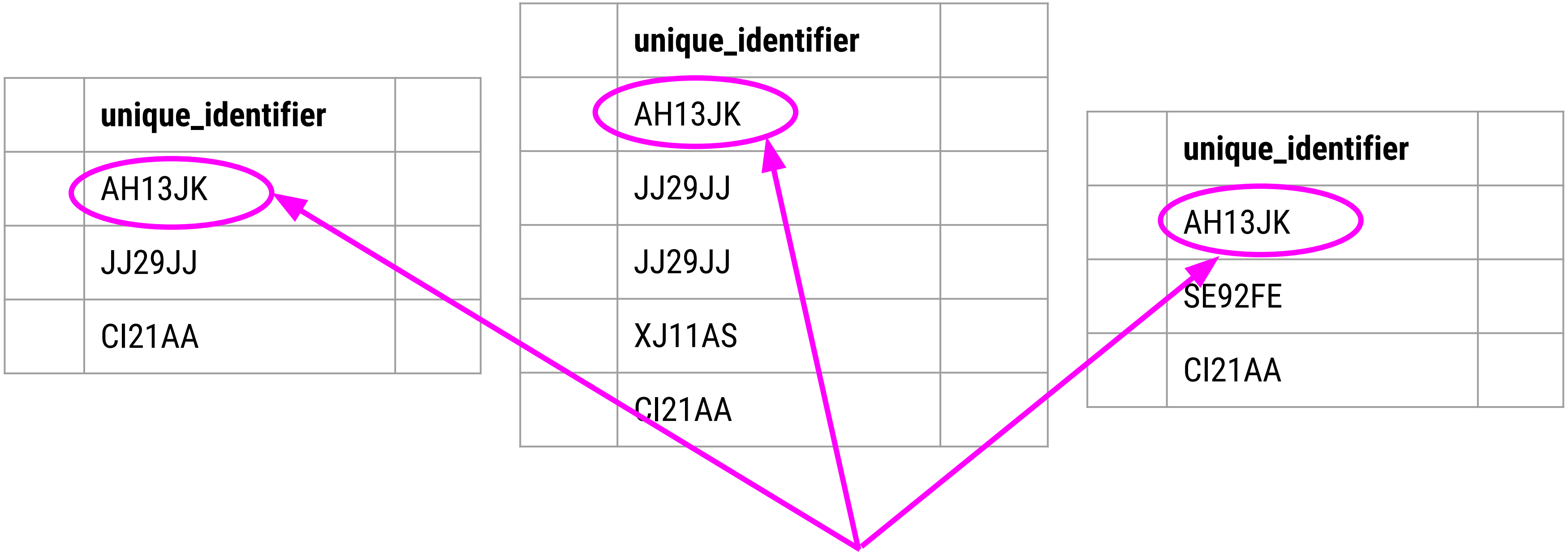
Relational Data

Getting Data

Three tables of information



	unique_identifier	
	AH13JK	
	JJ29JJ	
	CI21AA	

	unique_identifier	
	AH13JK	
	JJ29JJ	
	JJ29JJ	
	XJ11AS	
	CI21AA	

	unique_identifier	
	AH13JK	
	SE92FE	
	CI21AA	

entries are *related* to one another by their **unique identifier**

restaurant

name	id	address	type
Taco Stand	AH13JK	1 Main St.	Mexican
Pho Place	**JJ29JJ**	192 Street Rd.	Vietnamese
Taco Stand	XJ11AS	18 W. East St.	Fusion
Pizza Heaven	CI21AA	711 K Ave.	Italian

health inspections

name	id	inspection _date	inspector	score
Taco Stand	AH13JK	2018-08-21	Sheila	97
Pho Place	**JJ29JJ**	2018-03-12	D’eonte	98
Pho Place	**JJ29JJ**	2018-01-02	Monica	66
Taco Stand	XJ11AS	2018-12-16	Mark	43
Pizza Heaven	CI21AA	2018-08-21	Anh	99

rating

name	id	stars
Taco Stand	AH13JK	4.9
Pho Place	**JJ29JJ**	4.8
Taco Stand	XJ11AS	4.2
Pizza Heaven	CI21AA	4.7

Why relational data?

1. Efficient Data Storage
2. Avoids Ambiguity
3. Increases Data Privacy

restaurant

name	id	address	type
Taco Stand	AH13JK	1 Main St.	Mexican
Pho Place	**JJ29JJ**	192 Street Rd.	Vietnamese
Taco Stand	XJ11AS	18 W. East St.	Fusion
Pizza Heaven	CI21AA	711 K Ave.	Italian

Two different restaurants with the same name!

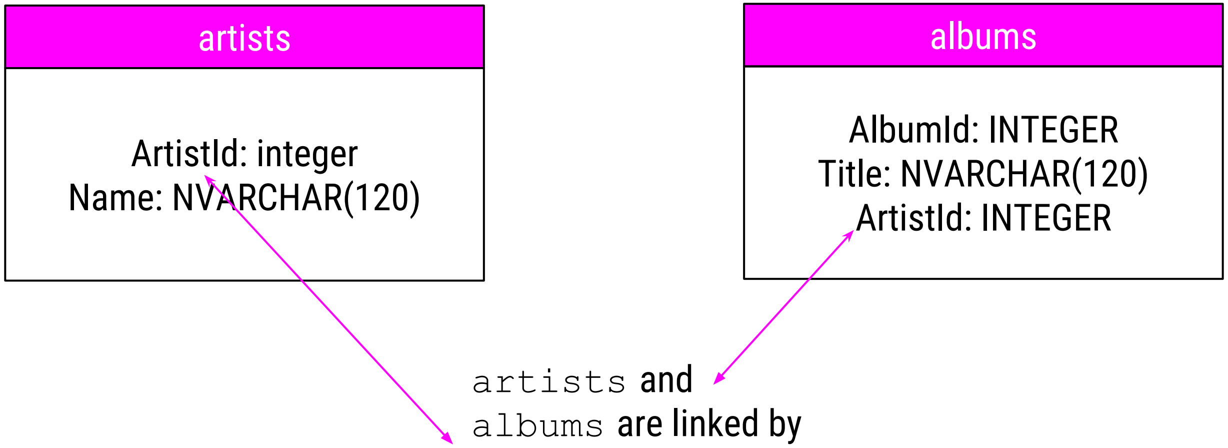
health inspections

name	id	inspection_date	inspector	score
Taco Stand	AH13JK	2018-08-21	Sheila	97
Pho Place	**JJ29JJ**	2018-03-12	D’eonte	98
Pho Place	**JJ29JJ**	2018-01-02	Monica	66
Taco Stand	XJ11AS	2018-12-16	Mark	43
Pizza Heaven	CI21AA	2018-08-21	Anh	99

rating

name	id	stars
Taco Stand	AH13JK	4.9
Pho Place	**JJ29JJ**	4.8
Taco Stand	XJ11AS	4.2
Pizza Heaven	CI21AA	4.7

chinook.db



Contains artists' name and ArtistId

artists

ArtistId: integer
Name: NVARCHAR(120)

albums

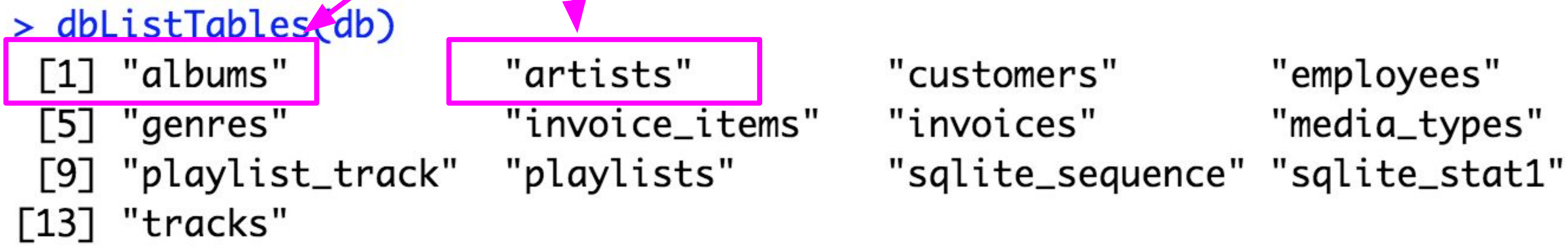
AlbumId: INTEGER
Title: NVARCHAR(120)
ArtistId: INTEGER

Contains data about a list of tracks. Each album corresponds to a single artist. But, a single artist can have many albums

`artists` and `albums` are linked by `ArtistId`

```
## install and load packages
## this may take a minute or two
install.packages("RSQLite")
library(RSQLite)
library(httr)

## specify driver
sqlite <- dbDriver("SQLite")

## download data
url <-
"http://www.sqlitetutorial.net/wp-content/uploads/2018/03/chinook
.zip"
GET(url, write_disk(tf <- tempfile(fileext = ".zip")))
unzip(tf)

## Connect to Database
db <- dbConnect(sqlite, 'chinook.db')

## list tables in database
dbListTables(db)
```

The two tables we'll work with throughout this lesson!



```
> dbListTables(db)
 [1] "albums"          "artists"         "customers"       "employees"
 [5] "genres"          "invoice_items"   "invoices"        "media_types"
 [9] "playlist_track"  "playlists"       "sqlite_sequence" "sqlite_stat1"
[13] "tracks"
```

```
## install and load packages
install.packages("dbplyr")
library(dbplyr)
library(dplyr)

## get two tables
albums <- tbl(db, "albums")
artists <- tbl(db, "artists")
```

artists

ArtistId	Name
1	AC/DC
2	Accept
3	Aerosmith

albums

AlbumId	Title	ArtistId
1	For Those About To Rock We Salute You	1
2	Balls to the Wall	2
3	Restless and Wild	2
6	Jagged Little Pill	4

Inner Join

artists	
ArtistId	**Name**
1	AC/DC
2	Accept
3	Aerosmith

albums		
AlbumId	**Title**	**ArtistId**
1	For Those About To Rock We Salute You	1
2	Balls to the Wall	2
3	Restless and Wild	2
6	Jagged Little Pill	4

Inner Join: include any row in both tables

artists

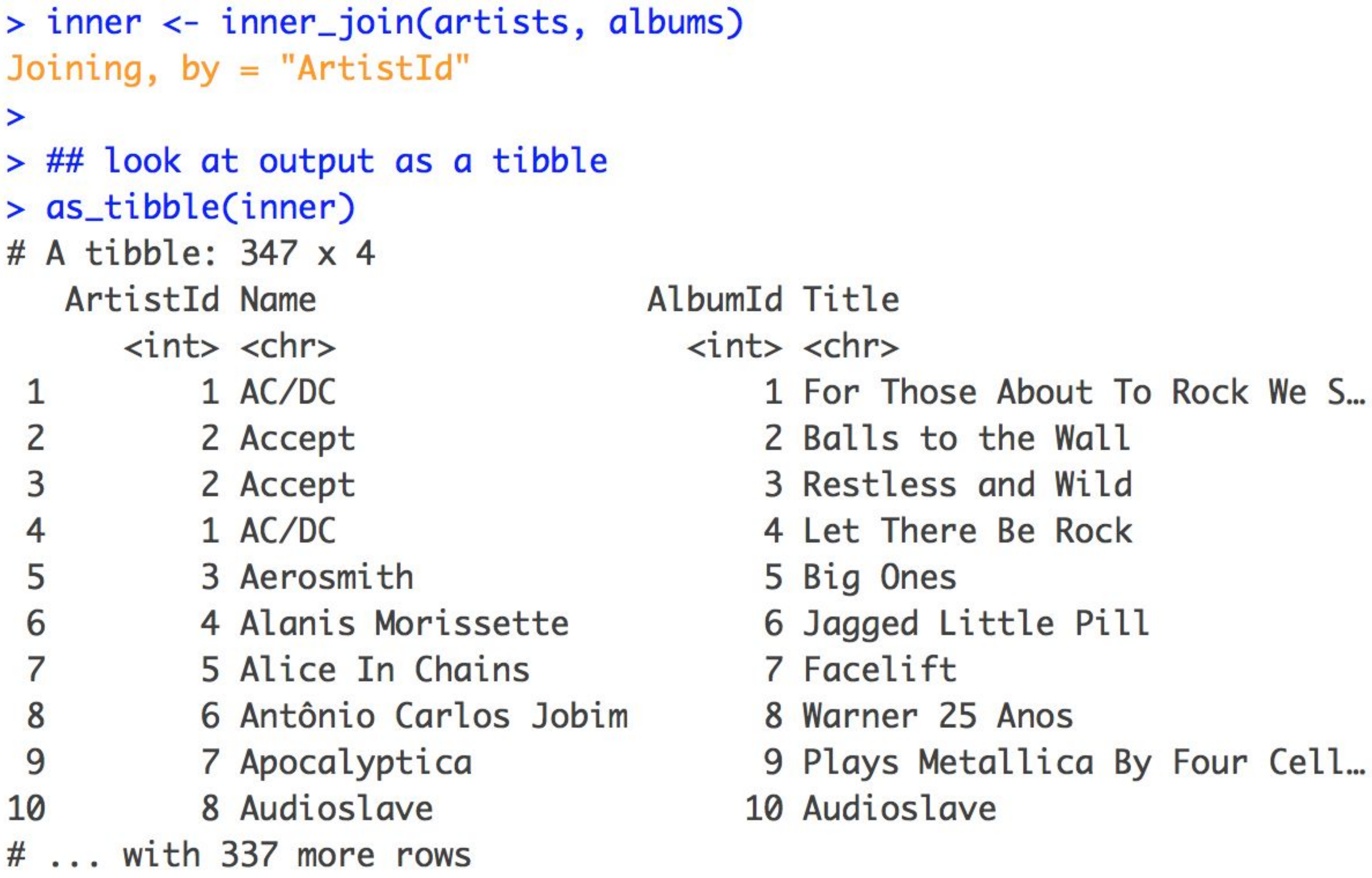
ArtistId	Name
1	AC/DC
2	Accept
3	Aerosmith

albums

AlbumId	Title	ArtistId
1	For Those About To Rock We Salute You	1
2	Balls to the Wall	2
3	Restless and Wild	2
6	Jagged Little Pill	4

`inner_join()`

ArtistId	Name	AlbumId	Title
1	AC/DC	1	For Those About To Rock We Salute You
2	Accept	2	Balls to the Wall
2	Accept	3	Restless and Wild



```
> inner <- inner_join(artists, albums)
Joining, by = "ArtistId"
>
> ## look at output as a tibble
> as_tibble(inner)
# A tibble: 347 x 4
   ArtistId Name                 AlbumId Title
      <int> <chr>                  <int> <chr>
 1        1 AC/DC                      1 For Those About To Rock We S…
 2        2 Accept                     2 Balls to the Wall
 3        2 Accept                     3 Restless and Wild
 4        1 AC/DC                      4 Let There Be Rock
 5        3 Aerosmith                  5 Big Ones
 6        4 Alanis Morissette          6 Jagged Little Pill
 7        5 Alice In Chains            7 Facelift
 8        6 Antônio Carlos Jobim       8 Warner 25 Anos
 9        7 Apocalyptica               9 Plays Metallica By Four Cell…
10        8 Audioslave                10 Audioslave
# ... with 337 more rows
```

Left Join

artists

ArtistId	Name
1	AC/DC
2	Accept
3	Aerosmith

albums

AlbumId	Title	ArtistId
1	For Those About To Rock We Salute You	1
2	Balls to the Wall	2
3	Restless and Wild	2
6	Jagged Little Pill	4

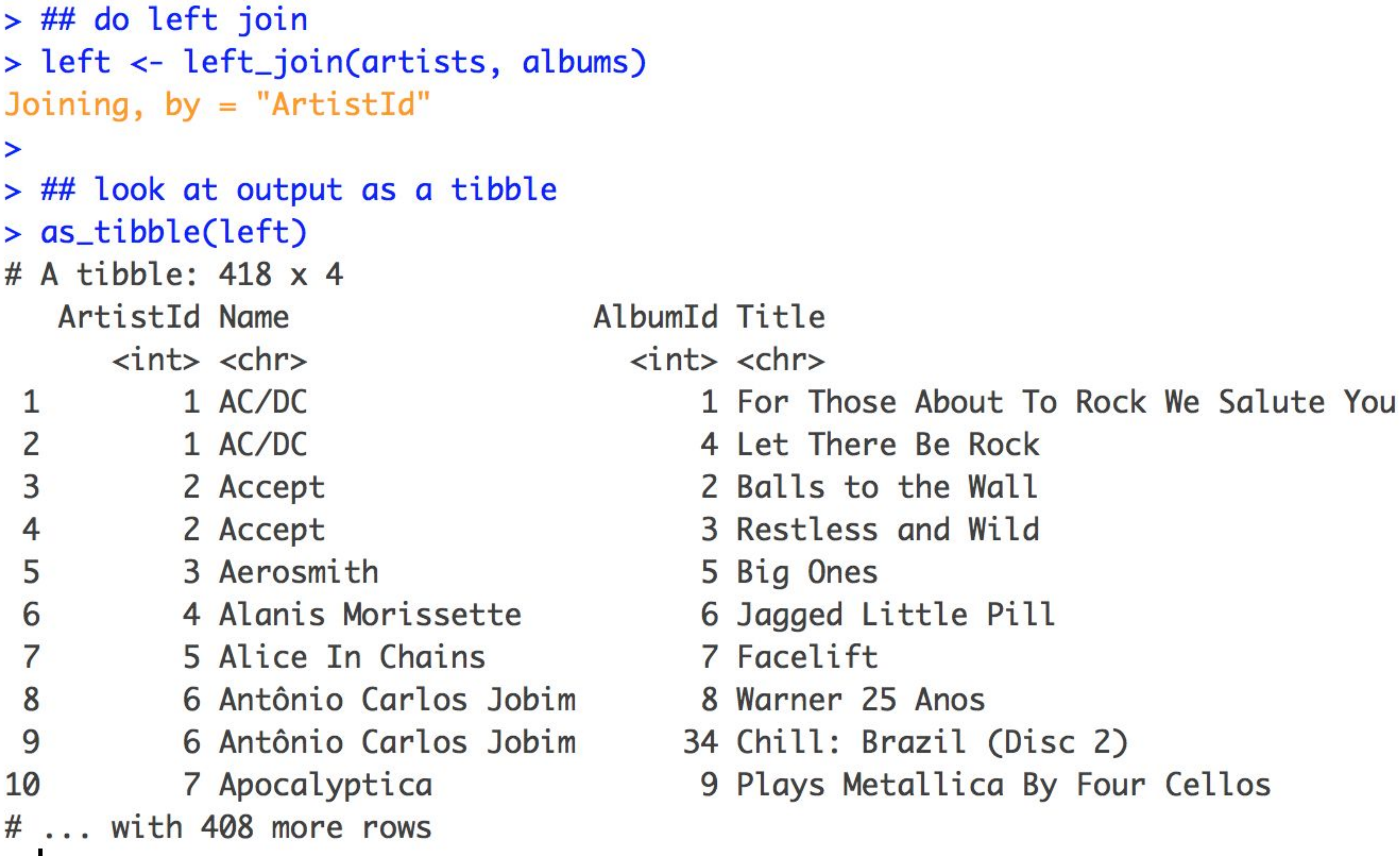
Left Join: include all rows in first table

albums		
AlbumId	**Title**	**ArtistId**
1	For Those About To Rock We Salute You	1
2	Balls to the Wall	2
3	Restless and Wild	2
6	Jagged Little Pill	4

artists	
ArtistId	**Name**
1	AC/DC
2	Accept
3	Aerosmith

`left_join()`

ArtistId	Name	AlbumId	Title
1	AC/DC	1	For Those About To Rock We Salute You
2	Accept	2	Balls to the Wall
2	Accept	3	Restless and Wild
3	Aerosmith	NA	NA



```
> ## do left join
> left <- left_join(artists, albums)
Joining, by = "ArtistId"
>
> ## look at output as a tibble
> as_tibble(left)
# A tibble: 418 x 4
   ArtistId Name                 AlbumId Title
      <int> <chr>                  <int> <chr>
 1        1 AC/DC                      1 For Those About To Rock We Salute You
 2        1 AC/DC                      4 Let There Be Rock
 3        2 Accept                     2 Balls to the Wall
 4        2 Accept                     3 Restless and Wild
 5        3 Aerosmith                  5 Big Ones
 6        4 Alanis Morissette          6 Jagged Little Pill
 7        5 Alice In Chains            7 Facelift
 8        6 Antônio Carlos Jobim       8 Warner 25 Anos
 9        6 Antônio Carlos Jobim      34 Chill: Brazil (Disc 2)
10        7 Apocalyptica               9 Plays Metallica By Four Cellos
# ... with 408 more rows
```

Right Join

artists

ArtistId	Name
1	AC/DC
2	Accept
3	Aerosmith

albums

AlbumId	Title	ArtistId
1	For Those About To Rock We Salute You	1
2	Balls to the Wall	2
3	Restless and Wild	2
6	Jagged Little Pill	4

Right Join: include all rows in 2nd table

artists

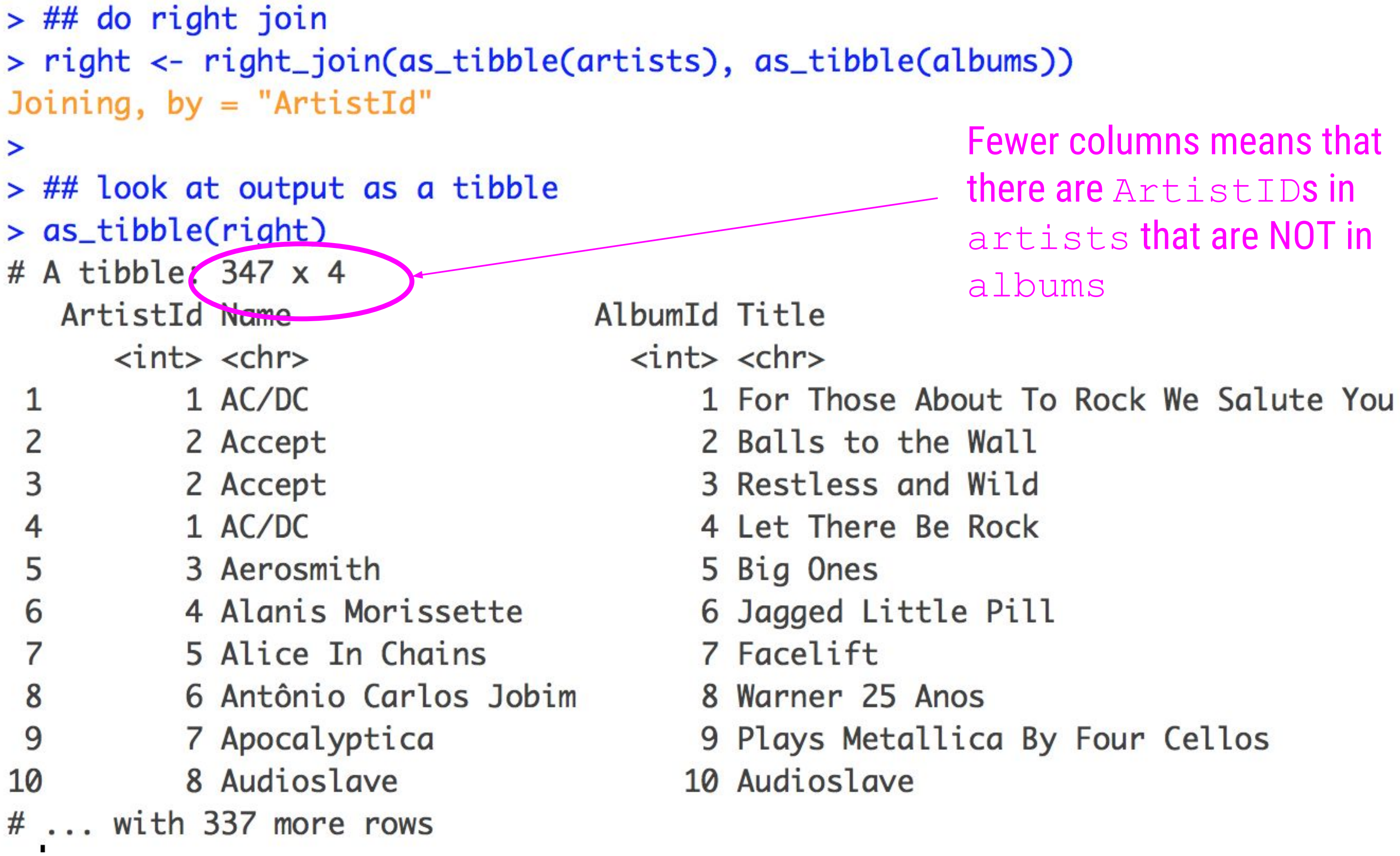
ArtistId	Name
1	AC/DC
2	Accept
3	Aerosmith

albums

AlbumId	Title	ArtistId
1	For Those About To Rock We Salute You	1
2	Balls to the Wall	2
3	Restless and Wild	2
6	Jagged Little Pill	4

`right_join()`

ArtistId	Name	AlbumId	Title
1	AC/DC	1	For Those About To Rock We Salute You
2	Accept	2	Balls to the Wall
2	Accept	3	Restless and Wild
4	NA	6	Jagged Little Pill



```
> ## do right join
> right <- right_join(as_tibble(artists), as_tibble(albums))
Joining, by = "ArtistId"
>
> ## look at output as a tibble
> as_tibble(right)
# A tibble: 347 x 4
   ArtistId Name                   AlbumId Title
      <int> <chr>                    <int> <chr>
 1        1 AC/DC                        1 For Those About To Rock We Salute You
 2        2 Accept                       2 Balls to the Wall
 3        2 Accept                       3 Restless and Wild
 4        1 AC/DC                        4 Let There Be Rock
 5        3 Aerosmith                    5 Big Ones
 6        4 Alanis Morissette            6 Jagged Little Pill
 7        5 Alice In Chains              7 Facelift
 8        6 Antônio Carlos Jobim         8 Warner 25 Anos
 9        7 Apocalyptica                 9 Plays Metallica By Four Cellos
10        8 Audioslave                  10 Audioslave
# ... with 337 more rows
```

Fewer columns means that there are `ArtistID`s in `artists` that are NOT in `albums`

Full Join

artists	
ArtistId	**Name**
1	AC/DC
2	Accept
3	Aerosmith

albums		
AlbumId	**Title**	**ArtistId**
1	For Those About To Rock We Salute You	1
2	Balls to the Wall	2
3	Restless and Wild	2
6	Jagged Little Pill	4

Full Join: include any row in *either* table

artists

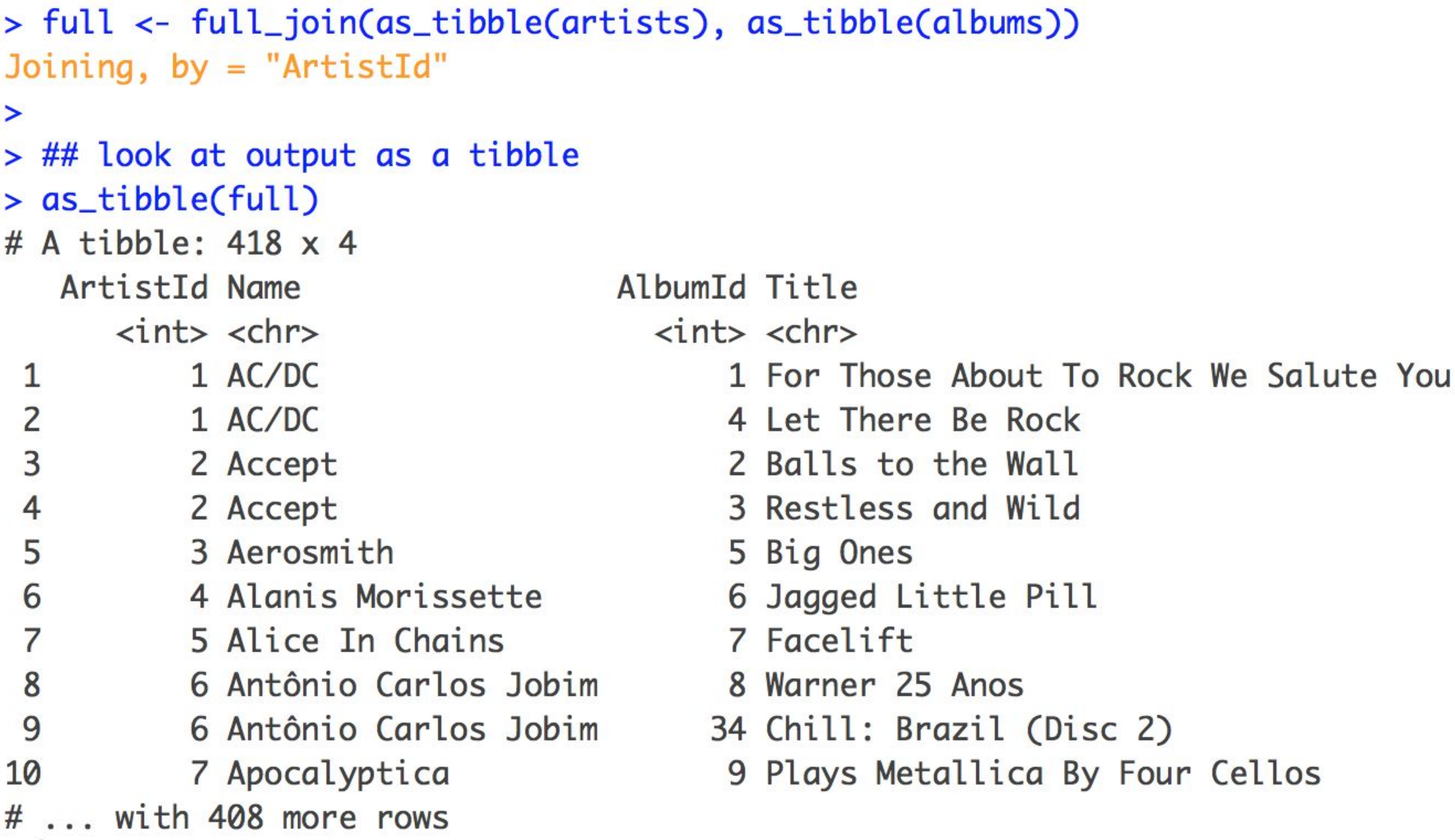
ArtistId	Name
1	AC/DC
2	Accept
3	Aerosmith

albums

AlbumId	Title	ArtistId
1	For Those About To Rock We Salute You	1
2	Balls to the Wall	2
3	Restless and Wild	2
6	Jagged Little Pill	4

`full_join()`

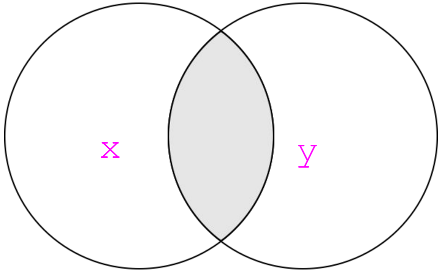
ArtistId	Name	AlbumId	Title
1	AC/DC	1	For Those About To Rock We Salute You
2	Accept	2	Balls to the Wall
2	Accept	3	Restless and Wild
3	Aerosmith	NA	NA
4	NA	6	Jagged Little Pill



```
> full <- full_join(as_tibble(artists), as_tibble(albums))
Joining, by = "ArtistId"
>
> ## look at output as a tibble
> as_tibble(full)
# A tibble: 418 x 4
   ArtistId Name                     AlbumId Title
      <int> <chr>                      <int> <chr>
 1        1 AC/DC                          1 For Those About To Rock We Salute You
 2        1 AC/DC                          4 Let There Be Rock
 3        2 Accept                         2 Balls to the Wall
 4        2 Accept                         3 Restless and Wild
 5        3 Aerosmith                      5 Big Ones
 6        4 Alanis Morissette              6 Jagged Little Pill
 7        5 Alice In Chains                7 Facelift
 8        6 Antônio Carlos Jobim           8 Warner 25 Anos
 9        6 Antônio Carlos Jobim          34 Chill: Brazil (Disc 2)
10        7 Apocalyptica                   9 Plays Metallica By Four Cellos
# ... with 408 more rows
```

inner

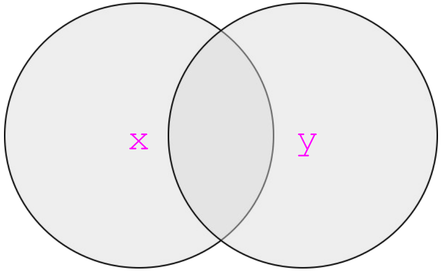
Include any row in both tables



`inner_join(x, y)`

full

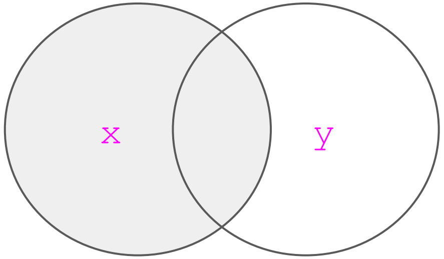
Include any row in either table



`full_join(x, y)`

left

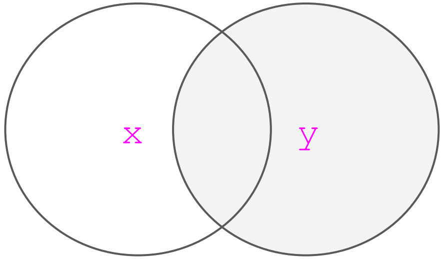
Include all rows in 1st table



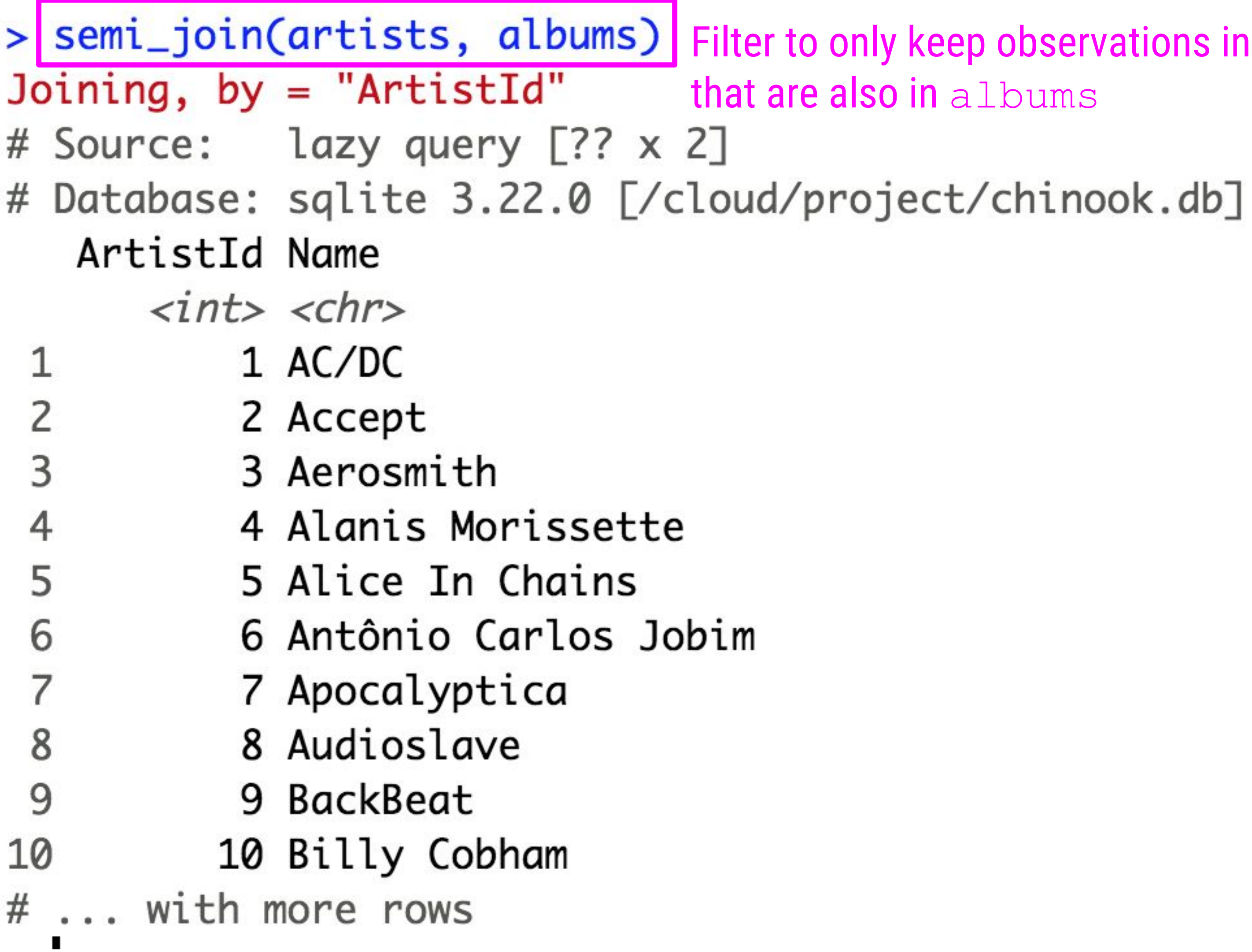
`left_join(x, y)`

right

Include all rows in 2nd table

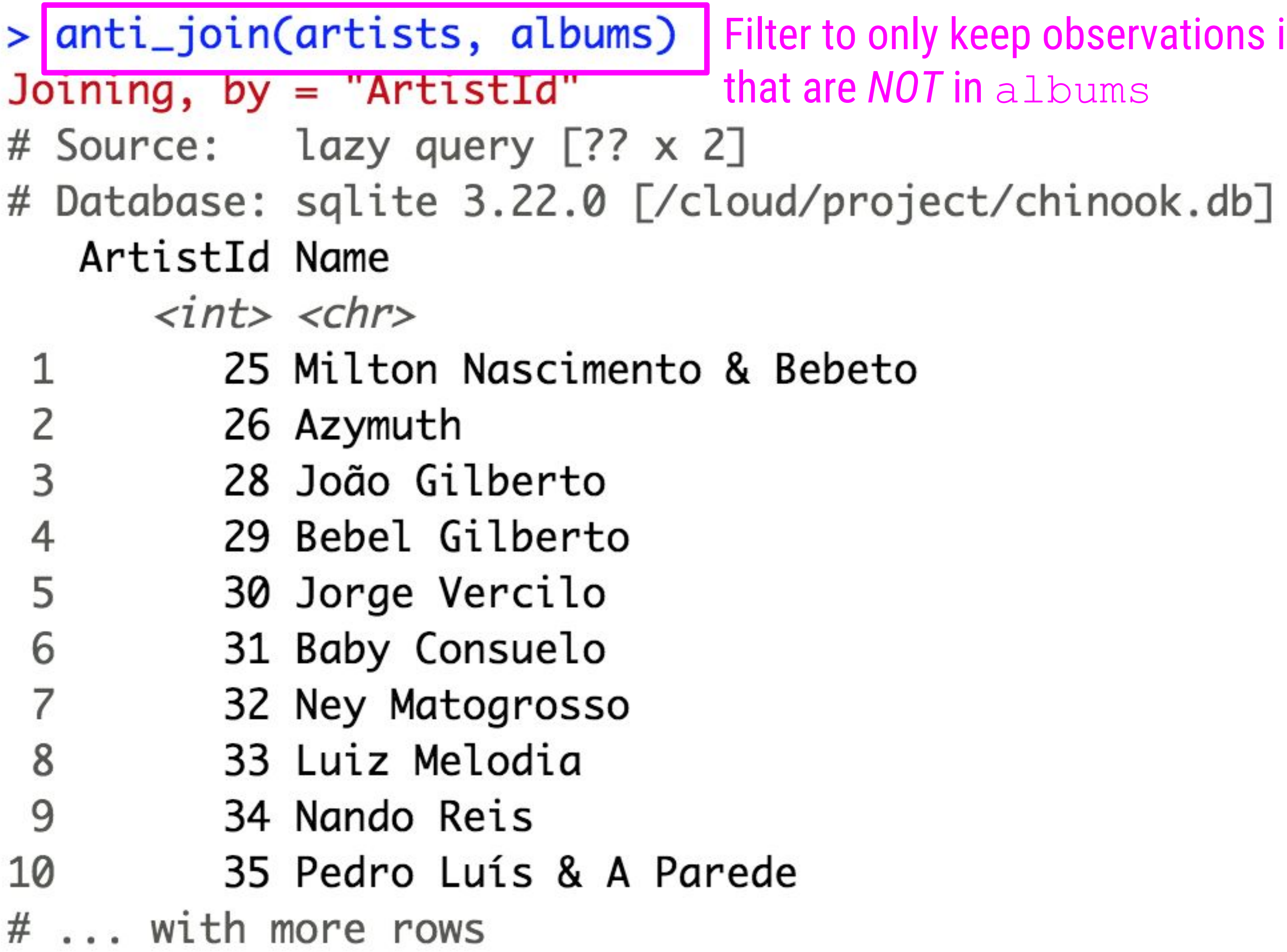


`right_join(x, y)`



```
> semi_join(artists, albums)
Joining, by = "ArtistId"
# Source:   lazy query [?? x 2]
# Database: sqlite 3.22.0 [/cloud/project/chinook.db]
   ArtistId Name
      <int> <chr>
 1        1 AC/DC
 2        2 Accept
 3        3 Aerosmith
 4        4 Alanis Morissette
 5        5 Alice In Chains
 6        6 Antônio Carlos Jobim
 7        7 Apocalyptica
 8        8 Audioslave
 9        9 BackBeat
10       10 Billy Cobham
# ... with more rows
```

Filter to only keep observations in `artists` that are also in `albums`



```
> anti_join(artists, albums)
Joining, by = "ArtistId"
# Source:   lazy query [?? x 2]
# Database: sqlite 3.22.0 [/cloud/project/chinook.db]
   ArtistId Name
      <int> <chr>
 1       25 Milton Nascimento & Bebeto
 2       26 Azymuth
 3       28 João Gilberto
 4       29 Bebel Gilberto
 5       30 Jorge Vercilo
 6       31 Baby Consuelo
 7       32 Ney Matogrosso
 8       33 Luiz Melodia
 9       34 Nando Reis
10       35 Pedro Luís & A Parede
# ... with more rows
```

Filter to only keep observations in `artists` that are *NOT* in `albums`

```
con <- DBI::dbConnect(RMySQL::MySQL(),
                      host = "database.host.com",
                      user = "janeeverydaydoe",
                      password =
rstudioapi::askForPassword("database_password")
)
```

Relational Data

- Relational Data & Databases
- Joins
  - Mutating Joins
    - inner_join
    - left_join
    - Right_join
    - full_join
  - Filtering Joins
    - semi_join
    - anti_join
- Connecting to a remote database server