

**Basi di Dati 1**  
Prof. L. Tanca e F. A. Schreiber  
**APPELLO DEL 1 OTTOBRE 2015**  
**Tempo: 2h30m**

Si consideri il seguente schema di base di dati, che vuole memorizzare informazioni relative ai viaggi di lavoro effettuati dai dipendenti di un'azienda.

VIAGGIODILAVORO (CFIncaricato, DataInizio, DurataGiorni, SpesaVitto, SpesaAlloggio,  
SpesaTrasporto, CittàDestinazione, CodiceCapitoloDiSpesa)  
DIPENDENTE (CodiceFiscale, Cognome, Nome, CittàResidenza, Ruolo)  
CITTÀ (NomeCittà, Nazione)  
CAPITOLODISPESA (Codice, Nome, Responsabile, Disponibilità)

Le spese relative a un viaggio di lavoro sono associate a un certo capitolo di spesa, e il viaggio di lavoro deve essere autorizzato dal dipendente che è responsabile di quel capitolo di spesa. L'attributo "Disponibilità" della relazione CAPITOLODISPESA rappresenta la spesa complessiva massima che può essere associata a quel capitolo. L'attributo "Ruolo" della relazione DIPENDENTE può assumere i valori "Impiegato", "Dirigente" oppure "Tecnico".

**A. LINGUAGGI FORMALI (6 punti).**

Esprimere in algebra relazionale (ottimizzata) e calcolo relazionale la seguente interrogazione:

*Trovare il codice fiscale di tutti i dipendenti che non hanno mai effettuato viaggi di lavoro in Francia, e di tutti quelli che non sono mai stati in viaggio due volte nella stessa nazione.*  
(4 punti)



### Calcolo relazionale

$$\{t \mid (\exists t1 \in \text{Dipendente} (t[\text{CodiceFiscale}] = t1[\text{CodiceFiscale}] \wedge \neg (\exists t2 \in \text{ViaggioDiLavoro}, \exists t3 \in \text{Città} (t2[\text{CFIncaricato}] = t1[\text{CodiceFiscale}] \wedge t2[\text{CittàDestinazione}] = t3[\text{NomeCittà}] \wedge t3[\text{Nazione}] = \text{'Francia'})))) \vee (\exists t4 \in \text{Dipendente} (t[\text{CodiceFiscale}] = t4[\text{CodiceFiscale}] \wedge \neg (\exists t5 \in \text{ViaggioDiLavoro}, \exists t6 \in \text{Città}, \exists t7 \in \text{ViaggioDiLavoro}, \exists t8 \in \text{Città} (t5[\text{CFIncaricato}] = t4[\text{CodiceFiscale}] \wedge t5[\text{CittàDestinazione}] = t6[\text{NomeCittà}] \wedge t7[\text{CittàDestinazione}] = t8[\text{NomeCittà}] \wedge t5[\text{CFIncaricato}] = t7[\text{CFIncaricato}] \wedge t6[\text{Nazione}] = t8[\text{Nazione}] \wedge t5[\text{DataInizio}] \neq t7[\text{DataInizio}])))\})$$

Esprimere in datalog la seguente interrogazione:

*Trovare le città della Germania verso cui non sono mai stati eseguiti viaggi con nazione di residenza dell'incaricato uguale a quella del dipendente che ha autorizzato il viaggio. (2 punti)*

CittàStessaNazRes(CittàDest) :- ViaggioDiLavoro(CFIncaricato, \_, \_, \_, \_, CittàDest, \_, CodiceCDS), Dipendente(CFIncaricato, \_, \_, CittàDip, \_), Città(CittàDip, Nazione), CapitoloDiSpesa(CodiceCDS, \_, Responsabile, \_), Dipendente(Responsabile, \_, \_, CittàResp, \_), Città(CittàResp, Nazione)

CittàScelta(NomeCittà) :- Città(NomeCittà, "Germania"), ¬CittàStessaNazione(NomeCittà)

?- CittàScelta(x)

### B. SQL: INTERROGAZIONI (7 punti).

Scrivere in SQL le seguenti interrogazioni:

- Per ogni responsabile, trovare la durata media dei viaggi di lavoro da lui autorizzati svolti da tecnici in città che sono state destinazione di almeno 10 viaggi. Visualizzare il risultato in ordine decrescente rispetto alla media calcolata. (3.5 punti)*

```
SELECT C.Responsabile, AVG(DurataGiorni) AS Media
FROM ViaggioDiLavoro AS V, Dipendente AS D, CapitoloDiSpesa AS C
WHERE V.CFIncaricato=D.CodiceFiscale AND V.CapitoloDiSpesa=C.Codice AND
      D.Ruolo='Tecnico' AND V.CittàDestinazione IN (
      SELECT CittàDestinazione
      FROM ViaggioDiLavoro
      GROUP BY CittàDestinazione
      HAVING COUNT(*)>=10
      )
GROUP BY C.Responsabile
ORDER BY Media DESC
```

2. *Trovare codice fiscale, cognome e nome dei dipendenti che hanno speso la cifra massima nell'eseguire viaggi di lavoro autorizzati da loro stessi aventi come destinazione la propria città di residenza. (3.5 punti)*

```
CREATE VIEW SpesaDip (CFDip, Cognome, Nome, Spesa) AS (  
  SELECT D.CodiceFiscale, D.Cognome, D.Nome, SUM(V.SpesaVitto +  
    V.SpesaAlloggio + V.SpesaViaggio)  
  FROM ViaggioDiLavoro AS V, CapitoloDiSpesa AS C, Dipendente AS D  
  WHERE V.CFIncaricato=D.CodiceFiscale AND V.CapitoloDiSpesa=C.Codice AND  
    V.CFIncaricato=C.Responsabile AND V.CittàDestinazione=D.CittàResidenza  
  GROUP BY D.CodiceFiscale, D.Cognome, D.Nome  
)
```

```
SELECT CFDip, Cognome, Nome  
FROM SpesaDip  
WHERE Spesa = (  
    SELECT MAX(Spesa)  
    FROM SpesaDip  
)
```

### C. DDL: Definizione di Schemi e Vincoli (3 punti)

1. Specificare in SQL la creazione delle tabelle DIPENDENTE e VIAGGIODILAVORO, definendo i vincoli di tupla e di dominio ritenuti opportuni ed esprimendo eventuali vincoli di integrità referenziale relativi a tutte le tabelle dello schema. (1 punto)

```
CREATE TABLE Dipendente (  
  CodiceFiscale CHAR(16) PRIMARY KEY,  
  Cognome VARCHAR(255),  
  Nome VARCHAR(255),  
  CittàResidenza VARCHAR(255) REFERENCES Città(NomeCittà) ON UPDATE  
    CASCADE ON DELETE NO ACTION,  
  Ruolo ENUM('Impiegato', 'Dirigente', 'Tecnico')  
)
```

```
CREATE TABLE ViaggioDiLavoro (  
  CFIncaricato CHAR(16) REFERENCES Dipendente(CodiceFiscale) ON UPDATE  
    CASCADE ON DELETE CASCADE,  
  DataInizio DATE,  
  DurataGiorni INTEGER,  
  SpesaVitto NUMERIC(7,2),  
  SpesaAlloggio NUMERIC(7,2),  
  SpesaTrasporto NUMERIC(7,2),  
  CittàDestinazione VARCHAR(255) REFERENCES Città(NomeCittà) ON UPDATE  
    CASCADE ON DELETE NO ACTION,  
  CodiceCapitoloSpesa VARCHAR(255) REFERENCES CapitoloDiSpesa(Codice)  
    ON UPDATE CASCADE ON DELETE NO ACTION,
```

**PRIMARY KEY** (CFIncaricato, DataInizio)  
)

2. Specificare in SQL il vincolo che verifichi che per ogni capitolo di spesa la spesa totale sostenuta non ecceda la disponibilità del capitolo di spesa stesso. (2 punti)

```
CREATE ASSERTION VerificaCapitoli CHECK (  
  NOT EXISTS (  
    SELECT C.Codice  
    FROM ViaggioDiLavoro AS V, CapitoloDiSpesa AS C  
    WHERE V.CapitoloDiSpesa=C.Codice  
    GROUP BY C.Codice, C.Disponibilità  
    HAVING SUM(V.SpesaAlloggio + V.SpesaVitto + V.SpesaTrasporto) >  
      C.Disponibilità  
  )  
)
```

#### **D. TEORIA (4 punti)**

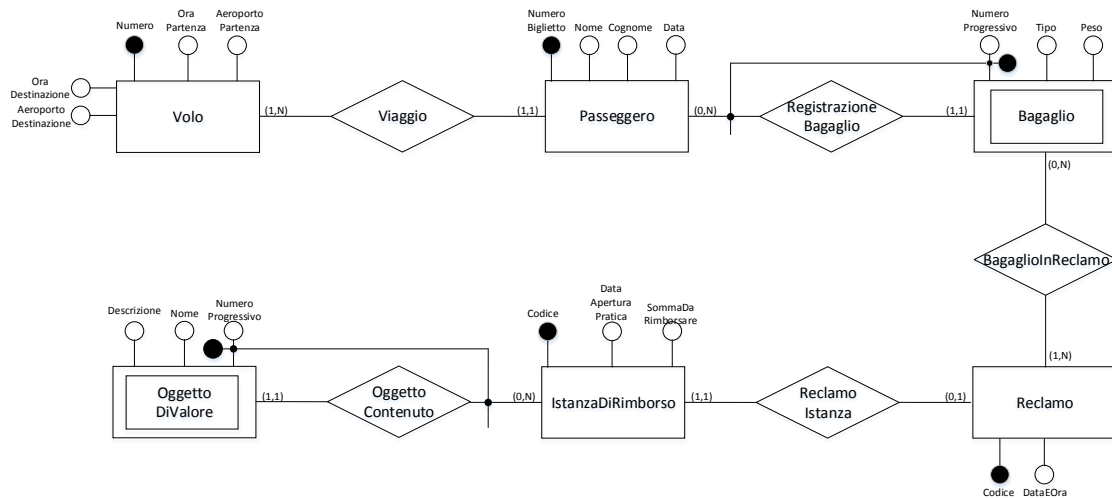
Descrivere il concetto di vincolo e la funzione dei vincoli in una base di dati, utilizzando parole proprie (e quindi evitando di ripetere testualmente quello che è scritto nelle slide del corso).

## E. PROGETTAZIONE DI UNA BASE DI DATI (12 PUNTI)

Progettare la base di dati relativa alla situazione descritta nel seguito. Svolgere il progetto concettuale usando il modello ER, e il progetto logico con il modello relazionale. Specificare le chiavi e gli eventuali vincoli d'integrità.

Si vuole progettare una base di dati per la gestione dei bagagli in un aeroporto. Ogni volo è caratterizzato dal numero di volo, dall'ora e aeroporto di partenza e dall'ora e aeroporto di destinazione. I passeggeri (identificati dal numero di biglietto e caratterizzati dal nome e cognome) utilizzano i voli in certe date. I passeggeri sono identificati dal numero di biglietto e dal nome e cognome. I passeggeri al check-in possono registrare i loro bagagli. Ogni bagaglio stivato è contraddistinto da un numero progressivo unico per ogni biglietto ed è caratterizzato dal tipo e dal peso. Nel momento in cui il bagaglio viene caricato su un volo l'uso di un lettore del codice a barre presente sul bagaglio permette di tenere traccia della lista di bagagli presente su un determinato volo. Se il bagaglio non arriva a destinazione, i passeggeri devono presentare reclamo all'aeroporto. Ogni reclamo è caratterizzato da un codice univoco, dalla data e ora del reclamo, dal riferimento al passeggero e dal riferimento ai bagagli smarriti. La maggior parte dei bagagli smarriti sono stati imbarcati su voli diversi da quelli in cui viaggia il passeggero. I bagagli per cui non si trova nessuna associazione con un volo vengono definiti "introvabili" e viene creata una istanza di rimborso per il passeggero. Ogni istanza di rimborso ha un codice univoco, il riferimento al reclamo, la data di apertura della pratica, la somma da rimborsare e la lista di oggetti di valore contenuta nella valigia. Ogni oggetto viene caratterizzato da un numero progressivo unico per istanza di rimborso, il nome dell'oggetto e una breve descrizione.

### Schema concettuale



### Schema logico

Volo (Numero, OraPartenza, AeroportoPartenza, OraDestinazione, AeroportoDestinazione)

Passeggero (NumeroBiglietto, Nome, Cognome, Data, NumeroVolo)

Bagaglio (NumeroBiglietto, NumeroProgressivo, Tipo, Peso)

Reclamo (Codice, DataEOra)

BagaglioInReclamo (CodiceReclamo, NumeroBiglietto, ProgressivoBagaglio)

IstanzaDiRimborso (Codice, DataAperturaPratica, SommaDaRimborsare)

OggettoDiValore (CodiceIstanza, NumeroProgressivo, Nome, Descrizione)

### Chiavi esterne

Passeggero.NumeroVolo → Volo.Numero

Bagaglio.NumeroBiglietto → Passeggero.NumeroBiglietti

BagaglioInReclamo.CodiceBagaglio → Reclamo.Codice

BagaglioInReclamo.(NumeroBiglietto, ProgressivoBagaglio) →  
Bagaglio.(NumeroBiglietto, NumeroProgressivo)

OggettoDiValore.CodiceIstanza → IstanzaDiRimborso.Codice