ICT502/ITS571

LECTURE 2

THE RELATIONAL MODEL

Muhammad Hamiz Mohd Radzi
Faiqah Hafidzah Halim

Contents

- Terminology
- Integrity Constraints
- Views

Objectives

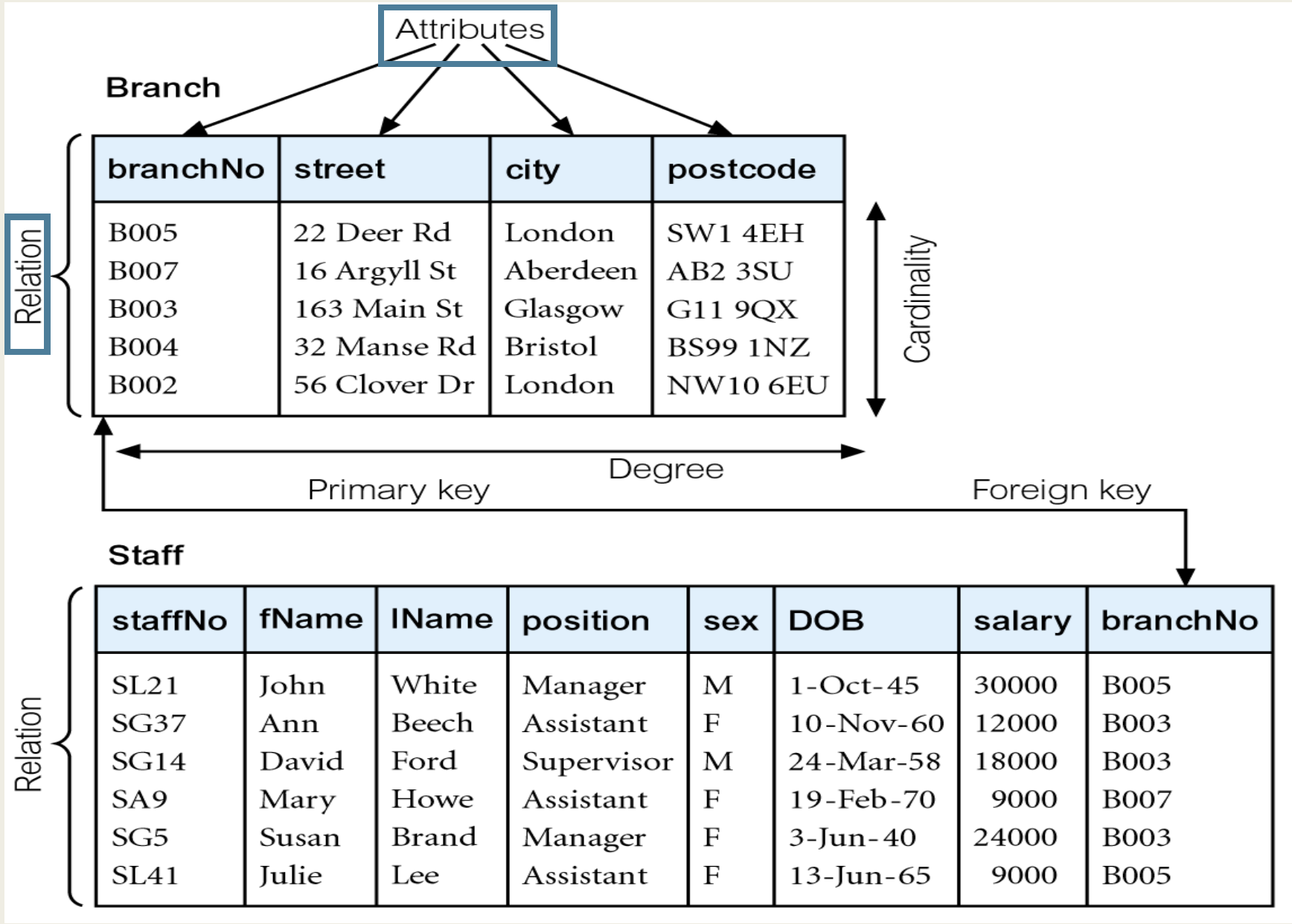
- At the end of this lesson, you should be able to:
  - *Define relation, attribute, domain, tuple, degree, cardinality, modality and relational database.*
  - *Understand the mathematical relation of database*
  - *Describe the relation and relational database schema*
  - *Explain the properties of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

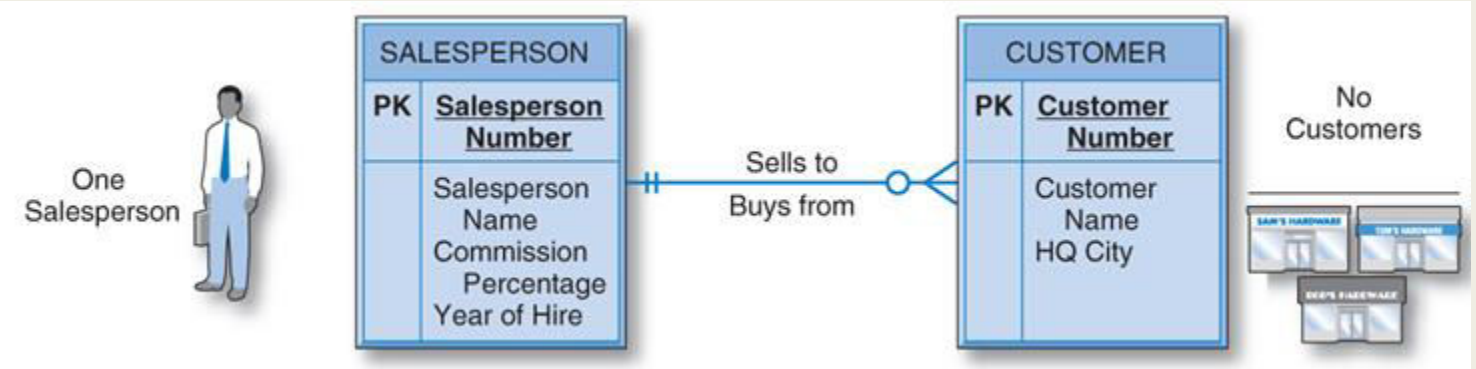
- The relational model is based on the mathematical concept of a **relation**, which is physically represented as a **table**.
- **Relation** : **table** with **columns** and **rows**.
- **Attribute** : Named **column** of a relation
- **Domain** : The set of allowable **values** for one or more attributes.


Attributes
Branch
Relation
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
Cardinality
Degree
Primary key
Foreign key
Staff
Relation
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 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

- **Tuple** is a **row** of a relation.
- **Degree** of a relation is the **number of attributes** it contains.
- **Cardinality** - the **maximum** number of entities that can be involved in a particular relationship. (Outer)
- **Modality** - The **minimum** number of entity occurrences that can be involved in a relationship. (Inner)



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, D_1 & D_2, where D_1 = {2, 4} and D_2 = {1, 3, 5}.
- Cartesian product, $\mathbf{D_1 X D_2}$, is set of all ordered pairs, where first element is member of D_1 and second element is member of D_2.
- $\mathbf{D_1 X D_2}$ = **{(2, 1), (2, 3), (2, 5), (4, 1), (4, 3), (4, 5)}**
- Alternative way is to find all combinations of elements with first from D_1 and second from D_2.

- 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:

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

- Consider three sets $D_1 = \{1, 3\}$, $D_2 = \{2, 4\}$, and $D_3 = \{5, 6\}$

- The Cartesian product $\mathbf{D_1 \times D_2 \times D_3}$ of these three sets is the set of all ordered triples such that the first element is from D_1, the second element is from D_2, and the third element is from D_3

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

- Any subset of these ordered triples is a relation.

- $D_1 \times D_2 \times \ldots \times D_n = \{(d_1, d_2, \ldots, d_n) \mid d_1 \in D_1, d_2 \in D_2, \ldots, d_n \in D_n\}$
- and is usually written as:

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

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

Database Relation

- Relation schema: a named relation defined by a set of attribute and domain name pairs.
- Relational database schema: Set of relation schemas, each with a distinct name.

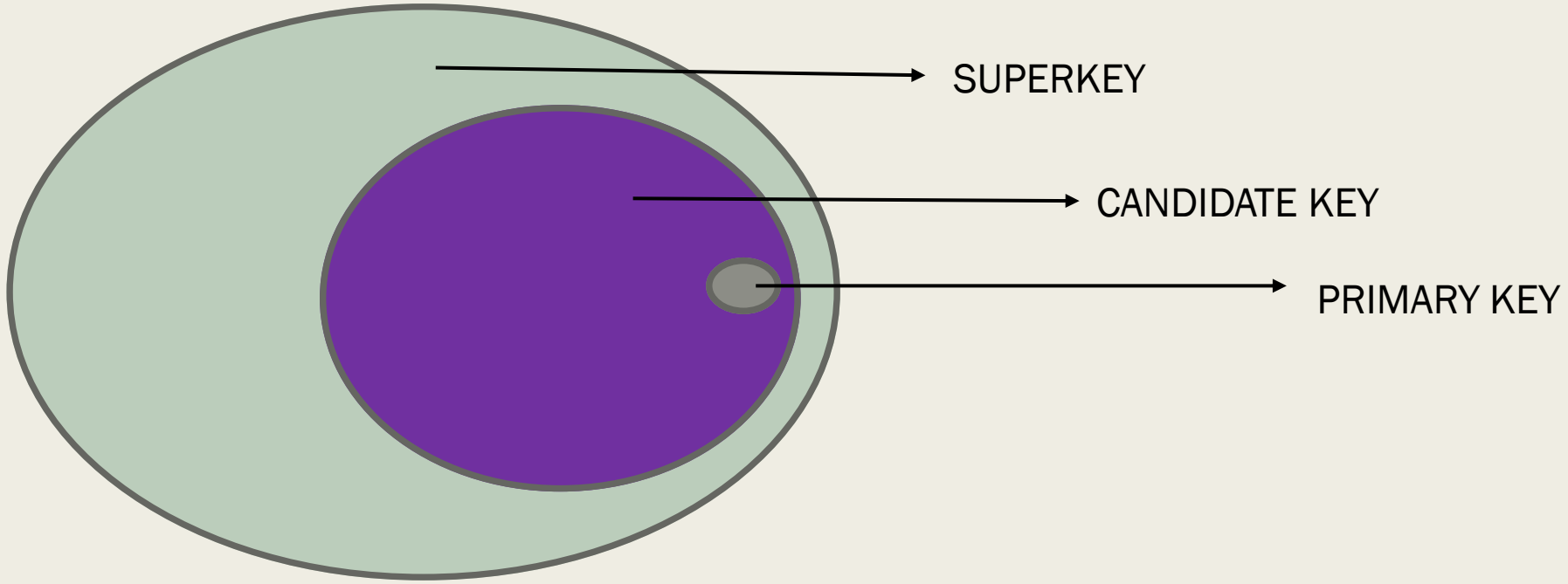
Properties of Relation

- **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.
- The important keys in a relation are:
  - *Superkey*
  - *Candidate Key*
  - *Primary Key*
  - *Alternate Key*
  - *Foreign Key*

- Superkey is an attribute, or set of attributes, that uniquely identifies a tuple within a relation.
- Candidate key is a minimal superkey with no redundancy.
- Primary key is the candidate key that is selected to identify uniquely within the key relation.



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

•Superkey:

- vehicleID
- carPlateNo
- engineID

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

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

•Candidate Key:

- vehicleID
- carPlateNo
- engineID

- vehicleID, carPlateNo
- vehicleID, engineID
- vehicleID, carPlateNo
- vehicleID, engineID
- carPlateNo, engineID

- vehicleID, carPlateNo, engineID

Primary Key: vehicleID

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

•Alternate Key:

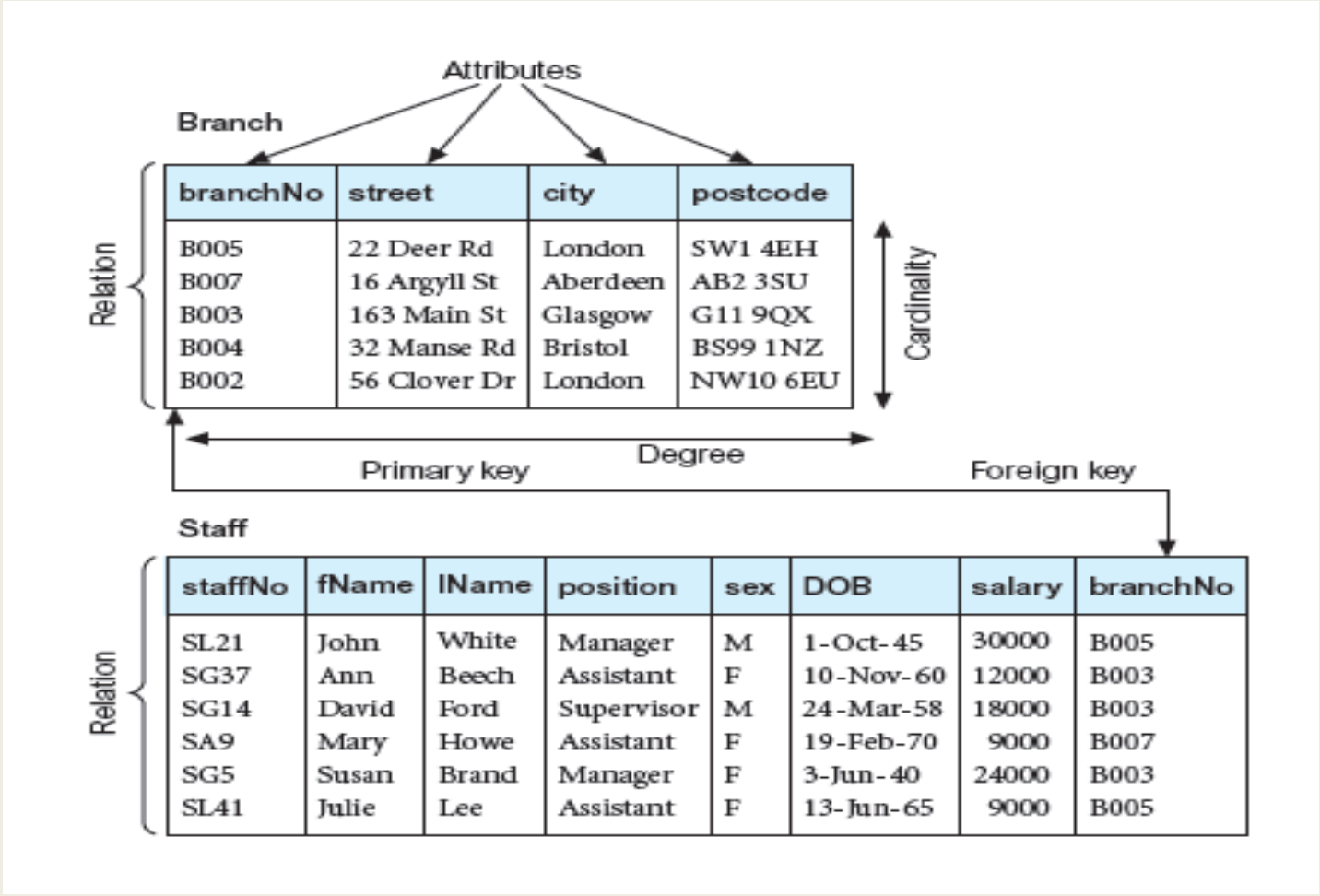
- carPlateNo
- engineID

- vehicleID, carPlateNo
- vehicleID, engineID
- vehicleID, carPlateNo
- vehicleID, engineID
- carPlateNo, engineID

- vehicleID, carPlateNo, engineID

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

- Foreign Key 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).



Integrity Constraints

- Integrity constraints is to make sure that the data is accurate.
- Inaccurate of data 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 **entity integrity** and **referential integrity**.
- Null value represents a value for an attribute that is currently **unknown** or is **not applicable** for this tuple.

Entity integrity

- The first integrity rule applies to the **primary keys** of base relations.
- Entity integrity can be defined as; in a base relation, no attribute of a primary key can be null.
- For example, in a STUDENT relation, studentID is considered as primary key.
- studentID **cannot be null** as it is the determinant for the other attributes like studentName, studentAddress etc.
- Hence, a primary key cannot be null and this kind of rule is called entity integrity.

Referential integrity

- The second integrity rule applies to **foreign keys**.
- Revolves around the circumstance of trying to **refer** to data in one relation in the database, based on values in another relation.
- By definition, referential integrity means that if a foreign key exists in a relation, either the foreign key value **must match** a primary key value of some tuple in its home relation or the foreign key value must be **wholly null**.
- For example, branchNo in the Staff relation is a foreign key targeting the branchNo attribute in the home relation, Branch.

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.

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.
- Modifications 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 are defined as:
  - *theoretically not updateable;*
  - *theoretically updateable;*
  - *partially updateable.*

Reference

- *Database Systems: A Practical Approach to Design, Implementation, and Management,* Thomas Connolly and Carolyn Begg, 5th Edition, 2010, Pearson.