



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Corso di Laurea in Informatica

Esame di Basi di Dati

Esame del 24 Febbraio 2020

Regole dell'esame:

- Non è possibile utilizzare alcun materiale, né appunti, né il libro.
- **Inserire le risposte nei riquadri che seguono i testi degli esercizi**
- L'uso della matita per mostrare la soluzione degli esercizi è fatta a rischio e pericolo degli studenti. Nessuna rivendicazione verrà accettata in caso parte della soluzione si cancelli, per via delle caratteristiche transitorie dei tratti a matita.
- Il massimo possibile è 32 punti
- Occorre avere acquisito almeno metà dei punti dell'Esercizio 3 (cioè 3.5 punti).
- **Coloro che siano sorpresi a copiare o a far copiare:**
 - Dovranno lasciare l'aula dell'esame
 - Riceveranno 0 punti "di ufficio"
 - Dovranno saltare l'appello successivo

Nome: _____ Num. Matric.: _____

Esercizio 1: Diagramma ER (8 punti)

*Si richiede di produrre lo schema concettuale Entità-Relazione di un'applicazione relativa ad una catena di **pista di pattinaggio**, evitando di introdurre entità non necessarie. Disegnare il diagramma ER nel riquadro della pagina che segue.*

Il sistema deve memorizzare le informazioni di ogni pista della catena. Di ogni pista, è di interesse sapere (1) la città in cui si trova, (2) l'anno di apertura e (3) il fatturato. Ogni città non può avere più di una pista. Inoltre, ogni pista è gestita da una persona, però una pista potrebbe temporaneamente non avere una persona che la gestisce. Si noti che una persona non può prendere in gestione più di una pista della catena.

Il sistema memorizza un'anagrafe delle persone: di ogni persona, è di interesse sapere il codice fiscale (identificativo), la data di nascita e la città di nascita.

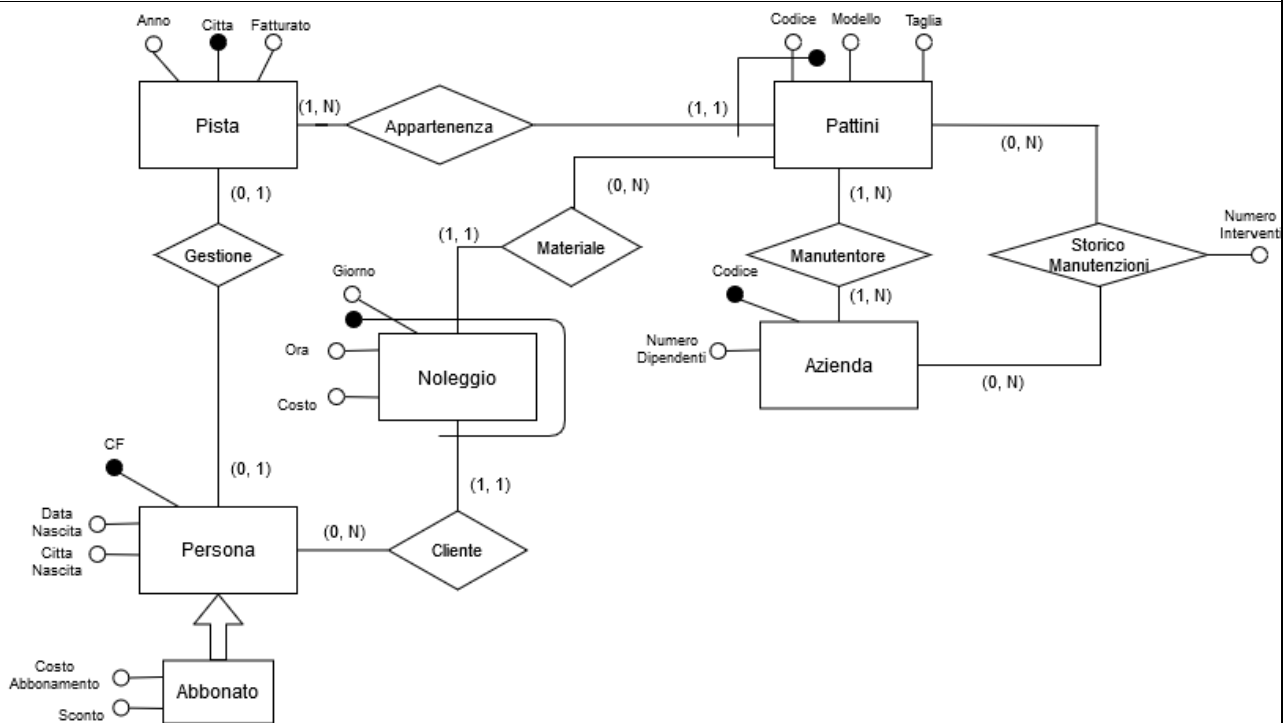
Quando le persone desiderano essere clienti di una pista, noleggiando un paio di pattini. Ogni paio di pattini è associato ad esattamente una pista.

Di ogni paio di pattini interessa il codice (unico all'interno della pista), il modello, la taglia. Inoltre, per ogni paio di pattini, è di interesse sapere quali aziende sono in grado di farne manutenzione. Si vuole anche memorizzare lo storico delle manutenzioni: per ogni paio di pattini, è di interesse sapere quante volte (intero) il paio è andato in manutenzione in ogni azienda in grado di mantenere. Di ogni azienda di riparazione pattini interessa il codice identificativo e il numero di dipendenti.

Come detto, ogni paio di pattini è noleggiato da persone per usarle nella pista associata. In particolare, si vuole sapere per ogni noleggio: (1) quale paio è stato preso, (2) il giorno e l'ora del noleggio e (3) il costo complessivo del noleggio. Si noti che ogni paio di pattini può essere noleggiato più volte dalla stessa persona, ma solo in giorni diversi.

Alcune persone sono clienti abbonati. Per tali clienti, è di interesse sapere il costo dell'abbonamento annuale (unico per cliente) e quale sconto (in percentuale) ottengono nei noleggi in virtù del loro abbonamento. Si noti che, se una persona è abbonata, non può prendere in gestione una pista (occorre modellare questo vincolo esplicitamente nello ER). Viceversa, una persona che non è abbonata può noleggiare pattini ed essere gestore di una pista. Non è necessario modellare il vincolo che non è possibile noleggiare pattini relativi all'eventuale pista in gestione.

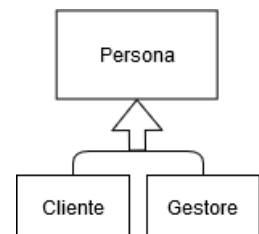
Soluzione Esercizio 1



Di seguito sono alcuni punti della soluzione che meritano un approfondimento:

- L'identificatore dell'entità *Pista* è la città, poiché non è possibile avere due piste nella stessa città. L'aggiunta di un codice come identificatore è errato perché non è previsto dalle specifiche, né necessario. Inoltre, tale codice come identificatore non permetterebbe di rappresentare il vincolo che non è possibile avere due istanze dell'entità *Pista* con lo stesso valore per l'attributo Città.
- Il codice non è sufficiente come identificatore dell'entità *Pattini* perché è unico per i pattini di una pista, ma non unico per tutti i pattini della catena.
- Si noti l'identificatore dell'entità *Noleggio*. L'identificatore modella che non è possibile che la stessa persona noleggi gli stessi pattini nello stesso giorno. Si osservi che l'ora non è parte dell'identificatore, altrimenti sarebbe possibile noleggiare i pattini ad orari diversi dello stesso giorno.
- **È errato realizzare *Noleggio* come relazione tra *Persona* e *Pattini* invece di entità come proposto sopra.** Se *Noleggio* fosse una relazione, non sarebbe possibile noleggiare gli stessi pattini più di una volta dalla parte della stessa persona: il testo indica che tale eventualità è possibile ma in giorni diversi.

L'ultima osservazione riguarda la possibilità qui a fianco (con *Persona* che generalizza le entità *Cliente* e *Gestore*). Tale soluzione è errata. La generalizzazione è *esclusiva*: vorrebbe dire che una persona può essere solamente cliente oppure gestore. Si noti che il fatto che la generalizzazione sia totale (freccia piena) o parziale (freccia vuota) non cambia nulla.



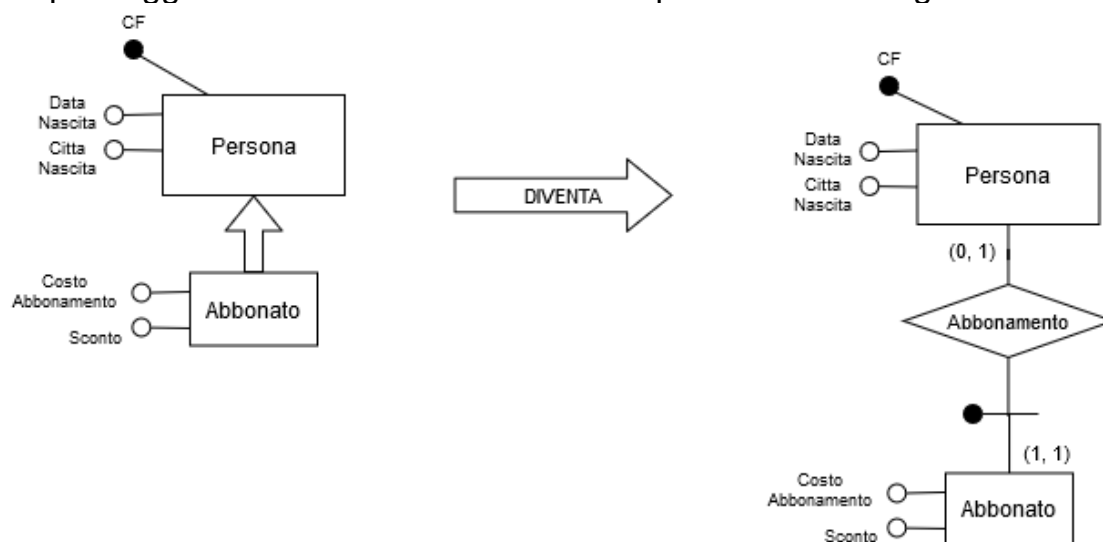
Nome: _____ Num. Matric.: _____

Esercizio 2: Progettazione Logica (7 punti)

A partire dallo ER concettuale dell'Esercizio 1, produrre uno schema relazionale del database nel riquadro sottostante, minimizzando i valori nulli. Indicare la chiave primaria, i vincoli di chiave esterne e gli attributi che ammettono valori nulli. Illustrare come ristrutturare il diagramma ER per essere direttamente traducibile in uno schema relazionale.¹

Soluzione Esercizio 2

Il primo passaggio è la ristrutturazione dello ER per rimuovere la generalizzazione:



Alla luce di quanto sopra, lo schema logico è il seguente:

PERSONA(CF, DataNascita, CittàNascita)

PISTA(Città, Fatturato, Anno, **Gestore**)

- Chiave Esterna: PISTA.Gestore → Persona.CF
- Chiave Aggiuntiva: Gestore

PATTINI(Codice, CittàPista, Modello, Taglia)

- Chiave Esterna: PATTINI.CittàPista → PISTA.Città

NOLEGGIO(CodicePattini, CittàPista, Persona, Giorno, Ora, Costo)

- Chiave Esterna: NOLEGGIO.(CodicePattini, CittàPista) → PATTINI.(Codice, CittàPista)
- Chiave Esterna: NOLEGGIO.Persona → Persona.CF

AZIENDA(Codice, NumeroDipendenti)

MANUTENTORE(CodicePattini, CittàPista, CodiceAzienda)

- Chiave Esterna: MANUTENTORE.(CodicePattini, CittàPista) → PATTINI.(Codice, CittàPista)
- Chiave Esterna: MANUTENTORE.CodiceAzienda → AZIENDA.Codice

STORICO(CodicePattini, CittàPista, CodiceAzienda, NumeroInterventi)

- Chiave Esterna: STORICO.(CodicePattini, CittàPista) → PATTINI.(Codice, CittàPista)
- Chiave Esterna: STORICO.CodiceAzienda → AZIENDA.Codice

ABBONATO(CF, Costo, Sconto)

¹ Allo scopo di mostrare la ristrutturazione del diagramma ER, è possibile semplicemente mostrare i cambiamenti apportati al diagramma nel riquadro soluzione dell'Esercizio 1, utilizzando una penna di diverso colore (non rossa!)

Esercizio 3: Algebra Relazionale & SQL (7 punti)

Si consideri la seguente base di dati per la registrazione di libri e dei loro autori, dove i libri sono identificati dal nome:

- AUTORE(CF, Nome, Eta)
- PUBBLICA(CF, Libro)

A. Nel riquadro, scrivere una Query in Algebra Relazione che restituisce il codice fiscale e il nome degli autori che hanno pubblicato almeno due libri. (2 punti).²

P1 = PUBBLICA

P2 = PUBBLICA

$\Pi_{\text{AUTORE.CF, AUTORE.NOME}}$

$(\text{AUTORE} \bowtie_{\text{AUTORE.CF=P1.CF}}$

$(\text{P1} \bowtie_{\text{P1.CF = P2.CF AND P1.Libro} \neq \text{P2.Libro}} \text{P2}))$

² Si assuma che l'operatore di join $A \bowtie B$ **senza condizioni** mantenga le tuple di $A \times B$ con valori uguali su attributi uguali (join naturale). Se **una condizione C è specificata**, \bowtie_C mantiene le tuple di $A \times B$ per cui la condizione C è vera.

Nome: _____ Num. Matric.: _____

- B. Nel riquadro, scrivere una query SQL per calcolare la media dell'età degli autori che hanno scritto almeno un libro con altri (cioè che hanno scritto almeno un libro che hanno più di un autore). (2.5 punti).

```
SELECT AVG(ETA),  
FROM AUTORE  
WHERE CF IN  
  
    (SELECT P1.CF  
     FROM PUBBLICA P1, PUBBLICA P2  
     WHERE P1.LIBRO=P2.LIBRO  
     AND P1.CF <> P2.CF)
```

- C. Nel riquadro, scrivere una query SQL che restituisce il libro o i libri con il numero più alto di autori. (2.5 punti)

```
CREATE VIEW AUTORI-PER-LIBRO  
(LIBRO, NUM_AUTORI) AS  
  
    SELECT LIBRO, COUNT(*)  
    FROM PUBBLICA  
    GROUP BY LIBRO  
  
SELECT LIBRO  
FROM AUTORI-PER-LIBRO  
WHERE NUM_AUTORI=  
  
    (SELECT MAX(NUM_AUTORI)  
     FROM AUTORI-PER-LIBRO)
```

Esercizio 4: Transazioni (5 punti)

Data la transazione $S = r_1(x)w_2(x)r_3(x)w_1(u)w_3(v)r_3(y)r_2(y)w_3(u)w_4(t)w_3(t)$, sapendo che S è conflict-serializzabile:

1. Mostrare/Spiegare se S è (o non è) view-serializzabile;
2. se S view-serializzabile, mostrare uno schedule seriale T che è view-equivalente a S , mostrando perché S and T sono view-equivalenti.

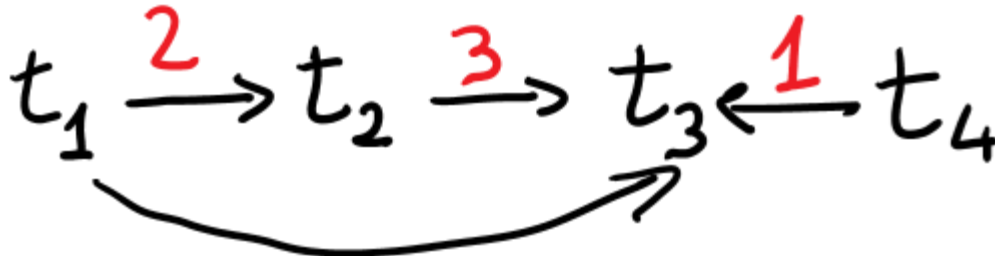
Parte 1

Siccome S è conflict-serializzabile, è anche view-serializzabile.

Parte 2

Se S è conflict-serializzabile, è possibile costruire un grafo dei conflitti di S che è aciclico. Il percorso nel grafo che tocca tutte le transazioni, fornisce un ordine delle transazioni. Scrivendo le operazioni nell'ordine delle transazioni e rispettando l'ordine all'interno delle transazioni, si ottiene uno schedule seriale che è conflict-equivalente e quindi anche view-equivalente.

Il seguente è il grafo dei conflitti:



Quindi, è possibile serializzare le transazioni nel seguente ordine: t_1, t_2, t_4, t_3 . Questo corrisponde allo schedule seriale $T = r_1(x) w_1(u) w_2(x) r_2(z) w_4(t) r_3(x) w_3(v) r_3(z) w_3(u) w_3(t)$, che è conflict-equivalente e quindi anche view-equivalente.

Si poteva ovviamente anche lavorare direttamente sulla view-equivalenza, guardando alla relazione leggi-da e alle scritture finali. In S , $r_3(x)$ legge da $w_2(x)$, mentre $r_1(x)$, $r_2(x)$ e $r_3(z)$ non leggono da scritture di transazioni in S . Le scritture finali sono $w_2(x)$, $w_3(t)$, $w_3(u)$ e $w_3(v)$. È facile verificare che lo schedule T ha una sola leggi-da come S , cioè $r_3(x)$ legge da $w_2(x)$, e le stesse scritture finali.

Nome: _____ Num. Matric.: _____

Esercizio 5: Quiz (5 punti)

Rispondere alle seguenti domande, sottolineando quale risposta è corretta (solo una è corretta).

Domanda 1 (1.5 Punti)

Data la relazione $R(A, B, C)$ con dipendenze funzionali $A \rightarrow BC$ e $B \rightarrow A$. Una delle seguenti affermazioni è vera. Quale?

- (1) AB è superchiave **ma non** è chiave; (Soluzione Corretta)
- (2) AB è **sia** chiave **che** superchiave;
- (3) AB **non** è **né** chiave **né** superchiave;
- (4) AB è chiave **ma non** è superchiave.

Domanda 2 (1 Punti)

Data un'istanza della relazione $R(A, B, C)$ dove l'unica chiave di R è $\{A, C\}$, (2) nessun attributo ammette valori nulli. Indicato con $|X|$ il numero di tuple di una relazione X , quale è vera tra le seguenti affermazioni relative al numero di tuple in $\pi_A(R)$?

- (1) È possibile sia $|\pi_A(R)| < |R|$ che $|\pi_A(R)| = |R|$. (Sol. Corretta)
- (2) È sempre il caso che $|\pi_A(R)| = |R|$.
- (3) Non è possibile che $|\pi_A(R)| = |R|$.
- (4) È sempre il caso che $|\pi_A(R)| = |\pi_{AC}(R)|$.

Domanda 3 (2.5 Punti)

Sia data una relazione $S(ABCD)$ con il set di dipendenze funzionali $\{C \rightarrow A, C \rightarrow D, D \rightarrow B\}$. Dopo aver trovato la chiave (che è una), dire quale affermazione (una) tra le seguenti è vera.

- (1) S è in forma normale di Boyce and Codd (BCNF) ma **non** è in terza forma normale (3NF).
- (2) S è **sia** in BCNF **che** in 3NF.
- (3) S **non** è in BCNF **ma** è in 3NF.
- (4) S **non** è **né** in BCNF **né** in 3NF. (Soluzione corretta con C unica chiave)