ISOM 661: Technology Bootcamp

Self-Assessment Test

Fall 2022

Survey

1. What is your experience with and how comfortable do you feel coding in SQL? Please elaborate on your answer.

For example:

* 1. How did you learn SQL?

*I learn SQL from online course on Coursera.*

* 1. Which courses (in-person or online) that used SQL have you taken?

*Intro to SQL by UC Davis.*

* 1. What kind of projects have you coded using SQL?

*A lot of ad-hoc analysis during internships, mostly queries.*

* 1. What was the hardest part of learning SQL?

*I’m still confused with the pipeline and ETL part.*

* 1. What do you like and don’t like about SQL?

*It’s an easy-learning and broad using script language for big data. Sometimes it’s hard to manage and analysis data from different databases at the same time. Also, for huge databases, efficiency usually need to be boosted constantly for convenient usage.*

* 1. What things about SQL do you still want to learn?

*The underlining mechanism of databases, including how to load and manage databases.*

* 1. On a scale from 1-5, how comfortable would you be doing a job interview that required SQL?

*3 for Data Analyst questions. 1.5 for Data Engineer questions.*

1. What is your experience with and how comfortable do you feel coding in R? Please elaborate on your answer.

For example:

* 1. How did you learn R?

*From stats courses at UC Berkeley.*

* 1. What kind of projects have you coded using R?

*A couple of internship projects, but nothing as complicate as machine learning problems.*

* 1. What do you like and don’t like about R?

*For stats works, it’s convenient. However, with its’ statistical essence, the syntax is relatively confusing compared to other ‘real’ programming languages.*

* 1. What things about R do you still want to learn?

*I want to learn how R is used practically in businesses. I seldom use R during data analysis works personally.*

* 1. On a scale from 1-5, how comfortable would you be doing a job interview that required R?

*Unknown (I seldom encounter an R question*

1. What is your experience with and how comfortable do you feel coding in Python? Please elaborate on your answer.

For example:

* 1. How did you learn Python?

*I learn and use from classes during my undergrad.*

* 1. Which courses (in-person or online) that used Python have you taken?

*UC Berkely cs61a, and a Data Analysis with Python class at Nankai.*

* 1. What kind of projects have you coded using Python?

*Except for school projects, I have a machine learning project as my graduation thesis. I also have several data analysis projects during working for Tesla.*

* 1. What was the hardest part of learning Python?

*For me currently, it’s the algorithm. I’m not comfortable with LeetCode algorithm questions during Data Scientist interviews* ***at all****.*

* 1. What do you like and don’t like about Python?

*I like it’s natural convenient as a programming language and as a data analysis script language. Meanwhile, this means there are short backs on its computing and building more fundamental programs.*

* 1. What things about Python do you still want to learn?

*Please tell me how to cope with the algorithm questions during data scientist interviews.*

* 1. On a scale from 1-5, how comfortable would you be doing a job interview that required Python?

*3 for data analyst position, 1.5 for data scientist position.*

1. Do you have any experience with programming languages other than SQL, R, or Python? Please elaborate.

I have learnt a bit C++ during my freshman year.

1. Do you have any experience using Linux systems? Please elaborate.

We use Unix as a virtual environmental to schedule tasks at Tesla. My computer is also MacBook, and I learnt some bash commands.

Section 1: SQL

1. *Select one.* Which of the following best describes DDL, DML, DQL, DCL, and TCL?
   1. All represent different dialects of SQL.
   2. All represent different types of SQL notation.
   3. All represent different SQL programming languages.
   4. None of the above.
2. *Select one.* In order of increasing complexity, which of the following specifies the correct ordering?
   1. Table < Schema < Data Warehouse < Database
   2. Schema < Table < Database < Data Warehouse
   3. Schema < Table < Data Warehouse < Database
   4. Table < Database < Schema < Data Warehouse
   5. None of the above.

For questions 3-5, assume the following tables exist in the database and that there are no data errors.

**CREATE TABLE** employee (first\_name **VARCHAR**(15) **NOT NULL**,  
 last\_name **VARCHAR**(15) **NOT NULL**,  
 employee\_id **INT NOT NULL**,  
 date\_of\_birth **DATE**,  
 address **VARCHAR**(30),  
 salary **DECIMAL**(10, 2),  
 department\_number **INT NOT NULL**,  
 **PRIMARY KEY** (employee\_id));  
  
**CREATE TABLE** department (department\_name **VARCHAR**(30) **NOT NULL**,  
 department\_number **INT NOT NULL**,  
 manager\_employee\_id **INT NOT NULL**,  
 manager\_start\_date **DATE**,  
 **PRIMARY KEY** (department\_number),  
 **UNIQUE** (department\_name),  
 **FOREIGN KEY** (manager\_employee\_id) **REFERENCES** employee (employee\_id));  
  
**CREATE TABLE** project (project\_name **VARCHAR**(15) **NOT NULL**,  
 project\_number **INT NOT NULL**,  
 department\_number **INT NOT NULL**,  
 **PRIMARY KEY** (project\_number),  
 **UNIQUE** (project\_name),  
 **FOREIGN KEY** (department\_number) **REFERENCES** department (department\_number));  
  
**CREATE TABLE** dependent (employee\_id **INT NOT NULL**,  
 dependent\_name **VARCHAR**(15) **NOT NULL**,  
 date\_of\_birth **DATE**,  
 relationship **VARCHAR**(8),  
 **PRIMARY KEY** (employee\_id, dependent\_name),  
 **FOREIGN KEY** (employee\_id) **REFERENCES** employee (employee\_id));

1. *Select all that apply.* Which of the following can be used to describe the query below and/or its result?

SELECT  
 department\_number,  
 date\_of\_birth dob,  
 count(employee\_id) count  
FROM  
 employee  
ORDER BY  
 3 DESC;

* 1. The result will have three columns: department\_number, date\_of\_birth, and count.
  2. The result will have its three columns ordered in descending order.
  3. The result will have the same number of rows as the employee table.
  4. This query results in a syntax error.
  5. None of the above.

1. *Select all that apply.* Which of the following can be used to describe the query below and/or its result?

SELECT a.employee\_id,  
 a.salary \* 2 double\_salary,  
 b.department\_name,  
 a.address residence  
FROM employee AS a  
 LEFT JOIN department AS b  
 ON a.employee\_id = b.manager\_employee\_id  
WHERE a.employee\_id IN (SELECT employee\_id FROM dependent);

* 1. There may be NULL values in the double\_salary column.
  2. There may be NULL values in the department\_name column.
  3. The result will contain only employees with dependents.
  4. This query results in a syntax error.
  5. None of the above.

1. *Select all that apply.* Which of the following can be used to describe the query below and/or its result?

SELECT b.department\_number,  
 SUM(a.salary) salary  
FROM employee AS a  
 LEFT JOIN project AS b  
 ON a.department\_number = b.department\_number  
WHERE a.salary < 1000000  
GROUP BY salary  
HAVING salary > 1000000;

* 1. The query is an example of a many-to-many join.
  2. This query returns an empty result (i.e. zero rows).
  3. This query results in a syntax error.
  4. None of the above.

Section 2: R Programming

1. *Select one.* Given the following R script, which of the statements below is true?

v <- c(1,2,3,4)  
x <- list(5:8)  
v\*x[1]

* 1. There is an error on line 1
  2. There is an error on line 2
  3. There is an error on line 3
  4. None of the above

1. *Select one*. Given the following R script, which of the statements below is true?

my\_function <- function(x)  
{  
 x[is.na(x)] <- sum(x, na.rm = TRUE);  
 x  
}  
y <- my\_function(c(1, 2, 3))  
z <- my\_function(c(1, 2, NaN))

* 1. There is a bug on line 1
  2. There is a bug on line 3
  3. y and z contain the same values
  4. y and z contain the same values, except for the last one

1. *Select one.* Which of the following is the correct way to define a for-loop?
   1. for i in (1:3) {

print(i)

}

* 1. for(i in 1:3) {

print(i)

}

* 1. for i in (1:3):

print(i)

* 1. None of the above

1. *Select one.* Using the R Documentation for maxCol, which of the following statements best describes the R script below?

# R DOCUMENTATION for maxCol

maxCol: Find Maximum Position in Matrix

Description: Find the maximum position for each row of a matrix, breaking ties at random.

Usage: max.col(m, ties.method = c("random", "first", "last"))

Arguments:  
 - m: numerical matrix  
 - ties.method: a character string specifying how ties are handled, "random" by default;

can be abbreviated; see ‘Details’.  
  
Value: index of a maximal value for each row, an integer vector of length nrow(m).

data <- data.frame(col1=c(2, 8, 1),  
 col2=c(7, 3, 5),  
 col3=c(9, 6, 4))  
colnames(data)[max.col(data, ties.method="first")]

* 1. The last line outputs: "col3" "col1" "col2"
  2. The last line outputs: 9 8 5
  3. The last line outputs: 8 7 9
  4. None of the above.

1. *Select one.* Which of the following will load the ggplot2 library to the R workspace?
   1. install.packages("ggplot2")
   2. import ggplot2
   3. library(ggplot2)
   4. None of the above.

Section 3: Python Programming

1. *Select all that apply*. Which of the following statements will result in a syntax error? Assume that data is a Pandas DataFrame and that none of the statements below will result in an IndexError or KeyError.
   1. data.iloc[[0,2]]
   2. data.iloc[[0,2],[1,3]]
   3. data.loc[(data.age >= 20), ['zip\_code', 'city']]
   4. data.iloc[(data.height < 6)]
2. *True or False.* In the Pandas library, pd.groupby() will return an instance of pd.DataFrame() with its rows grouped.
   1. True
   2. False
3. *True or False*. In the Python script below, the console prints: True.

stores = [(1601, 3), (1634, 2),  
 (1642, 1), (1547, 2),  
 (1963, 3), (1613, 3),  
 (1518, 1), (1644, 1),  
 (1148, 2), (1649, 2)]  
  
dates = ['2019-10-01', '2019-10-01',  
 '2019-10-01', '2019-10-01', '2019-10-01']  
  
x = []  
for store in stores:  
 for date in dates:  
 x.append({"store\_num": store[0], "date": date})  
  
y = [{"store\_num": store\_num, "date": date}  
 for store\_num, region\_id in stores  
 for date in dates]

print(x == y)

1. True
2. False

For questions 14-15, assume the following tables have been downloaded from the database into a CSV file and then read into memory using the Pandas function pd.read\_csv(). The tables employee, department, project, and dependent have been stored in the variables df\_employee, df\_department, df\_project, and df\_dependent, respectively.

**CREATE TABLE** employee (first\_name **VARCHAR**(15) **NOT NULL**,  
 last\_name **VARCHAR**(15) **NOT NULL**,  
 employee\_id **INT NOT NULL**,  
 date\_of\_birth **DATE**,  
 address **VARCHAR**(30),  
 salary **DECIMAL**(10, 2),  
 department\_number **INT NOT NULL**,  
 **PRIMARY KEY** (employee\_id));  
  
**CREATE TABLE** department (department\_name **VARCHAR**(30) **NOT NULL**,  
 department\_number **INT NOT NULL**,  
 manager\_employee\_id **INT NOT NULL**,  
 manager\_start\_date **DATE**,  
 **PRIMARY KEY** (department\_number),  
 **UNIQUE** (department\_name),  
 **FOREIGN KEY** (manager\_employee\_id) **REFERENCES** employee (employee\_id));  
  
**CREATE TABLE** project (project\_name **VARCHAR**(15) **NOT NULL**,  
 project\_number **INT NOT NULL**,  
 department\_number **INT NOT NULL**,  
 **PRIMARY KEY** (project\_number),  
 **UNIQUE** (project\_name),  
 **FOREIGN KEY** (department\_number) **REFERENCES** department (department\_number));  
  
**CREATE TABLE** dependent (employee\_id **INT NOT NULL**,  
 dependent\_name **VARCHAR**(15) **NOT NULL**,  
 date\_of\_birth **DATE**,  
 relationship **VARCHAR**(8),  
 **PRIMARY KEY** (employee\_id, dependent\_name),  
 **FOREIGN KEY** (employee\_id) **REFERENCES** employee (employee\_id));

1. *True or False.* In the Python script below, assume that output\_filepath points to a valid location on disk. The data stored in output\_filepath will have two columns: foo and bar.

import pandas as pd  
import numpy as np  
  
(df\_employee[['employee\_id', 'salary']]  
 .groupby(['employee\_id'])  
 .agg([('foo', 'mean'),  
 ('bar', np.std)])).to\_csv(output\_filepath, index=False)

* 1. True
  2. False

1. *Select one*. Which of the following best explains the importance of the PYTHONPATH environment variable?
   1. The PYTHONPATH environment variable points to the location where the Python programming language is installed.
   2. The PYTHONPATH environment variable points to the location where the executing Python program resides.
   3. The PYTHONPATH environment variable tells the Python interpreter where to locate the Python binary that is used to execute Python scripts.
   4. The PYTHONPATH environment variable tells the Python interpreter where to locate the module files imported into a program.