

DATA SCIENCE
&
MACHINE LEARNING
FOUNDATIONS



- La **DATA SCIENCE** è una disciplina che raccoglie un insieme di tecniche e modelli matematici, statistici e comunicativi con lo scopo di raccogliere, processare, analizzare e prevedere il comportamento di grandi quantità di dati.

- I processi di lavoro sui dati prevedono 3 fasi:

- **DATA ENGINEERING**:

- **TERMINAZIONE** E ARCHIVIAZIONE DEI DATI GREZZI;
 - TRASFERIMENTO E REPLICAZIONE DEI DATI GREZZI;
 - PRE - PROCESSING DEI DATI GREZZI, PER ADATTARLI ALL'ELABORAZIONE;

- **DATA ANALYTICS**:

- ESTRAZIONE DI INFORMAZIONI TRAMITE DATA MINING;
 - ANALISI DATI CON TECNICHE STATISTICHE;
 - CREAZIONE DI MODELLI PREDITTIVI E DI PATTERN RECOGNITION TRAMITE MACHINE LEARNING;

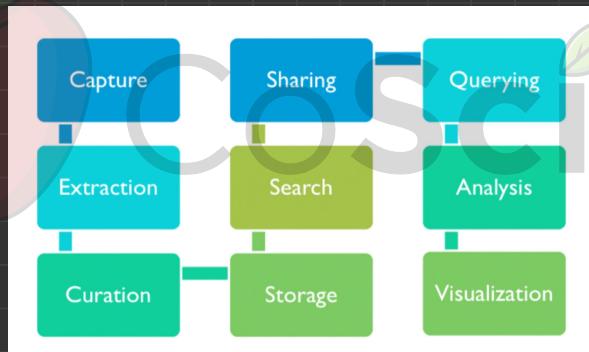
- **DATA NARRATIVE**:

- PRESENTAZIONE DEI RISULTATI TRAMITE GRAFICI E STORYTELLING.

- LE 3 ATTIVITA' HANNO COSTO DECRESCENTE IN TERMINI DI TEMPO E DENARO, MA GENERANO CRESCENTE VALORE ECONOMICO.

BIG DATA : USI E CARATTERISTICHE

- I **BIG DATA** SONO COLLEZIONI DI DATI ESTSE AL PUNTO DI NON ESSERE GESTIBILI CON I SISTEMI TRADIZIONALI (ES. DBMS), RICHIEDENDO PUNQUE TOOL SPECIFICI.



- LE CARATTERISTICHE FONDAMENTALI DEI BIG DATA SONO:

- **VOLUME**: ENORME QUANTITÀ DI DATI;

- **VELOCITY**: VELOCITÀ DI GENERAZIONE;

- **VARIETY**: VARIETÀ ED ETNOGENESIA;

- **VERACITY**: VERIDICITÀ ED AFFIDABILITÀ DEI DATI;
- **VALUE**: VALORE ECONOMICO ATTRIBUITO AI DATI;

• I BIG DATA TROVANO APPLICAZIONE IN MOLTISSIMI ABITATI:

- **BUSINESS**: SISTEMI DI RACCOLTA DI DATI, MODELLI PREDITTIVI, USER SEGMENTATION, ...
- **SCIENZA**: DATI DA OSSERVATORI ASTRONOMICI, ACCELERATORI DI PARTICELLE, ...

• L'AVVENTO DEI BIG DATA HA PERMESSO L'INDIVIDUAZIONE DI CONNESSIONI IN ESSI, CHE RAPPRESENTANO UN PESANTE INDIZIO PER UNA POSSIBILE RELAZIONE DI CAUSA - EFFETTO.

Associazione

ASPECTI CRITICI DEI BIG DATA

- DIFFICOLTÀ COMUNICATIVA TRA DATA ENGINEER E DATA SCIENTIST E STAKEHOLDER;
- DIFFICOLTÀ INTERPRETATIVA DELLE GRANDI QUANTITÀ DI DATI;
- DIFFICOLTÀ NELLA RACCOLTA DEI DATI, SPECIALMENTE SE DI ELEVATA QUALITÀ.

DATA PROFILING

DATA PROFILING: INTRODUZIONE

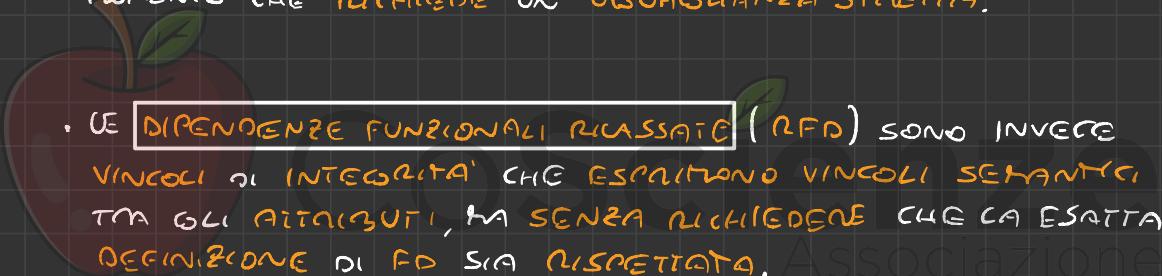
- IL **DATA PROFILING** È UNA BRANCA DELLA DATA SCIENCE CHE SI OCCUPA DI ESTRARE METADATI, CONNESSIONI E VINCOLI DAI DATI GREZZI IN MODO AUTOMATICO.
- I METADATI SONO ESTIMATTI TRAMITE SPECIFICI ALGORITMI DI DISCOVERY, E APPARTENGONO PRINCIPALMENTE ALLE SEGUENTI CATEGORIE:
 - VINCOLI DI UNICITÀ DI SINGOLI ATTRIBUTI O INSIEMI DI ATTRIBUTI, IL CUI DISCOVERY HA TEMPO ESPONENZIALE;
 - VINCOLI DI DIPENDENZA FUNZIONALE, DI CUI SI PUÒ CALCOLARE UNA COBERTURA MINIMA UGUALE ALLA POTENZA INTEGRALE, MA IL CUI EVENTUALE DISCOVERY È ESPONENZIALE;
 - VINCOLI DI **DIPENDENZA FUNZIONALE RICASSATA**, UNA FORMA PIÙ GENERALE DELLE DIPENDENZE FUNZIONALI;
 - VINCOLI DI INTEGRITÀ REFERENZIALE TRA SCHEMI NON SPECIFICATI IN MODO ESPLICATIVO, IL CUI DISCOVERY È ANCORA UNA VOLTA, O^I TEMPO ESPONENZIALE;

DIPENDENZE FUNZIONALI E CRITERI DI RIASSAMENTO

- Sia $R = \{A_1, A_2, \dots, A_m\}$ uno schema di relazione, siano $X, Y \subseteq R$; allora definiamo **DIPENDENZA FUNZIONALE** il **VINCOLO DI INTEGRITÀ**:

$$- X \rightarrow Y \stackrel{\text{def}}{\iff} \forall t_1, t_2 \in r, t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$$

- QUESTA DEFINIZIONE ESPONE UN VINCOLO SEMANTICO TRA GLI ATTRIBUTI X ED Y , MA RISULTA TOLTO RESTRITTIVA, DAL MOMENTO CHE RICHIEDE UN'UGUAGLIANZA STETICA.



- LE **DIPENDENZE FUNZIONALI PUASSATI** (RFD) SONO INVECE VINCOLI DI INTEGRITÀ CHE ESPONNO VINCOLI SEMANTICI TRA GLI ATTRIBUTI, MA SENZA RICHIEDERE CHE LA ESATTA DEFINIZIONE DI FD SIA RISPETTATA.

- ESEMPIO:

$$1.) \{CITTÀ, VIA\} \rightarrow CAP_{EQ}$$

- **ATTRIBUTE COMPARISON**:

- UNO DEI CRITERI DI RIASSAMENTO GUARDA IL CONFRONTO TRA I VALORI DEGLI ATTRIBUTI, CHE PUÒ AVVENIRE IN 2 DIVERSE MANIERE:

- CONFRONTO APPROSSIMATO (DISTANZA O SIMILANZA);

- RELAZIONE D'ORDINE (es. $t_1[X] \leq t_2[X] \Rightarrow t_2[Y] \leq t_2[Z]$);

- ENTREBI CONCETTI SONO GENERALIZZATI DA QUELLO DI **VINCOLO**, UN PREDICATO DEFINITO SUGLI ATTRIBUTI DI UNO STESSO OGGETTO;

- $\phi(A, B)$

- CHE VALE TRUE IN BASE A:

- SE LA RELAZIONE D'ORDINE È SODISFATTA;

- SE LA SIMILARITÀ O DISTANZA SUPERNA/NON SUPERNA UNA CERTA SOGLIA.

- ESEMPIO:

$$1.) \phi_1(A, B) = \begin{cases} \text{TRUE SE } A \leq B \\ \text{FALSE ALTRIMENTI} \end{cases}$$

CONCATENAZIONE ESEMPIO $A = t_1[X] \in B = t_2[X]$

$$2.) \phi_2(A, B) = \begin{cases} \text{TRUE SE } \delta_2(A, B) \leq 0.3 \\ \text{FALSE ALTRIMENTI} \end{cases}$$

CONCATENAZIONE ESEMPIO $A = t_1[X] \in B = t_2[X]$

• **EXTENT**:

- L'ALTO CITERIO DI RICASSAMENTO DELLE RFD RIQUARDA L'**EXTENT**, DENTRO IL QMDO ALL'ESTENSIONE ALLE TUPLE, FACENDO IN modo TALE CHE LA FD VALGA SOLO PER UN SOTTOINSIEME DELLE TUPLE, BASANDOSI SU:

- UNA **COVERAGE MEASURE** Ψ , DEFINITA SU UNA RFD Ψ , QUANTIFICA IL QMDO DI SODDISFAZIONE DI Ψ SU UN'ISTANZA DI RELAZIONE r :

$$- \quad \boxed{\Psi : D(X) \times D(Y) \rightarrow \mathbb{R}}$$

- ESISTONO VARI TIPOLOGIE DI MISURE DI COVERAGE:

- **COVERAGE**: MISURA DELLA COPRIVANZIA DEL MASSIMO SOTTOINSIEME $r_1 \subseteq r$ T.C. Ψ È VALIDA SU r_1 ;

- **δ_3 ENRON**: MINIMO NUMERO DI TUPLE DA MUOVERE AFFINCHÉ Ψ VALGA SU OGNI TUPLA;

• ESERCIZIO:

$$1.) \{ \text{CITA} \approx, \text{INDIRETTO} \approx \} \rightarrow \text{CAP}_{\text{EQ}}$$

CAP	CITA	INDIRETTO
84123	SALENNO	PIAZZA G. Mazzini
84123	SA	P. Mazzini
84123	SALENNO	VIA G. VICINANZA
84126	SA	PIAZZA VICINANZA

La CONFIDENZA VALE
3/4

- UN **INSIEME DI CONDIZIONI** C CHE DETERMINA IN modo UNIVOCO UN SOTTOINSIEME DI TUPLE SU CUI UNA RFD ψ È VERA:

- $D_C \stackrel{\text{def}}{=} \left\{ t \in D : \bigwedge_{i=1}^k C_i(t[A_i]) \right\}$, DOVE:
- $D \subseteq R$ È UN DOMINIO DEFINITO SU UN INSIEME DI ATTRIBUTI $\{A_1, A_2, \dots, A_n\}$;
- C_i È UN PREDICATO DEFINITO SUL DOMINIO DI A_i : $D(A_i)$ CHE FILTRA LE TUPLE SU CUI ψ È VERA.

DEFINIZIONE GENERALE DI RFD

- SIANO $R_1 = \{A_1, \dots, A_K\}$ E $R_2 = \{B_1, \dots, B_F\}$ DUE SCHEMI DI RELAZIONI APPARTENENTI ALLO SCHEMA DI DB R , ALLORA DEFINIAMO RFD ψ IL VINCOLO DI INTEGRITÀ:

- $D_{C_1} \times D_{C_2} : (X_1, X_2)_{\phi_1} \xrightarrow{\psi \geq \varepsilon} (Y_1, Y_2)_{\phi_2}$, DOVE:
- D_{C_1}, D_{C_2} SONO DOMINI CONDIZIONATI DA DUE INSIEMI DI CONDIZIONI C_1, C_2 , DEFINITI RISPECTIVAMENTE SU R_1 ED R_2 ;
- $X_1, Y_1 \subseteq R_1 \in X_2, Y_2 \subseteq R_2$;
- ψ È UNA COERCITIVE MEASURE CHE DEVE ASSUNGERE VALORE $\geq \varepsilon$;

- $\phi_1 \in \phi_2$ sono insiemi di vincoli definiti su $D(X_1), D(X_2) \subseteq D(Y_1), D(Y_2)$;

- DUE ISTANZE $r_1 \in D_{C_1} \in R_1 \subseteq D_{C_2}$ e $r_2 \in D_{C_2} \subseteq D_{C_1}$ soddisfano la RFD $\psi \stackrel{\text{def}}{\iff}$
 $\forall t_1, t_2 \in (r_1, r_2)$:

- $\phi(X_1, X_2) = \text{true}$, per ogni vincolo $\phi \in \phi_1$;

- $\phi(Y_1, Y_2) = \text{true}$, per ogni vincolo $\phi \in \phi_2$;

- LA COVERAGE MEASURE $\psi \geq \varepsilon$ su $R_1 \subseteq R_2$;

- USANDO QUESTA DEFINIZIONE, POSSIAMO AD ESEMPIO OTTENERE LA DEFINIZIONE DI FO CANONICA:

- $D_{\text{true}} : X_{\text{eq}} \xrightarrow{\psi_{\text{true}}(t)} Y_{\text{eq}}$, DOVRE:

- $D_{C_1} \times D_{C_2} = D_{\text{true}}$ INDICA CHE NON CI SONO RESTRIZIONI SULLE TUPLE SU CUI ψ SI APPLICA;

- $X_{\text{eq}} \in Y_{\text{eq}}$ INDICANO CHE $X = X_1 = X_2 \in Y = Y_1 = Y_2$;

- EQ INDICA CHE LA METRICA DI CONFRONTO USATA È L'UUGUALIANZA STRETTA =;

- $\Psi_{\text{err}(0)}$ È UNA COVERAGE MEASURE CHE INDICA LA NON AMMISSIBILITÀ DI TUTTE CHE NON RISPETTANO IL VINCOLO.

RFO CHE RICASSANO SULL'EXTENT

- IN GENERALE, UNA FD ψ POTREBBE NON VALERE SU CERTI TUTTI DEL DATASET, PER DIVERSI MOTIVI:

- ERRORE DI NORMALIZZAZIONE;

- VALORI NULLI O MANCATI;

- PRESENZA DI OUTLIER;

- UNA RFO CHE RICASSA SULL'EXTENT PUÒ USARE ALLORA UNA COVERAGE MEASURE ψ :

$$- D_{\text{rule}} : X_{\text{eq}} \xrightarrow{\psi(X,Y) \geq \varepsilon} Y_{\text{eq}},$$

- OPPURE PUÒ USARE UNA CONDIZIONE DI DOPPIO C.

$$- D_c : X_{\text{eq}} \xrightarrow{\Psi_{\text{err}(0)}} Y_{\text{eq}}$$

- C È UN INSERTE DI PREDICATI DEFINITI SUGLI ATTRIBUTI DELLA SCHEMA.

APPROXIMATE FUNCTIONAL DEPENDENCY (AFD)

- UN'APPROXIMATE FUNCTIONAL DEPENDENCY (AFD) È UNA RFD CLASSIFICATA SULL'ESTENTO CHE È VALIDA "QUASI" PER OGNI TUPLA, ED UTILIZZATA COME COVERAGE MEASURE:
 - IL ψ -ERRORE DIVENTA LA MINIMA PERCENTUALE DI TUPLE DA RIMUovere AFFINCHE' ψ VALGA SU TUTTI:

$$-\psi(X, Y) \stackrel{\text{def}}{=} \frac{\min\{|n_i| : n_i \in X \rightarrow Y \text{ vale in } \pi_i\}}{|n_i|}$$

- UNA CONFIANZA, DIVENDO IL MASSIMO NUMERO DI PERCENTUALE DI TUPLE SU CUI ψ VALE:

$$-\psi(X, Y) \stackrel{\text{def}}{=} \frac{\max\{|n_i| : n_i \in X \rightarrow Y \text{ vale in } \pi_i\}}{|n_i|}$$

- DUNQUE, DATO UN $\varepsilon \in [0, 1]$ COME SOGLIA, UNA AFD È DEFINITA COME UN RFD:

$$- D_{\text{TRUE}} : X_{\leq \varepsilon} \xrightarrow{\psi(X, Y) \leq \varepsilon} Y_{\leq \varepsilon}$$

ESEMPIO:

1-) IN UN DATASET CLINICO, POSSANO ESSERE VARIABILI:

- D_{TRUG}: Non Eq $\xrightarrow{\Psi(x,y) \leq 0.2}$ Gruppo sanguigno da

- IL 20% DI CASI POSSANO ESSERE ODOVUTI A DECCE DENTALI.

PURITY DEPENDENCIES (P.D.)

DATI 2 PARTIZIONI Tl'_S E Tl''_S DI UN INSERTE S, LA MISURA DI IMPURITÀ, SIA $X \in Tl'_S$ UN BLOCCO DI Tl'_S :

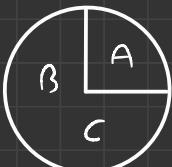
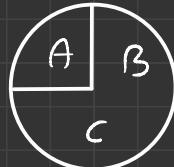
$$-\Theta(X) = 0 \Leftrightarrow$$

$$-\exists \text{ blocco } Y \in Tl''_S \text{ i.e. } X \subseteq Y$$

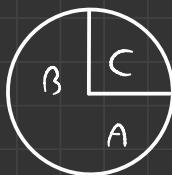
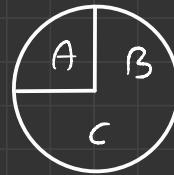
- Dunque il blocco $X \in Tl'_S$ è incluso in un solo blocco $Y \in Tl''_S$, e non interseca nessun altro $Z \in Tl''_S$;

- INOLTRE:

- $\Theta(X)$ CRESCE AL CRESCERE DEL NUMERO DI SOTTOSIEPI $Z \in Tl''_S$ INTERSECATI DA $X \in Tl'_S$;


 Π'_S

 Π''_S

$$\Theta(\Pi'_S, \Pi''_S) = 0$$


 Π'_S

 Π''_S

$$\Theta(\Pi'_S, \Pi''_S) > 0 \quad \text{POICHÉ } \{A, B\} \cap \{B, C\} \neq \emptyset \quad \text{E } \{A, B\} \cap \{A\} \neq \emptyset.$$

- LE **PURITY DEPENDENCIES** (PUD) SONO RFD CHE SI BASANO SU UNA MISURA DI IMPUREZZA; SONO:

- IN UN'ISTANZA DI UNA RELAZIONE R ;

- $X, Y \subseteq R$;

- $\Pi_X \in \Pi_Y$, PARTIZIONI DI R INDOTTI DA X O DA Y ;

- ALLORA UNA PUD È DEFINITA COME:

- **D_{TRUE}**: $X_{eq} \xrightarrow{\Theta(\Pi_X, \Pi_Y) \leq \varepsilon} Y_{eq}$, DOVE:

- Θ È UNA FUNZIONE CONCAVA E SUB-ADITIVA CHE CALCOLA LA PIÙ GRANDE MISURA DI IMPURETA' DEI BLOCCHI DI Π_X RISPETTO A Π_Y , E VICEVERSA.
- PONENDO $\epsilon = 0$, ACCORDI OTTENIAMO LE FD CANONICHE, IN QUANDO NON PONENDO ESSERCI IMPURETA':

- PER OGNI VALORE X DI X E Y DI Y , LE TUPLE t T.c. $t[X] = X$ DOVENDO ESSERE INCLUSE NELL' INSIEME DI TUPLE t I.c. $t[Y] = Y$,
- SE ESISTESSERO TUPLE TALI CHE $t_1[X] = t_2[X]$, MA $t_1[Y] \neq t_2[Y]$, ALLORA CI SAREBBERE UN'UNIONE SOTTOINSIEME CHE INTERSECA QUELLO DELLE TUPLE CON $t[X] = x$, DUNQUE L'IMPURETA' SAREBBERE > 0 ;

ESEMPIO:

1-) NEL DATA SET CLINICO:

Produttore	Nome	Categoria	...	Ingrediente Attivo	Prescrizione
Angelini	Aulin	NSAID		Nimesulide	Si
Dompè	Oki	NSAID		Ketoprofen	Si
Lisapharma	Arfen	NSAID		Ibuprofen	Si
...					
Zambon It	Spidifen	NSAID		Ibuprofen	No

► vale la seguente PuD

$D_{TRUE}: \text{IngredienteAttivo}_{EQ}$

$$\xrightarrow{\theta(\pi_{\text{IngredienteAttivo}}, \pi_{\text{Prescrizione}}) \leq 0.09} \text{Prescrizione}_{EQ}$$

NUMERICAL DEPENDENCIES (NFD)

- LE **NUMERICAL DEPENDENCIES** (NFD) SONO RFD CHE RIASSANO SUL NUMERO DI POSSIBILI VALORI ASSUNZIBILI DA Y DATO UN CERTO VALORE DI X .
- IN SOSTANZA, SE $t_1[X] = t_2[X]$, ALLORA $t_1[Y]$ E $t_2[Y]$ POSSONO AVERE SOLO K POSSIBILI VALORI; FORMALMENTE:

- $D_{EQ} : X_{EQ} \xrightarrow{\text{cond}(X, Y) \leq k} Y_{EQ}$, DOVE:

- $\text{cond}(X, Y) \stackrel{\text{def}}{=} |\pi_Y(\sigma_{X=x}(n))|$, INDICA IL NUMERO DI POSSIBILI VALORI DI Y ,

ESEMPIO:

1-1 $D_{EQ} : \text{NEPANTO} \xrightarrow{\substack{X \\ \text{NEPANTO CHECK-IN}}} Y_{EQ} \xrightarrow{\text{cond}(X, Y) \leq 10} \#SITANZA_{EQ}$

AD OGNI NEPANTO POSSONO ESSERE ASSOCIATE AL PIÙ 10 SITANZE.

PARTIAL DETERMINATION (PD)

- UN **RFD PROBABILISTICA** O **PARTIAL DETERMINATION** È DEFINITA COME

- $D_{TRUE} : X_{EQ} \xrightarrow{P(X, Y) \leq 1 - \epsilon} Y_{EQ}$, DOVE:

- $P(X, Y)$ è la probabilità che, prese 2 tuple t_1, t_2 t.c. $t_1[X] = t_2[X]$ allora anche $t_1[Y] = t_2[Y]$:

$$P(X, Y) \stackrel{\text{def}}{=} P(t_1[Y] = t_2[Y] \mid t_1[X] = t_2[X])$$

- con $\varepsilon = 0$ ottengiamo le FD canoniche.

- ESEMPIO:

1) Possibili canoni succ' attirando note su
naming e nomi scelti potremmo far sì
che la FD canonica Note \rightarrow Prenutone
non valga, ma verrà quella canonica stessa

$$- D_{\text{note}} : \text{Note}_{\text{EQ}} \xrightarrow{P(\text{Note}, \text{Prenutone}) \geq 0.97} \text{Prenutone}_{\text{EQ}}$$

**CONSTRAINED
DEPENDENCIES (CD)**

- Le **CONSTRAINED DEPENDENCIES (CD)** riassano sull'extent attraverso la specifica di un insieme di tuple su cui il vincolo di ugualanza è valido.

- DATA UNA FAMIGLIA DI VINCOLI L , ED UN VINCOLO $c \in L$, UNA CD È DEFINITA COME

$$- D_c : X_{\text{EQ}} \xrightarrow{\Psi_{\text{un}(c)}} Y_{\text{EQ}}$$

- DA QUESTA DEFINIZIONE, PONENDO $D_c = D_{\text{STAG}} \cup D_{\text{TIENI}}$ OTTENIAMO CHE
SIA CANONICHE.
- ESEMPIO:
- 1.) 2 RIGHE CON LA STESSA ESPERIENZA, DATA UNA
SPECIFICAZIONE IN PENATINA, DEVONO AVERE LO STESJO
VALORE DI STIPENDIO, SE LA LORO ESPERIENZA È ≥ 10 ANNI:

- $D_c : \text{ESPERIENZA}_{EQ} \xrightarrow{\Psi_{\text{STIPENDIO}}} \text{STIPENDIO}_{EQ}$

- $c = \left\{ t \in D : t[\text{ESPERIENZA}] \geq 10 \text{ anni} \wedge t[\text{SPECIFICAZIONE}] = "PENATINA" \right\}$

CONDITIONAL FUNCTIONAL DEPENDENCIES (CFD)

- LE CONDITIONAL FUNCTIONAL DEPENDENCIES (CFD) SONO SIMILI
ALLE CD, MA CON LA RESTRIZIONE CHE IL PREDICATO DI APPARTENZA
ALL'INSIEME DI TUPLE PER CUI LA DIPENDENZA VALE È FORMATO
SOLO DA UGUALIANZE.
- QUESTE SONO SPECIFICATE CON UN PATTERN TABLEAU, UNA
TABEGLIA CHE CONTIENE GLI ATTRIBUTI $A \in R$, ED I CUI VALORI
SONO:
 - VALORI DEL DOMINIO DI A CHE RESTRINCONO LE TUPLE
SU CUI LA DIPENDENZA È VERA;
 - UN SIMBOLO “-” CHE INDICA L'ASSUNZIONE DI RESTRIZIONI;

- IL PATTERN TABLEAU POSSONO USARE SOLO L'UGUAGLIANZA PER SPECIFICARE LE TUPLE, MA ESISTONO ESTENSIONI DELLE CFD, COME LE **[RCFD]**, CHE POSSONO SPECIFICARE ANCHE VINCOLI CON OPERAZIONI DI CONFRONTO ($<$, \leq , \geq , $>$, \neq).
- FORMALMENTE, DATO UN PATTERN TABLEAU T_n , UNA CFD È DEFINITA COME:

$$- D_{T_n} : X_{EQ} \xrightarrow{\psi_{n(0)}} Y_{EQ}$$

ESEMPIO

1-1 CONSIDERATO IL PATTERN TABLEAU PRECEDENTE, LA CFD SULL'ESPERIENZA E SU STIPENDIO PUÒ ESSERE ESPRESSA COME

$$- D_{T_n} : \text{ESPERIENZA}_{EQ} \xrightarrow{\psi_{n(0)}} \text{STIPENDIO}, \text{ CON:}$$

$$- T_n = \frac{\text{SPECIALIZZAZIONE}}{\text{PERCORSO}} \mid \frac{\text{ESPERIENZA}}{-} \mid \frac{\text{STIPENDIO}}{-}$$

RFD CHE RILASSANO
SUL CONFRONTO

- L'OBBIETTIVO DELLE RFD CHE RILASSANO SUL CONFRONTO È QUELLO DI ESPRIMERE CHE COINVOLGANO GRUPPI DI VALORI SIMILI PIUTTOSTO CHE IDENTICI; ESSE SONO DEFINITE COME:

$$D_{\text{func}} : X_{\phi_1} \xrightarrow{\Psi_{\text{fun}(0)}} Y_{\phi_2}, \text{ DOVE:}$$

- ϕ_1 E ϕ_2 SONO INSIEMI DI VINCOLI CHE ESPRIMONO PRESCRIZIONI SU UNA RELAZIONE D'ORDINE TRA GLI ELEMENTI DI $D(X)$ E $D(Y)$, OPPURE SUL VALORE DI UNA FUNZIONE DI DISTANZA O SIMILARITÀ SULLO STESSO DATARIO.

METRICAL FUNCTIONAL
DEPENDENCIES (MFD)

- LE METRICAL FUNCTIONAL DEPENDENCIES (MFD) AMMETTONO CONFRONTI NELLA PARTE DESTRA DELLA DIPENDENZA CHE TOLLERANO PICCOLE DIFFERENZE, SULLA BASE DI UNA FUNZIONE DI DISTANZA:

$$\phi : D(Y) \times D(Y) \longrightarrow \mathbb{R}$$

- SIA $\Delta_\phi(V)$ LA FUNZIONE CHE CALCOLA LA MASSIMA DISTANZA TRA GLI ELEMENTI DI UN INSIEME DI TUPLE V.

- DATA UNA SOGLIA DI TOLLENZA ϵ , UNA MFD VIENE DEFINITA COME:

$$D_{\text{func}} : X_{\text{eq}} \xrightarrow{\Psi_{\text{fun}(0)}} Y_{\text{max}} \Delta_\phi(S[Y]) \leq \epsilon, \text{ DOVE:}$$

- $\max_{S \in \overline{D}_X} (\Delta_\phi(S[Y]))$ INDICA LA MASSIMA DISTANZA TRA I VALORI DI $t_1[Y] \in t_2[Y]$, SAPENDO CHE $t_1[X] = t_2[X]$

- CON $\varepsilon = 0$ SI OTTENGONO LE FG CANONICHE.

- ESEMPIO:

1-1 Immaginiamo che la definizione di stipendio
tra i settori con uguali specializzazioni sia
essere max solo se

$$\begin{array}{c} X \\ \parallel \\ Y \end{array} \quad \text{Dunque: } \left\{ \text{SPECIALIZZAZIONE, ESPERIENZA} \right\} \xrightarrow{\Psi_{\text{funz}}}$$
$$\text{STIPENDIO}_{\max_{\text{SETT}}} \Delta_{\phi}(S[\gamma]) \leq 5000$$

NEIGHBOURHOOD
DEPENDENCIES (ND)

• LE NEIGHBOURHOOD DEPENDENCIES (NG) SFUCCANO IL CONCETTO DI CLOSELESS FUNCTION, UNA FUNZIONE CHE, PER OGNI ATTRIBUTO A E COPPIA DI TUPLE t_1, t_2 :

$$-\quad \Theta_A \in [0, 1] \stackrel{\text{def}}{=} \text{SIMILANZA DI } t_1[A] \in t_2[A]$$

• DATA UNA SOGLIA $\alpha \in [0, 1]$, IL PREDICATO DI VICINANZA SI DEFINISCE COME:

$$-\quad A^\alpha \stackrel{\text{def}}{=} \Theta_A \geq \alpha, \text{ CHE RICHIEDE UNA SIMILANZA TM } t_1[A] \in t_2[A] \text{ SUPERIORE AD } \alpha.$$

• DATE $\Theta_{A_1}, \Theta_{A_2}, \dots, \Theta_{A_m} \in \Theta_{\beta_1}, \Theta_{\beta_2}, \dots, \Theta_{\beta_m}$ CLOSELESS FUNCTION PER GLI ATTRIBUTI DI $X \in Y$ RISPETTIVAMENTE, CON SOGLIE $\alpha_1, \alpha_2, \dots, \alpha_m \in \beta_1, \beta_2, \dots, \beta_m$, UNA ND È DEFINITA COME:

$$- D_{TRUE} : X_{(\theta_{A_1} \geq \alpha_1 \wedge \dots \wedge \theta_{A_m} \geq \alpha_m)} \xrightarrow{\Psi_{true}(d)} Y_{(\theta_{B_1} \geq \beta_1 \wedge \dots \wedge \theta_{B_m} \geq \beta_m)}$$

ESEMPPIO:

1-) SI PUO' INDIVIDUARE PATIENTI CON ETÀ SICURE E SINTOMI SIMILI SIANO Ricoverati IN REPARTI SICILI:

$$- D_{TRUE} : \{ \text{SINTOMI}, \text{ETÀ} \}_{(\theta_{\text{SINTOMI}} \geq 0.85 \wedge \theta_{\text{ETÀ}} \geq 0.8)} \xrightarrow{\Psi_{true}(d)} \text{REPARTO}_{(\theta_{\text{REPARTO}} \geq 0.8)}$$

SIMILARITY FUNCTIONAL
DEPENDENCIES (SFD)

• LE **SIMILARITY DEPENDENCIES** (SFD) SONO RFD CHE RILASSANO SUL CONFRONTO SFUZZITANDO LE **RELAZIONI DI TOLLENZA**, UNA RELAZIONE DI EQUIVALENZA DEFINITA COME SEGUO:

$$- [t_1, \theta_A t_2 \stackrel{def}{\iff} |t_1[A] - t_2[A]| \leq \varepsilon], \text{ PER UN ATTRIBUTO } A.$$

• PER UN INSERTE X DI ATTRIBUTI INVECE:

$$- t_1 \theta_X t_2 \stackrel{def}{\iff} \forall A \in X, t_1 \theta_A t_2.$$

• A QUESTO PUNTO, UNA SFD E' DEFINITA COME:

$$- D_{true} : X_{\theta_X} \xrightarrow{\Psi_{true}(d)} Y_{\theta_Y}, \text{ CHE ESPRIME OLTRE:}$$

- $\forall t_1, t_2 \in M, t_1 \Theta_X t_2 \Rightarrow t_1 \Theta_Y t_2$, Dunque:
- $\epsilon = O \Rightarrow FD$ canoniche.

• ESEMPIO:

1-) Su dataset:

ID	Nome	Specializzazione	...	Stipendio	Tasse
1	George Johnson	Neurologia		\$218,000	\$62,500
2	Joe House	Cardiologia		\$314,000	\$94,200
3	Derek Williams	Pediatria		\$156,000	\$39,500
4	Henry Jones	Neurologia		\$222,000	\$63,000
5	Robert White	Pediatria		\$156,000	\$39,500
...					

POTERESSERE VALERE LA SFID:

- $D_{true}: Stipendio \Theta_{Stipendio} \xrightarrow{\text{Lem(0)}} Tasse \Theta_{Tasse}$

**HATCHING
DEPENDENCIES (HD)**

- LE **HATCHING DEPENDENCIES** (HD) HANNO LO SCOPO DI MODELLARE VINCOLI DI INTEGRITÀ REFERENZIALE RIASSATTI TRA SCHEMI DISTINTI.
- UNA HD CONFRONTA 2 RELAZIONI (R_1, R_2) , CON RISPETTIVI DOMINI $D_1 \in D_2$, ED È DEFINITA COME:

- $D_1 \times D_2 : (X_1, X_2) \xrightarrow{\text{Lem(0)}} (Y_1, Y_2) \rightleftharpoons$, Dunque:

- $X_1 = \{A_1, A_2, \dots, A_m\}, Y_1 = \{E_1, \dots, E_m\} \subseteq R_1$,
- $X_2 = \{B_1, B_2, \dots, B_m\}, Y_2 = \{F_1, \dots, F_m\} \subseteq R_2$,
- \approx è un predicato di confronto approssimato con una soglia ϵ di similarità su ciascun dominio degli attributi di $X_1 \in X_2$,
- \Rightarrow è un operazione di matching, che afferma che il valore di $t_1[Y_1]$ può essere trasformato in $t_2[Y_2]$ sostituendo un numero $\leq \epsilon$ di caratteristiche.

ESEMPIO:

1) DENTRATA \times DELL'ESAME:

$$((\text{NOTE}, \text{DATA NASCITA}), \{\text{CLIENTE}, \text{DATA NASCITA}\}) \underset{\approx}{\underset{\Phi_{\text{match}}}{\rightarrow}} (\{\text{INDIRIZZO PAZIENTE}, \text{INDIRIZZO CLIENTE}\}) \Rightarrow$$

► Consideriamo la relazione paziente di un DB medico

SIN	Nome	DataDiNascita	Sesso	...	Indirizzo
087-34-7789	Andrea White	1935-03-14	F		987 Jefferson, NV
087-11-3455	Mary Brown	1930-08-31	F		55 Fifth AV, NV
089-65-3325	Bill Mc Gregor	1970-12-21	M		100 Canal ST, NJ
...					

► e la relazione cliente di un DB di assicurazioni mediche

#Polizza	Cliente	DataNascita	...	Domicilio	Premio
35677651	Andreas White	03-14-1935		987 Jefferson AV, NV	\$2,400
35677712	M. Brown	08-31-1930		55 Fifth AV, NV	\$2,900
35677754	B. Gregor	12-21-1970		100 Canal Street, NJ	\$1,000
...					

COMPATIBLE DEPENDENCIES (C_oD)

- LE COMPATIBLE DEPENDENCIES (C_oD) SONO UNA GENERALIZZAZIONE DELLE ND E DELLE NFD, APPLICANDOLE AI DATASPACE ETERogenei.

- UTILIZZANO UN'OPERAZIONE DI CONFRONTO TRA DUE ATTRIBUTI $A_i, A_j \in S$ (IL DATASPACE):

$$- \boxed{t_1[A_i] \leftrightarrow_{ij} t_2[A_j]} \stackrel{\text{def}}{\iff} \begin{cases} t_1[A_i] = t_2[A_j] \circ \\ t_1[A_i] \approx t_2[A_j] \circ, \text{ SELETTI} \\ t_1[A_i] \Rightarrow t_2[A_j] \end{cases}$$

- UN'OPERAZIONE DI CONFRONTO APPROSSIMATO ≈ CON UNA SOGLIA E È UN'OPERAZIONE DI MATCH \Rightarrow ;

- DEFINITO QUINDI LA FUNZIONE DI CONFRONTO IL PREDICATO:

$$- \boxed{\Theta(A_i, A_j)} \stackrel{\text{def}}{=}$$

$$- \boxed{t_1[A_i] \leftrightarrow_{ij} t_2[A_j] \vee t_1[A_i] \leftrightarrow_{ii} t_1[A_i] \vee t_1[A_j] \leftrightarrow_{jj} t_2[A_j] \vee t_1[A_j] \leftrightarrow_{ji} t_2[A_i]}$$

- MENTO, DATI 2 INSIEMI DI ATTRIBUTI $X_1 \in X_2$:

$$- \boxed{\Theta(X_1, X_2)} \stackrel{\text{def}}{=} \bigwedge_{A_i \in X_1, A_j \in X_2} \Theta(A_i, A_j)$$

- DUNQUE, DATO UN INSIEME DI ATTRIBUTI, X_1 E X_2 , CONFRONTA LE TUPLE SU TUTTE LE COPPIE DI ATTRIBUTI CON L'OPERAZIONE DI CONFRONTO E RESTITUISCE L'AND TRA TUTTI I RISULTATI.
- A QUESTO PUNTO, DATO UN DATA SPACE ETERogeneo S , DEFINITO UNA COD COME:

$$- D_{true} : (X_1, X_2) \Theta(X_1, X_2) \xrightarrow{\Psi_{true}(0)} (Y_1, Y_2) \Theta(Y_1, Y_2)$$

, DOVE:

- $\Theta(X_1, X_2) \in \Theta(Y_1, Y_2)$ SONO FUNZIONI DI CONFRONTO.

ESEMPIO:

$$1) D_{true} : (STANZA, LETTO) \Theta(STANZA, LETTO) \xrightarrow{\Psi_{true}(0)}$$

$$(SESSO, GENERE) \Theta(SESSO, GENERE)$$

LA SEMANTICA E' CHE STABILITE STANZA E LETTO SONO DETERMINATI ANCHE SESSO E GENERE (IN SICHI DIFFERENTI).

DIFFERENTIAL
DEPENDENCIES (DD)

- LE DIFFERENTIAL DEPENDENCIES (DD) SONO UN ALTRO TIPO DI AFD CHE PERMETTE DI ESCLUDERE VINCOLI IN BASE AD UNA FUNZIONE DI DISTANZA PARTICOLARE.

- UNA FUNZIONE DI DIFFERENZA $\phi[\beta]$ ESPRIME UN VINCOLO SULLA DIFFERENZA TRA $t_1[\beta]$ E $t_2[\beta]$:

$$- \quad (t_1, t_2) \simeq \phi[\beta] \stackrel{\text{def}}{\iff} t_1[\beta] \in t_2[\beta] \text{ SONO ISPIRANTI IL VINCOLO}$$

- I VINCOLI POSSONO ESSERE SPECIFICATI CON QUALSIASI OPERAZIONE DI COMPARAZIONE ($=, \neq, >, \geq, <, \leq$), E SONO ASSOCIAZI A UNA SOGLIA ε .

- DEFINITA A QUESTO PUNTO UNA DD COTIE:

$$- D_{\text{nucl}} : X_{\phi_L} \xrightarrow{\psi_{\text{nucl}}} Y_{\phi_M}, \text{ DOVE:}$$

- ϕ_L E ϕ_M SONO FUNZIONI DI DIFFERENZA DEFINITE SUGLI ATTRIBUTI DI X ED Y

- LA DIPENDENZA STA AD INDICARE CHE SE $(t_1, t_2) \simeq \phi_L[X]$, ALLORA $(t_1, t_2) \simeq \phi_M[Y]$.

- ESEMPIO:

$$1) D_{\text{nucl}} : \text{DATA PRESCRIZIONE } \phi_L \xrightarrow{\psi_{\text{nucl}}} \text{DATA ESIGUIZIONE } \phi_M$$

DOVE PER ϕ_L , $\varepsilon = 0$, QUINDI DEVONO ESSERE VULCANI PER ϕ_M $\varepsilon \leq 5$.

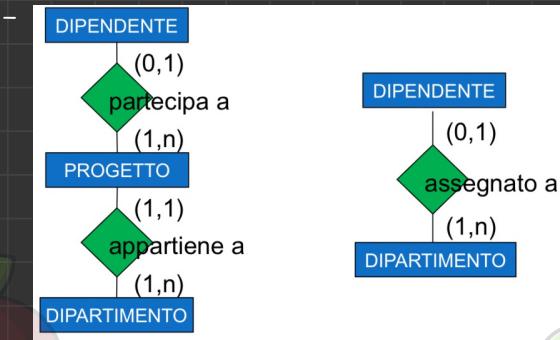
DATA INTEGRATION

DATA INTEGRATION:
INTRODUZIONE

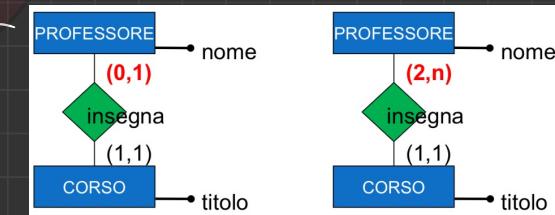
- LA DATA INTEGRATION È UNA BRANCIA DELLA DATA SCIENCE CHE SI OCCUPA DI INTEGRARE I DATI PROVENIENTI DA FONTI DIFFERENTI ED ETEROGENEE IN UN UNICO SCHEMA UNIFORME ED INTERROGANILE.
- LE SORGENTI DI INFORMAZIONE SONO SPESO:
 - DISTINTE;
 - AUTONOME;
 - ETEROGENEE;
- NONOSTANTE CO' LO SCHEMA RISULTANTE DEVE ESSERE UNIFORME IN TUTTO E PER TUTTO, E DOVE RISULTARE INTERROGANILE E GESTIONE IN MODO TRANSPARENTE.
- L'ETEROGENEITÀ NEGLI SCHEMI DI INTEGRARE PUO PORTARE A DIVERSE PROBLEMATICHE:
 - PROSPETTIVA DI PROGETTAZIONE DIFFERENTE;

- USO DI COSTRUTTI DI MODELLO DIFFERENTI;
- INCOMPATIBILITÀ DELLE SPECIFICHE;
- ESEMPIO:

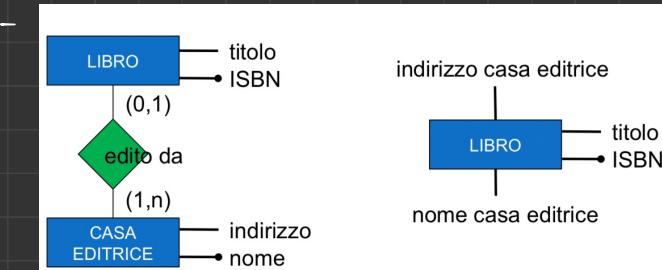
1-) DIVERSITÀ DI PROSPETTIVA:



2-) INCOMPATIBILITÀ DELLE SPECIFICHE:



3-) USO DI COSTRUTTI EQUIVALENTI:



CONCETTI COMUNI

- QUANDO 2 RELAZIONI R_1, R_2 PROVENIENTI DA SCHEMI DI OGNI DIVERSI SONO STATE CREATE PER MODELLENNO UNO STESSO CONCETTO, POSSONO VERIFICARSI DIVERSE SITUAZIONI:

- **IDENTITÀ**:

- R_1 ED R_2 SONO CREATE USANDO GLI STESSI COSTRUTTI, DUNQUE SONO PERFETTAMENTE UGUALI:

$$- R_1 = R_2$$

- **EQUIVALENZA**:

- R_1 ED R_2 SONO CREATE CON COSTRUTTI DIVERSI, HA È POSSIBILE TROVARE UNA CORRISPONDENZA BIETIVA TRA LE ISTANZE DI R_1 ED R_2

- ESEMPIO:

1-)



LIBRO			CASA EDITRICE	
ISBN	titolo	casa editrice	nome	indirizzo
123445	Il DFM	McGraw-Hill	McGraw-Hill	Via Ripamonti, 89
435454	Mi sembra logico	Apogeo	Apogeo	Via Verdi, 45
...

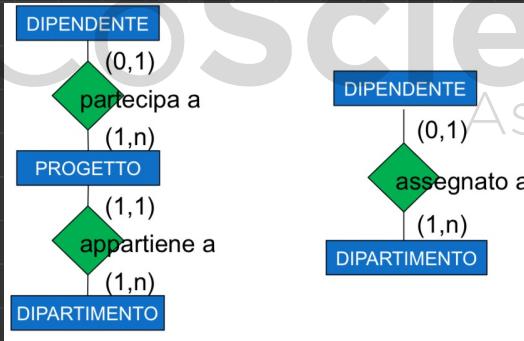
LIBRO				
ISBN	titolo	nome c.e.	Indirizzo c.e.	
123445	Il DFM	McGraw-Hill	Via Ripamonti, 89	
435454	Mi sembra logico	Apogeo	Via Verdi, 45	
...	

• **COTERMINALITÀ** :

- R_1 E R_2 SONO CREATE CON COSTRUTTI DIVERSI E NON EQUIVALENTI, MA CHE NON SONO IN CONTRASTO.

• ESEMPIO :

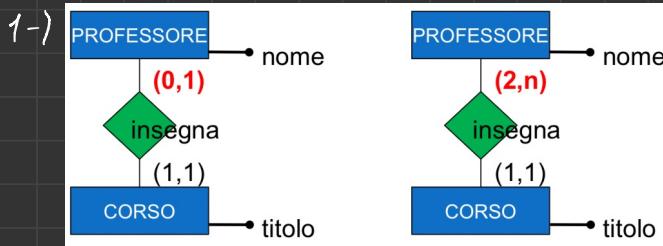
1-)



• **INCOMPATIBILITÀ** :

- R_1 E R_2 SONO CREATE CON COSTRUTTI DIVERSI E NON EQUIVALENTE, CHE SONO IN CONTRASTO A CAUSA DI UN CONFLITTO NELLE SPECIFICHE DELLO SCHEMA.

• ESEMPIO:



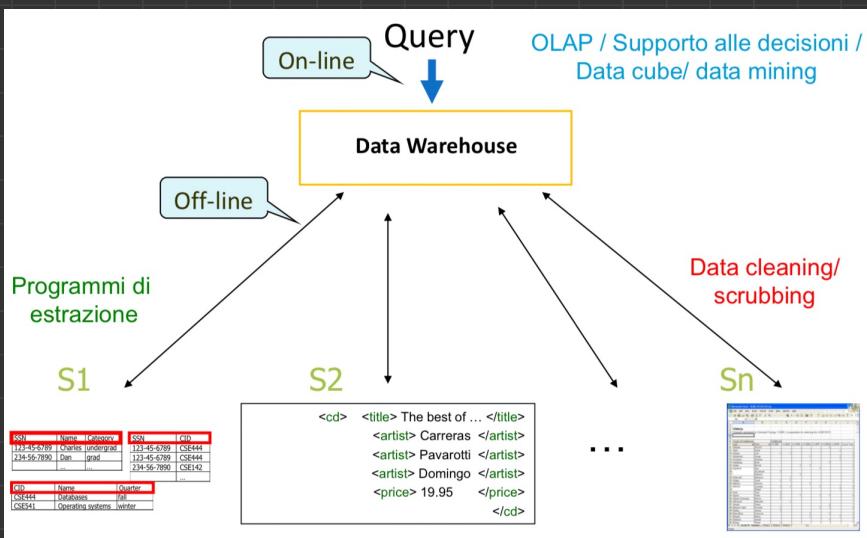
ARCHITETTURE DI
INTEGRAZIONE

- ESISTONO PRINCIPALMENTE 2 ARCHITETTURE DI INTEGRAZIONE DEI DATI:

- DATA WAREHOUSE;
- DATABASE VIRTUALE

[DATA WAREHOUSE]:

- I DATI INTEGRATI PROVENIENTI DA VERSI FONTI SONO CARICATI PERIODICAMENTE IN UN DATABASE MATERIALIZED, COMPOSTO DA FILE INDIPENDENTI DALLE FONTI ORIGINALI.
- HANNO LA CARATTERISTICA DI SEPARARE I DATI NESSI A DISPOSIZIONE PER L'INTERROGAZIONE DA QUELLI PER LE ANALISI STATISTICHE, IN FILE DIFFERENTI.
- PERIODICAMENTE, I DATI VENGONO AGGIORNATI.



VANTAGGI :

- SEPARA I DATI PER L'INTERROGAZIONE DAI DATI PER LE ANALISI STATISTICHE, PORTANDO AD UNA MAGGIOR EFFICIENZA.
- NON RICHIESTE REFRESH CONTINUI.

SVANTAGGI :

- I DATI NON SONO SEMPRE AGGIORNATI.

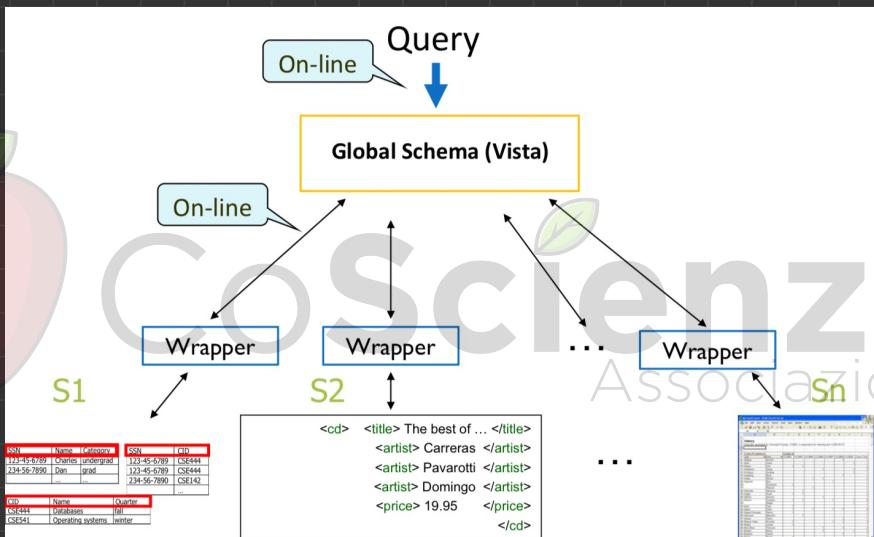
DATABASE VIRTUALE :

- I DATI SONO LASCIATI NELLE SORGENTI ORIGINALI E LO SCHEMA INTERROGABILE È RAPPRESENTATO DA UNA VISTA VIRTUALE CHE PERMETTE LE OPERAZIONI IN modo TRANSPARENTE.

- QUERY E OPERAZIONI SONO GESTITE TRAMITE DEI **WRAPPER**, CHE:

(i) TRADUCANO LA QUERY SULLO SCHEMA GLOBALE VIRTUALE IN UN INSIEME DI QUERY SUGLI SCHEMI ORIGINALI;

(ii) TRADUCANO I RISULTATI DELLE QUERY NEL FORMATO DELLO SCHEMA GLOBALE;



- **VANTAGGI**:

• I DATI SONO COSTANTEMENTE AGGIORNATI.

- **SVANTAGGI**:

• RICHIEDE CONTINUE TRADUZIONI DI QUERY E RISULTATI DALLO SCHEMA GLOBALE A QUELLI ORIGINALI.

- TRANCATA SEPARAZIONE DEI DATI PER LE QUOTIDIANI DATI PER LE ANALISI STATISTICHE.

ESI DELL' INTEGRAZIONE

- DA UN PUNTO DI VISTA LOGICO, L'INTEGRAZIONE PUÒ ESSERE SVOLGIBILE IN 3 FASI:

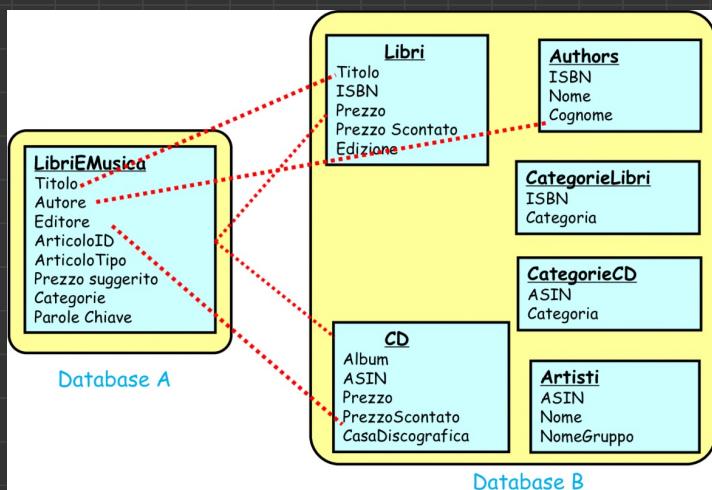
• SCHEMA MATCHING :

- SE CIASCUNO SCHEMA DA INTEGRARE HA NELLA CONCESSIONE DISTINTI DEL MONDO REALE, IL PROBLEMA DELL'INTEGRAZIONE NON SUSTIENE.

- LO SCHEMA MATCHING È LA FASE IN CUI SI INCONTRANO I CONCETTI CORRELATI ALL'INTERNO DEI DIVERSI SCHEMI DA INTEGRARE.

• ESEMPIO:

1-)



- ANALISI DEI CONFLITTI

- CONFLITTI DI ETEND GENEITÀ

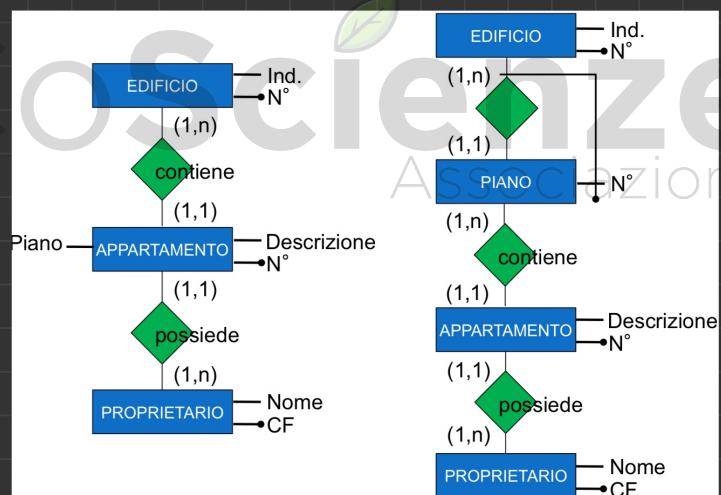
- DISCREPANZE NELL'USO DI FORMALISTI CON POTERE ESSESSIVO DIVERSO.

- CONFLITTI SEMANTICI

- PRESENZA DI SCHEMI SOAGENTE CHE INCORRANO LO STESSO CONCETTO, MA CON DIVERSI LIVELLI DI ABSTRAZIONE.

- ESEMPIO:

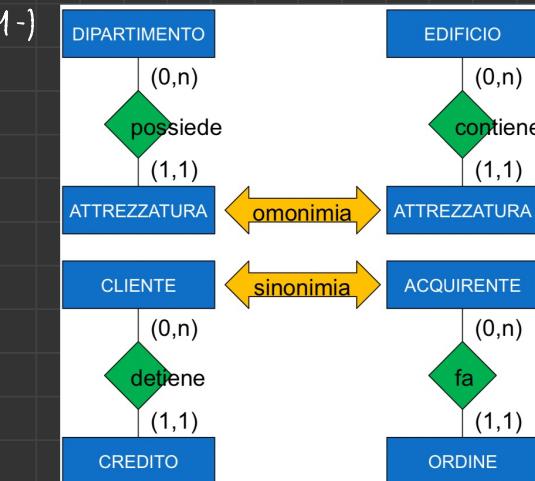
1-)



- CONFLITTI SUI NOMI

- PRESENZA DI OMOINOMIE, SINONIMIE O USO DELLO STESSO NOME PER INDICARE CONCETTI DISTINTI.

• ESEMPIO:



• STRUTTURALI:

- **TIPO**: COSTRUTTI DEL LINGUAGGIO DIVISI, A LIVELLO DI ENITÀ O ATTRIBUTO PER MODELLARE LO STESSO CONCETTO.

- **DIPENDENZA**: MODELLAZIONE DELLO STESSO CONCETTO, MA CON RELAZIONI, DIPENDENZE O VINCOLI DIFFERENTI.

- **CHIAVE**: CHIAVI DIFFERENTI TRA GLI SCHEMI CHE MODELLANO LO STESSO CONCETTO.

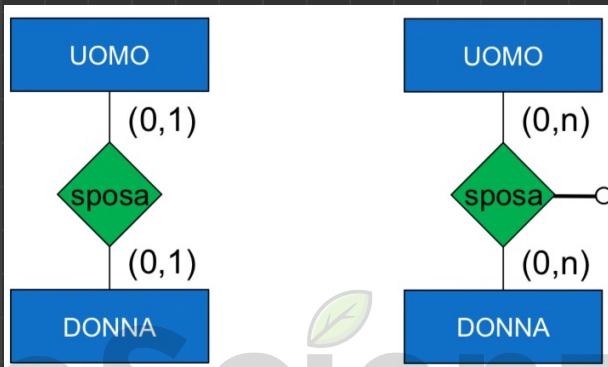
- **CONTRAMENTO**: DIFFERENTI VINCOLI, POLITICHE DI INSERT/UPDATE/DELETE IN SCHEMI CHE MODELLANO LO STESSO CONCETTO.

• ESERCIZIO:

1-) CONFLITTO DI TIPO:

- ATTRIBUTO SESSO: IN UNO SCHERMO
0,1 IN UN AUTOMATICO

2-) CONFLITTO DI DIPENDENZA:



- **INTEGRAZIONE DELLE SORGENTI**.

(i) RISOLUZIONE DEI CONFLITTI INDIVIDUATI;

(ii) PRODUZIONE DI UN NUOVO SCHERMO INTEGRATO CHE, PER QUANTO POSSIBILE, ESPRIMA LA STESSA SEMANTICA DELLE SORGENTI;

(iii) CREAZIONE DI UN MAPPING TRA LO SCHERMO
ORIGINALE E QUELLO INTEGRATO PER OGNI
CONCEPITO E/O RELAZIONE, IN 2 POSSIBILI MIGLI

- **GLOBAL AS VIEW**: LO SCHEMA GLOBALE INTEGRATO È CREATO IN FUNZIONE DI QUELLI SORGENTE LOCALI;
- **Local as View**: GLI SCHEMI GLOBALE È ESPRESSO IN modo INDEPENDENTE DA QUELLI LOCALI.

- **GLOBAL AS VIEW (Gav)**:

- NELLA MODALITÀ Gav LO SCHEMA GLOBALE È INTEGRATO, E' CREATO COME VISTA (VIRTUALE O MATERIALIZZATA), DEFINITA IN FUNZIONE NEGLI SCHEMI SORGENTE.
- OGNI CONCETTO DELLO SCHEMA INTEGRATO È UNA VISTA SUI CONCETTI NEGLI SCHEMI SORGENTE LOCALI.
- AD OGNI QUERY SULLO SCHEMA GLOBALE, BASTA SOSTituIRE I CONCETTI DELLO SCHEMA GLOBALE CON LE FUNZIONI DELLO SCHEMA LOCALE A CUI CORRISPONDONO.

- **VANTAGGI**:

- LA FASE DI UNFOLDING DELLE QUERI GLOBALE IN QUERI LOCALI È MOLTO SEMPLIFICATA.

• **SVANIAGGI** :

- RISULTA DIFFICILE DA ESPANDERE, POICHÉ ASSUMERE UNA SORGENTE IMPLICA LA RIDIFINIZIONE DI TUTTI I CONCETTI.

• **ESEMPIO** :

```
1-) // DB1 Magazzino  
ORDINI2011(chiaveO, chiaveC, data ordine, impiegato)  
CLIENTE(chiaveC, nome, indirizzo, città, regione, stato)  
  
// DB2 Amministrazione  
CLIENTE(chiaveC, piva, nome, tel, fatturato)  
FATTURE(chiaveF, data, chiaveC, importo, iva)  
STORICOORDINI2010(chiaveO, chiaveC, data ordine, impiegato)  
  
CREATE VIEW CLIENTE AS  
SELECT CL1.chiaveC, CL1.nome, CL1.indirizzo, CL1.città, CL1.regione,  
CL1.stato, CL2.tel, CL2.fatturato  
FROM DB1.CLIENTE AS CL1, DB2.CLIENTE AS CL2  
WHERE CL1.chiaveC = CL2.chiaveC;  
  
CREATE VIEW ORDINI AS  
SELECT * FROM DB1.ORDINI2011  
UNION  
SELECT * FROM DB2.STORICOORDINI2010;
```

• **LOCAL AS VIEW (Lav)** :

- NELLA MODALITÀ **Lav** LO SCHEMA GLOBALE È ESPRESSO IN FORMA INDEPENDENTE DALLE SORGENTI.
- OGNI CONCETTO DELLO SCHEMA INTEGRATO È ESPRESSO SOLO IN FUNZIONE DI QUEST'ULTIMA.
- AD OGNI QUERY GLOBALE, È NECESSARIO MAPPARE I CONCETTI NELLO SCHEMA GLOBALE NEGLI SCHEMI SORGENTE (QUERY NEWLIVING).

• **VANTAGGI :**

- NUOVI SERVIZI DA ESPANDERE AGGIUNGENDO NUOVI CONCETTI E/O SORGENTI.

• **SVANTAGGI :**

- L'UNFOLDING DELLE QUERY GLOBALI È CONNESSO, RICHIENDENDO QUERY REWRITING.

• **ESEMPIO :**

```
1-) // DB Globale
ORDINI(chiaveO, chiaveC, data ordine, impiegato)
CLIENTE(chiaveC, piva, nome, indirizzo, città, regione, stato, tel,
fatturato)
.....
// DB1 Magazzino
CREATE VIEW CLIENTE AS
SELECT chiaveC, nome, indirizzo, città, regione, stato
FROM DB.CLIENTE;

CREATE VIEW ORDINI2011 AS
SELECT * FROM DB.ORDINI
WHERE data > '31/12/2010' and data < ``1/1/2012'';
```

MAP - REDUCE

MAP - REDUCE : INTRODUZIONE

- COME AVVISTATO GIÀ ACCENNATO IN PRECEDENZA, L'AVVENTO DEI BIG DATA HA POSTO DIVERSE SPINE AL MONDO DELLA CORRUZIONE:
 - LA RETEORIZZAZIONE CENTRALIZZATA NON È PIÙ POSSIBILE;
 - LA CORRUZIONE SU UN SINGOLO CALCOLATORE NON È PIÙ POSSIBILE;
- UNA POSSIBILE PRIMA SOLUZIONE È LA PROGRAMMAZIONE PARALLELA E CONCORSUALE SU RETI, MA:
 - COPIARE I DATI SU TUTTI I CALCOLATORI COINVOLTI RISULTA COMPLESSO;
 - SCRIVERE PROGRAMMI CONCORSUALI È PARTICOLARMENTE DIFFICILE;
- **MAP-REDUCE** È UN PARADIGMA DI CORRUZIONE E RETEORIZZAZIONE DISTRIBUITA CHE:
 - USA UN FILE SYSTEM DISTRIBUITO PER LA RETEORIZZAZIONE;

- SFRUTTA UN PARADIGMA DI COMPUTAZIONE INTRINSECAMENTE PARALLELO E FACILE DA USARE NEI PROBLEMI GIUSTI;

FILE SYSTEM
DISTRIBUITO

- IL FILE-SYSTEM DISTRIBUITO DI MAP-REDUCE È PROGETTATO PER GARANTIRE:

- UN SISTEMA DI NAMING GLOBALE;
 - FAULT-TOLERANCE;

- FREQUENTI INTERROGAZIONI E MODIFICHE RANE;

- ESSO È BASATO SU 3 COMPONENTI PRINCIPALI:

- **CHUNK-SCANNER**

- OGNI FILE È SVOLVISO IN CHUNK DA 16-64 kB REPLICATI 2 O 3 VOLTE;

- OGNI REPLICA È MANTENUTA IN UN NODO DIVERSO, DETTO CHUNK- SERVER.

- **MASTER-NODE**:

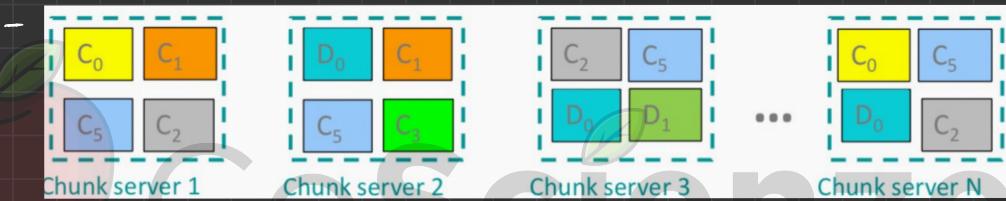
- GESTISCE I METADATI DEI FILE, IN PARTICOLARE LE INFRAZIONI SULLA POSIZIONE DEI CHUNK NELLA RETE;

- PUO' ESSERE REPLICATO A SUA VOLTA.

- **LIBERNIE CLIENT**:

- ESEGUVONO LA CONNESSIONE CON I MASTER-NODE ED I CHUNK-SERVER PER ACCEDERE AI FILE.

- LE COMPUTAZIONI SONO ESEGUITE SUGLI STESSI CHUNK-SERVER CHE MANTENGONO UN DETERMINATO CHUNK, IN TUO O NO NON DOVRA AFFRONTARE IL PROBLEMA DELLA COPIA DEI FILE:



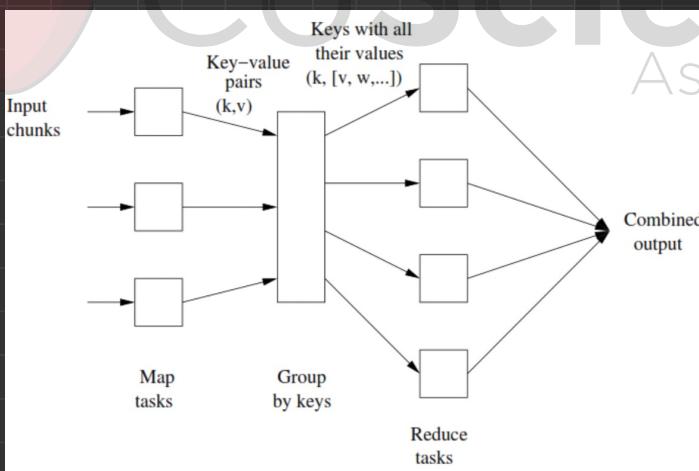
- IL PARADIGMA DI COMPUTAZIONE DI MAP-REDUCE PERMETTE DI ESEGUIRE COMPUTAZIONI PARALLELE SU GRANDI QUANTITA' DI DATI SPECIFICANDO SOLO 2 FUNZIONI:

- **MAP**: COME I DATI DA ELABORARE VENGONO SVISI E ELABORATI SU CIASCUN CHUNK-SERVER;

- **REDUCE**: COME I RISULTATI VENGONO COMBINATI CON UN'OPERAZIONE DI AGGREGAZIONE.

- AN ALTO LIVELLO, MAP-REDUCE OPERA CON SEGUENTI PASSI:

- (i) DOPO AVER ALLOCATO I TASK DI MAP E DI REDUCE AI VARI CHUNKS, I TASK DI MAP TRASFORMANO I RISPECTIVI CHUNKS ELABORANDO IN COPPIE CHIAVE-VALORE SECONDO LA LOGICA DI MAP SPECIFICATA;
- (ii) LE COPPIE CHIAVE-VALORE SONO ORDINATE PER CHIAVE DA UN MASTER CONTROLLER, CHE Poi LE SUDDIVIDE ANCHE TRA I VARI TASK REDUCE ALLOCATI;
- (iii) CIASCUN TASK DI REDUCE COMBINA I RISULTATI DI VARI TASK DI MAP (LE COPPIE CHIAVE-VALORE) SECONDO LA LOGICA DI REDUCE SPECIFICATA;



- ESEMPI:

- 1-) ANALIZZARE UN DOCUMENTO DI TESTO, E DETERMINARE CONTARE LE OCCORRENZE DI OGNI PAROLA.

• **MASTER CONTROLLER**:

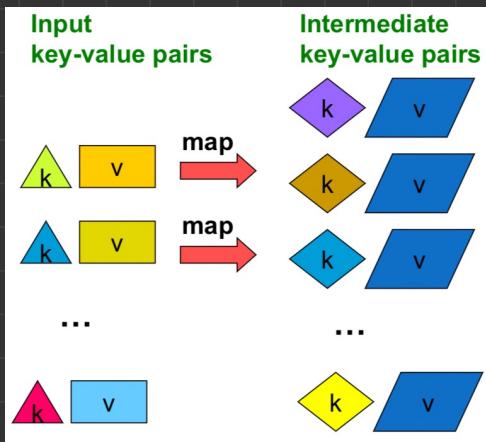
- IL MASTER CONTROLLER È UN PROCESSO CON IL COMITO DI ALLOCARE E CONTROLLARE IL WORKFLOW DEI TASK DI MAP E DI REDUCE:
 - STABILISCE IL NUMERO DI TASK MAP E REDUCE;
 - USA UNA FUNZIONE HASH h CHE MAPPÀ UNA CHIAVE NEL CORRISPONDENTI TASK DI REDUCE:

$$h : k \rightarrow \{0, 1, \dots, n-1\}$$

\uparrow \uparrow
CHIAVE TASK REDUCE

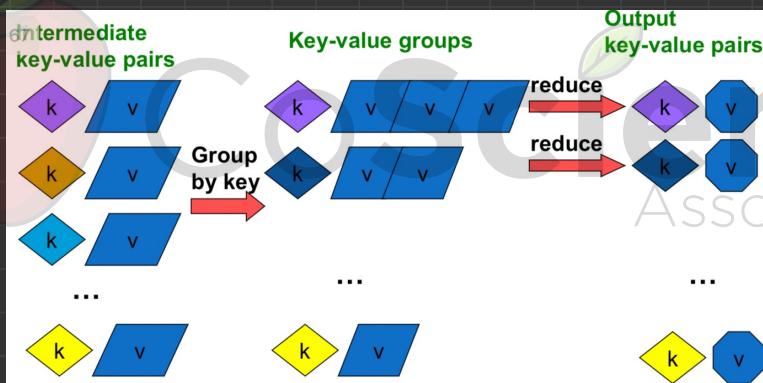
• **TASK MAP**:

- **INPUT**: UNA COPPIA CHIAVE-VALORE (k, v) (NOME DOCUMENTO);
- **OUTPUT**: UN INSIEME DI COPPIE CHIAVE-VALORE (k', v') ;



TASK REDUCE :

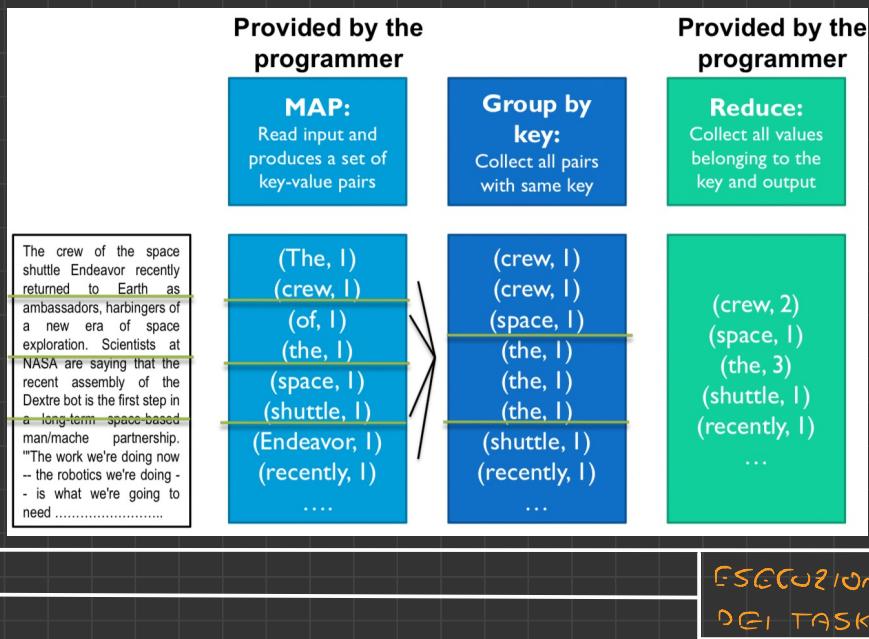
- **INPUT**: UN INSIEME DI COPPIE CHIAVE-VALORE $\langle k', v' \rangle$, DOVE $\langle v' \rangle$ È L'INSIEME DI VALORI ASSOCIAZI ALLA CHIAVE k' ;
- **OUTPUT**: UN INSIEME DI COPPIE CHIAVE-VALORE $\langle k', v'' \rangle$, DOVE v'' È UN VALORE AGGREGATO OTTENUTO DALLE COMBINAZIONI DEI VALORI $\langle v' \rangle$;
- TUTTI GLI OUTPUT DEI TASK REDUCE SONO Poi INSERITI IN UN UNICO FILE FINALE DI OUTPUT.



ESEMPIO :

```
map(key, value):  
1-) // key: document name; value: text of the document  
    for each word w in value:  
        emit(w, 1)
```

```
reduce(key, values):  
// key: a word; value: an iterator over counts  
    result = 0  
    for each count v in values:  
        result += v  
    emit(key, result)
```



ESECUZIONE
DEI TASK

• IL PROCESSO MASTER CONTROLLER :

- (i) ALLOCA UN WORKER AD OGNI UNITÀ DI ELABORAZIONE, CHE PUÒ ESEGUIRE UN TASK DI MAP O DI REDUCE;
- (ii) ASSEGNA 1 WORKER A CIASCUN TASK MAP E REDUCE, IN NUOVO TALE CHE AD OGNI CHUNK CORRISPONDA 1 TASK DI MAP;
- (iii) CREA UN FILE DI SCAMBIO PER OGNI TASK DI REDUCE, IN CUI È PRESENTE 1 COLONNA PER OGNI TASK DI MAP, CHE SCRIVE SU DI ESSA IL VALORE V' ASSOCIATO ALLA CHIAVE K' RELATIVA A TALE TASK;

- (iv) MONITORA CIASCUN TASK CON PUNTI PERIODICI, FACENDO VARIEGARE LO STATO DI TALI TASK TRA: "IDLE", "EXECUTING", "COMPLETED";

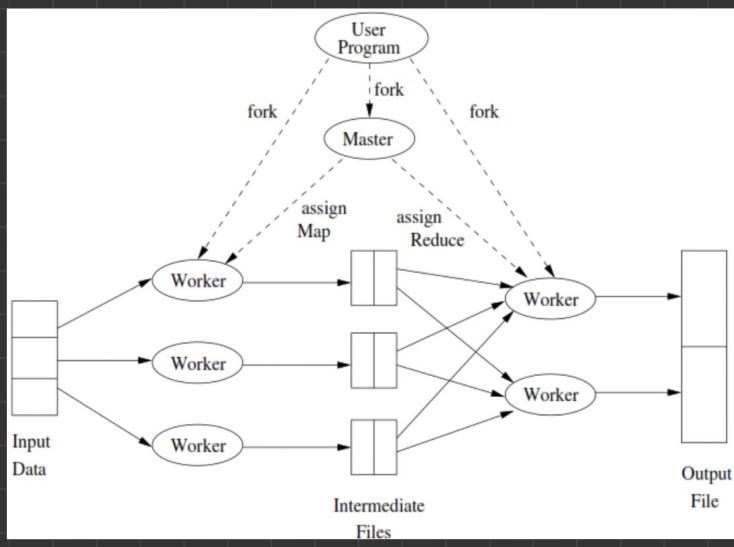
- SE IL MASTER CONTROLLER TROVA UNA FAILURE IN UN NODE CHE STA ESEGUENDO UN TASK DI MAP:

(i) I TASK DI MAP COINVOLTI SONO IMPOSTATI AD "IDLE" IN ATTESA DI ESSERE SCHEDULATI SU UN NUOVO WORKER SIA SE IN STATO "EXECUTING" CHE IN STATO "COMPLETED", POICHÉ GLI EVENTUALI RISULTATI SAREBBERO INACCESSIBILI;

(ii) INFORMA I TASK DI REDUCE COINVOLTI DEL CAMBIO DI WORKER;

- SE IL MASTER CONTROLLER TROVA UNA FAILURE IN UN NODE CHE STA ESEGUENDO UN TASK DI REDUCE:

(i) I TASK DI REDUCE COINVOLTI SONO IMPOSTATI AD "IDLE" IN ATTESA DI ESSERE SCHEDULATI SU UN NUOVO WORKER;



NUMERO DI TASK

- IL NUMERO DI TASK DI MAP DEV'ESSERE, IN GENERALE, MOLTO PIÙ ADO DI QUELLO DI NODI.
- INOLTRE, DATO CHE VIENE CREATO 1 FILE DI SCAMBIO PER OGNI FILE DI REDUCE ED 1 COLONNA IN ESSI PER OGNI TASK DI MAP, IL NUMERO DI TASK DI REDUCE DOVREBBE ESSERE < DI QUELLO DEI TASK DI MAP.
- LA MASSIMA PARALLELIZZAZIONE SI OTTIEDE CREANDO UN TASK DI REDUCE PER OGNI COPPIA:

- $(k', \langle v' \rangle)$, MA QUESTO PUÒ PORTARE A:

- FORTI ASIMMETRIE NELLA DISTRIBUZIONE DEL CARICO IN QUANTO ALCUNI $\langle v' \rangle$ POSSONO ESSERE TUTTO PIÙ GRANDI DI ALTRI;

- QUESTE INEGUAGLIANZE VENGONO RISOLTE:

- ALCANDO PIÙ TASK REDUCE DEL NUMERO DI NODI;

- RANDONIZZANDO L'ASSEGNAZIONE DELLE CHIAVI AI TASK DI REDUCE.

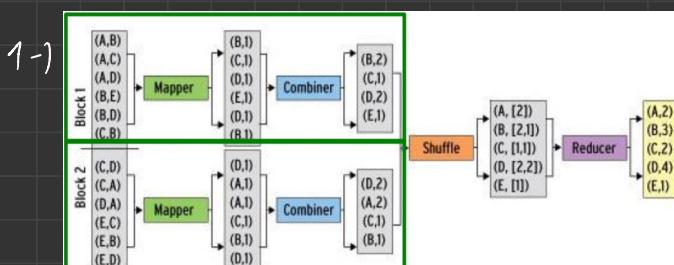
• **TASK DI BACKUP :**

- UN WORKER POTREBBE RISULTARE MOLTO LENTO A CAUSA DI PROBLEMI SULLA MACCHINA CHE LO ESEGUE.
- POSSONO ESSERE DUNQUE CREATE COPIE DELLO STESSO TASK, DETTE DI BACKUP, DELLE quali viene considerato il risultato solo del primo che termina.

• **COMBINERS :**

- SPESO UN TASK DI MAP PUÒ CREARE PIÙ COPIE DI VALORI ASSOCIATI ALLA STESSA CHIAVE $(k, v_1), (k, v_2), \dots, (k, v_m)$.
- PER SEMPLIFICARE IL TASK DI REDUCE, E RIDURRE L'USO DELLA RETE, POSSONO ESSERE USATI NEI COMBINER CHE COMBINANO GIÀ I RISULTATI NEL TASK DI MAP LOCALMENTE.
- PUÒ ESSERE APPLICATA SOLO SE L'OPERAZIONE EFFETTUATA DAL REDUCE È COMUTATIVA E ASSOCIAZIONALE.

• **ESEMPIO :**



DOCUMENT SIMILARITY

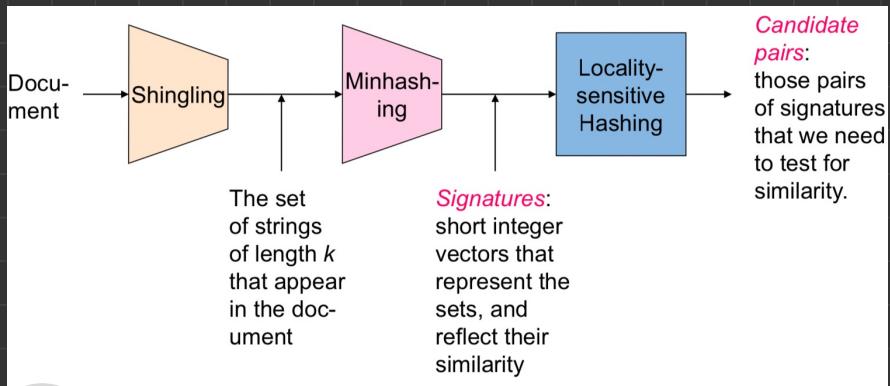
DOCUMENT SIMILARITY: INTRODUZIONE

- UN PROBLEMA COMUNE QUANDO SI LAVORA CON I BIG DATA E' QUELLA DI STABILIRE UNA MISURA DI SIMILARITA' TRA 2 DOCUMENTI, NOTA COME **DOCUMENT SIMILARITY**, ALLO SCORPO DI:
 - TROVARE DOPPLICATI / PLAGI;
 - TROVARE DOCUMENTI SIMILI CHE AFFRONTINO UNO STESSO ARGOMENTO;
- CI CONCERNEMENTO IN PARTICOLARE SULLA SIMILARITA' A LIVELLO DI PAROLE, Dunque CI INTERESSA CAPIRE SE 2 DOCUMENTI $A \in \beta$ SONO SIMILI, MA CON alcune DIFFERENZE (es. PLAGI).
- IL PROCESSO PER STABILIRE LA DOCUMENT SIMILARITY AVVIENE L'USO DI 3 TECNICHE:

(i) **SHINGLING**: CONVERSIONE DEI DOCUMENTI IN INSIGHI;

(ii) **HINHASHING**: CONVERSIONE DEGLI INSIEMI DEFINITI IN PAROLE CHE RINUNCANO LA DIMENSIONE DEGLI OGGETTI DA CONFRONTO MANTENENDO LE PROPRIETA' DI DISTANZA TRA DIESSI;

(iii) LOCAL-SENSITIVE HASHING: anziché confrontare tutte le fine, si confrontano solo quelle con una mappatura probabilistica di essere simili;



- Sia D un documento; un k -SHINGLE è una sequenza di k caratteri consecutivi in D .
- Un documento D è spesso rappresentato come l'insieme dei suoi k -SHINGLE:

$$- S_k = \{s : s \text{ è un } k\text{-SHINGLE su } D\}$$

- ESEMPIO:

1) $k=2$, $doc = abcabc$. L'insieme di 2-SHINGLE è:

$$- \{ab, bc, ca\}$$

• ESCAVAZIONI:

- CAMBIARE UNA PAROLA IN D HA EFFETTO SOLO SUL K -SHINGLE CHE SONO A DISTANZA DI $k-1$ CARATTERI DALLA PAROLA CAMBIATA.
- RIDORINARE I PARAGRAFI HA EFFETTO SOLO SUL 2 K -SHINGLE CHE SI TROVANO AL CONFINE TRA I 2 PARAGRAFI.
- DA QUESTE PROPRIETÀ, È CHIARO CHE CAMBIARE SINGOLE PAROLE O RIDORINARLE NON INFUENZA GLI SHINGLE, DUNQUE DOCUMENTI SIMILI \Rightarrow SHINGLE SIMILI.

• ESEMPIO:

1-1 "The dog which chased the cat" \rightarrow "The dog that
chased the cat"
SOLI 3 SHINGLE SONO CAMBIATI TRA LE 2 FRASI.

• SCELTA DI K:

- K TROPPO PICCOLO \Rightarrow MOLTI DEI K -SHINGLE GENERATI SONO PRESENTI IN TUTTI I DOCUMENTI, ANCHE SE NON SONO SIMILI;
- K TROPPO GRANDE \Rightarrow OGNI K -SHINGLE AMMAREGGIA PONTEGGI DI TESTO TROPPO GRANDI, DUNQUE ANCHE DOCUMENTI SIMILI AVRANNO K -SHINGLE DIVERSI;

- BISOGNA DUNQUE SEGUIRE K IN BASE AL TIPO DI DOCUMENTO, E' IN FONDO CHE I K-SHINGLE DI OGNI DOCUMENTO SIANO DISTINTI, MA NON ECESSIVAMENTE.

- SU UN ALFABETO Σ , I POSSIBILI K-SHINGLE SONO:

$$- |\Sigma|^k$$

- ESEMPIO:

1-) CON 27 CARATTENI, I POSSIBILI 5-SHINGLE SONO 27^5 .

2-) PER LE MAIL, K VENGUE SOLITAMENTE POSTO A 5, PER DOCUMENTI GENERALI $K=8$.

3-) IN ANTICOGLI DI CU, SI VUOLE VERIFICARE LA SIMILARITÀ, GLI SHINGLE SONO DEFINITI COME UN SECONDO DI PUNTEGGIATURA E Poi UNA Z PAROLE SUCCESSIVE.

- **HASHING DEGLI SHINGLE**

- PER RIDURRE L'USO DELLA MEMORIA, POSSIAMO USARE UNA FUNZIONE HASH CHE ASSOCIA AD OGNI STRINGA DI LUNGHEZZA K AD UN BUCKET-NUMBER:

$$- h: \sum_k \rightarrow \{1, 2, \dots, b\}$$
$$s \rightarrow h(s)$$

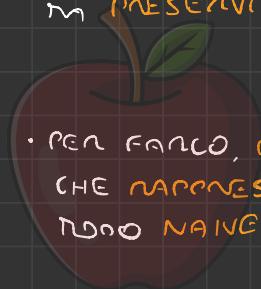
- AD OGNI SHINGLE E' ASSOCIAUTO UN BUCKET-NUMBER DETTO **TOKEN**, E IL DOCUMENTO DIVENTA UN INSIEDE DI TOKEN.

ESEMPIO:

- 1-) UN δ -SHINGLE (δ BYTE) PUÒ ESSERE RAPPRESENTATO CON UN TOKEN INTERO (4 BYTE).

**MINHASHING:
INTRODUZIONE**

- GLI INSIEMI DI SHINGLES SONO MOLTO GRANDI, DUNQUE ANCHE IL CONFRONTO DEI SOLI HASH RISUOVA COSTOSO.
- L'IDEA DEL **MINHASHING** È QUELLA DI TROVARE UNA RAPPRESENTAZIONE DI OGNI INSIEME, DETTA **SIGNSATURE**, CHE SIA MOLTO PIÙ PICCOLA MA PRESERVI LA DISTANZA TRA GLI INSIEMI.



- PER FARLO, COSTRUIMMO UNA **MATRICE DELLE CARATTERISTICHE** CHE RAPPRESENTA GLI INSIEMI DI SHINGLES S_1, S_2, \dots, S_m IN MODO NELLE:

- OGNI COLONNA M_j RAPPRESENTA UN INSIEME S_j ,
- OGNI RIGA M_i^j RAPPRESENTA UN ELEMENTO l_i^j (SHINGLE),

$$- \forall i, j \quad M_{i,j} = \begin{cases} 1 & \text{SE } l_i^j \in S_j \\ 0 & \text{ALTRIMENTI} \end{cases}$$

- LA SIMILARITÀ USATA PER CONFRONTARE GLI INSIEMI È CHIAMATA **SIMILARITÀ DI JACCARD**.

- SIANO $A \in B$ INSIEMI DISSETTI; DEFINISCIORA ALCUNA FUNZIONE DI SIMILARITÀ DI JACCARD:

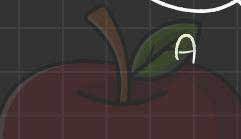
$$- \boxed{S_{\text{sim}}(A, B) \stackrel{\text{def}}{=} \frac{|A \cap B|}{|A \cup B|}}$$

- ESEMPIO:

1)



$$\Rightarrow S_{\text{sim}}(A, B) = \frac{3}{3+4} = \frac{3}{7}$$



- CONSIDERANO UNA PERMUTAZIONE CASUALE π DELLE RIGHE DELLA MATEMATICA DELLE CARATTERISTICHE; UNA FUNZIONE DI MINIASI ASSOCIA AD UNA COLOMNA S_j IL PRIMO INDICE DI UN SUO ELEMENTO PARI AD 1 NELLA PERMUTAZIONE π :

$$- \boxed{l_i(S_j) \stackrel{\text{def}}{=} \arg \min_i \{ \pi(M_{i,j}) : \pi(M_{i,j}) = 1 \} }, \text{ DOVE:}$$

- $\pi(M_{i,j})$ È L'ELEMENTO i,j NELLA MATEMATICA PERMUTATA.

- ESEMPIO :

1-)

Element	S_1	S_2	S_3	S_4	Element	S_1	S_2	S_3	S_4
a	1	0	0	1	b	0	0	1	0
b	0	0	1	0	e	0	0	1	0
c	0	1	0	1	a	1	0	0	1
d	1	0	1	1	d	1	0	1	1
e	0	0	1	0	c	0	1	0	1

Permutation

$$h(S_1) = h(S_4) = a; h(S_2) = c; h(S_3) = b;$$

- OSSERVAZIONE :

- LA PROBABILITÀ CHE 2 HINHASHI SIANO UGUALI È PARI ALLA SIMILARITÀ DI JACCARD:

$$P(h(S_1), h(S_2)) = \text{Sim}(S_1, S_2);$$

- PER LORO PERCORSI, DIVIDIAMO LE RIGHE IN 3 CATEGORIE:
- TIPO X : 1 IN S_1 ED S_2 ;
- TIPO Y : 1 IN S_1 E 0 IN S_2 O VICEVERSA;
- TIPO Z : 0 IN S_1 ED S_2 ;
- SI NOTI CHE SE L'INSIEME DI SHINGLE È COSTITUITO MENO, LA MATRIX H È SPANSA \Rightarrow RIGHE PRINCIPALMENTE DI TIPO Z.

- DUNQUE, DENOTANDO CON X, Y, Z IL NUMERO DI RIGHE DI TIPO X, Y, Z RISPECTIVAMENTE.

$$- \boxed{\text{Sim}(S_1, S_2) = \frac{X}{X+Y} = P(h(S_1) = h(S_2))} .$$

- ESEMPIO:

1.)

C₁ C₂

0 1 *

1 0 *

1 1 * * $\text{Sim}(C_1, C_2) =$

0 0 $2/5 = 0.4$

1 1 * *

0 1 *

SIGNATURE MATRIX

- PER DEFINIRE UNA SIGNATURE DI LUNGHEZZA ℓ :

(i) SI GENERANO m PERMUTAZIONI (LOGICHE, NON REALI)
CON ASSOCIANTE FUNZIONI NINHASH h_1, h_2, \dots, h_ℓ ;

(ii) LA SIGNATURE DEL DOCUMENTO S_J SAM:

$$- \boxed{h_1(S_J) h_2(S_J) \dots h_\ell(S_J)} ;$$

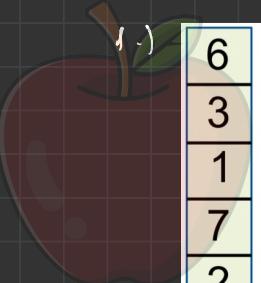
- DEFINIZIONE **SIGNATURE MATRIX** LA MATRICE CONTIENE LE SIGNATURE DEI DOCUMENTI, S_j SULLE COLONNE:

$$H \stackrel{\text{def}}{=} \| h_i(S_j) \|_{\ell^1 \times m}$$

- LA DIMENSIONE DI H DIPENDE DAL NUMERO DI DOCUMENTI MA NON DALLA LORO DIMENSIONE, DI CONSEGUENZA H È TOLTO PIÙ PICCOLA DI M .

- ESEMPIO:

1.)



6	7	1
3	6	2
1	5	3
7	4	4
2	3	5
5	2	6
4	1	7

0	1	1	0
0	0	1	1
1	0	0	0
0	1	0	1
0	0	0	1
1	1	0	0
0	0	1	0

3	1	1	2
2	2	1	3
1	1	3	5

Signature Matrix

- LA SIMILARITÀ TRA DUE SIGNATURE H_1, H_2 È DEFINITA COME IL NUMERO DI ELEMENTI HASH IN COMUNE TRA DI ESSI.
- LA SIMILARITÀ TRA H_1 E H_2 APPROSSIMA LA SIMILARITÀ DI JACCARD TRA LE RISPETTIVE COLONNE S_1 E S_2 .

• ESEMPIO :

1) NELL'ESEMPIO PRECEDENTE :

$$- \text{Sim}(\mathcal{S}_1, \mathcal{S}_2) = 1/4 \in \text{Sim}(\mathcal{H}_1, \mathcal{H}_2) = 1/3$$

$$- \text{Sim}(\mathcal{S}_2, \mathcal{S}_3) = 1/5 \in \text{Sim}(\mathcal{H}_2, \mathcal{H}_3) = 1/3$$

$$- \text{Sim}(\mathcal{S}_3, \mathcal{S}_4) = 1/5 \in \text{Sim}(\mathcal{H}_3, \mathcal{H}_4) = 0$$

$$- \text{Sim}(\mathcal{S}_1, \mathcal{S}_3) = 0 \in \text{Sim}(\mathcal{H}_1, \mathcal{H}_3) = 0.$$

IMPLEMENTAZIONE
TRAMITE FUNZIONI HASHI

- GENERARE NEGLIENDE ℓ PERMUTAZIONI È ESTREMAMENTE COSTOSO, VISTA LA DIMENSIONE DELLA MATRICE M , DUNQUE ESSE SONO SIMULATE USANDO UN INSIEME DI ℓ FUNZIONI HASHI h_1, \dots, h_ℓ COLLISION-RESISTANT.
- SE OGNI h_i È COLLISION-RESISTANT, SI PUÒ ASSUMERE CHE OGNI INDICE DI RIGA n SIA PERMUTATA NELLA POSIZIONE $h_i(n)$.
- INDICHEREMO CON $SIG(i, c)$ L'ELEMENTO DELLA SIGNATURE-MATRIX IN POSIZIONE i, c , DUNQUE IL PIÙ PICCOLO INOCHE PARI AD 1 NELLA COLONNA c NELLA PERMUTAZIONE GENERATA DA h_i .
- DUNQUE L'ALGORITMO PER GENERARE UNA FIRMA OPERA COME SEGUVE:

(z) PONE $SIG(\cdot, \cdot) = \infty$, $\forall \cdot, \cdot$;

(zz) Azienda n, calcola $l_1(n), \dots, l_k(n)$;

(nn) \forall colonna c, se l'elemento $M_{n,c} = 1$, allora pone:

$$- SIG(\cdot, c) = \min(SIG(\cdot, c), l_{\cdot}(n)).$$

ESEMPIO:

1 -)

	C1	C2
Row		
1	1	0
2	0	1
3	1	1
4	1	0
5	0	1

$h(x) = x \bmod 5$
$g(x) = (2x+1) \bmod 5$

(1) Row 1 has 1 in C1
 $h(1)=1 < SIG(h, C1)=\infty$
 $g(1)=3 < SIG(g, C1)=\infty$

	C1	C2
h	1	∞
g	3	∞

(3) Row 3 has 1 in C1 and C2

$$\begin{aligned} h(3)=3 &> SIG(h, C1)=1 \\ h(3) &> SIG(h, C2)=2 \\ g(3)=2 &< SIG(g, C1)=3 \\ g(3) &> SIG(g, C2)=0 \end{aligned}$$

	C1	C2
h	1	2
g	2	0

(4) Row 4 has 1 in C1
 $h(4)=4 > SIG(h, C1)=1$
 $g(4)=0 < SIG(g, C1)=2$
Signature Matrix Unchanged

Initial Signature Matrix:

Row	C1	C2
h	∞	∞
g	∞	∞

	C1	C2
h	1	2
g	3	0

$SIM(C1, C2)=1/5$ whereas $SIM(SIG(C1), SIG(C2))=0$
For bigger matrices the estimation gets closer !!

(5) Row 5 has 1 in C2
 $h(5)=0 < SIG(h, C2)=2$
 $g(5)=1 > SIG(g, C2)=0$

	C1	C2
h	1	0
g	2	0

LOCAL-SENSITIVE HASHING (LSH)

- DATO CHE LA DIMENSIONE DELLA SIGNATURE MATRIX DIPENDE COMUNQUE DAL NUMERO DI DOCUMENTI m , IL NUMERO DI COPIE DI SIGNATURE DA CONFRONTARE È $\mathcal{O}((\frac{m}{2})) = \mathcal{O}(m^2)$.
- L'IDEA DEL **LOCAL-SENSITIVE HASHING** (LSH) È QUELLA DI FAR SI' DA CONFRONTARE SOLO LE COPIE CHE HANNO ALTA PROBABILITA' DI ESSERE SIMILI.
- TALE PROBABILITA' È STABILITA APPLICANDO UNA FUNZIONE HASH h A PIÙ PARZIONI DELLA SIGNATURE MATRIX, COSÌ CHE A VALORI USUALI DI h SIA PROBABILE CHE CONFRONTOANO VALORI SIMILI DEI NINTASH.
- LA SIMILARITÀ O NENO, CHE OBTIENE IN SE UNA COPPIA DI COLONNE È CONSIDERATA AN ESSERE CONFRONTATA, È STABILITA IN BASE AD UNA SOGLIA τ , CHE PUÒ OBTENERE ANCHE NEI FALSI POSITIVI E NEGLATIVI:

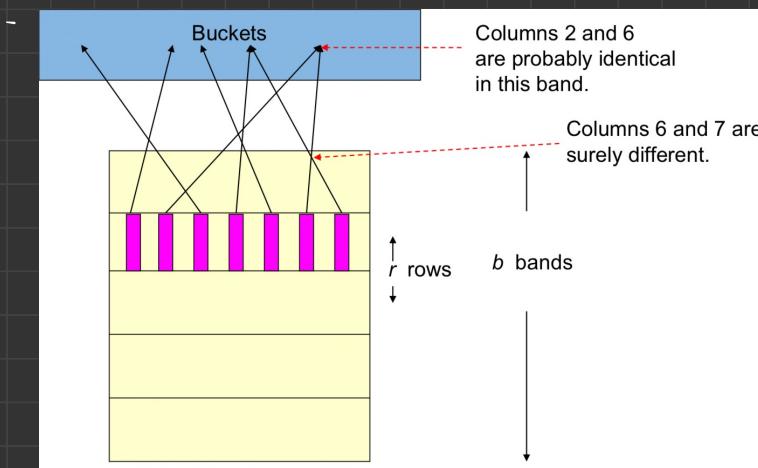
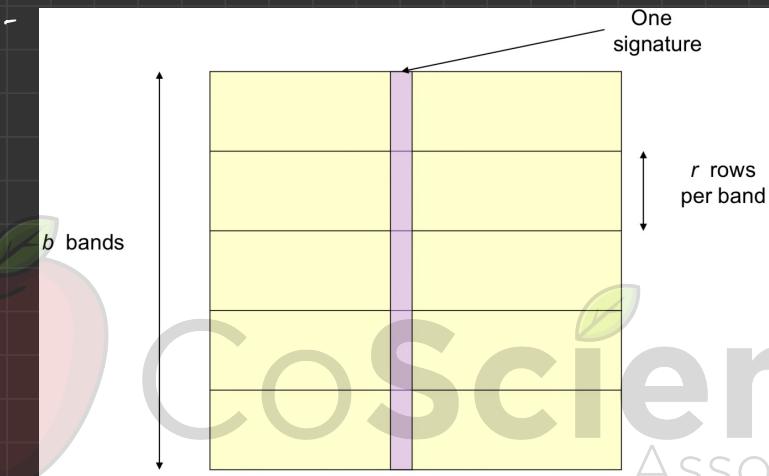
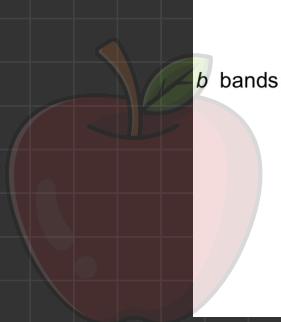
- $a \neq b \in h(a) = h(b) \Rightarrow a, b$ FALSI POSITIVI;
- $a \approx b \in h(a) \neq h(b) \Rightarrow a, b$ FALSI NEGATIVI.

- **SUDDIVISIONE IN BANDE**:

(r) LA SIGNATURE MATRIX H VIENE SUDDIVISA IN b BANDE DA m RIGHE CIASCUA;

(\Leftarrow) OGNI POSIZIONE DI COLONNA IN UNA BANDA VIENE MAPPATA IN UN BUCKET DI UNA HASH TABLE A k POSIZIONI (1 HASH TABLE PER OGNI BANDA);

(\Rightarrow) LE COPPIE CANDIDATE SONO QUELLE CHE COLLOCANO NELLO STESSO BUCKET PER ALMENO 1 BANDA.



- LE COSE SONO SCELTE IN BASE A MINIMIZZARE I FALSI POSITIVI E NEGLATIVI.
- SIA $s = \text{Sim}((c_1, c_2))$ LA SIMILARITÀ TRA I DOCUMENTI c_1 E c_2 , CON FINNE $\text{Sig}(c_1)$ E $\text{Sig}(c_2)$ (UGUALE ALLA PROBABILITÀ CHE $\text{Sig}(c_1) = \text{Sig}(c_2)$ SU 1 VALORE); ALLORA LA PROBABILITÀ:
 - $P((\text{Sig}(c_1), \text{Sig}(c_2)) \text{ COPPIA CORDINATA})$,
- PUO' ESSERE CALCOLATA COMPUTANDO:

(i) LA PROBABILITÀ CHE LE FINNE SIANO UGUALI SU OGNI RIGA DI UNA BANDA È s^n_j

(ii) LA PROBABILITÀ CHE LE FINNE SIANO DIFFERENTI SU ALMENO 1 RIGA IN UNA BANDA È $1 - s^n_j$;

(iii) LA PROBABILITÀ CHE LE FINNE SIANO DIFFERENTI SU ALMENO 1 RIGA IN OGNI BANDA (E DUNQUE QUELLA DI UN FALSO NEGATIVO) È $(1 - s^n)^b$;

(iv) LA PROBABILITÀ CHE LE FINNE SIANO UGUALI SU OGNI RIGA IN ALMENO UNA BANDA (E DUNQUE DI ESSERE UNA COPPIA CORDINATA) È $1 - (1 - s^n)^b$;

• ESEMPIO:

1-) SUPPOSIAMO 100.000 COLUMNS, SIGNATURE DI 100 INTGRI,
 $\tau = 0.80$.

- CI SONO $\binom{100.000}{2} \approx 5.000.000.000$ COPIE DI
FINE DI CONTINUANZA;

- SUPPOSIAMO IN $b=20$ BANDE DI $n=5$ RIGHE;

$$- (1 - s^n)^b = (1 - (0.8)^5)^{20} = 0.00035,$$

- CIRCA 1/3000 DEI HASH POSTI IN SUCCESSIVE
DIMENSIONI SONO FASSI NEGATIVI;

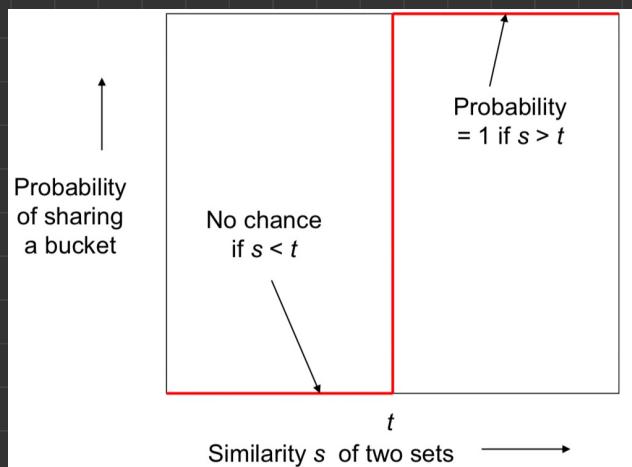
- LA PROBABILITÀ DI ESSERE UNA COPPIA CANDIDATA
SARÀ DI 0.9965;

• IL THRESHOLD τ È UNA FUNZIONE DI b E DI n CHE SI
COMPORTA INCALTAMENTE COME UNA STEP FUNCTION:

- RAPPRESENTA IL LIVELLO DI SIMILITÀ NECESSARIO
AFFINCHÉ LA PROBABILITÀ DI c_1, c_2 DI DIVENTARE
CORTE CANDIDATA SIA $\geq 1/2$;

- VALORI $s < \tau \Rightarrow$ BASSA PROBABILITÀ DI ESSERE
CANDIDATI;

- VALORI $s > \tau \Rightarrow$ ALTA PROBABILITÀ DI ESSERE
CANDIDATI;

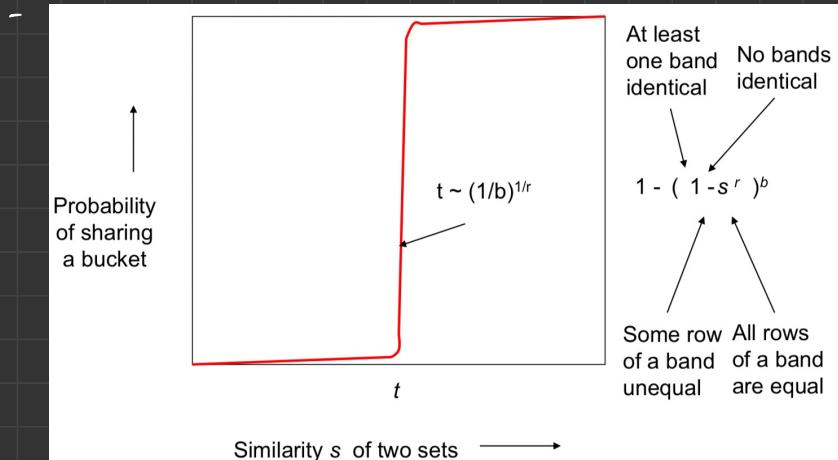


- PER OTTENERE UNA BUONA APPROSSIMAZIONE SI PONE:

-
$$t = (1/b)^{1/r}$$
, questo in quanto:

$$\lim_{b \rightarrow \infty} 1 - \left(1 - \left((1/b)^{1/r} \right)^r \right)^b = \lim_{b \rightarrow \infty} 1 - (1 - 1/b)^b = 1 - 1/e \approx$$

- ≈ 0.64



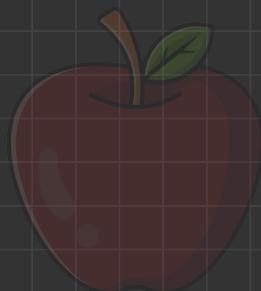
ESEMPIO :

1-) con $b = 20$, $n = 5$, $t = (1/b)^{1/n} \approx 0.55$ -

s	$1-(1-s^r)^b$
.2	.006
.3	.047
.4	.186
.5	.470
.6	.802
.7	.975
.8	.9996

} $s < t \Rightarrow$ non raggiungibile
candidate.

} $s > t \Rightarrow$ raggiungibile candidate.



CoScienze
Associazione

DATA MINING

KNOWLEDGE
DISCOVERY

- IL **[KNOWLEDGE DISCOVERY]** È UN PROCESSO DI ESTRAZIONE DI INFORMAZIONI UTILI DA GRANDI QUANTITÀ DI DATI, SOTTO FORMA DI **[PATTERN]**, OBTENENDO UN'INFORMATIZIONE DI ALTO LIVELLO BASATA SU UNA REGOLARITÀ.
- OGNI PATTERN SCOPERTO DURANTE IL KNOWLEDGE DISCOVERY DEVE ESSERE:
 - VALIDO SU UN CERTO NUMERO DI ISTANZE;
 - UTILE A PRENDERE DECISIONI;
 - COMPRENSIBILE DA UN ESSERE UMANO.



- IL PATTERN PIÙ SCHIPIO DA ESTRAE SONO LE **[REGOLE D'ASSOCIAZIONE]**, SIA $I = \{I_1, I_2, \dots, I_s\}$ UN INSIEME DI ITEM, ACCORDO UNA **[TRANSAZIONE]** È UN SOTTOINSIEME DI ITEM $T \subseteq I$, FUENTRE UNA BASE DI DATI $D = \{T_1, T_2, \dots, T_n\}$ È UN INSIEME DI TRANSAZIONI.

- ESEMPIO:

- 1-) I = INSIEME DEI PRODOTTI VENDUTI DAL SUPERMERCATO.
- 2-) T = CARTELLO DI UN UTENTE X.
- 3-) D = INSIEME DEI CARTELLI DI TUTTI GLI UTENTI.

- UNA REGOLA D'ASSOCIAZIONE È UN'IMPLICAZIONE PROBABILISTICA:
- $X \Rightarrow Y$ T.C. $X, Y \subseteq I$, su cui sono definiti, supponendo di conoscenza, sapendo che $\forall s \in I$:
- $\text{support}(S) \stackrel{\text{def}}{=} |\{T \in D : S \subseteq T\}| / |D|$ (numero di transazioni che contengono S);
- $\text{support}(X \Rightarrow Y) \stackrel{\text{def}}{=} \text{support}(X \cup Y)$ (rilevanza statistica);
- $\text{confidence}(X \Rightarrow Y) \stackrel{\text{def}}{=} \frac{\text{support}(X \cup Y)}{\text{support}(X)}$ (significatività dell'associazione);

ESEMPIO:

$$1) \text{ LATTE} \Rightarrow \text{UOVA}, \text{ support}(\text{LATTE} \Rightarrow \text{UOVA}) = 0.2 \\ \text{confidence}(\text{LATTE} \Rightarrow \text{UOVA}) = 0.3$$

- ▶ Items: valori associati ad un certo attributo in una certa relazione
- ▶ transazione: sottoinsieme di items, raggruppati rispetto al valore di un altro attributo (ad esempio un codice)

T1	111	201	01/05/1999 ink	1
	111	201	01/05/1999 milk	3
	111	201	01/05/1999 juice	6
T2	112	105	03/06/1999 pen	1
	112	105	03/06/1999 ink	1
	112	105	03/06/1999 milk	1
T3	113	106	10/05/1999 pen	1
	113	106	10/05/1999 milk	1
T4	114	201	01/06/1999 pen	2
	114	201	01/06/1999 ink	2
	114	201	01/06/1999 juice	4

2-) ▶ Analisi market basket

- ▶ * \Rightarrow uova
 - cosa si deve promuovere per aumentare le vendite di uova?
- ▶ Latte \Rightarrow *
- quali altri prodotti devono essere venduti da un supermercato che vende latte?

ALGORITMO
APRIORI

- SUPPONIAMO DI VOLER DETERMINARE OGNI REGOLA D'ASSOCIAZIONE $X \Rightarrow Y$ TALE CHE:

- $\text{support}(X \Rightarrow Y) \geq t_1;$

- $\text{confidence}(X \Rightarrow Y) \geq t_2.$

- IL PROBLEMA PUÒ ESSERE SCONPOSTO IN:

(I) DETERMINARE TUTTI I SOTTOINSIEMI $X \subseteq I$ TALI CHE
 $\text{support}(X) \geq t_1$, DETTI **FREQUENT ITEMSET**;

(II) TROVARE TUTTE LE ASSOCIATION RULE CON $\text{support}(X \Rightarrow Y) \geq t_1$
 $\& \text{confidence}(X \Rightarrow Y) \geq t_2;$

- L'**ALGORITMO APRIORI** RISOLVE QUESTI PROBLEMI, INTRODUCENDO UNA RICERCA ESAUSTRIVA INTELLIGENTE CHE INCORPORA DELLE TECNICHE DI PRUNING.

• ESEMPIO :

1-) ▶ Passo 1: estrazione dei frequent itemset

TRANSACTION ID	OGGETTI ACQUISTATI
1	A,B,C
2	A,C
3	A,D
4	B,E,F

▶ supporto minimo 50%

FREQUENT ITEMSET	SUPPORTO
{A}	75%
{B}	50%
{C}	50%
{A,C}	50%

2-) ▶ Passo 2: estrazione regole

- ▶ confidenza minima 50%
- ▶ Esempio: regola $A \Rightarrow C$
 - ▶ supporto $\{A,C\} = 50\%$
 - ▶ confidenza = supporto $\{A,C\}$ /supporto $\{A\} = 66.6\%$
- ▶ regole estratte
 - ▶ $A \Rightarrow C$ supporto 50%, conf. 66.6%
 - ▶ $C \Rightarrow A$ supporto 50%, conf. 100%

• L'algoritmo appena per la fase (I) opera così segue:

(i) INIZIALIZZA UN INSIEME DI FREQUENT ITEMSET

$$- L_1 = \{ I_j \in I : \text{support}(I_j) \geq t_1 \}$$

(ii) AL PASSO k COSTRUISE UN FREQUENT ITEMSET CANDIDATO,

C_k DI DIMENSIONE k :

o) AGGIUNGONO OGNI $A \in L_{k-1}$ A C_k (SOLUZIONE ESALUTIVA);

b) AGGIUNGENDO OGNI $A \in L_{k-1}$ A C_k SE E SOLO SE ASCA, S E' FREQUENT, DUNQUE DUNQUE $\text{support}(S) \geq t_1$ (SOLUZIONE OTIMIZZATA);

(ii) PER OGNI $A \in C_k$, A E' POSTO IN L_k SE E SOLO SE $\text{support}(A) \geq t_1$, DUNQUE SE A E' FREQUENT;

ESEMPIO:

1-) ▶ Base di dati D

TID	Items
100	1, 3, 4
200	2, 3, 5
300	1, 2, 3, 5
400	2, 5

- ▶ Supporto minimo 50% (cioè almeno 2 transazioni)
- ▶ nel seguito con supporto intendiamo il numero di transazioni e non la percentuale per comodità

SOLUZIONE ↗)

Scansione D (1)

C_1

L_1

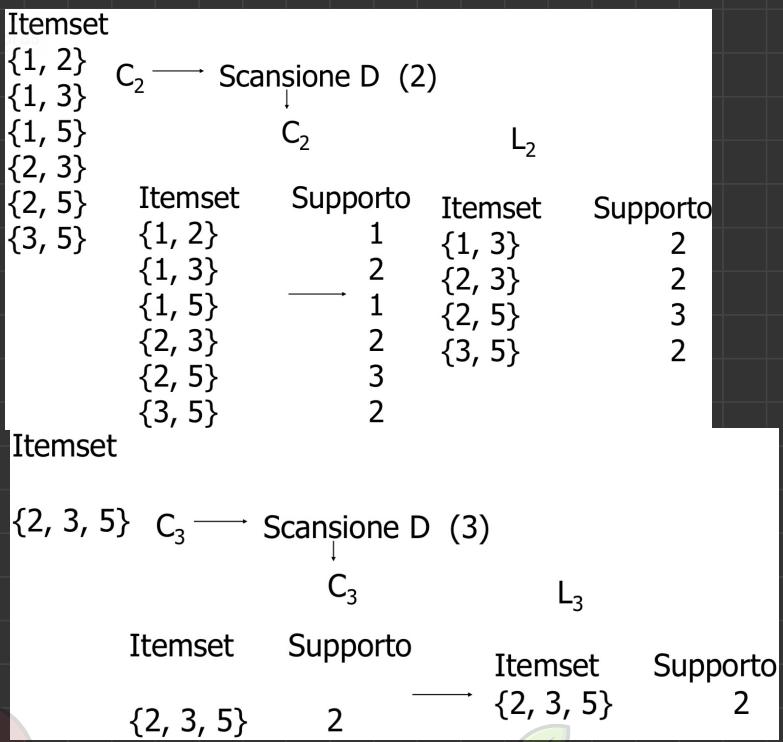
Itemset	Supporto (*4)	Itemset	Supporto
{1}	2	{1}	2
{2}	3	{2}	3
{3}	3	{3}	3
{4}	1	{5}	3
{5}	3		

Itemset				
{1, 2}		C ₂	Scansione D (2)	
{1, 3}				
{1, 4}		C ₂		L ₂
{1, 5}	Itemset	Supporto	Itemset	Supporto
{2, 3}	{1, 2}	1	{1, 3}	2
{2, 4}	{1, 3}	2	{2, 3}	2
{2, 5}	{1, 4}	1	{2, 5}	3
{3, 4}	{1, 5}	1	{3, 5}	2
{3, 5}	{2, 3}	2		
	{2, 4}	0		
	{2, 5}	3		
	{3, 4}	1		
	{3, 5}	2		

Itemset				
{1, 3 ,2}		C ₃	Scansione D (3)	
{1, 3, 4}				
{1, 3, 5}		C ₃		L ₃
{2, 3, 4}				
{2, 3, 5}				
{2, 5, 1}	Itemset	Supporto	Itemset	Supporto
{2, 5, 4}	{1, 3 ,2}	1	{2, 3, 5}	
{3, 5, 4}	{1, 3, 4}	1		
	{1, 3, 5}	1		
	{2, 3, 4}	0		
	{2, 3, 5}	2		
	{2, 5, 1}	1		
	{2, 5, 4}	0		
	{3, 5, 4}	0		

SOLUZIONE Q) :

Scansione D (1)				
		C ₁		L ₁
Itemset	Supporto (*4)		Itemset	Supporto
{1}	2		{1}	2
{2}	3	—	{2}	3
{3}	3		{3}	3
{4}	1		{5}	3
{5}	3			



2-) Supporto = 75% (3 transazioni su 4)

relazione:

Transid	custid	date	item	qty
111	201	01/05/1999	pen	2
111	201	01/05/1999	ink	1
111	201	01/05/1999	milk	3
111	201	01/05/1999	juice	6
112	105	03/06/1999	pen	1
112	105	03/06/1999	ink	1
112	105	03/06/1999	milk	1
113	106	10/05/1999	pen	1
113	106	10/05/1999	milk	1
114	201	01/06/1999	pen	2
114	201	01/06/1999	ink	2
114	201	01/06/1999	juice	4

SOLUZIONE a):

- ▶ Level 1:
 - ▶ L1: {pen} 1, {ink} 3/4, {milk} 3/4
- ▶ Level 2:
 - ▶ C2: {pen, ink}, {pen, milk}, {pen, juice},
{ink, milk}, {ink, juice}, {milk, juice}
 - ▶ L2: {pen, ink} 3/4, {pen, milk} 3/4
- ▶ Level 3:
 - ▶ C3: {pen, ink, milk}, {pen, ink, juice},
{pen, milk, juice}
 - ▶ L3: nessuno

SOLUZIONE b):

- ▶ Level 1:
 - ▶ L1: {pen} 1, {ink} 3/4, {milk} 3/4
- ▶ Level 2:
 - ▶ C2: {pen, ink}, {pen, milk}, {ink, milk}
 - ▶ L2: {pen, ink} 3/4, {pen, milk} 3/4
- ▶ Level 3:
 - ▶ C3: nessuno

• PER LA FASE (II), DATO CHE IL SUPORTO DEI FREQUENT ITEMSET
È GIÀ PER COSTRUZIONE $\geq t_1$, È NECESSARIO SOLO CALCOLARE LE
ASSOCIATION RULE LA CUI CONFIDENZA È $\geq t_2$.

• SIA L_N L'OUTPUT DELLA FASE (I), ALLORA PER OGNI FREQUENT
ITEMSET $X \in L_N$:

(1) DIVIDE X IN DUE SOTTOINSIEMI LHS E RHS, TALI CHE:

- $LHS \cup RHS = X$;

- $\text{confidence}(LHS \Rightarrow RHS) = \text{support}(LHS \cup RHS) / \text{support}(LHS) \geq t_c$.

• ESEMPIO:

1-)

- ▶ Frequent itemset: {pen} 1, {ink} 3/4, {milk} 3/4, {pen,ink} 3/4, {pen, milk} 3/4
- ▶ voglio costruire regole non banali (confidenza = 1)
- ▶ considero {pen,milk}
 - ▶ $\text{supporto}(\{\text{pen,milk}\}) = 3/4$
 - ▶ $\text{supporto}(\{\text{pen}\}) = 1$
 - ▶ $\text{supporto}(\{\text{milk}\}) = 3/4$
 - ▶ $\text{confidenza}(\text{pen} \Rightarrow \text{milk}) = 3/4$ non restituita
 - ▶ $\text{confidenza}(\text{milk} \Rightarrow \text{pen}) = 1$ restituita
- ▶ considero {pen,ink}
 - ▶ $\text{supporto}(\{\text{pen,ink}\}) = 3/4$
 - ▶ $\text{supporto}(\{\text{pen}\}) = 1$
 - ▶ $\text{supporto}(\{\text{ink}\}) = 3/4$
 - ▶ $\text{confidenza}(\text{pen} \Rightarrow \text{ink}) = 3/4$ non restituita
 - ▶ $\text{confidenza}(\text{ink} \Rightarrow \text{pen}) = 1$ restituita

ESTENSIONI DEGLI
ALGORITMI APPLICATI

• ITEM GERARCHICI:

• IN molti casi, gli item possono essere organizzati in modo gerarchico (es. categorie, sotto-categorie, ...).

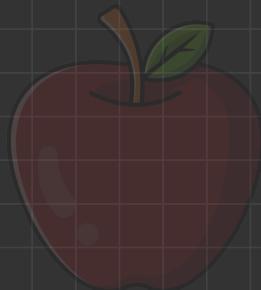
• In tali casi, il supporto può essere calcolato in modo gerarchico, in quanto questo di un item padre può essere ottenuto sommando il supporto degli item figli.

- ESEMPIO:



• **NEGOLE PERSONALIZZATE**:

- LE NEGOLE DI ASSOCIAZIONE POTREBBERO ESSERE DETERMINATE ANCHE IN MODO PERSONALIZZATO FILTRANDO SULLE TRANSAZIONI DI UN SINGOLO UTENTE, O SULLA BASE DI ALTE CONDIZIONI.



CoScienze
Associazione

IMPORTANTE!

DISCLAIMER

Il materiale contenuto nel drive è stato raccolto e richiesto tramite autorizzazione ai ragazzi frequentanti il corso di studi di Informatica dell'Università degli Studi di Salerno.

Gli appunti e gli esercizi nascono da un uso e consumo degli autori che li hanno creati e risistemati, per tanto non ci assumiamo la responsabilità di eventuali mancanze o difetti all'interno del materiale pubblicato.

Il materiale sarà modificato aggiungendo il logo dell'associazione, in tal caso questo possa recare problemi ad alcuni autori di materiale pubblicato, tale persona può contattarci in privato ed eliminaremo o modificheremo il materiale in base alle sue preferenze. Ringraziamo eventuali segnalazioni di errori così da poter modificare e fornire il miglior materiale possibile a supporto degli studenti.

Associazione CoScienze