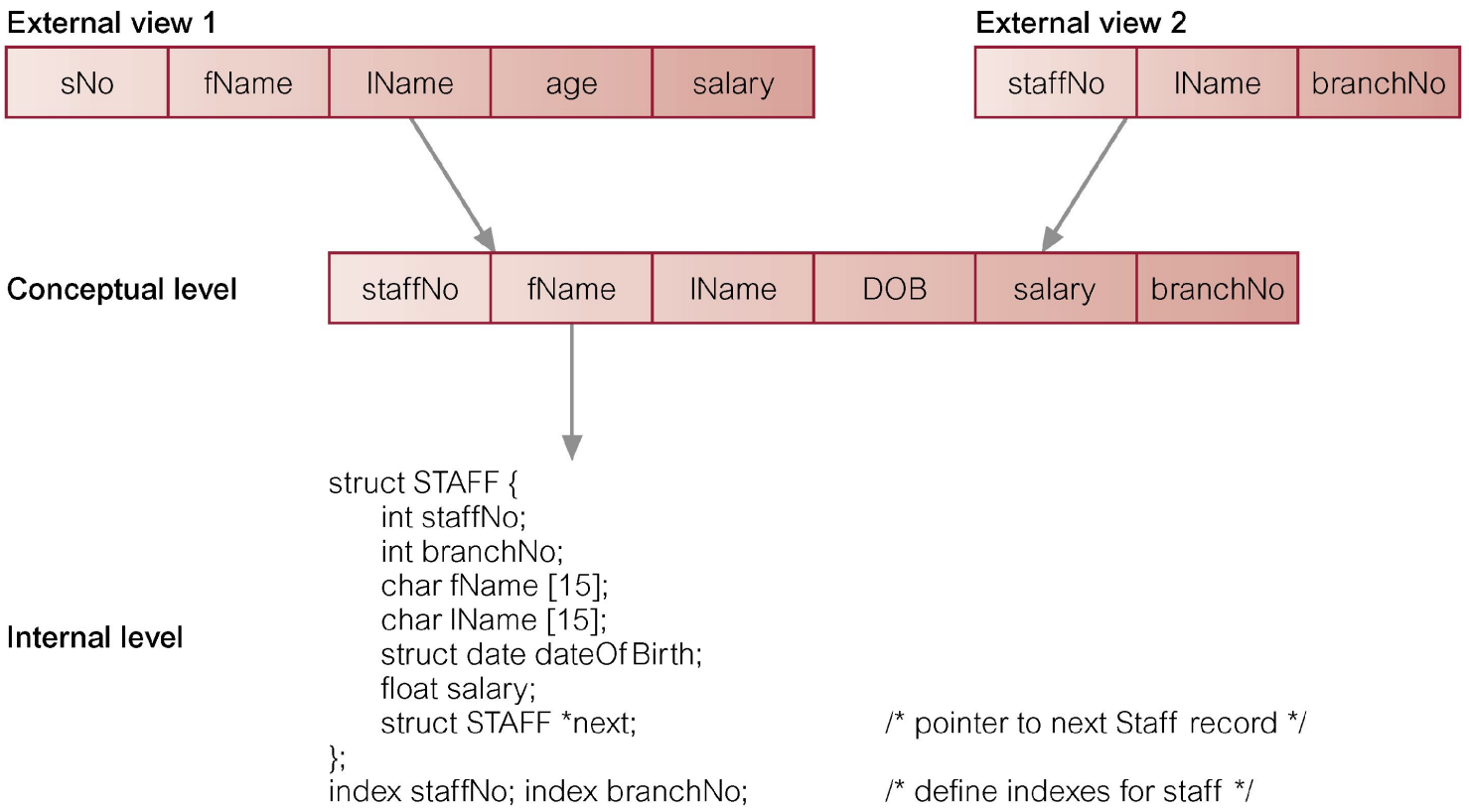
Differences between Three Levels of ANSI-SPARC Architecture



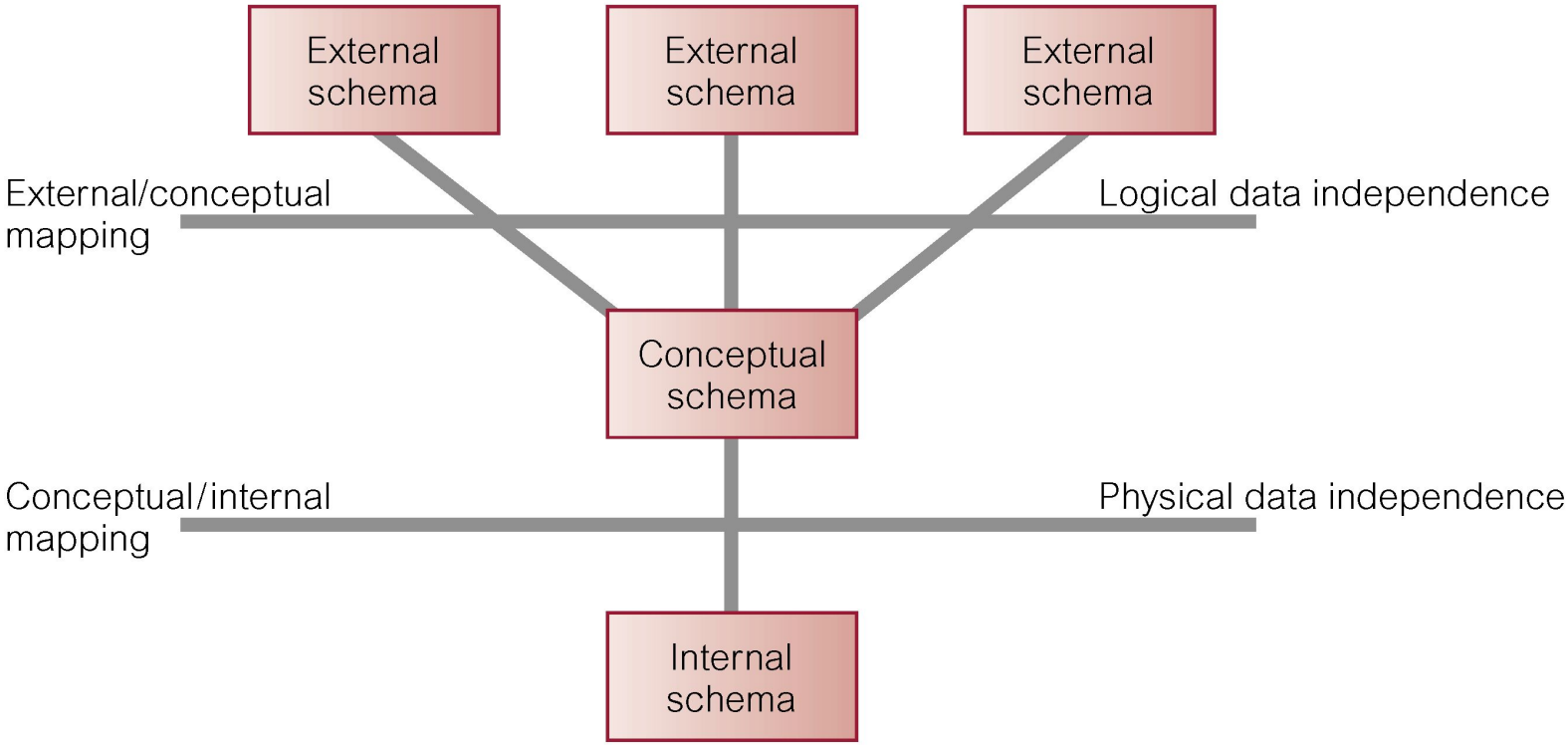
Data Independence

- Logical Data Independence
  - Refers to immunity of external schemas to changes in conceptual schema.
  - Conceptual schema changes (e.g. addition/removal of entities).
  - Should not require changes to external schema or rewrites of application programs.

Data Independence

- Physical Data Independence
  - Refers to immunity of conceptual schema to changes in the internal schema.
  - Internal schema changes (e.g. using different file organizations, storage structures/devices).
  - Should not require change to conceptual or external schemas.

Data Independence and the ANSI-SPARC Three-level Architecture



Data Model

- Data Model comprises:
  - A structural part
  - A manipulative part
  - Possibly a set of integrity rules

Data Model

- **Purpose**
  - **To represent data in an understandable way.**

Relational Systems

- Then, in 1970,E. F. Codd wrote “A Relational Model of Data for Large Shared Databanks” and introduced the relational model
- Information is stored as *tuples* or *records* in *relations* or *tables*
- Most modern DBMS are based on the relational model
- The relational model covers 3 areas:
  - Data structure
  - Data manipulation
  - Data integrity

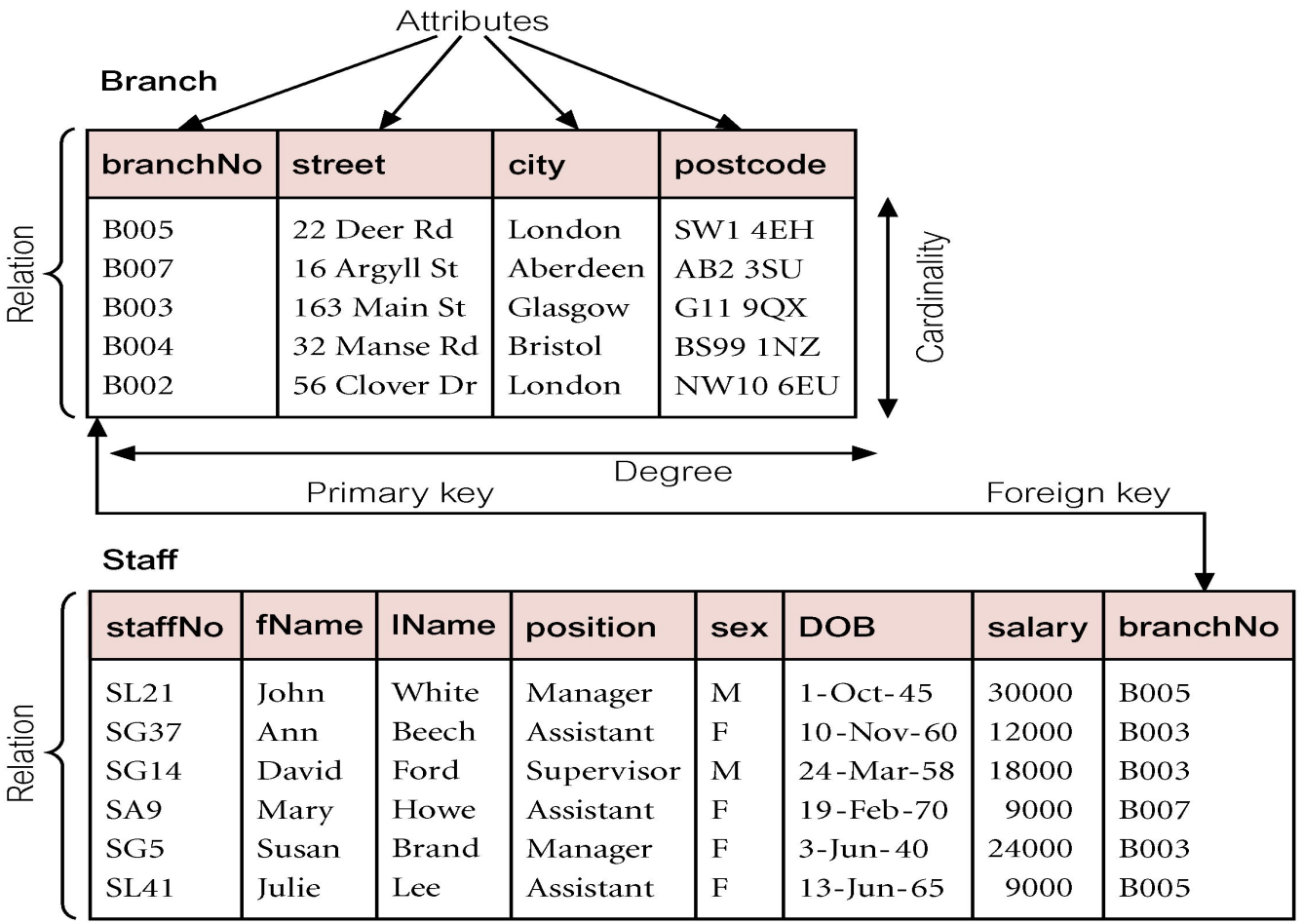
Relational Model Terminology

- A relation is a table with columns and rows.
- Attribute is a named column of a relation.
- Domain is the set of allowable values for one or more attributes.

Relational Model Terminology

- Tuple is a row of a relation.
- Degree is the number of attributes in a relation.
- Cardinality is the number of tuples in a relation.
- Relational Database is a collection of relations with distinct relation names.

Instances of Branch and Staff Relations



Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

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

Table 3.1 Alternative terminology for relational model terms.

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

Properties of Relations

- Relation name is distinct from all other relation names in relational schema.
- Each cell of relation contains exactly one atomic (single) value.
- Each attribute has a distinct name.
- Values of an attribute are all from the same domain.

Properties of Relations

- Each tuple is distinct; there are no duplicate tuples.
- Order of attributes has no significance.
- Order of tuples has no significance.

Candidate Keys

- A set of attributes in a relation is called a candidate key if, and only if,
  - Every tuple has a unique value for the set of attributes (*uniqueness*)
  - No proper subset of the set has the uniqueness property (*minimality*)

ID	First	Last
S139	John	Smith
S140	Mary	Jones
S141	John	Brown
S142	Jane	Smith

Candidate key: {ID}; {First,Last} looks reasonable but we may get people with the same name

{ID, First}, {ID, Last} and {ID, First, Last} satisfy uniqueness, but are not minimal

{First} and {Last} do not give a unique identifier for each row

Choosing Candidate Keys

- Important: don’t look just on the data in the table to determine what is a candidate key
- The table may contain just one tuple, so anything would do!
- Use knowledge of the real world – what is going to stay unique!

Primary Keys

- One Candidate Key is usually chosen to be used to identify tuples in a relation
- This is called the *Primary Key*
- Often a special ID attribute is used as the Primary Key

NULLs and Primary Keys

- Missing information can be represented using NULLs
- A NULL indicates a missing or unknown value
- More on this later...

- *Entity Integrity*: Primary Keys cannot contain NULL values

Foreign Keys

- *Foreign Keys* are used to link data in two relations. A set of attributes in the first (*referencing*) relation is a Foreign Key if its value always either
  - Matches a Candidate Key value in the second (*referenced*) relation, or
  - Is wholly NULL
- This is called *Referential Integrity*

Foreign Keys - Example

Department

DID	DName
13	Marketing
14	Accounts
15	Personnel

{DID} is a Candidate Key for Department - Each entry has a unique value for DID

Employee

EID	EName	DID	
15	John Smith	13	
16	Mary Brown	14	
17	Mark Jones	13	
18	Jane Smith	NULL	

{DID} is a Foreign Key in Employee - each Employee's DID value is either NULL, or matches an entry in the Department relation. This links each Employee to (at most) one Department

Foreign Keys - Example

Employee

ID	Name	Manager
E1496	John Smith	E1499
E1497	Mary Brown	E1498
E1498	Mark Jones	E1499
E1499	Jane Smith	NULL

{ID} is a Candidate Key for Employee, and {Manager} is a Foreign Key, which refers to the same relation - every tuple's Manager value is either NULL or matches an ID value

Referential Integrity

- When relations are updated, referential integrity can be violated
- This usually occurs when a referenced tuple is updated or deleted
- There are a number of options:
  - RESTRICT - stop the user from doing it
  - CASCADE - let the changes flow on
  - NULLIFY - make values NULL

Referential Integrity - Example

- What happens if
  - Marketing's DID is changed to 16 in Department?
  - The entry for Accounts is deleted from Department?

Department

DID	DName
13	Marketing
14	Accounts
15	Personnel

Employee

EID	EName	DID	
15	John Smith	13	
16	Mary Brown	14	
17	Mark Jones	13	
18	Jane Smith	NULL	

RESTRICT

- RESTRICT stops any action that violates integrity
  - You cannot update or delete Marketing or Accounts
  - You *can* change Personnel as it is not referenced

Department

DID	DName
13	Marketing
14	Accounts
15	Personnel

Employee

EID	EName	DID	
15	John Smith	13	
16	Mary Brown	14	
17	Mark Jones	13	
18	Jane Smith	NULL	

CASCADE

- CASCADE allows the changes made to flow through
  - If Marketing's DID is changed to 16 in Department, then the DIDs for John Smith and Mark Jones also change
  - If Accounts is deleted then so is Mary Brown

Department

DID	DName
13 → 16	Marketing
14	Accounts
15	Personnel

Employee

EID	EName	DID	
15	John Smith	13	16
16	Mary Brown	14	
17	Mark Jones	13	16
18	Jane Smith	NULL	

NULLIFY

- NULLIFY sets problem values to NULL
  - If Marketing's DID changes then John Smith's and Mark Jones' DIDs are set to NULL
  - If Accounts is deleted, Mary Brown's DID becomes NULL

Department

DID	DName
~~13~~ 16	Marketing
~~14~~	~~Accounts~~
15	Personnel

Employee

EID	EName	DID	
15	John Smith	13	NULL
16	Mary Brown	14	NULL
17	Mark Jones	13	NULL
18	Jane Smith	NULL	

Views

- Base Relation
  - Named relation corresponding to an entity in conceptual schema, whose tuples are physically stored in database.

- View
  - Dynamic result of one or more relational operations operating on base relations to produce another relation.

Views

- A virtual relation that does not necessarily actually exist in the database but is produced upon request, at time of request.
- Contents of a view are defined as a query on one or more base relations.
- Views are dynamic, meaning that changes made to base relations that affect view attributes are immediately reflected in the view.

Purpose of Views

- Provides powerful and flexible security mechanism by hiding parts of database from certain users.
- 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.
- Can simplify complex operations on base relations.

Updating Views

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

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