

## Lecture 9: August 4

*Lecturer: Anwar Mamat*

**Disclaimer:** *These notes may be distributed outside this class only with the permission of the Instructor.*

## 9.1 String Permute

```
1 used =[]
2 #recursive function
3 def permute(input_string , output_string ,index):
4     global used
5     if(index == len(input_string)):
6         print(output_string)
7         return
8     for i in range(len(input_string)):
9         if(used[i] == 0):
10            #print(input_string[i], end=",")
11            output_string[index] = input_string[i]
12            used[i] = 1
13            permute(input_string , output_string , index+1)
14            used[i] = 0
15 #iterative process
16 def permutel(input_string , output_string ,index):
17     global used
18     for i in range(len(input_string)):
19         if(used[i] == 0):
20            #print(input_string[i], end=",")
21            output_string[index] = input_string[i]
22            used[i] = 1
23            permute2(input_string , output_string ,1)
24            used[i] = 0
25
26 def permute2(input_string , output_string , index):
27     global used
28     for i in range(len(input_string)):
29         if(used[i] == 0):
30            used[i] = 1
31            #print(input_string[i], end=",")
32            output_string[index] = input_string[i]
33            permute3(input_string , output_string ,2)
34            used[i] = 0
35
36 def permute3(input_string , output_string ,index):
37     global used
38     for i in range(len(input_string)):
```

```

39     if(used[i] == 0):
40         used[i] = 1
41         #print(input_string[i], end=",")
42         output_string[index] = input_string[i]
43         permute4(input_string, output_string, 3)
44         used[i] = 0
45
46 def permute4(input_string, output_string, index):
47     global used
48     for i in range(len(input_string)):
49         if(used[i] == 0):
50             used[i] = 1
51             #print(input_string[i])
52             output_string[index] = input_string[i]
53             print(output_string)
54             used[i] = 0
55
56 def main():
57     global used
58     s = "ABCD"
59     out_string=[0 for i in range(len(s))]
60     used = [0 for i in range(len(s))]
61     permute(s, out_string, 0) #call permute1 for iterative process
62 main()

```

## 9.2 Sqlite3 Database

### 9.2.1 Database Example 1

```

1 import sqlite3
2
3 def main():
4     db_name = 'students.db'
5     db = sqlite3.connect(db_name)
6     db.row_factory = sqlite3.Row
7     db.execute('drop_table_if_exists_students')
8     db.execute('create_table_students(id_int_primary_key, name_text, email_text)')
9
10    db.execute('insert_into_students_values(5, "ethan", "000-4567")')
11
12    cursor = db.execute('select_*_from_students')
13    for row in cursor:
14        print(dict(row))
15        #print(row['name'])
16
17    db.execute('update_students_set_name="Alice10" where_id=1')
18    print('after_delete')
19    db.execute('delete_from_students_where_id=5')

```

```

20
21     db.commit()
22     #db.rollback()
23
24     cursor = db.execute('select *_from_students')
25     for row in cursor:
26         print(dict(row))    #print(row['name'])
27 main()

```

### 9.2.2 Database Example 2

```

1  import sqlite3
2
3  def update(db, row):
4      db.execute('update_tasks_set_name=?_where_id=?', (row['name'], row['id']))
5      db.commit()
6
7  def insert(db, row):
8      db.execute('insert_into_tasks(id, name, type)_values(?, ?, ?)',
9                 (row['id'], row['name'], row['type']))
10     db.commit()
11
12  def delete(db, id):
13     db.execute('delete_from_tasks_where_id={}'.format(id))
14     db.commit()
15
16  def disp_rows(db):
17     cursor = db.execute('select *_from_tasks_order_by_id')
18     for row in cursor:
19         print('id: {} \t name: {}'.format(row['id'], row['name']))
20  def main():
21     db = sqlite3.connect('tasks.db')
22     db.row_factory = sqlite3.Row
23     print('creating_table_tasks')
24     db.execute('drop_table_if_exists_tasks')
25     db.execute('create_table_tasks(id_int_primary_key, name_text, _type_int)')
26     db.execute('alter_table_tasks_add_column_finish_date_date')
27     print('insert_records')
28     insert(db, dict(id=1, name='take_out_trash', type=1))
29     insert(db, dict(id=2, name='wake_up_on_time', type=2))
30     insert(db, dict(id=3, name='wake_up_on_time', type=3))
31     for i in range(10):
32         insert(db, dict(id=i+4, name='abc', type=i*2))
33         db.commit()
34
35     disp_rows(db)
36     id = 3
37     delete(db, id)
38     print('after_delete')

```

```
39     disp_rows(db)
40     print('after_update')
41     update(db, dict(id=1,name='pick_up_trash'))
42     disp_rows(db)
43
44 main()
```