Exam information and practice exam

STATS 220 Semester One 2023

Exam information   How to revise for the exam!

Information about questions   Practice exam   Past exams

This practice exam is designed to give you an idea of the format of the exam and the kinds of questions you can expect to be given. **The project details referred to in the practice exam are from Semester One 2022.**

Please use Ed Discussion to post questions related to this practice exam, or general revision related questions 😺

Section 1   Section 2   Section 3

This section will contain questions based on knowledge of JSON and API queries, databases and SQL queries, and related data manipulations.

Total 22 marks

For Project 3, you sourced data about books from the Google Books API in a JSON data format.

Q7

When creating a new data frame as part of Project 3, you were instructed to “Select and rename two more numeric or character variables of your choice - do not select columns that contain lists or vectors”.

With reference to JSON data structures, explain why you were instructed not to select columns that contained lists or vectors.

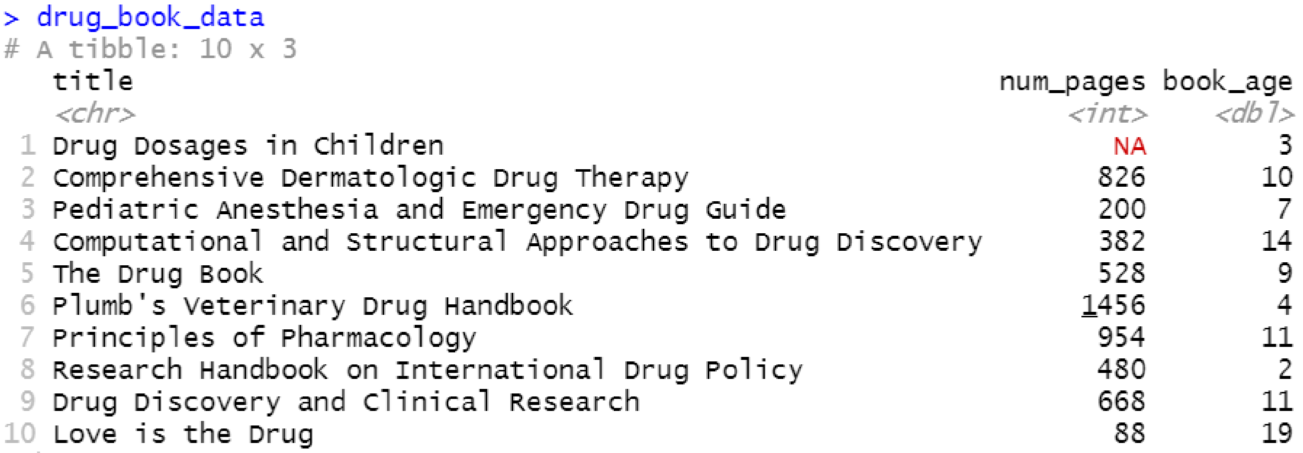
2 marks

JSON data structures can be nested, for example, books can have more than one category assigned to them in the Google Books API. If you tried to use one of these variables/columns, then your data would not be “tidy” or rectangular, which could have caused problems with data manipulations and visualisations.

Q8

Data was sourced from the Google Books API for books that had the word “drug” in their title (or subtitle).

R code and functions from {jsonlite} and {dplyr} were then used to create a new data frame drug_book_data .



```
> drug_book_data
# A tibble: 10 x 3
   title                                                       num_pages book_age
   <chr>                                                           <int>    <dbl>
 1 Drug Dosages in Children                                           NA        3
 2 Comprehensive Dermatologic Drug Therapy                           826       10
 3 Pediatric Anesthesia and Emergency Drug Guide                     200        7
 4 Computational and Structural Approaches to Drug Discovery         382       14
 5 The Drug Book                                                     528        9
 6 Plumb's Veterinary Drug Handbook                                 1456        4
 7 Principles of Pharmacology                                        954       11
 8 Research Handbook on International Drug Policy                    480        2
 9 Drug Discovery and Clinical Research                              668       11
10 Love is the Drug                                                   88       19
```

The code below provides the code used to create drug_book_data but some parts of the code have been replaced with numbers.

```
query <- "https://www.googleapis.com/books/v1/volumes?q=intitle:%22drug%22&startIndex=0&maxResults={1}"

response <- {2}(query, flatten = TRUE)

drug_book_data <- response$items %>%
  {3}(title = volumeInfo.title,
         published_date = volumeInfo.publishedDate,
         num_pages = volumeInfo.pageCount) %>%
  mutate(year_published = str_sub(published_date, 1, 4) %>% {4}(),
         {5} = 2022 - year_published) %>%
  {6}(title, num_pages, book_age)
```

Use the boxes below to enter the missing function, operation, argument name or value.

6 marks

{1} 10

{2} fromJSON

{3} rename

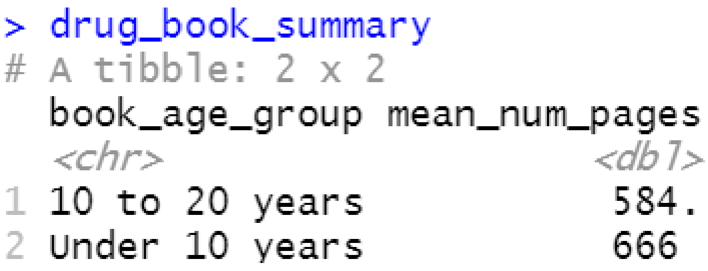
{4} as.numeric

{5} book_age

{6} select

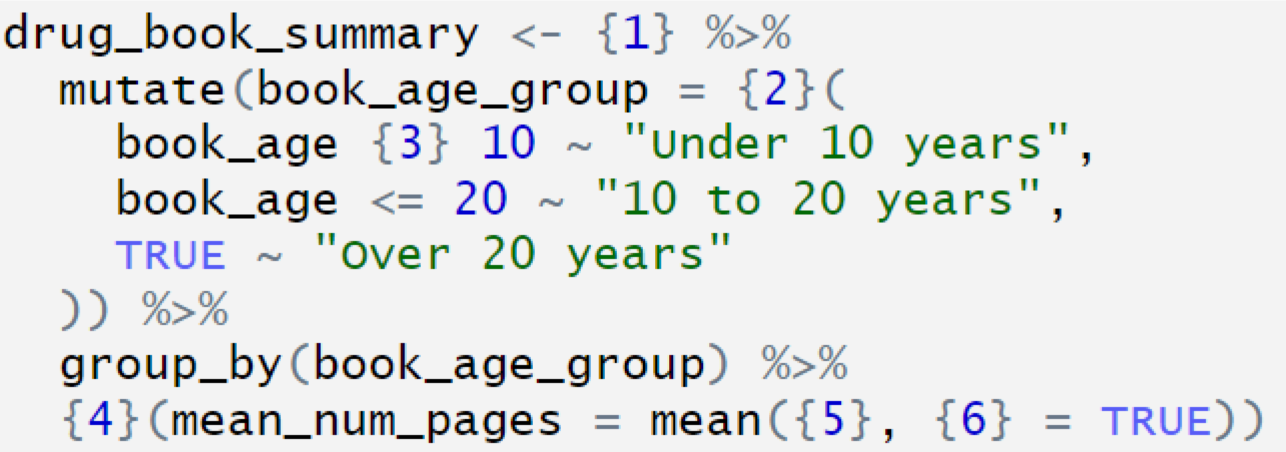
Q9

Further R code and functions from {dplyr} were then used to manipulate drug_book_data to create a new data frame drug_book_summary .



```
> drug_book_summary
# A tibble: 2 x 2
  book_age_group mean_num_pages
  <chr>                   <dbl>
1 10 to 20 years           584.
2 Under 10 years           666
```

The code below provides the code used to create drug_book_summary but some parts of the code have been replaced with numbers.



```
drug_book_summary <- {1} %>%
  mutate(book_age_group = {2}(
    book_age {3} 10 ~ "Under 10 years",
    book_age <= 20 ~ "10 to 20 years",
    TRUE ~ "Over 20 years"
  )) %>%
  group_by(book_age_group) %>%
  {4}(mean_num_pages = mean({5}, {6} = TRUE))
```

Use the boxes below to enter the missing function, operation, argument name or value.

6 marks

{1} drug_book_data

{2} case_when

{3} <

{4} summarise

{5} num_pages

{6} na.rm

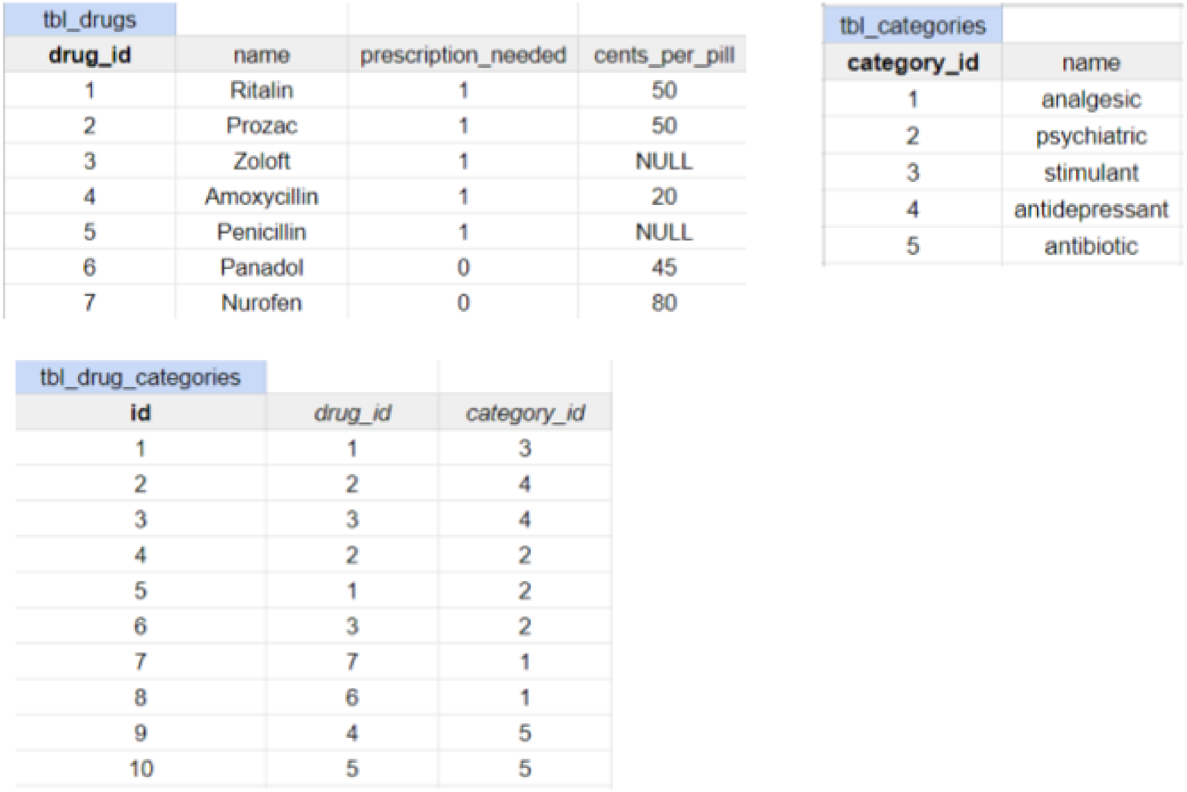
Q10

The data frame drug_book_summary does not contain any summaries for books over 20 years old. Is this because of the code used in Q9?

2 marks

The code is fine! Since the case_when() function first checks for books under 10 years, and then between 10 and 20 years, whatever is left over must be over 20 years (unless there are NA values). Therefore, it appears there are just no books that old in the drug_book_data data frame.

The following question refers to the database shown below.



tbl_drugs			
drug_id	name	prescription_needed	cents_per_pill
1	Ritalin	1	50
2	Prozac	1	50
3	Zoloft	1	NULL
4	Amoxycillin	1	20
5	Penicillin	1	NULL
6	Panadol	0	45
7	Nurofen	0	80

tbl_categories	
category_id	name
1	analgesic
2	psychiatric
3	stimulant
4	antidepressant
5	antibiotic

tbl_drug_categories		
id	*drug_id*	*category_id*
1	1	3
2	2	4
3	3	4
4	2	2
5	1	2
6	3	2
7	7	1
8	6	1
9	4	5
10	5	5

Q11

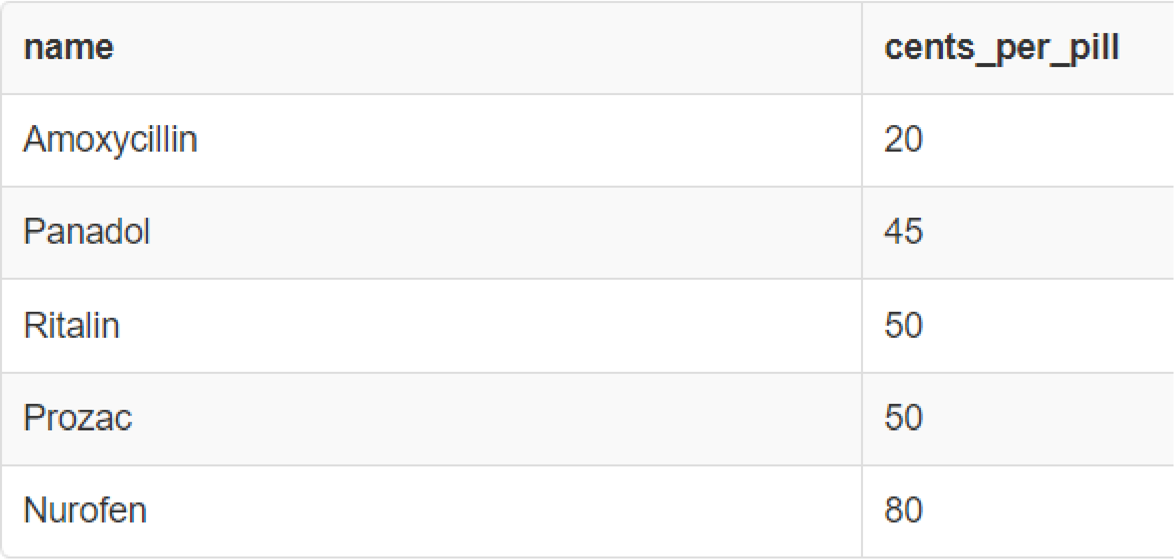
The database contains a drugs table (tbl_drugs), where each row is about a drug, and a categories table (tbl_categories), where each row is about a category. Explain why it also contains tbl_drug_categories.

2 marks

Each drug can have many categories (e.g. Ritalin is a stimulant, as well as a psychiatric), and each category can apply to many drugs (e.g. the analgesic category applies to both Panadol and Nurofen). The tbl_drug_categories is needed to be able to join tbl_drugs and tbl_categories while maintaining the third-normal form.

Q12

SQL code was used to produce the following output:



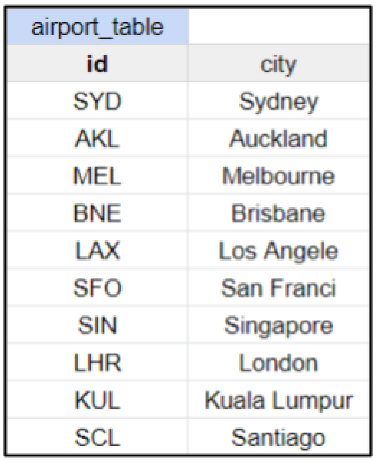
name	cents_per_pill
Amoxycillin	20
Panadol	45
Ritalin	50
Prozac	50
Nurofen	80

In addition to the SQL commands `SELECT` and `FROM`, two other commands were used for the query. Identify each of these commands and discuss how they would be used for the query.

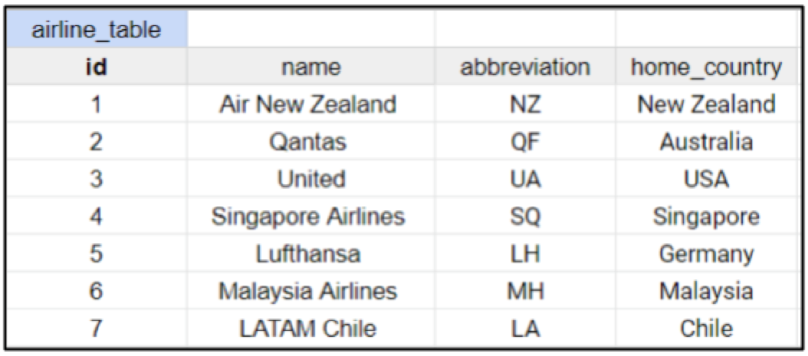
2 marks

The WHERE command is needed, as the output does not contain any NULL values for `cents_per_pill`.
The ORDER BY command is needed, as the values for `cents_per_pill` are ordered from lowest to highest.

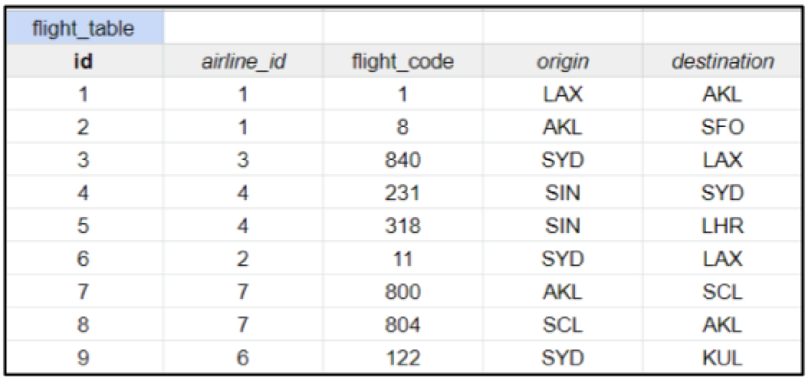
The following question refers to the database below:



airport_table	
id	city
SYD	Sydney
AKL	Auckland
MEL	Melbourne
BNE	Brisbane
LAX	Los Angele
SFO	San Franci
SIN	Singapore
LHR	London
KUL	Kuala Lumpur
SCL	Santiago



airline_table			
id	name	abbreviation	home_country
1	Air New Zealand	NZ	New Zealand
2	Qantas	QF	Australia
3	United	UA	USA
4	Singapore Airlines	SQ	Singapore
5	Lufthansa	LH	Germany
6	Malaysia Airlines	MH	Malaysia
7	LATAM Chile	LA	Chile



flight_table				
id	*airline_id*	flight_code	*origin*	*destination*
1	1	1	LAX	AKL
2	1	8	AKL	SFO
3	3	840	SYD	LAX
4	4	231	SIN	SYD
5	4	318	SIN	LHR
6	2	11	SYD	LAX
7	7	800	AKL	SCL
8	7	804	SCL	AKL
9	6	122	SYD	KUL

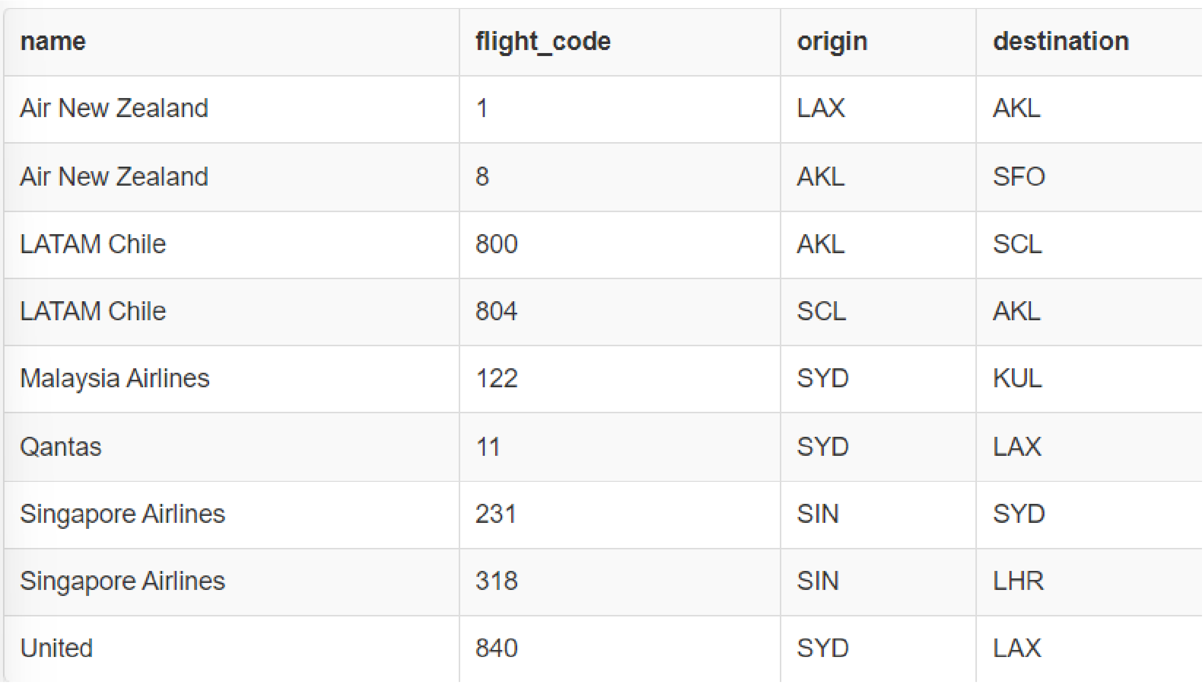
A SQL fiddle has been set up for this database, which you can use to help practice and develop your answers:

http://sqlfiddle.com/#!5/bc219 (http://sqlfiddle.com/#!5/bc219)

Q13

A SQL query needs to be written that selects the airline name, flight code, origin, and destination for flights.

Below is the table that should be the result of the query.



name	flight_code	origin	destination
Air New Zealand	1	LAX	AKL
Air New Zealand	8	AKL	SFO
LATAM Chile	800	AKL	SCL
LATAM Chile	804	SCL	AKL
Malaysia Airlines	122	SYD	KUL
Qantas	11	SYD	LAX
Singapore Airlines	231	SIN	SYD
Singapore Airlines	318	SIN	LHR
United	840	SYD	LAX

Discuss what kind of join should be used and specify what keys from what tables need to be joined.

As only the names of airlines that have flights are in the output, an inner join was used. The keys used for this join were the column id from the airline_table and the column airline_id from the flight_table.

2 marks