

BASI DI DATI 1 - PROFF. S.CERI, S.COMAI, G.POZZI, L.TANCA - A.A. 19/20
APPELLO DEL 22 FEBBRAIO 2020

Parte 1: LINGUAGGI DI INTERROGAZIONE - DURATA: 1 h 15 m

Lo schema seguente è utilizzato per gestire l'acquisto di biglietti per i voli di una compagnia aerea. Capienza indica il numero massimo di posti disponibili su un determinato volo.

VOLO (NUMERO, DATA, ORA, AEROPORTOPARTENZA, AEROPORTODESTINAZIONE, CAPIENZA, TIPOAEROMOBILE)

BIGLIETTO (NUMEROPOSTO, NUMEROVOLO, DATA, CFCLIENTE, PREZZO)

CLIENTE (CF, NOME, COGNOME, INDIRIZZO, CITTÀ, NAZIONE)

A. SQL (13 punti)

1. Specificare i comandi di creazione della tabella BIGLIETTO, definendo i vincoli di tupla e di dominio ritenuti opportuni ed esprimendo gli eventuali vincoli di integrità referenziale verso le altre tabelle (1 p.).
2. Per ciascun anno, trovare il singolo volo (numero e data) che ha generato il massimo ricavo per la compagnia (rispetto a tutti gli altri voli dello stesso anno) (3 p.).
3. Trovare il codice fiscale dei clienti che hanno volato con tutti i tipi di aeromobile (3 p.).
4. Trovare il codice fiscale, il cognome e nome dei clienti che nel 2019 hanno acquistato almeno due biglietti, ma mai per lo stesso aeroporto di destinazione (4 p.).
5. Esprimere il vincolo che verifichi che il numero di biglietti venduti per un volo sia al più superiore al 5% della capienza del volo stesso (2 p.).

B. Linguaggi formali (6 punti)

6. Formulare in Algebra Relazionale, Calcolo Relazionale e Datalog l'interrogazione A.3. (2+2+2 p.).

Parte 2: PROGETTAZIONE E TEORIA - DURATA: 1 h 15 m

C. Progetto Concettuale e Progetto Logico (13 punti)

Una Società di assicurazione per RCA (responsabilità civile auto) vuole tenere traccia dei propri clienti e dei loro sinistri. Per ogni cliente assicurato, sono noti i dati identificativi ed il codice fiscale: per ogni veicolo assicurato (autoveicolo, motoveicolo, autocarro, rimorchio) sono noti i dati del proprietario, il numero di telaio, il numero di targa e la potenza del veicolo (ove disponibile). Un cliente può ovviamente assicurare più veicoli, anche di proprietà NON sua.

Ogni polizza assicurativa riporta i dati del cliente, i dati del veicolo, la data di inizio e di fine della polizza, il premio annuo pagato dal cliente (costo della polizza), le eventuali estensioni (furto, incendio, atti vandalici), ed i vari massimali coperti (massimo danno pagabile dalla Società: per sinistro, per persona, per evento).

Per ogni sinistro, che può coinvolgere due o più veicoli, sono noti il numero targa e numero di telaio dei veicoli coinvolti, le polizze di copertura, i dati dei conducenti al momento dell'evento, nonché la data ed il luogo. Per i conducenti vengono memorizzati anche il tipo di patente (A, B, ...) e la data di scadenza. Per alcuni sinistri, tutti i veicoli sono coperti dalla medesima società di assicurazione: altri sinistri, invece, coinvolgono veicoli coperti da differenti Società, delle quali si memorizzano denominazione, agenzia, indirizzo.

A fronte di un sinistro, per ciascun cliente sottoscrittore della polizza di una delle vetture coinvolte nel sinistro, un perito della Società concorda un appuntamento, valuta i danni subiti dal veicolo stendendo una relazione, considera l'eventuale responsabilità del guidatore, e stima l'ammontare dei danni. Tale importo può essere bonificato direttamente all'autofficina di riparazione oppure al titolare della polizza.

- *Il progetto è valutato per completezza, correttezza, leggibilità, minimalità e autoesplicatività. Si ricordi che anche il progetto logico è un **grafo** i cui nodi e archi devono essere disposti coerentemente con quelli del progetto concettuale.*
- *Ricordare anche di specificare gli **identificatori** di tutte le entità e le **cardinalità** di tutte le associazioni, e di disporre i due grafi su due facciate affiancate, in modo da poterli osservare simultaneamente.*

D. Teoria (1 punto)

Cosa vuol dire per uno schema relazionale essere in Terza Forma Normale? Quali sono gli svantaggi di uno schema che non soddisfa questa condizione?

Ex. 1

```
CREATE TABLE Biglietto (  
    NumeroPosto VARCHAR(4),  
    NumeroVolo VARCHAR(5),  
    Data DATE,  
    CFCliente VARCHAR(16) NOT NULL REFERENCES Cliente(CF)  
        ON UPDATE CASCADE  
        ON DELETE NO ACTION,  
    Prezzo FLOAT NOT NULL,  
  
    PRIMARY KEY (NumeroPosto, NumeroVolo, Data),  
    FOREIGN KEY (NumeroVolo, Data) REFERENCES Volo(Numero, Data)  
        ON UPDATE NO ACTION  
        ON DELETE NO ACTION  
)
```

Ex. 2

```
CREATE VIEW Ricavi(Volo, Data, Somma) AS(  
    SELECT NumeroVolo, Data, SUM(Prezzo)  
    FROM Biglietto  
    GROUP BY NumeroVolo, Data  
)
```

```
SELECT R1.Volo, YEAR(R1.Data)  
FROM Ricavi R1  
WHERE R1.Somma = (  
    SELECT MAX(Somma)  
    FROM Ricavi R2  
    WHERE YEAR(R1.Data) = YEAR(R2.Data)  
)
```

Ex. 3

```
SELECT B1.CFCliente  
FROM Biglietto B1, Volo V1  
WHERE B1.NumeroVolo = V1.Numero AND B1.Data = V1.Data  
GROUP BY B1.CFCliente  
HAVING COUNT(DISTINCT V1.TipoAeromobile) = (  
    SELECT COUNT(DISTINCT V2.TipoAeromobile)  
    FROM Volo V2  
)
```

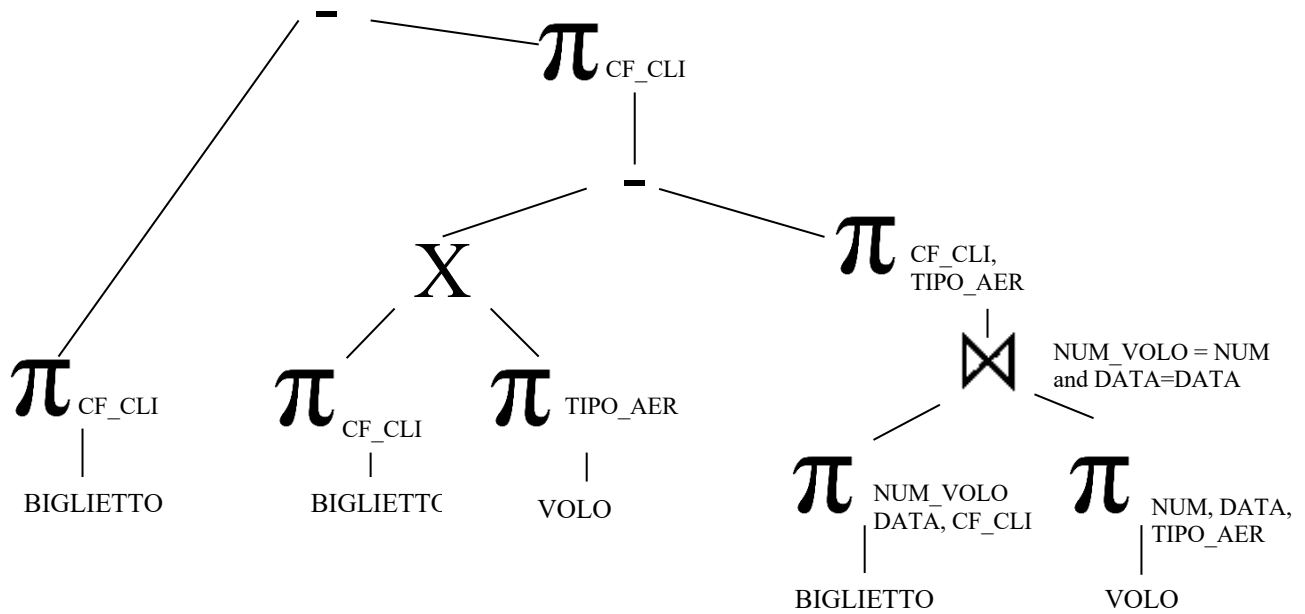
Ex. 4

```
SELECT B1.CF
FROM Biglietto B1
WHERE YEAR(B1.Data) = 2019
GROUP BY B1.CF
HAVING COUNT(*) >= 2
      AND COUNT(*) = (
        SELECT COUNT(DISTINCT V2.Destinazione)
        FROM Volo V2, Biglietto B2
        WHERE B2.NumeroVolo = V2.Numero
              AND B2.Data = V2.Data
              AND YEAR(B2.Data) = 2019
              AND B2.CF = B1.CF
      )
)
```

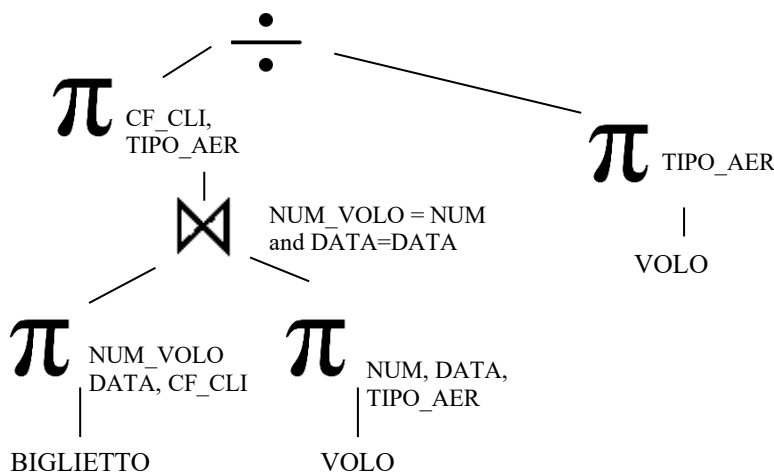
Ex. 5

```
CREATE ASSERTION Overbooking CHECK(
  NOT EXISTS(
    SELECT *
    FROM Volo V
    WHERE V.Capienza*1.05 < (
      SELECT COUNT(*)
      FROM Biglietti B
      WHERE B.NumeroVolo = V.Codice
            AND B.Data = V.Data
    )
  )
)
```

Algebra relazionale



Oppure:



Calcolo relazionale

$\{t \mid \exists t1 \in BIGLIETTO (t[CF_Cli]=t1[CF_Cli] \wedge \neg (\exists t2 \in BIGLIETTO, \exists t3 \in VOLO (t2[CF_Cli]=t1[CF_Cli] \wedge \neg (\exists t4 \in BIGLIETTO, \exists t5 \in VOLO (t4[CF_Cli]=t2[CF_Cli] \wedge t4[Num_Volo]=t5[Num] \wedge t4[Data]=t5[Data] \wedge t5[Tipo_Aer]=t3[Tipo_Aer])))))\}$

Oppure :

$\{t \mid \exists t1 \in BIGLIETTO (t[CF_Cli]=t1[CF_Cli] \wedge \neg (\exists t2 \in VOLO (\neg (\exists t3 \in BIGLIETTO, \exists t4 \in VOLO (t3[CF_Cli]=t1[CF_Cli] \wedge t3[Num_Volo]=t4[Num] \wedge t3[Data]=t4[Data] \wedge t4[Tipo_Aer]=t2[Tipo_Aer])))))\}$

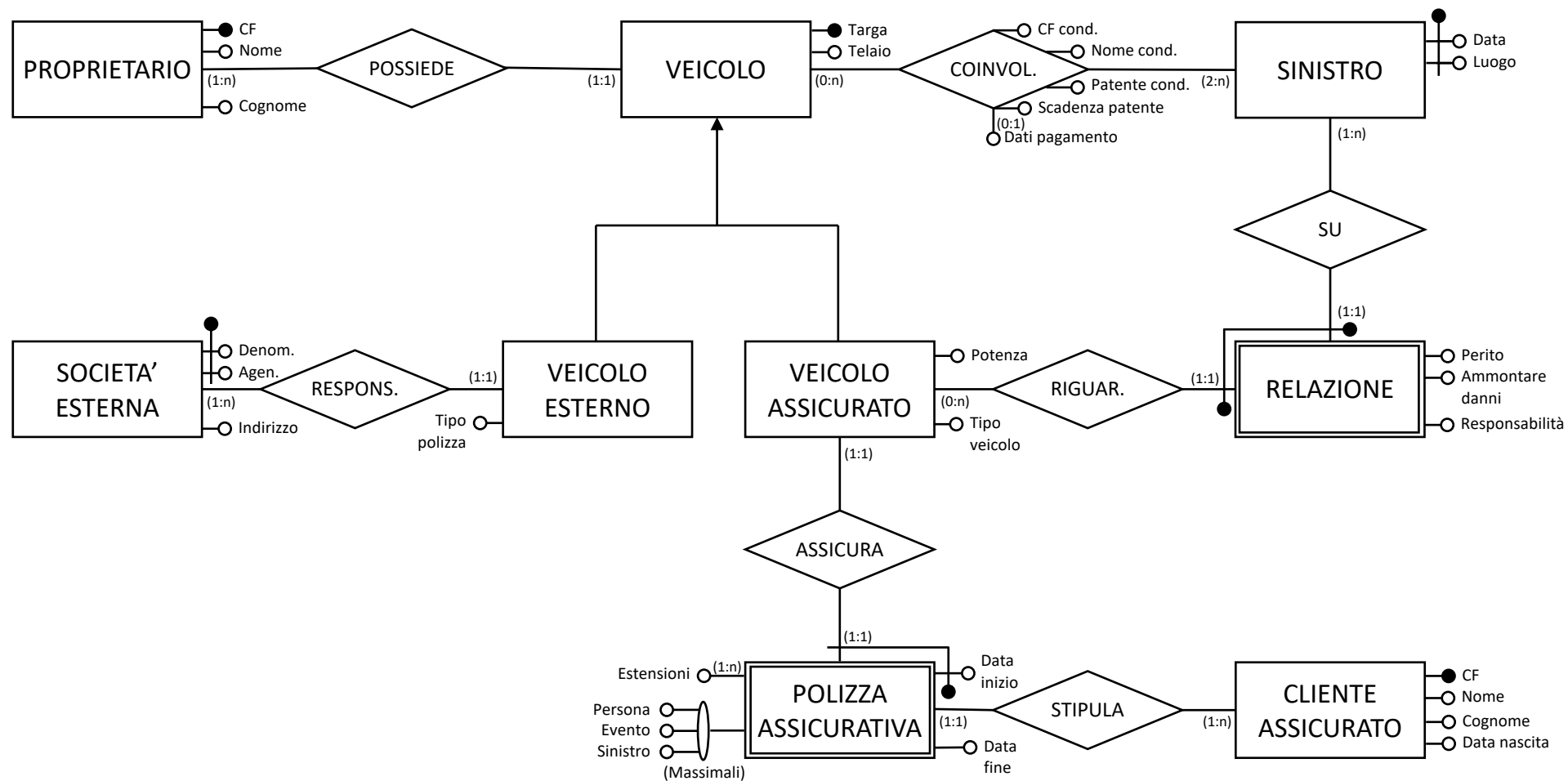
Datalog

$CLI_TIPO_USATO(cf, tipo) :- BIGLIETTO(_, num, data, cf, _), VOLO(num, data, _, _, _, tipo)$

$CLI_MANCA_TIPO(cf) :- BIGLIETTO(_, _, _, CF, _), VOLO(_, _, _, _, _, tipo),$
 $\neg CLI_TIPO_USATO(cf, tipo)$

$CLI_SCELTO(cf) :- BIGLIETTO(_, _, _, CF, _), \neg CLI_MANCA_TIPO(cf)$

$? - CLI_SCELTO(x)$



CLIENTE_ASSICURATO (CF, Nome, Cognome, Nascita)

POLIZZA (Targa, Data_inizio, Data_fine, Mass_Persona, Mass_Evento, Mass_Sinistro)

ESTENSIONE (Targa, Data_inizio, Estensione)

VEICOLO_ASSICURATO (Targa, Telaio, Potenza, Tipo_veicolo, CF_proprietario)

PROPRIETARIO (CF, Nome, Cognome)

VEICOLO_ASSICURATO_ESTERNAMENTE (Targa, Telaio, Tipo_polizza, Società, Agenzia)

SOCIETA'_ESTERNA (Denominazione, Agenzia, Indirizzo)

COINVOLTO (Targa, Data, Luogo, CF_conducente, Nome_conducente, Patente_conducente, Scadenza_patente, Dati_pagamento*)

SINISTRO (Data, Luogo)

RELAZIONE (Targa, Data, Luogo, Perito, Ammontare_danni, Responsabilità)

ERRORI GRAVISSIMI IN UNO SCHEMA CONCETTUALE:

Usare una entità debole al posto di una sottoclasse (es., CONDUCENTE entità debole di PERSONA)

Usare una sottoclasse al posto di una entità con relazione (es., PERITO sottoclasse di SINISTRO)

Considerare la società proprietaria del DB come una entità, oppure (nel caso del nostro esercizio) come elemento dell'entità SOCIETA'

Quando è stata inserita una relazione R tra due entità A e B, aggiungere gli identificatori - o altri attributi - dell'entità A sull'entità B.

Quando ci dovrebbe essere una relazione R tra due entità A e B, aggiungere invece identificatori - o altri attributi - dell'entità A sull'entità B.

In questo esame, alcune persone hanno considerato una gerarchia sotto MASSIMALE, le cui sottoclassi erano SINISTRO, PERSONA e SOCIETA', e poi hanno usato la stessa classe SINISTRO per rappresentare l'incidente specifico.