assignment3.Rmd

RuiCao_300423011

9/5/2019

Q1. a.

```
day1 <- as.Date('23-January-2001',format="%d-%B-%Y")
day1
```

```
## [1] "2001-01-23"
```

```
day2 <- as.Date('23/01/2001', format = "%d/%m/%Y")
day2
```

```
## [1] "2001-01-23"
```

```
day3 <- as.Date('01/23/01', format = "%m/%d/%y")
day3
```

```
## [1] "2001-01-23"
```

```
day4 <- as.Date('day 23 of January in the year 2001', format = "day %d of %B in the year %Y")
day4
```

```
## [1] "2001-01-23"
```

Q1.b walks on the moon: July 21, 1969, at 2:57 UTC

```
moon_walk <- as.POSIXct('July 21, 1969, 02:56', format="%B %d, %Y, %H:%M", tz = "UTC")
moon_walk
```

```
## [1] "1969-07-21 02:56:00 UTC"
```

```
format(moon_walk,usetz=TRUE, tz="ETC/GMT-12")
```

```
## [1] "1969-07-21 14:56:00 +12"
```

Q1.c

```
week_moon <- format(moon_walk, "%A")
week_moon
```

```
## [1] "Monday"
```

Q2.a.In a few sentences, give a brief explanation of what Primary and Foreign Keys are and how they differ.

In order for a table to qualify as a relational table it must have a primary key. The primary key: consists of one or more columns whose data contained within is used to uniquely identify each row in the table. A foreign key: is a set of one or more columns in a table that refers to the primary key in another table. A foreign key is that it is referring to a primary key

Unlike primary keys, foreign keys can contain duplicate values. Also, it is OK for them contain NULL values. A table is allowed to contain more than one foreign key, but just can have one primary key in one table.

Q2. b.Why are Primary and Foreign Keys an important part of a database?

A primary key ensures that data is unique in one column, but the foreign key is what makes the data stay consistent, as that is where the important data lies that needs to stay consistent and integral.

Q2. c.Identify the Primary and Foreign Keys (if any) in the the tables

artists : PK : ArtistId

invoices: PK: InvoicesId

FK: CustomerId

albums: PK: AlbumId

FK: ArtistId

tracks: PK: TrackId

FK: AlbumId, MediaTypeId, GenreId

Q2. d

```
library(DBI)
library(RSQLite)
chinook <- dbConnect(SQLite(), "chinook_data.sqlite")

albums_full <- read.csv("albums_full.csv", stringsAsFactors = FALSE)
artists_full <- read.csv("artists_full.csv", stringsAsFactors = FALSE)
custom <- read.csv("customers_full.csv", stringsAsFactors = FALSE)
genres_full <- read.csv("genres_full.csv", stringsAsFactors = FALSE)
inv_item <- read.csv("invoice_items_full.csv", stringsAsFactors = FALSE)
invoices_full <- read.csv("invoices_full.csv", stringsAsFactors = FALSE)
tracks_full <- read.csv("tracks_full.csv", stringsAsFactors = FALSE)

dbWriteTable(chinook, "albums", albums_full, overwrite = TRUE)
dbWriteTable(chinook, "artists", artists_full, overwrite = TRUE)
dbWriteTable(chinook, "custom", custom, overwrite = TRUE)
dbWriteTable(chinook, "genres", genres_full, overwrite = TRUE)
dbWriteTable(chinook, "invoice_item", inv_item, overwrite = TRUE)
dbWriteTable(chinook, "tracks", tracks_full, overwrite = TRUE)
```

```
select count(distinct GenreId) as Dis_GenderId
from genres
```

1 records

Dis_GenderId
25

e.

```
drop table if exists trac2
```

```
create table trac2 as
select GenreId, Name, total_tracks
from (
select * from genres g
inner join (
select GenreId, count(trackId) as total_tracks
from tracks
group by GenreId
) as tot_g on  g.GenreId = tot_g.GenreId
order by total_tracks DESC
)
```

```
select * from trac2
```

Displaying records 1 - 10

GenreId	Name	total_tracks
1	Rock	1297
7	Latin	579
3	Metal	374
4	Alternative & Punk	332
2	Jazz	130
19	TV Shows	93
6	Blues	81
24	Classical	74
21	Drama	64
14	R&B/Soul	61

f.

```
select * from trac2
where total_tracks = 0
```

0 records

GenreId	Name	total_tracks

Thus, there is no genre in the database with no track.

g.

```
SELECT Name, total_tracks,
       ROUND(total_tracks *100.0/(SELECT sum(total_tracks) FROM trac2 ),1) as PCT
FROM  trac2
order by PCT DESC
```

Displaying records 1 - 10

Name	total_tracks	PCT
Rock	1297	37.0
Latin	579	16.5
Metal	374	10.7
Alternative & Punk	332	9.5
Jazz	130	3.7
TV Shows	93	2.7
Blues	81	2.3
Classical	74	2.1
Drama	64	1.8
R&B/Soul	61	1.7

h.

```
SELECT Name, total_tracks,
       ROUND(total_tracks *100.0/(SELECT sum(total_tracks) FROM trac2 ),1) as PCT
FROM  trac2
where PCT >= 10
order by PCT DESC
```

3 records

Name	total_tracks	PCT
Rock	1297	37.0

Name	total_tracks	PCT
Latin	579	16.5
Metal	374	10.7

Q3.a

```
drop table if exists art2
```

```
create table art2 as
select ArtistId, count(*) as NumAlbums
from albums
group by ArtistId
order by ArtistId
```

```
drop table if exists ArtistInfo
```

```
create table ArtistInfo as
select a.ArtistId, Name as ArtistName, NumAlbums
from artists a
inner join art2 ar on ar.ArtistId = a.ArtistId
where NumAlbums >= 1
```

```
select * from ArtistInfo
```

Displaying records 1 - 10

ArtistId	ArtistName	NumAlbums
1	AC/DC	2
2	Accept	2
3	Aerosmith	1
4	Alanis Morissette	1
5	Alice In Chains	1
6	AntÃ´nio Carlos Jobim	2
7	Apocalyptica	1
8	Audioslave	3
9	BackBeat	1
10	Billy Cobham	1

b.

```
drop table if exists ArtistInfo
```

```
create table ArtistInfo(ArtistId integer, ArtistName text, NumAlbums integer)
```

```
insert into ArtistInfo( ArtistId, ArtistName, NumAlbums)
select a.ArtistId, Name as ArtistName, NumAlbums
from artists a
left join
  (select ArtistId, count(*) as NumAlbums
  from albums
  group by ArtistId
  order by ArtistId) as al2
on al2.ArtistId = a. ArtistId
where NumAlbums >= 1
```

```
select * from ArtistInfo
```

Displaying records 1 - 10

ArtistId	ArtistName	NumAlbums
1	AC/DC	2
2	Accept	2
3	Aerosmith	1
4	Alanis Morissette	1
5	Alice In Chains	1
6	AntÃ´nio Carlos Jobim	2
7	Apocalyptica	1
8	Audioslave	3
9	BackBeat	1
10	Billy Cobham	1

c.

```
drop table if exists Artists2
```

```
create table Artists2( ArtistId integer, Name text, Primary Key(ArtistId))
```

```
insert into Artists2
select * from artists
```

```
drop table if exists Gen2
```

```
create table Gen2( GenreId integer, Name text, PRIMARY KEY (GenreId))
```

```
insert into Gen2
select * from genres
```

```
drop table if exists ArtistGenres
```

```
create table ArtistGenres ( ArtistId integer, ArtistName text, GenreId integer, NumTracks intege
r,
primary key (ArtistId, GenreId),
foreign key (ArtistId) references Artists2 (ArtistId)
on delete restrict on update cascade,
foreign key (GenreId) references Gen2 (GenreId) on update cascade on delete restrict)
```

```
pragma foreign_key = on
```

```
drop table if exists art2
```

```
create table art2 as
select ArtistId, Name as ArtistName, AlbumId, Title as AlbumTitle
from(
select * from artists a
inner join albums al
on a.ArtistId = al.ArtistId)
```

```
insert into ArtistGenres (ArtistId, ArtistName, GenreId, NumTracks)
select ArtistId, ArtistName, GenreId, Count(*) as NumTracks
from art2 inner join tracks t
on art2.AlbumId = t.AlbumId
group by ArtistId, ArtistName, GenreId
```

```
select * from Artistgenres
```

Displaying records 1 - 10

ArtistId	ArtistName	GenreId	NumTracks
1	AC/DC	1	18
2	Accept	1	4
3	Aerosmith	1	15
4	Alanis Morissette	1	13

ArtistId	ArtistName	GenreId	NumTracks
5	Alice In Chains	1	12
6	AntÃ´nio Carlos Jobim	2	14
6	AntÃ´nio Carlos Jobim	7	17
7	Apocalyptica	3	8
8	Audioslave	1	14
8	Audioslave	4	12

d.

```
select * from(
  select ArtistId, artistName, Count(*) as Rec_Genres
  from ArtistGenres
  group by ArtistId)
  where Rec_Genres >1
```

Displaying records 1 - 10

ArtistId	artistName	Rec_Genres
6	AntÃ´nio Carlos Jobim	2
8	Audioslave	3
21	Various Artists	3
27	Gilberto Gil	3
81	Eric Clapton	2
82	Faith No More	2
84	Foo Fighters	2
88	Guns N’ Roses	2
90	Iron Maiden	4
92	Jamiroquai	3

e.

```
select sum(Quantity * UnitPrice) as Jamiroquai
from invoice_item
where TrackId in
(select TrackId
from Tracks
where AlbumId in
(select AlbumId
from Albums
where ArtistId = (select ArtistId
from Artists2
where Name = 'Jamiroquai')))
```

1 records

Jamiroquai
17.82

Q4.a

```
birthfile <- read.csv('birthfile.csv',stringsAsFactors = FALSE)
birthfile2018 <- read.csv('birthfile2018.csv', stringsAsFactors = FALSE)
RegionCodes <- read.csv('RegionCodes.csv',stringsAsFactors = FALSE)
vacc2014 <- read.csv('vacc2014.csv', stringsAsFactors = FALSE)
vacc2015 <- read.csv('vacc2015.csv', stringsAsFactors = FALSE)
vacc2016 <- read.csv('vacc2016.csv', stringsAsFactors = FALSE)
vacc2017 <- read.csv('vacc2017.csv', stringsAsFactors = FALSE)
vacc2018 <- read.csv('vacc2018.csv', stringsAsFactors = FALSE)
```

b.

```
birthfile$dob <- as.Date( birthfile$dob , format = "%Y-%m-%d")
birthfile2018$DOB <- as.Date(birthfile2018$DOB, format = "%d/%m/%Y")
vacc2014$date <- as.Date(vacc2014$date, format = "%d/%m/%Y")
vacc2015$date <- as.Date(vacc2015$date, format = "%d/%m/%Y")
vacc2016$date <- as.Date(vacc2016$date, format = "%d/%m/%Y")
vacc2017$date <- as.Date(vacc2017$date, format = "%d/%m/%Y")
vacc2018$date <- as.Date(vacc2018$date, format = "%Y-%b-%d")
```

c.

```
birthfile2018$sex <- ifelse( birthfile2018$Sex == "Female", "F",ifelse(birthfile2018$Sex == "Mal
e", "M",NA))

birthfile2018$Sex <- NULL
#testing
head(birthfile2018)
```

```
##           ID AgeOfMother        DOB        RegionName sex
## 1 VSF930437C          20 2018-04-04 Canterbury Region   M
## 2 AWB547492K          35 2018-05-21 Canterbury Region   M
## 3 NBZ730712D          30 2018-03-18 Canterbury Region   M
## 4 BEN284305V          25 2018-09-17 Canterbury Region   F
## 5 VNG756721J          20 2018-05-26 Canterbury Region   M
## 6 GBB883695B          35 2018-04-16 Canterbury Region   M
```

d.

```
birthfile2018 <- merge(birthfile2018, RegionCodes, by.x = "RegionName", by.y = "Region")
head(birthfile2018)
```

```
##          RegionName         ID AgeOfMother        DOB sex RegionID
## 1 Canterbury Region VSF930437C          20 2018-04-04   M       14
## 2 Canterbury Region AWB547492K          35 2018-05-21   M       14
## 3 Canterbury Region NBZ730712D          30 2018-03-18   M       14
## 4 Canterbury Region BEN284305V          25 2018-09-17   F       14
## 5 Canterbury Region VNG756721J          20 2018-05-26   M       14
## 6 Canterbury Region GBB883695B          35 2018-04-16   M       14
```

e.

```
birthfile2018$RegionName <- NULL
colnames(birthfile2018) <- c("id","MAge","dob", "sex","RegionID")
head(birthfile2018)
```

```
##           id MAge        dob sex RegionID
## 1 VSF930437C   20 2018-04-04   M       14
## 2 AWB547492K   35 2018-05-21   M       14
## 3 NBZ730712D   30 2018-03-18   M       14
## 4 BEN284305V   25 2018-09-17   F       14
## 5 VNG756721J   20 2018-05-26   M       14
## 6 GBB883695B   35 2018-04-16   M       14
```

f.

```
vacc2018 <- vacc2018[c("id", "stage", "date")]
v <- rbind(vacc2014, vacc2015, vacc2016, vacc2017, vacc2018)
nrow(v)
```

```
## [1] 158830
```

g

```
b <- rbind(birthfile, birthfile2018)
nrow(b)
```

```
## [1] 59882
```

h.

```
b$birthYear = format(b$dob, "%Y")

aggregate(cbind(b_c = id) ~ birthYear, b, function(x){NROW(x)})
```

```
##   birthYear   b_c
## 1      2014 11419
## 2      2015 12406
## 3      2016 12173
## 4      2017 12127
## 5      2018 11757
```

i.

```
date_3 <- reshape(v , idvar = "id", timevar = "stage", v.names = "date", direction = "wide")
```

j.

```
bir <- merge(b, date_3, by = "id", all.x = TRUE)
head(bir)
```

```
##           id MAge        dob sex RegionID birthYear     date.1     date.2
## 1 AAA170821D   30 2016-05-24   F       11      2016 2016-07-04 2016-08-16
## 2 AAA572357X   35 2017-05-17   M       14      2017 2017-06-30 2017-08-23
## 3 AAA731210C   30 2016-11-02   M       14      2016 2016-12-15 2017-01-31
## 4 AAA743866C   25 2017-10-14   M       14      2017 2017-11-27 2018-01-11
## 5 AAA916820V   25 2018-09-21   F       16      2018 2018-11-06 2018-12-22
## 6 AAC412779P   30 2014-03-11   F       15      2014       <NA>       <NA>
##       date.3
## 1 2016-11-23
## 2 2017-11-26
## 3 2017-05-23
## 4 2018-03-31
## 5       <NA>
## 6       <NA>
```

k.

```
head(bir,3)
```

```
##           id MAge        dob sex RegionID birthYear     date.1     date.2
## 1 AAA170821D   30 2016-05-24   F       11      2016 2016-07-04 2016-08-16
## 2 AAA572357X   35 2017-05-17   M       14      2017 2017-06-30 2017-08-23
## 3 AAA731210C   30 2016-11-02   M       14      2016 2016-12-15 2017-01-31
##       date.3
## 1 2016-11-23
## 2 2017-11-26
## 3 2017-05-23
```