



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

# Corso di Laurea in Informatica

## Esame di Basi di Dati

Esame del 12 Febbraio 2021

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 alla gestione delle vigne e delle vendemmie di un produttore di vini. Disegnare il diagramma ER nel riquadro della pagina che segue.*

Di ogni vigna interessa il codice (identificatore), la regione in cui si trova, il tipo di uva che vi si coltiva, e i vari produttori che ne sono stati proprietari, con la data in cui lo sono diventati. Si tenga presente che è consentito al più un cambio di proprietà al giorno.

Delle vigne interessano poi anche le vendemmie che sono state effettuate nelle vigne stesse, al massimo una all'anno.

Di ogni vendemmia interessa la vigna in cui è stata eseguita, l'anno in cui è stata eseguita, la valutazione del risultato che ha avuto (un voto da 1 a 10), la persona che ne è stata responsabile (ma non sempre questa informazione è disponibile) e le bottiglie di vino che la vendemmia ha prodotto.

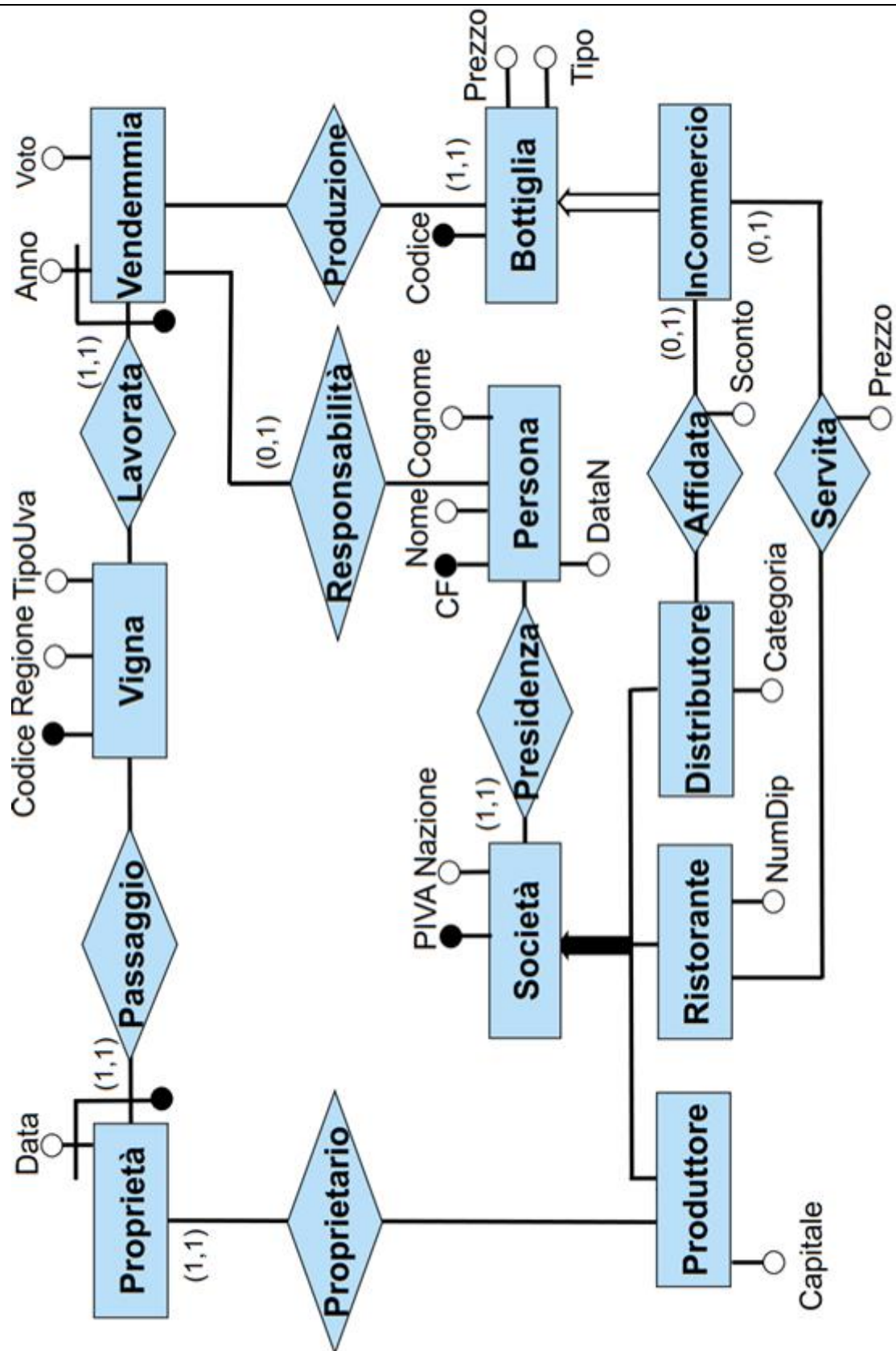
Ogni bottiglia ha un codice identificativo univoco ed è associato con un tipo, un prezzo standard. Non tutte le bottiglie vengono messe in commercio. Quando una bottiglia viene commercializzata, è di interesse sapere l'eventuale distributore che l'ha presa in carico per il suo commercio, con il relativo sconto applicato rispetto al prezzo standard (che può essere anche 0). Ogni bottiglia può essere venduta ad un ristorante: in tal caso, si vuole sapere quale è stato il ristorante e quale sia stato il prezzo di vendita.

Di ogni persona interessa il codice fiscale (identificativo), il nome, il cognome e la data di nascita.

Produttori, distributori e ristoranti sono le società di interesse per la nostra applicazione e di ogni società interessa la partita IVA (identificatore), la nazione in cui è registrata e la persona presidente. In aggiunta, dei produttori interessa il capitale sociale, dei distributori interessa la categoria e dei ristoranti interessa il numero di dipendenti.

Si noti che una società può essere una sola cosa tra produttore, distributore e ristorante.

## Soluzione Esercizio 1



Le cardinalità (1,n) omesse per leggibilità

Nome: \_\_\_\_\_ Num. Matric.: \_\_\_\_\_

## Esercizio 2: Progettazione Logica (7 punti)

A partire dallo ER dell'Esercizio 1, produrre uno schema relazionale del database nel riquadro sottostante. Indicare i vincoli di chiave e gli attributi che ammettono valori nulli. **Si consideri che, quando si accede al produttore, ristorante o distributore, si vuole sempre sapere la PIVA e la nazione. Tale operazione deve essere ottimizzata.** Illustrare come ristrutturare il diagramma ER per essere direttamente traducibile in uno schema relazionale.<sup>1</sup>

### Soluzione Esercizio 2 (Parziale)

Nella ristrutturazione:

1. L'entità *Società* viene rimossa e i suoi attributi duplicati per le entità figlie *Produttore*, *Ristorante* e *Distributore*;
2. La relazione *Presidenza* viene duplicata per ognuna delle entità figlie.

---

<sup>1</sup> 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 con le relazioni:

- **Programma** (Codice, NomeProgramma, Rete, Tipologia) che, ad ogni codice di programma televisivo, associa il nome del programma, la rete che lo trasmette e la tipologia.
- **Artista** (CProgramma, NomeArtista, compenso) che memorizza gli artisti che hanno partecipato ai diversi programmi, con i rispettivi compensi.

con chiave esterna Artisti.CProgramma→Programmi.Codice.

Si noti che ci possono essere due programmi con lo stesso nome ma codice diverso.

- A. Nel riquadro, scrivere una Query in Algebra Relazionale che restituisce i nomi dei programmi della Rete “Raiuno” in cui si esibiscono almeno due artisti. (2 punti).<sup>2</sup>

$V1 = \sigma_{Rete="Raiuno"} (Programma \bowtie_{Codice=CProgramma} Artista)$   
 $V1 = V2$   
 $\Pi_{V1.NomeProgramma}$   
 $(V1 \bowtie_{V1.Codice=V2.Codice \text{ AND } V1.NomeArtista \neq V2.NomeArtista} V2)$

---

<sup>2</sup> 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. Scrivere una query in Standard SQL che, per ogni codice programma, restituisce il nome del programma e la somma dei compensi per gli artisti del programma. (2.5 punti).

```
CREATE VIEW COMPENSO-PROGRAMMA  
(CODICE, COMPENSI) AS  
SELECT CPROGRAMMA, SUM(COMPENSO)  
FROM ARTISTA  
GROUP BY CPROGRAMMA;  
  
SELECT C.CODICE, NOMEPROGRAMMA, COMPENSI  
FROM COMPENSO-PROGRAMMA, PROGRAMMA  
WHERE COMPENSO-  
PROGRAMMA.CODICE=PROGRAMMA.CODICE
```

- C. Nel riquadro, scrivere una query in Standard SQL che restituisce la/e rete/i che trasmette più programmi di tipologia "Varietà" (2.5 punti)

```
CREATE VIEW PROGRAMMI-PER-RETE  
(RETE, NUM_PROGRAMMI) AS  
SELECT RETE, COUNT(*)  
FROM PROGRAMMA  
WHERE TIPOLOGIA="Varietà"  
GROUP BY CODICEBIBLIO;  
  
SELECT RETE  
FROM PROGRAMMI-PER-RETE  
WHERE NUM_PROGRAMMI =  
      (SELECT MAX(NUM_VISITATORI)  
       FROM VISITATORI-PER-BIBLIOTECA)
```

#### Esercizio 4: Normalizzazione (5 punti)

Sia data la relazione  $R(A,B,C,D,E,F)$  e l'insieme di dipendenze associato  $\{B \rightarrow A, E \rightarrow D, C \rightarrow B, CE \rightarrow F\}$ . Risolvere i seguenti punti:

- Trovare la/e chiave/i di  $R$ , motivando la risposta.
- Indicare quali dipendenze violano la BCNF, motivando la risposta
- Effettuare una decomposizione in BCNF senza perdita nel join, indicando le chiavi delle relazioni ottenute
- Indicare se la decomposizione ottenuta al punto c preserva le dipendenze. Motivare la risposta.

##### Punto a

Si considerano le seguenti chiusure

$B^+ = \{B, A\}$

$E^+ = \{E, D\}$

$C^+ = \{C, B, A\}$

$CE^+ = \{C, E, B, A, F, D\}$

Quindi,  $CE$  è una chiave della relazione.

##### Punto b

$B \rightarrow A$ ,  $E \rightarrow D$ ,  $C \rightarrow B$  violano la BCNF perché, rispettivamente,  $B$ ,  $E$  e  $C$  non sono superchiavi della relazione  $R$ .

##### Punto c

Usando  $B \rightarrow A$

$R_1(\underline{B}, A)$ , rimuovendo  $A$  da  $R$  che diventa  $R(B, C, D, E, F)$

Usando  $E \rightarrow D$

$R_2(\underline{E}, D)$ , rimuovendo  $D$  da  $R$  che diventa  $R(B, C, E, F)$

Usando  $C \rightarrow B$

$R_3(\underline{C}, B)$ , rimuovendo  $B$  da  $R$  che diventa  $R(C, E, F)$

La relazione  $R$  rimanente soddisfa  $CE \rightarrow F$ :  $R(\underline{C}, \underline{E}, F)$

##### Punto d

Sì, le dipendenze sono preservate poiché, per ogni dipendenza, esiste una relazione che contiene tutti gli attributi

Nome: \_\_\_\_\_ Num. Matric.: \_\_\_\_\_

### Esercizio 5: Quiz (5 punti)

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

#### Domanda 1 (1 Punto)

Data la query `SELECT * FROM S WHERE Z=4 ORDER BY X` sulla relazione  $S(X, Y, Z, W)$ . Quale dei seguenti indici in genere assicura le migliori performance in termini di velocità dell'esecuzione della query?

1. Indice B+Tree sulla coppia (X,Z)
2. Indice B+Tree su X
3. Indice Hash sulla coppia (X,Z)
4. Indice B+Tree sulla coppia (Z,X) <sup>3</sup>

#### Domanda 2 (2 Punti)

Date una qualsiasi istanza della relazioni  $R(\underline{A}, B, C)$  e  $S(\underline{D}, E, F)$  dove (1) le uniche chiavi di R e S sono rispettivamente  $\{A\}$  e  $\{D\}$ , (2) esiste chiave esterna:  $R.A \rightarrow S.D$ . Indicato con  $|X|$  il numero di tuple di una relazione X, quale è vera tra le seguenti affermazioni relative al numero di tuple in  $R \bowtie_{A=D} S$ ?

1. A volte  $0 \leq |R \bowtie_{A=D} S| < |R|$  ed altre volte  $|R \bowtie_{A=D} S| = |R|$
2.  $0 \leq |R \bowtie_{A=D} S| \leq |R| * |S|$
3. Sempre  $|R \bowtie_{A=D} S| = |R|$  <sup>4</sup>
4.  $0 \leq |R \bowtie_{A=D} S| \leq \max(|R|, |S|)$

---

<sup>3</sup> Questo indice permette di trovare la coppia  $(Z = 4, X = a)$  nell'indice dove  $a$  è il valore più piccolo presente per l'attributo X. Si itera sull'indice con coppie  $(Z=4, X=a')$  per valori crescenti di X fino a raggiungere la prima coppia con  $Z > 4$ . Vengono restituiti tutte le tuple associate con i valori di indice trovati, che sono già ordinate nel modo giusto

<sup>4</sup> Essendoci la chiave esterna specificata, ogni tupla di R è tale che l'attributo A è tra i valori dell'attributo S di una tupla di D. Quindi, ogni tupla di R rimane del prodotto cartesiano. Inoltre, poiché D è chiave primaria di S, ogni tupla di R è associata ad esattamente una tupla di S. Quindi, il numero di tuple del risultato è esattamente uguale alle tuple di R. Si noti che non è vero il contrario: ci sono tuple di S che non sono associate a tuple di R.

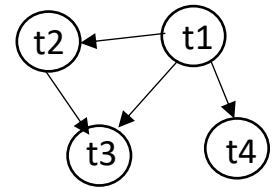


### Domanda 3 (2 Punti)

Dato il seguente schedule S, con grafo di conflitti in figura:

$W_1(A) R_2(A) R_2(B) W_2(D) R_3(C) R_1(C) W_3(B) R_4(A) W_3(C)$ .

Quale delle seguenti affermazioni è vera?



1. S è conflict-serializzabile ma **non** view-serializzabile
2. S **non** è conflict-serializzabile ma è view-serializzabile
3. S è sia conflict-serializzabile che view-serializzabile <sup>5</sup>
4. S **non** è né conflict-serializzabile né view-serializzabile

---

<sup>5</sup> Il grafo dei conflitti è aciclico, il che implica la conflict-serializzabilità, la quale implica a sua volta la view-serializzabilità