Chapter 4

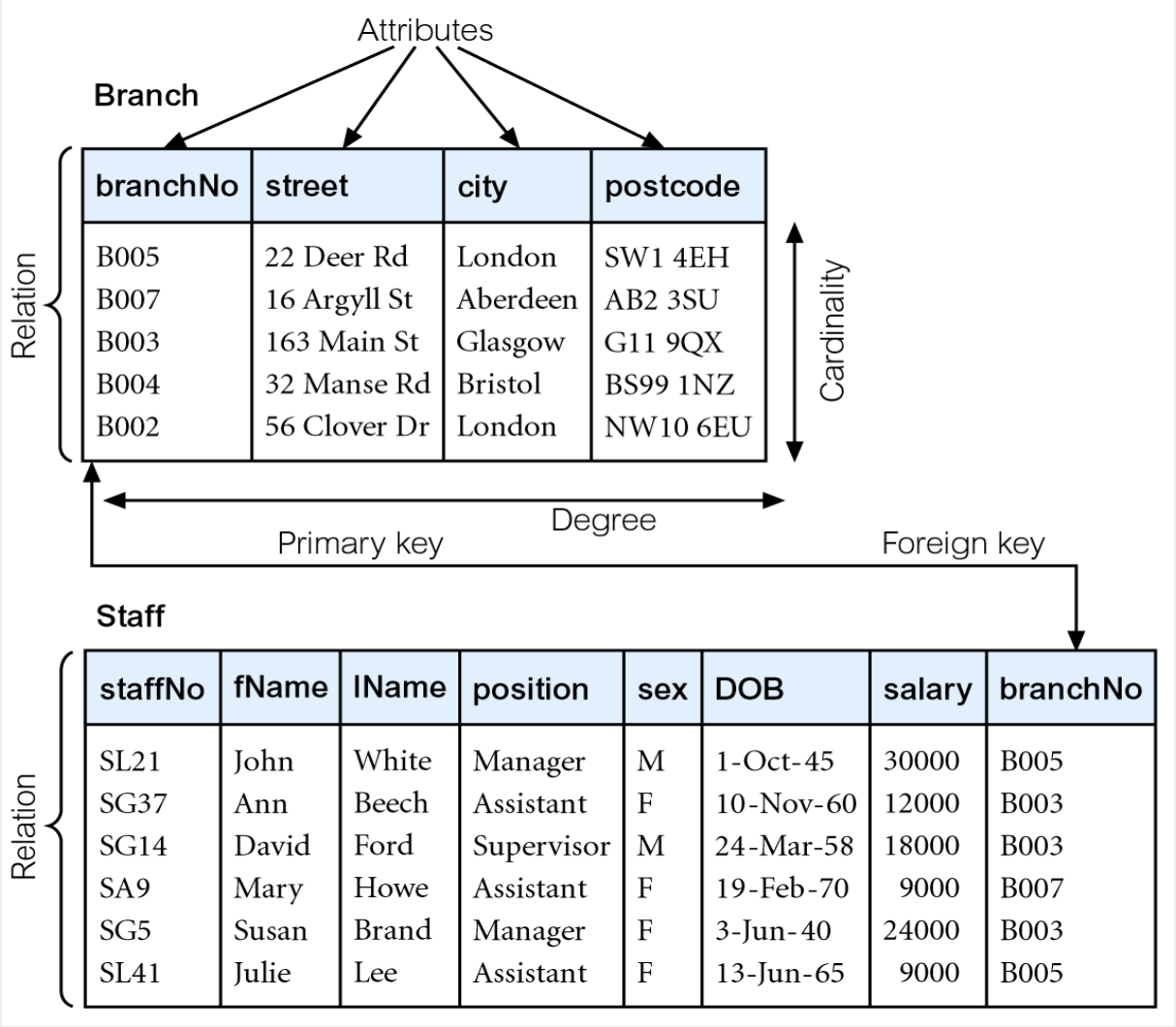
The Relational Model

Overview

I. This chapter covers the relational model, which provides a formal description of the structure of a database

II. The next chapter covers the relational algebra and calculus, which provides a formal basis for the query language for the database

Instances of Branch and Staff Relations



Relational Model Terminology

I. A relation is a table with columns and rows.

   A. Only applies to logical structure of the database, not the physical structure.

II. An attribute is a named column of a relation.

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

Relational Model Terminology

IV. A **tuple** is a row of a relation.

V. The **degree** is the number of attributes in a relation.

VI. The **cardinality** is the number of tuples in a relation.

VII. A **Relational Database** is a collection of normalized relations with distinct relation names.

- Typically means that the only shared columns among relations are foreign keys (eliminates redundancy)

Examples of Attribute Domains

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

Alternative Terminology for Relational Model

Formal terms	Alternative 1	Alternative 2
Relation	Table	File
Tuple	Row	Record
Attribute	Column	Field

Mathematical Definition of Relation

I. Consider two sets, D_1 & D_2, where $D_1 = \{2, 4\}$ and $D_2 = \{1, 3, 5\}$.

II. Cartesian product, $D_1 \times D_2$, is set of all ordered pairs, where first element is member of D_1 and second element is member of D_2.

$$D_1 \times D_2 = \{(2, 1), (2, 3), (2, 5), (4, 1), (4, 3), (4, 5)\}$$

A. Alternative way is to find all combinations of elements with first from D_1 and second from D_2.

Mathematical Definition of Relation

III. **Any subset of Cartesian product is a relation; e.g.**

$$R = \{(2, 1), (4, 3)\}$$

IV. **May specify which pairs are in relation using some condition for selection; e.g.**

A. **second element is 1:**

$$R = \{(x, y) \mid x \in D_1, y \in D_2, \text{ and } y = 1\}$$

1. $R = \{(2, 1), (4, 1)\}$

B. **first element is always twice the second:**

$$S = \{(x, y) \mid x \in D_1, y \in D_2, \text{ and } x = 2y\}$$

1. $S = \{(2,1)\}$

Mathematical Definition of Relation

V. **Consider three sets D_1, D_2, D_3 with Cartesian Product $D_1 \times D_2 \times D_3$; e.g.**

$$D_1 = \{1, 3\} \quad D_2 = \{2, 4\} \quad D_3 = \{5, 6\}$$

$$D_1 \times D_2 \times D_3 = \{(1,2,5), (1,2,6), (1,4,5), (1,4,6), (3,2,5), (3,2,6), (3,4,5), (3,4,6)\}$$

VI. **Any subset of these ordered triples is a relation.**

Mathematical Definition of Relation

VII. **Cartesian product of n sets $(D_1, D_2, \ldots, D_n)$ is:**

$$D_1 \times D_2 \times \ldots \times D_n$$

$$= \{(d_1, d_2, \ldots, d_n) \mid d_1 \in D_1, d_2 \in D_2, \ldots, d_n \in D_n\}$$

usually written as:

$$\prod_{i=1}^{n} D_i$$

VIII. **Any set of n-tuples from this Cartesian product is a relation on the n sets.**

Database Relations

I. **Relation schema**

   A. **Named relation defined by a set of attribute and domain name pairs.**

II. **Relational database schema**

   A. **Set of relation schemas, each with a distinct name.**

Properties of Relations

I. **Relation name is distinct from all other relation names in relational schema.**

II. **Each cell of relation contains exactly one atomic (single) value (e.g., a phone field may not contain multiple phone numbers)**

III. **Each attribute has a distinct name.**

IV. **Values of an attribute are all from the same domain.**

Properties of Relations

V. **Each tuple is distinct; there are no duplicate tuples.**

VI. **Order of attributes has no significance.**

A. **This differs from a mathematical relation in which the order of values matters (e.g., the tuple (2,1) is different than the tuple (1,2))**

Properties of Relations

VII. **Order of tuples has no significance, theoretically.**

A. **But practically the order may affect query optimization**

B. **Often we will order the relation on one or more attributes**

Relational Keys

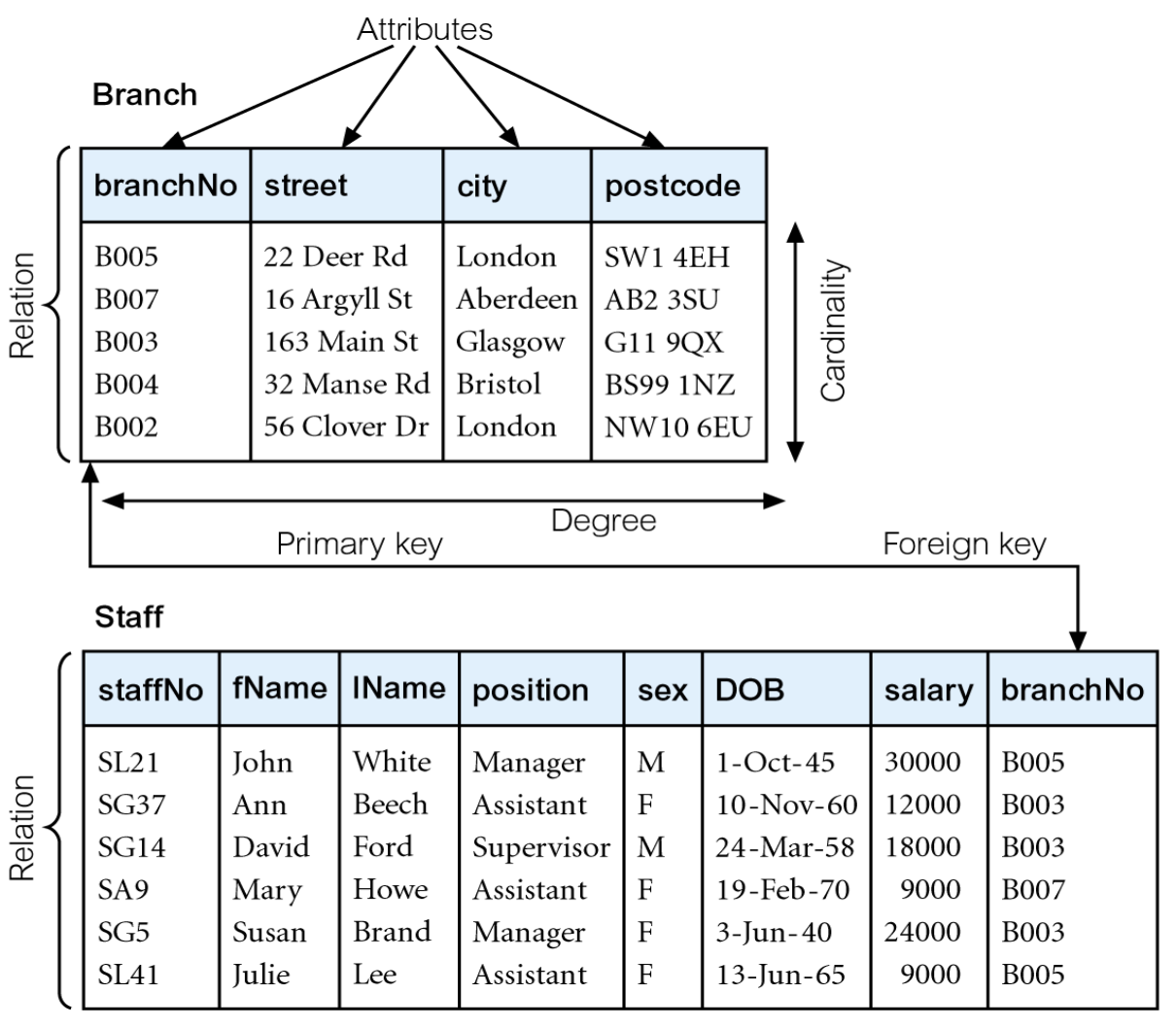
I. **Superkey:** An attribute, or set of attributes, that uniquely identifies a tuple within a relation.

II. **Candidate Key:** Superkey (K) such that no proper subset is a superkey within the relation.

   A. In each tuple of R, values of K uniquely identify that tuple (uniqueness).

   B. No proper subset of K has the uniqueness property (irreducibility).

Instances of Branch and Staff Relations



Relational Keys

III. **Primary Key:** Candidate key selected to identify tuples uniquely within relation.

IV. **Alternate Keys:** Candidate keys that are not selected to be primary key.

V. **Foreign Key:** Attribute, or set of attributes, within one relation that matches candidate key of some (possibly same) relation.

   A. Foreign key typically denotes a relationship with another relation

   B. Example: The BranchNo attribute in the Staff relation is a foreign key that denotes that a staff member belongs to a branch

The Hotel Schema

Hotel (hotelNo, hotelName, city)

Room (roomNo, hotelNo, type, price)

Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo)

Guest (guestNo, guestName, guestAddress)

Integrity Constraints

I. **Null**

A. **Represents value for an attribute that is currently unknown or not applicable for tuple.**

B. **Deals with incomplete or exceptional data.**

C. **Represents the absence of a value and is not the same as zero or spaces, which are values.**

D. **Example: In the Viewing relation, the Comment attribute might be Null if the client has not left a comment about that property**

E. **The presence of a large number of Nulls usually suggests that the relation should be decomposed into one or more subrelations-more on this later in the course**

Integrity Constraints

II. **Entity Integrity: In a base relation, no attribute of a primary key can be null.**

III. **Referential Integrity: If a foreign key exists in a relation, either the foreign key value must match a candidate key value of some tuple in its parent relation or the foreign key value must be wholly null.**

Integrity Constraints

IV. **General Constraints:** **Additional rules specified by users or database administrators that define or constrain some aspect of the enterprise.**

A. **Example: No staff member may have a salary that exceeds $100,000.00**

Views

I. **Base Relation**: A relation whose tuples are physically stored in database.

II. **View**: A dynamically derived relation created from a query on one or more base relations

A. Alternative names

1. derived relation
2. virtual relation

Views

B. A virtual relation that does not necessarily actually exist in the database but is produced upon request, at time of request.

C. Contents of a view are defined as a query on one or more base relations.

D. Views are dynamic, meaning that changes made to base relations that affect view attributes are immediately reflected in the view.

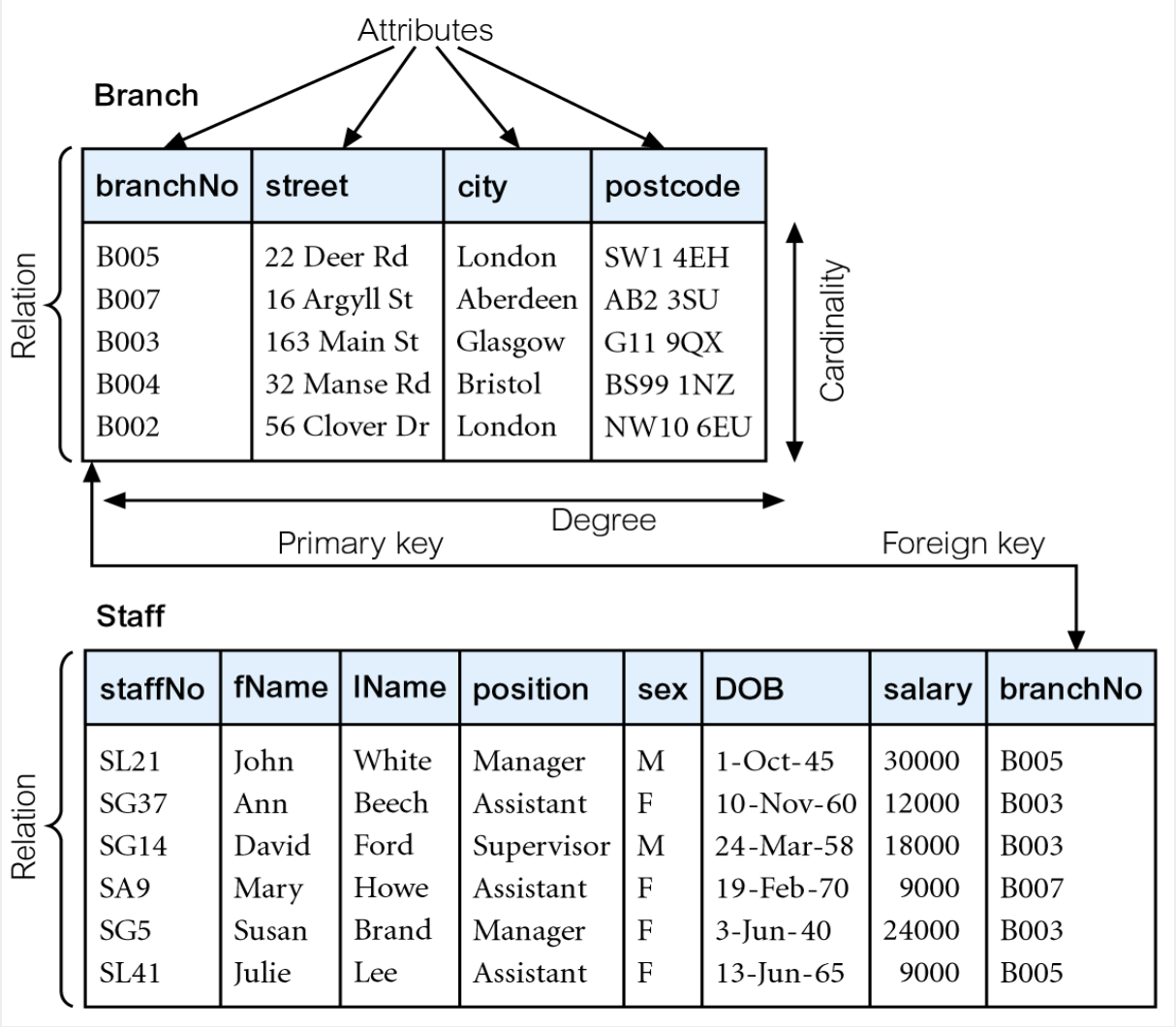
Purpose of Views

I. **Provides powerful and flexible security mechanism by hiding parts of database from certain users.**

II. **Permits users to access data in a customized way, so that same data can be seen by different users in different ways, at same time.**

III. **Can simplify complex operations on base relations.**

Instances of Branch and Staff Relations



Updating Views

I. **All updates to a base relation should be immediately reflected in all views that reference that base relation.**

II. **If view is updated, underlying base relation should reflect change.**

Updating Views

I. There are restrictions on types of modifications that can be made through views:

A. Updates are allowed if query involves a single base relation and contains a candidate key of base relation.

B. Updates are not allowed when the view is created from multiple base relations.

C. Updates are not allowed when the view's columns were formed from aggregation or grouping operations.