

Die Datenbank enthält folgende Daten über eine ehemalige Saison der ersten, zweiten und dritten Bundesliga:

```
Verein ( V_ID, Name, Liga )
Spiel ( Spiel_ID, Spieltag, Datum, Uhrzeit, Heim, Gast,
        Tore_Heim, Tore_Gast)
Spieler ( Spieler_ID, Vereins_ID , Trikot_Nr, Spieler_Name,
          Land, Spiele, Tore, Vorlagen )
Liga ( Liga_Nr, Verband , Erstaustragung , Meister ,
       Rekordspieler , Spiele_Rekordspieler)
```

Unterstrichene Attribute sind Primärschlüssel, kursive Attribute sind Fremdschlüssel.

Abfragen über eine Tabelle

1. Zeige alle verfügbaren Daten der Tabelle „Verein“ an.
2. Welche Vereine spielen in der zweiten Liga?
3. Wann wurde in der 1. Fußball-Bundesliga das erste Mal ein Spiel ausgetragen?
4. Wähle Liga_Nr, Verband und Rekordspieler aller drei Ligen aus.
5. Zeige alle Vereine an, die die Abkürzung „FC“ im Namen haben.
6. An welchem Tag fand das erste Spiel in der Saison statt?
7. Wie viele Tore wurden in der Saison geschossen ?
8. Welche Spieler haben in dieser Saison mehr als 15 Tore geschossen? Ordne die Liste so, dass der/die Spieler mit den meisten Toren am Anfang stehen.
9. Wie viele Spieler tragen die Trikot-Nr 12 ? Nenne die Ergebnisspalte „Anzahl“.
10. Welche deutschen Spieler (Land: D) haben in dieser Saison noch an keinem Spiel teilgenommen?
11. Zeige die Daten aller Spiele an, die am ersten Spieltag aller drei Ligen nach 17 Uhr begonnen haben.
12. Wie viele Tore wurden bisher durchschnittlich von den Spielern geschossen, die schon an mehr als 3 Spielen teilgenommen und mehr als 2 Vorlagen geliefert ha-

ben?

13. Suche alle Spiele, die im August stattgefunden und nach 19 Uhr begonnen haben.

Subquery (Unterabfragen)

14. Welcher Verein war Meister der zweiten Liga? Schreibe zuerst die Abfrage, die die Vereinsnummer des Meisters aus der Tabelle Liga ermittelt. Erstelle dann die Abfrage aus Verein, die den zugehörigen Namen findet und füge die erste als Subquery ein.
15. Welcher Spieler hat bisher für den Verein mit der ID 20 die meisten Tore geschossen?
16. Liste alle Spieler auf, die überdurchschnittlich viele Tore geschossen haben. Bestimme dazu in einer Subquery die durchschnittliche Anzahl Tore.

Abfragen über mehrere Tabellen durch Verknüpfung (Join)

17. Welche Spieler spielen für den Verein "SC Freiburg"? Gib auch die Trikotnummer und das Heimatland jedes Spielers sowie die Anzahl seiner Tore mit aus. Ordne die Ergebnisse aufsteigend nach der Trikotnummer.
18. Gib Trikotnummer, Name und die Anzahl der Tore aller Spieler der zweiten Liga aus, die bisher schon mehr als 10 Tore geschossen haben. Ordne nach der Anzahl ihrer Tore.
19. a) Welche Vereine haben bisher ein Heimspiel gegen den Verein mit der V_ID 10 gewonnen? Wie lauteten die Ergebnisse dieser Spiele?
b) Welche Vereine haben bisher ein Spiel gegen den Verein mit der V_ID 10 gewonnen? Wie lauteten die Ergebnisse dieser Spiele? Hinweis: hier muss man eine umfangreichere Bedingung in der ON-Klausel formulieren.
20. An welchen Tagen finden die Spiele der ersten Liga statt?
Hinweis: Jedes Datum darf nur einmal ausgegeben werden → DISTINCT
21. Wer (Name) hat am zweiten Spieltag gegen „Dynamo Dresden“ gespielt? Finde die V_ID von „Dynamo Dresden“ zuvor mit einer eigenen SQL-Abfrage heraus.

Baue diese Abfrage dann als SUBSELECT (Unterabfrage) in die ganze Abfrage mit ein: ...WHERE Heim = (SELECT „V_ID“ FROM....) .

22. Suche für jeden brasilianischen Spieler der ersten Liga heraus, an wie vielen Spielen er teilgenommen hat. Gib außerdem die Anzahl ihrer Tore und Vorlagen und den Namen des Vereins aus, für den sie spielen.
23. Welcher Spieler hat für den 1. FC Nürnberg die meisten Tore geschossen? Hinweis: in einer Subquery Verein und Spieler joinen und die maximale Anzahl Tore für Nürnberg herausfinden, Dann in der Hauptquery rausfinden, welcher Spieler von Nürnberg (JOIN) diese Anzahl Tore geschossen hat.
24. Welche Vereine haben am ersten Spieltag der ersten Liga gegeneinander gespielt, wie lauten die Ergebnisse? Hinweis: hier muss man zweimal auf die Tabelle Verein JOINen, mit jeweils unterschiedlichen ON-Bedingungen.
25. Gegen welche Vereine hat der „FC Schalke 04“ bisher Auswärtsspiele bestritten?
26. Wie viele Spiele hat „Hannover 96“ bis heute gewonnen?

Daten gruppieren

27. Wie viele Spieler hat jeder Verein der ersten Liga? Gib die Ergebnisse mit dem Vereinsnamen aus und ordne sie absteigend nach der Anzahl der Spieler.
28. Wie viele Tore sind bisher in jeder Liga gefallen?
29. Welche Vereine (Name, Liga) haben mindestens fünfmal unentschieden gespielt? Ordne das Ergebnis aufsteigend nach der Liga und absteigend nach der Zahl der Unentschieden.
30. Erstelle eine Liste in der für jeden Verein anzeigt, wie viele Heimspiele er gewonnen hat.
31. Gib die Liste aller Zweitligavereine mit Ihren in Auswärtsspielen erreichten Punkten aus.
Hinweis: Bei jedem Spiel gilt: Sieg = 3 Punkte, Unentschieden = 1 Punkt, Niederlage = 0 Punkte

32. Gib die aktuelle Spieltabelle der 1. Bundesliga aus. Diese beinhaltet für jeden Verein: Den Vereinsnamen, die Anzahl der bisher gespielten Spiele, die Anzahl der Siege, Unentschieden und Niederlagen, die geschossenen und erhaltenen Tore, die Tordifferenz und die Anzahl der Punkte.

33. Welcher Verein hat bisher die meisten Tore geschossen? (!)

Hinweis: Mit dem Vergleich Wert \geq ALL(...Unterabfrage...) in der HAVING-Klausel vergleicht man „Wert“ mit jeder einzelnen Zeile, die Unterabfrage liefert.

```
SELECT c.name, c.area FROM country
WHERE c.area > ALL (SELECT c1.area FROM country c1 JOIN encompasses e
ON c1.code = e.country WHERE c.code != c1.code)
```

→ Die WHERE-Bedingung ist jedesmal erfüllt, wenn in der Unterabfrage alle Flächen kleiner als c.area sind. (credits for this example to N.Nazar). Logischerweise darf die Unterabfrage nur eine Spalte liefern, sonst wird das mit dem Vergleich schwierig.

34. Gesucht sind Vereinsname, Spieler_ID, Trikotnummer und Name aller Spieler, die für den Verein spielen, der in dieser Saison die meisten Niederlagen erlitten hat (auch mehrere Vereine mit gleicher Anzahl möglich). (!)

Hinweis: \geq ALL(...)

35. Denke dir selbst eine interessante Fragestellung zur Datenbank „Bundesliga“ aus, erstelle die passende SQL-Anweisung und teste sie. Tausche die Fragestellung anschließend mit deinem Nachbarn aus und bearbeitet eure Aufgaben gegenseitig.