

**Esercizio 2**

E' data la query:

```
SELECT Nome, Voto
FROM STUDENTI, ESAMI
WHERE sessione = 'Invernale'
AND anno_iscrizione = '2004'
AND STUDENTI.mat = ESAMI.mat
```

sulle relazioni:

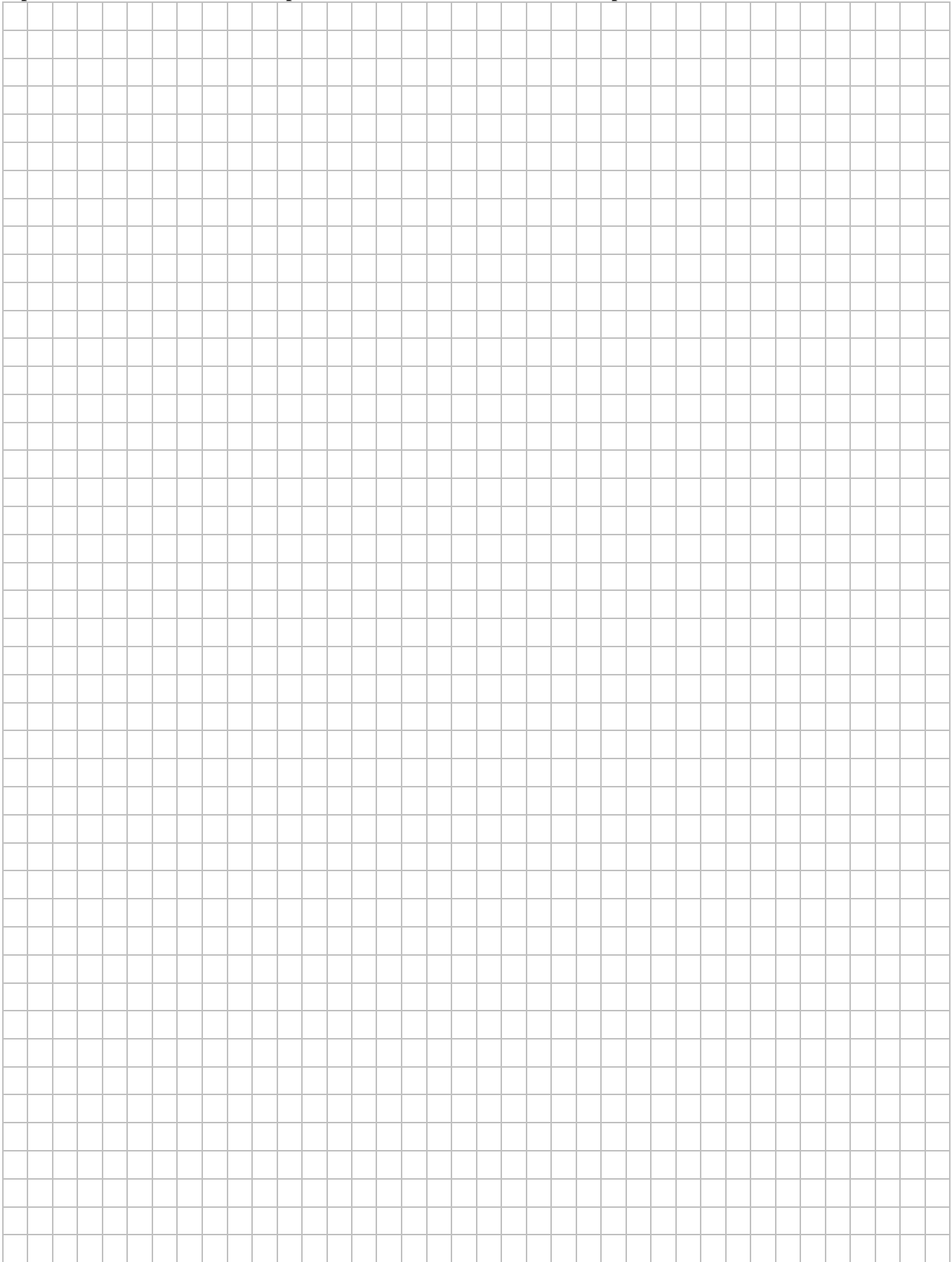
STUDENTI ( mat, nome, cognome, anno\_iscrizione )  
 ESAMI ( mat, codCorso, voto, data, sessione )

Si determini il miglior piano d'accesso per la risoluzione della query (per l'esecuzione del join si ipotizzi l'utilizzo degli algoritmi nested-loop e sort-merge ( $Z=3$ )), tenendo conto dei seguenti dati:

$NP_{STU} = 150$              $NT_{STU} = 35.000$              $NK_{anno\_iscrizione} = 20$   
 $NP_{ESA} = 600$              $NT_{ESA} = 800.000$              $NK_{sessione} = 4$   
 Su STUDENTI.mat è costruito un indice clustered con  $NL_{mat} = 20$  e  $h = 3$

**Svolgimento**

A large grid for writing the solution, consisting of approximately 30 columns and 30 rows of small squares.



**Esercizio 3**

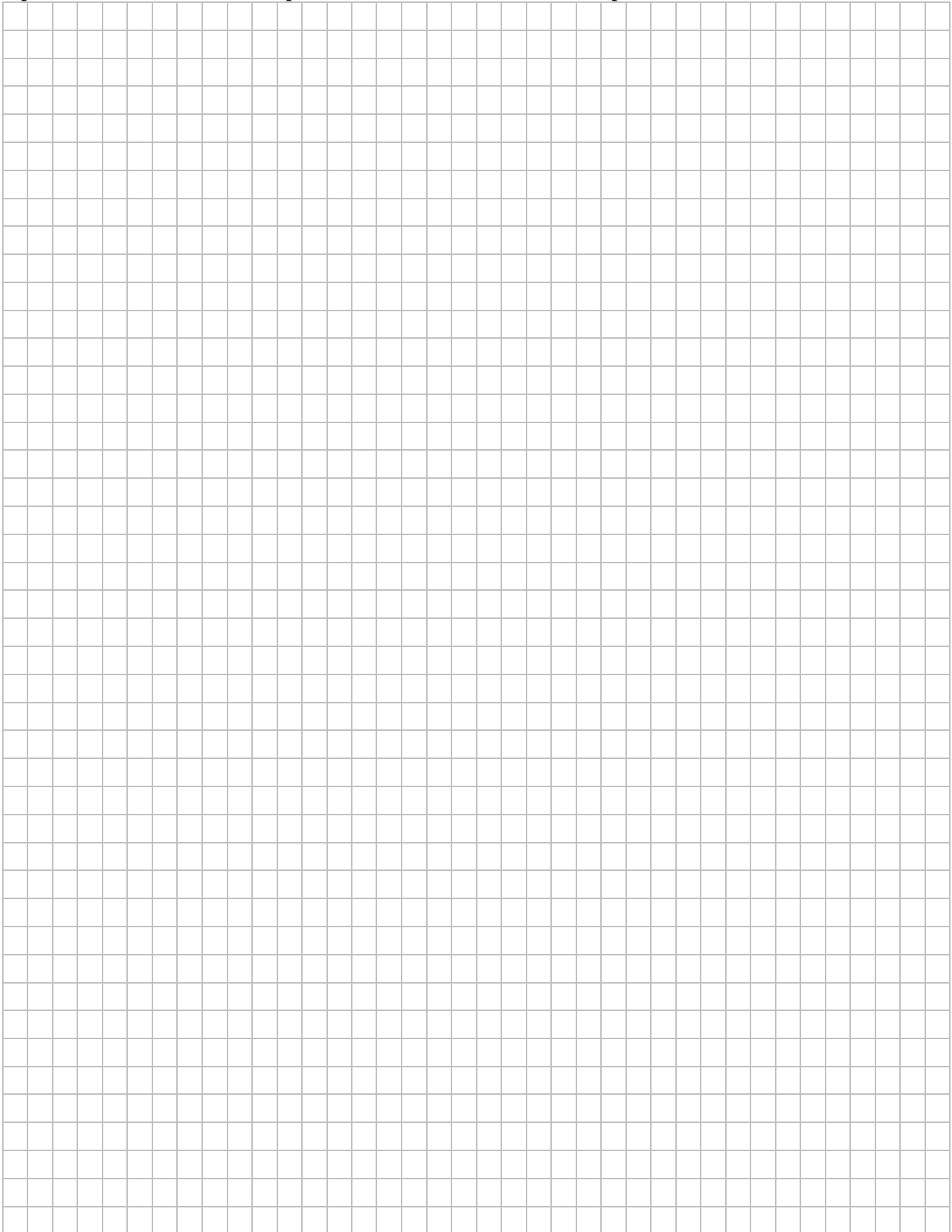
Dato il seguente schema relazionale:

SOCIO (tesserino, nome, indirizzo, città, età, telefono, data Iscrizione, corsiFrequentati)  
CORSO(codCorso, nome, descrizione, dataInizio)  
ISCRIZIONE (tesserino: SOCIO, codCorso: CORSO)

- 1. Scrivere la **stored procedure** che visualizza i corsi (nome, descrizione) che hanno più di 6 iscritti e non sono ancora iniziati.
- 2. Creare un **trigger** che tiene aggiornato il dato derivato “corsiFrequentati”. A fronte dell’inserimento di un nuovo iscritto incrementa il campo corsiFrequentati del relativo iscritto.

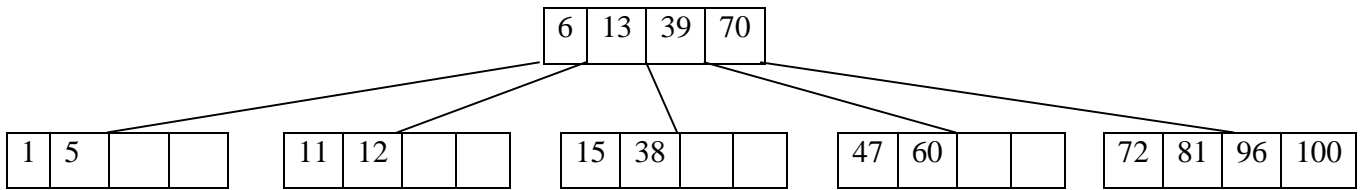
**Svolgimento**

The form consists of a large rectangular area filled with a fine grid of small squares, intended for the student to write their solution to the exercise. The grid is approximately 30 columns wide and 35 rows high.



**Esercizio 4**

Dato il seguente B-tree di ordine  $g=2$  mostrarne l'evoluzione a seguito della cancellazione della chiave 60.



**Svolgimento**

A large grid area for the student's solution, consisting of approximately 30 columns and 25 rows of small squares.

