dettaglio almeno due

a. Esse sono:

- i. **Anagrafica:** vi sono i **dati personali** del paziente per consentire eventuali contatti sia durante il ricovero (es. con i familiari) che successivamente alla dimissione.
- ii. **Accettazione:** vi sono riportati il **codice univoco** di identificazione assegnato al paziente, il **reparto** in cui sarà degente e il **motivo** del ricovero.
- iii. **Anamnesi**, si articola di 3 attributi:
  1. **di chi è** (personale o familiare):
    - a. Personale: riguarda il particolare paziente, quindi si possono riportare informazioni come le abitudini, lo stile di vita....
    - b. Familiare: per il paziente si va a registra dell'informazione legata ai familiari, per individuare fattori di rischio legati ad una certa ereditarietà.
  2. **di che tipo** (fisiologica o patologica):
    - a. Patologica: ci dice che tipo di patologie ha avuto il paziente.
    - b. Fisiologica: riguarda gli episodi legati alla crescita del paziente (esempio tipo di parti, vaccini, età pubertà, ecc.).
  3. **a quale evento è riferita** (prossima o remota):

- a. Prossima: : se riferisce di fatti legati all'attuale ricovero, quindi sintomi manifestatisi di recente.
  - b. Remota: se lontana al ricovero attuale, sia logicamente che temporalmente.
- iv. **Esame obiettivo:** contiene le **informazioni che si possono acquisire visitando il paziente** e usando strumenti ambulatoriali di base. Ci sono due tipi di esame obiettivo:
  - 1. **esame obiettivo generale:** insieme delle informazioni che vengono comunque sempre raccolte, indipendentemente dal motivo specifico del ricovero (tipicamente: peso, altezza, temperatura, pressione, frequenza cardiaca, ...);
  - 2. **esame obiettivo specifico:** insieme delle informazioni che riguardano più specificatamente il motivo del ricovero.
- v. **Diario Clinico Giornaliero:** vi si registra, durante tutto il ricovero, **tutto ciò che riguarda il paziente**, dai dati periodici (es. temperatura), alla terapia. Da un punto di vista di **medico-legale**, soprattutto per la somministrazione di farmaci o comunque gli interventi sul paziente, diventa importante disporre di informazioni come ora, dosaggio, via di somministrazione ecc. Dal punto di vista **informatico**, possiamo fare cose raffinate, in termini di integrità della cartella, per assicurarci che non ci siano alterazioni di informazioni. È composto da:
  - 1. **Diario medico.**
  - 2. **Diario infermieristico:** ingombrante, perché l'infermier\* implementa le cure.
- vi. **Indagini/Accertamenti:** vi si registrano, durante tutto il ricovero, i **risultati** delle visite specialistiche e gli esami.
- vii. **Dimissione:** vi si registra la conclusione del ricovero, e quindi:
  - 1. **SDO (Scheda di dimissione Ospedaliera).**
  - 2. **Lettera di dimissione:** informazioni riassuntive sul processo di cura, di solito indirizzata al medico di famiglia.

#### **4. Illustrare i diversi tipi di anamnesi presenti in cartella clinica**

- a. Spiegato sopra.

#### **5. Illustrare almeno 3 pro e 3 contro legati alla realtà fisica della cartella clinica cartacea**

- a. I pro sono i seguenti:
  - i. È trasportabile.

- ii. Non richiede **tecnologia**.
  - iii. È **immediata e facile da compilare e usare**, non richiede particolare addestramento del personale.
  - iv. L'**accesso ai dati** è molto diretto (se i dati sono pochi).
  - v. Facile da **duplicare** (fotocopia).
- b. I contro sono i seguenti:
- i. Un **solo utente alla volta** può accedere alla cartella e soltanto in loco (difficile il consulto a distanza).
  - ii. La ricerca di una cartella fra molte in uno schedario può essere **lunga e difficile**.
  - iii. Occupa fisicamente molto spazio (magazzini) e può essere **ingombrante/pesante** (es. pazienti con più problemi clinici).
  - iv. Anche all'interno di un ospedale, viaggia con **lentezza e difficoltà**.
  - v. La carta si **usura** facilmente ed è relativamente fragile.
  - vi. La duplicazione richiede **tempo e denaro**.
  - vii. **Non c'è privacy** nei dati.
  - viii. È facile **perdere "pezzi"**.
  - ix. Produrne e trasmetterne copia in formato cartaceo **costa** parecchio.

**6. Illustrare almeno 1 pro e 3 contro legati alla strutturazione dei dati nella cartella clinica cartacea**

- a. I pro:
- i. La carta è un **supporto informale**, con conseguente grande libertà e facilità nell'inserimento.
- b. I contro:
- i. Lo stile di **compilazione** è **"personale"**.
  - ii. Senza una struttura formale dei dati si fanno **più errori**.
  - iii. La **ricerca** all'interno della cartella è **difficoltosa**.
  - iv. La **ricerca manuale** tra le cartelle per cercare quelle che soddisfano certi criteri è:
    - 1. **Lunga**.

2. **Difficoltosa.**
3. **Costosa.**
4. **Poco efficace.**
5. **Non effettuabile** (troppo complessa).

**7. Definire la cartella clinica elettronica e illustrare quello che può fare in più (non quello che fa meglio!) rispetto a quella cartacea**

- a. Le cose sono le seguenti:
  - i. Registrare **tutte le informazioni** inerenti al processo clinico.
  - ii. Archiviare le informazioni in modo **sicuro in un unico luogo**.
  - iii. **Facilitare il reperimento** di informazioni sia orizzontali (stesso paziente) che verticali (più pazienti).

**8. Illustrare almeno 3 pro e 3 contro della cartella clinica elettronica**

- a. I pro sono i seguenti:
  - i. **Basso ingombro** fisico dei dati.
  - ii. **Facile duplicabilità** dei dati.
  - iii. **Trasportabilità virtuale** (LAN/modem e dispositivi wireless).
  - iv. **Accesso contemporaneo** di più utenti (stesso ospedale, ma anche teleconsulto).
  - v. **Protezione dell'accesso** da utenti non autorizzati.
  - vi. **Ricerca di dati:**
    1. **Facile.**
    2. **Veloce.**
    3. **Possibile fare ricerche complesse.**
  - vii. **Visione integrata dei dati dei pazienti.**
  - viii. **Supporto alla decisione clinica.**
  - ix. **Supporto inserimento dati.**
  - x. **Accesso a fonti di conoscenza.**



xi. **Supporto integrato alla comunicazione.**

b. I contro sono i seguenti:

- i. **Elevati requisiti tecnologici.**
- ii. **Sicurezza:** una volta superate le barriere, i dati possono essere più facilmente trafugabili di quelli cartacei.
- iii. **Richiesta di formalizzazioni e strutturazione dei dati spesso poco naturali.**
- iv. **Richiede addestramento del personale.**

**9. Illustrare gli aspetti fondamentali che vanno considerati nella progettazione di una cartella clinica elettronica**

a. Gli aspetti da considerare sono i seguenti:

i. **Ambito di applicazione:**

1. ambulatoriale;
2. reparto ospedaliero;
3. terapia intensiva.

ii. **Definizione del modello formale della cartella:**

1. orientata al **problema**: classificazione per problemi;
2. orientata **temporalmente**: collezione di dati sequenziali;
3. orientata alla **sorgente informativa**: classificazione per sorgente d'informazione.

iii. **Livello di attività della cartella:**

1. **passiva**: mero deposito di informazioni;
2. **attiva**: ausilio attivo alla diagnosi (generazione di allarmi, suggerimenti diagnostici o terapeutici, incorporazione d'accesso a protocolli diagnostici).

iv. **Aspetti informatici:**

1. **tipi di dati** e loro ingombro;
2. **accessibilità e sicurezza** dei dati;
3. **esportabilità** dei dati da una struttura ad un'altra, in presenza di cartelle diverse.

**10. Illustrare le differenze fra modelli dei dati della cartella clinica orientati temporalmente, orientati alla sorgente, e orientati al problema**

- a. Spiegato sopra.

**11. Illustrare la convenienza in diverse situazioni cliniche dei modelli dei dati orientati temporalmente, orientati alla sorgente, e orientati al problema.**

- a. Modello orientato **temporalmente** - Struttura adatta a seguire una sola malattia per volta. Adatta per:
  - i. Gestione di pazienti **ambulatoriali**;
  - ii. Gestione pazienti affetti da **malattie croniche** e soggetti a controlli periodici ripetitivi.
- b. Modello orientato alla **sorgente** - adatta in:
  - i. Pazienti ospedalizzati in **reparti** in cui si possono seguire, in molti casi, dei protocolli.
- c. Modello orientato al **problema** - adatta in:
  - i. **Medicina di base** (dove lo stesso paziente è seguito da un medico per anni e anni, spesso per problemi diversi).
  - ii. **Terapia intensiva**, dove di solito il paziente ha diversi problemi concomitanti.

**12. Illustrare cosa si intende con cartella attiva**

- a. Si intende una cartella con le seguenti caratteristiche:
  - i. Offre al clinico una **visione integrata dei dati** dei pazienti (es. dati che provengono da più sistemi possono essere analizzati insieme più facilmente, ad es. graficamente).
  - ii. Offre **supporto alla decisione clinica**.
  - iii. Funge da **supporto per l'inserimento dei dati**.
  - iv. Consente l'**accesso a fonti di conoscenza**, cioè la cartella per esempio suggerisce un protocollo da seguire per una malattia magari particolarmente critica per cui vale la pena attenersi ad un protocollo.
  - v. Offre **supporto integrato alla comunicazione**, per esempio il medico di medicina generale può interagire con i CPU per prenotare prestazioni per i propri assistiti, o può cominciare con i colleghi.

**13. Definire il significato di un protocollo e illustrarne in dettaglio almeno due vantaggi in clinica**

- a. Un protocollo è una **serie di istruzioni**, derivante da studi scientifici lunghi e internazionalmente riconosciuti, rappresentanti il **modo ritenuto “migliore”** per eseguire un determinato compito. Garantisce i seguenti vantaggi:
  - i. Garantiscono l'esecuzione delle istruzioni in modo **uniforme e riproducibile**.
  - ii. Forniscono un **supporto alla memoria** e propensione al rischio dell'operatore sanitario.
  - iii. Garantiscono uno **standard minimo** nell'erogazione delle cure ed è anche per i giovani medici un supporto educativo per l'apprendimento del modo corretto di svolgere un compito.
  - iv. Consentono di **sfruttare l'esperienza** di chi li ha messo a punto anche in situazioni complesse o clinicamente poco frequenti.
  - v. Offrono una chiara **demarcazione delle responsabilità e dei ruoli**.

**14. Illustrare due funzioni attive di una cartella clinica elettronica per il medico di famiglia**

- a. I seguenti sono alcuni esempi:
  - i. Gestire la **medicina di gruppo** (quando ad esempio più medici condividono l'ambulatorio).
  - ii. Gestione **protocolli**.

**15. Illustrare due delle possibili funzioni che una cartella clinica elettronica per il medico di famiglia offre in più rispetto ad una cartella clinica cartacea**

- a. Esse sono le seguenti:
  - i. Effettuare **calcoli statistici** sui pazienti.
  - ii. Consentire tramite una buona interoperabilità l'**interfacciamento con il CPU di diversi SIS** per prenotazioni di prestazioni o ricezioni di referti.

## PARTE 7

### 1. Illustrare cosa si intende per **privacy dei dati**

- a. **Privacy dei dati**: richiede che un'entità terza che **intercetta** i dati **non possa** **poterne comprendere** il significato.

### 2. Illustrare cosa si intende per **integrità dei dati**

- a. **Integrità dei dati**: richiede che un'entità terza **non possa poter alterare** l'**informazione** trasmessa dal mittente senza che il ricevente se ne accorga.

### 3. Illustrare cosa si intende per **identità dei partner**

- a. **Identità dei partner**: un'entità terza non deve potersi camuffare da mittente o far credere al mittente che il ricevente gli ha inviato dei dati. In altre parole, **deve essere sempre chiaro con chi si sta interloquendo**.

### 4. Illustrare cosa si intende per **non ripudiabilità degli impegni**

- a. **Non ripudiabilità degli impegni**: il mittente non può negare al ricevente di essere stato lui a mandare i dati, e viceversa il ricevente non può negare che i dati sono stati mandati dal mittente.

### 5. Perché un sistema di crittografia a chiave segreta è detto anche a chiave simmetrica?

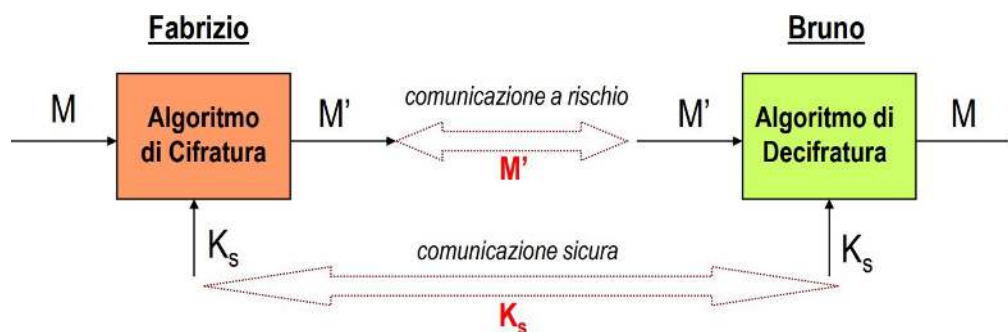
- a. Perché la chiave utilizzata per la cifratura è la stessa che viene utilizzata per la decifratura. L'algoritmo è dunque detto algoritmo simmetrico, cioè **la chiave usata per decifrare è la stessa che è stata usata per cifrare**.

### 6. Illustrare cosa si intende per **attacco esaustivo per un codice a chiave simmetrica**

- a. Un **attacco esaustivo** è una tecnica per tentare di violare un messaggio in cui si prova a decifrare un codice **provando tutte le chiavi possibili**. La sua efficacia dipende dal numero di chiavi possibili: più esso è alto, più tempo è richiesto dall'attacco per poter avere successo.

### 7. Su quali cose si basa la sicurezza di un sistema a chiave simmetrica?

- a. Esso prevede l'utilizzo di un'**unica chiave**, usata dal mittente per criptare il messaggio e dal ricevente per decriptarlo. Esso si basa sui seguenti principi:
  - i. **segretezza della chiave**;
  - ii. **pratica impossibilità di ricavare la chiave dal messaggio criptato**, pur conoscendo l'algoritmo utilizzato;
  - iii. (per renderlo sicuro) la chiave ha complessità tale da rendere comunque **irrealizzabile l'attacco esaustivo**.

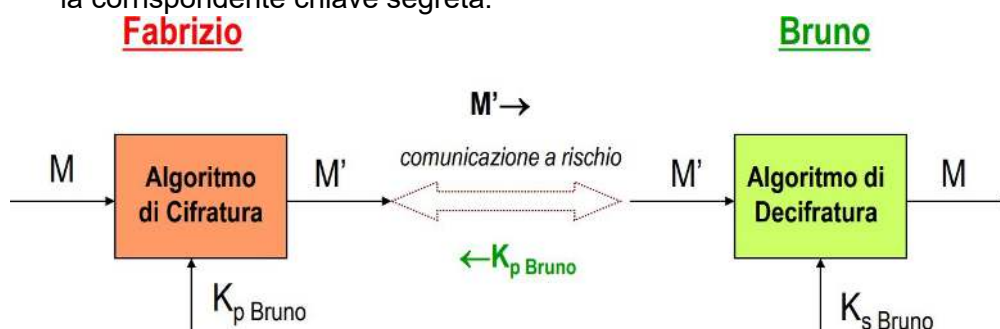


## 8. Quali sono i due limiti principali dei sistemi a chiave simmetrica?

- a. I limiti sono i seguenti:
  - i. è necessaria una chiave diversa per ogni interlocutore, portando quindi ad una **proliferazione delle chiavi**;
  - ii. è **difficile effettuare lo scambio delle chiavi** per la comunicazione, in quanto esse non possono viaggiare in chiaro.

## 9. Qual è l'idea di fondo dei sistemi di crittografia a chiave asimmetrica?

- a. Si chiamano così perché ogni utente ha **due chiavi**: una **chiave segreta** (detta anche privata), che nessuno può conoscere al di fuori del proprietario, **ed una chiave pubblica** (ogni chiave segreta è in un rapporto di corrispondenza alla chiave pubblica), così chiamata perché è invece conoscibile a tutti (e quindi "anche in chiaro"). Queste due chiavi sono tra di loro in un **rapporto speciale**, che si definisce di **alter ego**. Quello che viene codificato con un algoritmo asimmetrico sfruttando una delle due chiavi, è decodificabile solo e soltanto con l'altra chiave: **la chiave che riapre non è la stessa che ha chiuso**. Siccome una delle due chiavi deve rimanere segreta, bisogna anche che sia computazionalmente irrealizzabile quel procedimento di attacco esaustivo per cui si cerca di recuperare dalla chiave pubblica che è nota, la corrispondente chiave segreta.



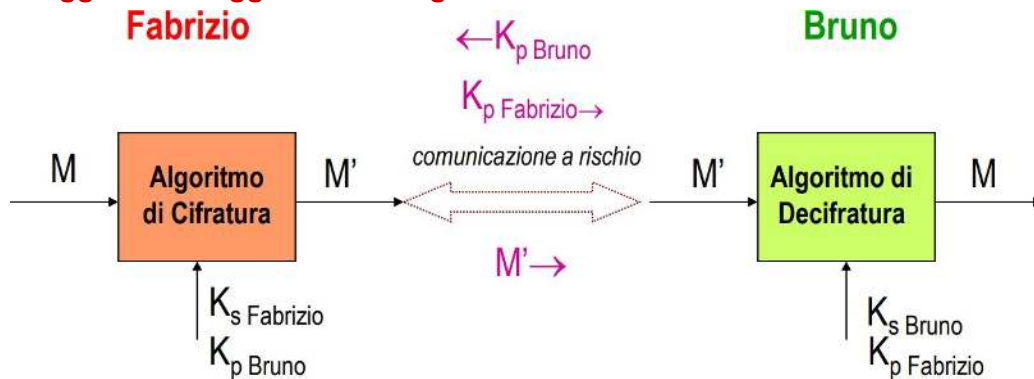
## 10. Perché non costituisce alcun problema la diffusione della chiave pubblica nella crittografia a chiave asimmetrica?

- a. Perché la forza dei metodi di generazione di coppie di chiavi sta nel fatto che sono basati su **funzioni "facili" da calcolare** (aritmeticamente parlando) in avanti, **ma estremamente "difficili" da invertire**. Di conseguenza anche avendo la chiave pubblica e conoscendo l'algoritmo il tempo e le risorse computazionali richieste per arrivare alla chiave privata rendono l'operazione sostanzialmente impossibile.

### 11. Quali sono i metodi che si possono usare per garantire l'identità dei partner in una trasmissione dati?

- a. Per garantire queste cose, si usa la variante della crittografia asimmetrica usando due chiavi. Per la codifica si procede così: gli interlocutori si scambiano la propria chiave pubblica, Fabrizio cifra il messaggio che deve spedire usando la sua chiave segreta e ciò serve a firmare il messaggio. Dopodiché sulla striscia di bit che è venuta fuori, che è già un messaggio criptato, fa girare da capo l'algoritmo simmetrico usando la chiave pubblica di Bruno. Il messaggio a questo punto parte, Bruno lo riceve e deve decriptarlo: dovrà andare al contrario, cioè dovrà prima fare la sua chiave segreta (cioè la chiave alter ego dell'ultima chiave usata per codificare), poi dovrà renderlo comprensibile decriptandolo con una passata dell'algoritmo asimmetrico, usando la chiave pubblica di Fabrizio. Lo schema è:
  - i. Sul canale di comunicazione a rischio prima viaggiano le due chiavi pubbliche: Fabrizio cifra con la sua chiave segreta (e questa è la firma che garantirà l'identità del mittente e la non ripudiabilità);
  - ii. dopodiché usa la chiave pubblica di Bruno, e manda  $M'$ . Una volta che Bruno riceve  $M'$ , prima passa con la chiave segreta, e poi con la chiave pubblica di Fabrizio, e così il messaggio è fruibile.
- b. Ecco che abbiamo garantito anche le due clausole di non ripudiabilità degli impegni e identità del mittente.

### 12. Vantaggi e svantaggi della crittografia asimmetrica



### 13. Vantaggi derivanti dalla "chiusura" per due volte con due chiavi diverse asimmetriche di un messaggio

- a. I vantaggi sono i seguenti:
  - i. è garantita l'**identità del mittente**;
  - ii. è garantita la **non ripudiabilità del messaggio** anche su canali a rischio.

#### 14. Illustrare l'idea della crittografia mista

- a. I sistemi di crittografia misti **combinano le tecniche chiave simmetrica e asimmetrica** in modo da fonderne i vantaggi. In parole povere, si utilizza il sistema (lento) **asimmetrico solo per comunicare la chiave segreta** (che in questi casi viene chiamata "chiave di sessione"). La chiave di sessione verrà poi usata per una **normale comunicazione basata su cifrari a chiave simmetrica**. In questo modo si risolve il problema della sicurezza nello scambio della chiave segreta, mentre la velocità di cifratura/decifratura rimane molto alta e non penalizza la comunicazione.

#### 15. Illustrare cos'è la firma digitale

- a. La firma digitale è un particolare tipo di firma elettronica basata su un sistema di **chiavi asimmetriche a coppia** che consente al titolare e al destinatario, rispettivamente, di **rendere manifesta e di verificare la paternità e l'integrità** di un documento informatico. Per essere valida ai sensi di legge, deve essere **certificata**: è equivalente a quella tradizionale. La cosa interessante è che **non è ripudiabile**, perciò è sempre riconducibile al titolare.

#### 16. . Illustrare perché si usano funzioni di hash per firmare digitalmente un documento

- a. Esse si usano perché consentono di **ridurre l'elevato costo computazionale** necessario per applicare la firma digitale, pur mantenendo le caratteristiche di quest'ultima.

#### 17. Definire una funzione di hash e le sue proprietà fondamentali

- a. Una funzione di hash trasforma un testo normale di lunghezza qualsiasi in una stringa di lunghezza fissata, chiamata digest. Ha le seguenti proprietà:
  - i. È **non invertibile** e **resistente alle contro-immagini**.
  - ii. È **resistente alle collisioni** (la probabilità che due messaggi diversi abbiano lo stesso digest è molto piccola).
  - iii. È **resistente alla correlazione** (di norma piccole variazioni del messaggio originale provocano evidenti variazioni nel digest).

#### 18. Come si invia in rete in modo sicuro un documento firmato? Come può il destinatario verificare l'integrità del documento ricevuto?

- a. Per spedire un documento assicurandone paternità ed integrità (non ci curiamo per semplicità della privacy), si deve:
  - i. **creare un'impronta** del documento con una funzione di hash stato dell'arte;
  - ii. **criptare l'impronta** (non il documento) usando la propria chiave privata (si fa in fretta perché l'impronta è "corta");
  - iii. **spedire su canale (anche insicuro) impronta criptata e documento**.
- b. Chi deve verificare l'autenticità (paternità e integrità) del documento:

- i. **decripta l'impronta** usando la chiave pubblica del proprietario;
- ii. applica al documento la **stessa funzione di hash** usata dal proprietario;
- iii. **confronta l'impronta** appena ottenuta con quella ricevuta;
- iv. se le impronte coincidono, il messaggio è arrivato integro, non è stato manomesso durante il tragitto.

## 19. Illustrare come si risolve il problema della paternità di una chiave pubblica

- a. È risolta tramite la **certificazione delle chiavi**. I vari stati UE hanno ciascuno un ente certificatore (per l'Italia l'AGID (Agenzia per l'Italia Digitale)), che rilascia ad un gruppo limitato di cittadini delle chiavi certificate. Quindi i certificatori:
  - i. **forniscono a chi ne fa richiesta una coppia di chiavi asimmetriche** e la certificazione delle stesse;
  - ii. **garantiscono l'identità dei soggetti** a cui hanno fornito tali chiavi;
  - iii. **mantengono i registri** delle chiavi pubbliche necessarie a verificare la titolarità del firmatario.
- b. Ovviamente il titolare di chiavi certificate ha la responsabilità di **conservare in modo sicuro la chiave privata** (e, ovviamente, di non farla usare ad altri): il certificato scade circa ogni 3 anni in modo da minimizzare anche le violazioni. Le firme digitali in pratica sono gestite con dispositivi tipo smart card, le quali sono dotate di un chip che è in grado di far funzionare l'algoritmo di firma direttamente sul chip della smart card, perché sulla smart card c'è la chiave privata.

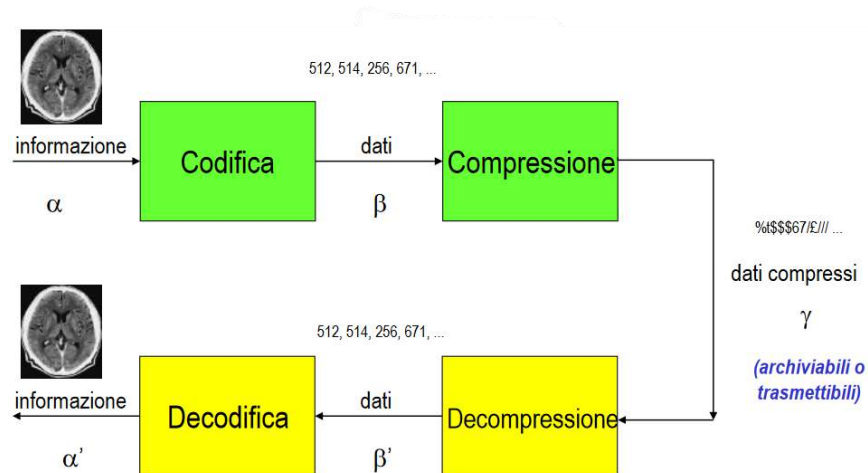


## PARTE 9-10

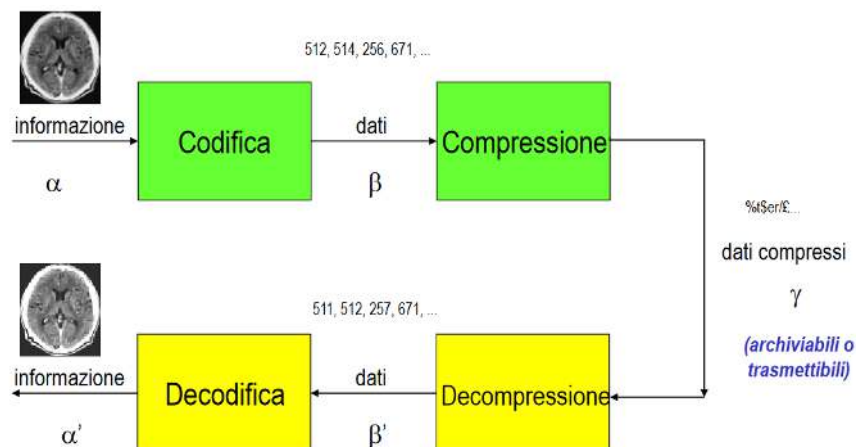
### 1. Illustrare la differenza fra tecniche di compressione lossless e lossy

a. Le tecniche di compressione possono essere suddivise in:

- i. **Lossless/reversibili: non c'è perdita d'informazione**, e di conseguenza con la decompressione consentono di ricostruire integralmente il messaggio originale. Si può basare sull'esistenza nell'informazione di una certa ridondanza e di una serie di pattern che consentono di predire parti di un messaggio non esplicitamente espresse. La tecnica consente solo una limitata compressione (fino a un massimo di un rapporto 1 a 10).



- ii. **Lossy/irreversibili: prevedono una moderata perdita di informazione**, e conseguentemente la decompressione non consente una ricostruzione esatta del messaggio originale. Si basa sul fatto che per alcuni tipi di stimoli (e.g. suoni e immagini) si può far leva non solo sulla ridondanza, ma anche sulla irrilevanza percettiva di certi dettagli. La compressione così operata comporta una perdita di informazione che però è sostanzialmente impercettibile. Si raggiungono così rapporti di compressione molto maggiori.



## 2. Illustrare, facendo riferimento al sistema uditivo, perché si può far leva sull'irrelevanza percettiva per comprimere un messaggio sonoro

- a. Dalla Serie di Fourier troncata non si potrà più ricostruire la forma d'onda di partenza, ma le differenze saranno impercettibili e comunque adeguate allo scopo. Infatti, si possono sfruttare due fattori:
  - i. La banda delle frequenze sonore della voce umana è molto più ristretta di quella delle frequenze udibili dall'orecchio umano. Questo significa che, se il messaggio da trasmettere è composto principalmente da voci, si può ridurre lo spettro delle frequenze conservate.
  - ii. Per molti scopi (come la registrazione di segreterie telefoniche o messaggi vocali) non è necessaria una ricostruzione particolarmente precisa del messaggio sonoro originale, ed è sufficiente mantenere una qualità tale che il messaggio resti comprensibile.

## 3. Spiegare perché nella compressione di segnali e immagini è utile un passo preliminare di decorrelazione

- a. La decorrelazione è un processo perfettamente reversibile e di norma il primo tacito passo di tutte le procedure di compressione di segnali e immagini, sia reversibili che non. Si definisce una funzione  $e(m, n) = f(m, n) - g(m, n)$ , dove  $g(m, n)$  è un **predittore dell'immagine** e  $f(m, n)$  è il **segnale originale**. L'immagine  $e(m, n)$  è il residuo (cioè quello che dell'immagine non so prevedere) ed è ciò che vado effettivamente a comprimere. La decorrelazione viene usata per rendere più efficace la successiva fase di compressione vera e propria, dato che tipicamente l'istogramma dei valori di  $e(m, n)$  è più "stretto" di quello dei valori di  $f(m, n)$ , ovvero **la distribuzione dei valori di  $e(m, n)$  sarà più concentrata** di quella di  $f(m, n)$ . La decorrelazione **riduce l'entropia** del messaggio sorgente.

## 4. Illustrare, usando anche un esempio, la codifica run length

- a. I dati possono presentare **sequenze che contengono più volte lo stesso simbolo** (i cosiddetti "run"). In quei casi non è conveniente ripetere lo stesso simbolo più volte, bensì il run può essere memorizzato, ad esempio, conservando **il valore del primo dato seguito da un simbolo speciale e dalla lunghezza del run** (length).
  - i. E.g. striscia di caratteri '\$\$\*\*\*\*\*55.62' e viene codificata in '\$\$^955.62', che è di 10 caratteri a differenza di quella di prima che ne ha 17. Ecco che abbiamo un run lungo 2 di dollari, un run lungo 10 di asterischi, un run lungo 2 di 5. Vedremo più avanti perché i run di lunghezza 2 non conviene codificarli, ma pensiamo allora a quello di lunghezza 10. La cosa che si può fare è usare per esempio il carattere speciale '^': lo si mette dopo il carattere che si ripete, e subito dopo si scrive quante volte deve essere ripetuto, quindi in questo caso l'asterisco, che viene ripetuto 9. Come possiamo osservare, la codifica di un singolo run si fa se il run ha una lunghezza minima di 3 carattere in questo caso; altra osservazione è che bisogna fissare anche una lunghezza massima del run, in questo caso è 10, di modo che la lunghezza del run si possa esprimere con una sola cifra in questo caso appunto abbiamo visto che la esprimiamo con il 9. Quindi in fase di progettazione di questo sistema di

compressione, bisogna stabilire anche, oltre alla lunghezza minima del run per cui vale la pena modificarlo, anche la lunghezza massima

**5. Illustrare come è possibile in alcuni casi usare a proprio vantaggio la rappresentazione al calcolatore dei numeri interi per creare dei run**

- a. Ripartiamo i **12 bit “utili” in due parti, 5 bit nel MSB** (Most Significant Byte) e **7 bit nel LSB** (Least Significant Byte). Il prefisso di ciascun byte identifica il tipo di informazione che il byte trasporta (111=MSB, 0=LSB) Nell'immagine TAC da comprimere, per via della correlazione fra i livelli di grigio, **i campioni dell'immagine differiscono spesso solo nel LSB**, per cui si potrà spesso evitare di conservare il MSB, con una riduzione di ingombro il decompressore troverà poi più byte LSB che byte MSB, ma attribuirà **ai byte senza MSB l'ultimo MSB trovato**.

**6. Definire la quantità di informazione media di una sorgente di simboli, precisando l'unità di misura**

- a. La quantità d'informazione media di una sorgente di simboli è data dalla sua **entropia**. Essa viene data dalla formula  $H(X) = \sum p_k(s_k) = \sum p_k \cdot \log_2 \left( \frac{1}{p_k} \right)$  dove  $p_k$  è la probabilità che un simbolo sia il simbolo  $s_k$ . Essa si misura in bit/simbolo. Si può anche dire che l'entropia misura **l'uniformità della distribuzione dei simboli** generati dalla sorgente.

**7. Dare i bound inferiore e superiore dell'entropia di una sorgente di simboli e illustrare sotto quali condizioni vengono raggiunti**

- a. L'entropia è **massima** se **ogni simbolo è equiprobabile**, ovvero se, dati M simboli,  $p(s_k) = \frac{1}{M} \quad \forall s_k$ . Nel qual caso l'entropia massima sarà  $H(X) = \sum \frac{1}{M} \cdot \log_2(M) = \log_2(M)$
- b. L'entropia di una sorgente X a M simboli è **minima**, e pari a zero, se un simbolo ha **probabilità unitaria**.

**8. Dare la definizione di codice**

- a. Un **codice** è una **funzione  $C(\cdot)$  che associa, a ciascuno dei simboli dell'alfabeto  $A = \{s_1, s_2, s_3, s_4, \dots, s_M\}$ , una stringa binaria** (detta parola). Saranno di interesse, ovviamente, solo **codici non singolari**, ovvero codici per cui, se  $s_p \neq s_q$ , allora  $C(s_p) \neq C(s_q)$ .

**9. Illustrare almeno due proprietà desiderabili per un codice a lunghezza variabile**

- a. Esse sono le seguenti:
  - i. **decodificabilità univoca**: ogni possibile concatenazione di codici corrisponde ad una e una sola sequenza di simboli;
  - ii. **decodificabilità istantanea**: dal flusso di dati codificati si può sempre stabilire quando si è completamente ricevuto un simbolo, senza dover aspettare di ricevere il seguito. Condizione necessaria e sufficiente affinché un codice sia istantaneamente decodificabile: nessun codice è il prefisso di un altro;

- iii. **efficienza:** codici più corti corrispondono a simboli più probabili, ovvero se  $p_1 \geq p_2 \geq \dots \geq p_M$  implica  $I_1 \leq I_2 \leq \dots \leq I_M$ .

#### 10. Illustrare l'idea alla base della codifica entropica

- a. L'idea delle tecniche di codifica entropica è quella di comprimere la sequenza di simboli utilizzando, per il singolo simbolo, parole binarie di lunghezza legata alla probabilità del simbolo. **I simboli più probabili verranno codificati con pochi bit, i simboli meno probabili con più bit.**

#### 11. Qual è il lower bound della lunghezza media per simbolo a cui può portare la tecnica di Huffman?

- a. L'**entropia della sorgente** rappresenta comunque il **lower bound della lunghezza media** (statistica) dei simboli codificati.

#### 12. Descrivere l'idea della tecnica di compressione LZW

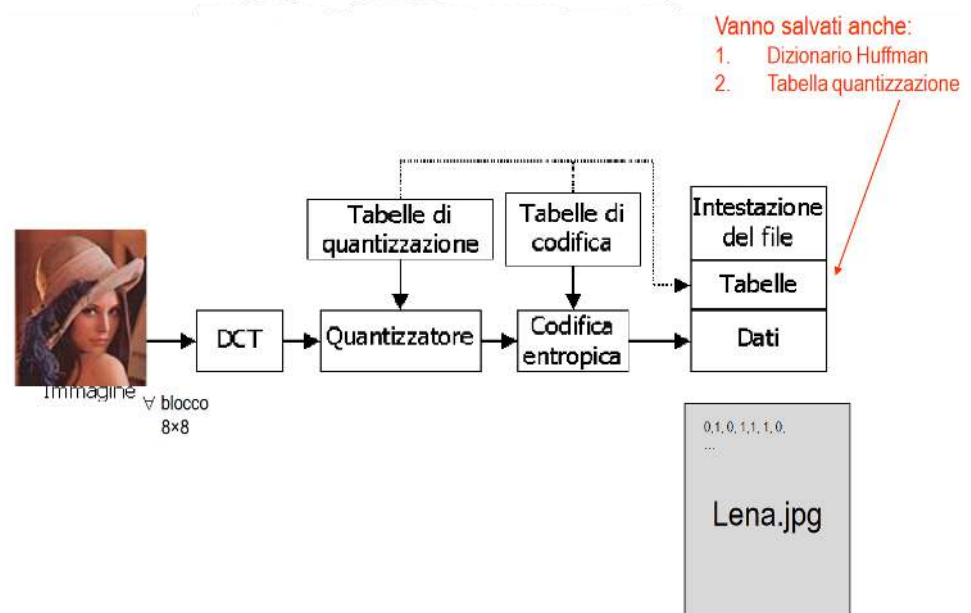
- a. Il metodo di compressione LZW costruisce, **per ogni testo da comprimere, uno specifico dizionario ad hoc** in cui sequenze di (due, tre, quattro, ...) **caratteri ASCII hanno un codice compatto**. I primi 256 posti del dizionario sono sempre e comunque occupati dalle codifiche ASCII (a 8 bit) dei singoli caratteri, e ad essi seguono numeri indicanti stringhe di codice ripetute nel testo. **Ogni messaggio genera un dizionario di codici diverso**. Per motivi di efficienza computazionale il dizionario dei codici deve comunque essere limitato. Quando la dimensione massima M del dizionario viene raggiunta, si può bloccare la crescita del dizionario o resettarlo.

#### 13. Descrivere in breve i due passi della compressione DCT delle immagini

- a. La Discrete Cosine Transform (DCT) è una **trasformata concettualmente simile alla trasformata discreta di Fourier** (usando tuttavia valori reali invece che complessi) e ci dice ancora come l'energia di un segnale nel tempo si può ripartire tra componenti (stavolta coseni) a varie frequenze. La compressione si svolge coi seguenti passi:
- Si divide la matrice immagine (se è a colori si dividono le tre matrici R, G e B oppure le Y, U, e V ottenute da R, G, e B) in **blocchi da 8x8 pixel**, aggiungendo eventualmente righe o colonne nulle per il completamento.
  - Per ogni blocco 8x8, di cui indichiamo i valori con  $f(m,n)$   $m,n=0\dots7$ , **si calcola la DCT-2D**, che produce ancora un blocco di 8x8 coefficienti (reali).
  - I coefficienti della DCT indicano com'è ripartita alle varie frequenze l'energia del segnale immagine (similmente alla DFT). In ciascun blocco 8 X 8, in alto a sinistra ci sono le basse frequenze spaziali e in basso a destra le alte frequenze spaziali. Per ogni blocco vengono **scelti gli L coefficienti a maggior energia** (con  $L \leq M$ ), e gli altri vengono considerati pari a 0.

#### 14. Descrivere in breve i passi principali della compressione JPEG delle immagini

- a. Lo standard JPEG (Joint Photographic Experts Group) è stato definito negli anni 1986–1992, da parte di gruppi di lavoro predisposti da due organizzazioni per utilizzare le più avanzate tecniche allora disponibili, consentendo all'utente di variare a suo piacere il rapporto di compressione, avendo comunque un algoritmo indipendente da contenuto, dimensione, e risoluzione dell'immagine e con complessità computazionale. I passi sono i seguenti:
- Divisione in blocchi 8x8.**
  - DCT per ogni blocco.**
  - Quantizzazione dei coefficienti della DCT** (passo irreversibile): Ogni elemento di ogni blocco 8x8 di coefficienti DCT viene diviso per il corrispondente coefficiente di una tabella di quantizzazione data basata su studi sulla percezione umana, e poi la parte intera del valore di tale divisione viene moltiplicata nuovamente per il corrispondente valore in tabella. L'effetto dell'operazione di divisione e del successivo arrotondamento all'intero più vicino è quello di schiacciare verso il basso i valori della DCT, portando a zero quelli che erano già piccoli.
  - Codifica finale dei coefficienti quantizzati (separata per componente DC e componenti AC):** In ogni blocco di coefficienti DCT l'elemento (0,0) ci dà, come nella DFT, il **valor medio** di luminosità, chiamato "**componente continua/DC**", il quale verrà poi compresso con codifica Huffman. I **valori successivi ("componenti AC")** vengono codificati usando una compressione run length operata "a zig-zag".



**15. Descrivere come si fa in JPEG ad ottenere rapporti di compressione via via più spinti**

- a. Si fa moltiplicando i valori delle tabelle di quantizzazione per un fattore di scala.

**16. Spiegare perché in JPEG conviene trattare i coefficienti DC dei vari blocchi in modo separato dai coefficienti AC**

- a. È ragionevole supporre che **blocchi di pixel contigui abbiano un livello medio di luminosità simile**, dunque risulta conveniente utilizzare una codifica ad hoc per i coefficienti DC dei vari blocchi dell'immagine. In particolare, si usa una **codifica differenziale di tipo Huffman**, dove si codifica la sequenza  $Diff_k = DC_k - DC_{k-1}$ , la quale ha entropia minore.

**17. Spiegare perché in JPEG è opportuno costruire la sequenza dei coefficienti AC di ogni blocco precedendo non per riga ma a zig-zag**

- a. Questo metodo consente di norma di **avere all'inizio della sequenza dei dati informativi (ed in coda una lunga sequenza di zeri)**.

**18. Illustrare come viene effettuata la codifica finale dei coefficienti AC dei vari blocchi nello standard JPEG**

- a. I 63 coefficienti AC vengono codificati con tecniche tipo run length e Huffman. In particolare, si procede a zig-zag, in modo da avere a inizio frequenza dati informativi e poi run di zeri. Si applica la Zero Run Length, una codifica specifica per run di zeri. Essa funziona così:
  - i. Si dedica un numero N di bit (di solito **6 bit**) alla rappresentazione di ciascun run.
  - ii. I **primi due bit (00)** identificano il **run**.
  - iii. I successivi **quattro bit indicano il numero di zeri** consecutivi meno 1. Pertanto, si arrivano a gestire fino ad un massimo di  $2^{(N-2)-1}$  zeri. Se ci sono più di 16 zeri, si codifica più volte il run.

**19. Spiegare in breve come si procede alla compressione di video nello standard MPEG**

- a. Essa funziona nel seguente modo:
  - i. Si dividono le immagini in **Group Of Pictures** (tipicamente 1 GOP=12 frame). In ordine di ingombro finale i vari frame sono:
    1. **frame intra (I)**: codificati come immagini statiche usando DCT;
    2. **frame predetti (P)**: codificati tramite il modello della correlazione fra frame (poco ingombro) che consente di riottenersi a partire dai frame I dello stesso GOP;
    3. **frame interpolati (B)**: codificati tramite il modello di interpolazione (pochissimo ingombro) a partire dai due frame adiacenti (I o P che siano).

# TELEMEDICINA

## 1. Dare una definizione di telemedicina

- a. La TELEMEDICINA (medicina a distanza) è l'**uso di informazioni mediche, scambiate da un luogo all'altro per via telematica, utili alla salute e all'educazione dei pazienti e delle professioni in ambito sanitario, e finalizzate al miglioramento terapeutico**. Essa può essere divisa nelle seguenti categorie:
  - i. **Telediagnosi, teleconsulto**: trasmissione di immagini e parametri vitali a distanza per avere un parere medico o medico-specialistico relativo alla diagnosi o alla terapia su particolari casi clinici.
  - ii. **Telemonitoraggio, teleassistenza, telesorveglianza**: realizza l'assistenza (monitoraggio, gestione della terapia, ...) direttamente presso l'abitazione degli assistiti, o comunque in strutture decentrate rispetto a quelle ospedaliere.
  - iii. **Telemedicina d'emergenza**: il collegamento fra mezzi mobili e strutture fisse al fine di consentire un primo inquadramento diagnostico e terapeutico nella fase di trasporto delle persone soccorse (ed eventualmente un telemonitoraggio di alcuni parametri vitali), nonché l'accertamento tempestivo della disponibilità di posti letto presso le strutture ospedaliere più idonee.
  - iv. **Telesoccorso**: strumenti (di norma a limitata richiesta di conoscenza tecnologica da parte degli utenti) che consentono, in condizioni di emergenza, di richiedere aiuto ad un centro di controllo, utilizzando trasmettitori portatili e una rete di comunicazioni. Spesso non richiedono un intervento da parte del portatore, o comunque richiedono un intervento limitato (es. pressione di un semplice pulsante).
- b. La Telemedicina è definita come l'integrazione, monitoraggio e gestione dei pazienti, nonché l'educazione dei pazienti e del personale, usando sistemi che consentano un pronto accesso alla consulenza di esperti ed alle informazioni del paziente, indipendentemente da dove il paziente o le informazioni risiedono.

## 2. Illustrare l'importanza della telemedicina di emergenza

- a. Essa consente di fornire già una **prima diagnosi già in fase di trasporto**, consentendo di attivare quanto prima le terapie necessarie e rientrare eventualmente nella GOLDEN HOUR.

## 3. Illustrare un esempio di uso di telemedicina in aree non raggiungibili

- a. Essa consente di fornire **assistenza medica anche in zone scarsamente raggiungibili** quali isole e comunità montane, o comunque zone dove la scarsa popolazione e la scarsa densità abitativa non consentono il mantenimento di ambulatori o di centri di assistenza specializzati. Esempi di contesti:
  - i. isole, località marittime;



- ii. missioni scientifiche in posti isolati (Antartica, spazio, ...);
- iii. carceri;
- iv. missioni militari;
- v. posti segnati da una calamità naturale.

#### 4. Illustrare un esempio di uso di telemedicina per malati cronici

- a. Un altro ambito è quello dell'**assistenza domiciliare**, fondamentale per pazienti a scarsa mobilità (come gli anziani) o per i pazienti fragili, per i quali lo spostamento verso il centro di cura e l'esposizione ad altri malati potrebbe comportare rischi elevati per la salute. Importante è spesso l'implementazione di strumenti che attuano la **trasmissione automatica di dati vitali**, che se anomali vengono analizzati e il paziente viene avvisato con un campanello di allarme: in molti casi si riescono ad evitare ospedalizzazioni che non sono indispensabili.

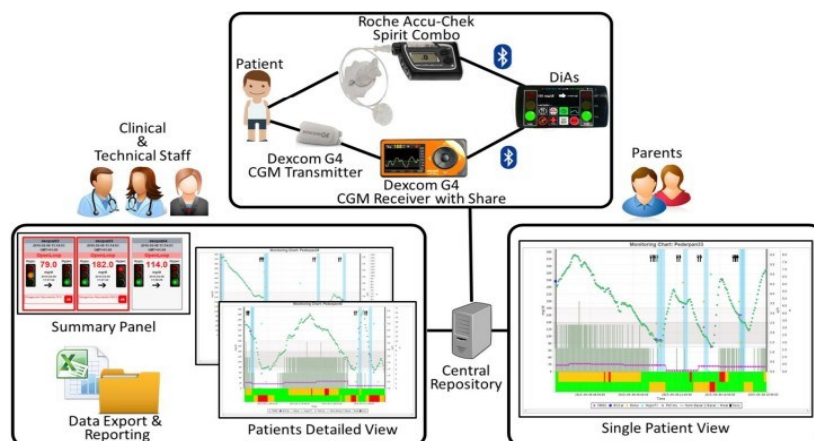
#### 5. Illustrare un esempio di uso di teleconsulto

- a. Esso è utile in casi in cui un **paziente possa avere difficoltà ad accedere ad uno specialista ma non a un medico di base**. Quest'ultimo potrà inoltrare allo specialista i dati necessari, il quale poi potrà visitare il paziente da remoto.

#### 6. Illustrare un esempio di problema di tipo legale connesso alla telemedicina

- a. I problemi di natura legale sono prevalentemente di due tipi:
  - i. **Sicurezza e privacy dei dati**: necessità di garantire la corretta e sicura preservazione dei dati, leggi sulla privacy (GDPR).
  - ii. **Responsabilità legale**: in caso di applicazione avventata o non applicazione della telemedicina.

#### 7. Illustrare schematicamente come la telemedicina è utilizzabile nella gestione del paziente diabetico



**Figure 5.** The remote monitoring architecture used for the adolescents camp held in Bardonecchia. The system was used by the clinical and technical staff for safety and research purposes and was proposed to the parents as a means of managing the disease of their children.



## 8. Illustrare cosa si intende per sistema di telemedicina multi-accesso

- a. Si intendono servizi di telemedicina che **consentano a medici e/o pazienti di interagire in modo “continuo”**, indipendentemente dal luogo in cui ci si trova e con una pletera di alternative per la connessione.

## DATA BASE – MODELLO RELAZIONALE

### 1. Spiegare la differenza fra n-upla e tupla

- a. Si definisce n-upla o tupla, una collezione o un **elenco ordinato di n oggetti**. Nella pratica:
  - i. La n-upla contiene dati **elementari**, distinguibili solo in base alla posizione.
  - ii. La tupla contiene dati **strutturati**, individuabili tramite il metadato associato (attributo) e, quindi, permutabili in qualsiasi modo.

### 2. Spiegare la differenza fra schema e istanza di una relazione

- a. La differenza è che:
  - i. Lo **schema** è la **struttura** di una relazione, ovvero il formato del record.
  - ii. L'**istanza** rappresenta i **valori** contenuti nella relazione (ovvero è la fotografia dello stato di una relazione in un determinato istante).

### 3. Spiegare i principali svantaggi della tabella universale

- a. Essa rappresenta i seguenti problemi:
  - i. **Duplicazione** delle informazioni (le stesse informazioni sono memorizzate più volte).
  - ii. **Difficoltà di aggiornare** le informazioni (devo cambiare molte righe. Ad es., se Pastore cambia indirizzo, se la retribuzione di un terapeuta cambia, se cambia il terapeuta di un trattamento...).
  - iii. **Condivisione** del file, segretezza dell'ultima colonna, ...
  - iv. Problemi di **consistenza**.

### 4. Definire un data base e un DBSM

- a. Un **data base** (o base di dati) è quindi una **raccolta di dati strutturati**, di varia natura e correlati logicamente tra loro, da cui l'operatore può evincere delle **informazioni**. In un data base si distingue una parte statica (metadati e loro interconnessione logica) ed una parte dinamica (dati elementari).

- b. Un Data Base Management System (**DBMS**) è un software che consente di **definire, costruire e manipolare una base di dati**. Essi gestiscono direttamente i file che memorizzano i dati elementari e i metadati, nonché i collegamenti logici tra questi.

## 5. Illustrare le caratteristiche principali di un DBMS

- a. Essi hanno le seguenti caratteristiche:
  - i. Grandi dimensioni: **consente di gestire grandi moli** di dati grazie allo sfruttamento della memoria secondaria.
  - ii. Condivisione: grazie alla gestione della concorrenza di eventi, **consente l'accesso simultaneo di più utenti** al DB ed evita così la duplicazione dei dati e i conseguenti problemi di consistenza/ridondanza/inefficienza/spreco di risorse.
  - iii. Persistenza: grazie all'uso della memoria secondaria, **i dati rimangono memorizzati indipendentemente dalla vita delle procedure** (es. le cosiddette viste esterne) che operano su di essi.
  - iv. **Affidabilità**: i DBMS hanno la capacità, in caso di malfunzionamento, di tornare all'ultimo stato consistente.
  - v. **Privatezza**: ciascun utente può eventualmente essere riconosciuto da un username ed avere autorizzazione ad effettuare soltanto certe operazioni.
  - vi. **Efficienza**: un DBMS ha la capacità di svolgere le operazioni in un tempo ragionevolmente breve e con risorse di calcolo e memoria accettabili.

## 6. Illustrare l'architettura a più livelli di un DBSM

- a. Un DBSM è architettato, dall'esterno (la GUI) verso l'interno (i dati), su 3 livelli:
  - i. **Schema esterno**: quanto è visibile dalla GUI.
  - ii. **Schema logico/concettuale**: corrisponde alla struttura delle tabelle, corrispondenze fra le tabelle...
  - iii. **Schema interno**: corrisponde alle strutture fisiche di memorizzazione dei dati.

## 7. Definire il prodotto cartesiano su due insiemi D1 e D2

- a. Il prodotto cartesiano dei due insiemi, D1 D2 è **l'insieme di tutte le possibili associazioni** fra gli elementi degli insiemi D1 e D2.

## 8. Spiegare la differenza tra relazione e prodotto cartesiano su D1 e D2

- a. Una relazione matematica (o "**relazione**") sugli insiemi D1 e D2 è un **sottoinsieme del prodotto cartesiano**  $D1 \times D2$ , ove D1 e D2 sono detti domini della relazione.
- b. In generale, è possibile definire prodotti cartesiani tra n insiemi,  $D1 \times D2 \times D3 \times D3 \dots \times Dn$ , e relazioni come loro sottoinsiemi. L'intero n indica il grado

della relazione, mentre la cardinalità di relazione è data dal numero delle n-uple che formano la relazione.

**9. Illustrare le proprietà di righe e colonne di una tabella**

- a. Le colonne rappresentano il **numero di attributi della tabella**, mentre le righe rappresentano il numero di entità distinte presenti nella tabella, ovvero la sua **cardinalità**.

**10. Definire il tipo di un attributo in una relazione**

- a. Un attributo è un **concetto che ha una struttura semplice e non possiede proprietà rilevanti associate**. Un attributo non ha esistenza autonoma ma è associato ad una entità o ad una relazione. Il dominio è l'insieme dei valori che possono essere assunti dai vari attributi di una relazione. Ogni dominio appartiene ad un determinato tipo, cioè stringa, numerico, alfanumerico, ...

**11. A partire da uno scontrino fiscale in nostro possesso, organizzare un DB che consenta di archiviare i dati di tutti gli scontrini che l'esercente eroga**

- a.

**12. Fare un esempio di fatto di base e di fatto derivato**

- a. Un **fatto di base** è un'informazione **direttamente ottenibile da una delle tabelle** costituenti un database, mentre un **fatto derivato** richiede la **messa in relazione** di più tabelle.

**13. Spiegare con un esempio perché è da evitare, quando possibile, la memorizzazione di fatti derivati**

- a. I fatti derivati sono **ridondanti**, e conseguentemente la loro memorizzazione richiede lo sfruttamento di spazio in memoria eccessivo.

**14. Spiegare la differenza tra chiave e superchiave di una relazione**

- a. Una **superchiave** per una relazione è un campo, o un insieme di campi, che in una relazione non può prendere lo stesso valore più di una volta. In altri termini, una superchiave **consente di identificare univocamente le tuple di una relazione**: non ci possono essere due tuple con lo stesso valore della superchiave.
- b. Un insieme di attributi K costituenti una superchiave di R è detto in particolare chiave di R se è una superchiave "minimale" (cioè se non esiste una superchiave  $K'$  tale che  $K' \subset K$ ) In altri termini, **nessun sottoinsieme degli attributi di una chiave forma ancora una chiave** (la chiave è l'insieme minimo degli attributi necessari ad identificare univocamente una riga della tabella).

**15. Dare una definizione di chiave primaria**

- a. Tra le varie chiavi di una relazione, il progettista del database ne individua una, detta **chiave primaria**, che **non può assumere (in alcuno dei suoi campi) valori nulli**.

**16. Fare un esempio di tabella che ammette solo chiavi composite**

- a. ID\_paziente,data-ora per una tabella VALORE\_GLICEMIA

### 17. Definire la chiave esterna di una tabella

- a. Una **chiave esterna** è un **attributo** (o eventualmente un insieme di attributi) che non è chiave primaria per la tabella in questione, ma **costituisce chiave primaria (eventualmente composita) per un'altra tabella**. Le chiavi esterne sono quelle che si sfruttano per mettere in corrispondenza tra loro le tabelle.

### 18. Illustrare un paio di vincoli intrarelazionali

- a. Sono vincoli che **coinvolgono il valore degli attributi** all'interno di una stessa relazione.

### 19. Illustrare un esempio di violazione del vincolo di integrità referenziale

- a. Un esempio è, in un database che gestisce la **prenotazione di esami universitari**, la presenza di uno studente nella tabella ESAMI che non sia presente nella tabella STUDENTI.

### 20. Illustrare il concetto di integrità referenziale

- a. Un vincolo di integrità referenziale associa ogni tupla della tabella referenziante R1 con la/le tupla/e della tabella referenziata R2 se ha in comune lo stesso valore nella chiave primaria. Esso impone che **tutti i valori presenti nelle chiavi di una tabella rappresentante una relazione siano presenti anche nelle tabelle indicanti gli oggetti**.
- b. L'integrità referenziale è una proprietà dei dati che stabilisce che tutti i loro riferimenti sono validi. Nel contesto dei database relazionali richiede che, **se il valore di un attributo (colonna) di una relazione (tabella) fa riferimento al valore di un altro attributo (nella stessa relazione o in una relazione diversa), il valore di riferimento deve esistere**.

### 21. Definire il modello relazionale dei dati

- a. Adottato dalla maggior parte dei DBMS attualmente in commercio, esso consente all'utente di **focalizzarsi su come i dati sono organizzati logicamente, senza preoccuparsi invece di come sono fisicamente memorizzati e gestiti internamente al sistema**. Il modello relazionale ci permette quindi di trattare i dati ad un livello logico senza interessarci del livello fisico. In altri termini, per accedere ai dati non è necessario conoscere le strutture fisiche (es. l'organizzazione del file contenente il DB) con cui sono memorizzati. Il concetto alla base di questo tipo di modello è quello di relazione. Si basa su 3 principi:
  - i. In una relazione **non** ci devono essere **tuple duplicate**.
  - ii. L'**ordine** delle tuple e degli attributi **non è importante**.
  - iii. **Ogni attributo** della relazione deve avere un **nome diverso** da tutti gli altri attributi della stessa relazione.

## 22. Definire grado e cardinalità di una tabella

- a. Il **grado** indica il numero di **attributi** della tabella, mentre la **cardinalità** è data dal numero delle **tuple** che la costituiscono.

## 23. Spiegare in quali casi si può dover usare il valore NULL in una relazione.

- a. Esso può essere usato se per alcune tuple di una relazione, il valore di un attributo non è disponibile perché **sconosciuto** o non **definibile**.

## 24. Illustrare mediante un esempio quando può servire introdurre un campo ad hoc per definire una chiave primaria

- a. Un esempio è l'**identificazione univoca** delle persone tramite **codice fiscale**, in quanto per nomi e cognomi molto comuni (e.g. Mattia Rossi) è potenzialmente possibile che ci siano più persone con lo stesso nome e cognome nate nella stessa città e nello stesso giorno.

# TEORIA EXTRA

## SISTEMI SANITARI

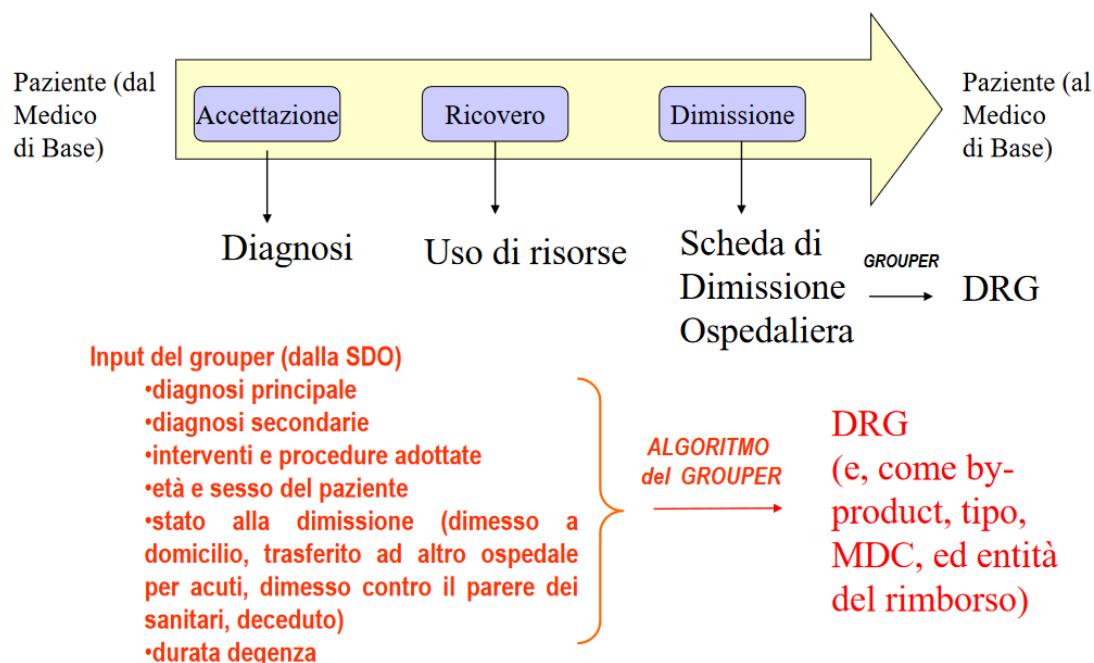
Un software certificato, detto grouper e in dotazione a tutte le strutture sanitarie, analizza l'attività svolta su un paziente durante il ricovero, desumibile dalla scheda di dimissione ospedaliera (SDO). A tale attività, e quindi ad ogni ricovero concluso, viene associato in modo "secco" uno e uno solo dei 506 DRG, ciascuno dei quali ha un diverso peso economico.

Il grouper opera inizialmente sulla sola diagnosi principale (in presenza di più diagnosi, è quella che, a giudizio soggettivo del medico, viene giudicata caratterizzare principalmente il ricovero), riconducendo il ricovero ad una delle 25 categorie diagnostiche principali (MDC, major disease category) Punti da notare:

- Le MDC sono importanti anche nella definizione delle tariffe di rimborso.
- Ogni DRG appartiene ad una sola MDC.
- Ogni MDC è contraddistinta da un costo medio della singola giornata di ricovero, che ha un ruolo importante nella determinazione delle tariffe T2 dei vari DRG in essa ricompresi.

Il grouper opera di solito in software con GUI user-friendly. Le diagnosi e gli interventi, desumibili dalla SDO e da usare come input del grouper, sono individuabili nel software attraverso menù successivi e da questo codificati automaticamente in codici standard internazionali.

## SCHEMA DEL MECCANISMO DI CALCOLO DEL DRG



Il sistema dei DRG fornisce agli enti finanziatori (Regioni), ma anche alle direzioni di ASL o di aziende ospedaliere, vari modi per quantificare la produttività a vari livelli di interesse. Alcuni esempi:

- la direzione di un'azienda ospedaliera può rapportare, reparto per reparto e/o paziente per paziente, costi reali e costi teorici di cura;

- la direzione di un' azienda ospedaliera può calcolare la produttività relativa (ovvero quanto rendono rispetto a quanto costano) dei vari reparti;
- una ASL può misurare la produttività relativa delle varie aziende ospedaliere (Az. Osp. X, Az. Osp. Y, ...) di sua competenza.

## CARTELLA CLINICA

Modalità di inserimento della diagnosi:

- Immissione libera – L'utente utilizza un manuale cartaceo.
  - Indice alfabetico: permette di reperire un codice partendo dalla patologia/proceduta.
  - Elenco sistematico: contiene l'elenco ordinato per codice → possibilità di errore.
- Supporto computerizzato all'immissione libera
  - Inserimento facilitato da strumenti per la navigazione nella terminologia.
- Immissione semi-strutturata
  - Possibile sono in ambienti particolari, l'utente è guidato nell'inserimento da liste predeterminate di scelte.

La qualità della codificazione in ambiente clinico può anche dipendere da:

- Distanza temporale tra momento di acquisizione reale dell'informazione e momento della sua effettiva codifica.
- Eventuale diversità degli attori (es. è lo staff medico che decide quali diagnosi sono rilevanti ai fini della cartella, ma è lo staff infermieristico che di solito deve in seguito fare la codifica).

Tipi di dati in cartella clinica:

1. Dati numerici (es. frequenza cardiaca).
2. Dati alfanumerici (es. colore urine).
3. Date del calendario.
4. Descrizioni a testo libero (es. refertazioni, quadri patologici, ...).
5. Segnali (es. ECG).
6. Immagini (es. TAC).
7. Suoni (es. fonocardiogramma).

Due esigenze in contrasto:

- È necessario proteggere i dati da usi non consentiti (leggi sulla "privacy").
- È necessario che gli operatori possano accedere ai dati in qualsiasi momento.

Soluzioni:

- Identificazione degli operatori con username/password e/o con smartcard.
- Verifica dei privilegi dell'operatore per l'effettuazione di determinate operazioni (a quali dati può accedere e come).
- Eventuale verifica della postazione da cui si collega l'utente.
- Politiche di gestione di password e autorizzazioni e loro monitoraggio continuo.
- Firma digitale (usando smartcard e chiavi certificate).

Ogni cartella clinica elettronica dovrà poter produrre e importare documenti in XML definiti sulla base di Document Type Definition (DTD) standard creati da Società Scientifiche Internazionali.

Tipiche attività del Medico di Medicina Generale (MMG o “medico di base”):

- diagnosi e cura di malattie acute;
- monitoraggio di malattie croniche;
- follow up di malattie passate (recupero, riabilitazione, ...).

Caratteristica della medicina di base:

- Il MMG vede il paziente più volte all'anno, ne conosce l'ambiente sociale e familiare, spesso ha in cura anche dei congiunti. Un modello di cartella clinica adatta al medico di base è quello di cartella orientata per problemi.

## SICUREZZA DATI

Indicazioni a tutela dei pazienti:

- i pazienti devono avere scelta libera se costituire o meno il database sanitario;
- in assenza del consenso il medico potrà usare solo le informazioni rese disponibili dal paziente o in precedenti prestazioni da lui fornite;
- la mancanza di consenso non deve incidere sulla possibilità di accedere alle cure;
- informazioni particolarmente sensibili (ad es. positività all'HIV) richiedono consenso specifico;
- il paziente deve essere informato su come è strutturata la cartella, chi vi ha accesso, che tipo di operazioni vi si può compiere, ...

Principali prescrizioni per i titolari del trattamento dei dati:

- garantito il diritto dei diritti contenuti nel Codice Privacy:
  - accesso ai dati;
  - rettifica dati;
  - integrazione dati;
- garantire la possibilità di “oscurare” alcuni dati;
- adottare elevate misure di sicurezza;
- comunicare al Garante eventuali fughe di dati o incidenti informatici entro 48 ore.

I dati si dividono in:

- **“dato personale”**: qualunque informazione relativa a persona fisica, giuridica, ente od associazione (es. il mio indirizzo, la mia data di nascita, ...);
- **“dati identificativi”**: dati personali che permettono l'identificazione diretta della persona (es. il mio Codice Fiscale, il numero della mia Carta d'identità, ...);
- **“dati sensibili”**: dati personali idonei a rivelare origine razziale/etnica, convinzioni religiose, opinioni politiche, preferenze sessuali, ..., nonché stato di salute (es. tutti i miei dati clinici, ...).

I sistemi informativi e i programmi informatici sono configurati riducendo al minimo l'utilizzazione di dati personali e di dati identificativi, in modo da escluderne il trattamento quando le finalità possono essere realizzate mediante dati anonimi od opportune modalità che permettano di identificare l'interessato solo in caso di necessità.

Per noi “progettisti” di sistemi:

- usare dati personali e identificativi solo se realmente necessario;
- proteggere il cammino per risalire dai dati all'identità della persona (es. nei data base).

I dati personali oggetto di trattamento sono:

1. trattati in **modo lecito** e secondo correttezza;
2. raccolti e registrati per **scopi determinati, espliciti e legittimi**, ed utilizzati in altre operazioni del trattamento in termini compatibili con tali scopi;
3. **esatti** e, se necessario, **aggiornati**;



4. **pertinenti, completi e non eccedenti** rispetto alle finalità per le quali sono raccolti o successivamente trattati;
5. conservati in una forma che consenta l'identificazione dell'interessato per un periodo di tempo non superiore a quello necessario agli scopi.

### Acquisire solo dati necessari e sufficienti e trattarli in modo corretto

Legge n.196 del 2003:

1. **Trattamento state-of-the-art** - I dati personali oggetto di trattamento sono custoditi e controllati, anche in relazione alle conoscenze acquisite in base al progresso tecnico (...), in modo da ridurre al minimo, mediante l'adozione di idonee e preventive misure di sicurezza, i rischi di:
  - a. distruzione o perdita;
  - b. accesso non autorizzato;
  - c. trattamento non consentito o non conforme alle finalità della raccolta.
2. **Politiche di controllo accessi e scadenza accrediti sono un obbligo di legge e i dati sensibili vanno criptati** - Il trattamento di dati personali effettuato con strumenti elettronici è consentito solo se sono adottate le seguenti misure minime:
  - a. autenticazione informatica;
  - b. adozione di procedure di gestione delle credenziali di autenticazione;
  - c. utilizzazione di un sistema di autorizzazione;
  - d. aggiornamento periodico dell'individuazione dell'ambito del trattamento consentito ai singoli incaricati e addetti alla gestione o alla manutenzione degli strumenti elettronici;
  - e. protezione degli strumenti elettronici e dei dati rispetto a trattamenti illeciti di dati, ad accessi non consentiti e a determinati programmi informatici;
  - f. adozione di procedure per la custodia di copie di sicurezza, il ripristino della disponibilità dei dati e dei sistemi;
  - g. tenuta di un aggiornato documento programmatico sulla sicurezza;
  - h. adozione di tecniche di cifratura o di codici identificativi per determinati trattamenti di dati idonei a rivelare lo stato di salute o la vita sessuale effettuati da organismi sanitari.

Capisaldi GDPR del 2016:

1. Si cura solo dei **dati personali**.
2. Scopi per i quali si raccolgono le informazioni e procedure di protezione devono essere **pre-dichiarati** in modo chiaro.
3. Non possono essere detenuti dati una volta venuto meno lo scopo (**oblio**).
4. Principio di responsabilità (**accountability**): non c'è una lista di operazioni o accorgimenti tecnici da seguire, ma l'obbligo di dimostrare di aver diligentemente adottato tutte le misure organizzative e di sicurezza possibili per proteggere riservatezza e integrità dei dati.
5. **Privacy by design e by default**: i sistemi vanno progettati e fatti funzionare considerando le problematiche di privacy.
6. Ruoli chiave: **titolare del trattamento, responsabile del trattamento, data protection officer (DPO)** (il DPO è obbligatorio in caso di trattamento di dati sensibili e da indicare al Garante).
7. **Valutazione di impatto**:
  - a. Obbligatoria nel caso di trattamento di dati sensibili (es. quelli biomedici) o comunque aventi carattere estremamente personale.
  - b. Il titolare (assistito dal responsabile e consultato il DPO) ha l'onere di effettuare l'analisi dei rischi inerenti libertà e diritti degli interessati (eventualmente dovrà attenuare o eliminare tali rischi o consultare il Garante).
8. **Sanzioni**: max{20 milioni di euro, 4% del fatturato}.

Il GDPR considera le seguenti tipologie di dati:

- **dato personale**: qualsiasi informazione riguardante una persona fisica identificata o identificabile («interessato»);

- **dati genetici:** dati relativi alle caratteristiche genetiche ereditarie o acquisite di una persona fisica che forniscono informazioni univoche su fisiologia o salute;
- **dati biometrici:** dati relativi a caratteristiche fisiche, fisiologiche o comportamentali di una persona fisica che ne consentono o confermano l'identificazione (es. immagine facciale).
- **dati relativi alla salute:** dati attinenti alla salute, compresa la prestazione di servizi di assistenza sanitaria.

Altre definizioni:

- **trattamento:** qualsiasi operazione (es. registrazione, strutturazione, conservazione, ...) applicata a dati personali;
- **titolare del trattamento:** la persona fisica o giuridica, l'autorità pubblica, il servizio che determina le finalità e i mezzi del trattamento di dati personali;
- **responsabile del trattamento:** la persona fisica o giuridica, l'autorità pubblica, il servizio che tratta dati personali per conto del titolare;
- **consenso dell'interessato:** volontà libera e inequivocabile con la quale si acconsente che i dati personali siano oggetto;
- **pseudonimizzazione:** trattamento dei dati personali in modo tale che i dati personali non possano più essere attribuiti a un/una interessat\* specific\* senza l'utilizzo di informazioni aggiuntive (tenute separate e protette);
- **violazione dei dati personali:** violazione che comporta accidentalmente o in modo illecito distruzione, perdita, modifica, divulgazione non autorizzata.

Principi GDPR:

1. **liceità, correttezza e trasparenza** - trattati in modo lecito, corretto e trasparente nei confronti dell'interessat\*;
2. **limitazione della finalità** - raccolti per finalità determinate, esplicite e legittime;
3. **minimizzazione dei dati** - adeguati, pertinenti e limitati a quanto necessario;
4. **esattezza** - esatti e, se necessario, aggiornati; devono essere adottate tutte le misure ragionevoli per cancellare o rettificare tempestivamente i dati inesatti;
5. **limitazione della conservazione** - conservati in una forma che consenta l'identificazione degli/delle interessat\* per un arco di tempo non superiore al conseguimento delle finalità;
6. **integrità e riservatezza** - trattati in maniera da garantire un'adeguata sicurezza dei dati personali, compresa la protezione, mediante misure tecniche e organizzative adeguate, da trattamenti non autorizzati o illeciti e dalla perdita, dalla distruzione o dal danno accidentali;
7. **responsabilizzazione** - è competente il titolare del trattamento.

È vietato il trattamento di dati sensibili o personali salvo che:

- l'interessat\* ha prestato il proprio **consenso esplicito**;
- il trattamento è necessario per tutelare un **interesse vitale** o per accertare un diritto in sede giudiziaria o per motivi di interesse pubblico;
- il trattamento riguarda dati personali resi **già pubblici** dall'interessat\* stess\*;
- il trattamento è necessario per **finalità di medicina** preventiva, medicina del lavoro, diagnosi, assistenza sanitaria o sociale, terapia;
- il trattamento è necessario per motivi di **sanità pubblica** (minacce per la salute pubblica, qualità e sicurezza dell'assistenza sanitaria e dei medicinali e dei dispositivi medici, ...).

Tecniche usate:

- la pseudonimizzazione e la cifratura dei dati personali;
- la capacità di assicurare, su base permanente, riservatezza, integrità, disponibilità e resilienza dei sistemi di trattamento;
- la capacità di ripristinare tempestivamente la disponibilità e l'accesso dei dati personali in caso di incidente fisico o tecnico;
- una procedura per valutare regolarmente l'efficacia delle misure tecniche e organizzative al fine di garantire la sicurezza.

# CRITTOGRAFIA

Algoritmo DES (Data Encryption Standard), con tutte le operazioni simmetriche (per decrittare si applica il punto 3-end con il punto 5 for  $j=16:1:1$ ):

1. Si sceglie una chiave **Ks da 64 bit**:
  - a. 56 dei 64 bit sono "liberi" ->  $2^{56}$  possibili chiavi;
  - b. ogni 7 bit, si determina un ottavo bit come bit di parità.
2. La sequenza in bit corrispondente ad es. al testo in chiaro viene divisa in **Q blocchi da 64 bit** (in pratica, per criptare un testo, si divide il testo in blocchi da 8 caratteri/codici ASCII).
3. Si considera il blocco j-esimo ( $j=1:Q$ ) da 64 bit e lo si **permuta** secondo una certa mappa P:
4. Ogni blocco permutato viene diviso nelle **due parti L e R** (left e right) da 32 bit ciascuna.
5. Per ciascuno dei Q blocchi, ripeto 16 volte un ciclo di **operazioni deterministiche di "confusione"** for  $m=1:16$ 
  - a. dalla chiave Ks a 64 bit si estrae, secondo una mappa data che varia con m, un **sottoinsieme Km di 48 bit**;
  - b. si pone:  $L_{new} = R$ ,  $R_{new} = L \text{ XOR } f(R, Km)$  dove f è una funzione binaria data (omissis) che opera su una stringa di 32 bit e su un'altra di 48 bit producendo una stringa di 32 bit;
  - c. si aggiornano **L = L<sub>new</sub> ed R = R<sub>new</sub>**.
6. Ciascuno dei Q blocchi ottenuti è permutato con la mappa P -1 (inversa della mappa del punto 3).

DES:

- Alta velocità di cifratura.
- La chiave è molto corta!

Chiave simmetrica:

- Necessità di un canale sicuro per lo scambio della chiave.
- Proliferazione chiavi (una per ogni coppia di interlocutori).

Chiave asimmetrica:

- Chiave pubblica  $K_p$ , così chiamata perché è invece conoscibile a tutti.
- una chiave segreta  $K_s$  (detta anche privata), che nessuno può conoscere al di fuori del proprietario, ricavabile dalla chiave pubblica.

Il codice RSA sfrutta il fatto che, dati due numeri interi p e q primi sufficientemente grandi:

- è molto facile stabilire il loro prodotto  $N = pq$  (problema diretto);
- è estremamente difficile risalire ad essi partendo dal loro prodotto (problema inverso).

In RSA, per un determinato utente:

- la chiave pubblica  $K_p$  è composta dalla coppia di interi (N,e);
- la chiave segreta  $K_s$  è composta dalla coppia di interi (N,d) dove d ed e sono interi ottenuti attraverso operazioni su p e q difficili da invertire.

Quindi, anche conoscendo la chiave pubblica  $K_p = (N,e)$  è estremamente difficile ottenere p e q e, quindi d, e quindi la chiave segreta  $K_s = (N,d)$ .

Algoritmo di generazione chiavi:

1. Scegliere **p e q primi** molto elevati (sono consigliati valori maggiori di  $10^{100}$ ).
2. Calcolare il valore di  **$N = p \cdot q$** .
3. Calcolare il valore di  **$z = (p-1) \cdot (q-1)$** .
4. Scegliere un intero  $e < N$  tale che e sia primo rispetto a z ( $e < N + \text{IsCoprime}(e, p-1) \ \& \ \text{IsCoprime}(e, q-1)$ ).

5. Determinare il più piccolo intero  $d$  tale che il resto intero della divisione tra  $e \cdot d$  e  $z$  sia 1  $\rightarrow \min(\text{mod}(e \cdot d, z) == 1)$ .
6. Formare le chiavi  $K_p = (N, e)$  e  $K_s = (N, d)$  e distruggere  $p$  e  $q$ .

Algoritmo cifratura RSA:

1. Il messaggio da cifrare (in binario) viene **suddiviso in blocchi di  $k$  bit** con  $2^k < N$ .
2. Per ogni blocco da  $k$  bit, si calcola il **valore numerico intero associato  $M$**  per conversione dal sistema binario al sistema decimale ( $M < 2^{(k-1)} < N$ ).
3. Per ciascun blocco si opera  **$r = \text{mod}(M^e, N)$** .
4. Ciascun valore  $r$  viene poi **ricconvertito in binario**.

Algoritmo decifratura RSA:

1. La striscia di bit ottenuta al passo **divisa in blocchi da  $k$** .
2. Essi **convertiti in decimale**, corrispondono alla sequenza degli  $M'$ .
3. Si fa per ciascun  $M'$  l'operazione  **$M'^d$** .
4. Si opera la **divisione per  $N$** , si ottengono i resti.
5. Tali **resti** vengono espressi in **binario**.

Algoritmo di hash stile digest:

- Consideriamo un messaggio di  $L$  caratteri,  $m_1, m_2, \dots, m_L$ , rappresentati con i codici ASCII su 8 bit (0-255 in decimale). Una potenziale semplice funzione di hash, con digest a  $n$  bit è:
  - $h(m_1, m_2, \dots, m_L) = \text{mod}(\sum_{i=1}^L m_i, 2^n)$

Algoritmo di XOR hash:

1. Se  $n$  è la lunghezza scelta per il digest, si divida il messaggio in  $Q$  blocchi da  $n$  bit (tipicamente risulterà  $n \ll Q$ ).
2. È quindi possibile, disponendo i  $Q$  blocchi di  $n$  bit per righe, costruire la matrice  $M$  (di dimensione  $Q \times n$ ), dove  $m_{i,j}$  è il  $j$ -esimo bit dell' $i$ -esimo blocco ( $i = 1, 2, \dots, Q; j = 1, \dots, n$ ).
3. Si computa  $c_j = m_{1,j} \text{ XOR } m_{2,j} \text{ XOR } \dots \text{ XOR } m_{Q,j}$
4. SE è Rotating hash:
  - a. Si trasforma la matrice  $M$  nella matrice  $M'$  (di dimensione  $Q \times n$ ) facendo, per ogni riga  $j$  di  $M$ , una rotazione a sinistra di  $(j-1)$  posizioni.
  - b.  $c_j = m'_{1,j} \text{ XOR } m'_{2,j} \text{ XOR } \dots \text{ XOR } m'_{Q,j}$

## COMPRESSIONE

Codifica run length:

- Un run può essere memorizzato, ad esempio, conservando il valore del primo dato seguito da un simbolo speciale (spesso  $\wedge$ ) e dalla lunghezza del run (length):
  - Poiché anche l'esistenza del run ha la sua codifica, il run dev'essere sufficientemente lungo (in questo esempio almeno 3).
  - A seconda del modo di codificare il run, ci possono essere dei vincoli sulla lunghezza massima (in questo esempio si codifica la lunghezza del run con una sola cifra, per cui al max la lunghezza può essere 10).

Codifica entropica di Huffman (-26% dell'ingombro, 2.2 bit/simbolo):

1. Riduzione della sorgente:
  - a.  $R = M$  numero di simboli.
  - b. Si elencano gli  $R$  simboli della sorgente in ordine di probabilità decrescente.
  - c. Si sommano le due probabilità più piccole, e si crea una nuova sorgente con  $R-1$  simboli.
  - d.  $R = R-1$ . Se  $R > 2$  si torna al passo 1.

## 2. Costruzione del codice:

- a. si procede a ritroso partendo dal codice ridotto
- b. Si assegna ad es. 0 al simbolo fittizio più probabile, 1 all'altro.
- c. Si va al passo indietro, per cui la sorgente ha un simbolo in più, appendendo, ad es. a destra, 0 ed 1 alla coppia dei simboli che nel passo in avanti erano stati accorpati (0 all'elemento della coppia più probabile, 1 all'altro).
- d. Se non si è già arrivati alla sorgente originale si torna a 2.

Il metodo di compressione LZW costruisce, per ogni testo da comprimere, uno specifico dizionario ad hoc in cui sequenze di (due, tre, quattro, ...) caratteri ASCII hanno un codice compatto. I primi 256 posti del dizionario sono sempre e comunque occupati dalle codifiche ASCII (a 8 bit) dei singoli caratteri.

Algoritmo di compressione:

1. Inizializza la tabella (0-255) con i codici ASCII dei caratteri singoli.
2. P = primo carattere in input.
3. WHILE non è finito il flusso dei caratteri in input:
  - a. C= carattere successivo in input
  - b. IF P+C è già nel dizionario ('+' qui significa concatenazione, in Matlab strcat(P,C))
    - i. P = P+C
  - c. ELSE
    - i. mando in output il codice di P e aggiungo P+C al dizionario.
    - ii. P = C
  - d. END
4. END
5. Mando in output il codice di P

Algoritmo di decompressione:

1. Inizializza la tabella (0-255) con i codici ASCII dei caratteri singoli.
2. OLD = primo codice in ingresso. Manda in output la traduzione di OLD.
3. WHILE non è finito il flusso di codici in ingresso
  - a. NEW = prossimo codice in ingresso
  - b. IF NEW non è nel dizionario sinora creato
    - i. S = traduzione di OLD
    - ii. S = S+C
  - c. ELSE
    - i. S = traduzione di NEW
  - d. END
  - e. manda in output S
  - f. C = primo carattere di S
  - g. aggiungi OLD+C al dizionario
  - h. OLD = NEW
4. END

La Discrete Cosine Transform (DCT) è una trasformata concettualmente simile alla trasformata discreta di Fourier (DFT) e ci dice ancora come l'energia di un segnale nel tempo si può ripartire tra componenti (stavolta coseni) a varie frequenze. Essa segue la seguente formula:

- $T_y(j) = \alpha(j) \sum_{n=0}^{N-1} y(n) \cos\left(\frac{j\pi(2n+1)}{2N}\right) \quad \text{con } j = 0, \dots, N-1, \quad \alpha(0) = \sqrt{1/N} \quad \alpha(j) = \sqrt{2/N}$
- Per  $j = 0$  (frequenza zero), la DCT restituisce un valore che ricorda la media del segnale (salvo un fattore di scala).

# DATABASE

Quando la mole di dati è molto grande, e fra essi sono previste frequenti ricerche, è cruciale la loro gestione efficace, sia per quanto riguarda la strutturazione che per quanto riguarda l'accesso. La gestione efficace dei dati consente di ottenere, velocemente e senza errori e/o omissioni, risposte ad interrogazioni (query) da cui produrre rapporti (report).

Almeno a livello ideale, la raccolta dei dati fatta in un DB ha le caratteristiche di:

- **permanenza nel tempo** (a differenza dei dati usati da un software attraverso la memoria centrale, il contenuto di un DB ha un tempo di vita indipendente dalle applicazioni che lo utilizza);
- **coerenza** (i dati sono necessari e sufficienti ad estrarre tutte le informazioni che un utente può richiedere).
- **Integrazione** (ogni informazione di interesse è ottenibile interrogando un'unica raccolta, in cui i dati sono correlati dal punto di vista logico).

Utenti di un DBMS:


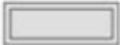








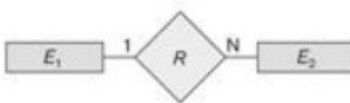
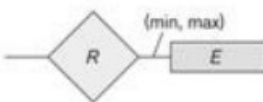
- I progettisti sono i responsabili della progettazione della base di dati (ad esempio: definizione delle tabelle, ...) (di usuale pertinenza anche del bioingegnere).
- Gli amministratori sono i responsabili del funzionamento e dell'amministrazione della base di dati (ad esempio: installazione, autorizzazioni di accesso, ...) (pertinenza informatica).
- I programmatori di applicazioni realizzano i programmi che accedono alla base di dati (ad esempio, utilizzando strumenti software ad hoc per creare interfacce verso la base di dati) (pertinenza informatica).
- Gli utenti finali usano il DB per le proprie attività, essenzialmente tramite le applicazioni rese disponibili dai programmatori di applicazioni (includono i nostri interlocutori, es. medici e personale sanitario in generale, ...).

Si può interagire con DBMS con due diverse categorie di linguaggi:

- **DDL (Data Definition Language)** linguaggio che consente di **definire gli schemi** della base di dati (es. collegamenti tra i metadati) e di gestire le autorizzazioni di accesso.
  - usato da amministratori e progettisti;
  - agisce sullo schema della BD.
- **DML (Data Manipulation Language)** linguaggio che consente di **manipolare** (inserire, modificare, cercare, ...) **i dati elementari**.
  - usato dai programmatori di applicazioni e quindi, indirettamente (e senza saperlo), dagli utenti finali;
  - agisce sull'istanza della BD.

SQL è un linguaggio che presenta, in forma integrata, sia funzionalità di DDL che di DML. Progettare una base di dati significa definirne struttura, caratteristiche e contenuto. Si tratta, come è facile immaginare, di un processo nel quale bisogna prendere molte decisioni delicate. Ci sono 3 fasi di progettazione database:

1. **Progettazione concettuale:** ha lo scopo di **descrivere le specifiche informali raccolte** sulla realtà di interesse mediante uno **schema concettuale**. Lo schema concettuale (noi utilizzeremo i **diagrammi E-R**) si pone ad un **alto livello di astrazione**, e non richiede di considerare i successivi aspetti implementativi.
2. **Progettazione logica:** consiste nella **traduzione dello schema concettuale nel modello di rappresentazione dei dati scelto** (per noi **relazionale**), detto anche schema logico. Tale schema è ancora **indipendente da dettagli fisici** (fase 3). In questa fase, le scelte progettuali tengono conto, tra l'altro, di **criteri di ottimizzazione**. Nel caso del modello relazionale dei dati, la tecnica comunemente utilizzata per ottimizzare la base dei dati è quella della normalizzazione.
3. **Progettazione fisica:** tratta **aspetti tecnologici avanzati** dipendenti dallo specifico ambiente operativo (organizzazione dei files, indici per l'accesso efficiente ai dati, ...) che sono di interesse solo per gli informatici "puri".

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	Total Participation of $E_2$ in $R$
	Cardinality Ratio 1: N for $E_1:E_2$ in $R$
	Structural Constraint (min, max) on Participation of $E$ in $R$

Lo strumento più usato è il **modello Entity-Relationship (E-R)** che, mediante interconnessione di diversi costrutti elementari (vd. figura a fianco), illustra graficamente il mini-mondo di interesse in un diagramma:

- semplice da comprendere;
- prescinde dal modello logico con cui verranno poi rappresentati i dati-

Definizioni di interesse:

- **Entità:** una classe di oggetti (fatti, cose, persone, ...) che, ai fini dell'applicazione di interesse, hanno proprietà comuni ed esistenza "autonoma".
- **Occorrenza di una entità:** particolare oggetto della classe.
- **Relazione** (o associazione): legame logico tra due entità significativo per l'applicazione di interesse. Può essere:
  - uno ad uno;
  - uno a molti;
  - molti a molti.
- **Occorrenza di associazione binaria:** una coppia di occorrenze di entità.

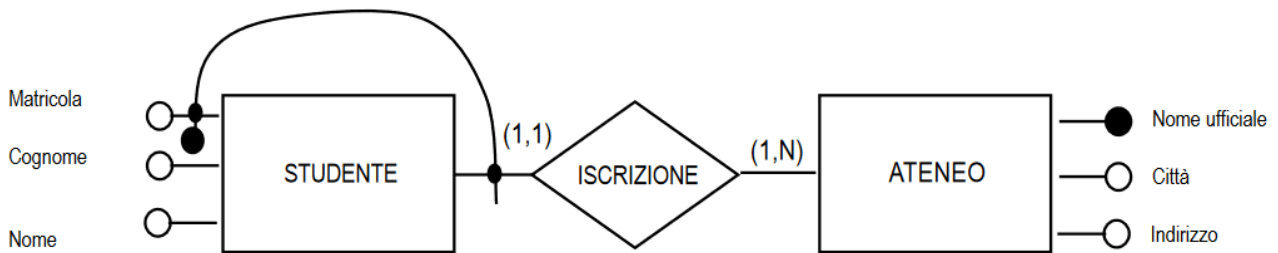
**Attributo:** descrive proprietà caratteristiche di entità o associazioni. Può essere atomico o composto, può essere:

- opzionale;
- multivalore;
- monovalore.
- **Cardinalità:** numero minimo e massimo di occorrenze di associazione.

In molti casi, uno o più **attributi** di una entità sono sufficienti ad individuare in maniera univoca le occorrenze dell'entità. Si parla allora di **identificatore interno**. Gli attributi facenti parte dell'identificatore devono avere cardinalità di attributo (1,1). Attributi multivalore o opzionali non possono far parte di identificatori. NB: alle associazioni non si applica la definizione di identificatore; infatti, le associazioni non hanno "vita propria", ed una occorrenza di associazione è di fatto individuata dalle occorrenze delle entità messe in collegamento.

Alcune entità **non hanno un identificatore interno** e si dicono per questo **deboli**. Quando per identificare una entità debole si sfrutta un'altra entità (con cui la prima è evidentemente associata con cardinalità (1,1)), si parla di **identificazione esterna**. Non si può comunque mai fare un'identificazione esterna se l'entità debole partecipa opzionalmente all'associazione o se è collocata da un lato "molti" dell'associazione (può servire introdurre un attributo ad hoc).

Ad es. STUDENTE in database nazionale → i numeri di matricola possono ripetersi, serve matricola + ateneo.



Le proprietà dei padri (attributi, identificatori, associazioni) si riportano anche alle entità “figlie” e quindi non vengono riportate nel diagramma (proprietà di “ereditarietà”).

I requisiti di un’applicazione provengono, nella maggior parte dei casi, da fonti diverse, quali:

- gli utenti/clienti finali (mediante interviste o documentazione già da loro predisposta);
- documentazione attinente al problema (moduli, regolamenti, procedure, normative, ...);
- realizzazioni preesistenti (applicazioni che si devono rimpiazzare o con cui si deve interagire).

Regole per la raccolta requisiti in linguaggio naturale:

- Scegliere il **corretto livello di astrazione**: evitare termini troppo generici o troppo specifici che rendono poco chiaro un concetto.
- **Evitare frasi contorte**: le definizioni devono essere semplici e chiare.
- **Standardizzare la struttura delle frasi**: utilizzare sempre lo stesso stile sintattico (scrivere sempre in un certo verso: es. “per <dato> rappresentiamo <insieme di proprietà>”).
- **Individuare sinonimi/omonimi e unificare i termini** (es. non usare “medico” e “dottore”).
- **Rendere esplicito il riferimento tra termini** (es. “la data dell’idoneità è riferita ai soli arbitri internazionali”).
- **Costruire un glossario dei termini**: utile per la comprensione e la precisazione dei termini usati. Il glossario contiene, per ogni termine: descrizione; possibili sinonimi; altri termini contenuti nel glossario con i quali esiste un legame logico.

Alcune “linee guida”:

1. Se un concetto ha **proprietà significative** e/o descrive classi di oggetti con **esistenza “autonoma”**, è opportuno rappresentarlo come **entità**.
2. Se un concetto ha una **struttura semplice e non possiede proprietà rilevanti associate** è opportuno rappresentarlo come **attributo** di un altro concetto.
3. Se sono state individuate **due entità e nei requisiti compare un concetto che le associa**, tale concetto può essere rappresentato da un’**associazione**.
4. Se uno o più concetti sono interpretabili come casi **particolari di un concetto “padre”**, è opportuno rappresentarli facendo uso di una **generalizzazione**.

Le due strategie sono, al solito:

1. **Strategia top-down**: inizialmente si trascurano i dettagli e si realizza uno **schema concettuale di massima** (“diagramma scheletro”) che viene poi **raffinato** entrando nel merito di un concetto alla volta.
2. **Strategia bottom-up**: inizialmente si descrivono **in dettaglio le varie componenti**, una alla volta, per poi procedere all’**integrazione nello schema concettuale finale** (utile per problemi complessi per lavorare in parallelo in più persone).



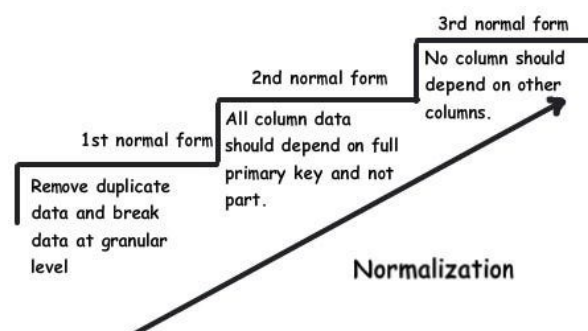
Dal diagramma E-R, per passare alla struttura logica del database relazionale si seguono di norma tre regole fondamentali:

1. un'associazione con cardinalità **(1:1)** può essere tradotta in **una sola tabella**;
2. un'associazione con cardinalità **(1:n)** può essere tradotta in **due tabelle**: una tabella si usa per l'**entità che può partecipare ad n occorrenze** (lato molti), mentre l'altra tabella si usa per ospitare i **dati dell'altra entità** (lato 1) e dell'associazione;
3. un'associazione con cardinalità **(n:m)** può essere tradotta in **tre tabelle**, una per **ciascuna delle due entità coinvolte** e una per l'**associazione**.

## OTTIMIZZAZIONE

Il processo di **normalizzazione** di un data base è il procedimento attraverso cui **le relazioni definite in una prima bozza vengono via via scomposte in modo da arrivare ad un insieme di relazioni nella forma normale di grado più elevato**. Le forme normali hanno **cinque gradi**:

- **La prima, la seconda e la terza forma normale** (la terza ha anche una variante, quella di Boyce-Codd) consentono di **evitare ridondanze determinate da dipendenze funzionali**.
- **La quarta e la quinta forma normale** riducono le ridondanze dovute alle cosiddette **dipendenze multivalore**.



Definizioni:

- **Dipendenza funzionale** (Functional Dependency, FD): **corrispondenza ad un solo valore tra un attributo e un altro**. La dipendenza funzionale è normalmente monodirezionale (dal "determinante" al "determinato"), ma a volte è bidirezionale.
  - si dice **completa** (FFD, Full Functional Dependency) quando il **determinante è "minimale"**, cioè **non contiene attributi non necessari**.
- **Attributo di una relazione**:
  - **Primario**: fa parte di almeno una chiave della relazione (NB: per individuare correttamente gli attributi primari è quindi cruciale ricordare che per una stessa relazione ci possono essere più chiavi).
  - **Non primario**: non fa parte di alcuna chiave della relazione.

Le varie forme sono:

1. **Prima forma normale (1NF)**: Una relazione è in Prima Forma Normale (abbreviato con 1NF, First Normal Form) se tutti i suoi **attributi assumono valori atomici, o semplici**.
  - a. Una relazione è in prima forma normale (1NF) se tutti i suoi attributi assumono valori atomici.
2. **Seconda forma normale (2NF)**: 1NF + **tutti i suoi attributi non primari sono in dipendenza funzionale completa da ogni possibile chiave**. NB: se la chiave sotto esame

è semplice (ovvero se è composta da un solo attributo), ogni dipendenza funzionale da essa è ovviamente completa.

- a. Una relazione 1NF è anche in seconda forma normale (2NF) se tutti i suoi attributi non primari sono in dipendenza funzionale completa da ogni possibile chiave.
3. **Terza forma normale (3NF):** 2NF + **non contiene dipendenze funzionali fra attributi non primari**. NB: se in una tabella c'è una dipendenza funzionale di un attributo non primario da un altro attributo non primario significa che c'è ridondanza + quando in una relazione in 2NF esiste un solo attributo non primario, allora la relazione è automaticamente in 3NF.
  - a. Una relazione 2NF è anche in terza forma normale (3NF) se non contiene dipendenze funzionali fra attributi non primari.
4. **Forma normale di Boyce-Codd (BCNF):** 3NF + **ogni attributo primario è in dipendenza funzionale completa (FFD) da ogni chiave di cui non fa parte**.
  - a. Una relazione in 3NF con una sola chiave ammissibile è ovviamente anche in BCNF. Si può dimostrare che una relazione in 3NF è anche in BCNF se, in ciascuna delle dipendenze funzionali che si possono individuare fra i vari attributi, il determinante è comunque una superchiave.
  - b. Una relazione in 3NF che ammette più chiavi è anche in forma normale di Boyce-Codd (BCNF) se ogni attributo primario è in dipendenza funzionale completa da ogni chiave di cui non fa parte.
5. **Quarta forma normale (4NF):** BCNF + **per ogni dipendenza multivalore  $x y$ , la relazione contiene solo attributi in  $x$  ed  $y$** .
6. **Quinta forma normale (5NF):** 4NF + **non si può più decomporre senza perdita di informazione**.

## ALGEBRA RELAZIONALE

In algebra relazionale, esistono un insieme di operazioni di base che hanno come input una o più relazioni e come output una nuova relazione. Ci sono le seguenti operazioni:

- **Operatori insiemistici: due tabelle (con lo stesso schema) → una tabella (mantiene lo schema).**
  - **Unione:**  $R(X) = R1(X) \cup R2(X)$  di due relazioni  $R1(X)$  e  $R2(X)$  definite sullo stesso schema  $X$  è la rappresentazione che contiene le tuple che appartengono ad  $R1$  oppure ad  $R2$ , oppure ad entrambe.
  - **Intersezione:**  $R(X) = R1(X) \cap R2(X)$  di due relazioni  $R1(X)$  e  $R2(X)$  è la relazione contenente le tuple che appartengono sia a  $R1(X)$  che a  $R2(X)$ .
  - **Differenza:**  $R(X) = R1(X) - R2(X)$  di due relazioni  $R1(X)$  e  $R2(X)$  è la relazione contenente le tuple che appartengono a  $R1(X)$  ma non a  $R2(X)$ .
  - NB:  $R1 \cap R2 = R1 - (R1 - R2)$
- **Operatori di ridenominazione, selezione e proiezione: una tabella → una tabella (con schema uguale o diverso).**
  - Ridenominazione  $\rho(R)$ : modifica il nome di uno o più attributi di una relazione  $R$ , lasciandone inalterato il contenuto. L'operazione di ridenominazione può essere fatta sul nome della relazione lasciando inalterati i nomi degli attributi.

$\rho_{\text{Nuovo Attributo } i\text{-esimo}, \text{Nuovo Attributo } j\text{-esimo} \leftarrow \text{Vecchio Attributo } i\text{-esimo}, \text{Vecchio Attributo } j\text{-esimo}}(R)$

- Selezione  $\sigma(R)$ : su una relazione  $R$ , indicata con  $\text{cond}(R)$ , produce un'altra relazione  $R_2$ , con lo stesso schema di  $R$ , che contiene di  $R$  solo le tuple che soddisfano la condizione  $\text{cond}$  (in altri termini  $\text{cond}(R)$  realizza un "filtraggio" di  $R$ ).  

$$\sigma_{\text{cond}}(R), \quad \vee \text{ OR} \quad \wedge \text{ AND} \quad \neg \text{ NOT}$$
- Proiezione  $\pi(R)$ : dati una relazione  $R$  sugli attributi  $X$  e un sottoinsieme  $YX$ , la proiezione di  $R$  su  $Y$ , che indichiamo con  $\pi_Y(R)$ , è la relazione che include le tuple di  $R$  ottenute considerandone solo i valori su  $Y$  (si eliminano tout court alcune colonne).
- **Operatori di join (prodotto cartesiano, join naturale, theta-join): due tabelle (con schemi anche diversi) → una tabella (con schema dato da una "giunzione" dei due schemi).**
  - **Prodotto cartesiano**: combina due relazioni (senza campi in comune)  $R_1 \times R_2$  è la relazione che si ottiene semplicemente affiancando in ogni modo possibile tuple della prima tabella e tuple della seconda tabella.
  - **Join**: un operatore che riunisce due relazioni combinandone i dati sulla base dei valori dei loro attributi:
    - **Join naturale** ( $X \bowtie Y$ ): operatore che concatena dati in relazioni diverse sulla base di valori uguali in attributi con lo stesso nome. Per attributi ha l'unione degli attributi delle relazioni di partenza (il campo, o i campi, in comune non vengono duplicati) e per tuple quelle ottenute giustapponendo le tuple della prima relazione a quelle della seconda relazione che presentano, su tutti gli attributi comuni, valori uguali.
      - Il numero di attributi della relazione prodotta dalla join  $R_1 \bowtie R_2$  è al più uguale alla somma dei numeri di attributi delle due relazioni  $R_1$  e  $R_2$ .
      - Nei casi pratici, il join è spesso fatto coinvolgendo una chiave esterna (NB: per usare il join naturale il nome degli attributi deve coincidere).
      - Il vincolo di integrità referenziale tra gli attributi comuni su cui si basa il join naturale di  $R_1$  e  $R_2$  evita che  $R_1$  faccia riferimento a valori inesistenti in  $R_2$ .
    - **Outer join**: si può "forzare" il fatto che tutte le tuple di una relazione (o di entrambe le relazioni) diano un contributo al risultato usando delle varianti della join chiamate outer join (join esterni). Esso può essere fatto a sinistra, a destra o da ambedue i lati.
    - **Theta-join**: un'operazione che consente di correlare due relazioni (aventi schemi disgiunti) sulla base del confronto tra i valori delle relazioni relativi ad attributi con nome diverso. La theta join fra due relazioni  $R_1$  e  $R_2$ , eseguita sulla condizione  $\text{cond}$ , si indica con  $R_1 \bowtie_{\text{cond}} R_2$ .  $\text{Cond}$  può essere un'uguaglianza (codice = codice\_fiscale).  
Equivale:  $R_1 \bowtie_{\text{cond}} R_2 = R_1 \bowtie (\sigma_{\text{cond}}(R_2))$

# MATLAB – FUNZIONI UTILI

- **Comandi iniziali:**

- `clc`: pulisce la Command Window
- `help <NomeComando>`: consulta l'help in linea per il comando NomeComando
- `lookfor <NomeComando>`: ricerca nel manuale la parola chiave NomeComando
- `demos`: dimostrazioni
- `exit`: chiude Matlab
- `what`: dice cosa è contenuto nel folder
- `open M-file`: apre M-file nel folder

- **Size:**

- `[n_rows, n_columns] = size(Matrix)`
- `n_columns = size(Matrix, 2)`
- `len = length(vector)`

- **Boolean:**

- `==` uguale
- `~=` diverso da
- `<` minore di
- `<=` minore o uguale
- `>` maggiore
- `>=` maggiore o uguale
- `&` and logico
- `|` or logico
- `~` not logico
- `and` – Logical AND `&`
- `or` – Logical OR `|`
- `not` – Logical NOT `~`
- `xor` – Logical EXCLUSIVE OR

- **Conversione tipi:**

- `stringa = num2str(numero)`
- `numero = str2num(stringa)`
- `ASCII_values = double(stringa)`
- `stringa = char(ASCII_values)`
- `strcmp(stringa_1, stringa_2)`
- `dec2bin(numero, nbit)` per ottenere le rappresentazioni da nbit

- **Gestione file:**

- `save file Variabile_1 Variabile_2 Variabile_3`: salva variabili in un file
- `load file Variabile_1 Variabile_2 Variabile_3`: carica variabili da un file

- **Generazione matrici di valori numerici:**

- `A = linspace(Min, Max, N)`: crea un vettore di N valori tra Min e Max, distribuiti linearmente
- `A = logspace(Min, Max, N)`: crea un vettore di N valori tra Min e Max, distribuiti logaritmicamente
- `A = Min:length_step:Max`
- `A = eye(N)`: matrice identità N×N

- `A = ones(n_rows, n_columns)`
- `A = diag([3, 5, 6])` matrice 3×3 con elementi sulla diagonale specificati
- `c = a*b'`: prodotto scalare con a e b vettori riga

- **Proprietà matrici:**

- `inv(X)`: matrice inversa di X
- `det(X)`: determinante di X
- `rank(X)`: rango di X
- `trace(X)`: traccia di X
- `eig(X)`: autovalori di X
- `poly(X)`: polinomio caratteristico di X
- `norm(X, p)`: norma p di X (matrice o vettore che sia)

- **Statistica:**

- `A = randn(n_rows, n_columns)`: crea una matrice con valori randomici distribuiti secondo  $N(0, 1)$ . Per avere media e varianza diversa, fare `Media+Varianza*randn(n_rows, n_columns)`
- `A = rand(n_rows, n_columns)`: matrice con elementi casuali distribuiti uniformemente in  $[0, 1]$
- `max(x)`, `min(x)`: massimo e minimo del vettore x (NB: procede per colonne se x è matrice)
- `sort(x)`: ordinamento ascendente del vettore x (per colonne se x è matrice)
- `mean(x)`, `median(x)`, `var(x)`, `std(x)`: media, mediana, varianza e sd campionaria di x (per colonne se x è matrice)

- **Manipolazione di matrici:**

- `A = find(condizione_in_matrix)`
- `A = fliplr(matrix)`: flippa la matrice da dx a sx
- `A = flipud(matrix)`: flippa la matrice da up a down

- **Matematica:**

- `cos`, `sin`, `cosh`, `sinh`, `tan`, `tanh`, `asin`, `asinh`, `acos`, `acosh`
- `log`, `log10`, `log2`, `exp`
- `abs`, `mod`, `sqrt`
- `round`, `floor`, `ceil`, `sign`, `fix`
- `polyval(p, x)`: calcola il valore del polinomio di coefficienti p in x
- `roots(p)`: radici del polinomio di coefficienti p
- `poly(r)`: determina i coefficienti del polinomio le cui radici sono r

- **Tipi di plot:**

- `plot(x, y)`
- `bar(x, y)`: produce un diagramma a barre
- `hist(y, m)`: suddivide l'intervallo dei valori compresi tra il minimo e il massimo di y in m "bin" (=sotto intervalli) di egual grandezza e calcola (e poi disegna) il numero di elementi di y compresi in ogni bin
- `stem(x, y)`: adatto quando si vuole mettere in evidenza il fatto che il segnale è a tempo discreto
- `semilogx(x, y)`: come plot ma l'asse x viene rappresentato in scala log10
- `semilogy(x, y)`: come plot ma l'asse y viene rappresentato in scala log10
- `loglog(x, y)`: come plot ma entrambi gli assi in scala log10

- **Specifiche plot:**

- `figure(n)`: apre la figura n. Se la figura n è già esistente, la rende la figura (attiva), ovvero quella su cui i plot avranno effetto
- `close`: chiusura finestra grafica corrente
- `close all`: chiusura di tutte le finestre grafiche
- `close(n)`: chiude a figura n
- `clf`: cancellazione grafici, riquadri, etc. dalla figura corrente
- `title(stringa)`: inserisce il titolo nella figura attiva; stringa può contenere sequenze LaTeX
- `xlabel(stringa)`: aggiunge il testo all'asse delle ascisse
- `ylabel(stringa)`: aggiunge il testo all'asse delle ordinate
- `grid on/off`: attiva/disattiva la griglia nella figura attiva
- `axis([xmin, xmax, ymin, ymax])`: specifica i range per ascisse e ordinate
- `axis tight`: gli assi finiscono al valore massimo/minimo dei dati (non rimane contorno)
- `axis equal`: fa in modo che incrementi unitari sui due assi abbiano la stessa lunghezza effettiva su schermo
- `axis square`: si impostano gli stessi min e max per entrambi gli assi (visualizzazione "quadrata")
- `axis normal`: si ritorna alla configurazione originale degli assi

```
function [out1, out2, ...] = NomeFunzione(in1, in2, ...)
```

```
% commenti per l'help online
```

```
out1 = ...
```

```
out2 = ...
```

```
return %superfluo se in fondo
```