

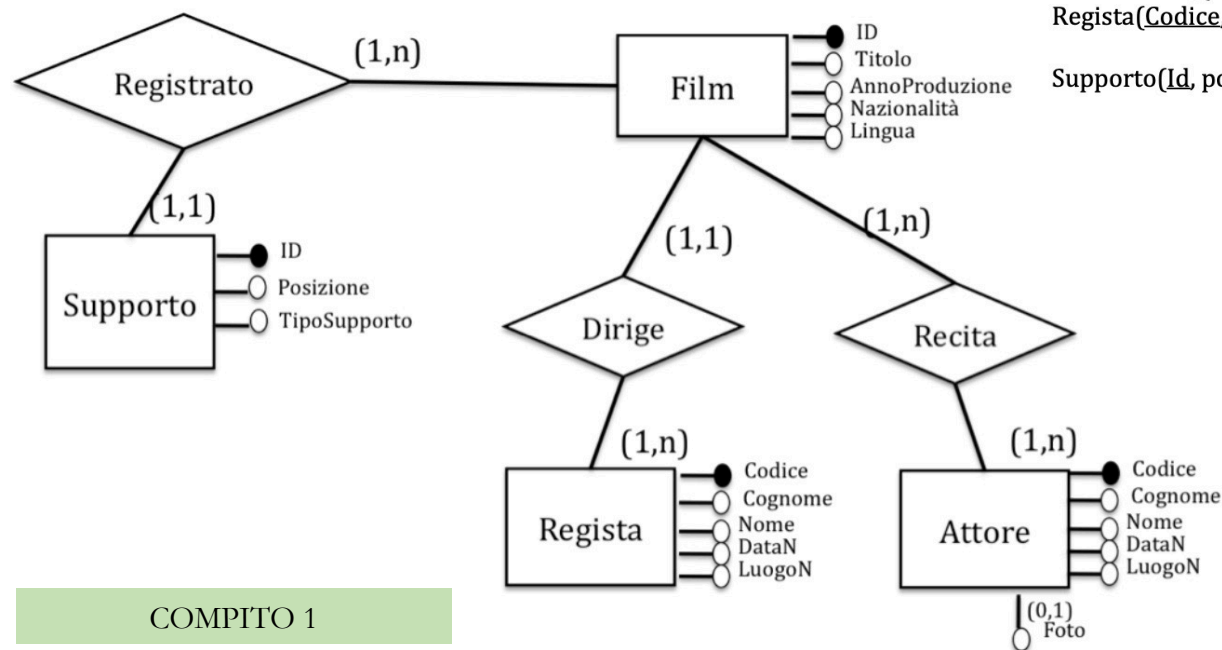
SOLUZIONI

Prof. Rosario Sorbello

COMPITI BASI DI DATI DEL 17 SETTEMBRE 2021

Gestire il proprio archivio di film. I film sono registrati su un supporto (su DVD). Di ogni supporto si memorizza la posizione nella propria videoteca. Ogni supporto può contenere un solo film. Di un film si tiene traccia del titolo, dell'anno di produzione, della nazionalità e della lingua. Un film è interpretato da attori ed è diretto da un regista. A un film possono partecipare uno o più attori. Di attori e registi si memorizza il codice, il cognome, il nome, la data e il luogo di nascita. Degli attori si può memorizzare inoltre anche una foto (non obbligatoriamente).

Schema Entità-Associazione:



Schema database relazionale:

Film(Id, titolo, annoproduzione, nazionalità, lingua, CodiceRegista)

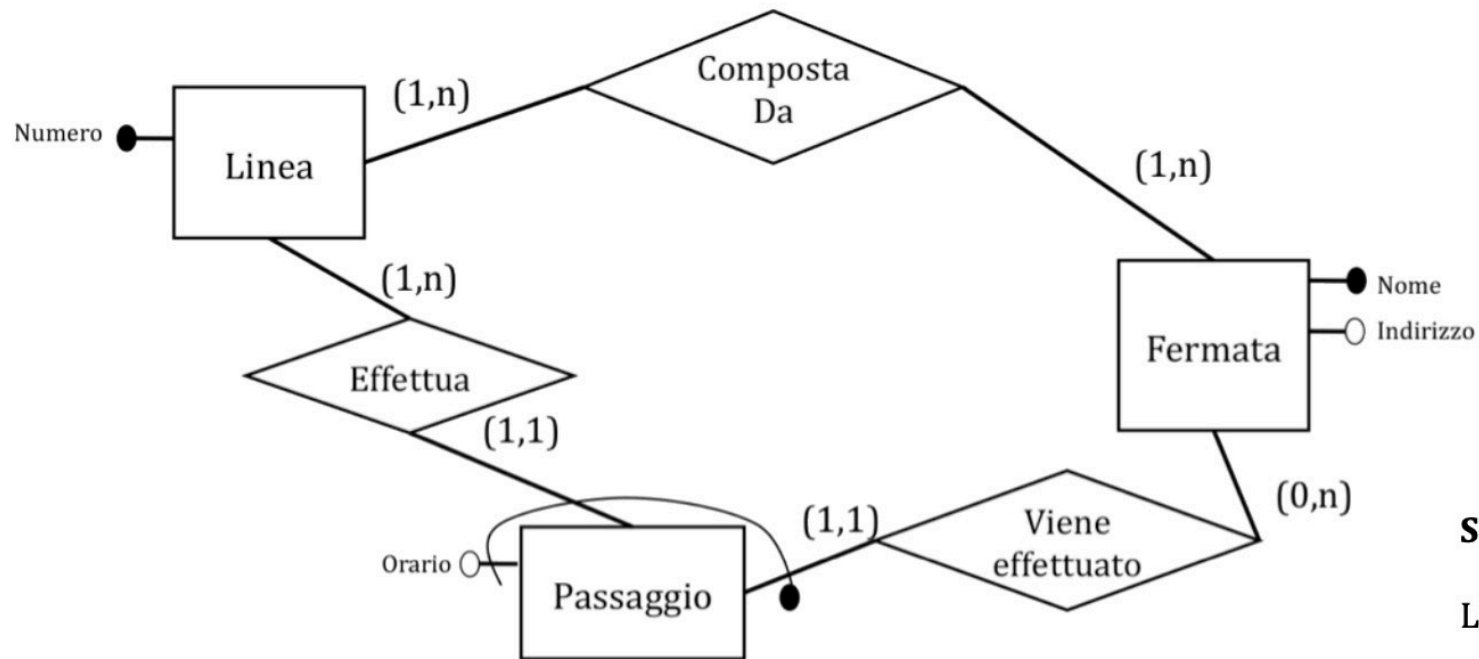
Attore(Codice, cognome, nome, dataN, luogoN, foto) (nota: foto può assumere valori nulli)

Recita(IdFilm, CodiceAttore)

Regista(Codice, cognome, nome, dataN, luogoN)

Supporto(Id, posizione, tiposupporto, IdFilm)

GESTIRE UNA LINEA DI TRASPORTO PUBBLICO DI AUTOBUS. UNA LINEA DI TRASPORTO PUBBLICO È CARATTERIZZATA DA UN NUMERO ED È COMPOSTA DA PIÙ FERMATE. DI OGNI FERMATA SI MEMORIZZA IL NOME E L'INDIRIZZO. TENIAMO INOLTRE TRACCIA DEI PASSAGGI. UNA LINEA EFFETTUA UN PASSAGGIO A UNA FERMATA IN DETERMINATI ORARI. SI PRESTI ATTENZIONE A NON TRASCURARE IL FATTO CHE UNA LINEA PUÒ EFFETTUARE PIÙ PASSAGGI ALLA MEDESIMA FERMATA.



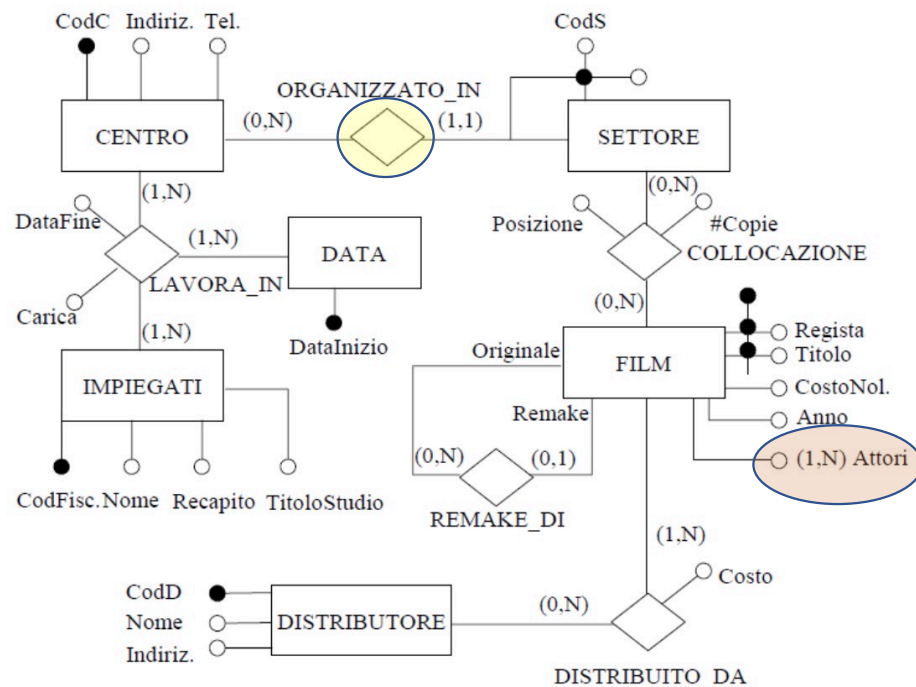
Schema database relazionale:

Linea(Numero)
CompostaDa(Linea, Fermata)
Fermata(Nome, Indirizzo)
Passaggio(Linea, Fermata, Orario)

COMPITO 1

DIVENTA RELAZIONE PERCHE' ATTRIBUTO MULTI-VALORE

NON SI TRADUCE



Schema logico relazionale normalizzato

IMPIEGATI (CodFisc., Nome, Recapito, TitoloStudio)

CENTRO (CodC, Indiriz., Tel.)

DATA (DataInizio)

LAVORA_IN (CodC, CodFisc., DataInizio, DataFine, Carica)

SETTORE (CodC, CodS)

FILM (Regista, Titolo, CostoNol., Anno)

ATTORI (Regista, Titolo, Attore)

COLLOCAZIONE (CodC, CodS, Regista, Titolo, #Copie, Posizione)

REMAKE_DI (RegistaRemake, TitoloRemake, RegistaFilmOriginale, TitoloFilmOriginale)

DISTRIBUTORE (CodD, Nome, Indiriz.)

DISTRIBUITO_DA (Regista, Titolo, CodD, Costo)

Altre traduzioni possibili del Diagramma E-R

La relazione REMAKE_DI poteva anche essere così tradotta:

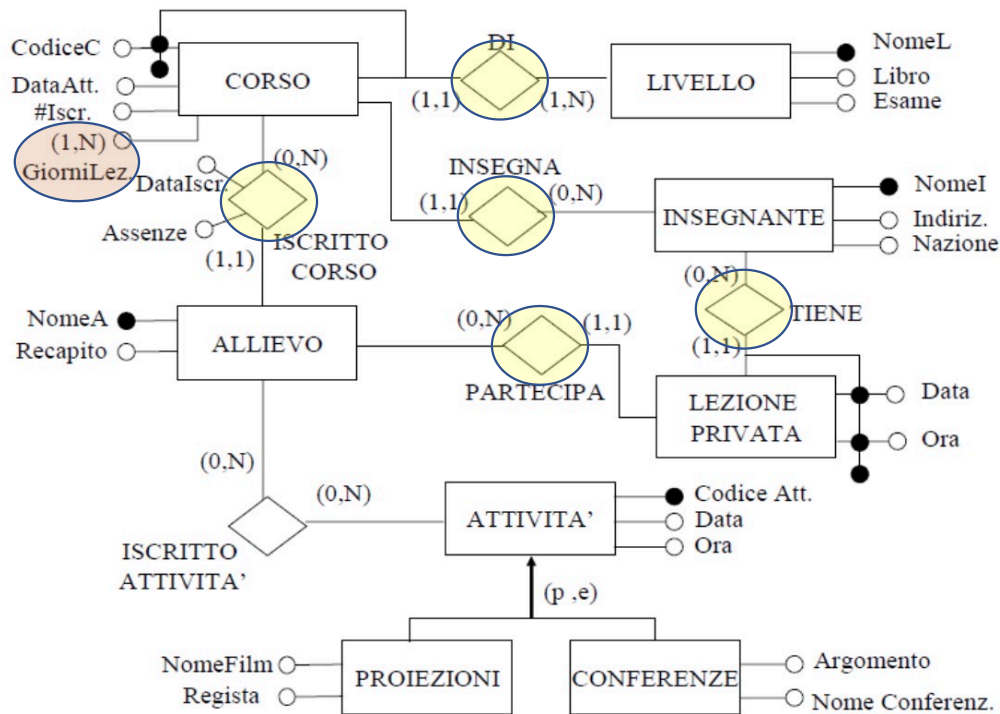
FILM (Regista, Titolo, CostoNol., Anno, RegistaFilmOriginale*, TitoloFilmOriginale*)

Questo tipo di traduzione è preferibile qualora la maggior parte dei film contenuti nella base di dati disponga di una versione remake.

COMPITO 2

DIVENTA RELAZIONE PERCHE' ATTRIBUTO MULTI-VALORE

NON SI TRADUCE



Schema logico relazionale normalizzato

LIVELLO (NomeL, Libro, Esame)

INSEGNANTE (NomeI, Indiriz., Nazione)

CORSO (NomeL, CodiceC, DataAtt., #Iscr., NomeI)

GIORNI LEZ. (NomeL, CodiceC, Giorno)

ALLIEVO (NomeA, Recapito, NomeL, CodiceC, DataIscr., Assenze)

LEZIONE PRIVATA (NomeI, Data, Ora, NomeA)

ATTIVITA' (Codice Att., Data, Ora)

ISCRITTO ATTIVITA' (NomeA, Codice Att.)

PROIEZIONI (Codice Att. Proiez., NomeFilm, Regista)

CONFERENZE (Codice Att. Conf., Argomento, Nome Conferenz.)

EREDITARIETA' SI
CONVERTE IN UN
RAPPORTO DI RELAZIONE

COMPITO 1

A	B	C	D	E
6	2	3	4	5
6	2	2	5	7
6	2	7	8	0
6	3	3	4	5
6	2	6	6	8
7	3	2	5	7
7	3	7	8	7
7	4	7	8	2
7	5	7	7	3
7	5	3	3	2
7	5	7	2	8
7	6	8	0	8
7	6	7	7	0

A	B	C	D	E
6	2	3	4	5
6	2	2	5	7
6	2	7	8	0
6	3	3	4	5
6	2	6	6	8
7	3	2	5	7
7	3	7	8	7
7	4	7	8	2
7	5	7	7	3
7	5	3	3	2
7	5	7	2	8
7	6	8	0	8
7	6	7	7	0

(A,B)= (6,2), (6,3), (7,3), (7,4), (7,5) e (7,6)

GRUPPI RIMASTI DOPO LA SELEZIONE

DOPO LA SELEZIONE

DOPO IL GROUP BY

```
select B, max(E)
from R
where C >= 3
group by A, B
having sum(D) > 7
```

DOPO L'HAVING

(A,B)=

(6,2): 3 tuple - max(e=(5,0,8))= **8**

(6,3): 1 tupla - max(e=(5))= **5**

(7,3): 1 tupla - max(e=(7))= **7**

(7,4): 1 tupla - max(e=(2))= **2**

(7,5): 3 tuple - max(e=(3,2,8))= **8**

(7,6): 2 tuple - max(e=(8,0))= **8**

B	MAX(E)
2	8
3	5
3	7
4	2
5	8
6	8

(A,B)=

– sum(d)= 4+8+6 =18

– sum(d)= 4 **NO**

– sum(d)= 8

– sum(d)= 8

– sum(d)= 7+3+2 =12

– sum(d)= 0+7 =7 **NO**

B	MAX(E)
2	8
3	7
4	2
5	8

RISULTATO FINALE

COMPITO 2

R					
A	B	C	D	E	
2	1	3	4	5	
2	1	2	5	7	
2	1	6	6	8	
2	1	7	8	0	
2	1	3	3	2	
3	1	3	4	5	
3	1	2	5	7	
3	1	7	8	1	
7	1	7	8	2	
8	1	7	7	3	
8	1	7	0	8	
9	1	7	7	0	
9	1	8	0	8	

R	A	B	C	D	E
2	1	3	4	5	
2	1	2	5	7	
2	1	6	6	8	
2	1	7	8	0	
2	1	3	3	2	
3	1	3	4	5	
3	1	2	5	7	
3	1	7	8	1	
7	1	7	8	2	
8	1	7	7	3	
8	1	7	0	8	
9	1	7	7	0	
9	1	8	0	8	

(A,B)= (2,1), (3,1), (7,1), (8,1) e (9,1)

GRUPPI RIMASTI DOPO LA SELEZIONE

11 su 13 tuple

DOPO LA SELEZIONE

```
select a, sum(e)
from r
where c > 2
group by a, b
having count (distinct c) > 1;
```

DOPO IL GROUP BY

DOPO L'HAVING

(A,B)=

(2,1): 4 tuple - sum(e)= 15 – count(distinct c)= 3

(3,1): 2 tuple - sum(e)= 6 – count(distinct c)= 2

(7,1): 1 tupla - sum(e)= 2 – count(distinct c)= 1 NO >1

(8,1): 2 tuple - sum(e)= 11 – count(distinct c)= 1 NO >1

(9,1): 2 tuple - sum(e)= 8 – count(distinct c)= 2

A	SUM(E)
2	15
3	6
7	2
8	11
9	8

A	SUM(E)
2	15
3	6
9	8

RISULTATO FINALE

PRODOTTO(COD_P, NOME_P, COLORE, TAGLIA, SEDE_MAGAZZINO)

FORNITORE(COD_F, NOME_F, NUM_SOCI, SEDE_FORNITORE)

FORNISCE(REF_F, REF_P, QUANTITA')

PRODOTTO				
(COD_P, NOME_P, COLORE, TAGLIA, SEDE_MAGAZZINO)				
P1	Maglia	Rosso	40	Torino
P2	Jeans	Verde	48	Milano
P3	Camicia	Blu	48	Roma
P4	Camicia	Blu	44	Torino
P5	Gonna	Blu	40	Milano
P6	Bermuda	Rosso	42	Torino

FORNISCE		
(REF_F, REF_P, QUANTITA')		
F1	P1	300
F1	P2	200
F1	P3	400
F1	P4	200
F1	P5	100
F1	P6	100
F2	P1	300
F2	P2	400
F3	P2	200
F4	P3	200
F4	P4	300
F4	P5	400

FORNITORE			
(COD_F, NOME_F, NUM_SOCI, SEDE_FORNITORE)			
F1	Andrea	2	Torino
F2	Luca	1	Milano
F3	Antonio	3	Milano
F4	Gabriele	2	Torino
F5	Matteo	3	Venezia

PRODOTTO(COD_P, NOME_P, COLORE, TAGLIA, SEDE_MAGAZZINO)

FORNITORE(COD_F, NOME_F, NUM_SOCI, SEDE_FORNITORE)

FORNISCE(REF_F, REF_P, QUANTITA')

DEFINIRE LA VISTA *PICCOLI_FORNITORI*, CHE RESTITUISCE IL CODICE, IL NOME, IL NUMERO DI SOCI E LA SEDE DEI FORNITORI CHE HANNO MENO DI 3 SOCI.

```
CREATE VIEW PICCOLI_FORNITORI AS
SELECT F.COD_F, F.NOME_F, F.NUM_SOCI, F.SEDE
FROM FORNITORE F
WHERE F.NUM_SOCI < 3;
```

USARE LA VISTA *PICCOLI_FORNITORI* PER SCRIVERE UNA INTERROGAZIONE SQL CHE MI RESTITUISCE I PICCOLI FORNITORI CON NUMERO DI SOCI PARI A 2 E I PRODOTTI CHE FORNISCONO CON UNA QUANTITA' MAGGIORE DI 150

```
SELECT CODF, NOMEF, COD_P, NOME_P
FROM PICCOLI_FORNITORI PF, FORNISCE FO, PRODOTTO PR
WHERE PF.COD_F=FO.REF_F AND FO.REF_P=PR.COD_P AND PF.NUM_SOCI=2 AND QUANTITA'>150
```

PRODOTTO				
(COD_P, NOME_P, COLORE, TAGLIA, SEDE_MAGAZZINO)				
P1	Maglia	Rosso	40	Torino
P2	Jeans	Verde	48	Milano
P3	Camicia	Blu	48	Roma
P4	Camicia	Blu	44	Torino
P5	Gonna	Blu	40	Milano
P6	Bermuda	Rosso	42	Torino

FORNISCE		
(REF_F, REF_P, QUANTITA')		
F1	P1	300
F1	P2	200
F1	P3	400
F1	P4	200
F1	P5	100
F1	P6	100
F2	P1	300
F2	P2	400
F3	P2	200
F4	P3	200
F4	P4	300
F4	P5	400

FORNITORE			
(COD_F, NOME_F, NUM_SOCI, SEDE_FORNITORE)			
F1	Andrea	2	Torino
F2	Luca	1	Milano
F3	Antonio	3	Milano
F4	Gabriele	2	Torino
F5	Matteo	3	Venezia

PRODOTTO(COD_P, NOME_P, COLORE, TAGLIA, SEDE_MAGAZZINO)

FORNITORE(COD_F, NOME_F, NUM_SOCI, SEDE_FORNITORE)

FORNISCE(REF_F, REF_P, QUANTITA')

DEFINIRE LA VISTA *FORNITORI_PRODOTTO* CHE MI RESTITUISCE
PER OGNI PRODOTTO IL CODICE E IL NOME DEL PRODOTTO
E IL NUMERO DI FORNITORI DIVERSI CHE L'HANNO FORNITO.

```
CREATE VIEW FORNITORI_PRODOTTO (COD_P, NOME_P, NUM_FORNITORI)
AS SELECT COD_P, NOME_P, COUNT(*)
FROM PRODOTTO P, FORNISCE F
WHERE P.COD_P=F.REF_P
GROUP BY COD_P, NOME_P;
```

USARE LA VISTA *FORNITORI_PRODOTTO* PER VISUALIZZARE IL CODICE E
IL NOME DEI PRODOTTI FORNITI DAL MAGGIOR NUMERO DI FORNITORI

```
SELECT COD_P, NOME_P
FROM FORNITORI_PRODOTTO FP1
WHERE FP1.NUMFORNITORI= (SELECT MAX(FP2.NUMFORNITORI)
                        FROM FORNITORI_PRODOTTO FP2);
```

PRODOTTO				
(COD_P, NOME_P, COLORE, TAGLIA, SEDE_MAGAZZINO)				
P1	Maglia	Rosso	40	Torino
P2	Jeans	Verde	48	Milano
P3	Camicia	Blu	48	Roma
P4	Camicia	Blu	44	Torino
P5	Gonna	Blu	40	Milano
P6	Bermuda	Rosso	42	Torino

FORNISCE		
(REF_F, REF_P, QUANTITA')		
F1	P1	300
F1	P2	200
F1	P3	400
F1	P4	200
F1	P5	100
F1	P6	100
F2	P1	300
F2	P2	400
F3	P2	200
F4	P3	200
F4	P4	300
F4	P5	400

FORNITORE			
(COD_F, NOME_F, NUM_SOCI, SEDE_FORNITORE)			
F1	Andrea	2	Torino
F2	Luca	1	Milano
F3	Antonio	3	Milano
F4	Gabriele	2	Torino
F5	Matteo	3	Venezia

INGEGNERE(**COD_I**, NOME_I, COGNOME_I, INDIRIZZO, PARTITA_IVA, CITTÀ_I)

AZIENDA(**COD_A**, NOME_A, TIPOLOGIA_AZIENDA, CITTÀ_A, SITOWEB)

SETTORE(**COD_S**, NOME_S, DESCRIZIONE)

CONSULENZE EFFETTUATE(**REF_I**, **REF_A**, **DATA**, *REF_S*, IMPORTO)

L'ATTRIBUTO TIPOLOGIAAZIENDA = {MICROIMPRESA, PICCOLAIMPRESA, MEDIAIMPRESA, GRANDEIMPRESA}

COMPITO 1

ESPRIMERE LE SEGUENTI INTERROGAZIONI IN SQL O IN ALGEBRA A SECONDA DELLA DOMANDA:

1. **ALGEBRA:** VISUALIZZARE IL CODICE, IL NOME E IL COGNOME DI CIASCUN INGEGNERE CHE HA EFFETTUATO CONSULENZE PRESSO *ALMENO DUE AZIENDE* DI TIPOLOGIA MEDIA IMPRESA.

PROJ_{COD_I, NOME_I, COGNOME_I} (INGEGNERE JOIN_{I.COD_I=CE1.REF_I} (SEL_{CE1.REF_A1<>CE2.REF_A2}
((SEL_{TIPOLOGIA_AZIENDA="MEDIA IMPRESA"} AZIENDA1) JOIN_{COD_A1=REF_A1} CONSULENZE_EFFETTUATE CE1)
JOIN_{CE1.REF_I=CE2.REF_I}
((SEL_{TIPOLOGIA_AZIENDA="MEDIA IMPRESA"} AZIENDA2) JOIN_{COD_A2=REF_A2} CONSULENZE_EFFETTUATE CE2))))

INGEGNERE(**COD_I**, NOME_I, COGNOME_I, INDIRIZZO, PARTITA_IVA, CITTÀ_I)

AZIENDA(**COD_A**, NOME_A, TIPOLOGIA_AZIENDA, CITTÀ_A, SITOWEB)

SETTORE(**COD_S**, NOME_S, DESCRIZIONE)

CONSULENZE EFFETTUATE(**REF_I, REF_A, DATA**, REF_S, IMPORTO)

COMPITO 1

L'ATTRIBUTO TIPOLOGIAAZIENDA = {MICROIMPRESA, PICCOLAIMPRESA, MEDIAIMPRESA, GRANDEIMPRESA}

ESPRIMERE LE SEGUENTI INTERROGAZIONI IN SQL O IN ALGEBRA A SECONDA DELLA DOMANDA:

2. ALGEBRA : VISUALIZZARE IL CODICE, IL NOME E IL COGNOME DI CIASCUN INGEGNERE CHE HA EFFETTUATO CONSULENZE NEL 2021 *PER TUTTE* LE AZIENDE DI TIPOLOGIA MEDIA IMPRESA DI PALERMO.

PROJ **cod_I**, NOME_I, COGNOME_I (INGEGNERE JOIN $COD_I=REF_I$

(PROJ REF_I, REF_A (SEL $CE.DATA \geq "1/1/2021"$ AND $CE.DATA \leq "31/12/2021"$ (CONSULENZE_EFFETTUATE))

÷

(REN $REF_A < -COD_A$ (PROJ COD_A (SEL $CITTA_A = "PALERMO"$ AND $TIPOLOGIA_AZIENDA = "MEDIA IMPRESA"$ AND $E.DATA \geq "1/1/2021"$ AND $CE.DATA \leq "31/12/2021"$ (AZIENDA))))))

INGEGNERE (COD_I, NOME_I, COGNOME_I, INDIRIZZO, PARTITA_IVA, CITTÀ_I)

AZIENDA (COD_A, NOME_A, TIPOLOGIA_AZIENDA, CITTÀ_A, SITOWEB)

SETTORE (COD_S, NOME_S, DESCRIZIONE)

CONSULENZE EFFETTUATE (REF_I, REF_A, DATA, REF_S, IMPORTO)

COMPITO 1

L'ATTRIBUTO TIPOLOGIAAZIENDA = {MICROIMPRESA, PICCOLAIMPRESA, MEDIAIMPRESA, GRANDEIMPRESA}

ESPRIMERE LE SEGUENTI INTERROGAZIONI IN SQL O IN ALGEBRA A SECONDA DELLA DOMANDA:

2. **SQL:** VISUALIZZARE IL CODICE, IL NOME E IL COGNOME DI CIASCUN INGEGNERE CHE HA EFFETTUATO CONSULENZE NEL 2021 *PER TUTTE* LE AZIENDE DI TIPOLOGIA MEDIA IMPRESA DI PALERMO.

SELECT COD_I, NOME_I, COGNOME_I

FROM INGEGNERE I

WHERE NOT EXISTS

(SELECT *

FROM AZIENDA A

WHERE A.TIPOLOGIA_AZIENDA="MEDIA IMPRESA" AND A.CITTÀ_A =" PALERMO"

AND NOT EXISTS

(SELECT *

FROM CONSULENZE_EFFETTUATE CE

WHERE CE.REF_I=I.COD_I AND CE.REF_A=A.COD_A AND

CE.DATA>="1/1/2021" AND CE.DATA<="31/12/2021")

INGEGNERE(**COD_I**, NOME_I, COGNOME_I, INDIRIZZO, PARTITA_IVA, CITTÀ_I)

AZIENDA(**COD_A**, NOME_A, TIPOLOGIA_AZIENDA, CITTÀ_A, SITOWEB)

SETTORE(**COD_S**, NOME_S, DESCRIZIONE)

CONSULENZE EFFETTUATE(**REF_I**, **REF_A**, **DATA**, *REF_S*, IMPORTO)

L'ATTRIBUTO TIPOLOGIAAZIENDA = {MICROIMPRESA, PICCOLAIMPRESA, MEDIAIMPRESA, GRANDEIMPRESA}

COMPITO 1

ESPRIMERE LE SEGUENTI INTERROGAZIONI IN SQL O IN ALGEBRA A SECONDA DELLA DOMANDA:

3. **SQL:** PER CIASCUN INGEGNERE DI TORINO CHE NON HA *MAI* EFFETTUATO CONSULENZE NEL SETTORE "EDILIZIA", VISUALIZZARE IL CODICE, IL NOME E IL COGNOME DELL'INGEGNERE E IL NUMERO COMPLESSIVO DI CONSULENZE EFFETTUATE.

```
SELECT COD_I, NOME_I, COGNOME_I, COUNT(*)
FROM CONSULENZE_EFFETTUATE CE, INGEGNERE I
WHERE CE.REF_I=I.COD_I AND I.CITTÀ_I="TORINO"
AND I.COD_I NOT IN
    (SELECT REF_I
     FROM CONSULENZE_EFFETTUATE CE, SETTORE S
     WHERE CE.REF_S=S.COD_S AND S.NOME_S="EDILIZIA")
GROUP BY COD_I, NOME_I, COGNOME_I
```

INGEGNERE(COD_I, NOME_I, COGNOME_I, INDIRIZZO, PARTITA_IVA, CITTÀ_I)

AZIENDA(COD_A, NOME_A, TIPOLOGIA_AZIENDA, CITTÀ_A, SITOWEB)

SETTORE(COD_S, NOME_S, DESCRIZIONE)

CONSULENZE EFFETTUATE(REF_I, REF_A, DATA, REF_S, IMPORTO)

L'ATTRIBUTO TIPOLOGIAAZIENDA = {MICROIMPRESA, PICCOLAIMPRESA, MEDIAIMPRESA, GRANDEIMPRESA}

COMPITO 1

ESPRIMERE LE SEGUENTI INTERROGAZIONI IN SQL O IN ALGEBRA A SECONDA DELLA DOMANDA:

4. **SQL:** PER CIASCUNA AZIENDA, VISUALIZZARE IL CODICE E IL NOME DELL'AZIENDA E IL CODICE, IL NOME E IL COGNOME DELL'INGEGNERE CHE HA EFFETTUATO IL *MAGGIOR NUMERO* DI CONSULENZE PRESSO QUELL'AZIENDA NELL'ANNO 2021.

```
SELECT COD_A, NOME_A, COD_I, NOME_I, COGNOME_I
FROM CONSULENZE_EFFETTUATE CE, INGEGNERE I, AZIENDA A
WHERE CE.REF_I=I.COD_I AND CE.REF_A=A.COD_A
AND CE.DATA>="1/1/2021" AND CE.DATA<="31/12/2021"
GROUP BY COD_A, NOME_A, COD_I, NOME_I, COGNOME_I
HAVING COUNT(*) >=ALL      (SELECT COUNT(*)
                             FROM CONSULENZE_EFFETTUATE CE2
                             WHERE CE2.DATA>="1/1/2021" AND CE2.DATA<="31/12/2021" AND CE2.REF_A=CE.REF_A
                             GROUP BY REF_I)
```


CC CORSA CAMPESTRE. (COD_CORSA, NOME_CORSA, CATEGORIA, CITTÀ_CORSA, REGIONE_CORSA, DATA, REF_ORG)
OR ORGANIZZATORE. (COD_ORG, NOME_O, COGNOME_O)
CO CONCORRENTE. (COD_C, NOME_C, COGNOME_C, DATA_NASCITA, INDIRIZZO, CITTÀ_C, REGIONE_C)
PA PARTECIPAZIONE CORSA (REF_C, REF_CORSA, POSIZIONE)

Esprimere le seguenti interrogazioni in SQL o ALGEBRA a seconda della domanda:

1. ALGEBRA: VISUALIZZARE IL CODICE, IL NOME E IL COGNOME DELLE PERSONE CHE HANNO ORGANIZZATO *ALMENO DUE CORSE CAMPESTRI* PER LA STESSA CATEGORIA IN CITTÀ DIVERSE.

PROJ COD_ORG, NOME_O, COGNOME_O (OR JOIN OR.COD_ORG=CC1.REF_ORG (SEL CC1.CATEGORIA=CC2.CATEGORIA AND
 CC1.CITTA_CORSA<>CC2.CITTA_CORSA (CC1 JOIN CC1.REF_ORG=CC2.REF_ORG CC2)

CC CORSA CAMPESTRE. (COD_CORSA, NOME_CORSA, CATEGORIA, CITTÀ_CORSA, REGIONE_CORSA, DATA, REF_ORG)
OR ORGANIZZATORE. (COD_ORG, NOME_O, COGNOME_O)
CO CONCORRENTE. (COD_C, NOME_C, COGNOME_C, DATA_NASCITA, INDIRIZZO, CITTÀ_C, REGIONE_C)
PA PARTECIPAZIONE CORSA (REF_C, REF_CORSA, POSIZIONE)

Esprimere le seguenti interrogazioni in SQL o ALGEBRA a seconda della domanda:

2. ALGEBRA: VISUALIZZARE IL CODICE, IL NOME E IL COGNOME DI CIASCUN CONCORRENTE CHE HA HA PARTECIPATO NEL 2021 A TUTTE LE CORSE DI CATEGORIA *UNDER40* DI PALERMO.

PROJ_{COD_C, NOME_C, COGNOME_C} (CO JOIN_{COD_C=REF_C}
 (PROJ_{REF_C, REF_CORSA} (PA)

÷

REN_{REF_CORSA<-COD_CORSA} (PROJ_{COD_CORSA} (SEL_{CATEGORIA='UNDER40' AND CITTÀ_CORSA='PALERMO' AND DATA>='1/1/2021' AND DATA<='31/12/2021', CC))))}

CC CORSA CAMPESTRE (COD_CORSA, NOME_CORSA, CATEGORIA, CITTÀ_CORSA, REGIONE_CORSA, DATA, REF_ORG)
OR ORGANIZZATORE (COD_ORG, NOME_O, COGNOME_O)
CO CONCORRENTE (COD_C, NOME_C, COGNOME_C, DATA_NASCITA, INDIRIZZO, CITTÀ_C, REGIONE_C)
PA PARTECIPAZIONECORSA (REF_C, REF_CORSA, POSIZIONE)

Esprimere le seguenti interrogazioni in SQL o ALGEBRA a seconda della domanda:

2. SQL: VISUALIZZARE IL CODICE, IL NOME E IL COGNOME DI CIASCUN CONCORRENTE CHE HA HA PARTECIPATO NEL 2021 A TUTTE LE CORSE DI CATEGORIA *UNDER40* DI PALERMO.

```
SELECT COD_C, NOME_C, COGNOME_C
FROM CONCORRENTE CO
WHERE NOT EXISTS (SELECT *
                  FROM CORSA_CAMPESTRE CC
                  WHERE CC.CATEGORIA='UNDER40' AND CC.CITTÀ_CORSA='PALERMO'
                  AND CC.DATA>='1/1/2021' AND CC.DATA<='31/12/2021'
                  AND NOT EXISTS (SELECT *
                                FROM PARTECIPAZIONECORSA PA
                                WHERE PA.REF_C=CO.COD_C AND PA.REF_CORSA=CC.COD_CORSA))
```

CC CORSA CAMPESTRE (COD_CORSA, NOME_CORSA, CATEGORIA, CITTÀ_CORSA, REGIONE_CORSA, DATA, REF_ORG)
OR ORGANIZZATORE (COD_ORG, NOME_O, COGNOME_O)
CO CONCORRENTE (COD_C, NOME_C, COGNOME_C, DATA_NASCITA, INDIRIZZO, CITTÀ_C, REGIONE_C)
PA PARTECIPAZIONECORSA (REF_C, REF_CORSA, POSIZIONE)

Esprimere le seguenti interrogazioni in SQL o ALGEBRA a seconda della domanda:

3. SQL: VISUALIZZARE IL CODICE, IL NOME E IL COGNOME DI CIASCUNA PERSONA CHE, NELLA REGIONE PIEMONTE, HA ORGANIZZATO ALMENO UNA CORSA CAMPESTRE PER LA CATEGORIA *UNDER 14* MA NESSUNA CORSA CAMPESTRE PER LA CATEGORIA *UNDER 20*.

```
SELECT COD_ORG, NOME_O, COGNOME_O
FROM ORGANIZZATORE OR
WHERE OR.COD_ORG IN (SELECT REF_ORG
                     FROM CORSA_CAMPESTRE CC1
                     WHERE CC.CATEGORIA='UNDER 14' AND CC.REGIONE_CORSA='PIEMONTE')
AND OR.COD_ORG NOT IN (SELECT REF_ORG
                     FROM CORSA_CAMPESTRE CC2
                     WHERE CC.CATEGORIA='UNDER 20' AND CC.REGIONE_CORSA='PIEMONTE')
```

CC CORSA CAMPESTRE (COD_CORSA, NOME_CORSA, CATEGORIA, CITTÀ_CORSA, REGIONE_CORSA, DATA, REF_ORG)
OR ORGANIZZATORE (COD_ORG, NOME_O, COGNOME_O)
CO CONCORRENTE (COD_C, NOME_C, COGNOME_C, DATA_NASCITA, INDIRIZZO, CITTÀ_C, REGIONE_C)
PA PARTECIPAZIONECORSA (REF_C, REF_CORSA, POSIZIONE)

Esprimere le seguenti interrogazioni in SQL o ALGEBRA a seconda della domanda:

4. SQL: VISUALIZZARE IL CODICE, IL NOME E IL COGNOME DI CIASCUN CONCORRENTE CHE HA PARTECIPATO A TUTTE LE CORSE CAMPESTRI ORGANIZZATE NELLA PROPRIA REGIONE, MA NON HA *MAI* CONSEGUITO LA PRIMA POSIZIONE IN NESSUNA CORSA CAMPESTRE.

```
SELECT COD_C, NOME_C, COGNOME_C
FROM CONCORRENTE CO, PARTECIPAZIONECORSA PA, CORSA_CAMPESTRE CC
WHERE CO.COD_C=PA.REF_C AND PA.REF_CORSA=CC.COD_CORSA AND CC.REGIONE_CORSA=CO.REGIONE_C
AND CO.COD_C NOT IN
    (SELECT PA2.REF_C
     FROM PARTECIPAZIONECORS PA2
     WHERE PA2.POSIZIONE=1)
GROUP BY COD_C, NOME_C, COGNOME_C
HAVING COUNT(*) =
    (SELECT COUNT(*)
     FROM CORSA_CAMPESTRE CC2
     WHERE CC2.REGIONE_CORSA=CO.REGIONE_C) PASSAGGIO DI PARAMETRI
```