Domanda 3 (30%)
Si consideri una base di dati con le relazioni (entrambe con indice sulla chiave primaria)

R1(A,B,C), R2(D,E,F)

Eseguendo le interrogazioni seguenti, su Postgres, si rilevano le seguenti scelte per l'operatore di join:

1.	`select *` `from R1 join R2 on C=D`	Hash join

2.	`select *`
	`from R1 join R2 on C=D`
	`where B>=41 AND B<=45`
		Nested loop join
		con accesso diretto alla relazione interna

Motivare ciò, valutando, per ciascuna delle due interrogazioni, il costo di un piano di esecuzione con hash join e uno con nested loop join (e che, per l'operazione 2, includa la selezione), supponendo che

- le relazioni abbiano N_1=2.000.000 ed N_2=4.000.000 ennuple, (con fattore di blocco f_1=20 e f_2=40)
- l'attributo B in R1 abbia circa 200.000 valori diversi (compresi fra 1 e 200.000 e distribuiti uniformemente)
- entrambi gli indici abbiano p=4 livelli (radice e foglie incluse) e fattore di blocco massimo f_i=50
- l'operazione possa contare su un numero di pagine di buffer pari a circa q=500.

Rispondere riempiendo la tabella sottostante, indicando il costo in modo sia simbolico sia numerico.

	Hash join	Nested loop join con accesso diretto ...
1.		
2.		

Domanda 3 (30%)
Si consideri una base di dati con le relazioni (entrambe con indice sulla chiave primaria)

R1(A,B,C), R2(D,E,F)

Eseguendo le interrogazioni seguenti, su Postgres, si rilevano le seguenti scelte per l'operatore di join:

1.	`select *` `from R1 join R2 on C=D`	Hash join

2.	`select *`
	`from R1 join R2 on C=D`
	`where B>=41 AND B<=45`
		Nested loop join
		con accesso diretto alla relazione interna

Motivare ciò, valutando, per ciascuna delle due interrogazioni, il costo di un piano di esecuzione con hash join e uno con nested loop join (e che, per l'operazione 2, includa la selezione), supponendo che

- le relazioni abbiano N_1=2.000.000 ed N_2=4.000.000 ennuple, (con fattore di blocco f_1=20 e f_2=40)
- l'attributo B in R1 abbia circa 200.000 valori diversi (compresi fra 1 e 200.000 e distribuiti uniformemente)
- entrambi gli indici abbiano p=4 livelli (radice e foglie incluse) e fattore di blocco massimo f_i=50
- l'operazione possa contare su un numero di pagine di buffer pari a circa q=500.

Rispondere riempiendo la tabella sottostante, indicando il costo in modo sia simbolico sia numerico.

	Hash join	Nested loop join con accesso diretto ...
1.	Blocchi prima relazione N1/f1 = 100.000 Blocchi prima relazione N2/f2 = 100.000 Quadrato del n pagine = 500x500=250.000 Si può eseguire in due passate 3 (N1/f1 + N2/f2) = 600.000	Scansione di R1 e poi per ogni ennupla di R1 accesso diretto a R2 con l'indice su D Costo scansione di R1: N1/f1 Accesso diretto a R2: p+1-2 N1/f1 + N1 x (p+1-2) = ca 6.000.000
2.	Scansione di R1 con selezione e poi hash join della selezione con R2. L'hash join può quindi essere eseguito in UNA passata, perché la selezione di R1 entra nel buffer N1/f1 + N2/f2 = 200.000	Scansione di R1 con selezione e poi, per ogni ennupla della selezione, accesso diretto a R2 con l'indice su D N1/f1 + 50 x (p+1-2) = ca 100.000

Domanda 3 (30%)
Si consideri una base di dati con le relazioni (entrambe con indice sulla chiave primaria)

R1(A,B,C), R2(D,E,F)

Eseguendo le interrogazioni seguenti, su Postgres, si rilevano le seguenti scelte per l'operatore di join:

1.	`select *` `from R1 join R2 on C=D`	Hash join

2.	`select *`
	`from R1 join R2 on C=D`
	`where B>=41 AND B<=45`
		Nested loop join
		con accesso diretto alla relazione interna

Motivare ciò, valutando, per ciascuna delle due interrogazioni, il costo di un piano di esecuzione con hash join e uno con nested loop join (e che, per l'operazione 2, includa la selezione), supponendo che

- le relazioni abbiano N_1=2.000.000 ed N_2=4.000.000 ennuple, (con fattore di blocco f_1=20 e f_2=40)
- l'attributo B in R1 abbia circa 200.000 valori diversi (compresi fra 1 e 200.000 e distribuiti uniformemente)
- entrambi gli indici abbiano p=4 livelli (radice e foglie incluse) e fattore di blocco massimo f_i=50
- l'operazione possa contare su un numero di pagine di buffer pari a circa q=500.

Indicare come cambiano i costi, per l'operazione 2, se sull'attributo B è definito un indice.

	Hash join	Nested loop join con accesso diretto ...
2bis.		

Domanda 3 (30%)
Si consideri una base di dati con le relazioni (entrambe con indice sulla chiave primaria)

R1(A,B,C), R2(D,E,F)

Eseguendo le interrogazioni seguenti, su Postgres, si rilevano le seguenti scelte per l'operatore di join:

1.	`select *` `from R1 join R2 on C=D`	Hash join

2.	`select *`
	`from R1 join R2 on C=D`
	`where B>=41 AND B<=45`
		Nested loop join
		con accesso diretto alla relazione interna

Motivare ciò, valutando, per ciascuna delle due interrogazioni, il costo di un piano di esecuzione con hash join e uno con nested loop join (e che, per l'operazione 2, includa la selezione), supponendo che

- le relazioni abbiano N_1=2.000.000 ed N_2=4.000.000 ennuple, (con fattore di blocco f_1=20 e f_2=40)
- l'attributo B in R1 abbia circa 200.000 valori diversi (compresi fra 1 e 200.000 e distribuiti uniformemente)
- entrambi gli indici abbiano p=4 livelli (radice e foglie incluse) e fattore di blocco massimo f_i=50
- l'operazione possa contare su un numero di pagine di buffer pari a circa q=500.

Indicare come cambiano i costi, per l'operazione 2, se sull'attributo B è definito un indice.

	Hash join	Nested loop join con accesso diretto ...
2bis.	Selezione su R1 con l'indice su B e hash-join del risultato con R2 p + 50 + (N2/f2) = 100.000	Selezione su R1 con l'indice su B e per ogni ennupla della selezione, accesso diretto a R2 con l'indice su D p + 50 + 50 x (p+1-1) = ca 250