Basi di Dati - Prof. G. Polese	**Anno Accademico 2009/2010**
Classwork N.1	**del 12/4/2010**

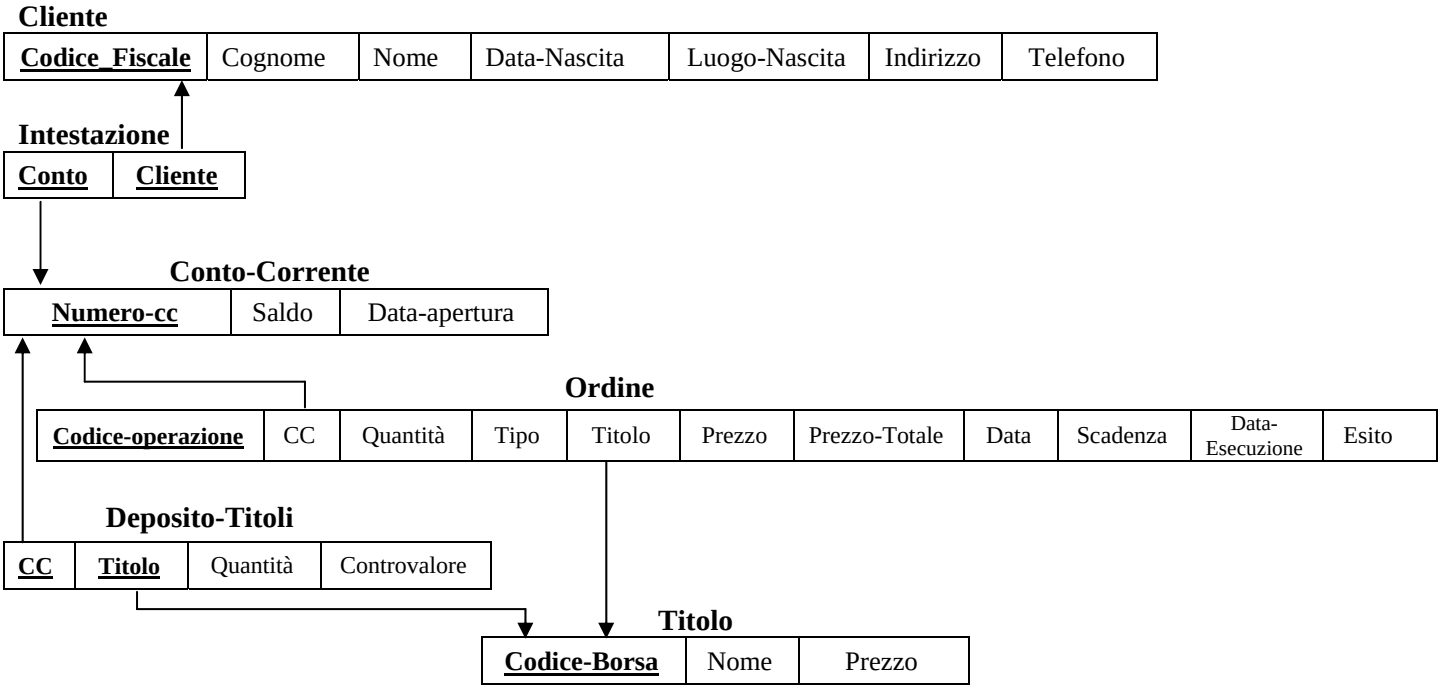
Nome e Cognome: **Matricola/Alias:**

(Scrivere solo nello spazio bianco. Se necessario, usare il retro del foglio. Non sono ammessi elaborati su fogli diversi.)

Una banca vi ha commissionato l'applicazione per memorizzare gli ordini di borsa da parte dei suoi clienti. Da un lato ci sono quindi i possessori di conto corrente presso la banca, mentre dall'altra ci sono i vari titoli di borsa. Ciascun correntista può richiedere un ordine di acquisto di un titolo, specificando la data di scadenza dell'ordine, il tipo di operazione (acquisto o vendita), la quantità ed il prezzo desiderato (questa è una semplificazione rispetto alla realtà). L'operazione andrà a buon fine solo se nell'intervallo di tempo specificato l'azione raggiunge il prezzo desiderato e il programma fa in tempo a completare l'operazione. Anche in questo caso facciamo l'ipotesi semplificativa che l'operazione non viene mai eseguita parzialmente, quindi ci sono due possibile esiti finali: *eseguito* o *non eseguito*. Ciascun ordine eseguito con successo da luogo alla variazione del deposito titoli, in quanto occorre aggiungere eventuali titoli acquistati (o aumentarne la quantità se già se ne sono acquistati altri uguali in precedenza) ed eliminare quelli venduti (o sottrarre la quantità venduta se questa non esaurisce la disponibilità del relativo titolo). Ciascun ordine andato a buon fine è inoltre accompagnato dal prezzo totale dato dal prodotto del prezzo per il numero di titoli acquistati o venduti nell'ambito dell'ordine.

Esercizio1 (10 Punti)

Disegnare graficamente lo schema logico relazionale del database relativo all'applicazione descritta, includendo sottolineature per indicare chiavi primarie e frecce per eventuali chiavi esterne. Il conto corrente è caratterizzato da un numero di conto, più dati anagrafici dell'intestatario ed il saldo disponibile. Ciascun titolo è invece caratterizzato da un codice di borsa, un nome, un prezzo corrente (non ci interessa modellare la dinamica quotidiana del prezzo). Gli attributi delle tabelle per memorizzare ordini e deposito titoli si possono evincere dal testo riportato sopra.



Tradurre inoltre lo schema disegnato nel DDL di SQL, specificando le seguenti politiche di gestione dell'integrità referenziale:

1) In caso di modifica del nome e/o codice di un titolo, ripercuotere la modifica su tutti i depositi di clienti che ne possiedono delle quote, mentre ordini che fanno riferimento ad esso possono essere posti a null.
2) In caso di cancellazione di un cliente eliminare il suo deposito titoli e gli ordini impartiti.

```
CREATE TABLE Cliente (
        Codice_Fiscale      CHAR(16)        PRIMARY KEY,
        Cognome             VARCHAR(20)     NOT NULL,
        Nome                VARCHAR(20)     NOT NULL,
        Data-Nascita        DATE            NOT NULL,
        Luogo-Nascita       VARCHAR(20)     NOT NULL,
        Indirizzo           VARCHAR(30),
        Telefono            VARCHAR(15)
)

CREATE TABLE Conto-Corrente (
        Numero_cc           CHAR(10)        PRIMARY KEY,
        Saldo               NUMERIC(10,2)   DEFAULT 0,
        Data-Apertura       DATE
)

CREATE TABLE Intestazione (
        Conto               CHAR(10)    References    Conto-Corrente(Numero-cc),
        Cliente             CHAR(16)    References    Cliente(Codice_Fiscale),

        PRIMARY KEY(Conto, Cliente)
)

CREATE TABLE Titolo(
        Codice_Borsa        CHAR(10)        PRIMARY KEY,
        Nome                VARCHAR(15)     NOT NULL,
        Prezzo              NUMERIC(9,2)
)

CREATE TABLE Ordine (
        Codice_operazione   CHAR(10)    PRIMARY KEY,
        CC                  CHAR(10)    References    Conto-Corrente(Numero-cc)
                                ON UPDATE CASCADE    ON DELETE CASCADE,
        Titolo              CHAR(10)    References    Titolo(Codice_Borsa)
                                ON UPDATE SET NULL    ON DELETE SET NULL
        Quantità            INTEGER,
        Tipo                VARCHAR(8),
        Prezzo              NUMERIC(9,2),
        Prezzo-Totale       NUMERIC(10,2),
        Data                DATE,
        Scadenza            DATE,
        Data-Esecuzione     DATE,
```

```
    Esito           VARCHAR(12)
)

CREATE TABLE    Deposito-Titoli (
    CC              CHAR(10)    References  Conto-Corrente(Numero-cc)
                        ON UPDATE CASCADE   ON DELETE CASCADE,
    Titolo          CHAR(10)    References  Titolo(Codice_Borsa)
                        ON UPDATE CASCADE
    Quantità        INTEGER,
    Controvalore    NUMERIC(10,2),
)
```

Esercizio2 (10 punti)
Usando l'algebra relazionale, scrivere le seguenti query:

1) Elencare i dati dei clienti che hanno acquistato una certa azione (denotarla con codice di borsa TIT_xxx) almeno una volta.

$PROJ_{Codice\_Fiscale,\ Cognome,\ Nome}$ (Cliente $JOIN_{Cliente=Codice\_Fiscale}$
(Intestazione $JOIN_{Conto=Numero\text{-}cc}$ (Conto-Corrente
$JOIN_{CC=Numero\text{-}cc}(SEL_{Esito='Eseguito'\ AND\ Titolo=TIT\_xxx}(ORDINE))$)))

2) Elencare i dati dei clienti che in un dato mese non hanno effettuato ordini.

$PROJ_{Codice\_Fiscale,\ Cognome,\ Nome}$ (Cliente) –

$PROJ_{Codice\_Fiscale,\ Cognome,\ Nome}$ (Cliente $JOIN_{Cliente=Codice\_Fiscale}$
(Intestazione $JOIN_{Conto=Numero\text{-}cc}$ (Conto-Corrente
$JOIN_{CC=Numero\text{-}cc}$ ($SEL_{Data\text{-}Esecuzione >= '01/X/Y'\ AND\ Data\text{-}Esecuzione <= '31/X/Y'}$ (ORDINE)))))

Esercizio3 (10 Punti)
Usando il linguaggio SQL scrivere una query per elencare i dati di clienti che in un dato anno hanno speso almeno 100.000 € in acquisto di titoli.

```
SELECT      Codice_Fiscale, Cognome, Nome
FROM        Cliente JOIN Intestazione I on I.Cliente = Codice_Fiscale JOIN
            Conto-Corrente on Conto=Numero-cc JOIN Ordine on CC = Numero-cc
WHERE       Tipo='Acquisto' AND Data-Esecuzione  LIKE '??-??-2010'
            AND Esito = 'Eseguito'
GROUP BY    CC
HAVING      SUM (Prezzo-Totale) >= 100.000
```