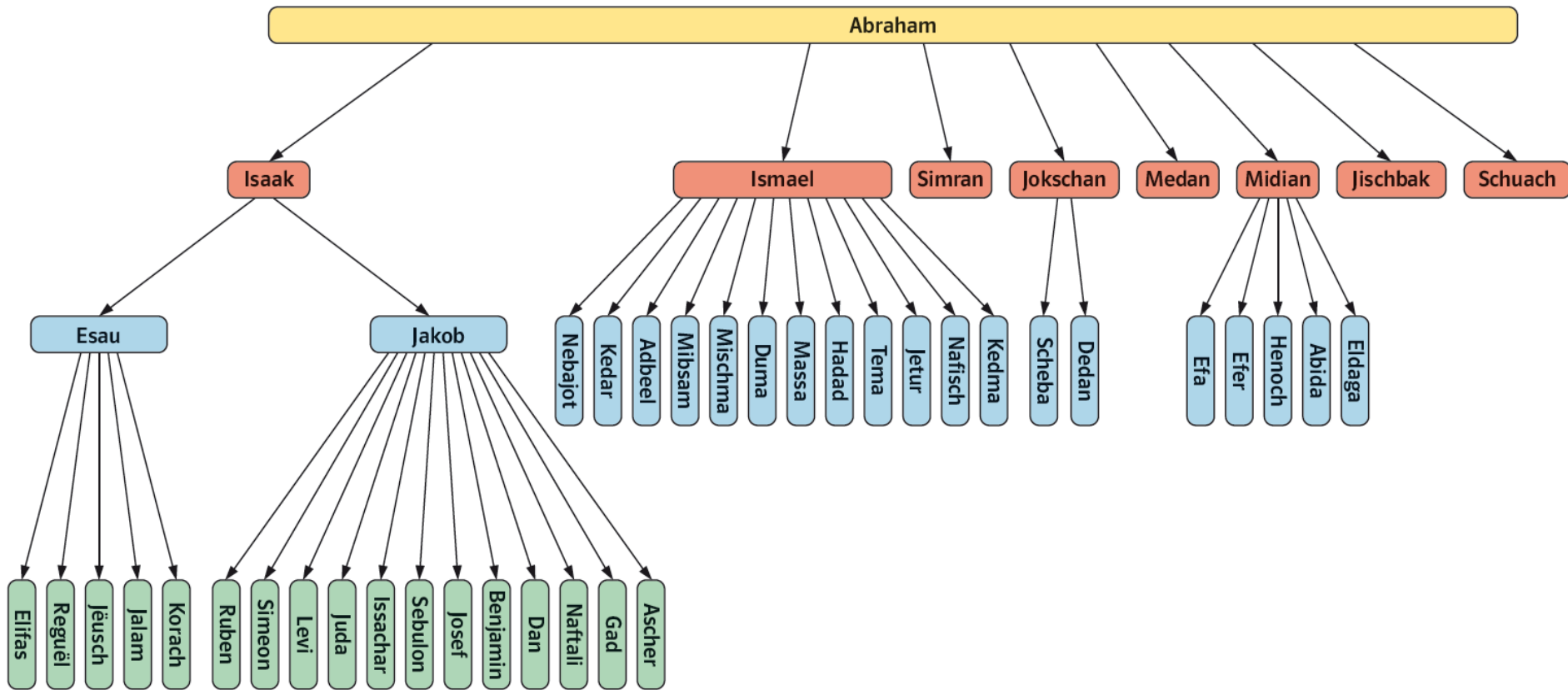
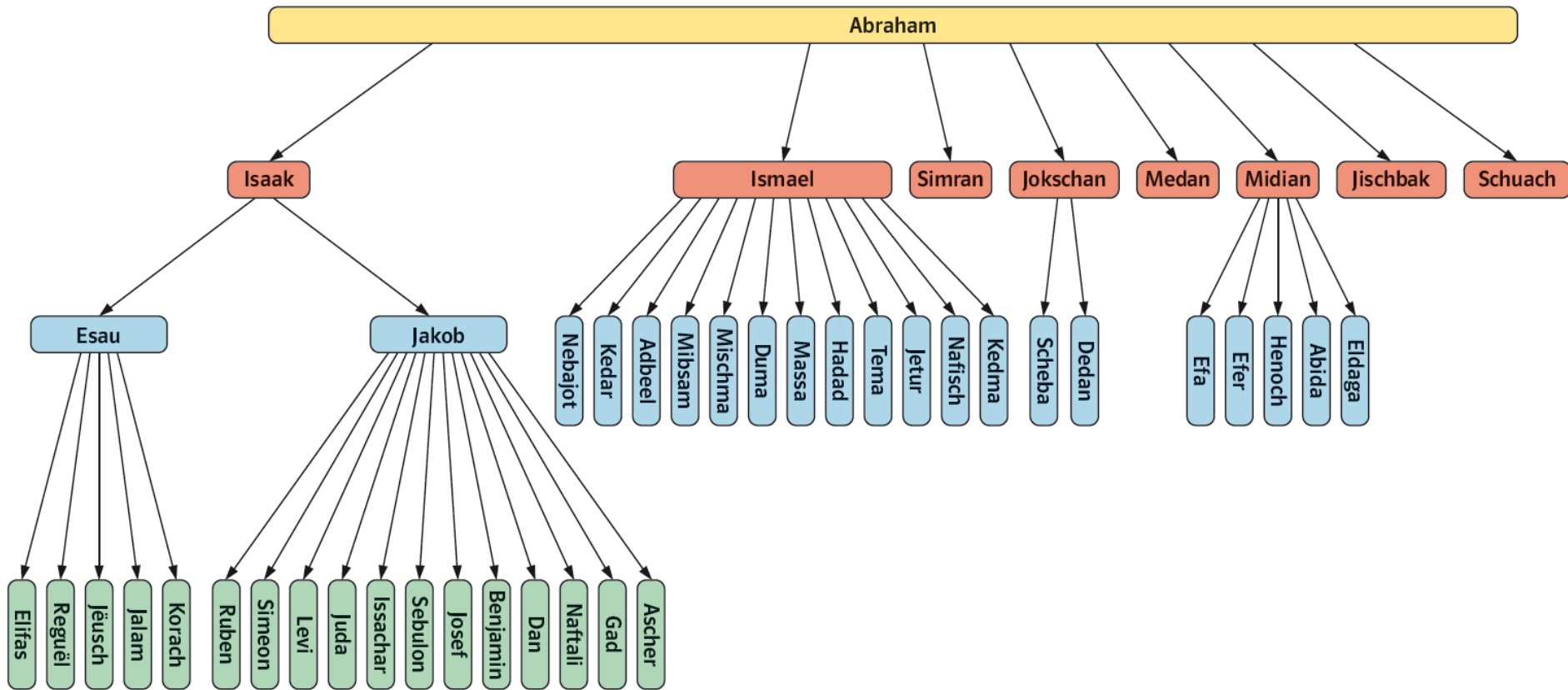


4 Hierarchische Baumstrukturen



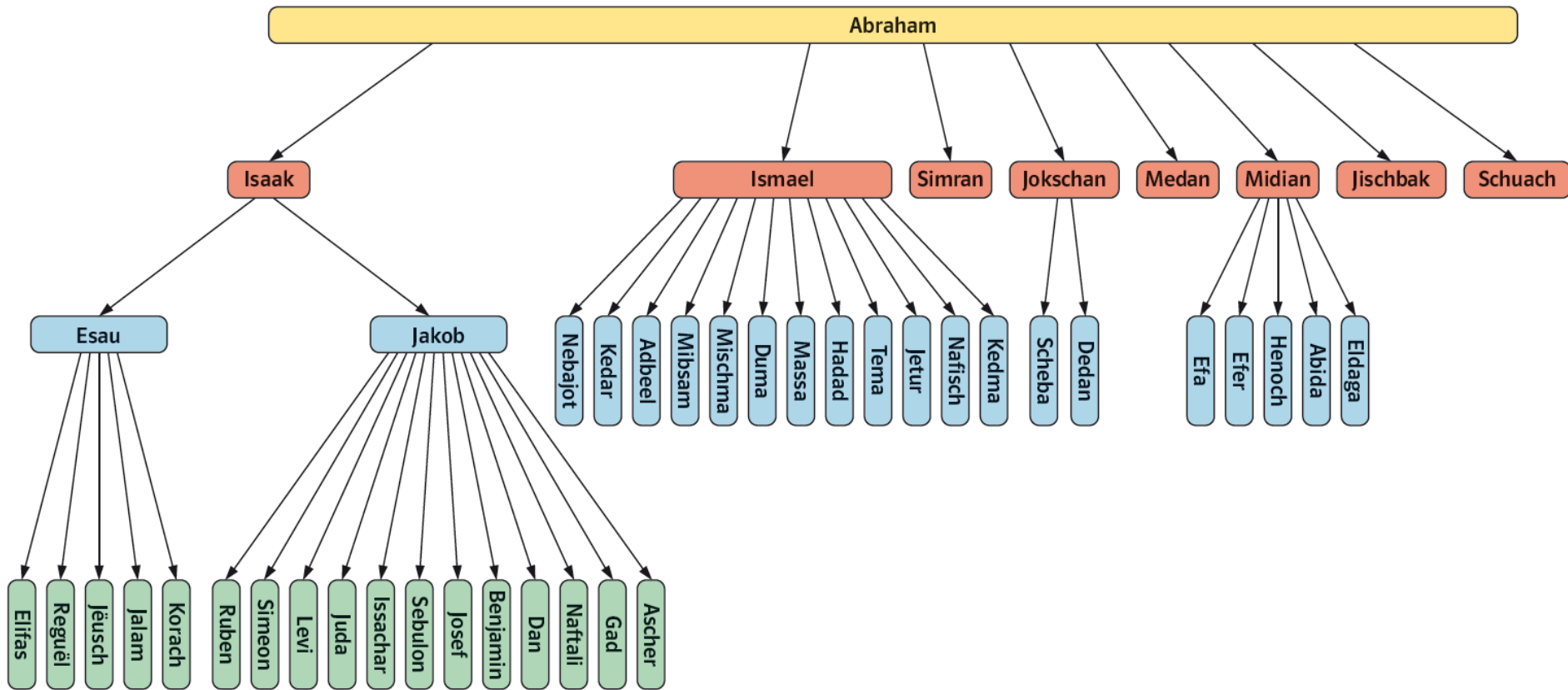
Wie viele Nachfahren von Abraham sind dokumentiert?

4 Hierarchische Baumstrukturen



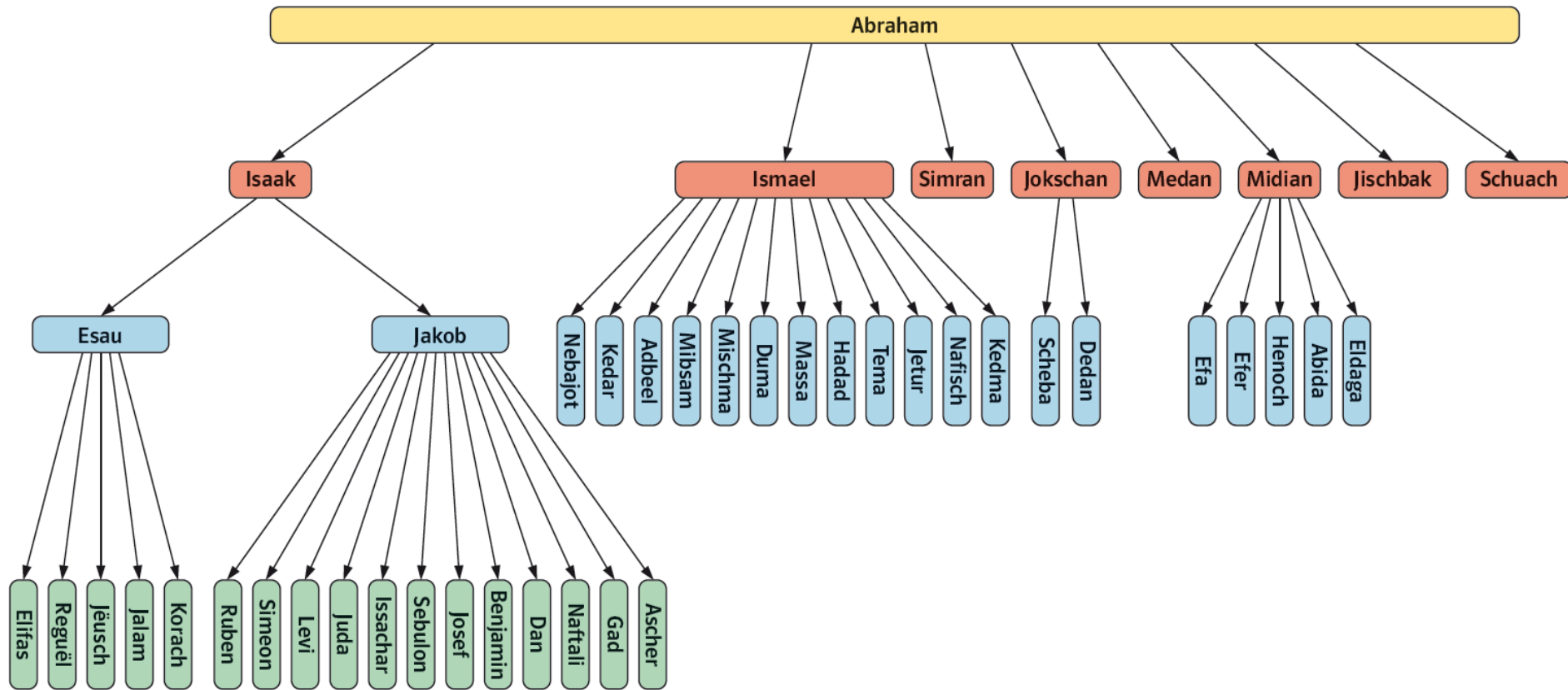
In welcher Generation nach Abraham befindet sich der Nachfahre Benjamin?

4 Hierarchische Baumstrukturen



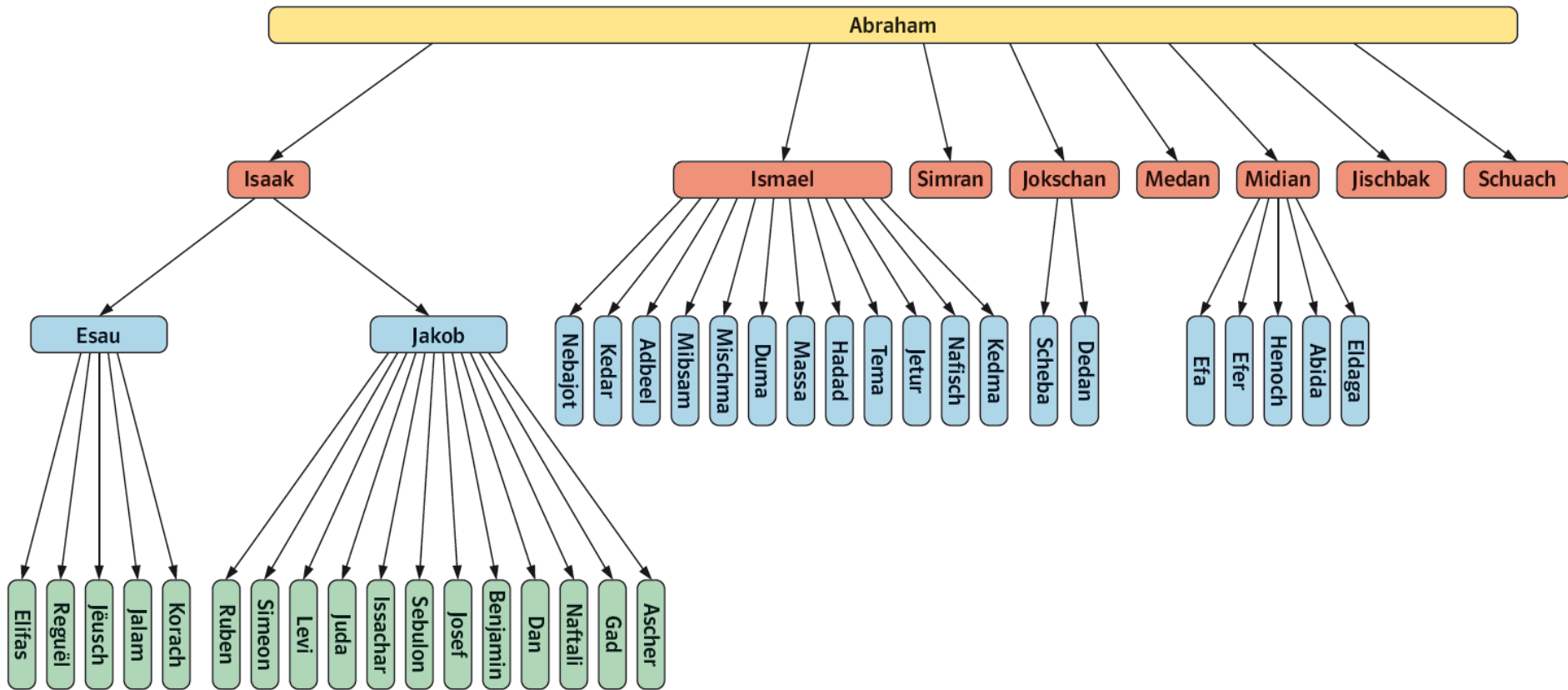
Wer waren die Söhne von Midian?

4 Hierarchische Baumstrukturen



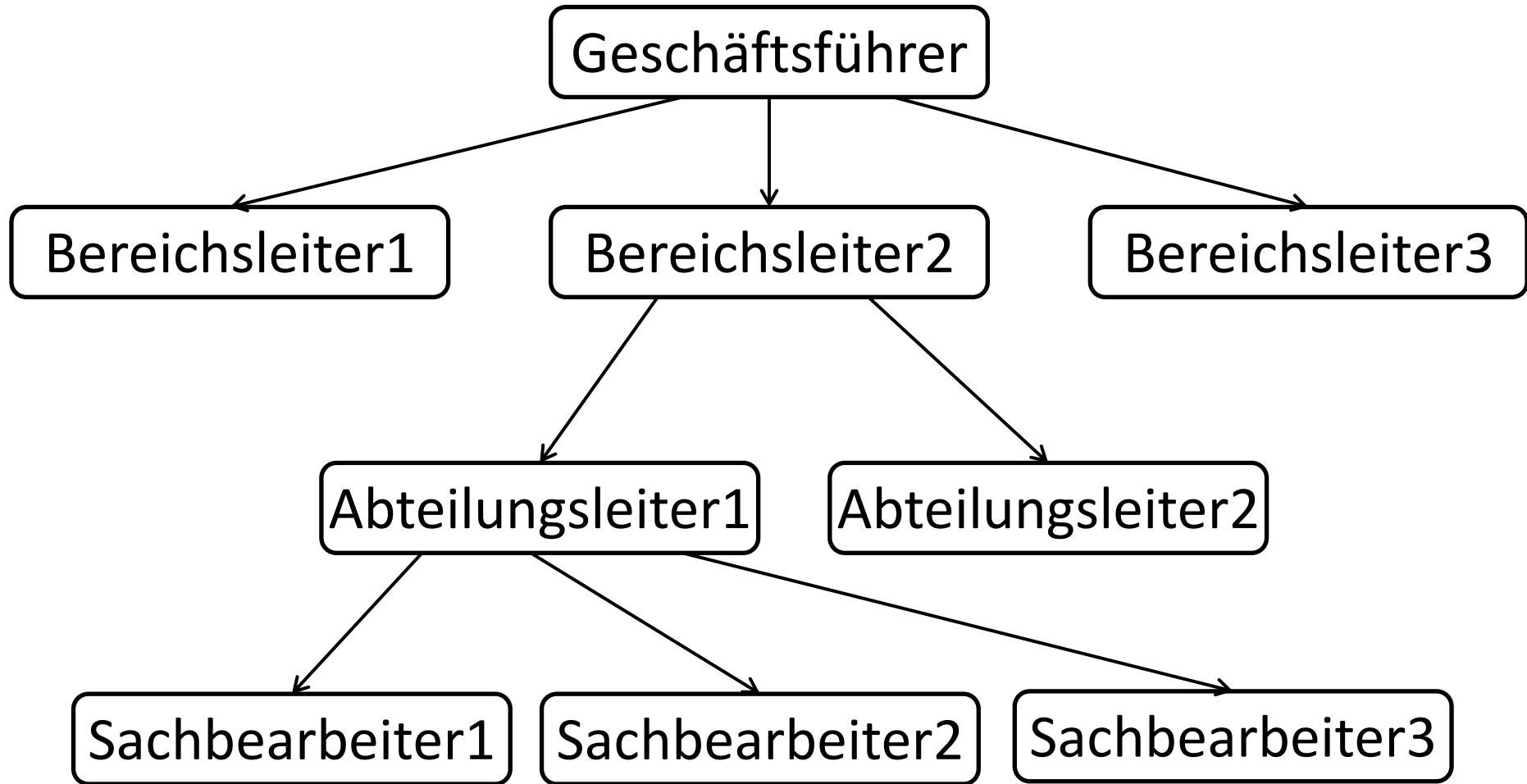
Waren Jakob und Kedar Cousins?

4 Hierarchische Baumstrukturen



Wer war der Vater von Esau?

S. 76/1 Unternehmenshierarchie



Unternehmen
Baum b
Unternehmen() mitarbeiterEinfuegen(int s_v, int s, String n, String h) datenAusgeben() anzahlMitarbeiterGeben() tiefeGeben() inhaltSuchen(int schluessel) vorgesetzterGeben(int schluessel)

Baum
Baumelement wurzel
Baum() wurzelSetzen(Datenelement de) alleDatenAusgeben() anzahlElementeGeben() kindknotenEinfuegen(Datenelement de_M, Datenelement de_K) tiefeGeben() inhaltSuchen(Datenelement de) elterninhaltGeben(Datenelement de)

<abstract> Baumelement
baumdatenAusgeben() anzahlDatenknotenGeben() kindknotenEinfuegen(DE de_M, DE de_K) tiefeGeben() inhaltSuchen(Datenelement de) elterninhaltGeben(DE de, DE de_M)

Abschluss

```
baumdatenAusgeben()  
anzahlDatenknotenGeben()  
kindknotenEinfuegen(DE de_M, DE de_K)  
tiefeGeben()  
inhaltSuchen(Datenelement de)  
elterninhaltGeben(DE de, DE de_M )
```

Datenknoten

```
Datenelement inhalt  
Baumelement[] kinderfeld  
int anzahlKindknoten  
Datenknoten(DE inh)  
baumdatenAusgeben()  
anzahlDatenknotenGeben()  
kindknotenEinfuegen(DE de_M, DE de_K)  
tiefeGeben()  
inhaltSuchen(Datenelement de)  
elterninhaltGeben(DE de, DE de_M )
```

<abstract> Datenelement

```
istGleich(DE de)  
schluesselGeben()  
datenAusgeben()
```

Mitarbeiter

```
int schluessel  
String name  
String hierarchie  
Mitarbeiter(int s, String nn, String h)  
istGleich(DE de)  
schluesselGeben()  
datenAusgeben()
```


Öffne S76-A1-Vorlage:

- Vervollständige die Methode `mitarbeiterEinfuegen(...)` und die dazu notwendigen weiteren Methoden in den entsprechenden Klassen.
- Ebenso:
 - ✓ `datenAusgeben()` (nur in DK)
 - ✓ `anzahlMitarbeiterGeben()`
 - ✓ `tiefeGeben()` (nur in Abschluss und DK)
 - ✓ `inhaltSuchen(int schluessel)`
 - ✓ `vorgesetzterGeben(int schluessel)`

In Unternehmen:

```
public void mitarbeiterEinfuegen(int s_v, int s, String n, String h){  
    b.kindknotenEinfuegen(new Mitarbeiter(s_v, "", ""),  
                           new Mitarbeiter(s, n, h));  
}
```

In Baum:

```
public void kindknotenEinfuegen(Datenelement de_M,  
                                Datenelement de_K){  
    wurzel.kindknotenEinfuegen(de_M, de_K);  
}
```

In Abschluss:

```
public void kindknotenEinfuegen(Datenelement de_M,  
                                Datenelement de_K){ }
```

In Datenknoten:

```
public void kindknotenEinfuegen(Datenelement de_M, Datenelement de_K){  
    if (inhalt.istGleich(de_M)){  
        if (anzahlKindknoten < kinderfeld.length){  
            kinderfeld[anzahlKindknoten]=new Datenknoten(de_K);  
            anzahlKindknoten++;  
        }  
        else {  
            System.out.println("Dem Vorgesetzten kann kein Mitarbeiter mehr  
                                zugeteilt werden.");  
        }  
    }  
    else {  
        for (int i=0; i<anzahlKindknoten;i++){  
            kinderfeld[i].kindknotenEinfuegen(de_M, de_K);  
        }  
    }  
}
```

In Datenknoten:

```
public void baumdatenAusgeben(){  
    //Preorder  
    inhalt.datenAusgeben();  
    for (int i=0; i<anzahlKindknoten;i++){  
        kinderfeld[i].baumdatenAusgeben();  
    }  
}
```

In Unternehmen:

```
public int anzahlMitarbeiterGeben(){  
    return b.anzahlElementeGeben();  
}
```

In Baum:

```
public int anzahlElementeGeben(){  
    return wurzel.anzahlDatenknotenGeben();  
}
```

In Abschluss:

```
public int anzahlDatenknotenGeben(){  
    return 0;  
}
```

In Datenknoten:

```
public int anzahlDatenknotenGeben(){  
    int anzahl = 0;  
    for(int i=0; i<anzahlKindknoten;i++){  
        anzahl = anzahl + kinderfeld[i].anzahlDatenknotenGeben();  
    }  
    return anzahl+1;  
}
```

In Abschluss:

Wurzel: Tiefe 0

```
public int tiefeGeben(){  
    return -1;  
}
```

In Datenknoten:

```
public int tiefeGeben(){  
    int tiefe=-1;  
    for (int i=0; i<anzahlKindknoten;i++){  
        int t = kinderfeld[i].tiefeGeben();  
        if (tiefe<t) tiefe = t;  
    }  
    return tiefe +1;  
}
```

In Unternehmen:

```
public Datenelement inhaltSuchen(int schluessel){  
    return b.inhaltSuchen(new Mitarbeiter(schluessel, "", ""));  
}
```

In Baum:

```
public Datenelement inhaltSuchen(Datenelement de){  
    return wurzel.inhaltSuchen(de);  
}
```

In Abschluss:

```
public Datenelement inhaltSuchen(Datenelement de){  
    return de;  
}
```


In Datenknoten:

```
public Datenelement inhaltSuchen(Datenelement de){  
    //könnte man noch effizienter gestalten;  
    //Abbruch wenn in einem Kindknoten Übereinstimmung entdeckt  
    //wurde
```

```
        if (inhalt.istGleich(de)){de = inhalt; }  
        else {  
            for (int i=0; i<anzahlKindknoten;i++){  
                de = kinderfeld[i].inhaltSuchen(de);  
            }  
        }  
        return de;  
    }
```

In Unternehmen:

```
public Datenelement vorgesetzterGeben(int schluessel){  
    return b.elterninhaltGeben(new Mitarbeiter(schluessel, "", ""));  
}
```

Gib den Inhalt des Mutterknotens als 2. Parameter mit durch!

In Baum:

```
public Datenelement elterninhaltGeben(Datenelement de){  
    return wurzel.elterninhaltGeben(de, null);  
}
```

In Abschluss:

```
public Datenelement elterninhaltGeben(Datenelement de,  
Datenelement de_M){  
    return null;  
}
```

In Datenknoten:

```
public Datenelement elterninhaltGeben(Datenelement de,  
Datenelement de_M){  
    Datenelement de_rueck = null;  
    if (inhalt.istGleich(de)){  
        return de_M;  
    }  
    else {  
        for (int i=0; i<anzahlKindknoten;i++){  
            if (de_rueck == null){  
                de_rueck = kinderfeld[i].elterninhaltGeben(de, this.inhalt);  
            }  
        }  
        return de_rueck;  
    }  
}
```