

## SECONDO PARZIALE - Basi di Dati e Lab

11/06/2018

Corso di Laurea in Ingegneria Informatica

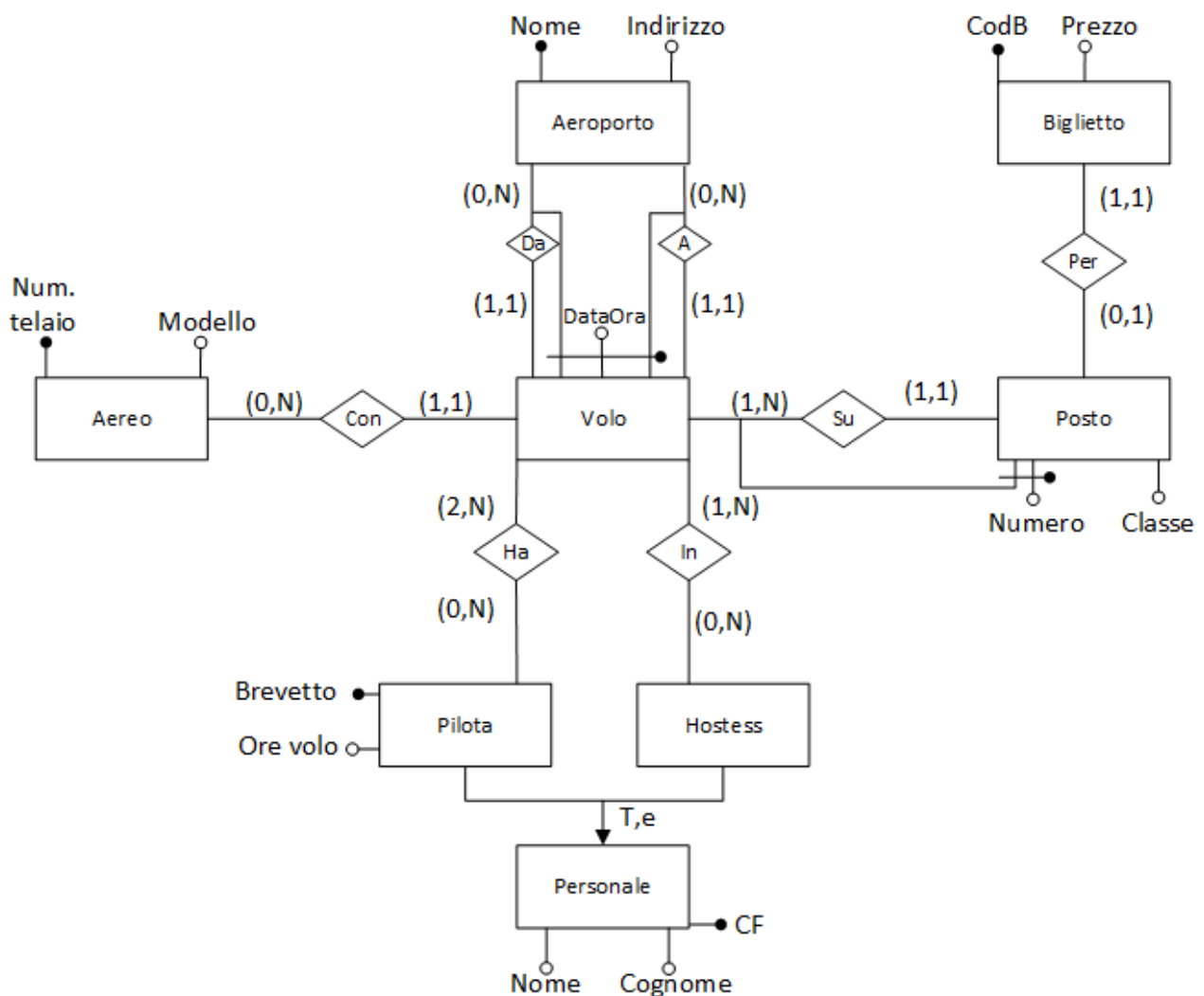
Prof. Sonia Bergamaschi

### Esercizio 1 (punti 6)

- Definizione di foreign key in SQL (3 punti)
- Definizione di seconda forma normale ed esempio di violazione (3 punti)

### Esercizio 2 (punti 3)

Tradurre lo schema E/R in schema logico relazionale, evidenziando gli eventuali vincoli non esprimibili nel modello.



### Esercizio 3 (punti 24)

Sia dato il seguente schema relazionale di un sistema di gestione dei treni.

Treno (id treno, modello, anno, num\_posti)

Stazione (cod stazione, nome, città)

**AK** (nome)

Tratta (id tratta, id\_treno, partenza, arrivo, data\_partenza, ora\_partenza, durata)

**FK:** id\_treno **References** Treno **NOT NULL**

**FK:** partenza **References** Stazione **NOT NULL**

**FK:** arrivo **References** Stazione **NOT NULL**

A) Si richiede di scrivere in algebra relazionale e in SQL le seguenti interrogazioni:

1. Selezionare i nomi delle stazioni da cui nel 2010 sono partiti tutti i modelli di treni presenti. **(punti 2+2)**.
2. Selezionare i dati dei treni che sono passati per la città di Modena (partenza o arrivo) nel mese di gennaio del 2012. **(punti 2+2)**
3. Selezionare i dati delle stazioni da cui non ci sono state partenze nel mese di dicembre 2016. **(punti 2+2)**

B) Si richiede di scrivere in SQL le seguenti interrogazioni:

4. Selezionare i dati dei treni che nel 2011 sono arrivati almeno 10 volte nella stessa città, mostrando anche la città. **(punti 3)**
5. Selezionare i dati dei treni che nel 2016 hanno effettuato più ore di lavoro (somma delle durate di percorrenza). **(punti 3)**
6. Creare una vista che mostri per le stesse tratte (stessa partenza e stesso arrivo): nomi delle stazioni di arrivo/partenza, modello del treno che ha la durata di percorrenza minore, e la relativa durata di percorrenza. **(punti 3)**
7. Selezionare le città in cui il numero di partenze è superiore al numero di arrivi. **(punti 3)**

## SOLUZIONE

### Esercizio 2 (punti 6)

Aereo (Num telaio, modello)

Aeroporto (Nome, indirizzo)

Pilota (CF, Brevetto, Ore volo, Nome, Cognome)

**AK**(Brevetto)

Hostess (CF, Nome, Cognome)

Volo (Partenza, Destinazione, DataOra, Num telaio)

**FK** Num telaio **REFERENCES** Aereo

**FK** Partenza **REFERENCES** Aeroporto

**FK** Destinazione **REFERENCES** Aeroporto

Ha (Partenza, Destinazione, DataOra, CF pilota)

**FK** CF\_pilota **REFERENCES** Pilota

**FK** Partenza, Destinazione, DataOra **REFERENCES** Volo

**Nota:** almeno 2 piloti per volo

In (Partenza, Destinazione, DataOra, CF)

**FK** CF **REFERENCES** Hostess

**FK** Partenza, Destinazione, DataOra **REFERENCES** Volo

Posto (Partenza, Destinazione, DataOra, Numero, Classe)

**FK** Partenza, Destinazione, DataOra **REFERENCES** Volo

Biglietto (CodB, Prezzo, Partenza, Destinazione, DataOra, Numero)

**AK** Partenza, Destinazione, DataOra, Numero

**FK** Partenza, Destinazione, DataOra, Numero **REFERENCES** Posto

### Esercizio 3

1. Selezionare i nomi delle stazioni da cui nel 2010 sono partiti tutti i modelli di treni presenti. (punti 2+2)

Si prendono le tratte del 2010

$$S1 = \sigma_{data\_partenza \geq 2010-1-1 \text{ AND } data\_partenza \leq 2010-12-31}(Tratta)$$

Si estraggono le stazioni di partenza e modelli dei treni

$$S2 = \pi_{partenza,modello}(Treno \bowtie S1)$$

Si divide S2 per i modelli di treni esistenti ottenendo così le stazioni da cui sono partiti tutti i modelli e si mette in join con le stazioni visto che si vogliono i nomi

$$\pi_{nome}(Stazione \bowtie (S2 \div \pi_{modello}(Treno)))$$

```
SELECT DISTINCT nome
FROM stazione s
WHERE NOT EXISTS (
    SELECT *
    FROM treno t
    WHERE NOT EXISTS (
        SELECT *
        FROM tratta tr
        JOIN treno t1 ON tr.id_treno = t1.id_treno
        WHERE t1.modello = t.modello
        AND tr.partenza = s.cod_stazione
        AND YEAR(data_partenza) = 2010
    )
)
```

2. Selezionare i dati dei treni che sono passati per la città di Modena (partenza o arrivo) nel mese di gennaio del 2012. (punti 2+2)

Si prendono le stazioni di Modena

$$S1 = \sigma_{città='Modena'}(Stazione)$$

Si prendono le tratte di gennaio 2012

$$S2 = \sigma_{data\_partenza \geq '2012-01-01' \text{ AND } data\_partenza \leq '2012-12-31'}(Tratta)$$

Infine si estraggono i treni che sono partiti/arrivati a Modena

$$Treno \bowtie \pi_{id\_treno}(S1 \bowtie_{S2.partenza=S1.codstazione} (S2)) \cup (S1 \bowtie_{S2.arrivo=S1.codstazione} (S2))$$

```
SELECT DISTINCT t.*
FROM treno t
JOIN tratta tr ON t.id_treno = tr.id_treno
JOIN stazione sp ON tr.partenza = sp.cod_stazione
JOIN stazione sa ON tr.arrivo = sa.cod_stazione
WHERE tr.data_partenza >= '2012-01-01'
AND tr.data_partenza < '2012-01-31'
AND (sp.città = 'Modena' OR sa.città = 'Modena')
```

3. Selezionare i dati delle stazioni da cui non sono state partenze nel mese di dicembre 2016. (punti 2+2)

$$S1 = \pi_{partenza} \left( \sigma_{data\_partenza \geq '1-12-2016' \text{ AND } data\_partenza \leq '31-12-2016'}(Tratta) \right) \\ Stazione \bowtie (\pi_{cod\_stazione}(Stazione) - S1)$$

```
SELECT *
FROM Stazione S
WHERE S.cod_stazione NOT IN (
    SELECT partenza
    FROM Tratta
    WHERE data_partenza >= '1-12-2016' AND data_partenza <= '31-12-2016'
)
```

4. Selezionare i dati dei treni che nel 2011 sono arrivati almeno 10 volte nella stessa città (punti 3)

```
SELECT T.*, S.Città
FROM Treno T
JOIN Tratta TR ON T.id_treno = TR.id_treno
JOIN Stazione S ON S.cod_stazione = TR.arrivo
WHERE YEAR(TR.data_partenza) = 2011
GROUP BY T.*, S.Città
HAVING COUNT(*) >= 10
```

5. Selezionare i dati dei treni che nel 2016 hanno effettuato più ore di lavoro (somma delle durate delle tratte) (punti 3)

```
SELECT t.*, SUM(durata) AS ore_lavoro
FROM treno t
JOIN tratta tr ON t.id_treno = tr.id_treno
WHERE tr2.data_partenza >= '1-1-2016'
AND tr2.data_partenza <= '31-12-2016'
GROUP BY t.id_treno
HAVING SUM(durata) >= ALL (
    SELECT SUM(durata)
    FROM tratta tr2
    WHERE tr2.data_partenza >= '1-1-2016'
    AND tr2.data_partenza <= '31-12-2016'
    GROUP BY tr2.id_treno
)
```

6. Creare una vista che mostri per le stesse tratte (ossia stessa partenza e stesso arrivo) i nomi delle stazioni, il modello del treno che ha la durata percorrenza minore su quella tratta e la relativa durata di percorrenza. (punti 3)

```
CREATE VIEW tempo_pcorrenza_min AS
SELECT DISTINCT t.modello, sp.nome AS 'Partenza', sa.nome as 'Arrivo', durata
FROM treno t
JOIN tratta tr ON t.id_treno = tr.id_treno
JOIN stazione sp ON tr.partenza = sp.cod_stazione
JOIN stazione sa ON tr.arrivo = sa.cod_stazione
WHERE durata <= ALL (
    SELECT durata
```

```
FROM tratta tr2
WHERE tr2.partenza = tr.partenza AND tr2.arrivo = tr.arrivo
)
```

**7. Selezionare le città in cui il numero di partenze è superiore al numero di arrivi (punti 3)**

```
SELECT città
FROM stazione s
JOIN tratta t ON t.partenza = s.cod_stazione
GROUP BY città
HAVING COUNT(*) > (
    SELECT COUNT(*)
    FROM stazione s1
    JOIN tratta t1 ON t1.arrivo = s1.cod_stazione
    WHERE s1.città = s.città
    GROUP BY s1.città
)
```