The Relational Model

Objectives

- At the end of this lesson, you should be able to:
  - ✓ Define relation, attribute, domain, tuple, degree, cardinality, modality and relational database.
  - ✓ Understand the mathematical relation of database.
  - ✓ Describe the relation and relational database schema.
  - ✓ Explain the properties of relation.
  - ✓ Identify the super, candidate, primary, alternate and foreign key.
  - ✓ Describe the nulls, entity and referential integrity, and general constraint.
  - ✓ Explain base relation, view, purpose of view and updating views.

Introduction

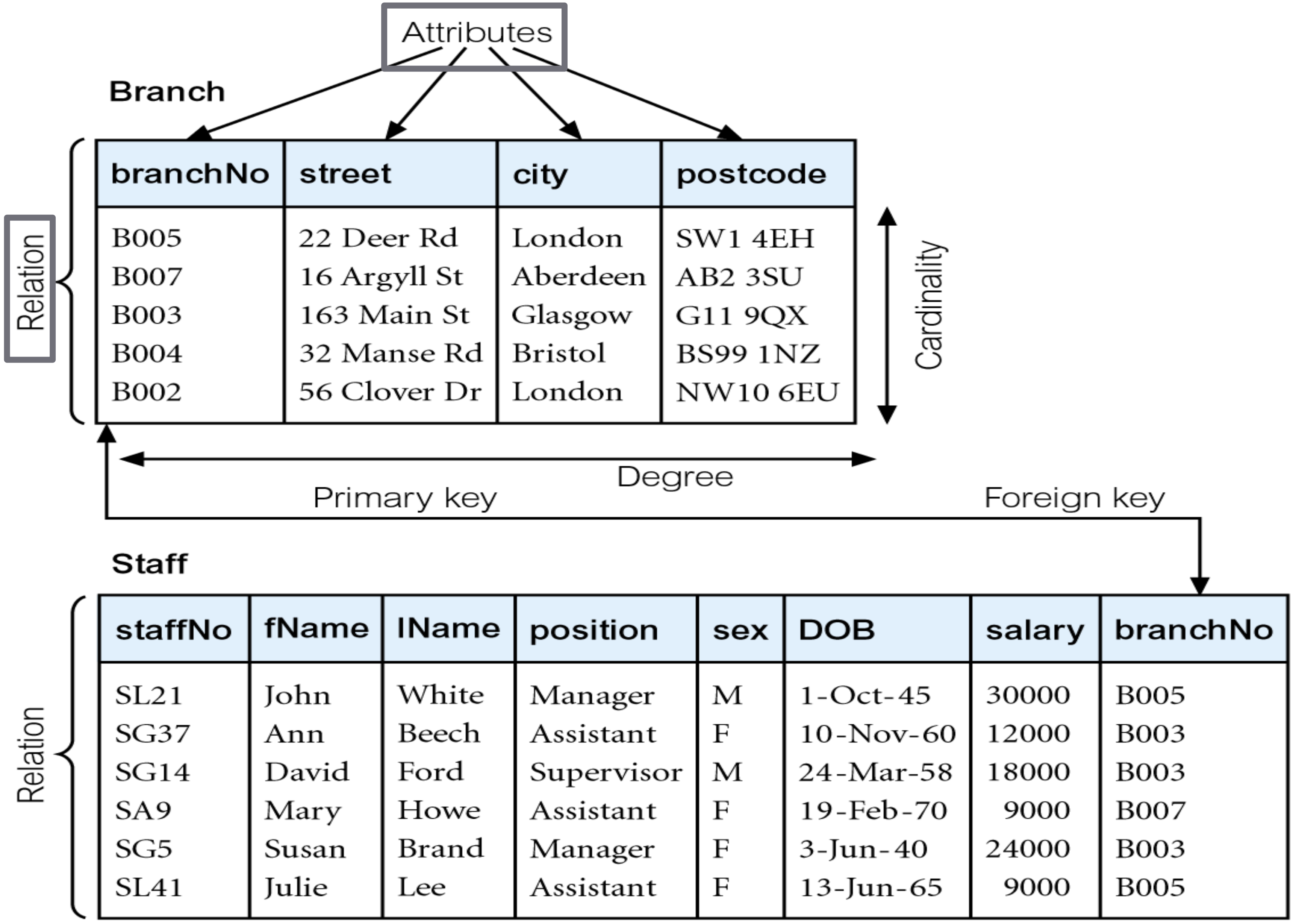
- The relational model was first proposed by E. F. Codd in his seminal paper ‘A relational model of data for large shared data banks’ (Codd, 1970).
- Data appears to be stored in what we have been referring to as **simple, linear files.**
- Relational databases are based on **mathematics.**
- A relational database is a **collection of relations** that, as a group, contain the data that describes a particular business environment.

Terminology

- The relational model is based on the mathematical concept of a relation, which is physically represented as a table.
- Relation : table with columns and rows.
- Attribute : Named column of a relation
- Domain : The set of allowable values for one or more attributes.
- Tuple is a row of a relation.
- Degree of a relation is the number of attributes it contains.
- Cardinality is the number of tuples in a relation.

Attribute : Named column of a relation

Relation : table with columns and rows.



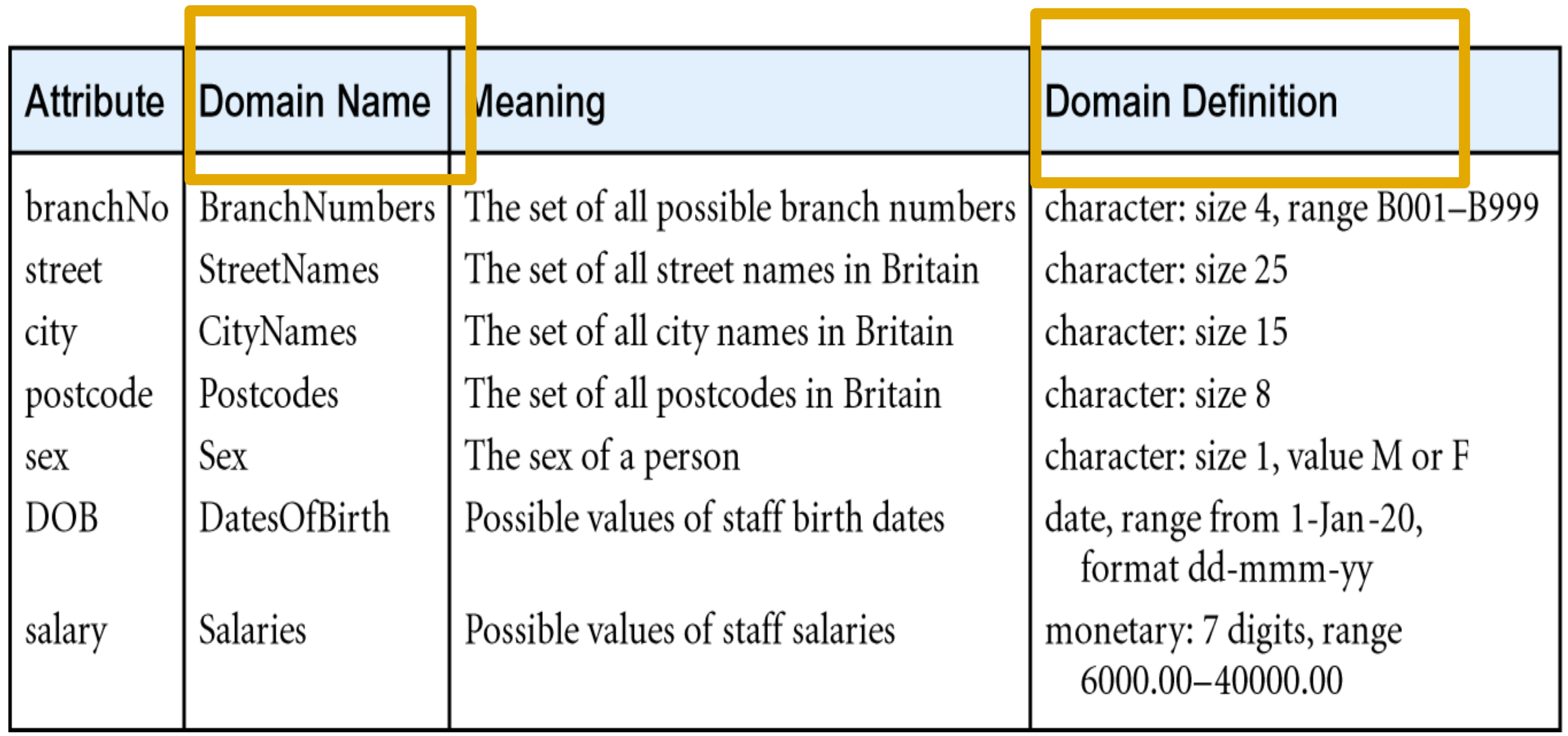
Branch

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

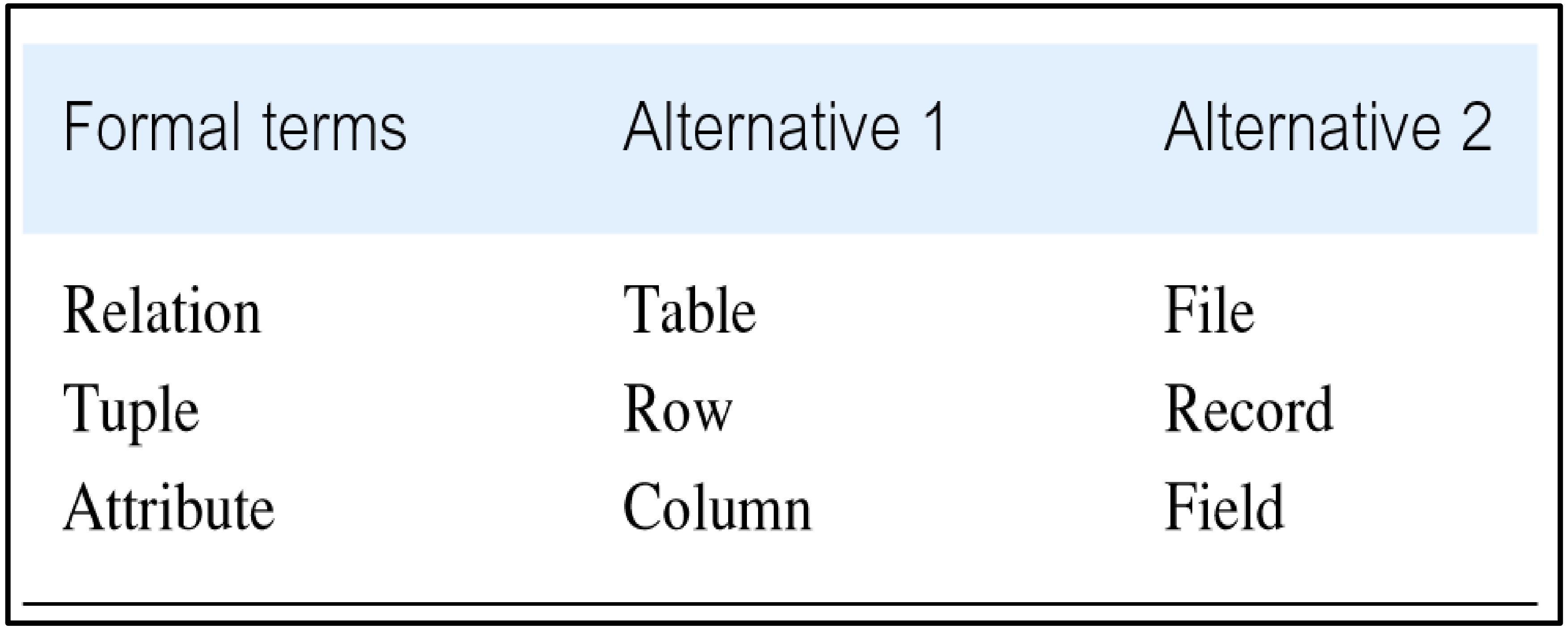
staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee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

Example of Attributes Domain



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



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

Mathematical Relation

- To understand the true meaning of the term ***relation***, we have to review some concepts from mathematics.
- Consider two sets, **D1 & D2**, where **D1 = {2, 4} and D2 = {1, 3, 5}**
- Cartesian product, ***D1 X D2***, is set of all ordered pairs, where first element is member of D1 and second element is member of D2.
- ***D1 X D2*** **= {(2, 1), (2, 3), (2, 5), (4, 1), (4, 3), (4, 5)}**

- 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:

  $R = \{(x, y) \mid x \in D1, y \in D2, \text{ and } y = 1\}$

- first element is always twice the second:

  $S = \{(x, y) \mid x \in D1, y \in D2, \text{ and } x = 2y\}$

- Consider three sets D1 = {1, 3} , D2 = {2, 4}, and D3 = {5, 6}
- The Cartesian product ***D1* × *D2* × *D3*** of these three sets is the set of all ordered triples such that the first element is from D1, the second element is from D2, and the third element is from D3
- ***D1* × *D2* × *D3* = {(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.

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

and is usually written as:

$$\boxed{\prod_{i=1}^{n} D_1}$$

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

Database Relations

Relation schema: a named relation defined by a set of attribute and domain name pairs

- Example of stafs' relation schema
  STAFF (staffid, staffname, staffemail, staffaddress, deptid)
- Example of departments' relation schema

  DEPT (deptid, deptname, locid)
- Example of locations' relation schema

  LOCATION (locid, locname, city)

Database Relations

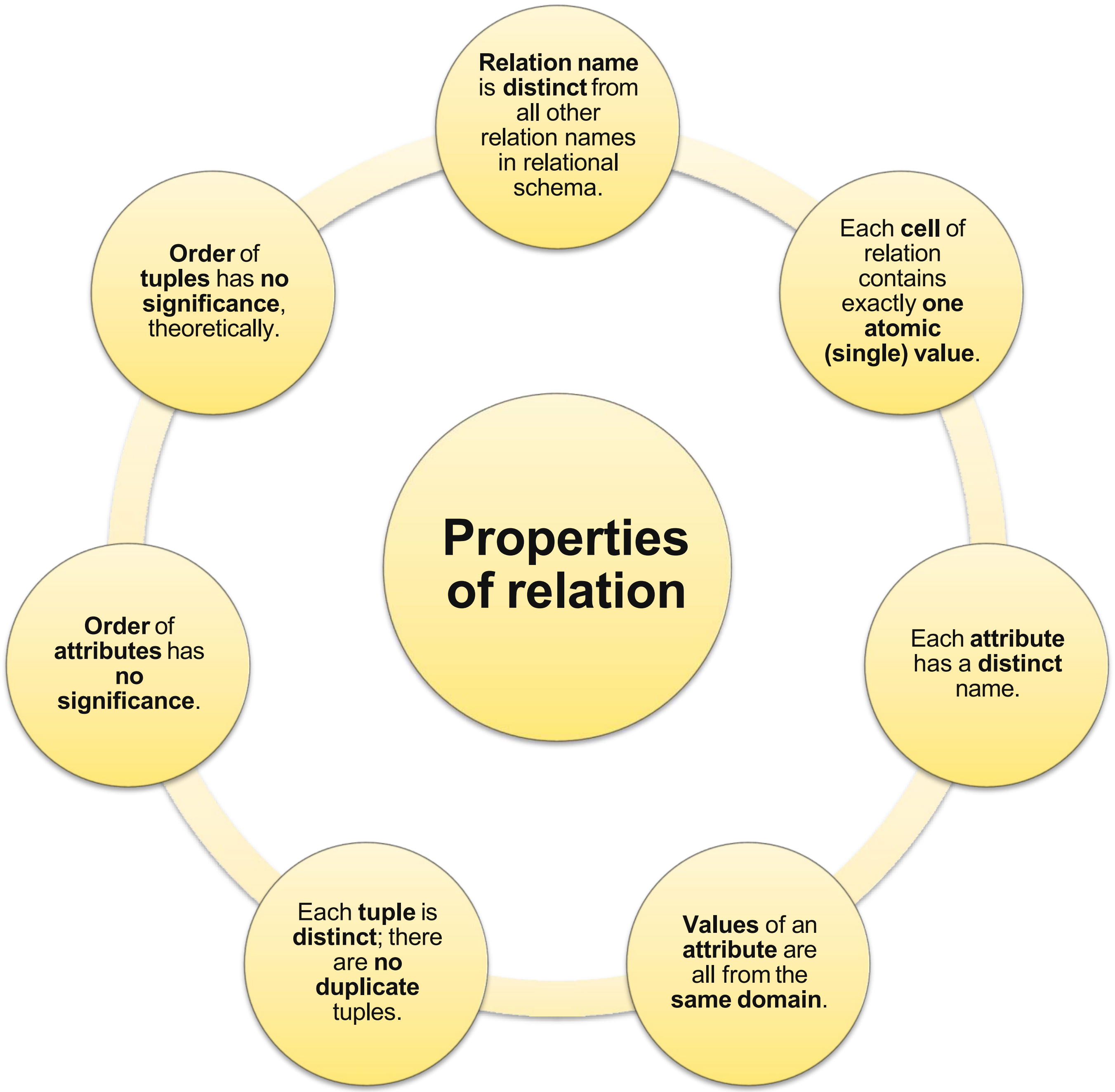
Relational database schema: Set of relation schemas, each with a distinct name.

- Example of Human Resource relational schema which combine of all previous relation schemas

  STAFF (<u>staffid</u>, staffname, staffemail, staffaddress, deptid*)

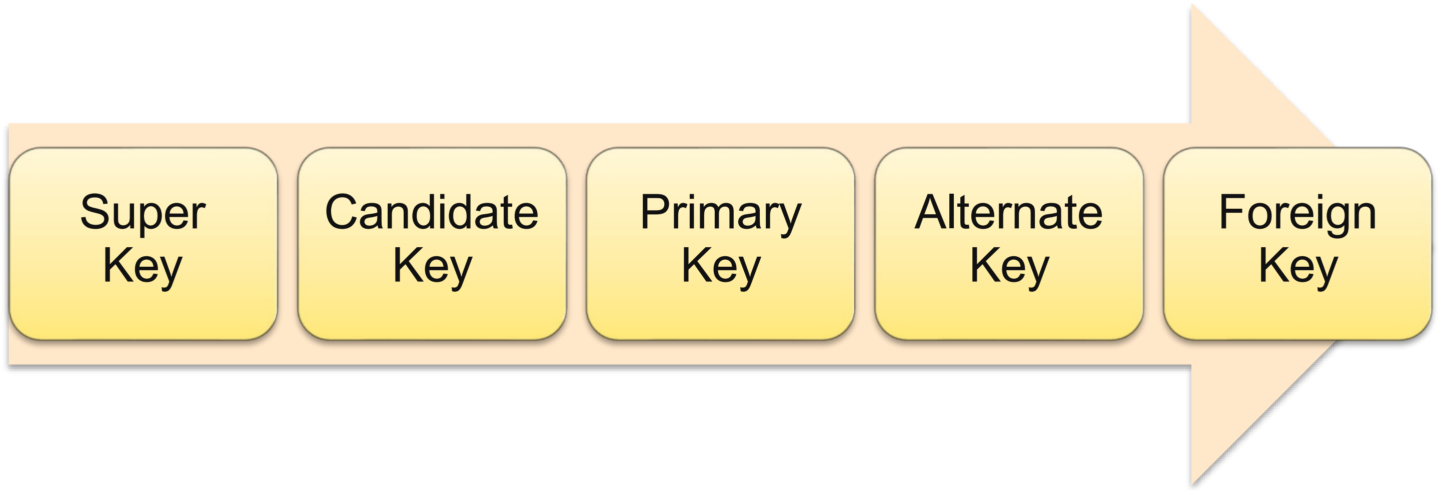
  DEPT (<u>deptid</u>, deptname, locid*)

  LOCATION (<u>locid</u>, locname, city)

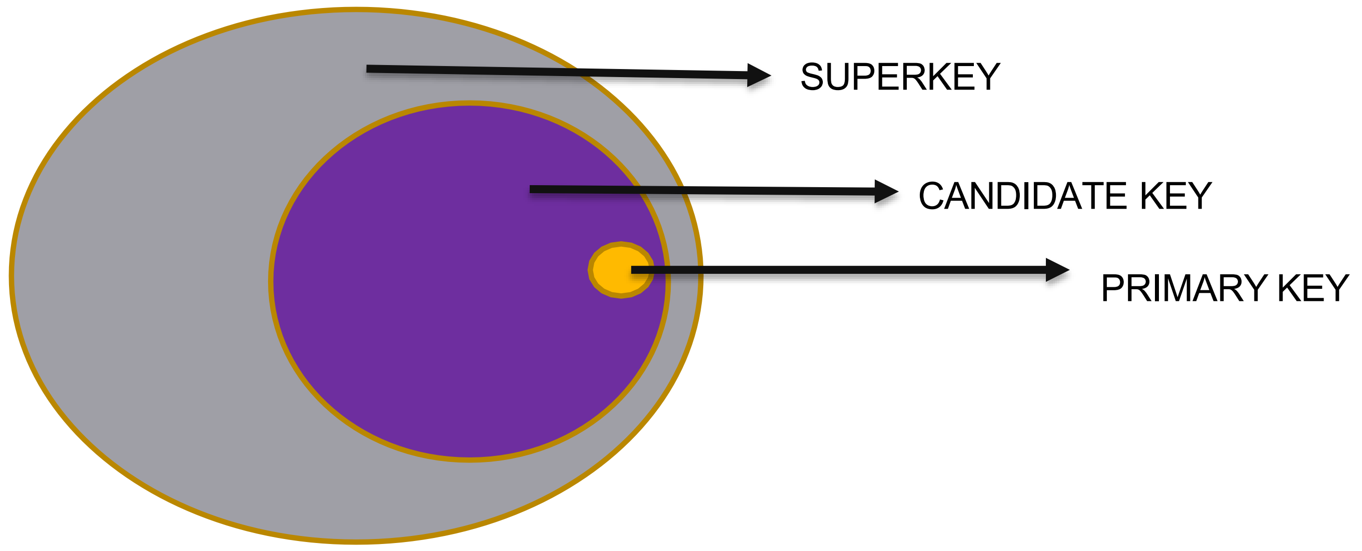

Properties of relation
Relation name is **distinct** from all other relation names in relational schema.
Each **cell** of relation contains exactly **one atomic (single) value**.
Each **attribute** has a **distinct** name.
Values of an **attribute** are all from the **same domain**.
Each **tuple** is **distinct**; there are **no duplicate** tuples.
Order of **attributes** has **no significance**.
Order of **tuples** has **no significance**, theoretically.

Relational Keys

- As stated previously, there are no duplicate tuples within a relation.
- Therefore, we need to be able to identify one or more attributes (called relational keys) that uniquely identifies each tuple in a relation.



- **Super key** is an attribute, or set of attributes, that uniquely identifies a tuple within a relation.
- **Candidate key** is a minimal super key or a super key with no redundant attribute .
- **Primary key** is the candidate key that is selected to identify tuples uniquely within the key relation.



Example

vehicleID	carPlateNo	engineID	carName
101	WTY 1234	9999876	Proton Pesona
102	WXY 4567	5644321	Perodua Alza
103	CDA 3389	6667889	Proton Pesona
104	MCD 1745	1277653	Honda City

Superkey:

vehicleID
carPlateNo
engineID

vehicleID, carPlateNo
vehicleID, engineID
vehicleID, carName
carPlateNo, engineID
carPlateNo, carName
engineID, carName

vehicleID, carPlateNo, engineID
vehicleID, carPlateNo, carName
vehicleID, engineID, carName
carPlateNo, engineID, carName
vehicleID, carPlateNo, engineID, carName

Example

vehicleID	carPlateNo	engineID	carName
101	WTY 1234	9999876	Proton Pesona
102	WXY 4567	5644321	Perodua Alza
103	CDA 3389	6667889	Proton Pesona
104	MCD 1745	1277653	Honda City

Candidate key:

vehicleID

carPlateNo

engineID

Primary Key: vehicleID

Example

vehicleID	carPlateNo	engineID	carName
101	WTY 1234	9999876	Proton Pesona
102	WXY 4567	5644321	Perodua Alza
103	CDA 3389	6667889	Proton Pesona
104	MCD 1745	1277653	Honda City

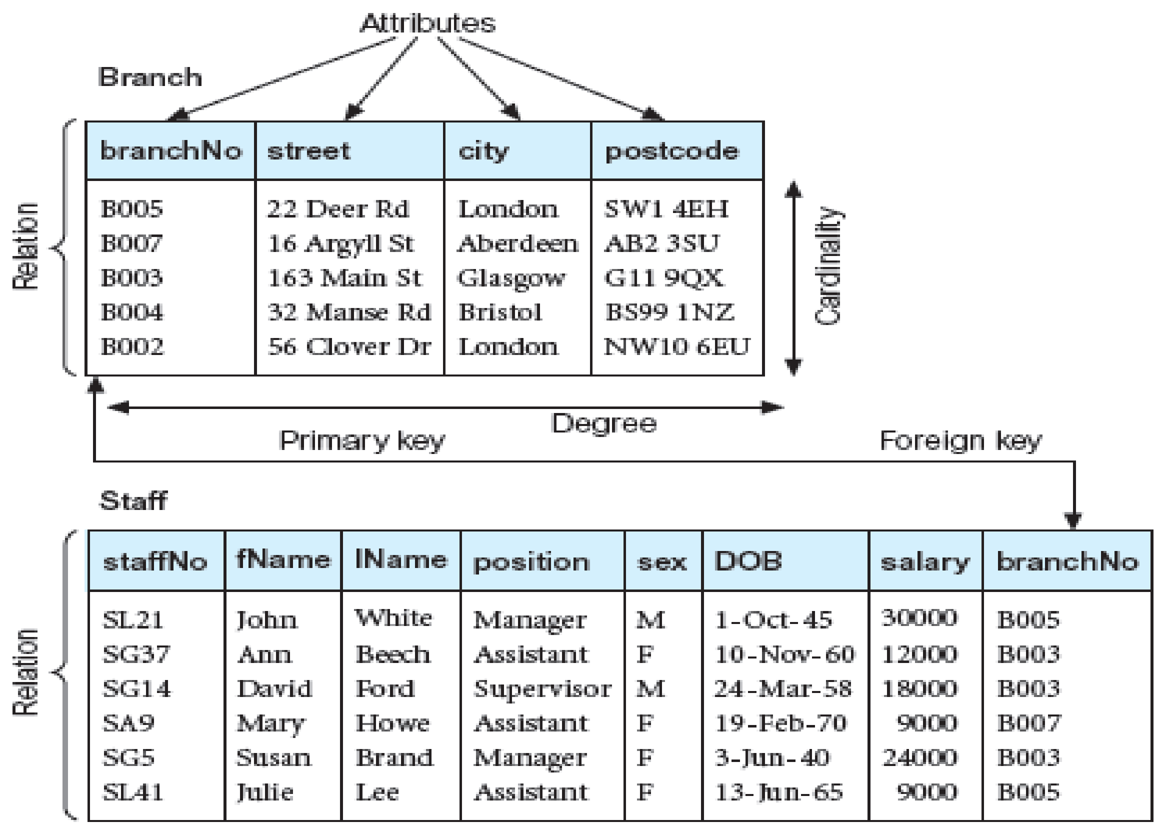
Alternate key:

carPlateNo

engineID

Alternate Key is a candidate key that was not chosen to be the primary key of the relation.

Foreign Key (FK) is an attribute or group of attributes that serves as the primary key of one relation and also appears in another relation (foreign key in this relation). FK establishes the relationship among tables.



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

Integrity Constraints

- Integrity constraints is to make sure that the data is accurate.

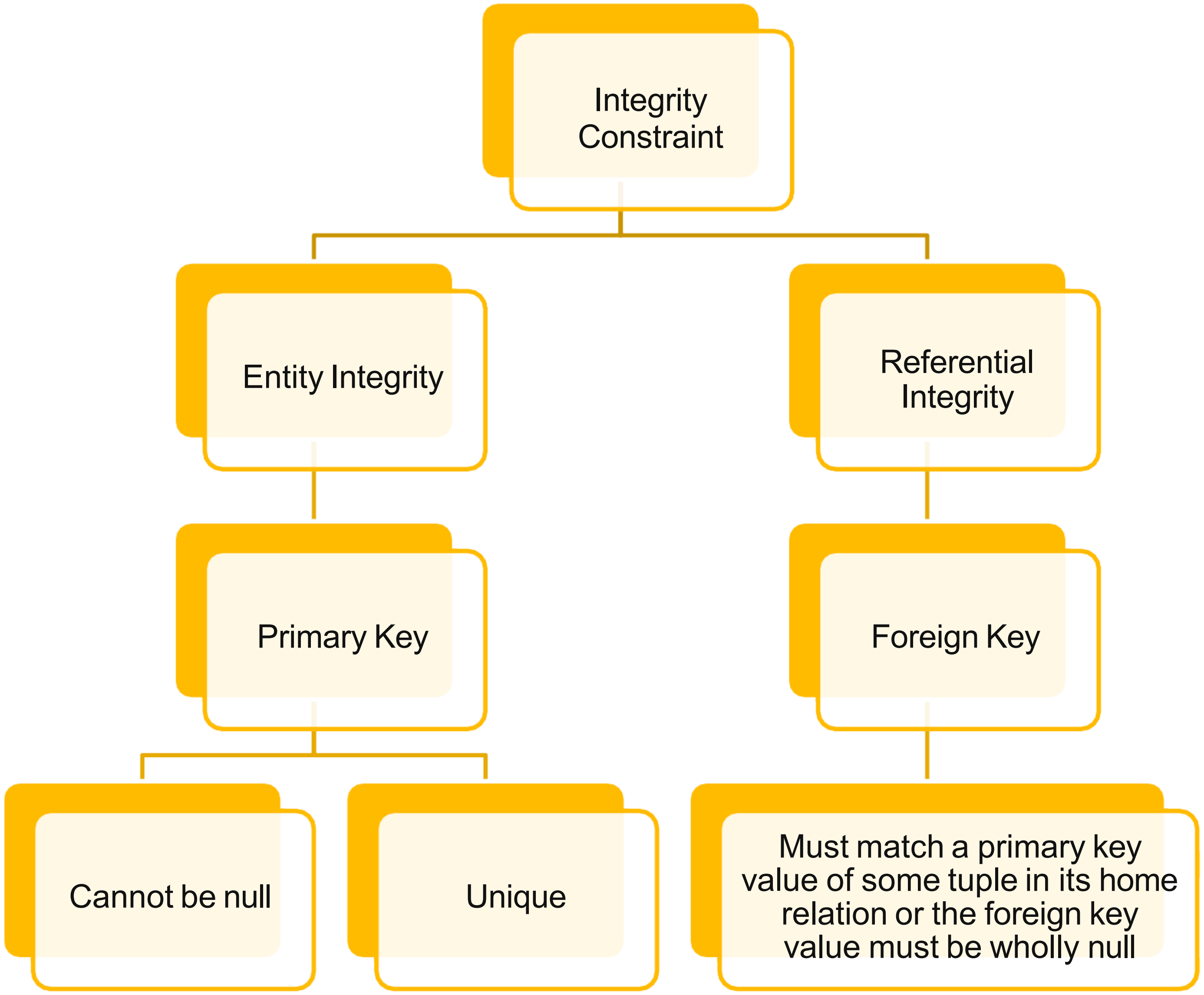
- Data inaccuracy could leave the whole database system unreliable and can affect the company from running smoothly.

- There are two important integrity rules, which are constraints or restrictions that apply to all instances of the database.

- The two principal rules for the relational model are known as

  i. **entity integrity**
  ii. **referential integrity**

- Null value represents a value for an attribute that is currently unknown or is not applicable for this tuple.


Integrity Constraint
Entity Integrity
Referential Integrity
Primary Key
Foreign Key
Cannot be null
Unique
Must match a primary key value of some tuple in its home relation or the foreign key value must be wholly null

Insert, delete, update for referential integrity

	PARENT TABLE	CHILD TABLE
INSERT	No Problem	Cannot be done if there is no match primary key in parent's table
DELETE	Cannot be done if there is a match foreign key in child's table	No Problem
UPDATE	Cannot be done if there is a match foreign key in child's table	Cannot be done if there is no match existing primary key in parent's table

Delete & update actions for reference rows

- Early relational DBMSs did not provide any control mechanisms for referential integrity.

- Modern relational DBMSs provide sophisticated control mechanisms for referential integrity:

  ✓ **Delete rules**
  ✓ Insert rules
  ✓ **Update rules**

The **ON DELETE** and **ON UPDATE** action associated with each foreign key in database is one of "**NO ACTION**", "**RESTRICT**", "**SET NULL**", "**SET DEFAULT**" or "**CASCADE**".

- If an action is not explicitly specified, it defaults to "**NO ACTION**".

- **NO ACTION**: Configuring "**NO ACTION**" means just that: when a parent key is modified or deleted from the database, **no special action** is taken.

- **RESTRICT**: The "**RESTRICT**" action means that the application is **prohibited** from deleting (for **ON DELETE RESTRICT**) or modifying (for **ON UPDATE RESTRICT**) a parent key when there exists one or more child keys mapped to it.

- **SET NULL**: If the configured action is "**SET NULL**", then when a parent key is deleted (for **ON DELETE SET NULL**) or modified (for **ON UPDATE SET NULL**), the child key columns of all rows in the child table that mapped to the parent key are set to contain **SQL NULL** values.

- **SET DEFAULT:** The "**SET DEFAULT**" actions are similar to " SET NULL", except that each of the child key columns is set to contain the columns **default value** instead of NULL.

- **CASCADE**: A "**CASCADE**" action **propagates** the delete or update operation on the parent key to each dependent child key.

- For an "ON DELETE CASCADE" action, this means that each row in the child table that was associated with the deleted parent row is also deleted.

- For an "**ON UPDATE CASCADE**" action, it means that the values stored in each dependent child key **are modified** to match the **new parent** key values.

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

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.
- **Virtual relation** that does not necessarily exist in the database but can be produced upon request by a particular user, at the time of request.

Views

- 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.
- Modification's restrictions:
  - ✓ Updates are allowed if query involves a single base relation and contains a candidate key of base relation.
  - ✓ Updates are not allowed involving multiple base relations.
  - ✓ Updates are not allowed involving aggregation or grouping operations.

Classes of Views

- theoretically not updateable;
- theoretically updateable;
- partially updateable.

Review Questions

Question 1

a) Differentiate between a superkey and a primary key

(4 marks)

Question 2

Consider the relational schema below:

STUDENT (StudentID, Name, GradYear)

COURSE (CourseID, CourseTitle)

PREREQUISITE (CourseID, PrereqCourseID)

ENROLLMENT (StudentID, CourseID, Semester, Year, Grade)

a) Give the primary key and foreign key (if any) for each relation.

(4 marks)

References

Thomas Connolly and Carolyn Begg, Database Systems:A Practical Approach to Design, Implementation, and Management, 6th Edition, Pearson, 2015, ISBN: 978- 01329432