

Cognome e nome: .....

Matricola: .....

### Sistemi Informativi - Prova d'esame del 20/12/2000

**Regole del gioco:** 1) vietato scambiare informazioni con altri; 2) motivare ampiamente tutte le risposte; 3) indicare con chiarezza, cognome, nome e numero di matricola.

## Esercizio 1

Considera il seguente schema di basi di dati relazionale:

**Compagnie**(Codice, Nome, Indirizzo, Telefono)

**Aeroporti**(Codice, Città, Nome)

**TipiAereo**(Codice, Nome, NumeroPosti)

**Voli**(Compagnia, Numero, AeroportoDestinazione, OraPartenza, OraArrivo, TipoAereo)

Supponi che diverse compagnie possano dare lo stesso numero a diversi voli (per esempio: ALITALIA volo 123 e KLM volo 123 sono due voli diversi), ma che una stessa compagnia dia ai suoi voli tutti numeri diversi.

1.1) Indica le chiavi primarie delle relazioni

**Compagnie**(Codice, Nome, Indirizzo, Telefono)

**Aeroporti**(Codice, Città, Nome)

**TipiAereo**(Codice, Nome, NumeroPosti)

**Voli**(Compagnia, Numero, AeroportoDestinazione, OraPartenza, OraArrivo, TipoAereo)

1.2) Indica i vincoli di integrità referenziale

Tra l'attributo Compagnia di **Voli** e la relazione **Compagnie**

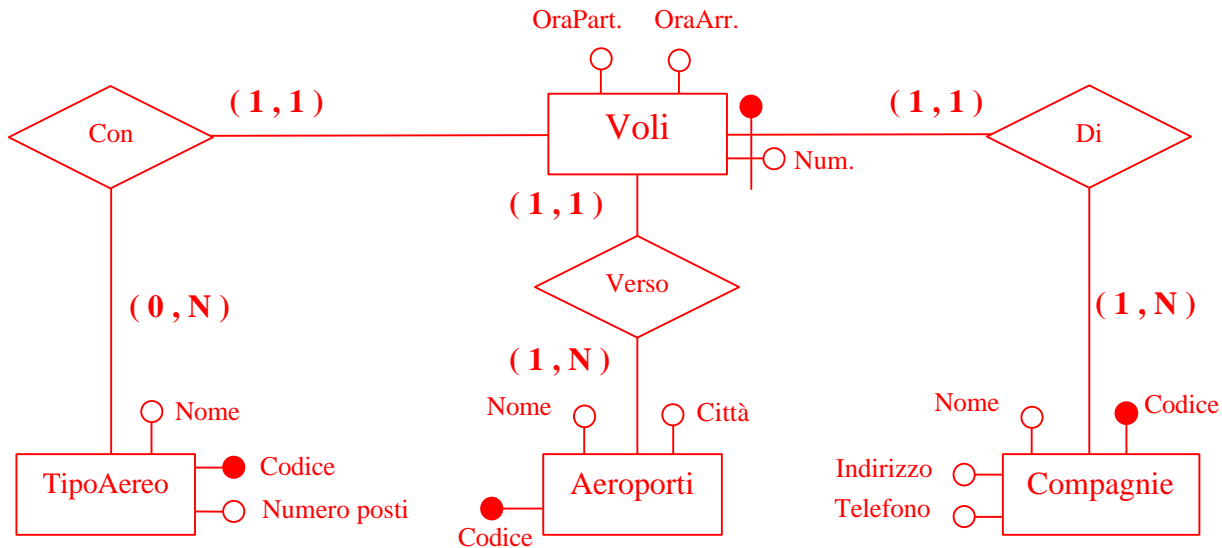
Tra l'attributo AeroportoDestinazione di **Voli** e la relazione **Aeroporti**

Tra l'attributo TipoAereo di **Voli** e la relazione **TipiAereo**

Cognome e nome: .....

Matricola: .....

1.3) Mostra uno schema E-R dal quale lo schema relazionale sopra specificato possa essere stato ottenuto nella fase di progettazione logica



1.4) Formula una interrogazione in SQL che trova l'ora di partenza dei voli per Milano che arrivano prima delle ore 10

```
SELECT OraPartenza
FROM Voli, Aeroporti
WHERE Voli.AeroportoDestinazione = Aeroporti.Codice AND
      Aeroporti.Nome = 'Milano' AND
      Voli.OraArrivo < 10
```

oppure

```
SELECT OraPartenza
FROM Voli
WHERE OraArrivo < 10 AND
      AeroportoDestinazione = ANY (SELECT Codice
                                   FROM Aeroporti
                                   WHERE Nome = 'Milano')
```

1.5) Formula la stessa interrogazione dell'esercizio 1.4 in algebra relazionale.

$$\pi_{\text{OraPartenza}}(\sigma_{\text{OraArrivo} < 10 \wedge \text{Città} = \text{'Milano'}}(\text{Aeroporti} \bowtie_{\text{Codice} = \text{AeroportoDestinazione}} \text{Voli}))$$

oppure

$$\pi_{\text{OraPartenza}}(\sigma_{\text{Città} = \text{'Milano'}}(\text{Aeroporti}) \bowtie_{\text{Codice} = \text{AeroportoDestinazione}} \sigma_{\text{OraArrivo} < 10}(\text{Voli}))$$

Cognome e nome: .....

Matricola: .....

- 1.6) Formula una interrogazione in SQL che trova i voli per Milano che partono prima delle ore 8, producendo il numero del volo, il nome e il numero di telefono della compagnia e l'orario di partenza e di arrivo.

```
SELECT Voli.Numero, Compagnie.Nome, Compagnie.Telefono,
       Voli.OraPartenza, Voli.OraArrivo
FROM Voli, Aeroporti, Compagnie
WHERE Voli.AeroportoDestinazione = Aeroporti.Codice AND
       Voli.Compagnia = Compagnie.Codice AND
       Aeroporti.Nome = 'Milano' AND
       Voli.OraPartenza < 8
```

oppure

```
SELECT Voli.Numero, Compagnie.Nome, Compagnie.Telefono,
       Voli.OraPartenza, Voli.OraArrivo
FROM Voli, Compagnie
WHERE Voli.Compagnia = Compagnie.Codice AND
       Voli.OraPartenza < 8 AND
       Voli.AeroportoDestinazione = ANY (
           SELECT Codice
           FROM Aeroporti
           WHERE Nome = 'Milano')
```

- 1.7) Formula una interrogazione in SQL che trova i nomi dei tipi di aerei utilizzati dalla compagnia di nome 'ALITALIA'.

```
SELECT DISTINCT TipiAereo.Nome
FROM TipiAereo, Voli, Compagnie
WHERE Voli.Compagnia = Compagnie.Codice AND
       Voli.TipoAereo = TipiAereo.Codice AND
       Compagnie.Nome = 'Alitalia'
```

oppure

```
SELECT Nome
FROM TipiAereo
WHERE Codice = ANY (
           SELECT TipoAereo
           FROM Voli
           WHERE Compagnia = ANY (
               SELECT Codice
               FROM Compagnie
               WHERE Nome = 'Alitalia'))
```

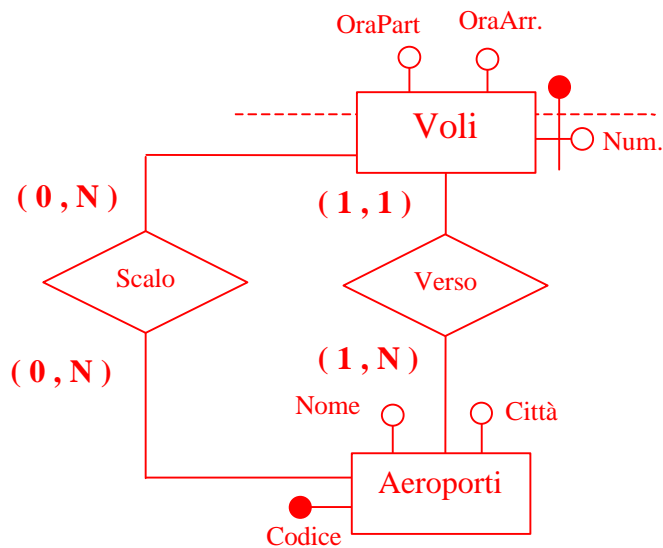
oppure ancora

Cognome e nome: .....

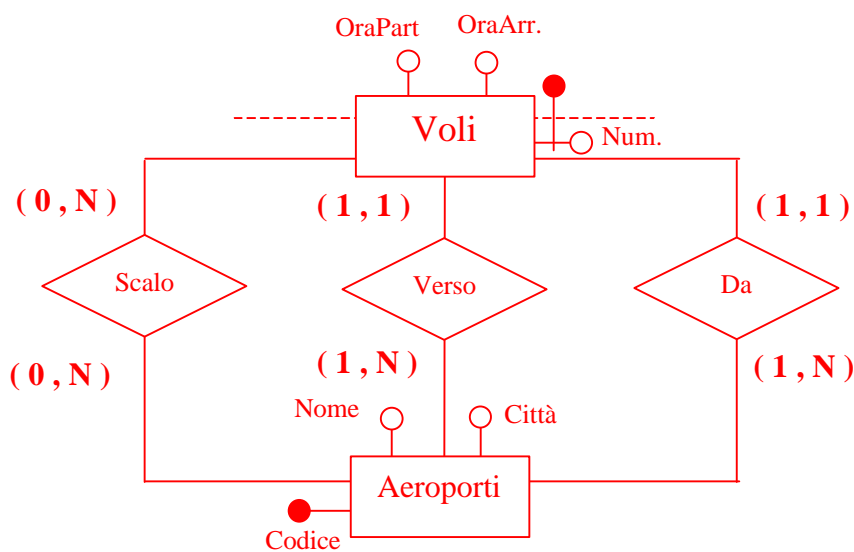
Matricola: .....

```
SELECT Nome
FROM TipiAereo
WHERE Codice = ANY (SELECT TipoAereo
                    FROM Voli, Compagnie
                    WHERE Voli.Compagnia = Compagnie.Codice AND
                      Compagnie.Nome = 'Alitalia')
```

1.8) Modifica lo schema E-R prodotto in risposta alla domanda 1.3 in maniera che sia possibile per un volo avere uno o più scali intermedi



1.9) Modifica lo schema E-R in maniera da considerare anche gli aeroporti di partenza



Cognome e nome: .....

Matricola: .....

## Esercizio 2

Considera il seguente schema di base di dati relazionale:

**Impiegati**(Matricola, Nome, Stipendio, Direttore)

**Dipartimenti**(Codice, Nome, Direttore)

2.1 Spiega in quali casi le due interrogazioni seguenti possono dare risultati diversi

|       |  |
|-------|--|
| 2.1.A | <pre>SELECT AVG(Stipendio) FROM Impiegati WHERE Direttore = ANY (SELECT Direttore                         FROM Dipartimenti)</pre> |
| 2.1.B | <pre>SELECT AVG(Stipendio) FROM Impiegati as I, Dipartimenti as D WHERE I.Direttore = D.Direttore</pre>                            |

Le due interrogazioni danno due risultati diversi nel caso in cui (erroneamente o meno) due dipartimenti abbiano lo stesso direttore. In tal caso l'interrogazione 2.1.A calcola lo stipendio medio degli impiegati sottoposti ad un direttore, mentre la 2.1.B considera due volte lo stipendio degli impiegati con il direttore condiviso.

2.2) Quale (o quali) delle seguenti interrogazioni SQL determina gli impiegati che guadagnano più del loro direttore?

|       |  |
|-------|--|
| 2.2.A | <pre>SELECT I.Nome FROM Impiegati AS I, Impiegati AS D WHERE I.Direttore = D.Matricola AND       I.Stipendio = D.Stipendio</pre>   |
| 2.2.B | <pre>SELECT I.Nome FROM Impiegati AS I WHERE I.Stipendio &gt; ALL (SELECT D.Stipendio                         FROM Impiegati AS D                         WHERE I.Direttore = D.Matricola)</pre> |
| 2.2.C | <pre>SELECT I.Nome FROM Impiegati AS D WHERE D.Stipendio &gt; ANY (SELECT I.Stipendio                         FROM Impiegati AS I                         WHERE I.Matricola = D.Direttore)</pre> |

La 2.2.A non va bene perché ha solo una condizione di ugualianza sullo stipendio.

La 2.2.B va bene: il risultato dell'interrogazione annidata è lo stipendio del direttore dell'impiegato corrente.

La 2.2.C non va bene: non è corretto l'uso della variabile I (target list, prima riga) che è definita solo nella clausola FROM annidata. Se la prima riga fosse "SELECT D.Nome" l'interrogazione sarebbe corretta, essendo uguale all'interrogazione 2.2.B in cui la variabile "I" è stata scambiata ovunque con "D".

Cognome e nome: .....

Matricola: .....

2.3) Quale (o quali) delle seguenti interrogazioni SQL determina gli impiegati che guadagnano lo stipendio più alto?

|       |  |
|-------|--|
| 2.3.A | SELECT Nome<br>FROM Impiegati<br>WHERE Stipendio = ALL (SELECT MAX(Stipendio)<br>FROM Impiegati) |
| 2.3.B | SELECT Nome<br>FROM Impiegati<br>WHERE Stipendio > ALL (SELECT Stipendio<br>FROM Impiegati)      |
| 2.3.C | SELECT Nome<br>FROM Impiegati<br>WHERE Stipendio > ANY (SELECT Stipendio<br>FROM Impiegati)      |

La 2.3.A va bene. Il risultato dell'interrogazione annidata è lo stipendio massimo.

La 2.3.B non va bene: si impone che lo stipendio dell'impiegato sia maggiore di *tutti* gli stipendi, e quindi anche di se stesso. Il risultato dell'interrogazione è sempre nullo.

La 2.3.C non va bene: vengono selezionati gli impiegati il cui stipendio sia maggiore di almeno un altro stipendio (ANY). Il risultato dell'interrogazione sono tutti gli impiegati eccetto quello che guadagna meno.