HOMEWORK WEEK 2
(handout for students)

TASK 1 (SQL)

- MySQL Index:

NOTE: this is a very common question in tech interviews.

- The most important index types we need to know about are:
  - **Single column and multi-column index**

Single column index: A single-column index is an index based on the values in one column of a table; A single-column index should be created when retrieval will be executed using one column only as the key
Example: Location column is a single column index.

Original Table:

Location:	year	make	model
1	2016	TOYOTA	PRIUS
2	2016	HONDA	CIVIC
3	2017	CHEVROLET	SILVERADO
4	2017	TOYOTA	MDX
5	2017	ACURA	TL

```
Syntax : CREATE INDEX [index name]
ON [Table name]([column1])
```

Multi-column : A multicolumn index is an index based on the values in multiple columns of a table.

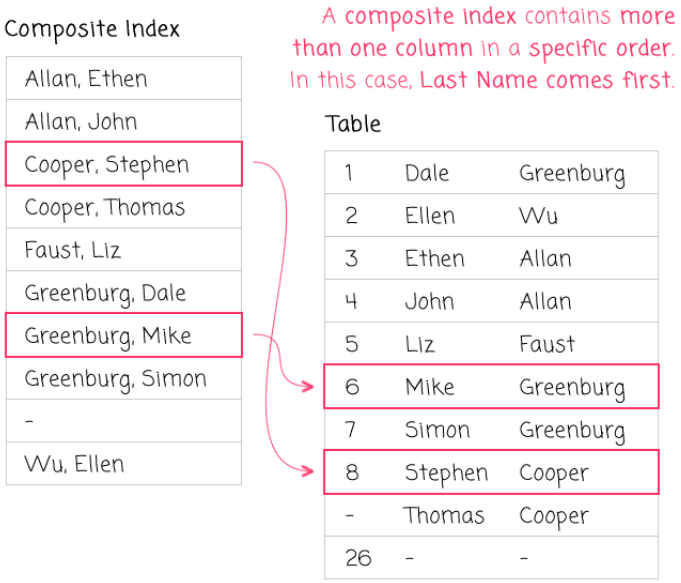
Syntax

```
CREATE INDEX [index name]
ON [Table name]([column1, column2, column3,...]);
```

- **Composite index**

a composite index, multiple columns in the same table (sometimes called a

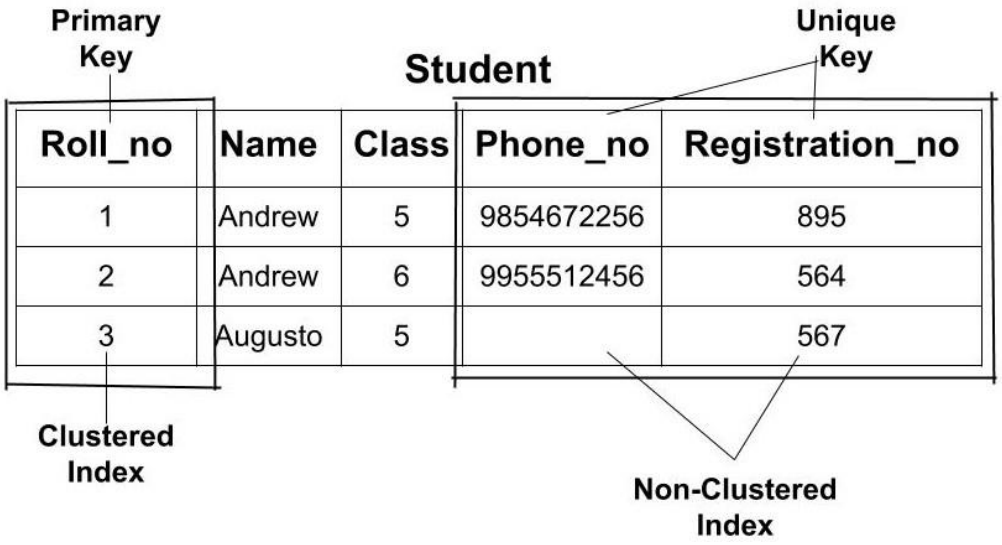
"compound" index). When an index contains multiple columns, the order of the columns in the index matters. You can only leverage a composite index to search for columns in the same order in which the index provides them



- **Clustered index**

Every table in the database, no matter how small, should have a primary key (sometimes called a "clustered index"). The primary key is the column - or set of columns - that uniquely identifies every individual record in a table. This primary key is then used to search for and act upon a single, specific record

Example:



Write definitions for each types and provide an example (you can find examples online, but you need to write them down for each type)

Add a new index to the 'Sweet' table in Bakery database (any column -explain your choice)

CREATE UNIQUE INDEX IDX_ITEM_NAME ON SWEET(ITEM_NAME);

ITEM_NAME is also currently unique, so we may want to enforce uniqueness easily with an index.

Add a new index (multi-column) to the table 'Accounts' in the Bank database (explain your choice of columns).

CREATE INDEX IDX_NAME
ON ACCOUNTS(ACCOUNT_HOLDER_NAME, ACCOUNT_HOLDER_SURNAME)

IDX_NAME could be useful when we are searching customers' full name, could save a lot of time.

TASK 2 (SQL)

- Study the extra curriculum topics about Triggers and Events (**sql_session6 slide deck marked HOMEWORK SLIDES)**
- Review and run exercises provided to understand how Triggers and Events work (use scripts in your handouts **example_event.sql** and **example_trigger.sql**)

(you do not need to submit anything for this task, but you may come across these topics in the future assessments and interviews)

TASK 3 (Python)

Question 1

I am building some very high-quality chairs and need exactly four nails for each chair. I've written a program to calculate how many nails I need to buy to build these chairs.

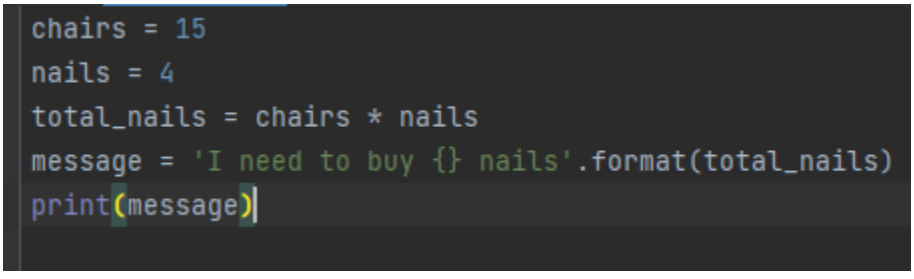
```
chairs = '15'

nails = 4

total_nails = chairs * nails

message = 'I need to buy {} nails'.format(total_nails)

print(message)
```

When I run the program, it tells me that I need to buy 15151515 nails. This seems like a lot of nails.

Is my program calculating the total number of nails correctly? What is the problem? How do I fix it?

Answer: No, because you treated the number of chairs as string instead integer

Right answer: remove the quote sign



Question 2

I'm trying to run this program, but I get an error. What is the error telling me is wrong?

How do I fix the program?

```
my_name = Penelope

my_age = 29

message = 'My name is {} and I am {} years old'.format(my_name, my_age)

print(message)
```

Answer: the data type for my_name variable should be string, so quote sign is needed

Right answer below:

```
my_name = 'Penelope'

my_age = 29

message = 'My name is {} and I am {} years old'.format(my_name, my_age)

print(message)
```

Question 3

I have a lot of boxes of eggs in my fridge and I want to calculate how many omelettes I can make. Write a program to calculate this.

Assume that a box of eggs contains six eggs and I need four eggs for each omelette, but I should be able to easily change these values if I want. The output should say something like:

"*You can make 9 omelettes with 6 boxes of eggs*"

```
boxes_eggs = int(input())

num_omelette = boxes_eggs * 6 // 4

ans_2= f'You can make {num_omelette} omelettes with {boxes_eggs} boxes of eggs'

 print(ans_2)
```

Question 4

Complete a series of tasks to format strings

```
# Task 1 - Replace the (.) character with (!) instead. Output should be "I love coding!"

my_str = "I love coding."

my_str = my_str.replace('.' , '!')

print(my_str)
```

```
# Task 2 - Reassign str so that all of its characters are lowercase.

my_str_1 = "EVERY Exercise Brings Me Closer to Completing my GOALS."

my_str_1 = my_str_1.lower()

print(my_str_1)
```

```
# Task 3 - Output whether this string starts with an A or not

my_str_2 = "We enjoy travelling"
```

```
ans_1 = my_str_2.startswith('A')

print(ans_1)
```

```
# Task4 - What is the length of the given string?

my_str_3="1.458.001"

print(len(my_str_3))
```

Question 5

Complete a series of tasks to slice strings

```
# Task 1 - Slice the word so that you get "thon".

wrd="Python"

ans_1=  wrd[2:]

print(ans_1)
```

```
# Task 2 - Slice the word until "o". (Pyth)

wrd="Python"

ans_1= wrd[:4]

print(ans_1)
```

```
# Task 3 - Now try to get "th" only.
```

```
wrd="Python"

ans_1=wrd[2:4]

print(ans_1)
```

```
# Task 4 - Now slice the word with steps of 2, excluding first and last characters

wrd="Python"

# Type your answer here.

ans_1=wrd[1:5:2]

print(ans_1)
```

Question 6

Explain what this program does

```
for number in range(100):

        output = 'o' * number

        print(output)
```

Answer: printing ‘o’ on the each line, ascending number of ‘o’ by adding one more’o’ until there are 99 ‘o’ in the 99th row.

Question 7

Your boss really likes calculating VAT on their purchases. It is their favourite hobby. They've written this code to calculate VAT and need your help to fix it.

```
def calculate_vat(amount):

        amount * 1.2
```

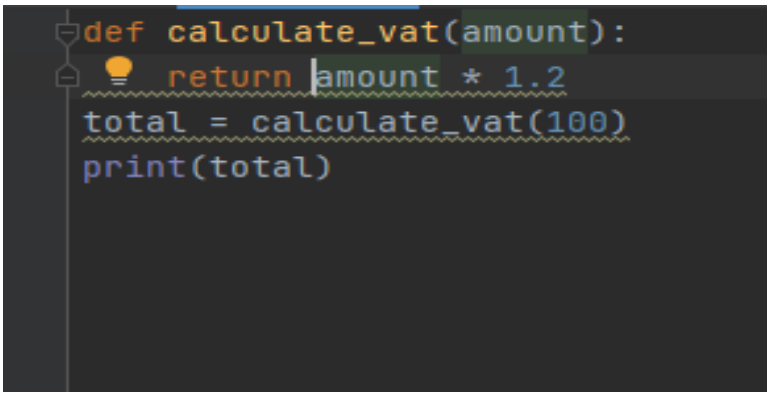
```
total = calculate_vat(100)

print(total)
```

When your boss runs the program they get the following output:

```
None
```

Your boss expects the program to output the value 120 . What is wrong? How do you fix it?

Answer: because the boss's function didn't return anything.



Question 8

Write a new function to print a 'cashier receipt' output for a shop (any shop – clothes, food, events etc).

It should accept 3 items, then sum them up and print out a detailed receipt with TOTAL.

For example:

Input:

Item_1_name = 'Trainers'

Item_1_price = 50.45

Item_2_name = 'T-shirt

Item_2_price = 12

Output:

Trainers 50.45

T-shirt.....12.00

TOTAL 62.45

Solution1:

```
def cashier_receipt(Item_1_name, Item_1_price, Item_2_name, Item_2_price, Item_3_name, Item_3_price):
    return f'{Item_1_name}.....{Item_1_price}\n' \
           f'{Item_2_name}.....{Item_2_price}\n' \
           f'{Item_3_name}.....{Item_3_price}\n' \
           f'Total.....{Item_1_price + Item_2_price + Item_3_price}'


print(cashier_receipt('Trainers', 50.45, 'T - shirt', 12, 'Dress', 30))
```

Solution 2:

ain.py print.py

```
def cashier_receipt(items):
    counter = 0
    for key in items:
        counter += items[key]
    return counter


items = {'trainers': 50.45, 'T-shirt': 12, 'dress': 30}

for key in (items):
    print(f'{key}.....{items[key]}\n')

print(f'Total.....{cashier_receipt(items)}')
```