Fair Use Notice:

The material used in this presentation i.e., pictures/graphs/text, etc. is solely intended for educational/teaching purpose, offered free of cost to the students for use under special circumstances of Online Education due to COVID-19 Lockdown situation and may include copyrighted material - the use of which may not have been specifically authorised by Copyright Owners. It's application constitutes Fair Use of any such copyrighted material as provided in globally accepted law of many countries. The contents of presentations are intended only for the attendees of the class being conducted by the presenter.

DATABASE SYSTEMS (SW215)

REFERENCIAL KEYS

By : HIRA NOMAN

THE DATABASE TERMS OF REFERENCE

Entity Modelling	Normalisation	RDBMS
Entity	Relation	Table
Entity Occurrence	Tuple	Row or Record
Attribute	Domain	Column or Field
Organisational Information	Model Data	Data

DATABASE KEYS

- Keys are, as their name suggests, a key part of a relational database and a vital part of the structure of a table.
- They ensure each record within a table can be uniquely identified by one or a combination of fields within the table.
- They help enforce integrity and help identify the relationship between tables.
- There are four main types of keys, **Super key, Candidate key, Primary key and Foreign key.**

1. Super Key

- A Super key is any combination of fields within a table that uniquely identifies each record within that table.

2. Candidate Key

- A candidate key is a subset of a super key.
- A candidate key is a single field or the least combination of fields that uniquely identifies each record in the table. The least combination of fields distinguishes a candidate key from a super key.
- Every table must have at least one candidate key but at the same time can have several.

- As an example, we might have a student_id that uniquely identifies the students in a student table. This would be a candidate key. But in the same table we might have the student's first name and last name that also, when combined, uniquely identify the student in a student table. These would both be candidate keys.
- In order to be eligible for a candidate key it must pass certain criteria.
  - ❖ It must contain unique values.
  - ❖ It must not contain null values.
  - ❖ It contains the minimum number of fields to ensure uniqueness.
  - ❖ It must uniquely identify each record in the table.
- Once your candidate keys have been identified you can now select one to be your primary key.

3. Primary Key

- A primary key is a candidate key that is most appropriate to be the main reference key for the table.
- As its name suggests, it is the primary key of reference for the table and is used throughout the database to help establish relationships with other tables.
- As with any candidate key the primary key must contain unique values, must never be null and uniquely identify each record in the table.
- As an example, student_id might be a primary key in a student table.
- In the table below we have selected the candidate key studentId to be our most appropriate primary key.

Primary keys are mandatory for every table. Each record must have a value for its primary key. When choosing a primary key from the pool of candidate keys always choose a **single simple key** over a **composite key**.

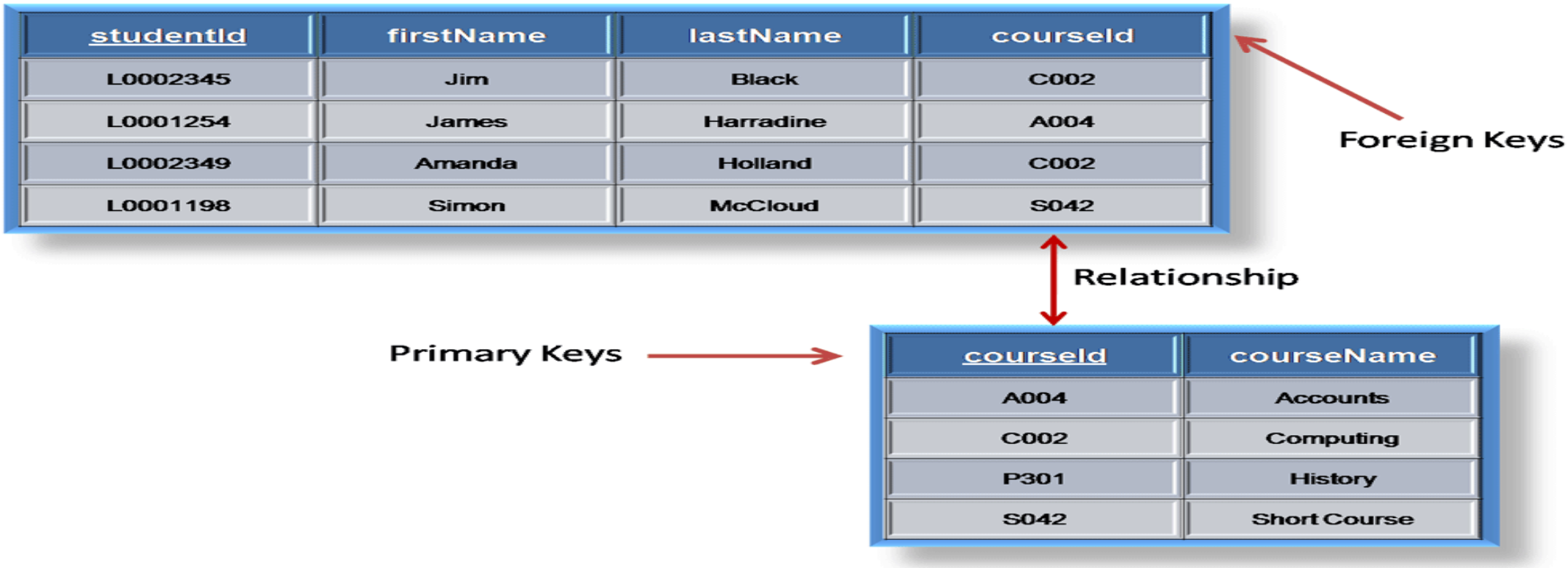
Primary Keys

StudentId	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042
L0023487	Peter	Murray	P301
L0018453	Anne	Norris	S042

4. Foreign Key

- A foreign key is generally a primary key from one table that appears as a field in another where the first table has a relationship to the second. In other words, if we had a table A with a primary key X that linked to a table B where X was a field in B, then X would be a foreign key in B.

- An example might be a student table that contains the course_id the student is attending. Another table lists the courses on offer with course_id being the primary key. The 2 tables are linked through course_id and as such course_id would be a foreign key in the student table.



CLASSIFICATION OF KEYS

1. Simple Key

- Any of the keys described before (i.e., primary, or foreign) may comprise one or more fields, for example if firstName and lastName was our key this would be a key of two fields where as student_id is only one.
- A simple key consists of a single field to uniquely identify a record.
- In addition, the field cannot be broken down into other fields, for example, student_id, which uniquely identifies a particular student, is a single field and therefore is a simple key.
- No two students would have the same student number.

2. Compound Key

- A compound key consists of more than one field to uniquely identify a record.
- A compound key is distinguished from a composite key because each field, which makes up the primary key, is also a simple key in its own right.
- An example might be a table that represents the modules a student is attending. This table has a student_id and a phone_no as its primary key. Each of the fields that make up the primary key are simple keys because each represents a unique reference when identifying a student in one instance and a Phone number in the other.

3. Composite Key

- A composite key consists of more than one field to uniquely identify a record. This differs from a compound key in that one or more of the attributes, which make up the key, are not simple keys.
- Taking the example from compound key, imagine we identified a student by their firstName + lastName. In our table representing students our primary key would now be firstName + lastName. Because firstName + lastName represent a unique reference to a student, they are not each simple keys, they have to be combined in order to uniquely identify the student. Therefore, the key for this table is a composite key.

TASK A

Student_Id	College_Id	Student_Name	Rtu_Roll_No	Father_Name	Address	Branch_Id	Session
1	1210038	Pankaj	12emccs138	Suresh	Alwar	1	2012-16
2	1210039	Priya	12emccs139	Ram	Mathura	1	2012-16
3	1310048	Rahul	13emmce210	Sambhu	Alwar	3	2013-17
4	1310078	Neeraj	13emcve078	Nathu	Jaipur	5	2013-17
5	1210047	Sandeep	12emccs047	Om Parkash	Alwar	1	2012-16
6	1210048	Sanjeev	12emece048	Omi	Delhi	2	2012-16

SUPER KEY ?

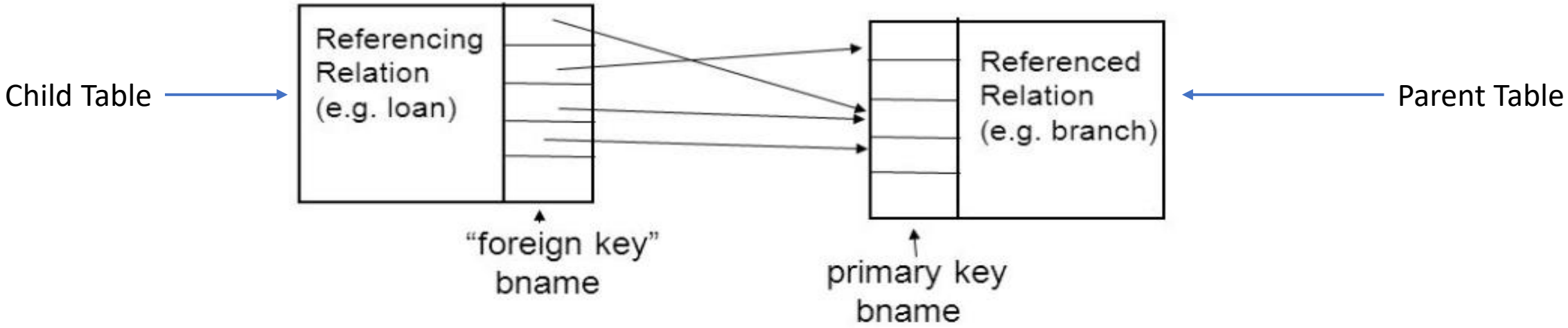
CANDIDATE KEY ?

PRIMARY KEY ?

FOREIGN KEY ?

REFERENCIAL INTEGRITY

- Referential integrity (RI) is a term used with relational databases to describe the integrity of the business relationships represented in the schema. It ensures that relationships between tables remain consistent.
- Referential integrity (RI) refers to the integrity of reference between data in related tables.
- Referential integrity is a well-understood relational constraint.
- The referential integrity constraints state that if a foreign key in the first table refers to the primary key of the second table then every value of foreign key in the first table should either be null or present in the second table.
- A referential integrity test asserts that all the foreign keys in a given column link to a correct record in the parent table.



Ref Integrity:

ensure that:

foreign key value → primary key value

(note: need not to ensure ←, i.e., not all branches have to have loans)

Student (First Table)

Roll_no	Student_name	Age	Course_id
1	Andrew	18	78
2	Angel	19	16
3	Priya	20	56
4	Analisa	21	

Foreign Key

This value is not allowed because this value is not defined as a primary key in the course table.

The value can be NULL as the student(Analisa) may not have taken any course.

Primary Key

Course (Second Table)

Primary Key

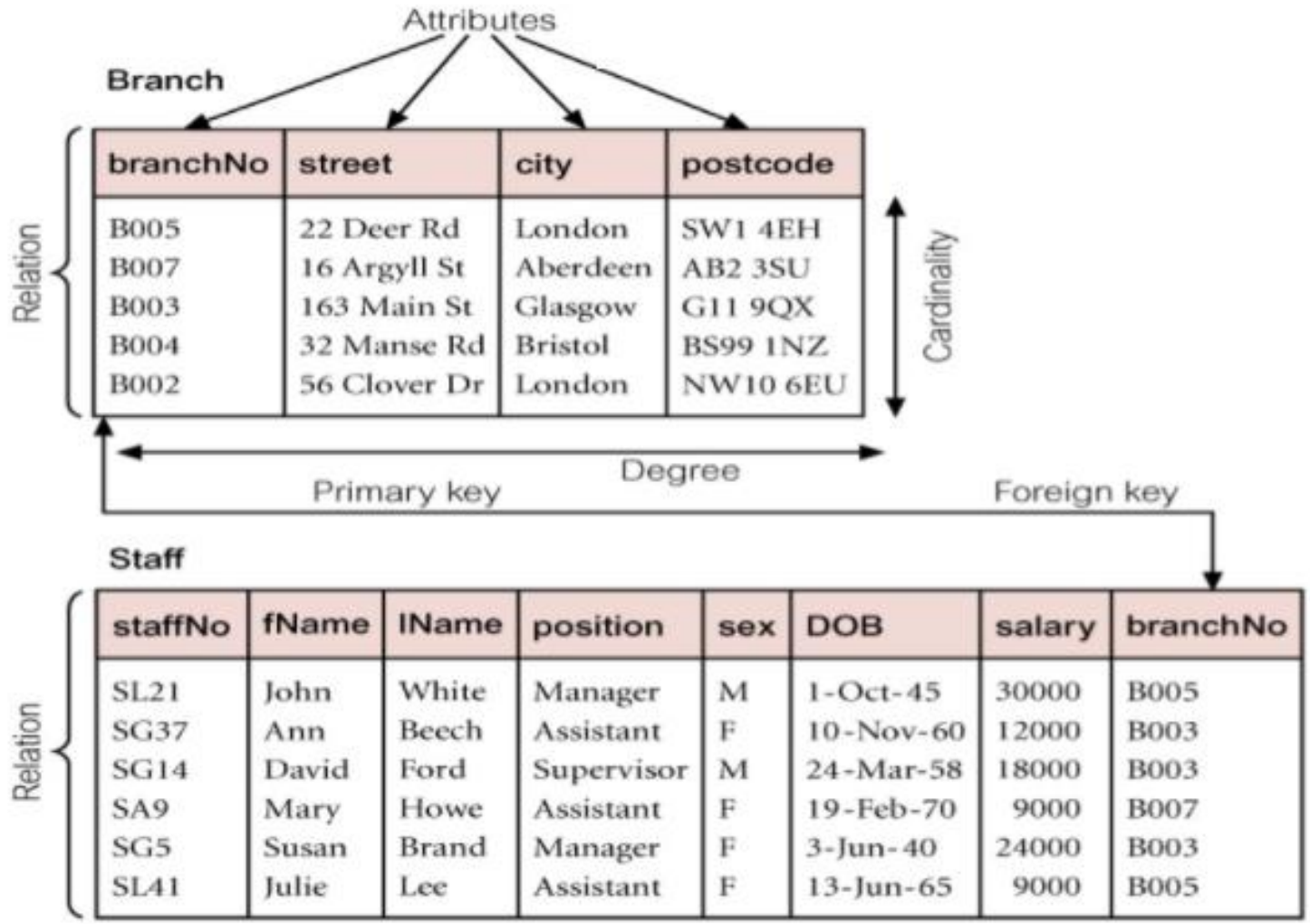
Course_id	Course_name	Duration (months)
78	Big Data	4
56	Algorithm	2

REFERENTIAL INTEGRITY

EMPLOYEE

EMP_ID	EMP_NAME	SALARY
123	Jack	30000
142	Harry	60000
164	John	20000
	Jackson	27000

Not allowed as primary key can't contain a NULL value


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

IMPLEMENTING REFERENTIAL INTEGRITY

- Leverage database functionality, such as **foreign key constraints** and triggers.

- Using this approach, when data is inserted, updated, or deleted in the database, it will enforce referential integrity.