Matricola	
Cognome	
Nome	

Basi di Dati (Modulo Teoria) Prova scritta

Durata: 2 ore e 15 minuti

Avvertenze: è severamente vietato consultare libri e appunti.
DOMANDE PRELIMINARI (è necessario rispondere in modo sufficiente alle seguenti tre domande per poter superare la prova scritta con esito positivo; in caso di mancata o errata risposta a queste domande il resto del compito non verrà corretto)

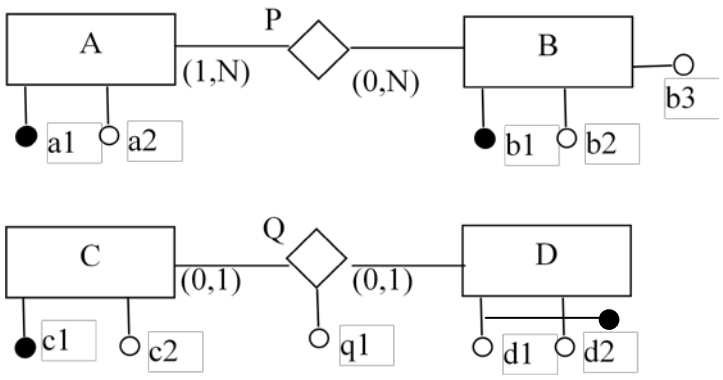
a) Si illustri il costrutto di entità del modello Entità-Relazioni

Il costrutto di entità del modello Entità-Relazioni rappresenta una classe di oggetti (fatti, persone, cose) della applicazione di interesse con proprietà comuni e con esistenza "autonoma".

Esempio: persona, città, studente, corso

Rappresentazione grafica

b) Dato il seguente schema concettuale nel modello ER, si produca la sua traduzione nel modello relazionale.



A(<u>a1</u>,a2)
B(<u>b1</u>,b2,b3)
P(<u>a1</u>,<u>b1</u>)

C(<u>c1</u>,c2,)
D(<u>d1</u>,<u>d2</u>)
Q(<u>c1</u>,<u>d1</u>,<u>d2</u>,q1)

c) Date le due seguenti relazioni: R1(<u>A</u>, B, C) e R2(<u>D</u>, E, F) (tutti gli attributi sono di tipo numerico) scrivere;

c.1) un'espressione in algebra relazionale che restituisca i valori distinti contenuti nell'attributo F di R2;

c.2) un'espressione ottimizzata dell'algebra relazionale che contenga un join naturale e una selezione su R2, e produca come risultato le tuple t di R2 tali che t[E]>=5 e tali che esiste una tupla t' di R1 dove t[D]=t'[C] (non sono ammesse altre selezioni oltre a quella su R2).

- C.1

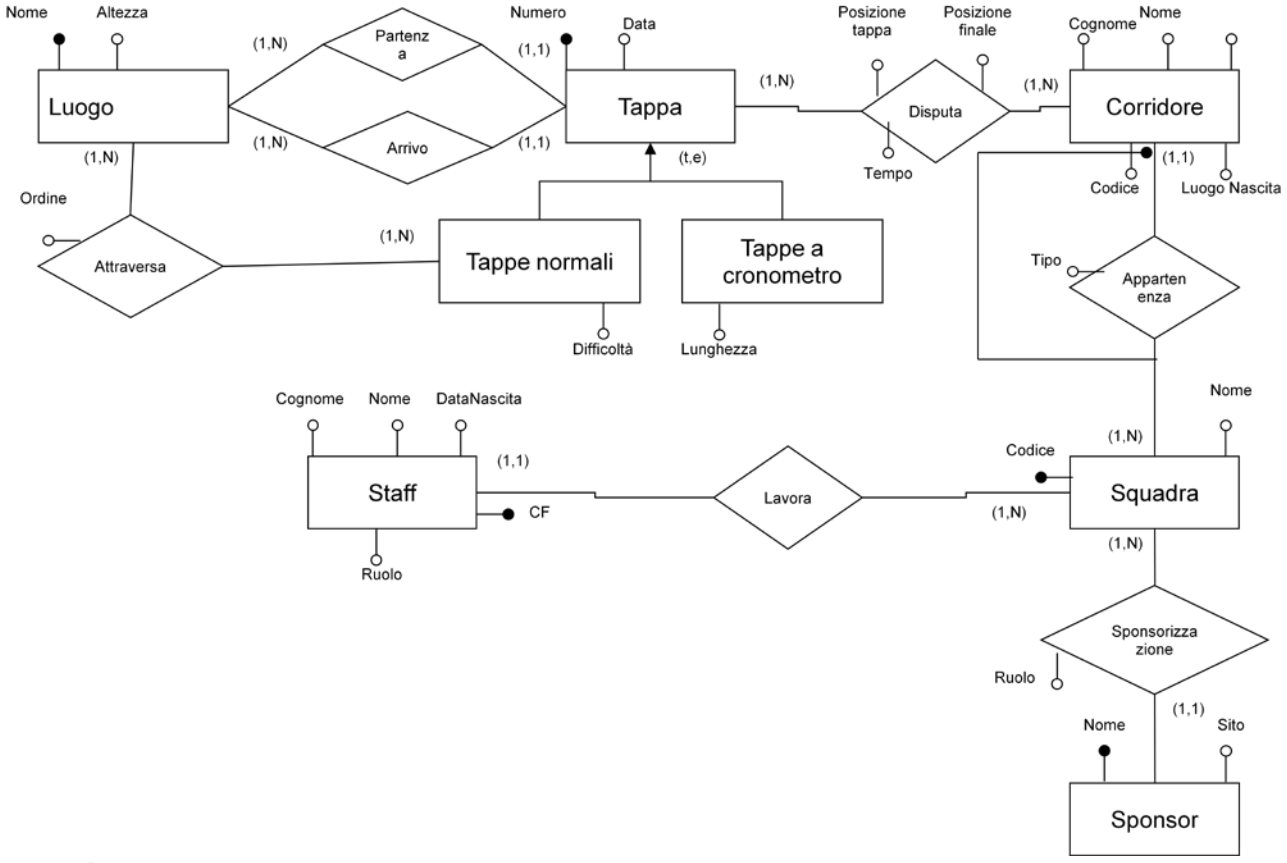
  $\Pi_F (R2)$

- C.2

  $(\sigma_{E>5} (R2)) \quad (\Pi_D (\rho_{D \leftarrow C} (R1)))$

Esercizio 1
Progettare la base di dati relativa alla situazione descritta nel seguito. Progettare lo schema concettuale utilizzando il modello entità-relazione e lo schema relazionale della base di dati (indicare esplicitamente per ogni relazione dello schema relazionale: le chiavi primarie, gli attributi che possono contenere valori nulli e i vincoli di integrità referenziale). Non aggiungere attributi non esplicitamente indicati nel testo.

Si richiede di progettare la base di dati di un'applicazione relativa all'ultima edizione del Giro d'Italia. Di ogni tappa, identificata da un numero d'ordine, interessa la data in cui è stata disputata, il luogo di partenza, ed il luogo di arrivo. Esistono due tipologie di tappe: le tappe a cronometro e le tappe normali. Delle tappe a cronometro interessa la lunghezza. Delle tappe normali interessa il grado di difficoltà, e gli eventuali luoghi intermedi attraversati dalla tappa, con l'ordine di attraversamento. Una stessa tappa non può passare due volte per lo stesso luogo. Di ogni corridore interessa la squadra di appartenenza (una ed una sola), con il tipo di appartenenza (ad es., capitano, gregario, ecc.), il codice (unico nell'ambito della squadra), il nome, il cognome, la data di nascita ed il luogo di nascita. Interessano inoltre le tappe portate a termine dal corridore, con la posizione nella tappa e la posizione complessiva al termine della tappa. Nel caso di tappe a cronometro interessa anche il tempo impiegato dal corridore. Di ogni squadra interessa il codice (identificativo), il nome, gli sponsor e la composizione dello staff tecnico (direttore sportivo, massaggiatori, meccanici, …). Ogni squadra può essere sponsorizzata da più di uno sponsor; per ogni sponsor si conoscono il nome (identificativo), il sito web e il ruolo (sponsor principale o secondario). Si fa presente che ad una squadra deve appartenere almeno un corridore. I membri dello staff tecnico sono caratterizzati dal codice fiscale, nome e data di nascita, il ruolo. Ogni membro appartiene allo staff di una sola squadra. Di ogni luogo rilevante per la base di dati interessa il nome (identificativo) e l'altezza sul livello del mare.

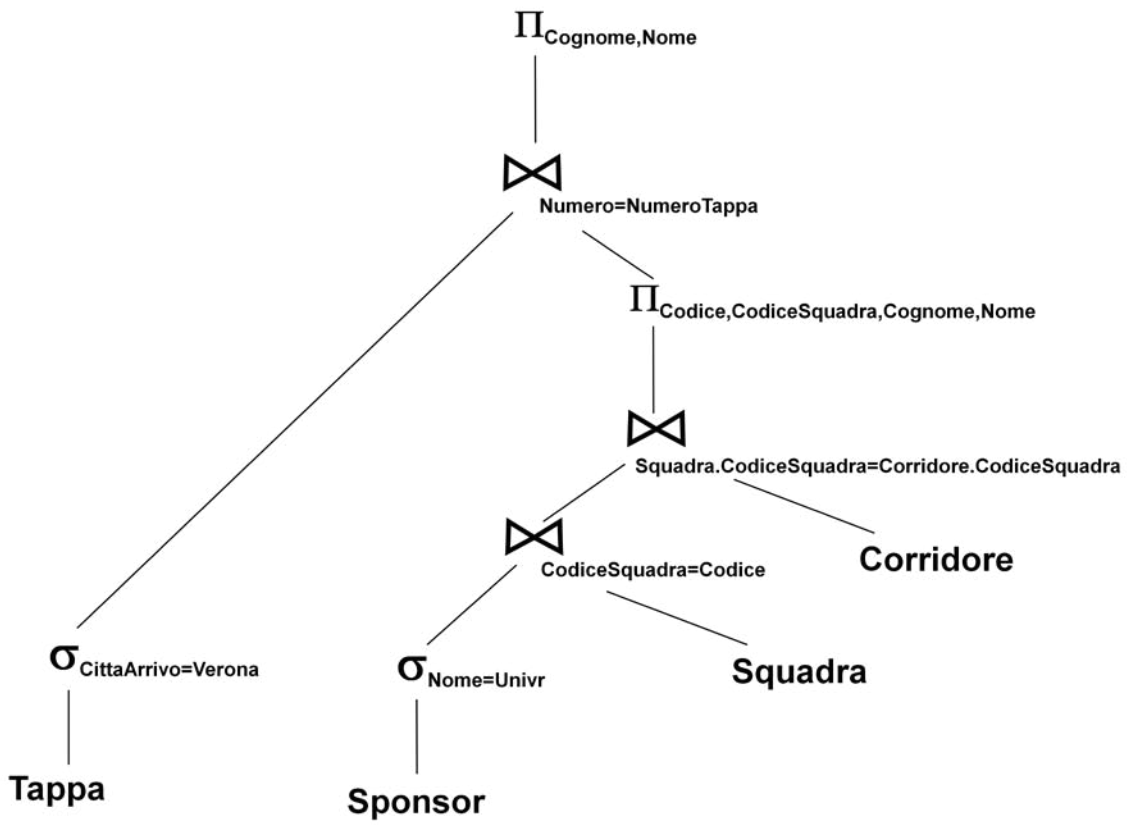


Progetto logico
Tappa (Numero, Data, CittàArrivo, CittàPartenza, Difficoltà, Lunghezza, Tipo)
Luogo (Nome,Altezza)
Attraversamento (NumeroTappa, NomeLuogo, Ordine)
Squadra (Codice, Nome)
Corridore (Codice, CodiceSquadra, Nome, Cognome, DataNascita, LuogoNascita)
Disputa (CodiceCorridore, CodiceSquadra, NumeroTappa, PosizioneTappa, PosizioneFinale, Tempo)
Sponsor (Nome, SitoWeb, CodiceSquadra, Ruolo)
Staff (CF, Nome, Cognome, DataNascita, Ruolo, CodiceSquadra)

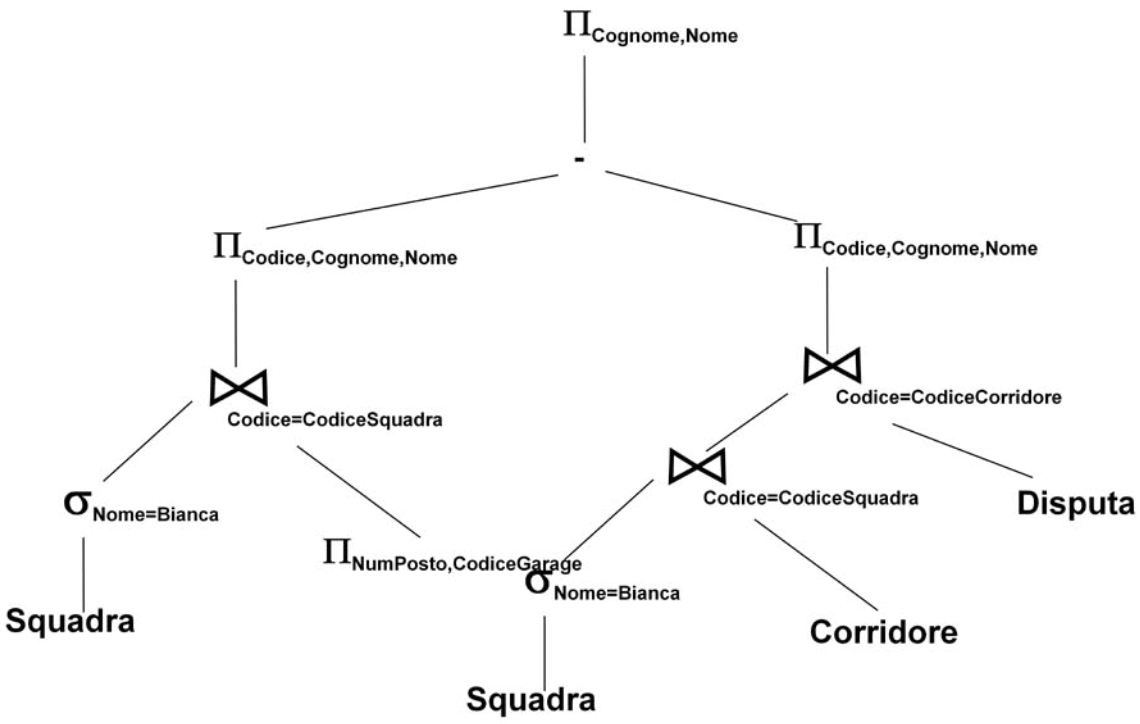
Esercizio 2

Dato lo schema relazionale dell'esercizio 1, esprimere in algebra relazionale ottimizzata le seguenti interrogazioni:

2.a) *Trovare il nome e cognome dei corridori della squadra sponsorizzata da UniVr che hanno disputato la tappa che arriva a Verona.*



2.b) *Trovare il nome e cognome dei corridori della squadra "Bianca" che non hanno disputato alcuna tappa.*



Esercizio 3

Dato il seguente schema relazionale, (chiavi primarie sottolineate), in cui il campo Categoria della tabella POSTO_AUTO può assumere i valori “Normale”, “Convenzionato” o “Mezzo_Eccezionale”:

GARAGE (Cod Garage, Indirizzo, Città, CodiceFiscaleProprietario)
POSTO_AUTO(NumPosto, CodiceGarage, Categoria)
CATEGORIA(Codice, Altezza, Larghezza, Tariffa_oraria)
AUTO (Targa, AnnoImm, Marca, Modello, CodFiscaleProprietario)
OCCUPAZIONE(PostoAuto, Garage, TargaAuto, DataInizio, OraInizio, NumeroOre, CostoTotale)

Vincoli di integrità: POSTO_AUTO.CodiceGarage → GARAGE,
OCCUPAZIONE.PostoAuto → POSTO_AUTO,
OCCUPAZIONE.Garage → GARAGE,
OCCUPAZIONE.TargaAuto → AUTO

Formulare in SQL le seguenti interrogazioni (definire viste solo dove è necessario):

3.a *Fornire informazioni sui posto auto (di qualunque garage) che sono stati occupati sempre dalla stessa automobile nel mese di gennaio 2006.*

```
SELECT PostoAuto, Garage
FROM Occupazione
WHERE DataInizio BETWEEN 01/01/06 AND 31/01/06
GROUP BY PostoAuto, Garage
HAVING COUNT(DISTINCT(TargaAuto))=1
```

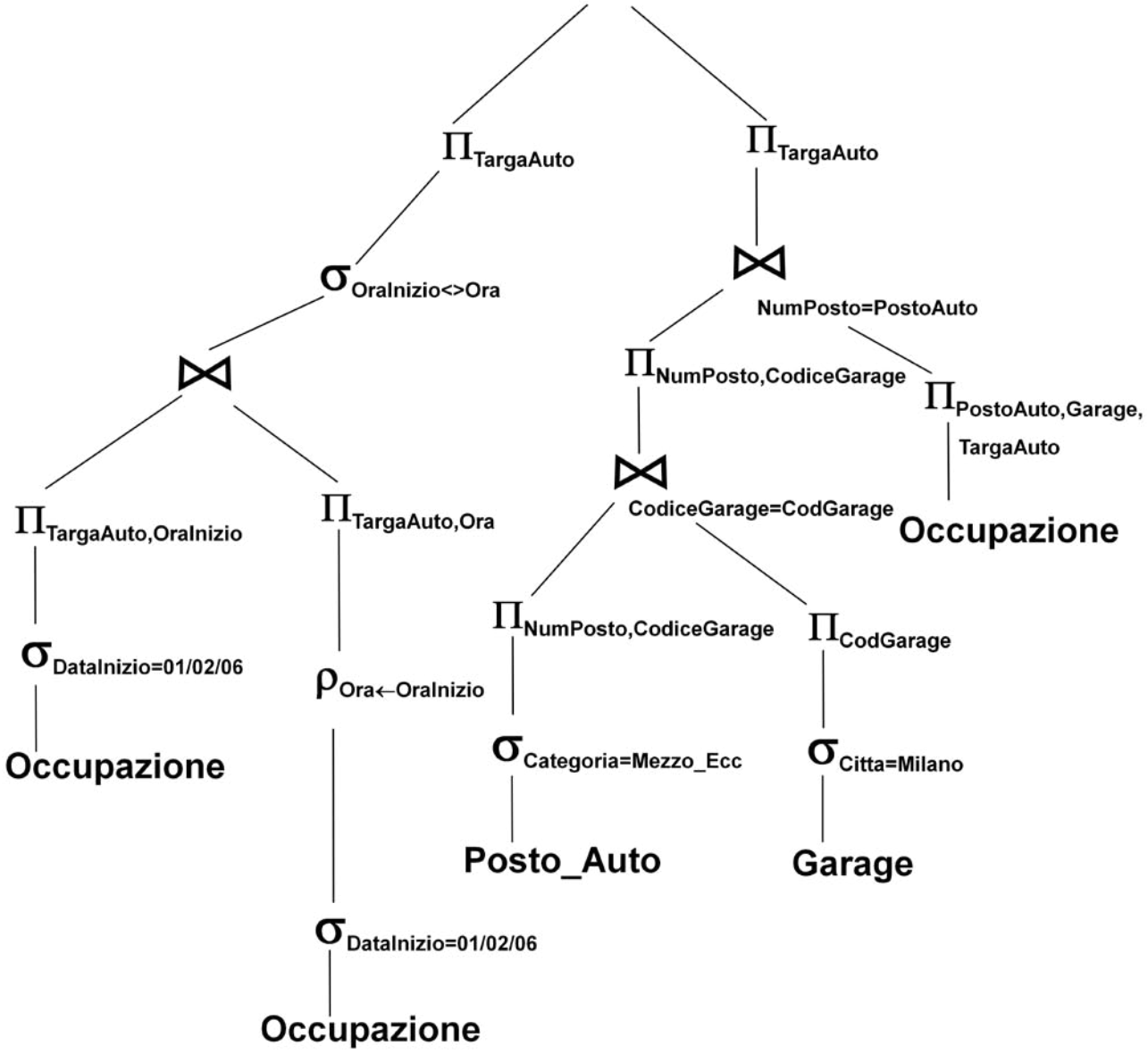
3.b *Determinare il posto auto non convenzionato e il relativo garage che nel mese di novembre 2006 abbia fruttato il massimo ricavo.*

```
CREATE VIEW RicavoPostoNovembre (Posto, Garage, Ricavo) AS
SELECT PostoAuto, Garage, SUM(CostoTotale)
FROM Occupazione, Posto_Auto
WHERE Occupazione.PostoAuto = Posto_Auto.NumPosto
AND Occupazione.Garage = Posto_Auto.CodiceGarage
AND (Categoria=“Normale” OR Categoria=“Mezzo_Eccezionale”)
AND DataInizio BETWEEN 01/11/06 AND 30/11/06
GROUP BY PostoAuto, Garage

SELECT PostoAuto, Garage
FROM RicavoPostoNovembre
WHERE  Ricavo = (SELECT MAX(Ricavo) FROM RicavoPostoNovembre)
```

Formulare in algebra relazionale ottimizzata la seguente interrogazione:

3.c *Trovare le automobili che hanno effettuato almeno 2 posteggi il giorno 1/2/06 ma che non siano mai state parcheggiate in posti di categoria"Mezzo_Eccezionale" a Milano.*



Esercizio 4

Lo studente illustri il concetto di View-Serializzabilità.

Date le seguenti definizioni:

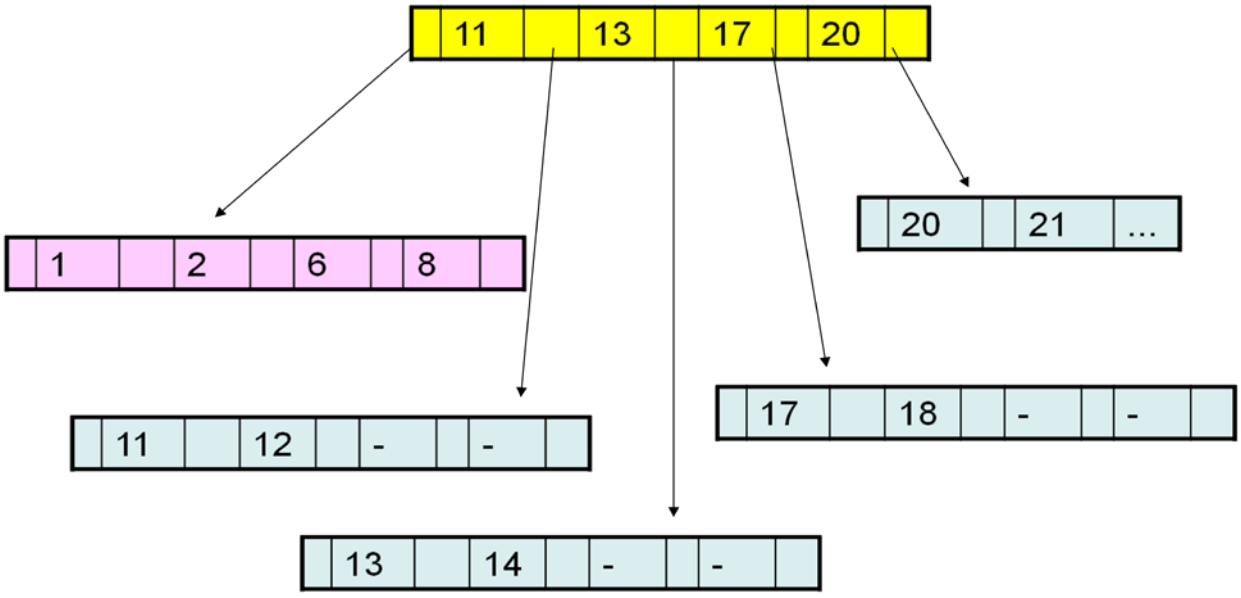
- *ri*(*x*) ***legge-da*** *wj*(*x*) in uno schedule *S* se *wj*(*x*) precede *ri*(*x*) in *S* e non c'è *wk*(*x*) fra *ri*(*x*) e *wj*(*x*) in *S*
- *wi*(*x*) in uno schedule *S* è ***scrittura finale*** *se* è l'ultima scrittura dell'oggetto *x* in *S*

Due schedule si dicono ***view-equivalenti*** *(Si* ≈V *Sj*) se hanno la stessa relazione *legge-da* e le stesse *scritture finali*. Uno schedule è ***view-serializzabile*** se è view-equivalente ad un qualche schedule seriale (Schedule in cui le transazioni si presentano una dopo l'altra).

Esercizio 5

Data la seguente lista di valori chiave L=(1,2,17,18,20,21,13,14,6,8,11,12)

5.a costruire un possibile B+-tree (fan-out=5) che contenga i seguenti nodi foglia: (1,2,6,8) (11,12), (13,14), (17,18), (20,21);



5.b mostrare l'albero dopo l'inserimento del valore chiave 3.