DB Module

NgeeAnn

School of InfoComm Technology

SQL - Join Tables

Databases

Day 2

Prepared by Staff of School of ICT

Topics

- *Join* Tables
  - Cartesian Product
  - *Join* Conditions
  - Two Table *Join*
  - Multiple Table *Join*
  - Self *Join*

Recap - Concept of Keys

- Types of key:
  - Candidate Key
  - Primary Key
  - Alternate Key
  - Foreign Key

Candidate Key

- *Candidate Key*
  - An attribute or attributes that uniquely identifies each tuple in a relation

Example: Branch Relation

BranchNo	Address	TelNo
1	1,Tulip Plaza	61111111
2	2, Hibiscus Mall	62222222

Candidate Keys :

BranchNo, Address and TelNo

Primary & Alternate Key

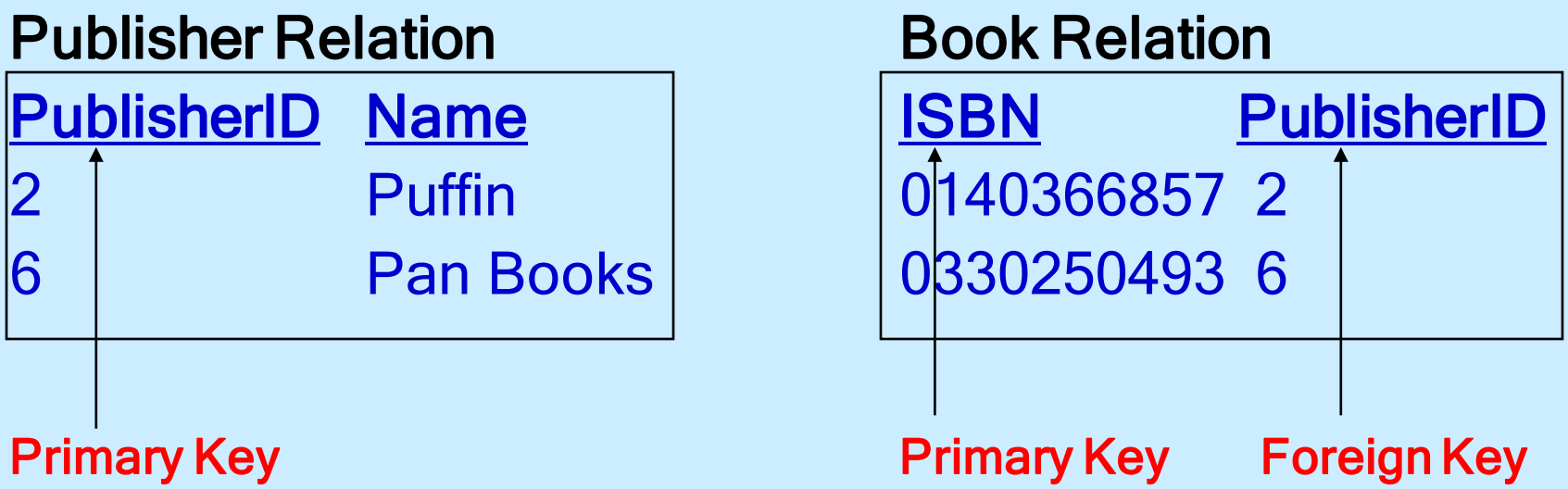
- *Primary Key*
  - The Candidate Key chosen to uniquely identify each tuple

    What is the Primary Key of Branch Relation?
  - Each relation has only *ONE* primary key
- *Alternate Keys*
  - Candidate Keys that are not chosen as Primary Key

    What are the Alternate Keys of Branch Relation?

Foreign Key

- Used in a relation to create relationship with other relation (or same) in a database



Composite Key

- A key that consists of *more than 1* attribute

Relation	Primary Key
Staff	StaffID
BookCopy	ISBN and CopyNo

Join Tables

- Sometimes query needs data from *more than one* table.

List ISBN & title of all books and their corresponding publisher's name

- ISBN, title from which table?
- Publisher's name from which table?

Join Tables

```
SELECT ISBN, Title, Name
FROM Book, Publisher
```

- Does the above query produce correct results?

Cartesian Product (Cross Join)

SELECT ISBN, Title, Name

FROM Book, Publisher

The result is the product of two tables

→ *Cartesian Product*

Each row in the *Book* table is *joined* with each row in the *Publisher* table to make up a new row in the query output

Cartesian Product

Book (2 rows)

ISBN	….	PublisherID	….
11111111111	…	1	…
22222222222	…	2	…

Publisher (3 rows)

PublisherID	Name
1	PrenticeHall
2	Thomson
3	Sams

Query Output (no. of rows =)

ISBN	….	PublisherID	….	PublisherID	Name
11111111111	…	1	…	1	PrenticeHall
11111111111	…	1	…	2	Thomson
11111111111	…	1	…	3	Sams
22222222222	…	2	…	1	PrenticeHall
22222222222	…	2	…	2	Thomson
22222222222	…	2	…	3	Sams

Eliminate Non Meaningful Rows

Query Output :

ISBN	….	PublisherID	….	PublisherID	Name
1111111111	…	1	…	1	PrenticeHall
1111111111	…	1	…	2	Thomson
1111111111	…	1	…	3	Sams
2222222222	…	2	…	1	PrenticeHall
2222222222	…	2	…	2	Thomson
2222222222	…	2	…	3	Sams

- Which rows are not meaningful?

Eliminate Non Meaningful Rows

Correct Query Output :

ISBN	….	PublisherID	…	PublisherID	Name
1111111111		1		1	PrenticeHall
2222222222		2		2	Thomson

Conditions for meaningful rows:

- The 2 tables must have a common attribute
  - What is the common attribute? PublisherID
  - The common attribute? Foreign Key of *BOOK*
  - The common attribute? Primary Key of *Publisher*

Inner Join

```
SELECT Book.*, Publisher.*
FROM Book INNER JOIN Publisher
ON Book.PublisherID = Publisher.PublisherID
```

- Where is the **join** condition?
- Why prefix **column** name with **table** name?
- Must every column name be prefixed by table name?

Table Aliases

- *Shorthand* for table names
- To *reduce* typing
- Active for *current* SQL statement only

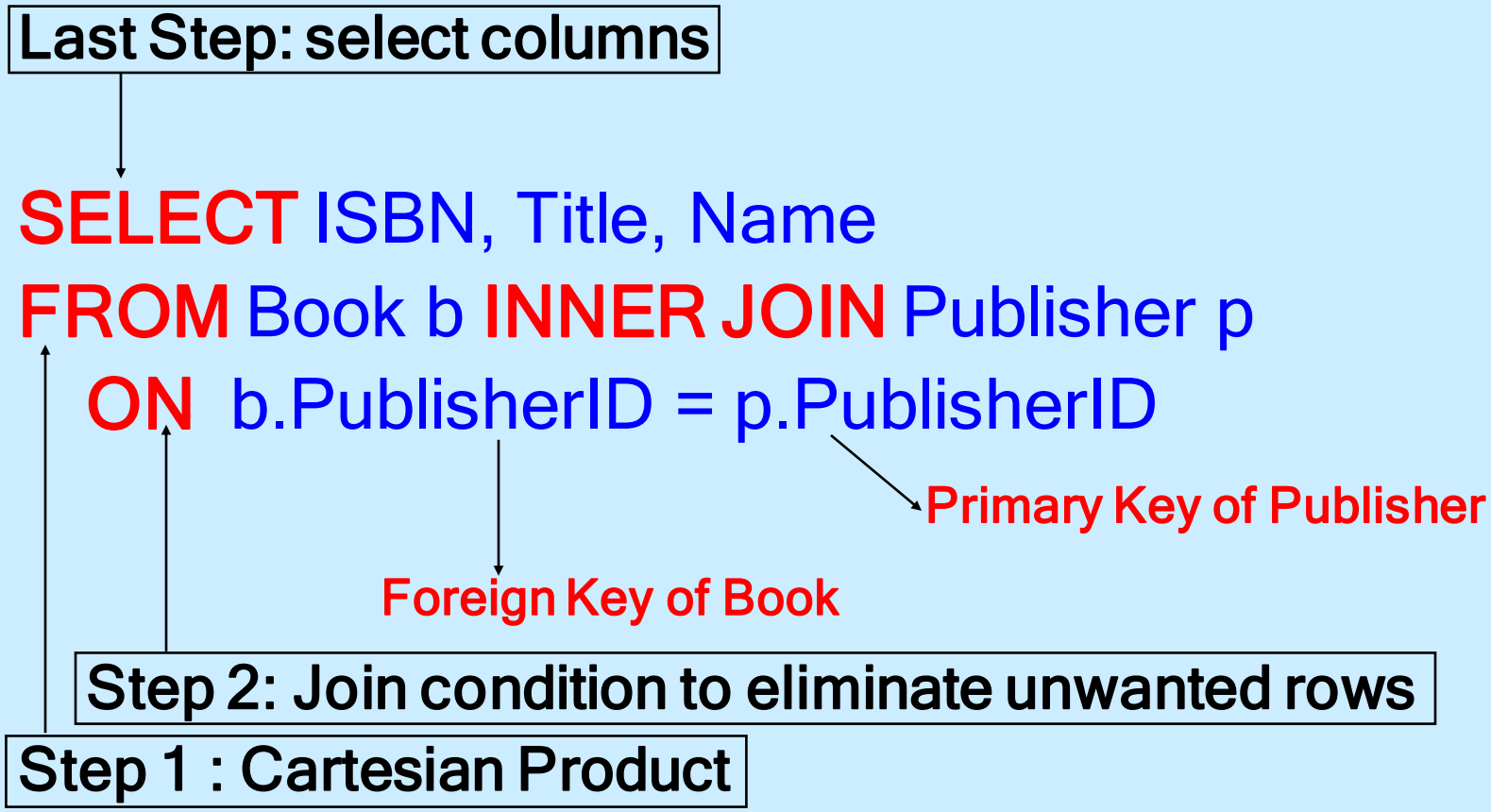
Table Aliases

```
SELECT book.*, Publisher.*
FROM Book INNER JOIN Publisher
ON Book.PublisherID = Publisher.
   PublisherID
```

Replaced by :

```
SELECT b.*, p.*
FROM Book b INNER JOIN Publisher p
ON  b.PublisherID = p.PublisherID
```

Steps in Joining Tables



Three Table Join

- **List ISBN, title for all books with corresponding publisher name & book category description**

  - *ISBN*, *Title* from which table ?
  - *Publisher Name* from which table ?
  - *Book Category Description* from which table ?

  - What is the Primary Key of *Publisher* ?
  - What is the Primary Key of *BookCategory* ?
  - What are the Foreign Keys of *Book* ?

Three Table Join

- List ISBN, title for all books with corresponding publisher name & book category description

```
SELECT ISBN, Title, Name, Description
FROM Book b
INNER JOIN Publisher p
   ON b.PublisherID = p.PublisherID
INNER JOIN BookCategory bc
   ON  b.BookCat = bc.BookCat
```

Self Join

- A table can be joined to itself

List names of all staff & names of their supervisors

```
SELECT s.Name, sup.Name
FROM Staff s INNER JOIN Staff sup
ON s.SupervisorID = sup.StaffID
```

Summary
Step in Joining Tables