Solutions for the exercise session 2

1. SQL queries

Problem 1.1: Bikeshare trip durations in San Francisco.

1. Chatgpt uses a different dataset (instead of `san_francisco_bikeshare` it searches for a table in `san_francisco dataset`).
2. The code given by chatgpt does not actually need to join tables, we only need table that contains information on durations of trips (i.e. the table `bikeshare_trips`).

Below the correct and simplified code is presented.

```
SELECT start_station_id AS station_name,
       AVG(trips.duration_sec / 60) AS avg_rent_time_minutes
FROM `bigquery-public-data.san_francisco_bikeshare.bikeshare_trips` AS trips
GROUP BY station_name
ORDER BY avg_rent_time_minutes DESC
```

Problem 1.2: NHTSA Traffic Fatalities dataset.

1. The dataset `bigquery-public-data.nhtsa_traffic_fatalities.accident_2020` seems to be unavailable, we analyze the dataset

   `bigquery-public-data.nhtsa_traffic_fatalities.accident_2016`.
2. No need to filter the year, since we already access the data with specific year.
3. No need to set the limit, since there might be several days of week with the maximal number of motor accidents.

Below the correct and simplified code is presented.

```
SELECT FORMAT_TIMESTAMP('%A', TIMESTAMP(timestamp_of_crash)) AS day_of_week,
       COUNT(*) AS total_fatalities
FROM `bigquery-public-data.nhtsa_traffic_fatalities.accident_2016`
GROUP BY day_of_week
ORDER BY total_fatalities DESC
```

Problem 1.3: Austin longest duration bike rides.

```
SELECT start_station_name
FROM `bigquery-public-data.austin_bikeshare.bikeshare_trips` AS A
ORDER BY duration_minutes DESC
LIMIT 5
```

Problem 1.4: Austin stolen bikes.

```
SELECT COUNT(trip_id) AS count
FROM `bigquery-public-data.austin_bikeshare.bikeshare_trips` AS A
WHERE A.end_station_name = "Stolen"
```

Problem 1.5: Swiss customers

First we create our database.

```
library(DBI); library(RSQLite)
con <- dbConnect(SQLite(), ":memory:")

DBI::dbExecute(con, "
CREATE TABLE customers (
  customer_id INTEGER PRIMARY KEY,
  name TEXT NOT NULL,
  country TEXT NOT NULL
);
")
```

```
[1] 0
```

```
DBI::dbExecute(con, "
CREATE TABLE orders (
  order_id INTEGER PRIMARY KEY,
  customer_id INTEGER NOT NULL,
  order_date TEXT NOT NULL, -- ISO8601
  amount REAL NOT NULL,
  FOREIGN KEY(customer_id) REFERENCES customers(customer_id)
);
")
```

```
[1] 0
```

```
DBI::dbWriteTable(con, "customers",
  data.frame(customer_id=1:5,
             name=c("Ada","Bruno","Chloé","Dinesh","Elena"),
             country=c("CH","FR","CH","IN","US")),
  append=TRUE)

DBI::dbWriteTable(con, "orders",
  data.frame(order_id=1:10,
             customer_id=c(1,1,3,3,3,3,4,4,5,5),
             order_date=as.character(as.Date("2025-09-20")
             + c(0,1,0,2,5,6,1,7,3,9)),
             amount=c(120,75,300,42,88,150,60,500,20,220)),
  append=TRUE)

DBI::dbListTables(con)
```

```
[1] "customers" "orders"
```

Now we select names of swiss customers.

```
DBI::dbGetQuery(con, "
    SELECT name
    FROM customers
    WHERE country = 'CH'
    ORDER by name
    LIMIT 3;
")
```

```
   name
1   Ada
2 Chloé
```

Though we only display top 3 names, there was no difference if we would not set the limit (check it!).

Problem 1.6: Average number of orders per customer.

```
DBI::dbGetQuery(con, "
  SELECT c.name, AVG(o.amount) AS avg
  FROM customers AS c
  INNER JOIN orders AS o
    ON c.customer_id = o.customer_id
  GROUP BY c.name
  ORDER BY avg DESC
  LIMIT 3;
")
```

```
    name avg
1 Dinesh 280
2  Chloé 145
3  Elena 120
```

Problem 1.7: Customers with no orders.

```
DBI::dbGetQuery(con, "
  SELECT c.customer_id, c.name
  FROM customers c
  LEFT JOIN orders o
    ON o.customer_id = c.customer_id
  WHERE o.order_id IS NULL;
")
```

```
  customer_id  name
1           2 Bruno
```

2. Accessing data via API keys.

Problem 2.1: Weather in Bern.

```r
# packages
library(httr2)
library(dplyr)

# store your key in an env var before class:
Sys.setenv(OPENWEATHERMAP_API_KEY = your_key)
okey <- Sys.getenv("OPENWEATHERMAP_API_KEY")

resp <- request("https://api.openweathermap.org/data/2.5/weather") |>
  req_url_query(q = "Bern", units = "metric", appid = okey) |>
  req_perform()

wx <- resp_body_json(resp, simplifyVector = TRUE)

# extract a compact summary row
wx_row <- tibble::tibble(
  city        = wx$name,
  country     = wx$sys$country,
  temperature = wx$main$temp,
  feels_like  = wx$main$feels_like,
  humidity    = wx$main$humidity,
  wind_ms     = wx$wind$speed,
  condition   = wx$weather$description,
  timestamp   = as.POSIXct(wx$dt, origin = "1970-01-01", tz = "UTC")
)
wx_row
```

Problem 2.2: Information on OMDB movies.

```r
library(httr2)
Sys.setenv(OMDB_API_KEY = your_key)
mkey <- Sys.getenv("OMDB_API_KEY")
# Search first
s <- request("https://www.omdbapi.com/") |>
  req_url_query(apikey = mkey,
  s = "The Lord of the Rings: The Fellowship of the Ring", y = 2001) |>
  req_perform() |>
  resp_body_json(simplifyVector = TRUE)
head(s$Search)

# Then fetch details by IMDb id
movie <- request("https://www.omdbapi.com/") |>
  req_url_query(apikey = mkey, i = "tt0120737") |>
  req_perform() |>
  resp_body_json(simplifyVector = TRUE)

unlist(movie[c("Title","Year","Genre","Director","imdbRating")])
```

```python
import requests
import pandas as pd

url = "https://en.wikipedia.org/wiki/Python_(programming_language)"
headers = {
    # Be polite & descriptive per Wikimedia’s policy:
    "User-Agent": "Timofei-Education (timofei98shashkov@gmail.com) requests"
}
resp = requests.get(url, headers=headers, timeout=60)
resp.raise_for_status()  # will raise if still not 200

tables = pd.read_html(resp.text)  # requires lxml or html5lib installed
```

```
<string>:2: FutureWarning: Passing literal html to 'read_html' is deprecated and will be rem
```

```python
print(len(tables))
```

```
17
```

```python
print(tables[5].shape)
```

```
(16, 4)
```

Problem 2.3: Alpha Vantage

```r
library(httr2)
```

```
Warning: package 'httr2' was built under R version 4.3.3
```

```r
library(purrr)
```

```
Warning: package 'purrr' was built under R version 4.3.2
```

```r
library(dplyr)
```

```
Warning: package 'dplyr' was built under R version 4.3.2


Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
akey <- Sys.getenv("ALPHAVANTAGE_API_KEY", unset = "demo")
resp <- request("https://www.alphavantage.co/query") |>
  req_url_query(`function` = "TIME_SERIES_DAILY", symbol = "IBM", apikey = akey) |>
  req_perform()
js <- resp_body_json(resp, simplifyVector = TRUE)

daily <- js[["Time Series (Daily)"]]
prices <- imap_dfr(daily, ~ tibble::tibble(
  date   = as.Date(.y),
  open   = as.numeric(.x[["1. open"]]),
  high   = as.numeric(.x[["2. high"]]),
  low    = as.numeric(.x[["3. low"]]),
  close  = as.numeric(.x[["4. close"]]),
  volume = as.numeric(.x[["5. volume"]])
)) |>
  arrange(date)
prices
```

```
# A tibble: 100 x 6
   date        open  high   low close  volume
   <date>     <dbl> <dbl> <dbl> <dbl>   <dbl>
 1 2025-06-05  265.  268.  265.  267. 2659478
 2 2025-06-06  268.  270.  268.  269. 2495543
 3 2025-06-09  268.  273.  267.  272. 4331464
 4 2025-06-10  273.  277.  273.  276. 5163507
 5 2025-06-11  277.  282.  275.  282. 4656034
 6 2025-06-12  282.  283.  280.  281. 3418007
 7 2025-06-13  278.  280.  276.  277. 3243824
 8 2025-06-16  279.  284.  279.  282. 3685321
 9 2025-06-17  281.  285.  281.  283. 3069556
10 2025-06-18  285   287.  283.  283. 3534110
# i 90 more rows
```

3. Webscraping exercises

We will explore the webpage Python (programming language).

Problem 3.1. Hunting for table CSS selector.

Using selector gadget extension to hunt for CSS selectors for sections. The selector `.mw-heading2` describes exactly what we wanted.

Problem 3.2. Scraping a table.

Next we scrape the table `Summary of Python 3's built-in types` in **R** using `html_table`.

```
library(rvest)

url <- "https://en.wikipedia.org/wiki/Python_(programming_language)"
tables <- url %>%
  read_html() %>%
  html_table(fill = TRUE)
tables
```

```
[[1]]
# A tibble: 3 x 6
  X1    X2                                                         X3    X4    X5    X6
  <lgl> <chr>                                                      <lgl> <chr> <lgl> <chr>
```

```
1 NA    "This article has multiple issues. Please help ~ NA    This~ NA    This~
2 NA    "This article needs to be updated. The reason g~ NA    <NA>  NA    <NA>
3 NA    "This article may contain unverified or indiscr~ NA    <NA>  NA    <NA>

[[2]]
# A tibble: 1 x 2
  X1    X2
  <lgl> <chr>
1 NA    This article needs to be updated. The reason given is: Information rela~

[[3]]
# A tibble: 1 x 2
  X1    X2
  <lgl> <chr>
1 NA    This article may contain unverified or indiscriminate information in em~

[[4]]
# A tibble: 22 x 2
   Python              Python
   <chr>               <chr>
 1 ""                  ""
 2 "Paradigm"          "Multi-paradigm: object-oriented,[1]procedural (imperati~
 3 "Designed by"       "Guido van Rossum"
 4 "Developer"         "Python Software Foundation"
 5 "First appeared"    "20 February 1991; 34 years ago (1991-02-20)[2]"
 6 ""                  ""
 7 "Stable release"    "3.14.0[3] \n   / 7 October 2025; 21 days ago (7 October~
 8 ""                  ""
 9 "Typing discipline" "Duck, dynamic, strong;[4]optional type annotations[a]"
10 "OS"                "Cross-platform including e.g. for mobile/Android[b]"
# i 12 more rows

[[5]]
# A tibble: 10 x 1
   X1
   <chr>
 1 "This article is part of a series on"
 2 "Python"
 3 ""
 4 "Python frameworks\nBlueBream CherryPy CubicWeb Django FastAPI Flask Google ~
 5 "Python libraries\nappJar Anaconda Apache MXNet Apache Singa Astropy Beautif~
 6 "Python IDEs\nAtom / Pulsar Codelobster EasyEclipse Eclipse Emacs Eric Geany~
 7 "Python implementations\nActivePython CLPython CPython Cython Intel Dist. fo~
 8 "See alsoHistory of Python List of Python books List of Python conferences L~
 9 "Computer programming portal\nPython Programming (Wikibook)"
10 ".mw-parser-output .navbar{display:inline;font-size:88%;font-weight:normal}.~

[[6]]
# A tibble: 16 x 4
   Type                      Mutability Description             `Syntax examples`
   <chr>                     <chr>      <chr>                   <chr>
 1 bool                      immutable  Boolean value           "TrueFalse"
 2 bytearray                 mutable    Sequence of bytes       "bytearray(b'Som~
 3 bytes                     immutable  Sequence of bytes       "b'Some ASCII'b\~
 4 complex                   immutable  Complex number with re~ "3+2.7j3 + 2.7j5~
 5 dict                      mutable    Associative array (or ~ "{'key1': 1.0, 3~
 6 types.EllipsisType        immutable  An ellipsis placeholde~ "...Ellipsis"
 7 float                     immutable  Double-precision float~ "1.33333"
 8 frozenset                 immutable  Unordered set, contain~ "frozenset({4.0,~
 9 int                       immutable  Integer of unlimited m~ "42"
10 list                      mutable    List, can contain mixe~ "[4.0, 'string',~
11 types.NoneType            immutable  An object representing~ "None"
12 types.NotImplementedType immutable  A placeholder that can~ "NotImplemented"
13 range                     immutable  An immutable sequence ~ "range(-1, 10)ra~
14 set                       mutable    Unordered set, contain~ "{4.0, 'string',~
15 str                       immutable  A character string: se~ "'Wikipedia'\"Wi~
16 tuple                     immutable  Tuple, can contain mix~ "(4.0, 'string',~

[[7]]
# A tibble: 5 x 3
  vtePython                                                      vtePython vtePython
  <chr>                                                          <chr>     <chr>
1 "Implementations"                                              "Circuit~ ""
2 "IDEs"                                                         "eric\nI~ ""
3 "Topics"                                                       "WSGI\nA~ ""
4 "Designer"                                                     "Guido v~ ""
5 "Software (list)\nPython Software Foundation\nPython Conf~ "Softwar~ "Softwar~

[[8]]
# A tibble: 3 x 2
  `vteProgramming languages`                             vteProgramming langu~1
  <chr>                                                  <chr>
1 "Comparison\nTimeline\nHistory"                        "Comparison\nTimeline~
2 "Ada\nALGOL\nSimula\nAPL\nAssembly\nBASIC\nVisual Basi~ "Ada\nALGOL\nSimula\n~
3 "Lists: Alphabetical\nCategorical\nGenerational\nNon-E~ "Lists: Alphabetical\~
# i abbreviated name: 1: `vteProgramming languages`
```

```
[[9]]
# A tibble: 2 x 2
  `vtePython web frameworks`                              vtePython web framew~1
  <chr>                                                   <chr>
1 "CherryPy\nCubicWeb\nDjango\nFastAPI\nFlask\nGrok\nNev~ "CherryPy\nCubicWeb\n~
2 "Comparison"                                            "Comparison"
# i abbreviated name: 1: `vtePython web frameworks`

[[10]]
# A tibble: 4 x 2
  `vteDifferentiable computing`              `vteDifferentiable computing`
  <chr>                                      <chr>
1 "General"                                  "Differentiable programming\nInfo~
2 "Hardware"                                 "IPU\nTPU\nVPU\nMemristor\nSpiNNa~
3 "Software libraries"                       "TensorFlow\nPyTorch\nKeras\nscik~
4 "Portals\nComputer programming\nTechnology" "Portals\nComputer programming\nT~

[[11]]
# A tibble: 9 x 4
  `vteFree and open-source software` vteFree and open-source softw~1 ``    ``
  <chr>                              <chr>                           <chr> <chr>
1 "General"                          "Alternative terms for free so~ <NA>   <NA>
2 "Softwarepackages"                 "Audio\nBioinformatics\nCodecs~ <NA>   <NA>
3 "Community"                        "Free software movement\nHisto~ <NA>   <NA>
4 "Organisations"                    "Free Software Movement of Ind~ <NA>   <NA>
5 "Licenses"                         "AFL\nApache\nAPSL\nArtistic\n~ Type~ "Com~
6 "Types and standards"              "Comparison of licenses\nContr~ <NA>   <NA>
7 "Challenges"                       "Digital rights management\nLi~ <NA>   <NA>
8 "Related topics"                   "Forking\nGNU Manifesto\nMicro~ <NA>   <NA>
9 "Portal\n Category"                "Portal\n Category"             <NA>   <NA>
# i abbreviated name: 1: `vteFree and open-source software`

[[12]]
# A tibble: 1 x 2
  X1                  X2
  <chr>               <chr>
1 Types and standards "Comparison of licenses\nContributor License Agreement\nC~

[[13]]
# A tibble: 8 x 8
  `vteStatistical software` vteStatistical softw~1 ``    ``    ``    ``    ``
  <chr>                     <chr>                  <chr> <chr> <chr> <chr> <chr>
```

```
1 Public domain             "Dataplot\nEpi Info\n~ <NA>   <NA> <NA>   <NA> <NA>
2 Open-source               "ADMB\nDAP\ngretl\nja~ <NA>   <NA> <NA>   <NA> <NA>
3 Freeware                  "BV4.1\nCumFreq\nSegR~ <NA>   <NA> <NA>   <NA> <NA>
4 Commercial                "Cross-platform\nData~ Cros~ "Dat~ Wind~ "BMD~ Exce~
5 Cross-platform            "Data Desk\nGAUSS\nGr~ <NA>   <NA> <NA>   <NA> <NA>
6 Windows only              "BMDP\nEViews\nGenSta~ <NA>   <NA> <NA>   <NA> <NA>
7 Excel add-ons             "Analyse-it\nUnistat ~ <NA>   <NA> <NA>   <NA> <NA>
8 Comparison • Category     "Comparison • Categor~ <NA>   <NA> <NA>   <NA> <NA>
# i abbreviated name: 1: `vteStatistical software`
# i 1 more variable: `` <chr>

[[14]]
# A tibble: 3 x 2
  X1             X2
  <chr>          <chr>
1 Cross-platform "Data Desk\nGAUSS\nGraphPad InStat\nGraphPad Prism\nIBM SPSS S~
2 Windows only   "BMDP\nEViews\nGenStat\nLIMDEP\nLISREL\nMedCalc\nMicrofit\nMin~
3 Excel add-ons  "Analyse-it\nUnistat for Excel\nXLfit\nRExcel"

[[15]]
# A tibble: 4 x 4
  `vteNumerical-analysis software` `vteNumerical-analysis software`  ``    ``
  <chr>                            <chr>                             <chr> <chr>
1 Free                             "Advanced Simulation Library\nAD~ Disc~ Fort~
2 Discontinued                     "Fortress"                        <NA>  <NA>
3 Proprietary                      "DADiSP\nFEATool Multiphysics\nG~ <NA>  <NA>
4 Comparison                       "Comparison"                      <NA>  <NA>

[[16]]
# A tibble: 1 x 2
  X1           X2
  <chr>        <chr>
1 Discontinued Fortress

[[17]]
# A tibble: 3 x 2
  `Authority control databases` `Authority control databases`
  <chr>                         <chr>
1 International                 GNDFAST
2 National                      United StatesFranceBnF dataCzech RepublicIsrael
3 Other                         IdRefYale LUX
```

```r
length(tables)
```

```
[1] 17
```

```r
dim(tables[[6]])
```

```
[1] 16  4
```

Alternatively a table can be scraped using, for example, selector gadget.

Problem 3.3. Count domain frequences.

```r
library(rvest)
library(urltools)
```

```
Warning: package 'urltools' was built under R version 4.3.3


Attaching package: 'urltools'

The following object is masked from 'package:httr2':

    url_parse
```

```r
library(dplyr)

url <- "https://en.wikipedia.org/wiki/Python_(programming_language)"
links <- read_html(url) %>%
  html_nodes("a") %>%
  html_attr("href") %>%
  na.omit()

links <- links[grepl("^http", links) | grepl("^/wiki/", links)]
links <- unique(links)[1:20]
links <- ifelse(grepl("^/wiki/", links), paste0("https://en.wikipedia.org", links), links)

domains <- domain(links)
table(domains)
```

```
domains
    af.wikipedia.org    als.wikipedia.org     an.wikipedia.org 
                   1                    1                    1 
    ar.wikipedia.org     as.wikipedia.org donate.wikimedia.org 
                   1                    1                    1 
    en.wikipedia.org 
                  14 
```

Problem 3.4. Ethical issues.

You can add options to executable code like this

```r
library(polite)
```

```
Warning: package 'polite' was built under R version 4.3.3
```

```r
robotstxt::paths_allowed("https://en.wikipedia.org/wiki/Python_(programming_language)")
```

```

 en.wikipedia.org
```

```
[1] TRUE
```