

```

2 filename="./address-book-database.csv"
3
4 # Controlla se il file esiste
5 if [[ ! -f $filename ]]; then
6     echo "Errore: file $filename non trovato"
7     exit 1
8 fi
9
10 # Controlla se l'argomento è stato fornito
11 if [[ -z $1 ]]; then
12     echo "Errore: nessun argomento fornito"
13     exit 1
14 fi
15
16 # Comandi principali
17 if [[ $1 == "view" ]]; then
18     # Visualizza il contenuto del file in modo formattato e ordinato
19     head -n 1 "$filename" | column -s "," -t
20     tail -n +2 "$filename" | column -s "," -t | sort -k 4 -t ","
21 elif [[ $1 == "insert" ]]; then
22     # Inserisce un nuovo record nel file
23     caso1=$(head -1 "$filename" | cut -d "," -f 1)
24     echo -n "${caso1}: "
25     read var
26     inserimento="$var"
27     for (( i = 2; i <= 6; i++ )); do
28         caso1=$(head -1 "$filename" | cut -f $i -d ",")
29         echo -n "${caso1}:" "
30         read var
31         inserimento="$inserimento,$var"
32         if (( $i == 4 )); then
33             email=$(echo $inserimento | cut -f 4 -d ",")
34             match=$(grep -w "$email" "$filename")
35             if [[ ! -z $match ]]; then
36                 echo "Errore: Mail già presente"
37                 exit 1
38             fi
39         fi
40         if (( $i == 6 )); then
41             echo "$inserimento" >> "$filename"
42             echo "Added"
43         fi
44     done
45 elif [[ $1 == "search" ]]; then
46     # Cerca un record nel file
47     if [[ -z $2 ]]; then
48         echo "Errore: specificare il termine da cercare"
49         exit 1
50     fi
51     IFS=$'\n'
52     cerco=$2
53     trovato=0
54     for line in $(tail -n +2 "$filename"); do
55         linea_verifica=$(echo "$line" | grep -w "$cerco")
56         if [[ ! -z $linea_verifica ]]; then
57             trovato=1
58             IFS=","
59             for (( i = 1; i <= 6; i++ )); do
60                 indice=$(head -1 "$filename" | cut -f $i -d ",")
61                 campo=$(echo "$linea_verifica" | cut -d "," -f $i)
62                 echo "${indice}": $campo
63             done
64             echo -e
65         fi
66     done
67     if (( $trovato == 0 )); then
68         echo "Not Found"
69     fi
70 elif [[ $1 == "delete" ]]; then

```

```

71     # Elimina un record dal file
72     if [[ -z $2 ]]; then
73         echo "Errore: specificare il termine da eliminare"
74         exit 1
75     fi
76     elimina=$2
77     verifica=$(grep -w -c "$elimina" "$filename")
78     if (( verifica == 0 )); then
79         echo "Cannot find any record"
80         exit 1
81     fi
82     sed -i "/$elimina/d" "$filename"
83     echo "Deleted"
84 else
85     echo "Errore: argomento fornito non valido"
86     exit 1
87 fi
88
89 # Ripristina l'IFS originale
90 IFS=$' \t\n'

```

5 -> $\neg f$ → controllo se il file ESISTE
 ↳ $\neg f$ → è un file ? 1) CONTROLLO FILE (ESISTE)

$\neg \neg f \rightarrow$ non è un file ?

11 - 14 → -z → STRINGS MULT
 ↳ bruto force
 FORKING 2) BRUTALISMO (FORKING)

17 -> view → if \neg

→ \neg right
 HEAD \neg right \rightarrow N RIGHT }
 { N & N RIGHT
 (\rightarrow C BYTE
 | 0 min right + A(L)

TAIL -N + 2 | COLUMN -S "", "-T"

↳ GRANIT COSTA

DI HEAD H^o
PREIXE LE ULTIME

COLUMN → DIVIDE PER "", &
METTE SPATI

1, 2, 3, 4, 5 REC

1 { 2 } 3 { 4 } 5
1 { 2 } 3 { 4 } 5

COLUMN → SORT

-S / SEPARATORE
DIVISIONE
-T SEPARATORE
DATA > FATTI
ORDINE
PER COLONNE
1 2 3
4 5 6
7 8
2 5
3 6

-T CHIAVE
DI ORDINAZIONE
-T delimitatore
-V eliminare
righe duplicate

SORT -T & -T ""

↳ ordinare la colonna e delimitare con

in ordine
numerico
+ 02 + 0

3) LA VIEW

↳ SELECT la prima RIGA (HEAD) e ordina tutte le colonne

↳ LE TABELLE → TABL (SELECT LA PRIMA)

ORDINE TEATRICO ED OPERA CAZARE

PRERTE "STRICTI"

SI VOLGE ITERAZIONE COMPO

→ bis solto SI PUÒ STAMPARE IL PRIMO
E PROSEGUIRE CON SECONDO

++ IC(HO)

bREVETTO UND NUOVA

RICHI

↳ ECHO - N ND

- CASO 1 → HEAD(-1) → PRIMO IL PRIMO REG

PRIMO
COMPO

↳ STAMPA IL CASO
↳ h ot a CUT -d, " -f
↳ DECIMOTRORI

↳ L'ESCI C'INPUT PER REG

DO OGNI ITERAZIONE

ITERAZIONE → è come un array → oggi VOLGO PIANO PIANO

↳ RISULTATO HA PRIMO VEDI LETTERA

Se ISOLANDO IL PRIMO VOLGO PER POTER INIZIALIZZARE
IL INSERIMENTO, COME POSSE UN BROSSO

ITERAZIONI SUGLI ELEMENTI

→ CASO -1 → HEAD -1 RISULTATO | CUT -1 f_i -d, "

CICLO

f_i
↳ COMBINAR IL
COMPO

1 | 2 | 3 | 4 | 5

i_1 i_2 i_3 i_4 i_5

→ oggi ho visto little SU INSERIMENTO → inserimento = " {i_1, v_k}"

Dopo aver inizializzato INSERIMENTO, DEVO
RIEMPIRE ITERAZIONE

i=4

children verificando se il campo non sia già presente

↳ LO SALVIAMO PER VERIFICARLO → EMAIL

EMAIL = ECHO + INSERIMENTO

↳ STAMPARE INSERIMENTO → SISTEMA DOGLI PRENDIAMO IL CAMPO

↳ CUT -d ";" -f n

VDR DI APPROVATO → RICEVERE EMAIL

↳ MATCH → grep -w

{
- i giorni MARZO/APRILE/MAGGIO
- M MOSTRA le righe in cui sono inseriti
- w eseguirà solo parole intere
- x eseguirà anche sotto parole

-w VS
-x
di eseguire
di PAROLE INTERE
WRITING ONCE
PIÙ DI PAROLE

↳ un match stampa catturando l'email utente

↳ GREP -w "if EMAIL" "FILE NAME" → in effetti grep -w è l'item cost in questo caso

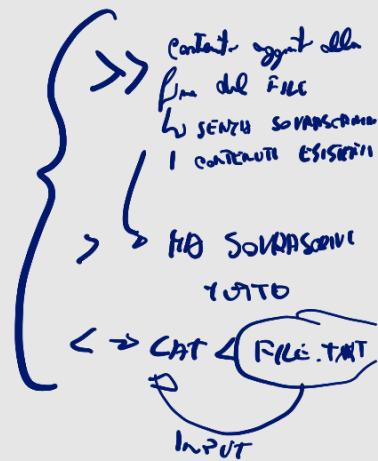
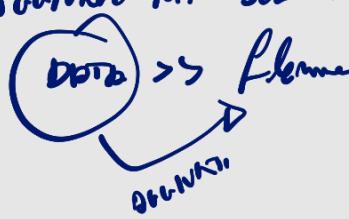
↳ se niente trova allora if [! -z match] è vero → quindi [! -z ?] è falso → esce con EXIT

CONTROLLO SU ITERATORI

varianti: file → EMAIL NON presente → avviene tutto sul file

FILTRO ITERATORI

↳ >>



-> vuoto

SEARCH → SEARCH OK

se scrivendo SEARCH SENZA METTI → NON HA SENTITO → CONTROLLO SULL'INPUT → if [! -z ?] →

·IFS → OSCURO → UN WORD C'È DENTRO

·TROVATO CON VARI BB CERCARE → ESEMPIO ITALIANO PREMI

·TROVATO → VDR DI APPROVATO → O → SE È 1 ALLORA È TROVATO
·ITERA SU TUTTE LE LINEE (ESCLUSO IL PRIMO) → (TAIL -N (+2)) → SETTING

↳ & LINEA → LINEA | awk -w "cerco" → \$2

↳ SE NON È RUOTO → [! -z ?] → TROVATO = 1

SE LO TROVO STAMPO QUINDI → ESEMPI

→ PREMERE IL ULTIMO

BHO → FUNZIONE
E BOSS

INDEX:
{ (HEAD → "FIND NAME") CUT -f \$1 -d ";" } INDEX
1:
2:
3:
4:
5:
| MAPP

→ IFS = ";"
1, 2, 3, 4, 5
|
1: → PEC

2:

→ PRENDO LA PRIMA RIGA
(HEAD -1)

Lo stampo il valore -f di definito da "1.."
 CAMPO: AVERO SPazio LO LINEA DAL MODO CHE SHVO
 LO RIGA NON HO IL BIGLIETTO:
 DI RICREARE I CAMPi → ITENDO SUL
 RIUSCITO

→ stampo la linea
 ↳ echo {line | cut -d ":" -f i} → primo campo grande

Sono tutto → echo {indirizzo}: campo

ECHO -e → shift carattere spaziale

→ SE NON TROVO NIENTE, STO CERCO

3:

IFS = \$' \n ' PER IL FILE
 ↳ TALK ~ +2 FILE → USATO
 come "DELIMITATORI" → LO SPAZIO
 E TROPPO OLTRE RIGA SINGOLARE

EITERAZIONE E SCOSTRUZIONE

CONTROLLA OCCORRENZA

DELETE → CLASSICO CONTROCCO → -z (NUOTO)

→ VERIFICA CHE CI SIA QUALcosa DA ELIMINARE → grep -w c

↳ == > NON TROVATO

ELSE SET -i → JED → STREG EDITOR → MANPOD È TRASFORMATO TESTO
 ↳ agire sul file è non su QUELLO TEMPORANEO

SED -i "/\$ELIMINA/d" "\$file"
 ↳ SETTA
 ↳ ERGE LE LINEE NEL FILE

ERRORE COMMESSI

- 1) CONTROLLO SUL FILE → -f
- 2) NON E' IT
 QUANDO SI CREA UNO VOTO L'ERRONEA
- 3) IN INSERT → IF [(-2 MATCH?) → SE È VUOTO PERCHÉ DOVREVA FERMARSI
 ↳ COSÌ MI METTI INIFICIO DI SEARCH

CODICE DIVERSO → SO GET HUB

SE È (controlla) UN FILE ALLORA ENTRA NEL BLOCCO

VIEW CLOSEST → CANCELLA -T -S ";" | head -m
 ↳ -T → secca branch
 ↳ -S ";" → DELIMITATORI

STAMPA IN OGNI

TUTTI "+(+2)"
 ↳ CANCELLA -T -S ";"
 ↳ SART (+1) IN
 ↳ SECONDO "

CONTROLLA → CONTROLLA SU NUMERO DI ARREDAMENTI

- PDSBTI
- ↳ controllo sulle occorrenze → GREP -w $\langle c \rangle$ numero OCCORRENZE
ES: → quando c'è qualcosa sicuramente \rightarrow le celle non iterabili su
 - ↳ each ha per (i) una FOR ITEM TUTTE LE CELLE \rightarrow le stringhe
 - INPUT → PRIMA RIGA → SOTTO CUD -D " " -f DIRETTORE IN TUTTO IL FILE
 - INSERT → perché ~~USO~~ ~~ROTA~~ con "INSERIRE"
 $\langle new_entry = " " \rangle \rightarrow$ che non SERVIRÀ UN COSE
 - \rightarrow TUTTE quelle quando si usano $\langle c \rangle$
 - \rightarrow CLASSICO controllo se i=4 \Rightarrow grey - e - h = 1
 - DEBUTTO \rightarrow CLASSICO CON RULLO SUBITO IHPOT E
SUL PENSARE DI CLIMINARE GLI INDICI
 - \rightarrow se non trovi nulla \rightarrow sub -i "/..":d ecc