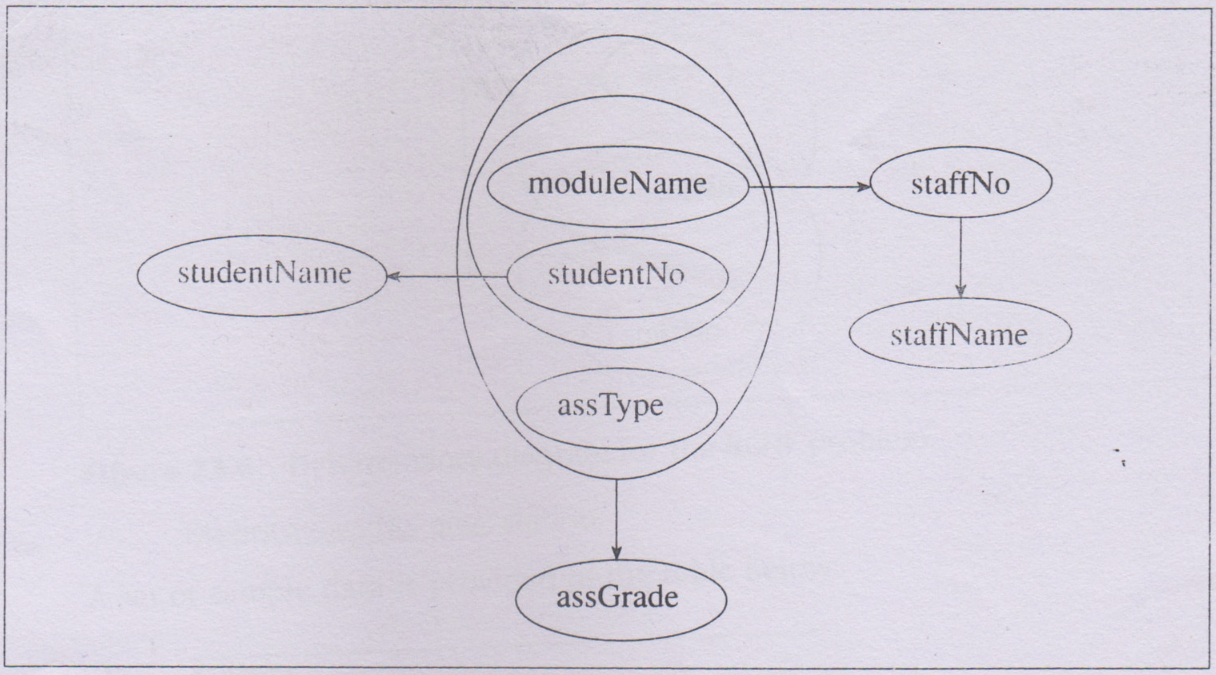

Figure 23.5 Determinancy diagram for the academic schema

for the data set decomposed in the early section of this chapter is illustrated in figure 23.5.

First, second, third normal forms and Boyce-Codd normal form are all about functional dependencies. Fourth and fifth normal forms are about non-functional dependencies.

First normal form concerns repeating groups. Since functional dependencies document one-to-many relationships between data they are an explicit representation of repeating groups. Second normal form concerns part-key dependencies. Here we are segmenting functional dependencies out from a compound key. Third normal form concerns inter-data dependencies. Here we are identifying determinants within the non-key attributes of a table.

23.9 Boyce–Codd Normal Form

Boyce-Codd normal form (BCNF) is a stronger normal form than 3NF and is designed to cover anomalies that arise when there is more than one candidate key in some set of data requirements.

Suppose we have introduced a scheme of majors and minors into our degree schemes at a university. The business rules relevant to that part of this domain covering majors are listed below:

1. Each student may major in several areas.
2. A student has one tutor for each area.
3. Each area has several tutors but a tutor advises in only one area.
4. Each tutor advises several students in an area.

A diagram incorporating all these business rules is illustrated in figure 23.6.

On the basis of these business rules a schema is produced in 3NF represented in the bracketing notation below:

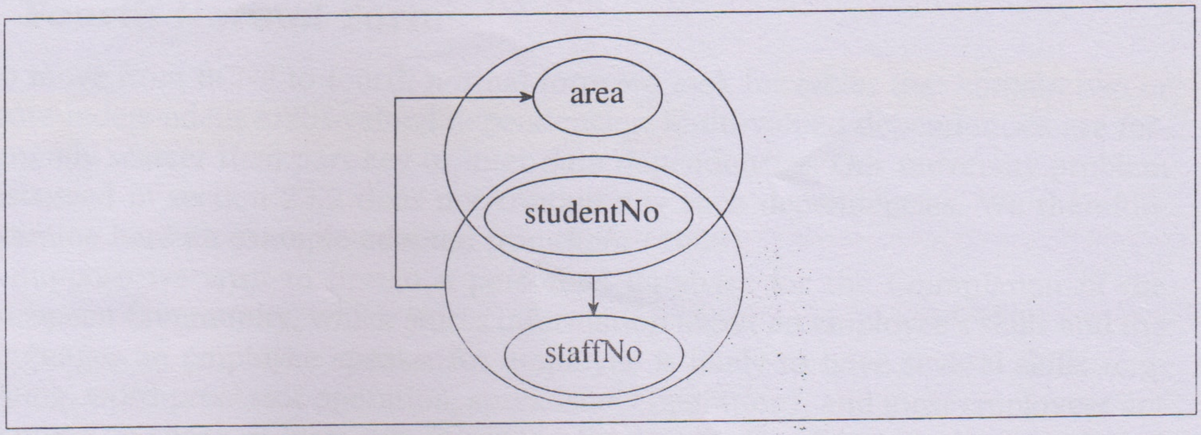

Figure 23.6 Determinancy diagram for the BCNF problem

Majors(<u>studentNo</u>, <u>area</u>, staffNo)

A set of sample data is provided in the table below.

Majors

StudentNo	***Area***	***StaffNo***
123456	Computer Science	234
234567	Information Systems	345
123456	Software Engineering	456
234567	Graphic Design	678
345678	Information Systems	567

This schema is in 3NF because there are no partial dependencies and no inter-data dependencies. However, anomalies will still arise when we come to update this relation. For instance:

1. Suppose student 123456 changes one of her majors from computer science to information systems. Doing this means that we lose information about staffNo 234 tutoring on computer science. This is an update anomaly.
2. Suppose we wish to insert a new row to establish the fact that staffNo 789 tutors on computer science. We cannot do this until at lest one student takes this area as their major. This is an insertion anomaly.
3. Suppose student 345678 withdraws from the university. In removing the relevant row we lose information about staffNo 567 being a tutor in the area of information systems. This is a deletion anomaly.

These anomalies occur because there are two overlapping candidate keys in this problem. R. F. Boyce and E. F.Codd identified this problem and proposed a solution in terms of a stronger normal form known as BCNF. A relation is in BCNF if every determinant is a candidate key. The schema above can be converted into BCNF in one of two ways. The two schemas are presented in bracketing notation below:

Schema 1:
StudentTutors(<u>studentNo</u>, <u>staffNo</u>)
TutorAreas(<u>staffNo</u>, area)
Schema 2:
StudentTutors(<u>studentNo</u>, <u>area</u>)
TutorAreas(<u>staffNo</u>, area)

23.10 Fourth Normal Form

To move from BCNF to fourth normal form we look for tables that contain two or more independent multi-valued dependencies. Multi-valued dependencies are fortunately scarcer than part-key or inter-data dependencies. Our university problem discussed in section 23.2 does not contain any such dependencies. We therefore examine here an example adapted from Kent (1983).

Suppose we wish to design a personnel database for the Commission of the European Community, which stores information about an employee's skills and the languages an employee speaks. An employee is likely to have several skills (e.g. typing, word-processor operation, spreadsheet operation), and most employees are required to speak at least two European languages. Our first attempt at a design for this system might aggregate all the data together in one table as below:

EUEmployees

employeeNo	***skill***	***language***
0122443	Typing	English
0122443	Typing	French
0122443	Dictation	English
0221133	Typing	German
0221133	Dictation	French
0332222	Typing	French
0332222	Typing	English

However, we wish to add the restriction that each employee exercises skill and language use independently. In other words, typing as a skill is not inherently linked with the ability to speak a particular language.

Under fourth normal form these two relationships should not be represented in a single table as in figure 23.7(A). This is evident when we draw the determinancy diagram as in figure 23.7(B). Having two independent multi-valued dependencies means that we must split the table into two as below:

EUSkills

employeeNo	***skill***
0122443	Typing
0122443	Dictation
0221133	Typing
0221133	Dictation
0332222	Typing

EULanguages

employeeNo	***language***
0122443	English
0122443	French
0221133	German
0221133	French
0332222	French

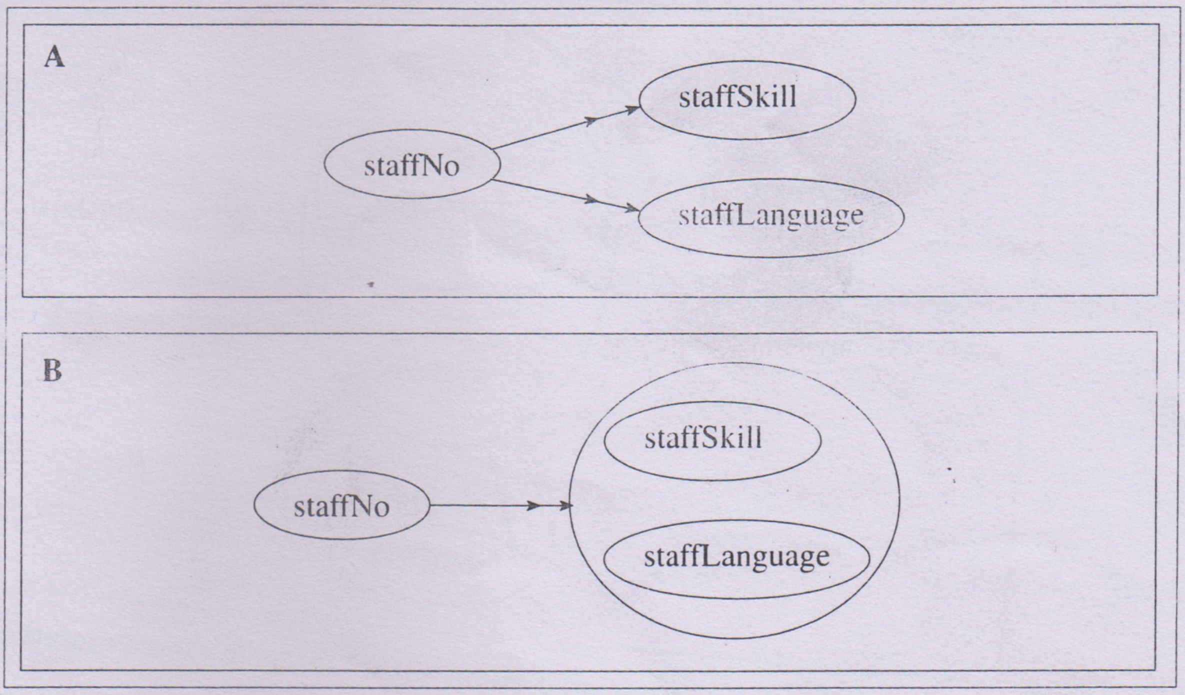

Figure 23.7 Formal normal form

23.11 Fifth Normal Form

Generally speaking, a fourth normal form table is in fifth normal form if it cannot be non-loss decomposed into a series of smaller tables. Consider the table below which stores information about automobile agents, automobile companies and automobiles:

Outlets

agent	***company***	***automobile***
Jones	Ford	Car
Jones	Vauxhall	Van
Smith	Ford	Van
Smith	Vauxhall	Car

If agents represent companies, companies make products, and agents sell products, then we might want to record which agent sells which product for which company (figure 23.8). To do this we need the structure above. We cannot decompose the structure because although agent Jones sells cars made by Ford and vans made by Vauxhall he does not sell Ford vans or Vauxhall cars. Fifth normal form concerns interdependent multi-value dependencies, otherwise known as join dependencies.

23.12 SUMMARY

1. **Normalisation is the process of producing a schema not subject to file maintenance anomalies.**
2. **We have discussed two complementary approaches to bottom-up data analysis: non-loss decomposition and determinancy diagramming.**
3. **Non-loss decomposition is a step-by-step approach to producing a fully normalised schema.**