Chapter 7:
LOGICAL DATABASE DESIGN

NORMALIZATION (PART 2)

DETERMINANCY DIAGRAMS

Determinancy Diagramming

- Classic normalization is described as a process of non-loss decomposition.
- Non-loss decomposition is a design process guaranteed to produce a data set free from file maintenance problems.

Determinancy Diagramming

- Disadvantages of using this as a database design technique:
- It requires all of the data set to be in place before the process can begin.
- For any reasonable large data set the process is extremely time-consuming, difficult to apply and prone to human error.

ADVANTAGE OF DETERMINANCY DIAGRAM

- Main advantage of the determinancy diagramming technique is:
- It provides a mechanism for designing a database incrementally.
- No need to have a complete data set in hand to begin the process of design.

ADVANTAGE OF DETERMINANCY DIAGRAM

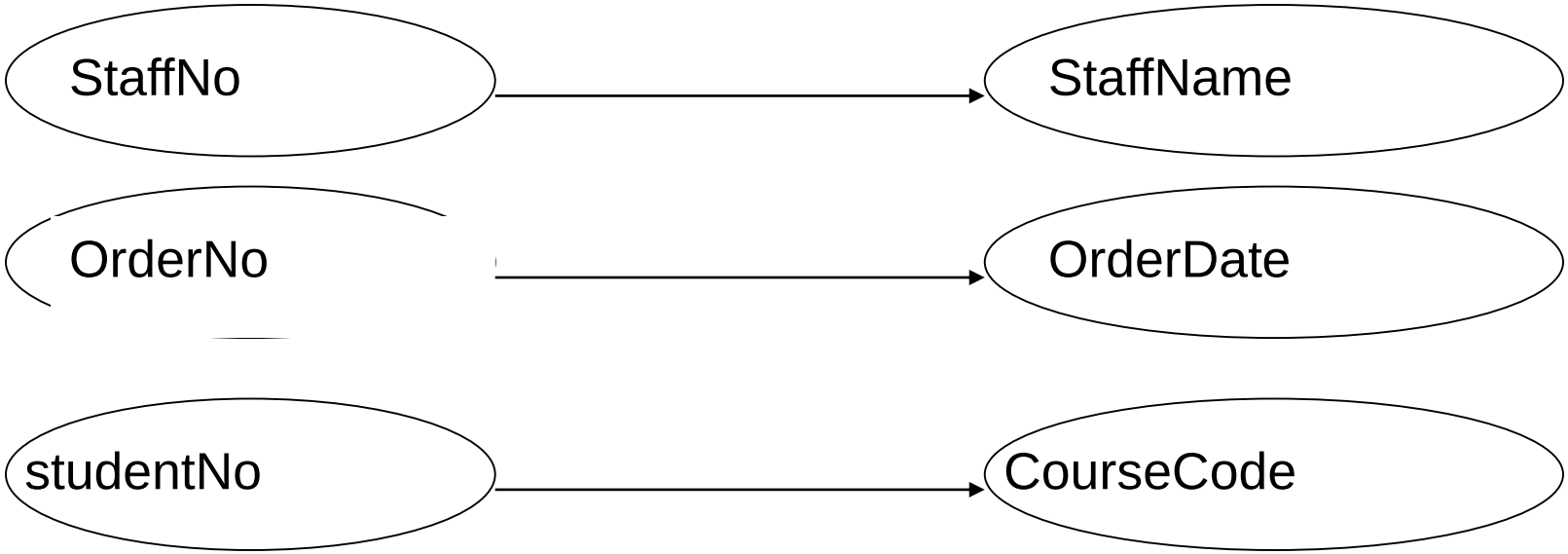
- Only a small collection of central data items are required to start design.
- Core data items can then be continuously added until all the dependencies are fully documented.
- Once documented, the determinancy diagram can be transformed into a relational schema in a couple of steps.

DEFINITION

- Def : A diagram which documents the determinancy or dependency between data items. (Graphical Notation alternative to Normalization)
- Data items are drawn on a determinancy diagram as labeled ovals, circles or bubbles.
- Functional dependency is represented between two data items by drawing a single-headed arrow from the determinant data item to the dependent data item.

DEPENDENCY DIAGRAM

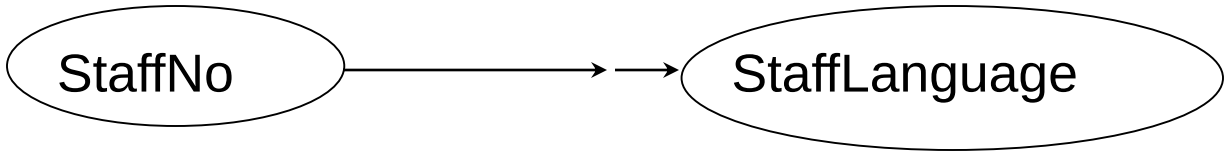
- Functional Dependencies



Non-Functional (Multi-Valued) Determinancy

- Not all dependencies can be modeled in terms of functions.
- Data item B is said to be non-functionally dependent on data item A if for every value of data item A there is a delimited set of values for data item B.
- The mapping is no longer functional because it is one to many.

Non-Functional (Multi-Valued) Determinancy



For every staffNo we can identify a delimited set of language codes which apply to that staff member.

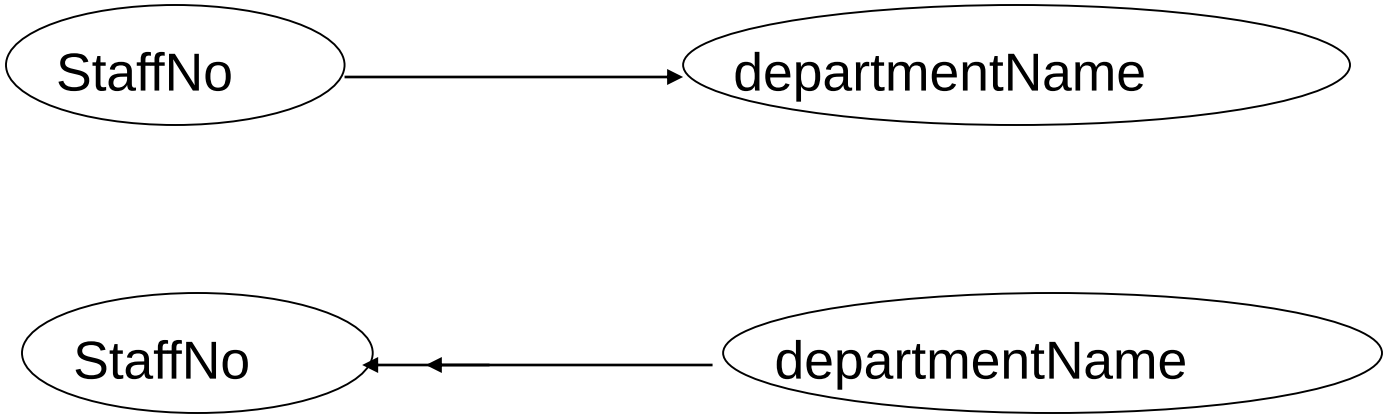
- Multi-Valued or non-functional dependency is indicated by drawing a double-headed arrow from the determinant to the dependent data item

Pragmatic of Determinancy Diagramming

- Dependencies between any two data items may be diagrammed as A to B or B to A, but not both.
- A single-valued dependency in one direction could be a multi-valued dependency in the opposite direction.

EXAMPLE

StaffNo and departmentName



WHICH DIRECTION TO CHOOSE?

Always chose the direction of the functional dependency.

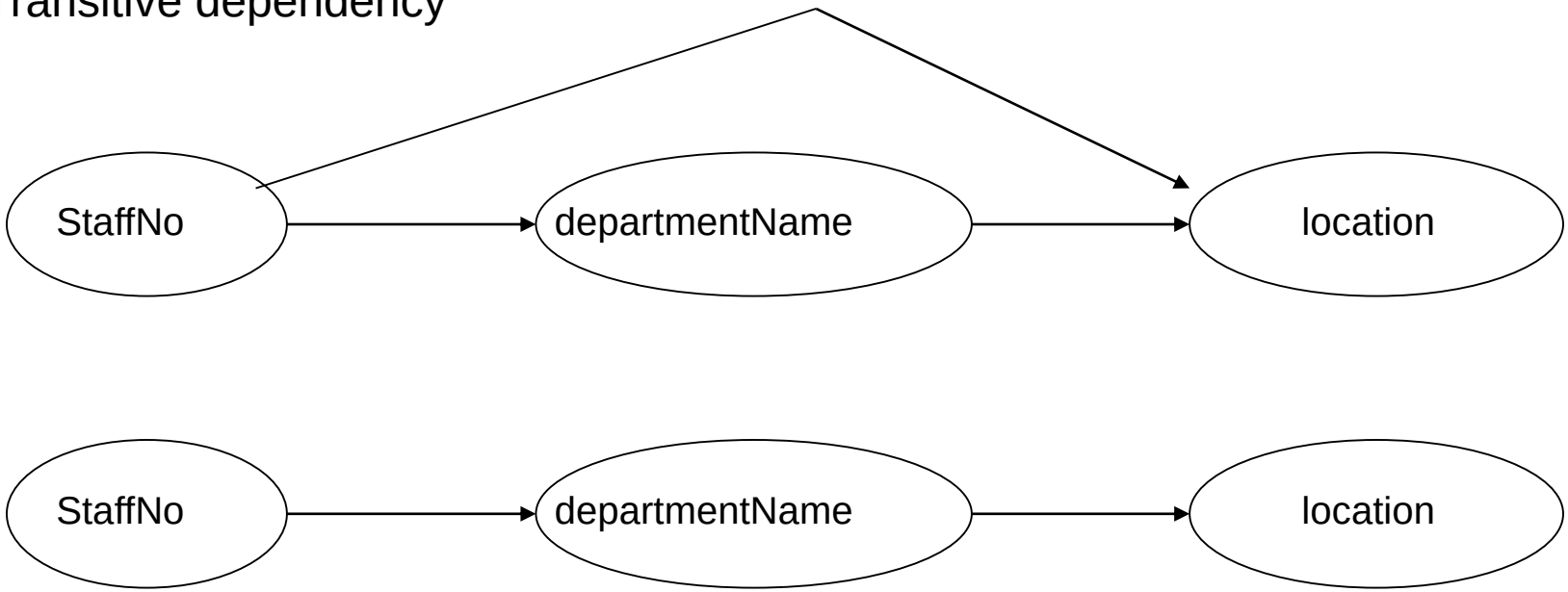
- It reduces the number of compound keys required.
- If a functional or non-functional dependency exists in both directions then we choose either.

Example

- StaffNo and extensionNo
- StaffNo might functionally determine ExtensionNo and ExtensionNo in turn functionally determine StaffNo
- More data items are likely to be dependent on staffNo than upon extensionNo.
- Choose the one that is most likely to be influence by other dependencies arising from the data item.

COMPOUND & TRANSITIVE DETERMINANCY

1. Transitive dependency



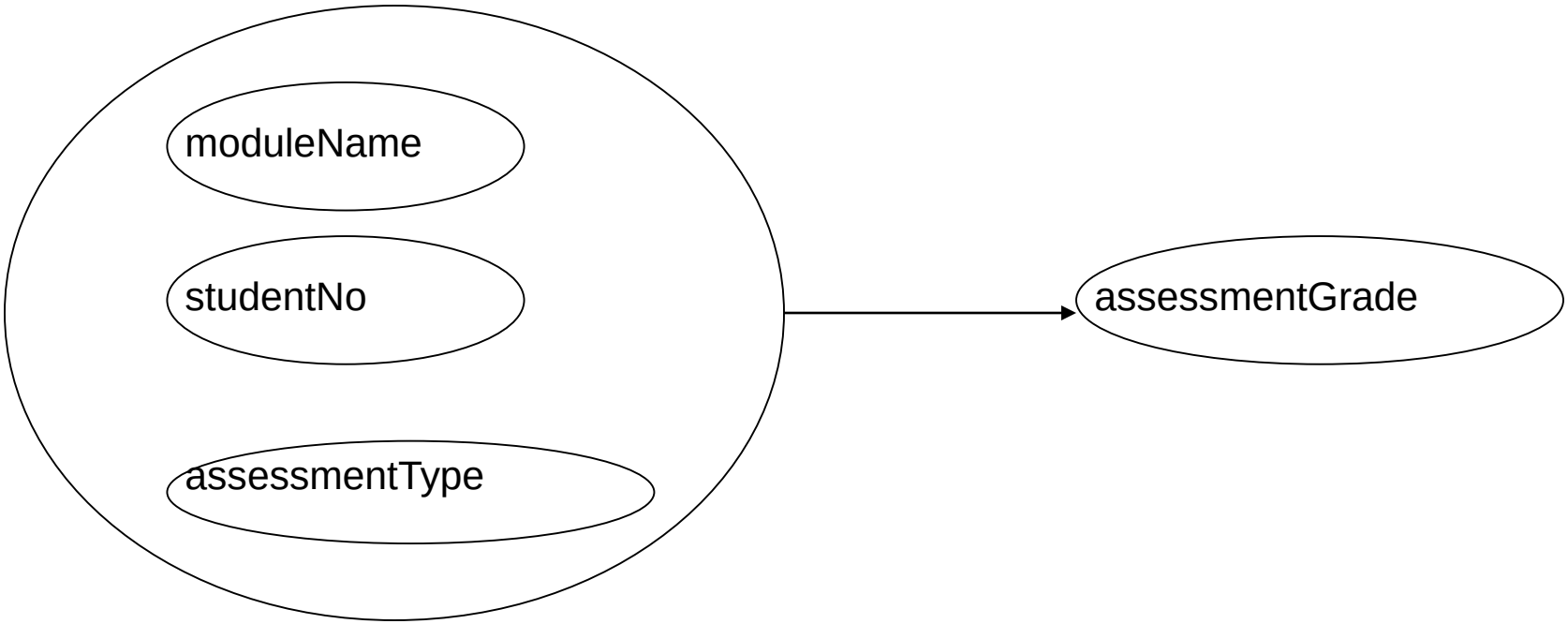
COMPOUND & TRANSITIVE DETEMINANCY

- A functional dependencies exists from
- StaffNo to departmentName
- DepartmentName to location
- StaffNo to location
- Also refers as transitive dependencies
- Any situation in which A determines B, B determines C, A also determines C can be simplifies into a chain A to B and B to C.

COMPOUND DETERMINANCY

- Sometimes a combination of two or more data items gives a dependent relationship.
- Example: ModuleName, StudentNo, and AssessmentType is used to determine assessmentGrade.
- How do we show this in determinancy diagram?

COMPOUND DETERMINANCY



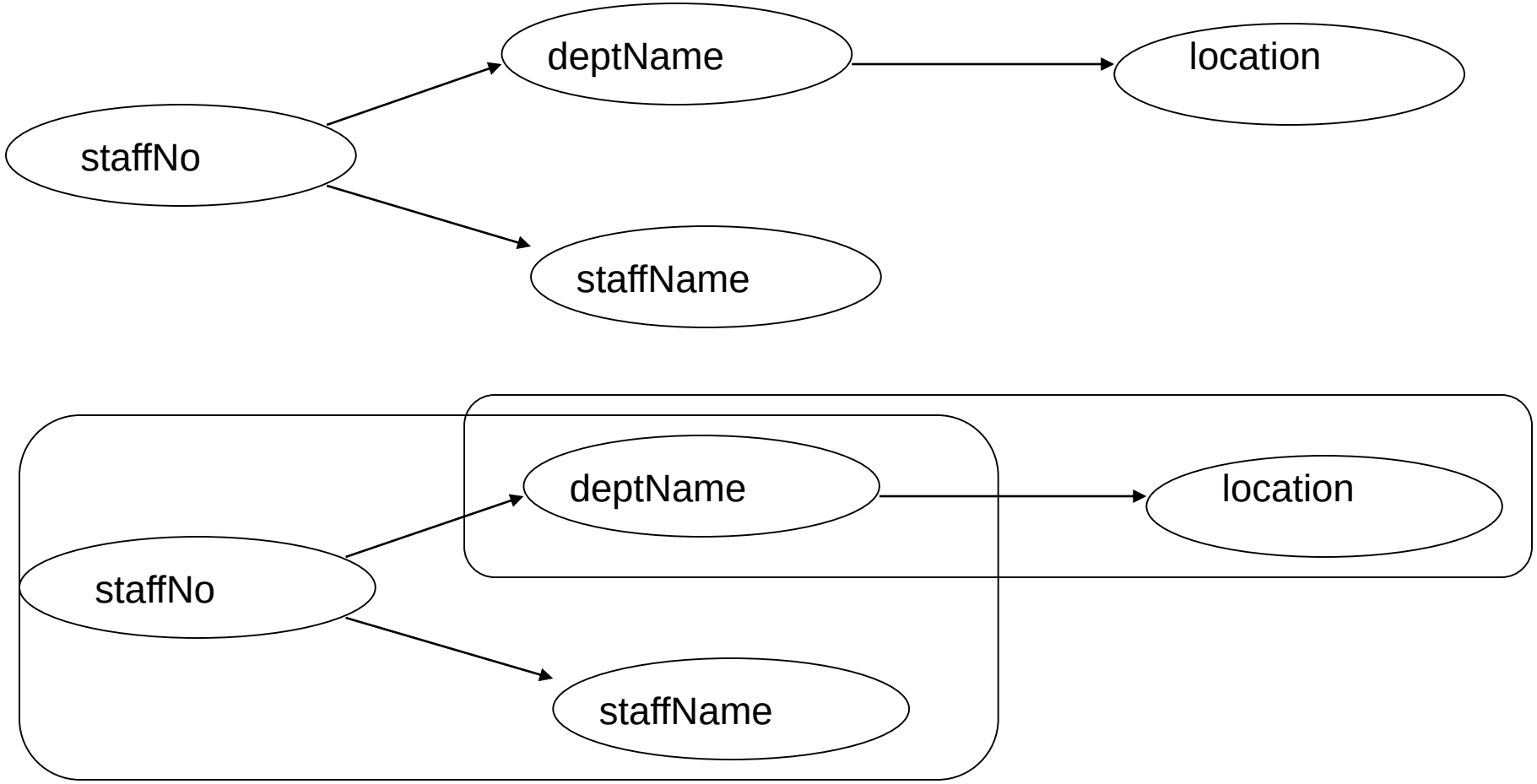
COMPOUND DETERMINANCY

- A compound determinant is drawn as an enclosing bubble around two or more data item bubbles. The functional dependency is drawn from the outermost bubble.

Accommodating Functional Dependencies

- Accommodation is the process of transforming the dependency diagram into a set of table structures by applying the following rule:
- Every functional determinant becomes the primary key of a table. All immediate dependent data items become non-key attributes of the table.
- This is frequently referred as the Boyce-Codd rule.

EXAMPLE


deptName
location
staffNo
staffName
deptName
location
staffNo
staffName

FUNCTIONAL DEPENDENCY DIAGRAM

- Draw boundaries around data items of a determinancy diagram that form elements of table structures.
- The number of determinants (oval with arrows emerging) indicate the number of tables required.
- Foreign keys are ovals with arrows both emerging and converging.

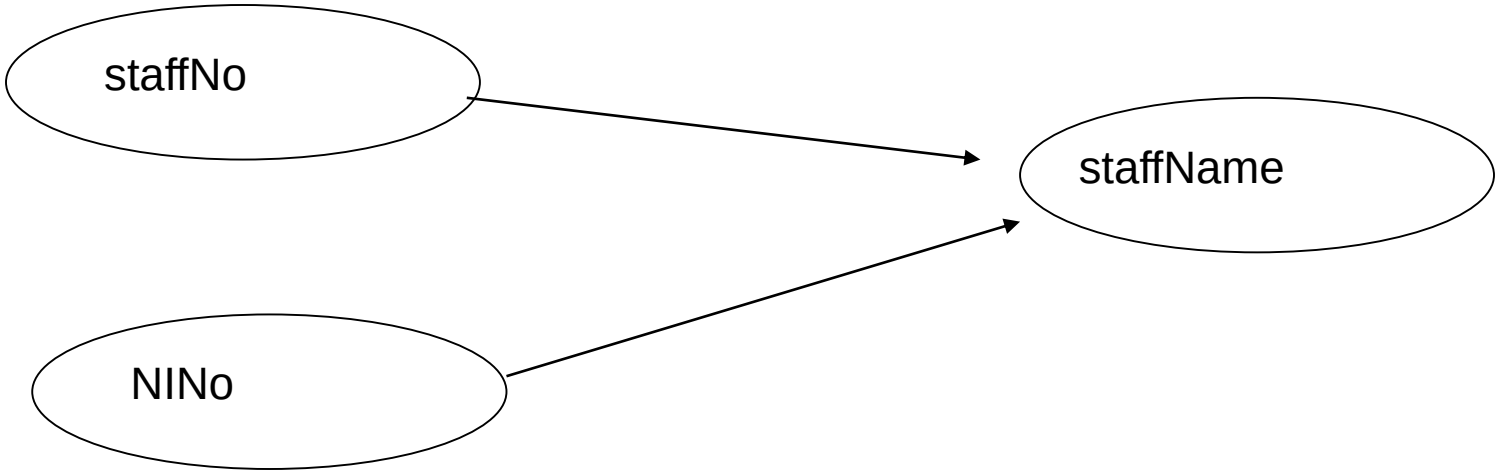
TABLE STRUCTURES

- Applying the above rule gives the following structures in the bracketing notation:
- Staff(staffNo, staffName, deptName…)
- Department(deptName, location, …)

CANDIDATE KEYS

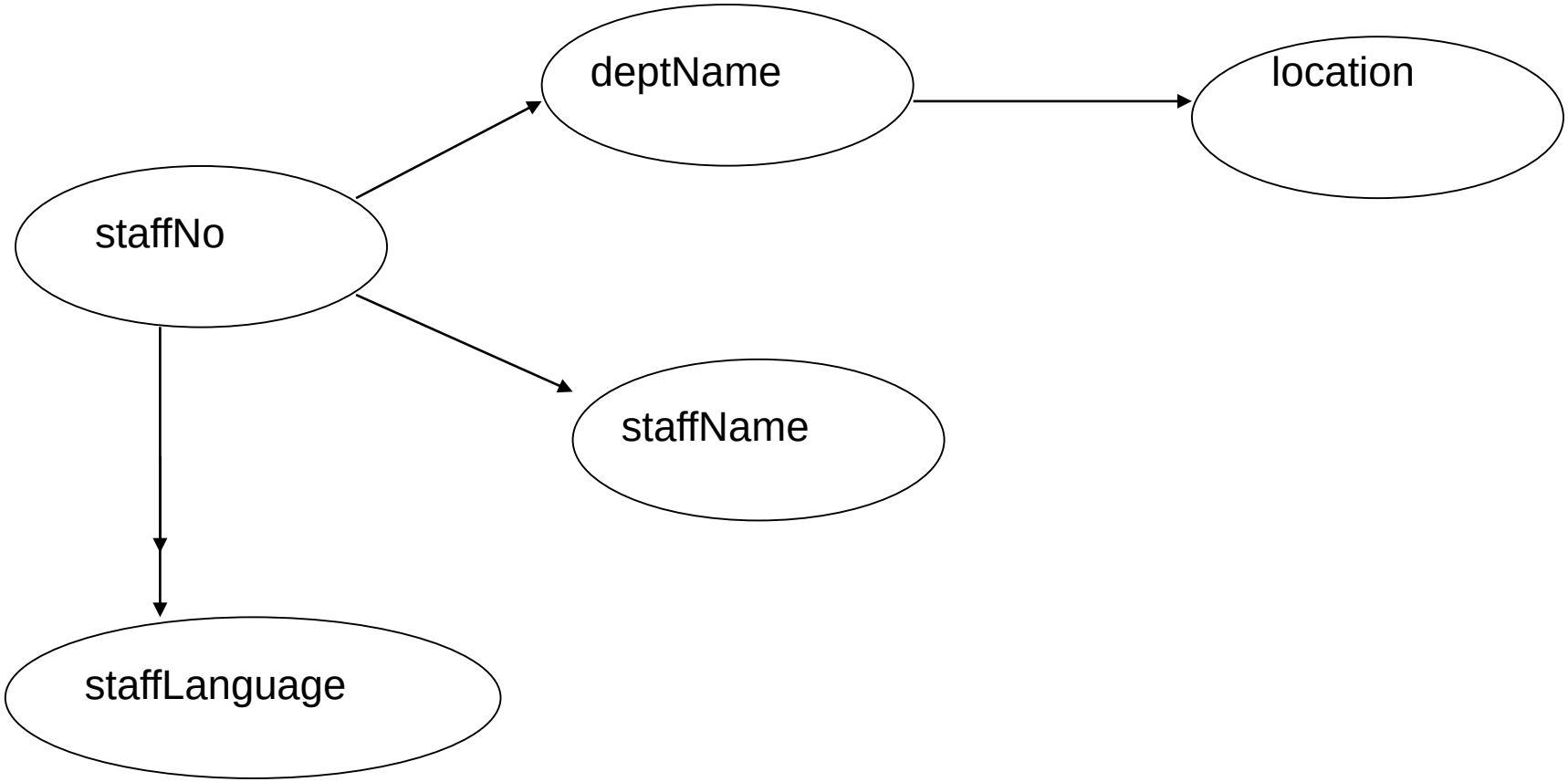
- A determinancy diagram is useful in identifying candidate keys.
- A candidate key is any data item that can act in the capacity of a primary key for a table.
- Represented as competing determinants
- So we choose one to be the primary key and the second one we make it a dependent data items in the table

CANDIDATE KEYS



National Identity Number (or even National Insurance number!) and StaffNo both can be used as Primary key

Accommodating Non-Functional Dependencies



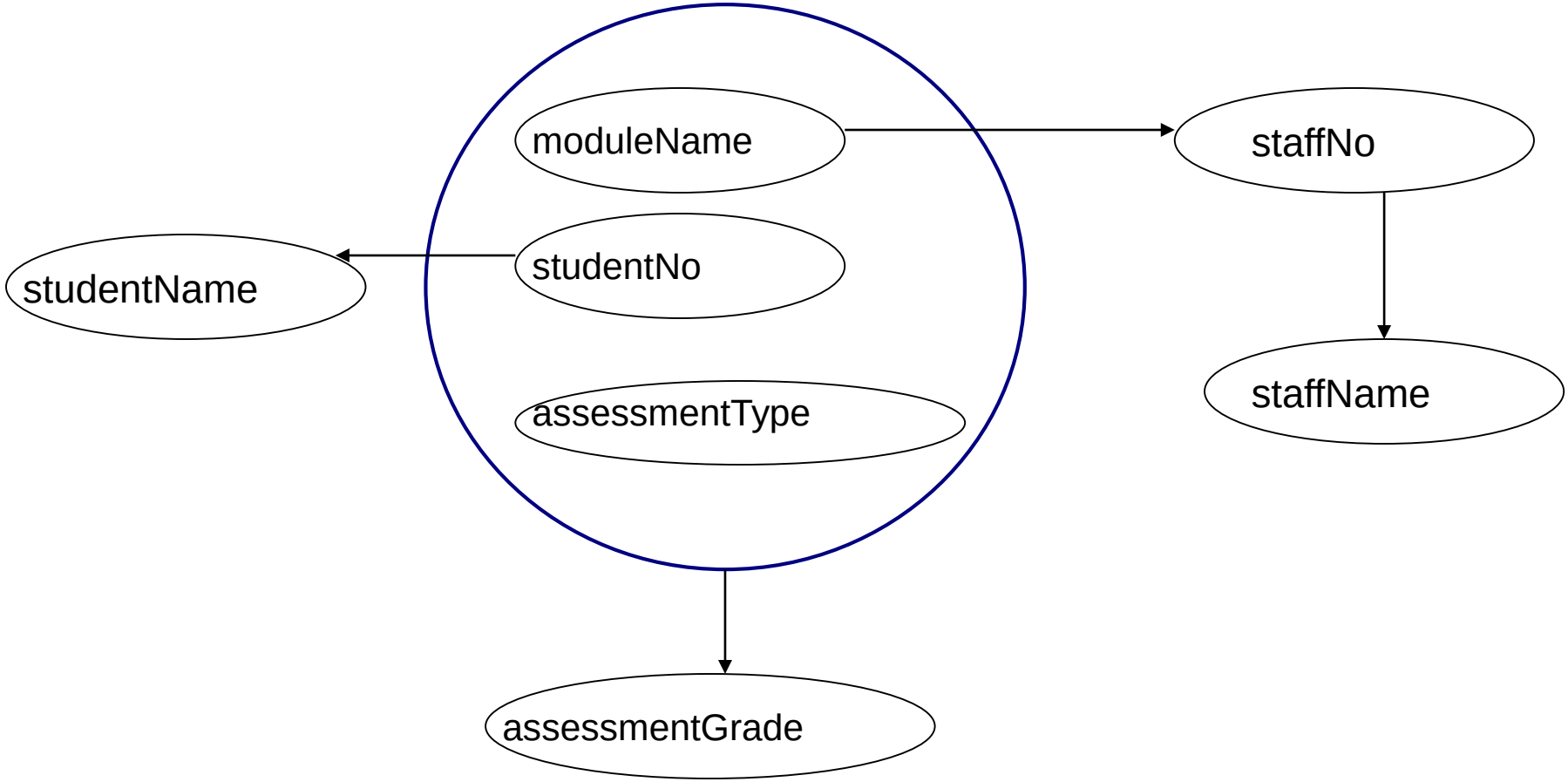
Accommodating Non-Functional Dependencies

- Apply the following rule on:
- staffNo
- deptName
- staffName
- Location
- staffLanguage
- Every non-functional determinant becomes part of the primary key of a table. (Boyce Codd rule)

Accommodating Non-Functional Dependencies

- Every non-functional determinant becomes part of the primary key (make up a compound key from the determinant & dependent data items in a non-functional relationship)
- Need 2 tables to represent the functional dependencies + 1 table to record multivalued dependencies between StaffNo and StaffLanguage.
- Staff(staffNo, staffName, deptName…)
- Department(deptName, location, …)
- Language(staffNo, staffLanguage, …)

Determinancy Diagram for the academic schema



Determinancy Diagrams & the Normal Forms

- First normal form concerns repeating groups.
- Second normal form concern part-key dependencies.
- Third normal form concerns inter-data dependencies

Boyce-Codd Normal Form

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

Boyce-Codd Normal Form

- 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:

Boyce-Codd Normal Form

- Each student may major in several areas.
- A student has one tutor for each area.
- Each area has several tutors but a tutor advises in only one area.
- Each tutor advises several students in an area.
- Draw a determinancy diagram for the above business rule.
- Show the result in Bracketing notation

DATA

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

PROBLEMS

- The above schema is in 3NF because:
- There are no partial dependencies
- No interdata dependencies.
- However, anomalies will still arise when we come to update this relation.
- Update anomaly
- Deletion anomaly
- Insertion anomaly

ANSWER

- Reason for these anomalies is because there are two candidate keys in this problem.
- Solution is to apply BCNF

ANSWER

- Schema1:
- StudentTutors(studentNo, staffNo)
- TutorsAreas(staffNo, area)
- Schema2:
- StudentTutors(studentNo, area)
- TutorsAreas(staffNo, area)

Fourth Normal Form

- We look for tables that contain two or more interdependent multi-valued dependencies
- The latter are fortunately scarcer than part-key or inter-data dependencies.

Example

- Suppose we wish to design a personnel database for the Commission of the European community about employee's skill and spoken languages
- An employee is likely to have several skills (e.g. Typing, word processing operations, spreadsheet) and most of them are required to speak at least two languages.
- Restriction: Employee exercises skill & language use independently i.e. skill not inherently linked to speak a particular language

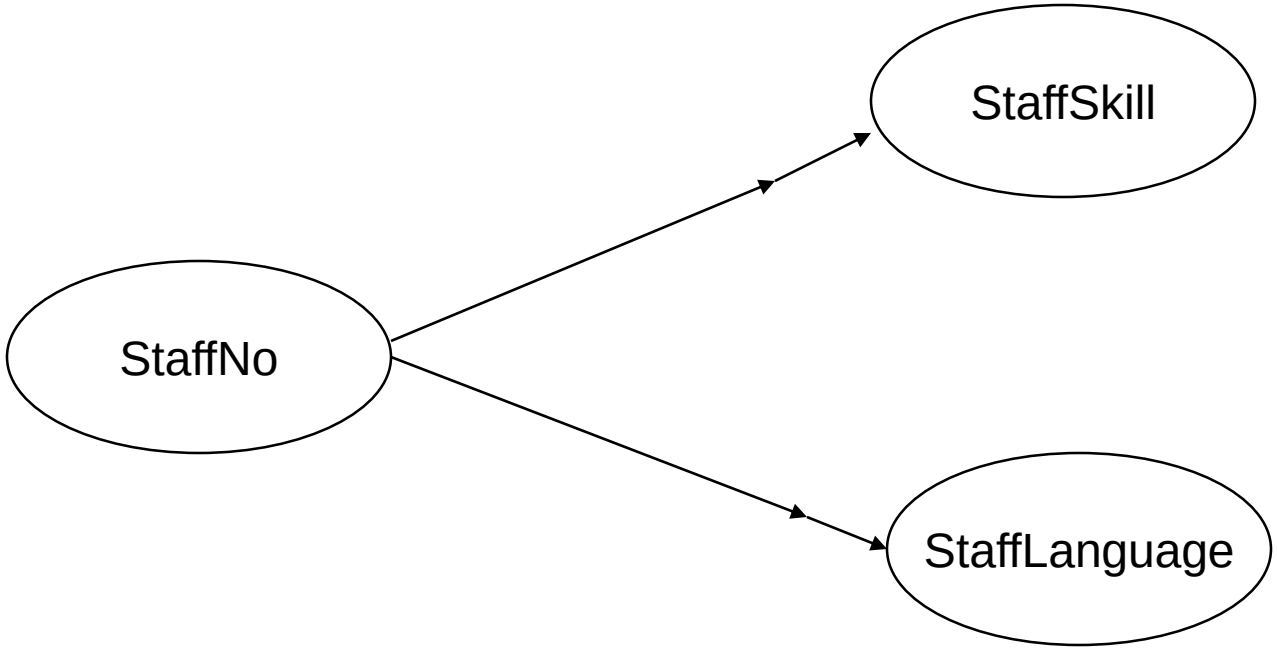
Fourth Normal Form- example

- EUEmployees

EmployeeNo	Skill
122443	Typing
122443	Dictation
221133	Typing
221133	Dictation
332222	Typing

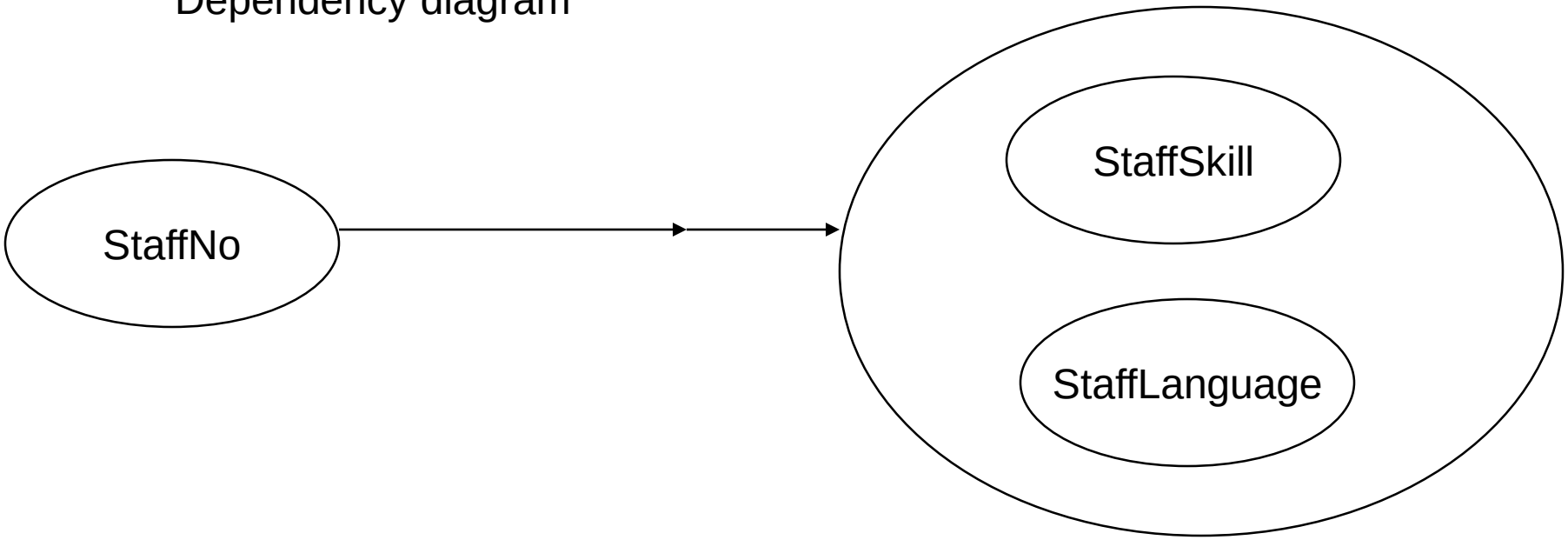
EmployeeNo	Language
122443	English
122443	French
221133	German
221133	French
332222	French

Fourth Normal Forms



Fourth Normal Forms

Dependency diagram



Fifth Normal Form

- A table is in fifth normal form if and only if it is in fourth normal form and it cannot be further non-loss decomposed into a series of smaller tables .(To eliminate all redundancy)

Case Example

Suppose:

1. Agents represent companies. The latter makes products. Agents sell product.

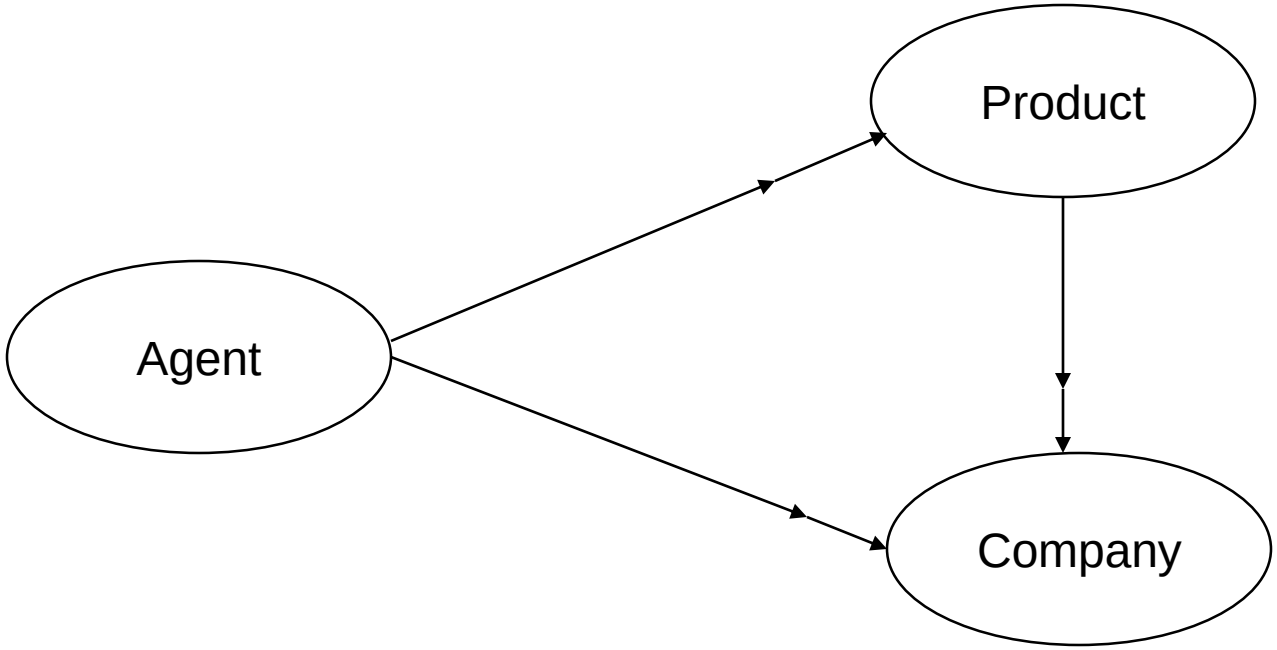
2. And we want to record which agents sells which product for which company

Agent	Company	Automobile
Jones	Ford	Car
Jones	Vauxhall	van
Smith	Ford	van
Smith	Vauxhall	Car

Fifth Normal Form

- We cannot decompose the above structure because, although agent Jones sells cars made by Ford and vans made by Vauxhall, he does not sell Ford vans and Vauxhall cars.
- 5th Normal form concerns interdependent multi-value dependencies, otherwise known as join dependencies.

Fifth Normal Form



Agent	Company
Jones	Ford
Jones	Vauxhall
Smith	Ford
Smith	Vauxhall

Agent	Automobile
Jones	Car
Jones	van
Smith	van
Smith	Car

Automobile	Company
Car	Ford
van	Vauxhall
van	Ford
Car	Vauxhall