

BASI DI DATI

**APPUNTI
PERSONALI**

Schemi logico relazionale (modello relazionale)

È un modello logico di rappresentazione dei dati in un database. Si basa sulla teoria degli insiemi e sulla logica del primo ordine. È strutturato attorno al concetto matematico di relazione (tabella). Per trattarla ci si avvale di Algebre relazionale e calcolo relazionale.

Struttura base

- Attributo o campi dato
- tipo di dato e dominio su quel tipo
- valore per ciascun attributo
- tuple, esempio o record

NOME	MATRICOLA	SESSO
LAURA	294163	F
LUCA	271422	M
LORENZO	266292	M
MARIELLA	239478	F

Vincoli di integrità referenziale

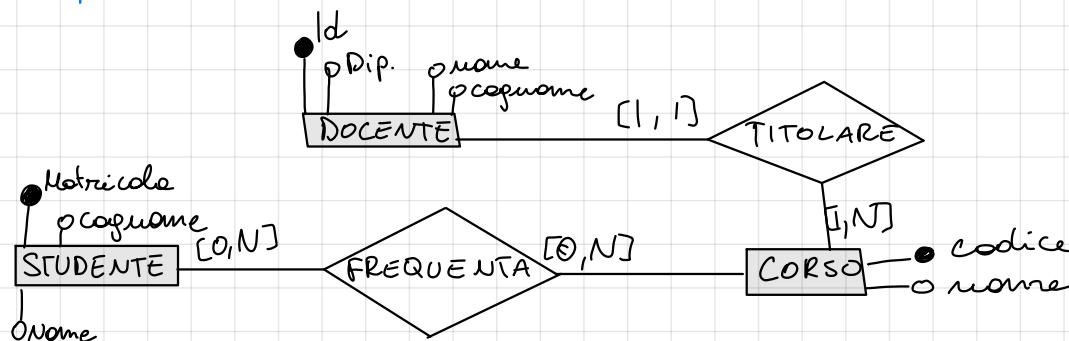
Quando si crea una relazione fra due tabelle è possibile stabilire i vincoli sopracitati: l'integrità referenziale è un insieme predefinito di vincoli o regole che il programma per accertarsi che la relazione sia valida. Es:

- il campo correlato nella 1a tab. deve essere primary key
- i campi correlati devono avere lo stesso tipo di dati
- le tabelle devono appartenere allo stesso database
- [...]

Regole di derivazione: — SCHEMA CONCETTUALE —

1. Ogni ENTITÀ diventa una relazione (tabelle)
2. Ogni ATTRIBUTO di entità diventa un attributo della relazione (colonne).
3. L'identificatore univoco diventa chiave primaria
4. L'ASSOCIAZIONE di tipo 1:N viene rappresentata aggiungendo al lato N una foreign key
5. L'associazione 1:1 { a partecipazione reciprocamente totale } foreign key
6. L'associazione 1:1 { a partecipazione parziale totale } foreign key

Esempio: schema concettuale → relazionale

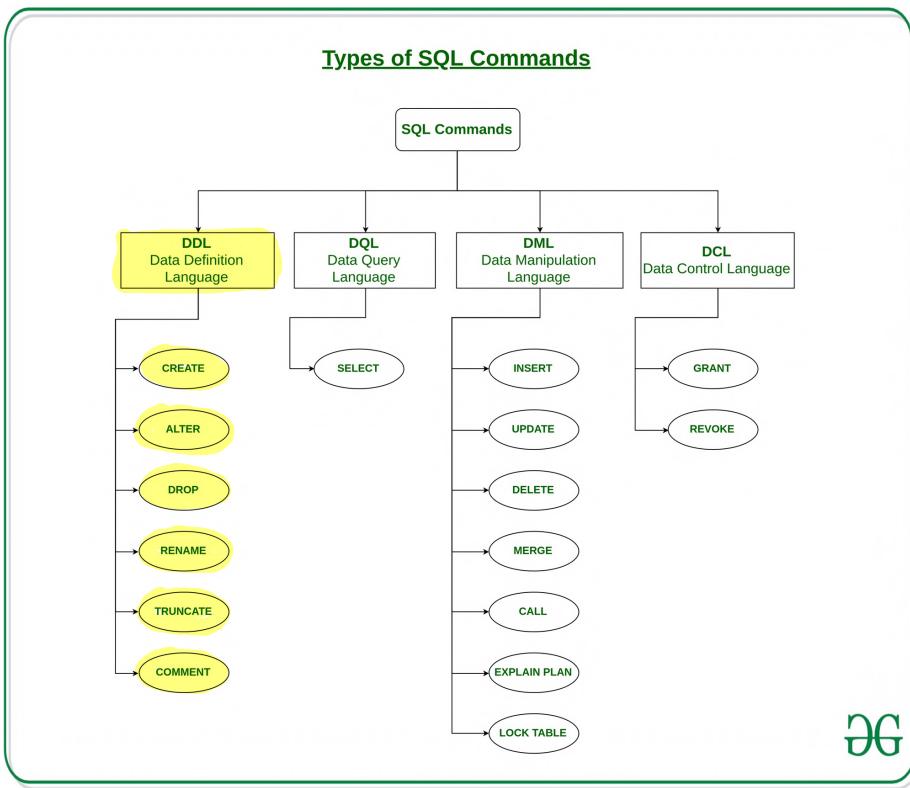


- 1) Entità: Docente (Id, cognome, nome, dip)
 Corso (codice, nome)
 Studente (matricola, cognome, nome)

- 2) Relazioni: 1:N lato N foreign key Corso (codice, nome, id_FK)
 N:N nuova tab. coppia chiave Frequenza (matricola, codice)

DDL - Data definition language

- Il DDL di SQL permette di definire schemi di relazione (tabelle)
- Permette di specificare vincoli, a livello di riga (tuple) & tabella



Il DDL contiene i comandi che definiscono per le tabella, come:

- > Create (!)
- > Alter (?)
- > Drop (?)
- > Rename (?)
- > Truncate (x)
- > Comment (?)

Vincoli e chiavi - primarie ed esterne -

Primary Key:

Una chiave primaria viene utilizzata solo per individuare ogni riga presente in una tabella. Può far parte dello stesso record corrente oppure essere un campo artificiale e non avere che fare con il record corrente.

Può consistere di uno o più campi presenti in una tabella. Se vengono usati più campi come chiave primaria, si dice chiave composta.

Foreign Key:

Una chiave esterna rappresenta uno o più campi che fanno riferimento alle chiavi primarie di un'altra tabella. Lo scopo è quello di garantire l'integrità referenziale dei dati. Cioè sono consentiti solo valori che si ritiene debbano apparire nel database.

Funzioni SQL:

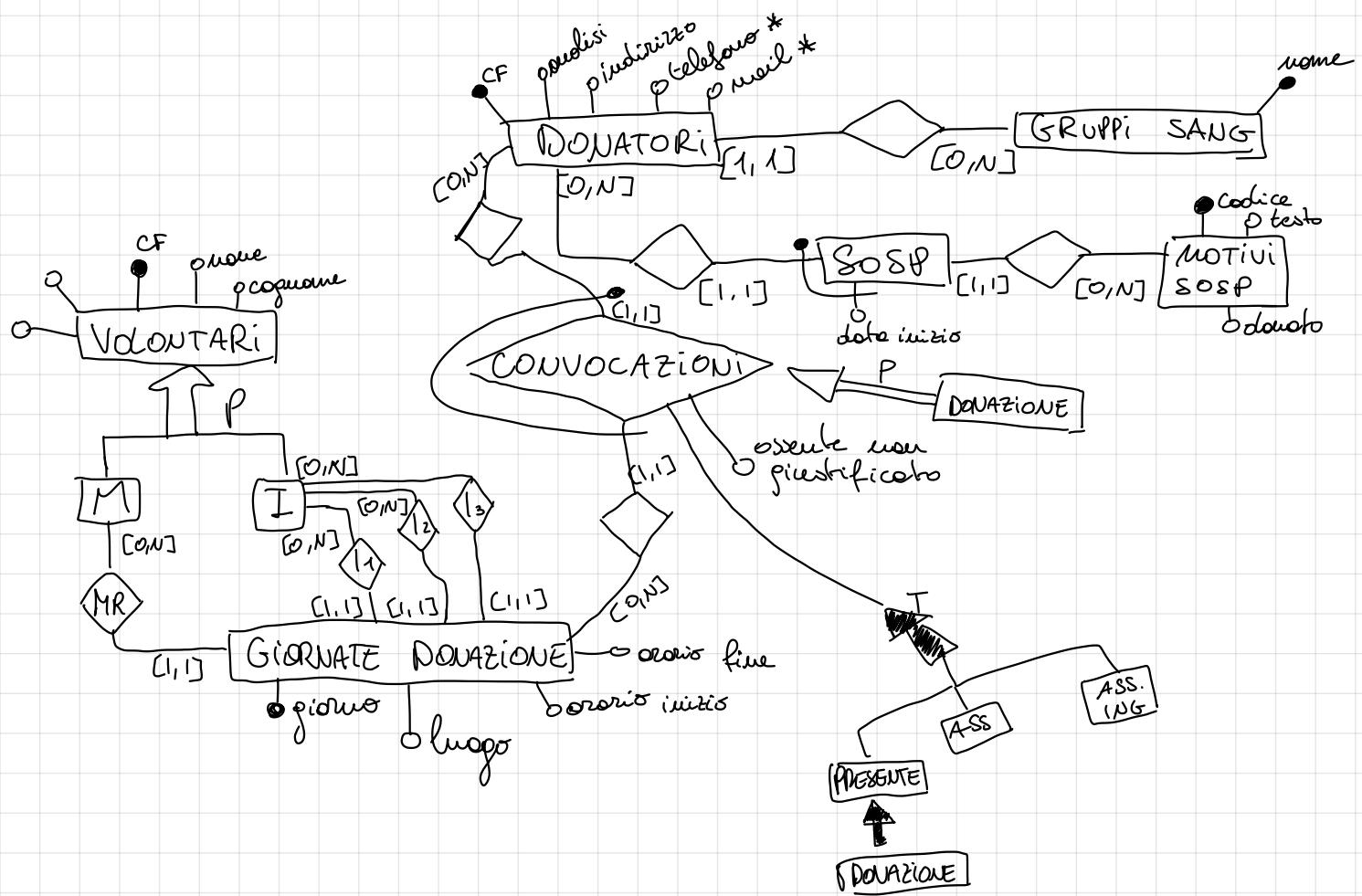
JOIN: Le tabelle possono relazionarsi fra loro di modo da estrarre info da più tabelle con una unica query. Esistono tipi diversi di join necessari per creare query complicate.

Esempio: Supponendo di avere un database di un negozio online con le tabelle:

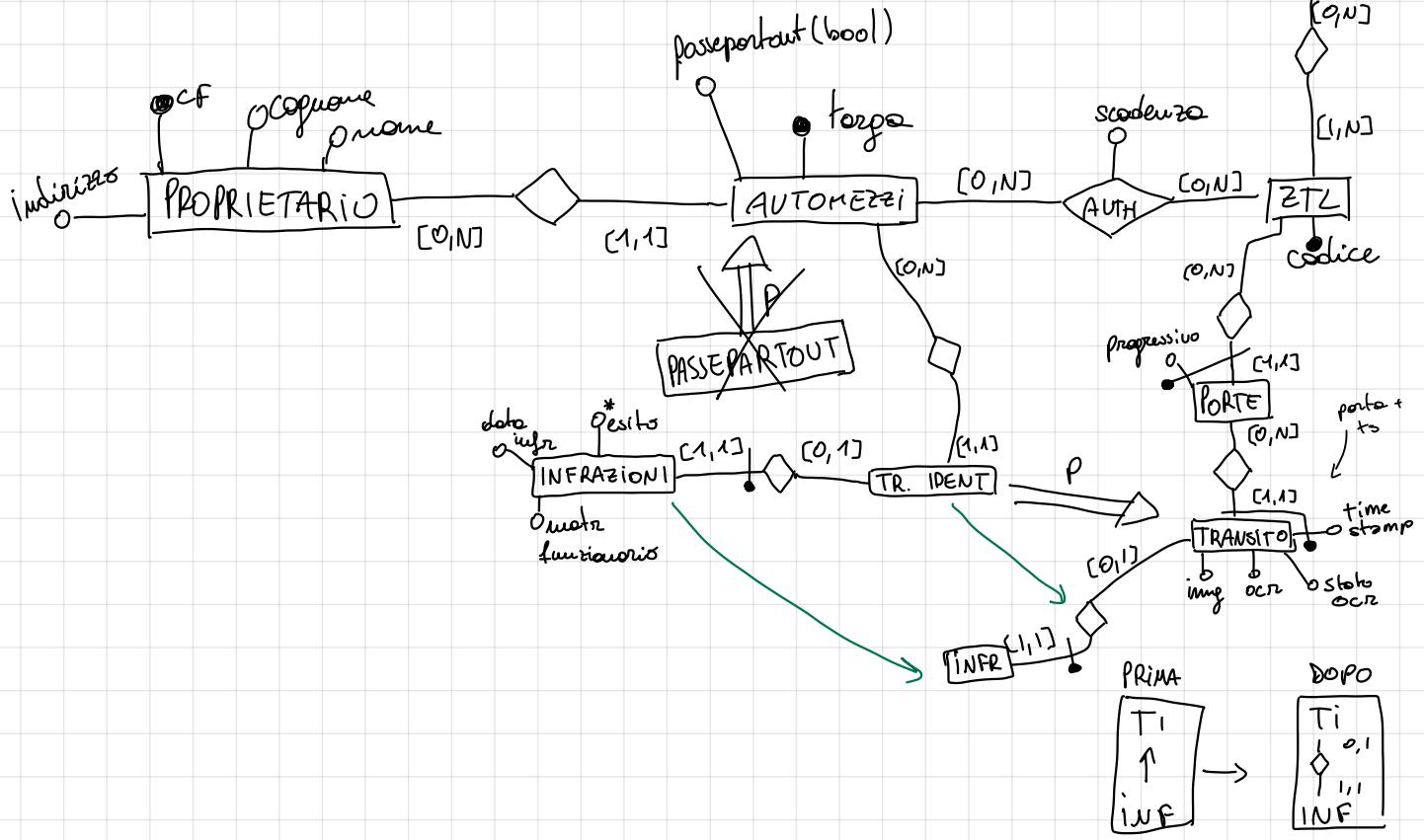
acquisti	clienti
- id	- id (PK)
- id - cliente (fk)	- nome
- prezzo	- cognome
- articolo	

INNER JOIN: (default join):

Appello 2005 06 13 Schema concettuale



12/12/20

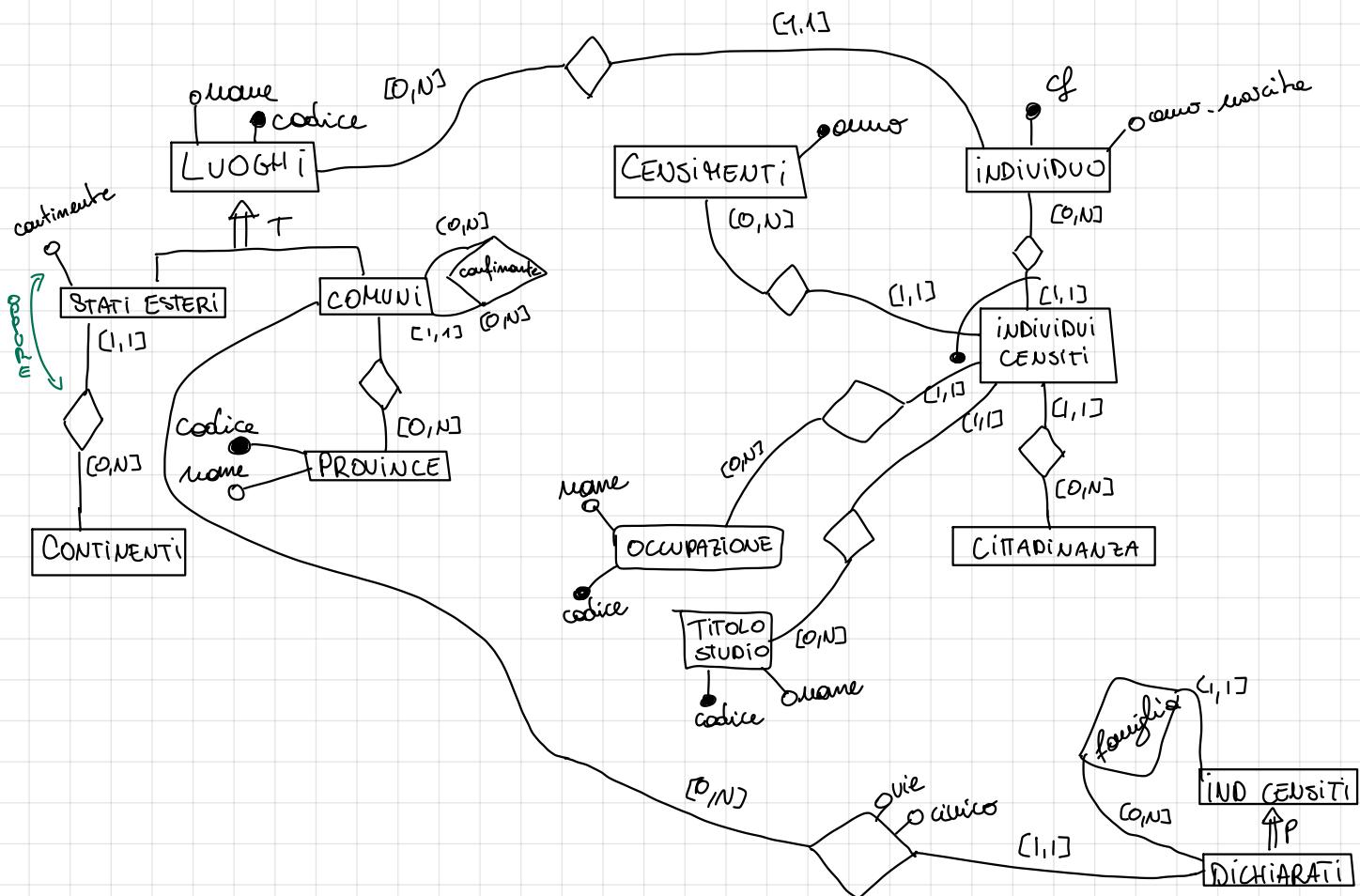


19/12/20

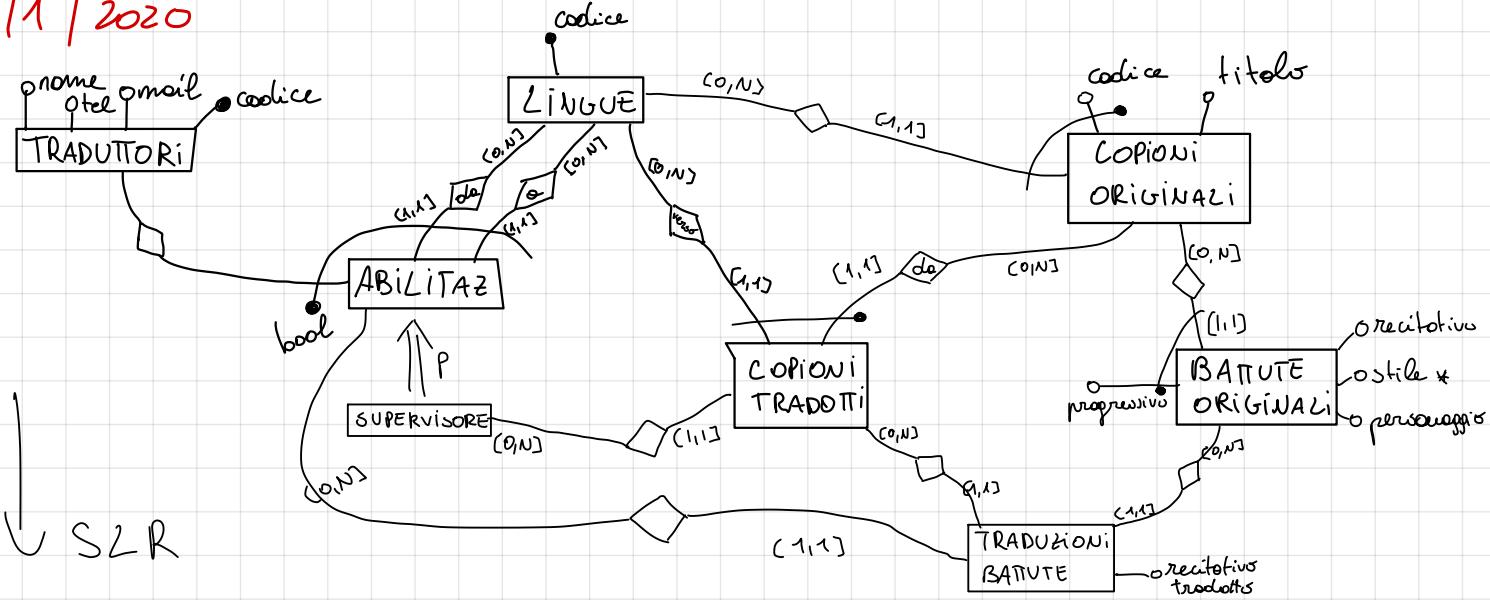
De recuperare le lezioni precedente:

- triggers - timestamp - schedule
- profilo dei conflitti

20/12/19



9/1/2020



— lingue (_codice_)

```
Create table lingue (
codice varchar(10) not null primary key);
```

—copioni originali (_lingua fk_, _codice_, titolo)

```
Creare table copioni_originali (
Lingua varchar 10 not null references lingue(codice),
Codice integer not null,
Titolo varchar 200 not null,
Progr integer not null,
Primary key (lingua_copione, codice_copione, progr)
```

IL MODELLO E-R

Modello entity-relationship: I costrutti principali del modello

Entità: rappresentano classi di oggetti (fatti, cose, persone) che hanno proprietà comuni ed esistenza autonoma ai fini dell'applicazione di interesse. Un'occurrenza di un'entità è un oggetto o istanza della classe che l'entità rappresenta. Il nome è univoco.

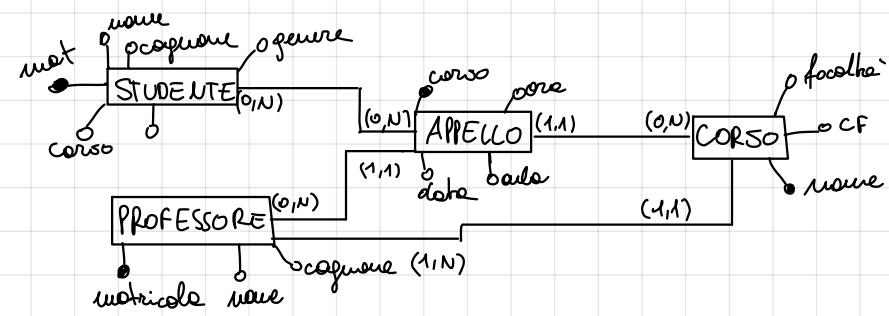
Associazione: le associazioni (relazioni) rappresentano un legame fra due o più entità.

Attributi: Si usano per descrivere entità e associazioni. Gli oggetti della stessa classe entità o associazione hanno gli stessi attributi.

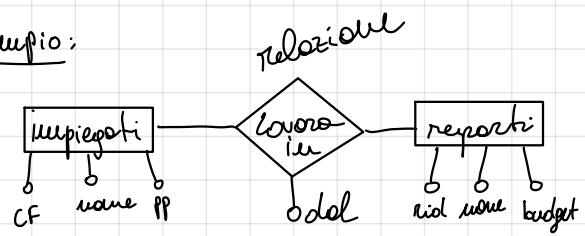
Esempio: gestione studenti di un corso di laurea

entità: - studente - corso - appello - professore

studente: - nome - cognome - matricola ecc.



Esempio:



25-11-2015 A

5)

Musei (name, città)

Opere (codice, artista, titolo, anno, museo, ...)

*

Prestiti (opere fk, museo fk, data-i, data-f)

Restauri (opere, data-i, data-f)

L'opera è di un, altro museo in prestito

Select codice-museo where (opere.codice = prestiti.opera AND opere.museo = prestiti.museo)

L'opera è del museo, non in prestito

(musei.name = opere.museo and codice not in (select opere from prestiti);

d'opere non è in restauro ad oggi → current_date > data-i;

where codice not in (select opere from restauri where AND
current_date < data-f;
data-i ↗ ↘ data-f
curr ↗ ↘ not in ↗ ↘ current-date
data-i ↙ ↘ data-f

6) Un museo uffizi propone mostre Giotto

15 aprile - 5 giugno → prestiti ⇒ ottere table prestiti per tenere traccia

Prestiti (opere, museo, data-inizio, data-fine)

→ insert into table prestiti (opere, museo, d-i, d-f)

select codice, 'Uffizi', '15 aprile 2015', '5 giugno 2015'
from opere

where artista = 'Giotto' AND museo != 'Uffizi';

7) elencare artisti che (in tutti i musei) hanno almeno 10 opere
ordinandoli in base al numero di opere, nome artista
x ogni artista, n musei che hanno almeno una sua opera.

select artista, count(*) as num-opere, count(DISTINCT museo) as num-musei
from opere
group by artista having count(*) >= 10
order by num-opere, artista;

2015 04 27 B

Vincolo → no due sezioni elett. stessa indirizzo
oltre tutte sezioni odd constraint no same odd unique (indirizzo)

Vista elettorato passivo → persone suff. a candidarsi:

{+25 anni e vivi
tessere rilasciate da set ≠ a candidatura
Tessere emesse prima elezioni}

indicare distn, n° tessere
e dati demografici

create or replace view elettorato_passivo (distretto, n_tessere, cf, nome, cognome, data_morte) as
select te.numero, te.sezione-distretto, p.nome, p.cognome, p.codice-fiscale, p.data-morte
from persone p, tessere_elettorali te, candidature c

where te.persone = p.codice_fiscale and
cast('01-04-15' as date) - p.data_morte > 25 * 365 and {
p.data_morte is null and } 1
te.persone = c.persone and } join
c.distretto != te.sezione-distretto and } 3
te.emissione < cast('01-04-15' as date) } 2

Per candidarsi + - no.co.
dn, cf

1 25 anni età (vivi)
2 tessere → distr. ≠ da candidatura
3 tessere → emissione < elezioni

1) persone (cf, nome, cognome, data_morte, n_tessere)
2) tessere elettorali (sezione -> ol)
3) tessere elettorali (data emissione) + numero
→ candidature.

Cancellare delle candidature chi non è elett. passivo

delete from candidature c
where c.persone not in (select cf from elett_passivo);

Vista:

Ogni distretto → ogni candidato → n. voti scrutinati nelle sez. del distretto che
indicano preferenze per quel candidato

Vista: calcola x ogni distretto n tot voti scrutinati nelle sez di quel distretto

Con le visto calcola % voti ottenuti per ogni distretto per i candidati

create or replace view voti_per_candidato (#, nome, cognome set
distretto) as
select sezione-distretto, count(*) as voti
from Voti-scrutinati
where candidato is not null
group by sezione-distretto, candidato

create or replace view voti_per_distretto (distretto, voti_tot) as
select distretto, count(*) as voti_tot
from Voti-scrutinati
group by sezione-distretto

Voti-totali / voti-per-candidato
VT VC

Voti
VC / 100
VT
voti-tot

Select f.districto, c.candidato, (c.voti * 100) / t.voti-tot as per_voti
from voti-per-sezione c, numero_voti_totale t;

25-11-15 B

1) altezza e larghezza $0 < x \leq 5$ metri (espr am)
+ dipinto $< 10 \text{ m}^2$

Alter table opere add constraint dim

check ((larghezza between 1 and 500) and (tipo != 'dipinto' or larghezza * lunghezza < 10000));
lunghezza " " " "

2) Prestiti avvenuti da fine novembre 2015 \rightarrow fine dell'anno
update prestiti set data-fine '31-12-2015' where data-fine between '1-11-2015' and
'30-11-2015');

3) Eleniare musei che non hanno mai concordato un'opera in prestito.

Select museo from opere where not exist (Select opere from prestiti where

Musei(museo, citta)

Artisti

Opere(opere, museo, codice)

Prestiti(opere, museo, d_i, d_f)

opere.codice =
prestiti.opere);

Vista opere fruibili

opere \leftarrow del museo

opere \leftarrow in prestito da altri musei

opere \leftarrow non in restouro e non in prestito

Create view opere_fruibili(museo, codice) as
Select(museo, codice) from opere where

- non restouro

dove il codice delle opere che sono state restourate ha la data di fine
< di oggi

- non in prestito

dove codice opere del museo non ha

date fra d_i e d_f nei prestiti

opere.codice not in (select opere from prestiti

where date_i < current_date and date_f > current_date)

|
opere.codice in (select opere from prestiti where
prestiti.date-fine < cost('today' as date) AND
" ".date-initio > cost('today' as date);

- prestito da altro museo

| opere.(prestiti.opere = opere.codice) and

| museo in (select museo from prestiti
where d_i < curr and d_f > curr)

- del museo

| il nome del museo è = al museo dell'opera

| musei.nome = opere.museo

query per view || select opere.museo, opere.codice from opere, musei, prestiti
where
- di pup e non in prestito
 $(opere.museo = musei.nome) \text{ and } opere.codice \notin (\text{select opere from prestiti where d-iz-inizio} < \text{current_date} \text{ and d-f-fine} > \text{current_date})$ or
- è in prestito da qualcuno
 $(prestiti.opera = opere.codice) \text{ and opere.museo in } (\text{select museo from prestiti where d-i} < \text{current_date} \text{ and d-f} > \text{current_date})$ and
- non in restauri
opere.codice not in (select opere from restauri where d-i < curr and d-f > curr);

Note: Cosa è un trigger?

Il trigger è una procedura che viene eseguita in maniera automatica in coincidenza di un determinato evento, come per esempio la cancellazione di un record di una tabella. Così facendo si ha modo di specificare e mantenere vincoli anche complessi.

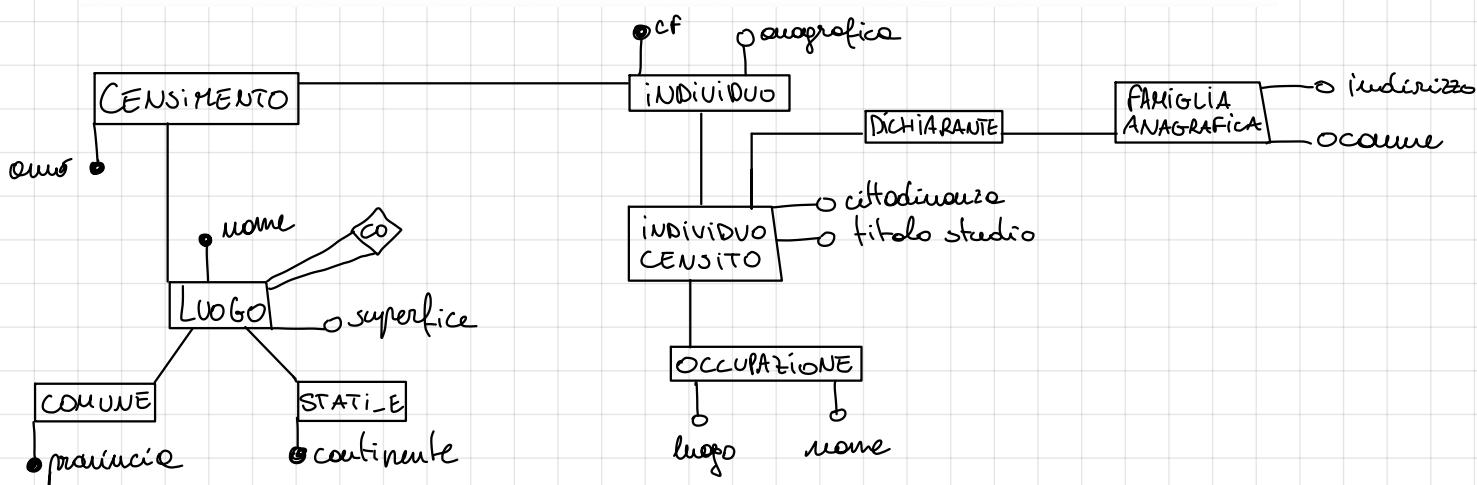
Secondo portiole 2007-02-22

Note

1. Su tutti i fogli contenenti le soluzioni indicare, IN STAMPATELLO, la data dell'appello ed il proprio cognome, nome e numero di matricola
2. Non è consentita la consultazione di alcunché.
3. L'orario di consegna scritto alla lavagna è tassativo.
4. Il testo del compito va consegnato insieme ai fogli con le soluzioni.

Esercizi

1. Mostrare uno schema concettuale che descriva (una frazione de-) i dati raccolti in occasione di una serie di censimenti anagrafici nazionali. Sono di interesse le seguenti informazioni:
 - (a) Il censimento, identificato dall'anno di svolgimento.
 - (b) Il luogo, distinto in comune (italiano) o stato estero. I comuni sono raggruppati in province; gli stati esteri sono raggruppati in continenti.¹ Per ogni comune si conosce la superficie (in chilometri quadrati) e quali altri comuni sono con esso confinanti.
 - (c) L'individuo, di cui interessa il codice fiscale, il luogo e l'anno di nascita nonché, per ogni censimento al quale ha partecipato, la cittadinanza, il titolo di studio (nessuno, licenza elementare, licenza media inferiore, ecc.), la tipologia di occupazione prevalente (lavoratore dipendente, lavoratore autonomo, studente, ecc.) ed il comune presso il quale tale occupazione è svolta.
 - (d) La famiglia anagrafica, identificata all'interno del censimento da uno degli individui che la compongono (il "dichiarante") e della quale interessa conoscere il comune di residenza e l'indirizzo (via e numero civico). Si noti che, al variare dei censimenti, un individuo può o meno essere censito e può appartenere a famiglie diverse; fissato un censimento, ogni individuo censito deve far parte di una ed una sola famiglia anagrafica.
 - (e) La relazione di parentela, affinità o altro (dichiarante, coniuge, genitore, figlio, convivente, ecc.) esistente tra il dichiarante e gli individui che compongono la sua famiglia anagrafica.
2. Tradurre lo schema concettuale dell'esercizio precedente in uno schema logico relazionale, codificando opportunamente i vincoli dello schema. (Nota: è consentito utilizzare una notazione semi-formale, purché sufficientemente chiara e completa).



CENSIMENTO (anno)

LUOGO (name, confine, superficie)

COMUNE (provincia, name fk)

STATI_E (name fk, continente)

INDIVIDUO (CF, d-n, luogo fk)

INDIVIDUO_CENSITO (CF, individuo fk, cittadinanza, titolo studio, censimento fk, occupazione fk)

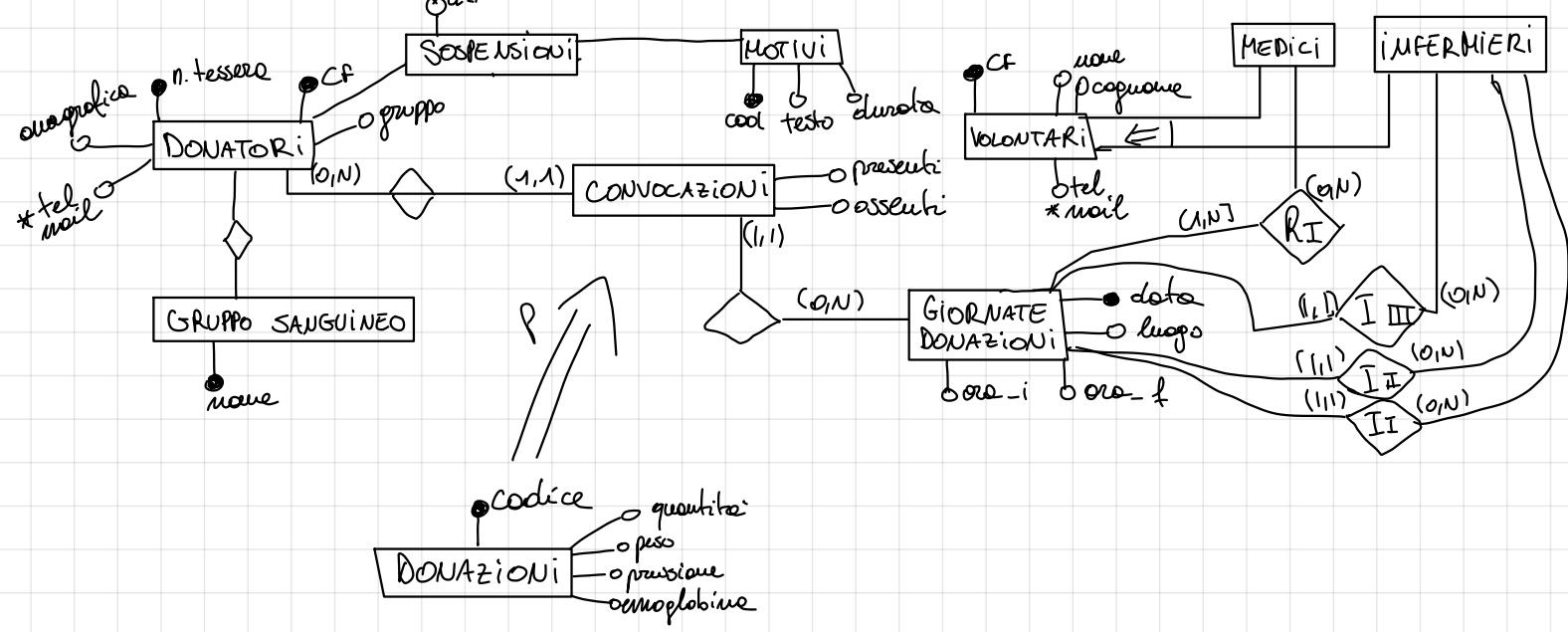
DICHIARANTE (censimento fk, individuo fk, residenza fk, indirizzo)

FAMIGLIA_A (name)

OCCUPAZIONE (luogo fk, name)

13/06/05

Associazione donatori → soci
gruppo
data analisi
onografica + tel remail
mo' essere sospeso



DONAZIONI (codice, peso, pressione, emoglobina, donatore fr_+ , quantità)

DONATORI (n. tessera, gruppo_{fr}, d-ultime-analisi, CF, nome, cognome, indirizzo, d-n, tel*, mail*)

SOSPENSIONI (olaiatore, emotivo)

CONVOCAZIONI (presenti, assenti)

VOLONTARI (medici ex, infermieri ex, nuove, cognome, cognome, indirizzo*, tel*)

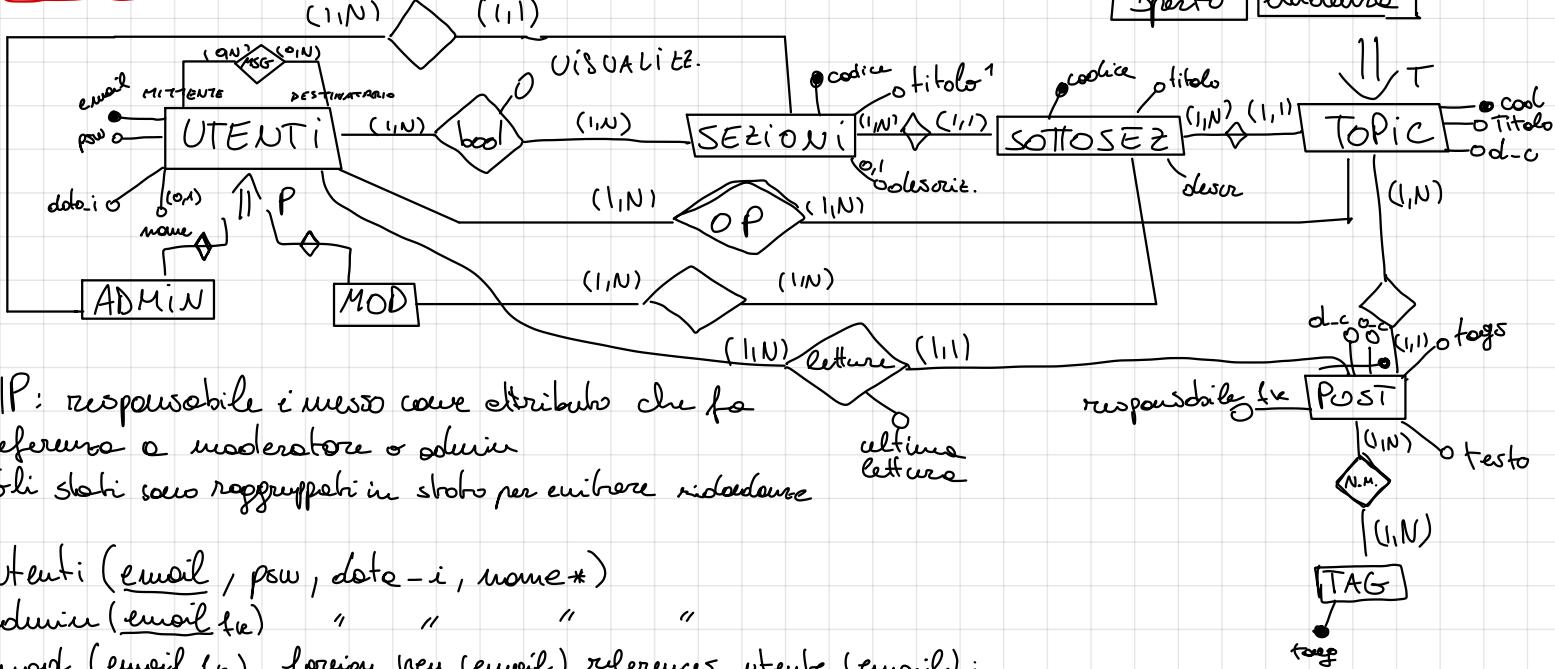
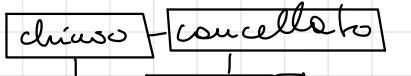
INFERMIERI (nuove, copiare, CF, indirizzo*, tel*)

MEDICI(nome, cognome, cf, indirizzo*, tel*)

GIORNATE-SON(date, luogo, ora-i, ora-f)
GP-name & luogo

GRUPPO_SANG(nome)

23/02/2010



P: responsabile è messo come attributo che fa riferimento a moderatore o amministratore
Gli stessi sono raggruppati in stessa per evitare ridondanza

utenti (email, psw, data-i, name*)

admin (email fk) " " " "

mod (email fk) foreign key (email) references utente (email);
msg (data, ora, codice, mittente fk, oggetto, testo, dest fk)

↳ Foreign key (mittente) references utente (email);

sezioni (codice, titolo, descrizione*, amministratore fk)

↳ odd constraint titolo_unico unique (titolo)

↳ foreign key (amministratore) references admin (email);

sottosezioni (codice, titolo, descriz., moderatore fk)

↳ foreign key (moderatore) references mod (email);

topic (codice, titolo, data-c, stato)

alter table topic odd constraint stato check (stato in 'chiuso', 'aperto', 'evidenze', 'concessato');

tag (testo)

post (utente fk, data-c, ore-c, tags, testo, responsabile fk)

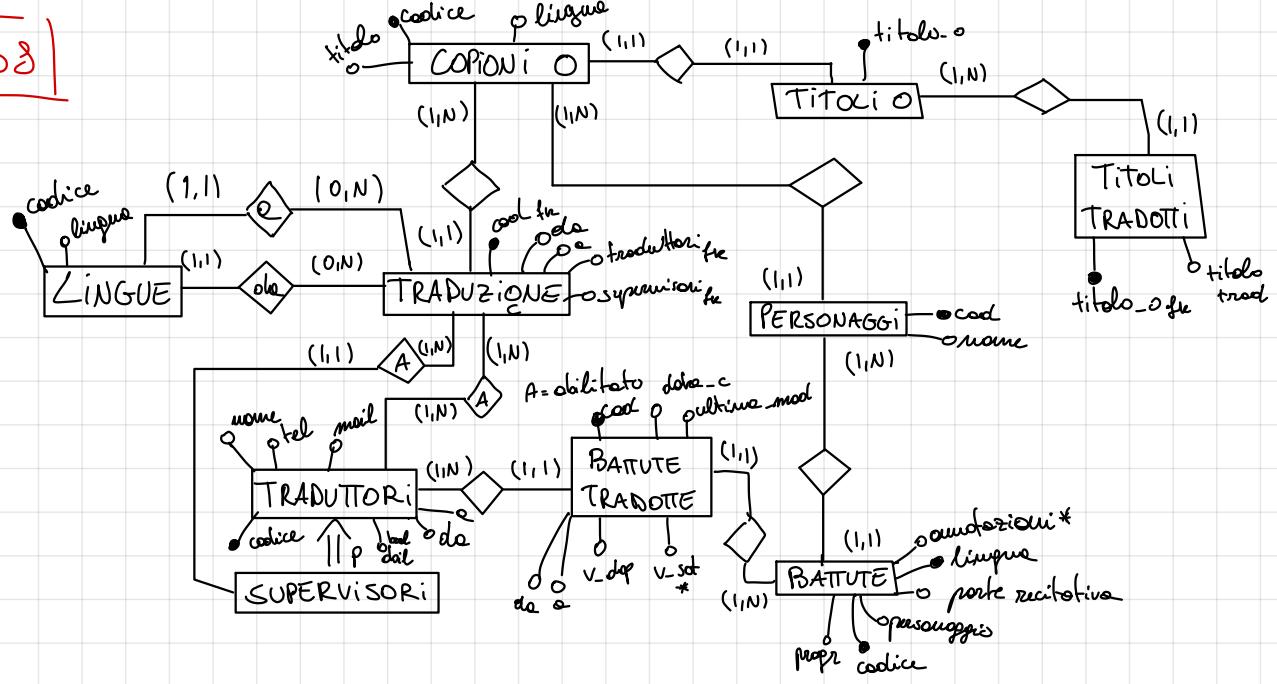
foreign key (responsabile) references utente (email)

lettura (utente fk, post, ultimo lett)

visualizzabile (utenti fk, sezioni fk, bool visualizz)

check responsabile in (select * from admin
union select * from mod)

22/06/2003



1) distingue (codice , manu - lingue)

2 Copie - o (codice, titolo su, lingue su)

titoli - o (titolo)

$\text{titoli_t} + (\underline{\text{titolo_of}}, \text{titolo_t})$

s copiorum - + (codice fm, de-l₁, o-l₂, superioribus, traductis fm)

abilitazioni ([traduttore, supervisore] fr, [de-l, o-l] fr)

traduttori (nome, codice, tel, mail, [do-l, o-l]fr)

Supervisor (codice ^{root}, nome, tel, mail, [de..l, a..l] file)

battute - o (codice, lingue fr., personaggi fr., recitativo, ammortazioni *)

• bottute-t (^{advice}_{fn} [de-l, e-l] fr, v.-drop, V-soft*, data-c, ultime-med)

personaggi (codice, nome)

ALGEBRA RECAZIONALE

Riepilogo:

- **Proiezione (Π)**: op. unario per estrarre colonne da una relazione $\Pi_{\text{lista-attributi}}(R)$
 → lo schema del risultato contiene i soli attributi contenuti in.
 la relazione risultato contiene le tuple ristrette agli attributi nella lista.
- **Selezione (σ)**: op. unario per selezionare righe da una relazione $\sigma_{\text{condizione}}(R)$
 expr. booleane tipo: attributo op costante o attributo, op attributo
 $OP = \{ <, \leq, >, \geq, =, \neq \}$ esponibile con conn. logici $\wedge, \vee, (\text{and}, \text{or})$
- **Unione (\cup)**: unione di due relazioni R_1 e R_2 contiene tutte le tuple (prese 1a volta) presenti in R_1 , R_2 o entrambi. Schema risultato = schema R_1 .
- **Intersezione (\cap)**: intersezione di due rel. R_1 e R_2 , contiene tutte le tuple (prese 1a volta) presenti contemporaneamente in R_1 e R_2 . Schema ris = schema R_1 .
- **Differenza ($-$)**: contiene le tuple presenti in R_1 ma non in R_2 . Sch.ris = sch R_1 .
- **Ridenominazione (ρ)**: $\rho(y \leftarrow x)(R) \quad \rho(R(F), E)$
 $E \rightarrow \text{expr. alg. rel. } R, \text{ istanza di rel.}$
 $F \rightarrow \text{lista di ridenom.}$
- **Prodotto cartesiano (\times)**: restituisce un'istanza di relazione il cui schema contiene tutti i campi di R_1 (ordine originale) seguiti da tutti i campi di R_2 . (err. ridenomin.)
- **Join condizionale (\bowtie_c)**: è una select fra R_1 e R_2 applicata sul prodotto cartesiano $R_1 \times R_2$.
- **Equi join (\bowtie_c)**: è un join in cui la condizione di join è composta solo da uguaglianze (eventuali AND). Lo schema risultato contiene tutti gli attributi di R_1 e gli attr. R_2 che non compaiono nelle cond di join.
- **Natural join (\bowtie)**: È un equijoin in cui le uguaglianze sono specificate su tutti gli attributi aventi lo stesso nome in R_1 e R_2 .

Props. natural join:

- Il join è commutativo e associativo $\rightarrow R_1 \bowtie R_2 = R_2 \bowtie R_1$ e $R_1(R_2 \bowtie R_3) = R_2(R_1 \bowtie R_3)$
- Se le due relazioni non hanno attributi in comune $R_1 \bowtie R_2 = \emptyset$ il join si riduce a prodotto cartesiano semplice $R_1 \times R_2$.

Viste:

Altro uso sono le espressioni a cui viene assegnato un nome: si possono usare le viste dentro altre espressioni > semplifica la scrittura di expr. complesse.
 sintassi: $V = E$ con V nome visto (non influenza su interoperabilità)

Massimo e minimo:

Min/Max assoluto: dato lo schema relazionale $R(A, B)$ trovare il min/max B in R .
→ determiniamo il minimo: $\pi_B(R) - \pi_B(R \bowtie_{B>B_1} (\rho(A_1, B_1 \leftarrow A, B)))$
Nella seconda parte vengono trovati tutti quei valori che non sono il minimo → per questo viene fatto un join fra R e se stesso, con tutti gli attributi rimanenenti.
↳ vengono tenute tutte le tuple tranne quelle in cui B assume il valore minore.

Min/Max relativo: trovare per ogni A il min/max B in R
supponiamo il max: $\pi_{A,B}(R) - \pi_{A,B}(R \bowtie_{A=A_1 \text{ and } B=B_1} (\rho(A_1, B_1 \leftarrow A, B)) (R))$

Ottimizzazione algebrica:

Ha lo scopo di trovare una espressione che sia equivalente a quella data, ma più efficiente.
Il costo può essere valutato in termini delle dimensioni di risultati intermedi.

Trasformazioni:

- 1) atomizzazione delle selezioni
- 2) idempotenza proiezioni
- 3) anticipazione della selezione proiezione rispetto al join

1) Una congiunzione di selezioni → sequenza di selezioni atomiche: $\sigma_{F_1} \text{ and } F_2(E) = \sigma_{F_2}(\sigma_{F_1}(E))$

2) Una proiezione può essere trasformata in una sequenza di proiezioni che eliminano i vari attributi in varie fasi: $\pi_A(E) \stackrel{\text{sost}}{=} \pi_A(\pi_{A,B}(E))$

3a) $\sigma_F(E_1 \bowtie E_2) = E_1 \bowtie \sigma(F)(E_2)$ Se la condizione F coinvolge attributi della sottoespressione E_2 .

3b) $\pi_{X_1, Y_2}(E_1 \bowtie E_2) = E_1 \bowtie \pi_{Y_2}(E_2)$ Combinare l'idempotenza delle proiezioni, permette di com E_1 su X_1 , E_2 su X_2 $Y_2 \subseteq X_2 \wedge (X_1 \cap X_2) \subseteq Y_2$ //: eliminare subito da ciascuna relazione gli attributi che non comparevano nel risultato e non sono coinvolti nel join.

ESERCIZI ALGEBRA RELAZIONALE:

Dal DB costituito nel modo seguente:

(1)

LIBRI (codice-libro, autore, titolo)

UTENTI (codice-utente, nome, cognome)

PRESTITI (codice-utente, codice-libro, data-prestito)

Si esprimono le seguenti interrogazioni:

- 1) titoli dei libri presi in prestito il 13/07/2021
- 2) Autori dei libri presi in prestito da Paolo Bianchi.
- 3) Codici degli utenti che hanno preso in prestito libri scritti da C oppure Z
- 4) titoli dei libri presi in prestito da Paolo Bianchi il 13/07/2021
- 5) Gli utenti che hanno preso in prestito solo libri di Camilleri.

1) $S_1 = \sigma_{data_prestito=13/07/2021} (Prestiti) \quad //: \pi_{titolo} (S_1 \bowtie Libri)$
 oppure direttamente $\pi_{titolo} (\sigma_{data_prestito=13/07/2021} (Prestiti) \bowtie Libri)$

2) $S_1 = \sigma_{nome='Paolo' \text{ AND } cognome='Bianchi'} (Utenti)$
 $S_2 = S_1 \bowtie Prestiti \quad S_3 = S_2 \bowtie Libri \quad \pi_{autore} (S_3)$

3) $S_1 = SEL_{autore} = "camilleri" \text{ OR autore } "de luca" (Libri)$
 $S_2 = S_1 \bowtie Prestiti$
 $\pi_{codice_utente} (S_2)$

4) $S_1 = \sigma_{nome='Paolo' \text{ AND } cognome='Bianchi'} (Utenti)$
 $S_2 = \sigma_{data_prestito=13/07/2021} (Prestiti)$
 $S_3 = \pi_{codice_libro} (S_1 \bowtie S_2)$
 $S_4 = S_3 \bowtie Libri$
 $\pi_{sel} (S_4)$

5) $S_1 = \sigma_{autore} = "Camilleri" (Libri)$ $T_1 = \sigma_{autore = 'Camilleri'} (Libri)$
 $S_2 = \pi_{codice_libro} (S_1)$ $T_2 = \pi_{codice_libro} (T_1)$
 $S_3 = S_2 \bowtie Prestiti$ $T_3 = T_2 \bowtie (Prestiti)$
 $S_4 = \pi_{codice_utente} (S_3)$ $T_4 = \pi_{codice_utente} (T_3)$
 $S_4 - T_4$

(2)

CD (codice_col, autore, titolo)

Clienti (codice-cliente, nome, cognome, codice_fiscale)

Noleggi (codice-cliente, codice_cd, data)

- 1) Autore / titolo CD noleggiati da P.B il 21/02/2021
- 2) Nome e cognome clienti che hanno noleggiato CD rodolpheos il 12/01/2021
- 3) Titolo CD noleggiati da cod cli C123 o G123
- 4) Clienti con stesso nome ma \neq cod_fiscale

1) $S_1 = \text{Noleggi} \bowtie \text{Clienti}$
 $S_2 = \sigma_{\text{data_noleggio} = 20/11/2021} \text{AND nome} = "P" \text{ AND cognome} = "B" (S_1)$
 $S_3 = \pi_{\text{codice_cd}} (S_2)$
 $\Pi_{\text{autore}, \text{titolo}} (S_3 \bowtie \text{CD})$

2) $S_1 = \sigma_{\text{data_noleggio} = 20/11/2021} (\text{Noleggio})$
 $S_2 = \sigma_{\text{autore} = "Radiohead"} (\text{CD})$
 $S_3 = \pi_{\text{codice_cli}} (S_1 \bowtie S_2)$
 $\Pi_{\text{nome}, \text{cognome}} (S_3 \bowtie \text{Clienti})$

3) $\Pi_{\text{titolo}} (\sigma_{\text{codice_cli} = "G123" \text{ OR codice_cli} = "C123"} (\text{Noleggio}) \bowtie \text{CD})$

4) Ridenominiamo la tabella clienti:
 \downarrow
 $p(\text{clienti}_1 (\text{nome}_1 \leftarrow \text{nome}, \text{cognome}_1 \leftarrow \text{cognome}, \text{codice_f}_1 \leftarrow \text{cf}), \text{clienti})$
 $\Pi_{\text{nome}, \text{cognome}, \text{codice_f}} (\text{Clienti} \bowtie \text{clienti}. \text{nome} = \text{clienti}_1. \text{nome}_1 \text{ AND}$
 $\text{clienti}. \text{cognome} = \text{clienti}_1. \text{cognome}_1 \text{ AND codice_f} = \text{codice_f}_1 (\text{Clienti}_1)$

(3) Viaggiatori (codice_viaggiatore, nome, cognome)
treni (codice_treno, provenienza, destinazione)
prenotazioni (codice_viagg, codice_tre, data)

- 1) Nome e cognome dei viaggiatori che il 10/11/2010 hanno prenotato posti su treni Pisa \rightarrow Roma
- 2) Elenco delle date in cui i viaggiatore di nome Marco hanno prenotazioni.
- 3) Provenienza e destinazione di treni sui cui c'è almeno una prenotazione.

1) $S_1 = \text{SEL data} = "10/11/2010" \text{ from (prenotazioni)}$
 $S_2 = \text{Sel provenienza} = "Pisa" \text{ AND destinazione} = "Roma" \text{ from (treni)}$
 $S_3 = \pi_{\text{codice_viaggiatore}} (S_1 \bowtie S_2)$
 $S_4 = S_3 \bowtie \text{Viaggiatori}$
 $\Pi_{\text{nome}, \text{cognome}} (S_4)$

2) $S_1 = \text{SEL cognome} = "bianchi" \text{ from viaggiatori}$
 $S_2 = \Pi_{\text{data}} (S_1 \bowtie \text{prenotazioni})$

3) $\Pi_{\text{provenienza}, \text{destinazione}} (\text{treni} \bowtie \text{prenotazioni})$

(4) Clienti (codice_cliente, nome, cognome)
Film (codice_film, anno, genere)
Noleggio (codice_cliente, codice_film, data)

- 1) Nome e cognome dei clienti che hanno noleggiato film di fantascienza
- 2) titolo e anno di film di fantascienza
- 3) titolo di film già noleggiati da Paolo bianchi
- 4) cognome clienti che in data 10/12/2020 hanno noleggiato film girati nel 1965

1) $\Pi \text{ nome, cognome } ((\sigma_{\text{genere} = \text{'fantascienza'}}(\text{film}) \bowtie \text{Noleggio}) \bowtie \text{Clienti})$

2) $\Pi \text{ titolo, anno } (\sigma_{\text{genere} = \text{'fantascienza'}}(\text{film}))$

3) $S_1 = \sigma_{\text{genere} = \text{'giallo'}}(\text{film})$

$S_2 = \sigma_{\text{cognome} = \text{"Bianchi"} \text{ and nome} = \text{"Paolo}}(\text{clienti})$

$S_3 = \Pi \text{ codice_cliente } (S_1 \bowtie \text{Noleggio})$

$S_4 = S_3 \bowtie S_2$

$\Pi \text{ titolo}(S_4)$

4) $S_1 = \sigma_{\text{data} = 10/11/11}(\text{Noleggio})$

$S_2 = \sigma_{\text{genere} = \text{"Fantascienza"} \text{ OR anno} = 1965}(\text{film})$

$S_3 = \Pi \text{ codice_cliente } (S_1 \bowtie S_2)$

$\Pi \text{ cognome}(S_3 \bowtie \text{Clienti})$

(5)

Esoni (codice-exame, materia, professore)

Aule (codice-aule, nome, edificio, copertura)

Prenotazioni (codice-aule, codice-exame, data)

1) Edificio e nome delle aule prenotate per gli esami di informatica il 10/11/2000

2) Nome e copertura delle aule prenotate per gli esami del prof. Bianchi

3) Edificio e nome aule che hanno almeno 100 posti e hanno prenotazioni 10/11/2000

4) Sime ma non prenotate

5) Nome e edificio aule prenotate il 10/12/2000

1) $\Pi \text{ edificio, nome } ((\sigma_{\text{materia} = \text{"informatica}}(\text{Esame}) \bowtie \sigma_{\text{data} = "10/10/10"}(\text{Prenotazioni})) \bowtie (\text{Aule}))$

2) $\Pi \text{ nome, copertura } (\sigma_{\text{professore} = \text{"Bianchi}}(\text{Esoni}) \bowtie \text{Prenotazioni}) \bowtie \text{Aule})$

3) $\Pi \text{ edificio, nome } (\sigma_{\text{copertura} \geq 100}(\text{Aule}) \sigma_{\text{data} = 10/11}(\text{Prenotazioni}))$

4) $\Pi \text{ edificio, nome } (\sigma_{\text{copertura} \geq 100}(\text{Aule})) - \Pi \text{ edificio, nome } (\sigma_{\text{data} = "10/11"}(\text{Prenotazioni}))$

5) $\Pi \text{ nome, edificio } (\sigma_{\text{data} = 10/12}(\text{Prenotazioni})) \bowtie \text{Aule}$

Codificare un vincolo di chiave primaria (rel musei (nome, città))

proj

M1 (nome1, nome2) : $\Pi \{ \text{nome}, \text{città} \} (\text{musei})$

M2 (nome2, città2) := M1

$M1 \bowtie (\text{nome1} = \text{nome2} \text{ AND } \text{città1} = \text{città2}) \quad M2 = \{ \}$

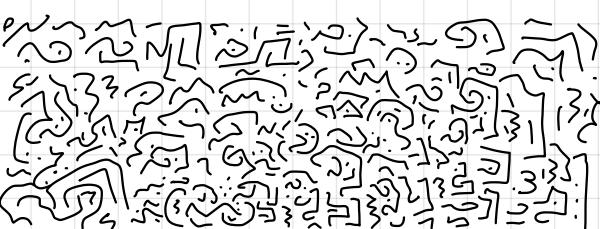
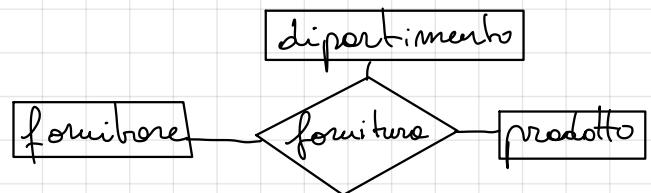
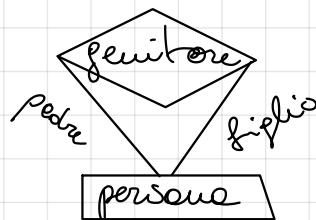
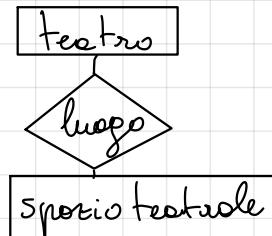


Diagramma Entità-Relazione

Entità: rappresentano concetti complessi e di rilievo che descrivono classi di oggetti con esistenza autonoma:

I teatro

Relazioni: sono legami logici, tra due o più entità.



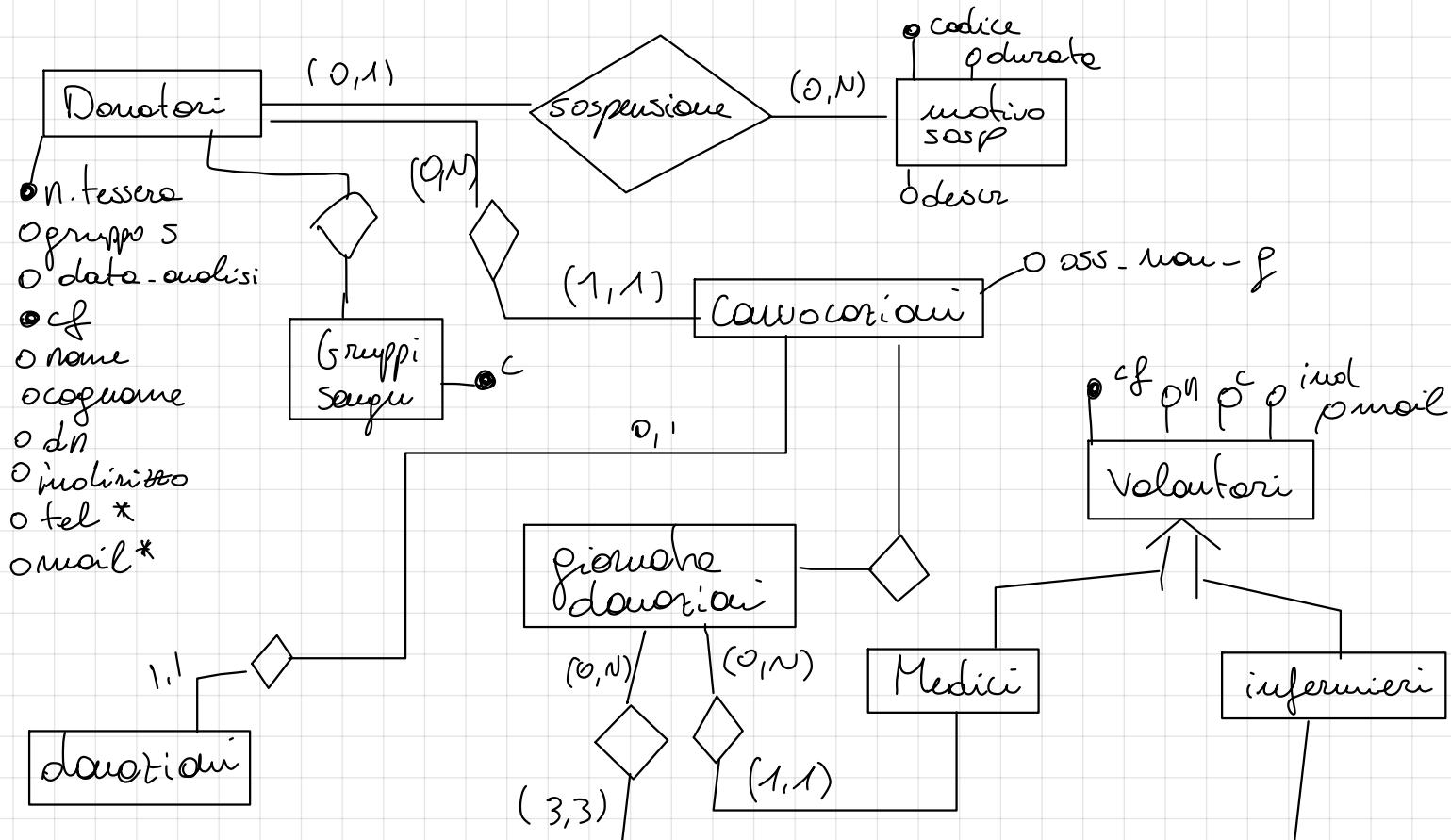
Cardinalità:



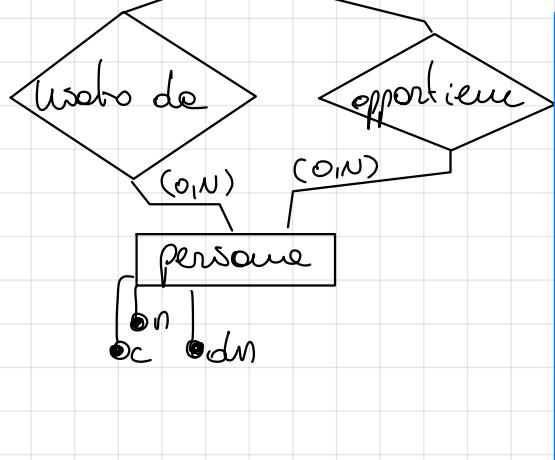
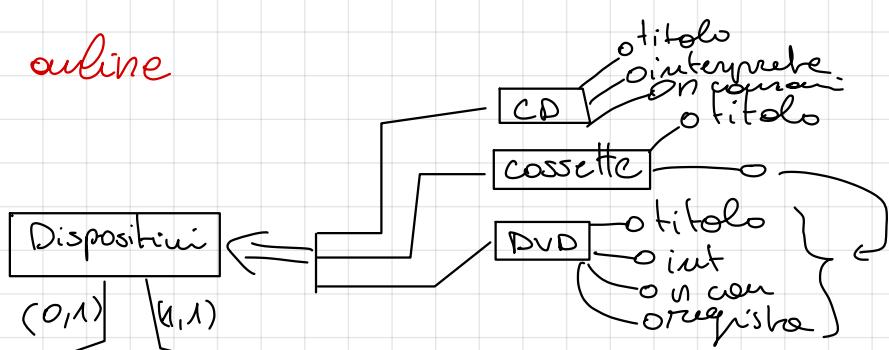
● a quale uni può iscriversi un studente?

● quanti studenti possono iscriversi all'università?

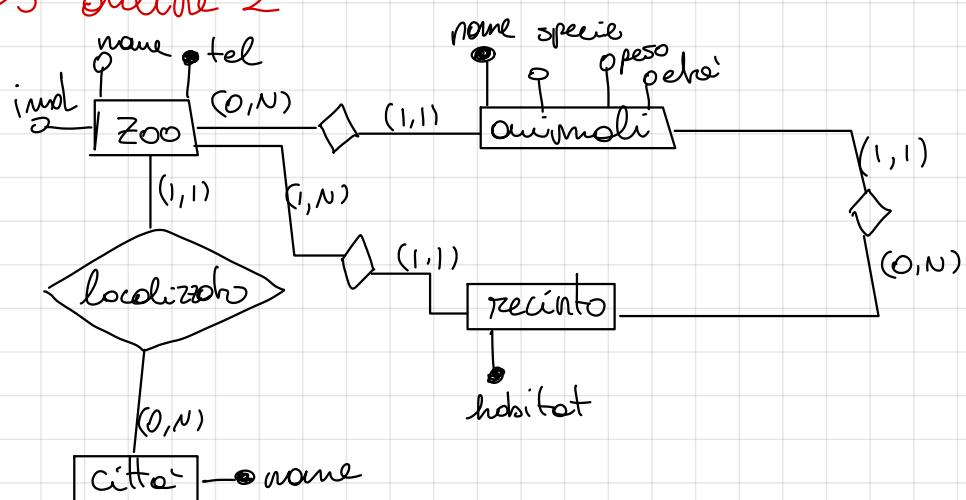
ER - 13/6/2005



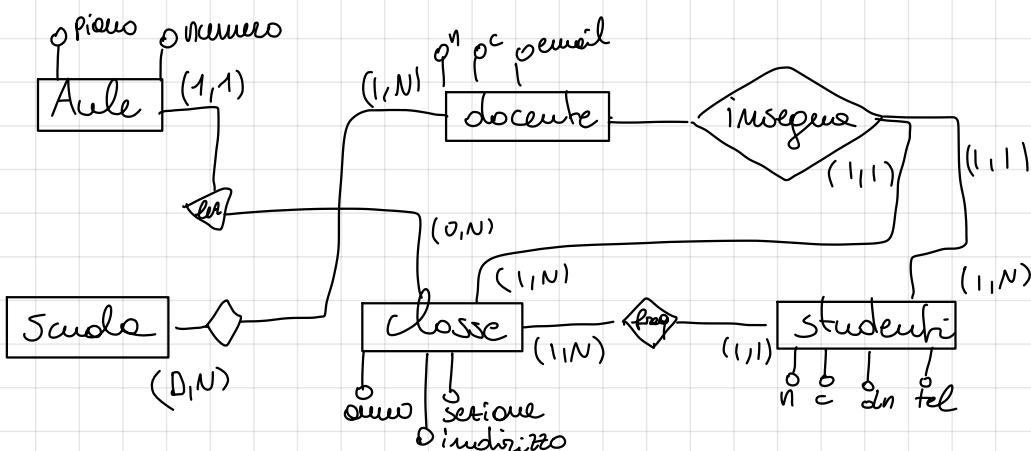
Es online



Es online 2



Es online 3



Note:

Musei (name, citta)

Artisti (name, e_n, e_um*, nazionalita*)

Opere (codice, artista*, titolo, anno, museo_fn, e, l, p*)

prestiti (opera_fn, museo_fn, date_iniz, date_fi*)

Restauri (opera_fn, d_i, d_f)