INTRODUCTION TO DATABASES

KAL ACADEMY

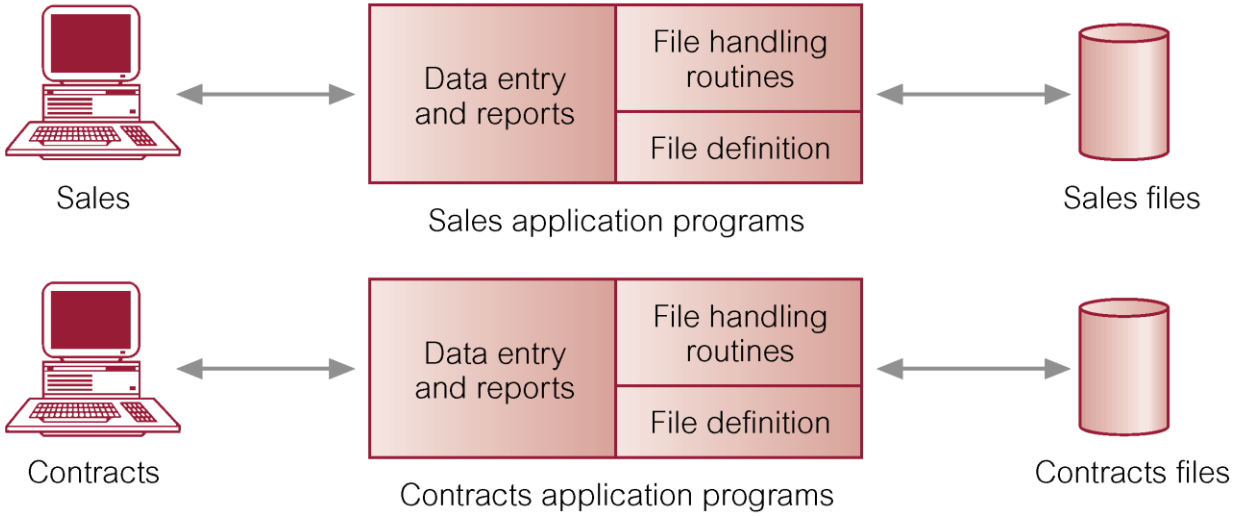
EXAMPLES OF DATABASE APPLICATIONS

- Purchases from the supermarket
- Purchases using your credit card
- Booking an airline flight
- Using the library
- Using the Internet
- Studying at a university

FILE-BASED SYSTEMS

- File-based systems are collections of application programs that perform services for the end users (e.g. create reports).
- Each program defines and manages its own data.

FILE-BASED PROCESSING



Sales Files

- **PropertyForRent** (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
- **PrivateOwner** (ownerNo, fName, lName, address, telNo)
- **Client** (clientNo, fName, lName, address, telNo, prefType, maxRent)

Contracts Files

- **Lease** (leaseNo, propertyNo, clientNo, rent, paymentMethod, deposit, paid, rentStart, rentFinish, duration)
- **PropertyForRent** (propertyNo, street, city, postcode, rent)
- **Client** (clientNo, fName, lName, address, telNo)

What’s wrong with this picture?

LIMITATIONS OF FILE-BASED APPROACH

- Separation and isolation of data
  - Each program maintains its own set of data
  - Users of one program may be unaware of potentially useful data held by other programs
- Duplication of data
  - Same data is held by different programs
  - Wasted space and potentially different values and/or different formats for the same item
- Data dependence
  - File structure is defined in the program code

DATABASE APPROACH

- The modern database approach arose as a solution to these problems.
- Database definition: a collection of logically related data and a description of this data.
- Logically related data comprises entities, attributes, and relationships.
- System catalog (metadata) provides description of data to enable program–data independence.

DATABASE MANAGEMENT SYSTEM (DBMS)

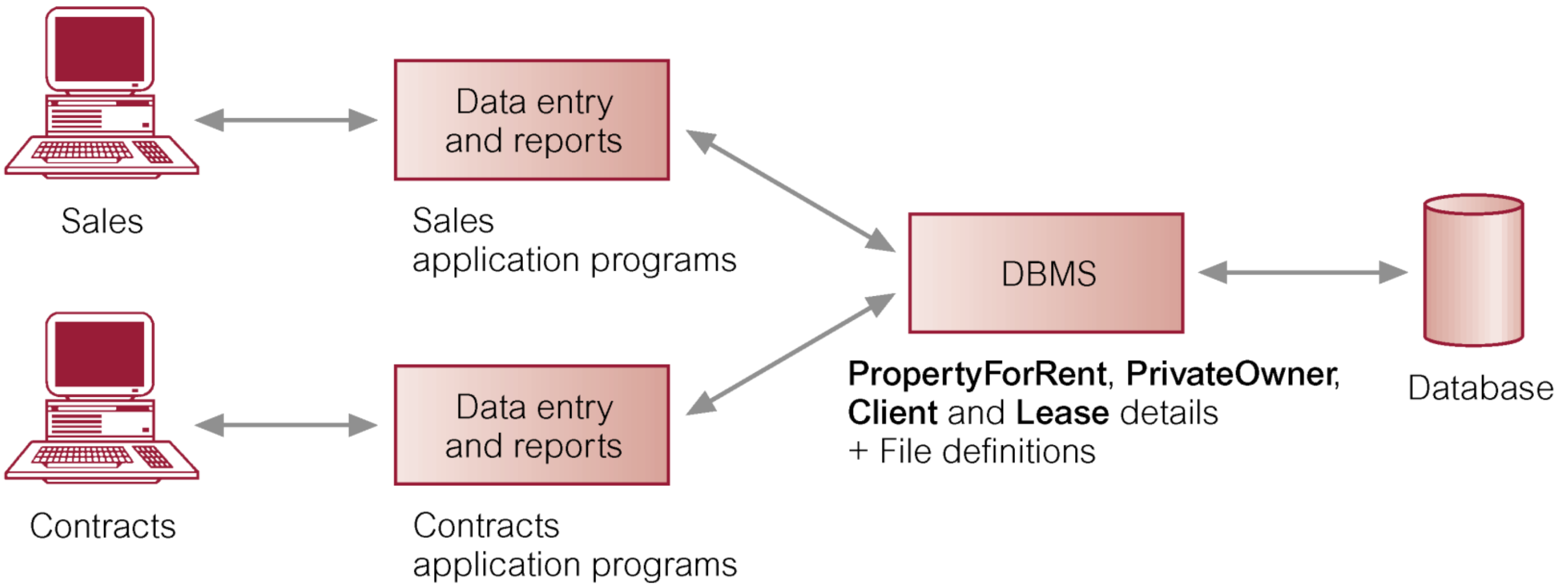
- The Database Management System (DBMS) is a software system that enables users to define, create, and maintain the database and which provides controlled access to this database.

- The DBMS provides a view mechanism:
  - Allows each user to have his or her own view of the database
  - Provides users with only the data they want or need to use
  - A view is essentially some subset of the database

Views

- Benefits of views include:
  - Reduced complexity (for the user)
  - Enhanced security
  - Customize the appearance of the database
  - Present a consistent, unchanging picture of the structure of the database, even if the underlying database is changed

DATABASE MANAGEMENT SYSTEM (DBMS)



PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)
Lease (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentFinish)

ADVANTAGES OF DBMS

- Control of data redundancy / consistency
- Improved data integrity
- Improved security
- Economy of scale
- Increased concurrency
- Improved backup and recovery services
- Increased productivity

DISADVANTAGES OF DBMS

- Complexity
- Size
- Additional hardware costs
- Cost of the DBMS
- Cost of conversion
- Performance
- Higher impact of a failure

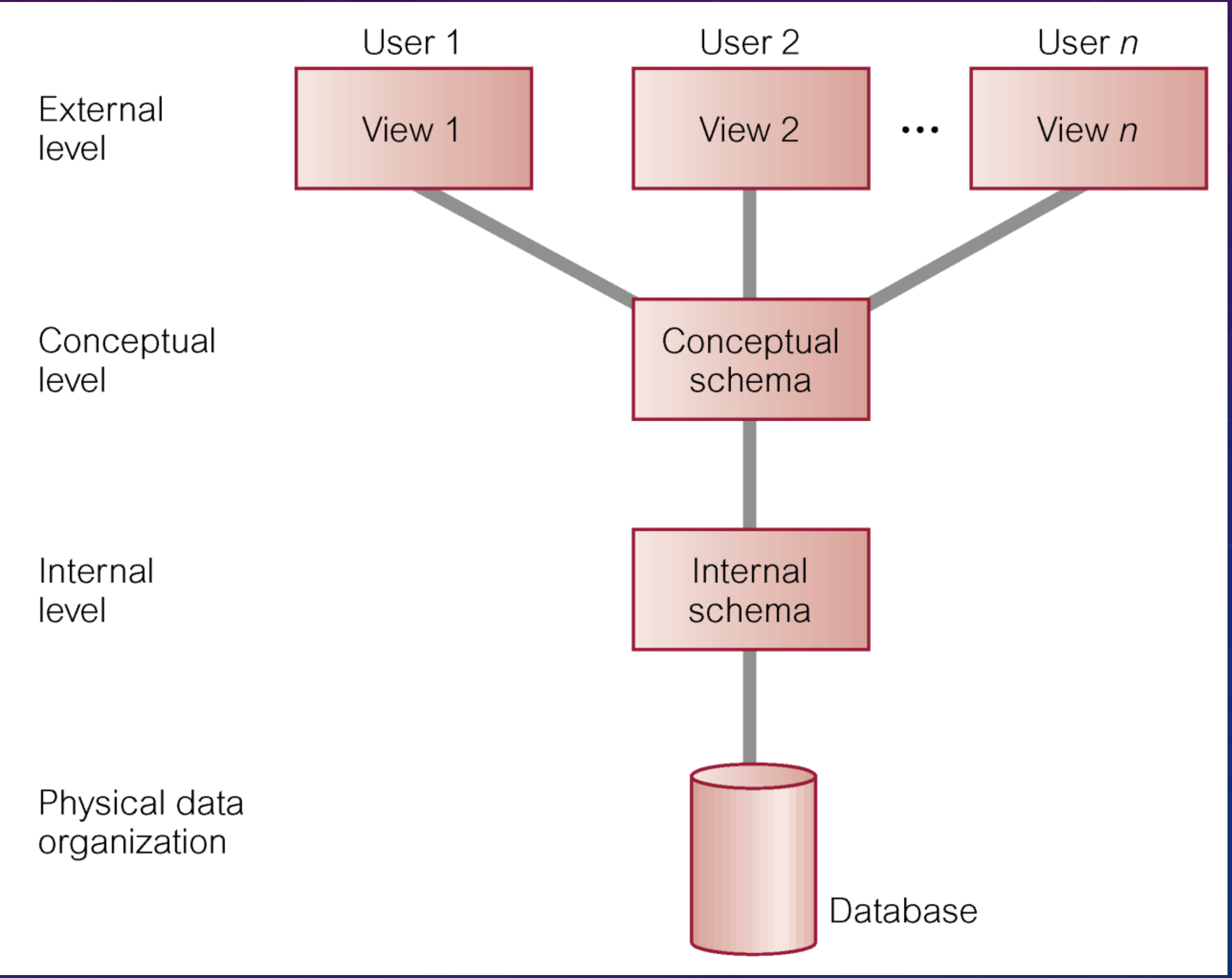
DATABASE ENVIRONMENT

KAL ACADEMY

OBJECTIVES OF THREE-LEVEL ARCHITECTURE

- **All users should be able to access same data, but have a different and customizable view.**
- **A user's view is immune to changes made in other views.**
- **Users should not need to know physical database storage details.**
- **Administrators should be able to change database storage structures without affecting the users' views.**

ANSI-SPARC THREE-LEVEL ARCHITECTURE



ANSI-SPARC THREE-LEVEL ARCHITECTURE

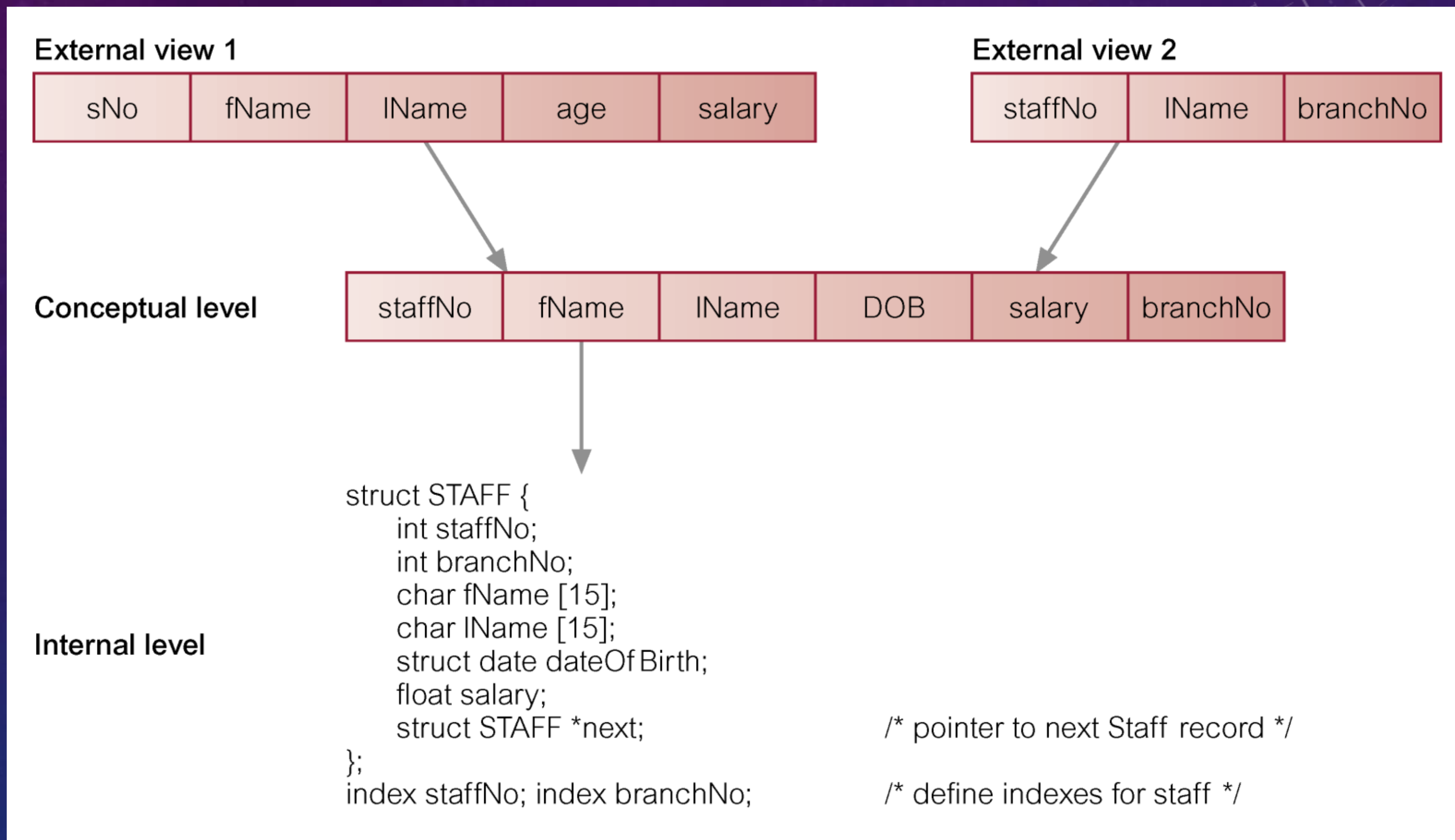
- **External Level**
  - **Users' view of the database**
  - **Describes that part of database that is relevant to a particular user**

- **Conceptual Level**
  - **Community view of the database**
  - **Describes what data is stored in database and relationships among the data**
  - **Constraints and security also defined**

ANSI-SPARC THREE-LEVEL ARCHITECTURE

- **Internal Level**
  - **Physical representation of the database on the computer**
  - **Describes how the data is stored in the database**

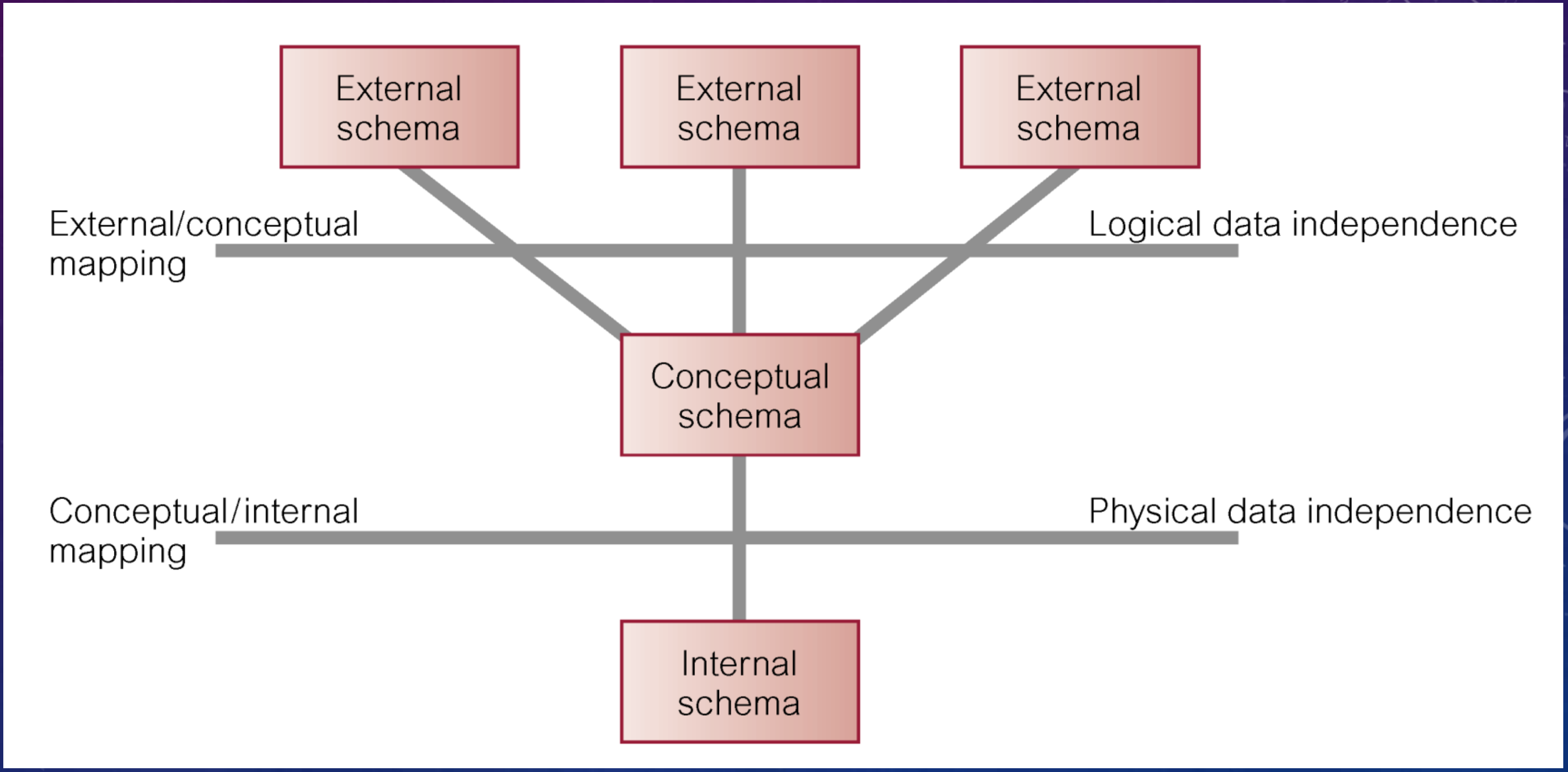
DIFFERENCES BETWEEN THREE LEVELS OF ANSI-SPARC ARCHITECTURE



DATA INDEPENDENCE

- **Logical Data Independence**
  - **Refers to immunity of external schemas to changes in conceptual schema (e.g. addition / removal of entities)**
- **Physical Data Independence**
  - **Refers to immunity of conceptual schema to changes in the internal schema (e.g. different file organization, storage structures/devices)**

DATA INDEPENDENCE AND THE ANSI-SPARC THREE-LEVEL ARCHITECTURE



THE RELATIONAL MODEL

KAL ACADEMY

RELATIONAL MODEL TERMINOLOGY

- A relation is a table with columns and rows.
  - Does not apply to physical layout, only the conceptual and external levels of the architecture
- An attribute is a named column of a relation.
- A domain is the set of allowable values for one or more attributes.

RELATIONAL MODEL TERMINOLOGY

- **A tuple is a row of a relation.**
- **The degree of a relation is the number of attributes it contains.**
- **The cardinality of a relation is the number of tuples it contains.**
- **A relational database is a collection of normalized relations with distinct relation names.**

INSTANCES OF BRANCH AND STAFF RELATIONS

Attributes

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

Relation

Cardinality

Degree

Primary key

Foreign key

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

Relation

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

MATHEMATICAL DEFINITION OF RELATION

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

- **The 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.**

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

MATHEMATICAL DEFINITION OF RELATION

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

  $R = \{(2, 1), (4, 1)\}$
- May specify which pairs are in relation using some condition for selection; e.g.
  - second element is 1:

    $R = \{(x, y) \mid x \in D_1, y \in D_2, \text{and } y = 1\}$
  - first element is always twice the second:

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

MATHEMATICAL DEFINITION OF RELATION

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

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

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

- Any subset of these ordered triples is a relation.

MATHEMATICAL DEFINITION OF RELATION

- The Cartesian product of n sets (D_1, D_2, . . ., D_n) is:

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

- Any set of n-tuples from this Cartesian product is a relation on the n sets.
- So, given any number of domains (i.e. sets), a relation on those domains would be a subset of the Cartesian product of the domains.

DATABASE RELATIONS

- **Relation schema**
  - Named relation defined by a set of attributes and corresponding domains
- **Relational database schema**
  - Set of relation schemas, each with a distinct name

PROPERTIES OF RELATIONS

- **Each cell of relation contains exactly one atomic (single) value.**
- **Each attribute in a relation has a distinct name.**
- **Values of an attribute are all from the same domain.**
- **Each tuple is distinct; there are no duplicate tuples.**
- **The order of the attributes has no significance.**
- **The order of the tuples has no significance (theoretically).**

RELATIONAL KEYS

- **Superkey**
  - **An attribute, or a set of attributes, that uniquely identifies each tuple within a relation**

- **Candidate Key**
  - **Superkey (K) such that no proper subset is a superkey within the relation**
  - **In each tuple of R, values of K uniquely identify that tuple (uniqueness).**
  - **No proper subset of K has the uniqueness property (irreducibility).**

RELATIONAL KEYS

- **Primary Key**
  - **Candidate key selected to identify tuples uniquely within relation**
- **Alternate Keys**
  - **Candidate keys that are not selected to be primary key**
- **Foreign Key**
  - **Attribute, or set of attributes, within one relation that matches candidate key of some (possibly same) relation**

INSTANCES OF BRANCH AND STAFF RELATIONS

Attributes

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

Relation

Cardinality

Degree

Primary key

Foreign key

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

Relation

RELATIONAL INTEGRITY

- Before we discuss relational integrity we need to discuss the concept of null.
- Null:
  - Represents value for an attribute that is currently unknown or not applicable for tuple
  - Deals with incomplete or exceptional data
  - Represents the absence of a value and is not the same as zero or spaces, which are values

RELATIONAL INTEGRITY

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

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

RELATIONAL INTEGRITY

- **Enterprise Constraints**
  - **Additional rules specified by users or database administrators.**