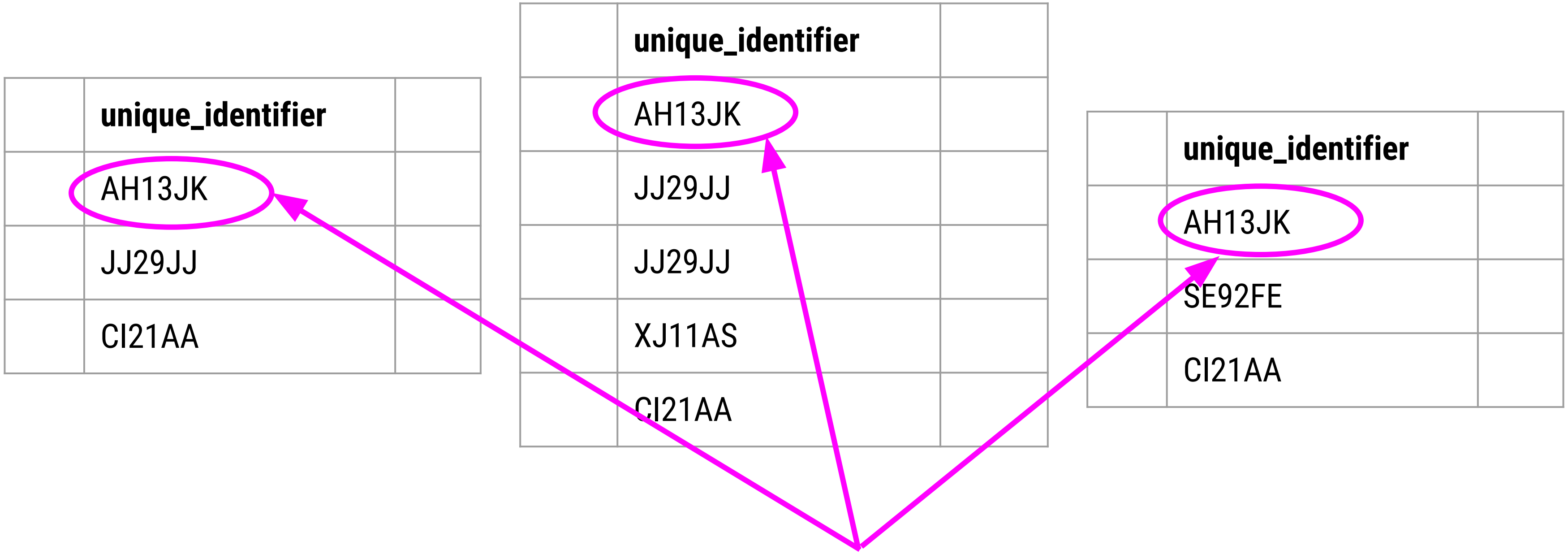
Getting data from databases

Getting and Cleaning 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

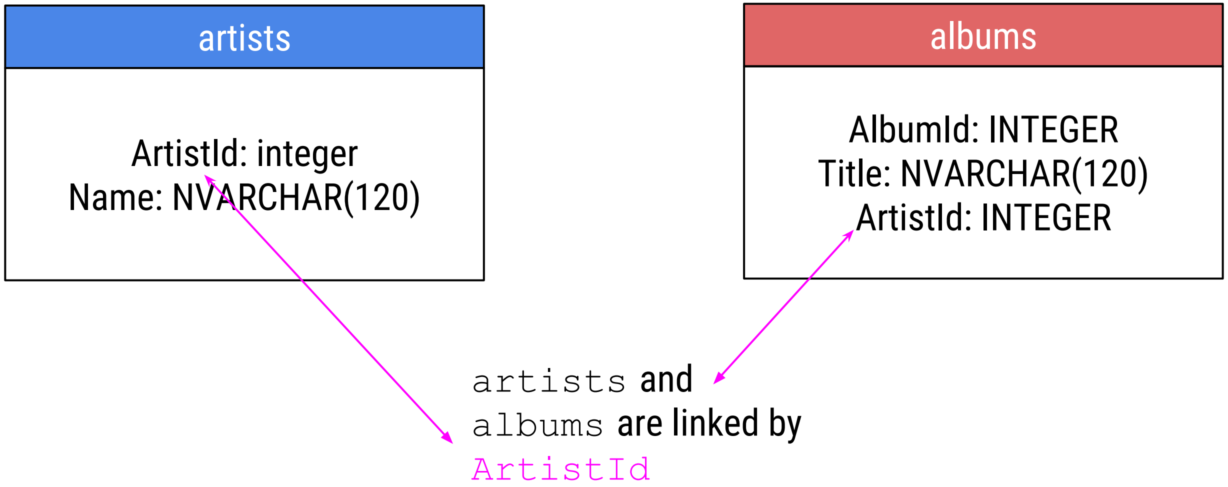
name	id	inspection _date	inspector	score
Taco Stand	AH13JK	2018-08-21	Sheila	97
Pho Place	**JJ29JJ**	2018-03-12	D’eonte	98
Pho Place	**JJ29JJ**	2018-01-02	Monica	66
Taco Stand	XJ11AS	2018-12-16	Mark	43
Pizza Heaven	CI21AA	2018-08-21	Anh	99

rating

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

Created by vectorpouch - Freepik.com

chinook.db



Contains artists' name and ArtistId

artists

ArtistId: integer
Name: NVARCHAR(120)

albums

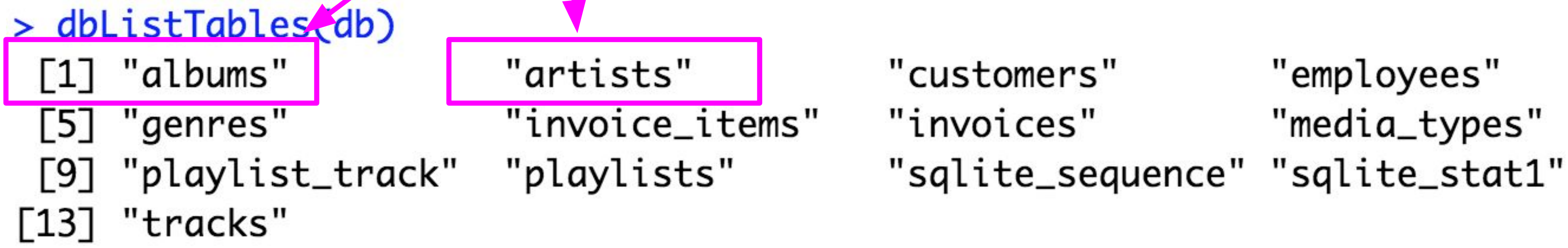
AlbumId: INTEGER
Title: NVARCHAR(120)
ArtistId: INTEGER

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

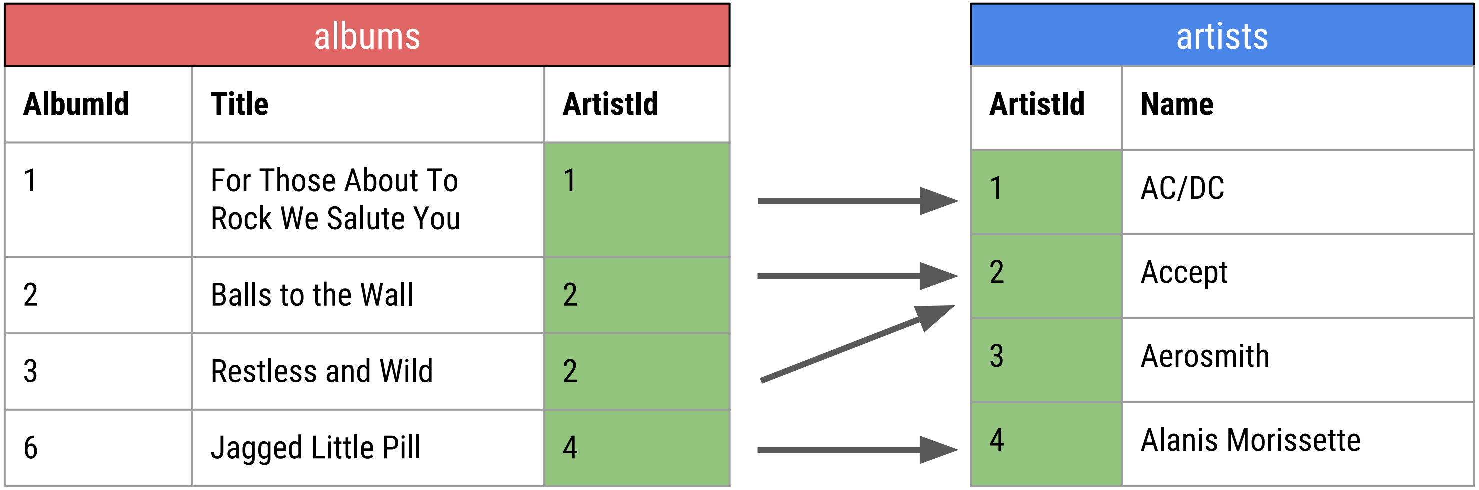
ArtistId	Name
1	AC/DC
2	Accept
3	Aerosmith
4	Alanis Morissette
5	Alice in Chains

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
4	Let there Be Rock	1
5	Big Ones	3
6	Jagged Little Pill	4

These two tables have the ArtistId column in common

Left join to create a single table with all albums and their artist



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
4	Alanis Morissette

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

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

Summarizing:
Getting data from databases

Getting and Cleaning Data