

## Esercitazione 8 – Mercoledì 21 gennaio 2015 (2 ore) DDL e progettazione

### Testi degli esercizi

#### Esercizio 1 (Tema d'esame del 20 settembre 2012)

Si consideri il seguente schema di base di dati che vuole tenere traccia delle attività di una piscina nell'anno 2012/2013.

VASCA (CodVasca, Profondità)

CORSO (CodCorso, Descrizione, Tipo, DataInizio, DataFine)

CALENDARIO (CodiceCorso, Vasca, Giorno, OraInizio, Insegnante, Durata)

PERSONA (CE, Nome, Cognome, Indirizzo, Città, Tel)

FREQUENZA (CodiceCorso, Vasca, Giorno, OraInizio, Cliente, DataDiIscrizione)

Il campo Tipo nella tabella CORSO specifica se il corso è “Monosettimanale” o “Bisettimanale”. La tabella PERSONA contiene i dati di clienti e insegnanti.

- a) Considerando lo schema di base di dati in testa al tema d'esame, specificare in SQL la creazione delle tabelle CORSO e FREQUENZA, definendo i vincoli di tupla e di dominio ritenuti opportuni ed esprimendo eventuali vincoli di integrità referenziale relativi a tutte le tabelle dello schema.
- b) Specificare in SQL il vincolo che controlla che la data di iscrizione non sia posteriore alla data di fine del corso.
- c) [*NON RICHIESTO DAL TEMA D'ESAME*] Specificare in SQL il vincolo che controlla che in uno stesso giorno non ci siano più di 50 persone che frequentano lezioni.
- 

#### Esercizio 2

STUDENTE (Matricola, CodiceFiscale, DataNascita, NumeroEsamiSostenuti)

Definire in SQL la tabella Studente sapendo che:

- La matricola è un codice costituito da 6 caratteri.
  - Il codice fiscale, anche se non è chiave primaria, non ammette duplicati.
  - La data di nascita non deve mai assumere un valore nullo, e non deve essere precedente al 1900.
  - L'attributo NumeroEsamiSostenuti di default assume valore 0, e non può essere negativo.
-

### **Esercizio 3 (Tema d'esame del 3 luglio 2012)**

Si richiede di progettare la base di dati di un'azienda che fabbrica e vende divani su commessa. L'azienda mantiene un catalogo dei modelli base che i clienti possono considerare come riferimento per la definizione del proprio ordine. Di ogni modello interessa il codice identificativo, il nome e la descrizione. Ogni modello può essere realizzato in diversi tessuti e in diversi colori. Di ogni tessuto si registra il codice del tessuto, il nome e l'insieme di colori disponibili. Di ogni divano ordinato interessa il numero progressivo (unico per ogni modello), il modello di riferimento, il numero di posti, il tessuto e il colore scelto, il numero di giorni previsti per la sua realizzazione e il prezzo. Di ogni divano interessa inoltre sapere i dettagli dell'ordine: il cliente che ha effettuato l'ordine e la data e ora. Ciascun cliente può effettuare al più un ordine in una certa ora di una certa data. Di ogni cliente interessa il codice fiscale, la data di nascita, numero di telefono e l'indirizzo di residenza composto da via e città di residenza. Gli ordini si suddividono in processati e non processati. Di ogni ordine non processato interessa la data prevista per la consegna. Di ogni ordine processato interessa la consegna (una ed una sola) attraverso cui è stato processato. Di ogni consegna, che può essere relativa a più di un ordine, interessa la data, l'ora, la durata, e il corriere usato. Di ogni corriere interessa il codice, il nome, il numero di telefono e la città in cui opera.

---

### ***Esercizio per casa n. 1 (Tema d'esame del 11 settembre 2014)***

VOLO (Codice, Compagnia, AeropPartenza, AeropArrivo, OraPart, OraArrivo, TipoAereo)  
PRENOTAZIONE (IdPasseggero, CodVolo, DataVolo, Nome, Cognome, Email, DataNascita)  
CHECKIN (IdPasseggero, CodVolo, DataVolo, Posto, OraEffettivaCheckIn, Priorità)

C1) Considerando lo schema di base di dati in testa al tema d'esame, specificare in SQL la creazione delle tabelle VOLO, PRENOTAZIONE e CHECKIN, definendo i vincoli di tupla e di dominio ritenuti opportuni ed esprimendo eventuali vincoli di integrità referenziale relativi a tutte le tabelle dello schema.

C2) Considerando lo schema di base di dati in testa al tema d'esame, specificare in SQL il vincolo che "la prenotazione è sempre obbligatoria".

---

### ***Esercizio per casa n. 2 (Tema d'esame del 5 settembre 2012)***

Si richiede di progettare la base di dati per una catena di supermercati. Ogni supermercato è identificato da un codice ed è caratterizzato da indirizzo, città, numero di telefono e nome del responsabile. I supermercati della catena rilasciano delle carte fedeltà. Ogni carta fedeltà è caratterizzata da un codice univoco, dal cliente titolare della carta e dai punti accumulati. Per ogni cliente titolare di una carta fedeltà si tiene traccia delle seguenti informazioni: codice fiscale, nome, cognome, indirizzo e recapito telefonico (facoltativo).

I prodotti commercializzati dalla catena di supermercati sono caratterizzati da un codice univoco, dal nome del prodotto, dal nome del produttore e dal prezzo consigliato. Si vuole tenere traccia degli scontrini emessi dai vari supermercati della catena. Ogni scontrino è caratterizzato da data di emissione, cassa da cui è stato emesso, numero progressivo giornaliero (il numero progressivo giornaliero è univoco all'interno della cassa), elenco dei prodotti acquistati con relative quantità e prezzi. Per ogni scontrino si deve tenere traccia anche dell'eventuale tessera fedeltà usata durante l'acquisto. Ogni cassa del supermercato è identificata da un codice univoco ed è caratterizzata dal supermercato presso cui è collocata, dalla data in cui è stata installata e dal modello.

## Soluzioni

### Esercizio 1

a) Considerando lo schema di base di dati in testa al tema d'esame, specificare in SQL la creazione delle tabelle CORSO e FREQUENZA, definendo i vincoli di tupla e di dominio ritenuti opportuni ed esprimendo eventuali vincoli di integrità referenziale relativi a tutte le tabelle dello schema.

```
CREATE TABLE Corso (  
    CodCorso VARCHAR(255) PRIMARY KEY,  
    Descrizione VARCHAR(255),  
    Tipo ENUM('Monosettimanale', 'Bisettimanale'),  
    DataInizio DATE,  
    DataFine DATE  
)
```

```
CREATE TABLE Frequenza (  
    CodiceCorso VARCHAR(255),  
    Vasca VARCHAR(255),  
    Giorno DATE,  
    OraInizio TIME,  
    Cliente CHAR(16) REFERENCES Persona(CF) ON UPDATE CASCADE ON  
        DELETE CASCADE,  
    DataDiIscrizione DATE,  
    PRIMARY KEY (CodiceCorso, Vasca, Giorno, OraInizio, Cliente),  
    FOREIGN KEY (CodiceCorso, Vasca, Giorno, OraInizio) REFERENCES  
        Calendario(CodiceCorso, Vasca, Giorno, OraInizio) ON UPDATE CASCADE ON  
        DELETE NO ACTION  
)
```

b) Specificare in SQL il vincolo che controlla che la data di iscrizione non sia posteriore alla data di fine del corso.

```
CREATE ASSERTION VerificaDate CHECK (  
    NOT EXISTS (  
        SELECT *  
        FROM Frequenza AS f, Corso AS c  
        WHERE f.CodiceCorso=c.CodCorso AND f.DataDiIscrizione>c.DataFine  
    )  
)
```

c) *[NON RICHIESTO DAL TEMA D'ESAME]* Specificare in SQL il vincolo che controlla che in uno stesso giorno non ci siano più di 50 persone che frequentano lezioni.

```
CREATE ASSERTION VerificaNumeroPresenti CHECK (  
    50 >= ALL ( SELECT COUNT(*)  
                FROM Frequenza  
                GROUP BY Giorno  
            )  
)
```

---

## Esercizio 2

STUDENTE (Matricola, CodiceFiscale, DataNascita, NumeroEsamiSostenuti)

Definire in SQL la tabella Studente sapendo che:

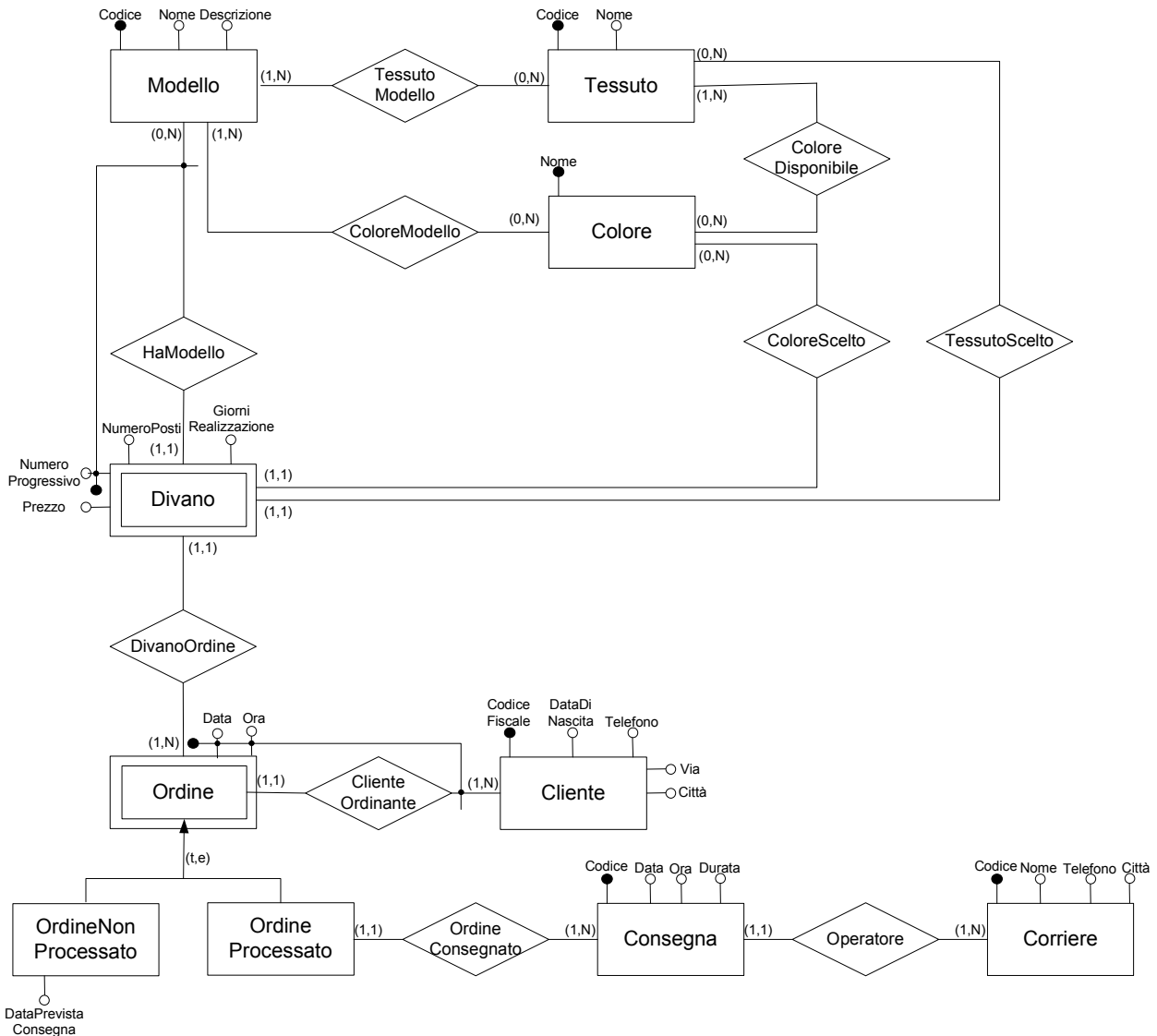
- La matricola è un codice costituito da 6 caratteri.
- Il codice fiscale, anche se non è chiave primaria, non ammette duplicati.
- La data di nascita non deve mai assumere un valore nullo, e non deve essere precedente al 1900.
- L'attributo NumeroEsamiSostenuti di default assume valore 0, e non può essere negativo.

```
CREATE TABLE Studente (  
    Matricola CHAR(6) PRIMARY KEY,  
    CodiceFiscale CHAR(16) UNIQUE,  
    DataNascita DATE NOT NULL CHECK (DataNascita >= '1900-01-01'),  
    NumeroEsamiSostenuti INTEGER DEFAULT 0 CHECK (NumeroEsamiSostenuti >= 0)  
)
```

---

### Esercizio 3

#### Schema concettuale



#### Schema logico

Modello(Codice, Nome, Descrizione)

Tessuto(Codice, Nome)

TessutoModello(Modello, Tessuto)

Colore(Nome)

ColoreModello(Modello, Colore)

Divano(Modello, NumeroProgressivo, Prezzo, NumeroPosti, GiorniRealizzazione, Colore, Tessuto, DataOrdine, OraOrdine, Cliente)

Ordine(Data, Ora, Cliente, Tipo, DataPrevistaConsegna\*, Consegna\*)

Cliente(CodiceFiscale, DataDiNascita, Telefono, Via, Città)

Consegna(Codice, Data, Ora, Durata, Corriere)

Corriere(Codice, Nome, Telefono, Città)

*Chiavi esterne*

TessutoModello.Modello → Modello.Codice

TessutoModello.Tessuto → Tessuto.Codice

ColoreModello.Modello → Colore.Nome

ColoreModello.Colore → Colore.Nome

Divano.Modello → Modello.Codice

Divano.(DataOrdine, OraOrdine, Cliente) → Ordine.(Data, Ora, Cliente)

Ordine.Cliente → Cliente.CodiceFiscale

Ordine.Consegna → Consegna.Codice

Consegna.Corriere → Corriere.Codice

---

***Esercizio per casa n. 1***

C1) Considerando lo schema di base di dati in testa al tema d'esame, specificare in SQL la creazione delle tabelle VOLO, PRENOTAZIONE e CHECKIN, definendo i vincoli di tupla e di dominio ritenuti opportuni ed esprimendo eventuali vincoli di integrità referenziale relativi a tutte le tabelle dello schema.

**CREATE TABLE** Volo (

Codice **VARCHAR**(255) **PRIMARY KEY**,

Compagnia **VARCHAR**(255),

AeropPartenza **VARCHAR**(255),

AeropArrivo **VARCHAR**(255)

OraPart **TIME**,

OraArrivo **TIME**,

TipoAereo **VARCHAR**(255)

)

**CREATE TABLE** Prenotazione (

IdPasseggero **VARCHAR**(255),

CodVolo **VARCHAR**(255) **REFERENCES** Volo(Codice) **ON UPDATE CASCADE ON DELETE NO ACTION**,

DataVolo **DATE**,

Nome **VARCHAR**(255),

Cognome **VARCHAR**(255),

Email **VARCHAR**(255),

**PRIMARY KEY** (IdPasseggeri, CodVolo, DataVolo)

)

```
CREATE TABLE CheckIn (  
    IdPasseggero VARCHAR(255),  
    CodVolo VARCHAR(255),  
    DataVolo DATE,  
    Posto INTEGER,  
    OraEffettivaCheckIn TIME,  
    Priorità BOOLEAN,  
    PRIMARY KEY (IdPasseggero, CodVolo, DataVolo),  
    FOREIGN KEY (IdPasseggero, CodVolo, DataVolo) REFERENCES  
        Prenotazione(IdPasseggero, CodVolo, DataVolo) ON UPDATE CASCADE ON  
        DELETE CASCADE  
)
```

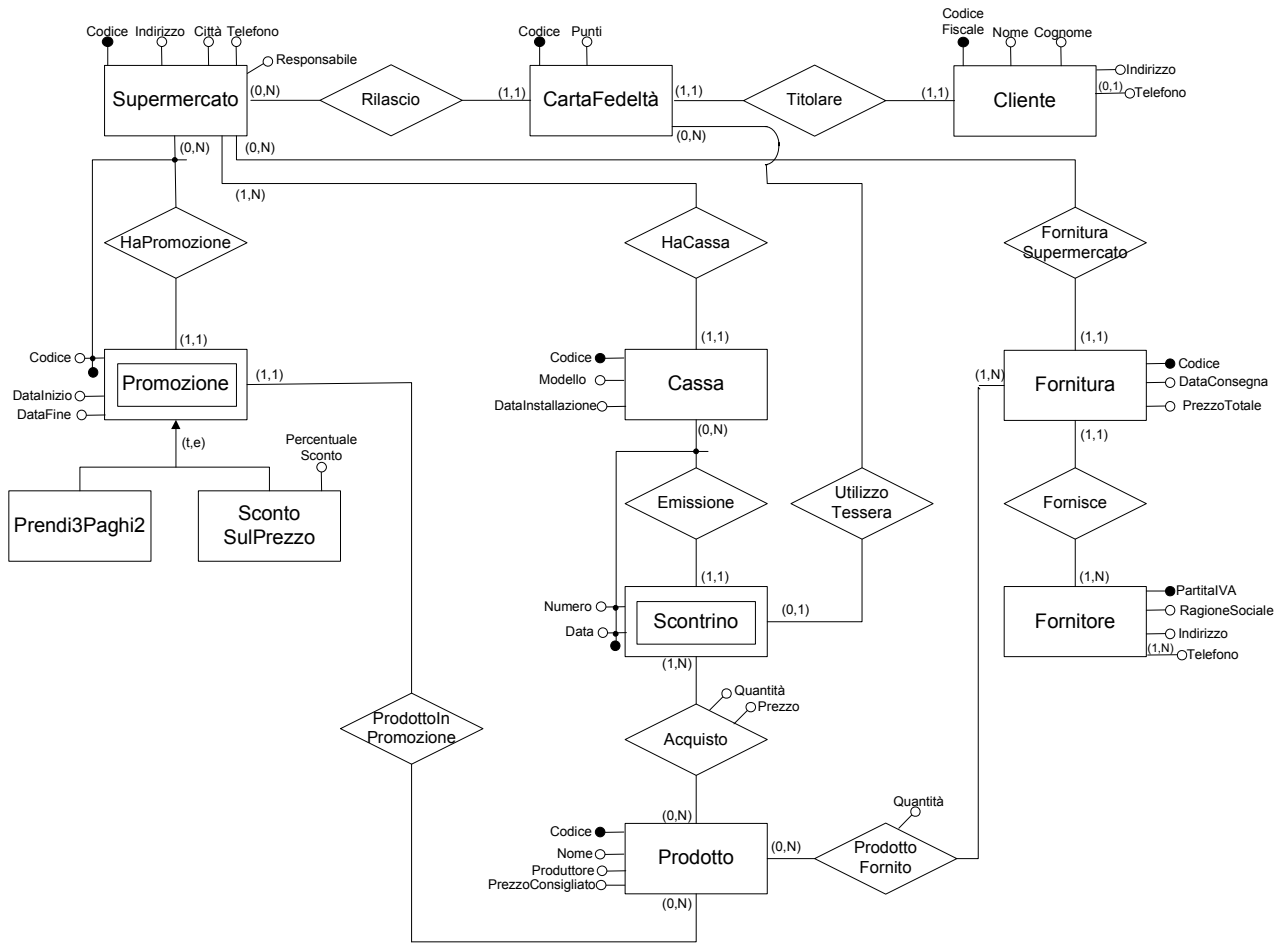
C2) Considerando lo schema di base di dati in testa al tema d'esame, specificare in SQL il vincolo che "la prenotazione è sempre obbligatoria".

```
CREATE ASSERTION PrenotazioneObbligatoria CHECK (  
    NOT EXISTS (  
        SELECT *  
        FROM CheckIn  
        WHERE (IdPasseggero, CodVolo, DataVolo) NOT IN (SELECT IdPasseggero,  
            CodVolo, DataVolo FROM Prenotazione)  
    )  
)
```

---

## Esercizio per casa n. 2

### Schema concettuale



### Schema logico

Supermercato(Codice, Indirizzo, Città, Telefono, Responsabile)

CartaFedeltà(Codice, Punti, SupermercatoRilascio, CodiceFiscale, Nome, Cognome, Indirizzo, Telefono)

Prodotto(Codice, Nome, Produttore, PrezzoConsigliato)

Promozione(Supermercato, Codice, Indirizzo, Città, Telefono, Tipo, PercentualeSconto\*)

Cassa(Codice, Modello, DataInstallazione, Supermercato)

Scontrino(Cassa, Numero, Data, Tessera\*)

Acquisto(CassaScontrino, NumeroScontrino, DataScontrino, Prodotto, Quantità, Prezzo)

Fornitore(PartitaIVA, RagioneSociale, Indirizzo)

TelefonoFornitore(PartitaIVA, Telefono)

Fornitura(Codice, Supermercato, DataConsegna, PrezzoTotale, Fornitore)

### Chiavi esterne

CartaFedeltà.SupermercatoRilascio → Supermercato.Codice

Promozione.Supermercato → Supermercato.Codice

Cassa.Supermercato → Supermercato.Codice

Scontrino.Cassa → Cassa.Codice

Scontrino.Tessera → CartaFedeltà.Codice

Acquisto.(CassaScontrino, NumeroScontrino, DataScontrino) → Scontrino.(Cassa, Numero, Data)



Acquisto.Prodotto → Prodotto.Codice  
TelefonoFornitore.PartitaIVA → Fornitore.PartitaIVA  
Fornitura.Supermercato → Supermercato.Codice  
Fornitura.Fornitore → Fornitore.PartitaIVA