

Databases

MicroSoft SQL-server / MySQL-server

Some walk through

Functions For a DBMS #1

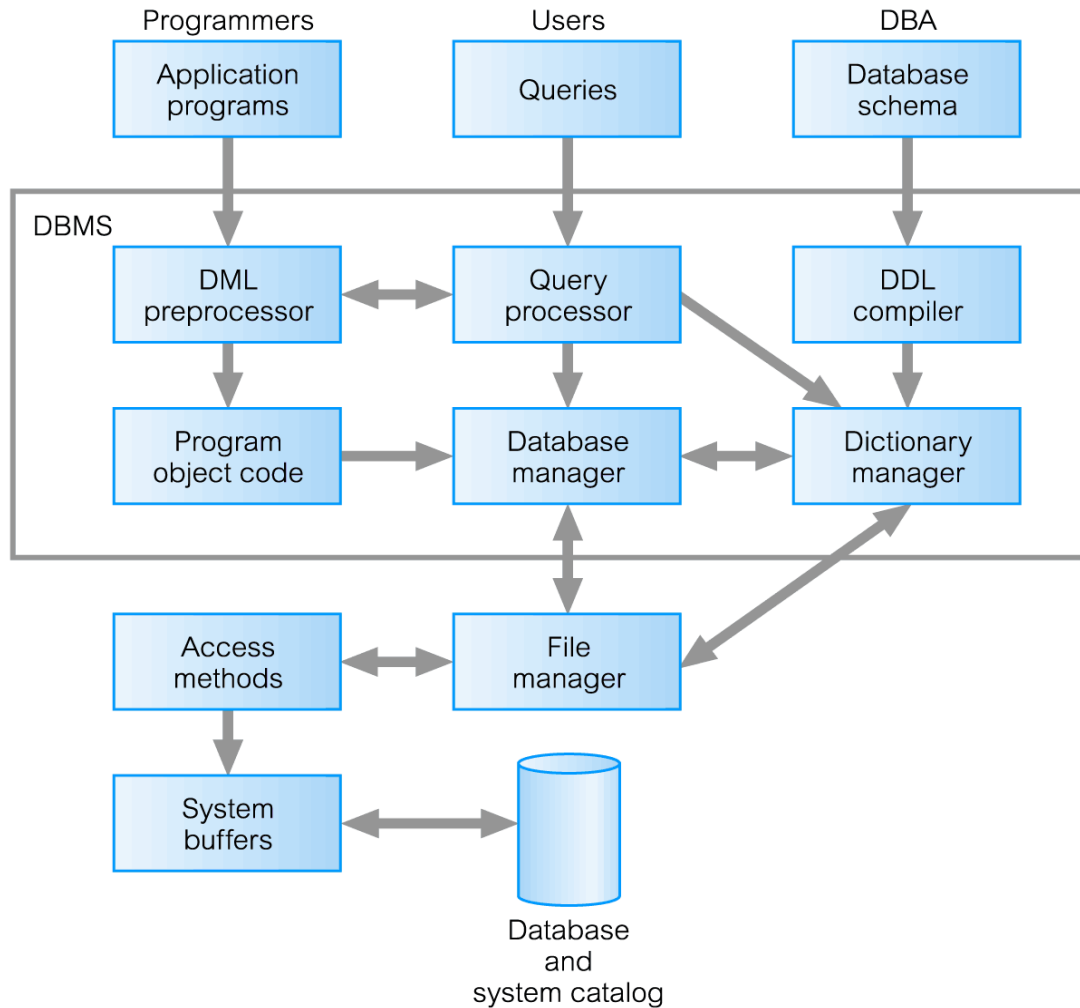
- **Data storage, retrieval and update**
- **A user-accessible catalogue**
- **Transaction support**
- **Concurrency control services**
- **Recovery services**

Functions For a DBMS #2

- **Authorization services**
- **Support for data communication**
- **Integrity services**
- **Services to promote data independence**
- **Utility services**

Database (Server)

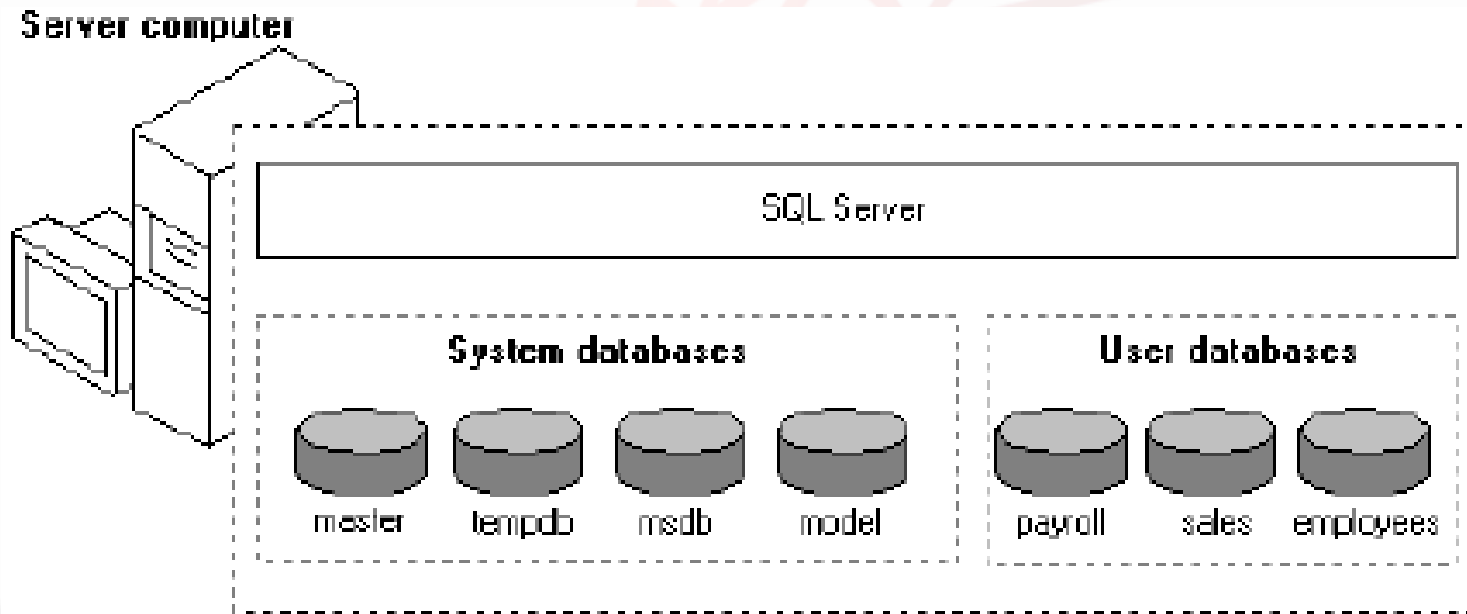
~ not a single system



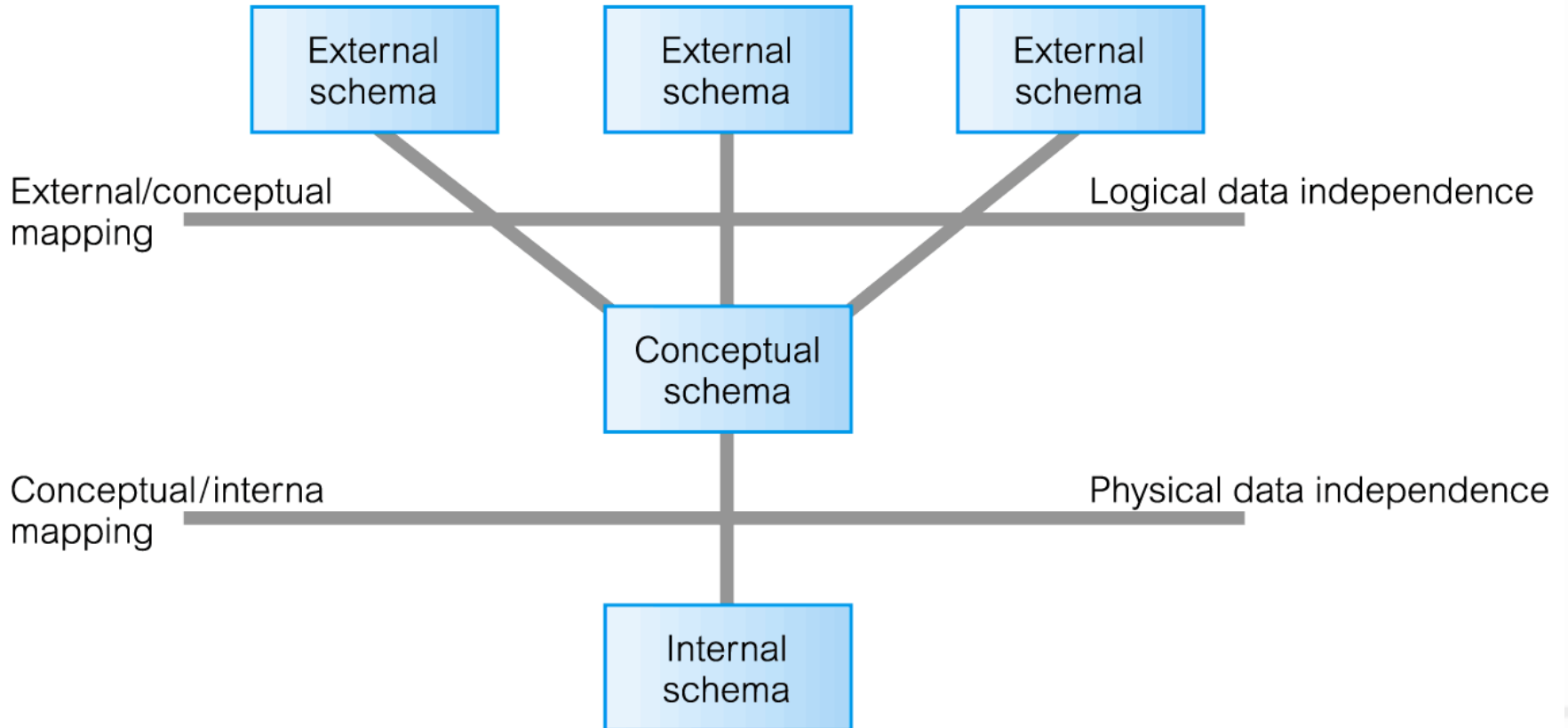
Micr
SQL

© 2012

Databases - system and user

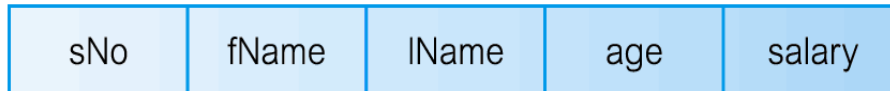


DBMS: ANSI-SPARC

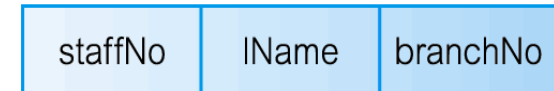


Example

External view 1



External view 2



Conceptual level



Internal level

```
struct STAFF {
    int staffNo;
    int branchNo;
    char fName [15];
    char lName [15];
    struct date dateOfBirth;
    float salary;
    struct STAFF *next;           /* pointer to next Staff record */
};
index staffNo; index branchNo; /* define indexes for staff */
```

Description of the 3 levels #1

- **External Level**
 - **Users' view of the database.**
 - **Describes that part of database that is relevant to a particular user.**
 - **Description in SQL**
- **Conceptual Level**
 - **Community view of the database.**
 - **Describes what data is stored in database and relationships among the data.**
 - **Description in SQL**

Description of the 3 levels #2

- **Internal Level**

- **Physical representation of the database on the computer.**
- **Describes how the data is stored in the database.**
- **Description in some low level language (C, Assembler)**

Terminology of the relational model (logical)

Relation A relation is a table with columns and rows.

Attribute An attribute is a named column of a relation.

Domain A domain is the set of allowable values for one or more attributes.

Tuple A tuple is a row of a relation.

Degree The degree of a relation is the number of attributes it contains.

Cardinality The cardinality of a relation is the number of tuples it contains.

Relational database A collection of normalized relations with distinct relation names.

Example of relational terms used

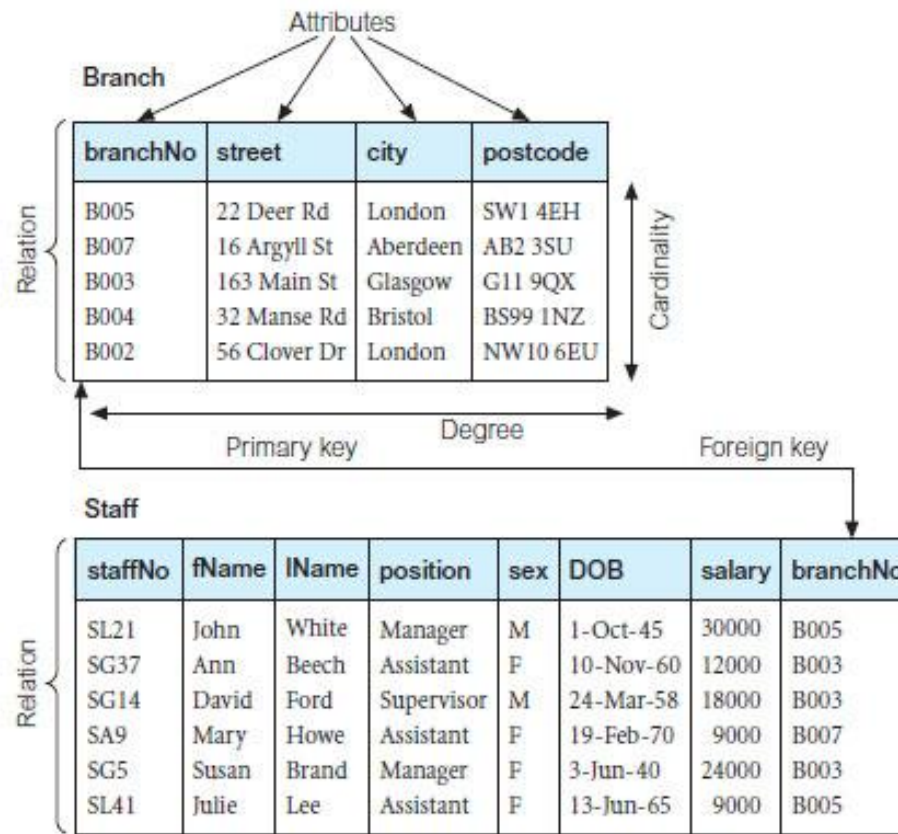


Figure 3.1
Instances of the Branch and Staff relations.

Attribute	Domain Name	Meaning	Domain Definition
branchNo	BranchNumbers	The set of all possible branch numbers	character: size 4, range B001–B999
street	StreetNames	The set of all street names in Britain	character: size 25
city	CityNames	The set of all city names in Britain	character: size 15
postcode	Postcodes	The set of all postcodes in Britain	character: size 8
sex	Sex	The sex of a person	character: size 1, value M or F
DOB	DatesOfBirth	Possible values of staff birth dates	date, range from 1-Jan-20, format dd-mmm-yy
salary	Salaries	Possible values of staff salaries	monetary: 7 digits, range 6000.00–40000.00

Figure 3.2
Domains for some attributes of the Branch and Staff relations.

Anatomy of a relational database (physical), I

- From **entity** to **table**
- From **tuple** to **record or Row**
- From **attribute** to **field or Column**
- **Keys** are fields with specific roles:
 - **Primary key**: uniquely identifies each record in a table
 - **Foreign key**: points from a field in a table to the primary key of another table
- See “SQL queries for mere mortals” page 6+7+9 figure 1-1, 1-2, 1-3

Anatomy of a relational database (physical), II

- Tables can be combined together - this is called **relationships**:
 - **One-to-one** (aka 1:1)
 - **One-to-many** (aka 1:N)
 - **Many-to-many** (aka M:N)
- See “SQL queries for mere mortals” page 12-14 figure 1-5 – 1-8

ER modelling

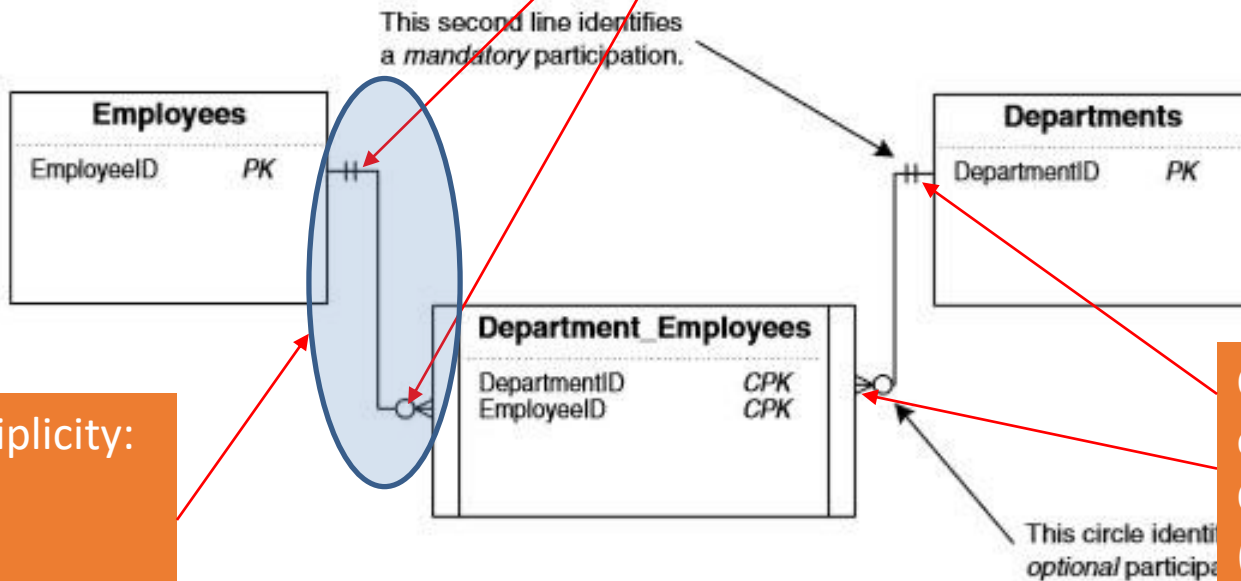
- ER diagrams shows entities and their relationships
- In this course we are going to use the syntax called ***Crow's foot***
- Each box represents an ***entity***
- Lines connecting the entities are called ***relationships***

Multiplicity

Away from entity:
Participation
(minimum)

Close to entity:
Cardinality
(maximum)

Multiplicity:
1:1
1:N
M:N



20 Diagramming the type of participation for the *Departments* and *Department_Employees* tables

Participation in ER diagrams

- The type of **participation** assigned to a given table determines, whether a record must exist in that table, before you can enter a record into the other table
- Two types of participation:
 - **Mandatory:** at least one record must exist
 - **Optional:** no requirements
- See “SQL queries for mere mortals” page 47+48 figure 2-19 + 2-20

Alternative ER notations

- Two different notations:
 - Chen
 - UML
- The examples are copy/pasted from “*Database Systems: A Practical Approach to Design, Implementation, and Management*”

ER diagram Chen notation

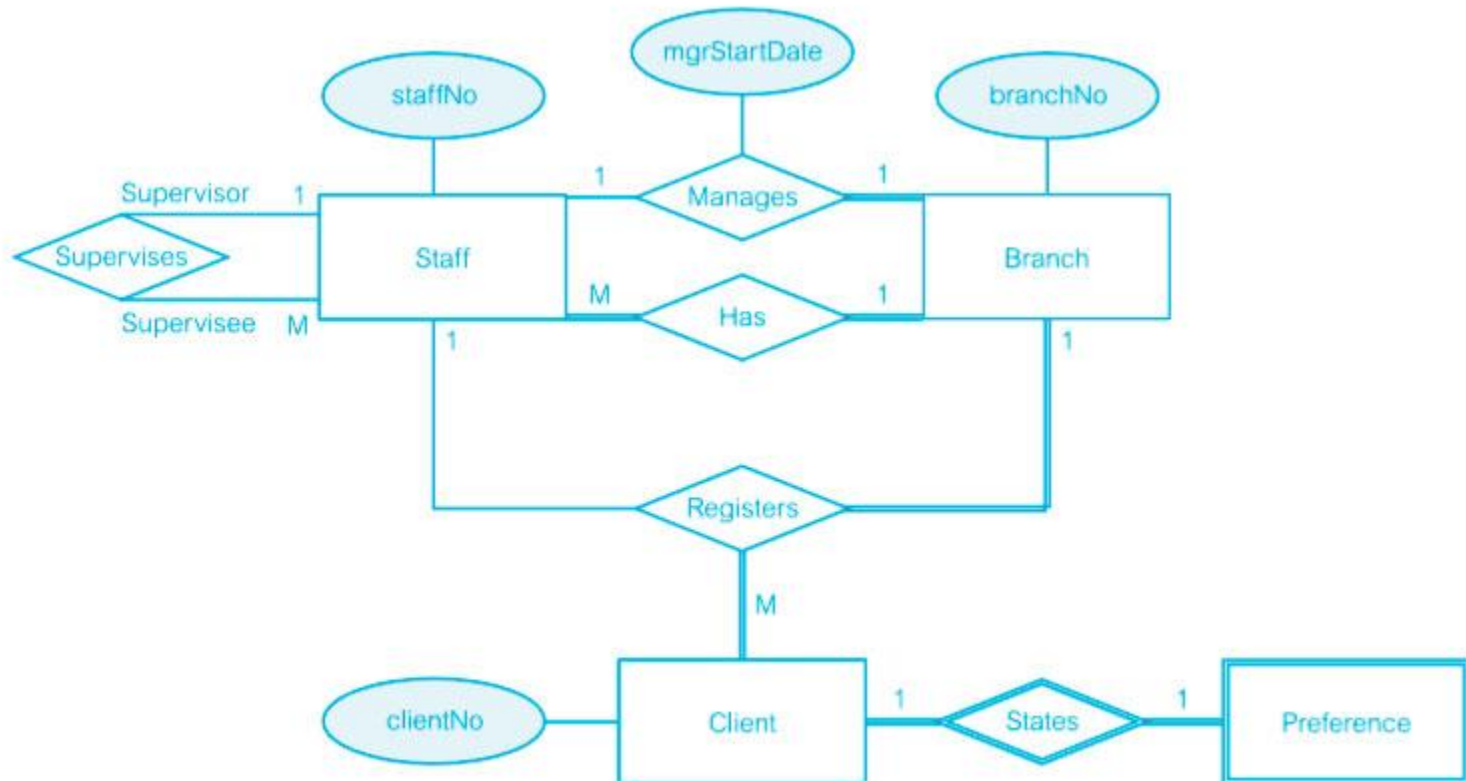


Figure F.1 Part of the ER diagram shown in Figure 11.1 redrawn using the Chen notation.

ER diagram UML notation

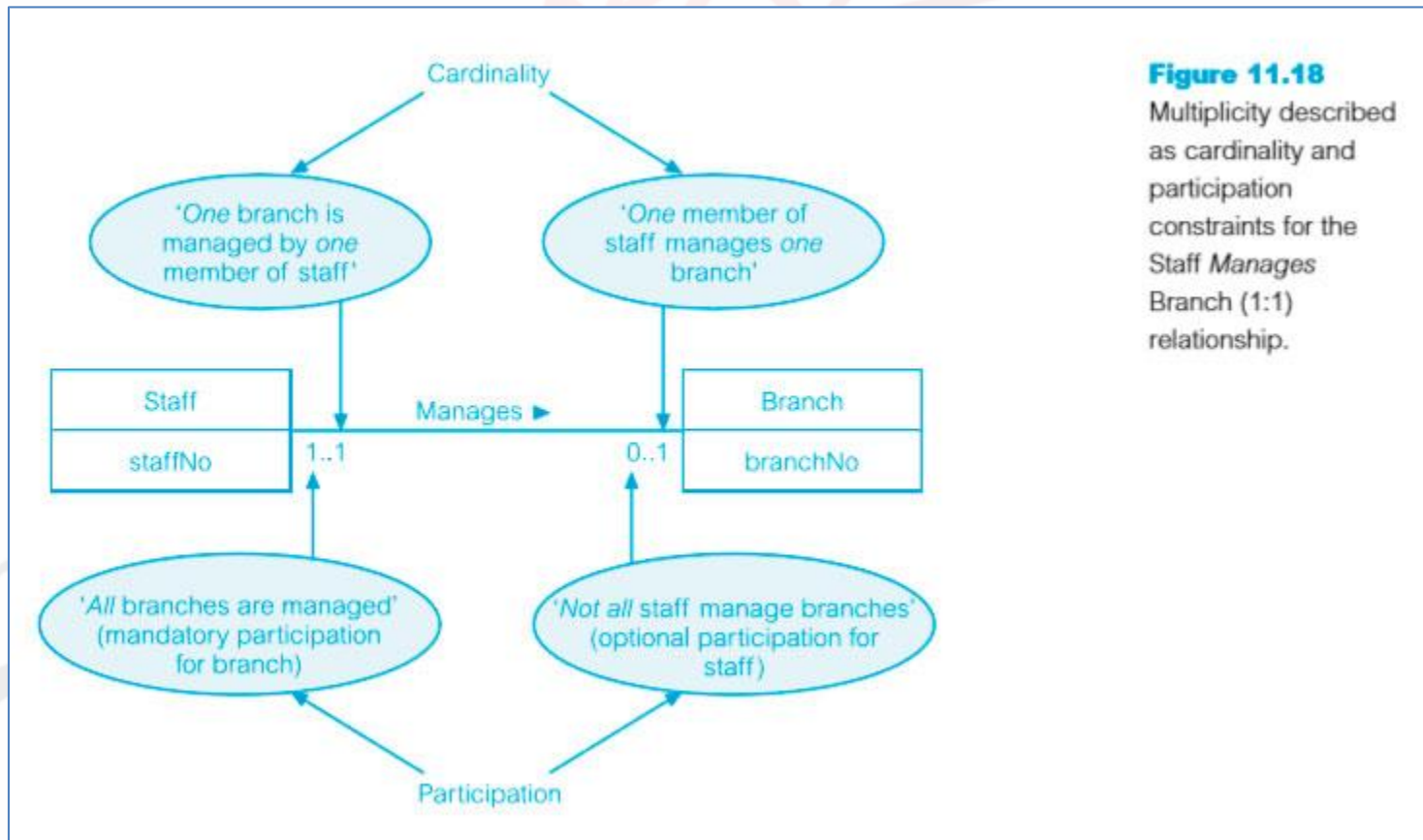


Figure 11.18
Multiplicity described as cardinality and participation constraints for the Staff *Manages* Branch (1:1) relationship.