

README עבור תוכנת סימולציה RA

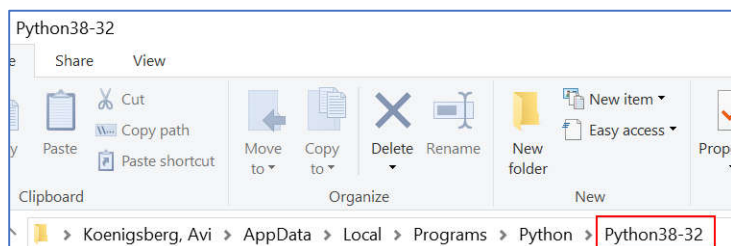
לתוכנה אין UI מרגש, אז לא להתלהב.

יש צורך להשתמש בפייתון, אז אם אין לכם במחשב, תורידו, או באסה לכם.

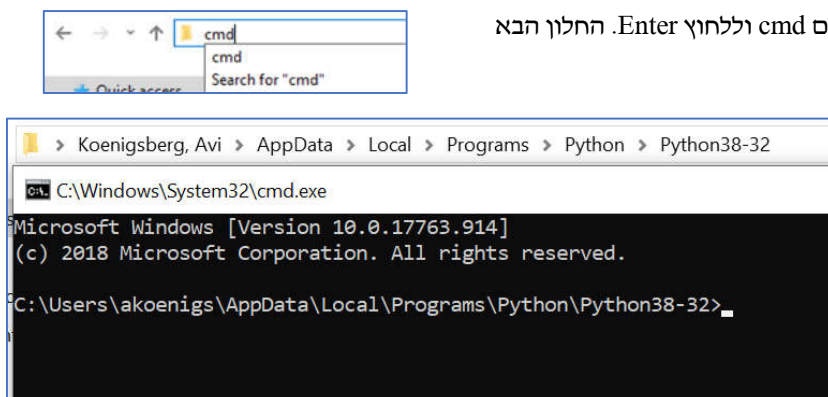
התוכנה משתמשת בספריה פייתונית בשם prettytable על מנת להציג טבלאות בצורה נוחה. הספריה דורשת הורדה.

1. הורדת הספריה prettytable:

א. כניסה לתיקיה בה נמצאת התוכנה של פייתון



ב. בשורת הpath, לרשום cmd וללחוץ Enter. החלון הבא ייפתח:



ג. להריץ את הפקודה

python -m pip install prettytable

הפקודה הזו תוריד את הספריה.

2. מה כלול:

א. הסקריפט כולל שני קבצים. הראשון בשם Table.py הוא קובץ שמכיל את המחלקות והפונקציות, והשני בשם table_env.py זה סביבה בשביל לעשות שטויות ולשחק. אין צורך לערוך את הקובץ Table אבל חופשי.

ב. בקובץ Table יש את המחלקות הבאות:

I. Table : ליצירת סכימות והוספת מופעים.

II. Condition : ליצירת תנאים (לשימוש באופרטור select ובאופרטור cond_join [שזה בעצם conditional join]).

III. סוגי תנאים:

cond_v – השוואת עמודה לערך מסויים, כמו (year <= 3). (value)
cond_c – השוואת עמודה לעמודה אחרת, כמו (R.pid = S.pid). (column)
cond_p – תנאי שמכיל תנאים אחרים ומפעיל עליהם לוגיקה נתונה, כמו (pid = 20 & year < 3) (parent)

3. שימוש במחלקה Table :

הבנאי של המחלקה מקבל כפרמטר ראשון את שם הטבלה, ושאר פרמטרים יהיו מחזוריות המייצגות את העמודות.

```
>>> T = Table("DB", "col_a", "col_b")
>>> T.print()
Table DB:
+-----+
| col_a | col_b |
+-----+
```

הוספת שורה לטבלה: לטבלה יש מתודה בשם add. המתודה מכריחה להשתמש במילות מפתח המתאימות לסכימה של הטבלה.

```
>>> T = Table("Students", "sid", "name")
>>> T.add(101, "Maoz")
Traceback (most recent call last):
  File "<pyshell#45>", line 1, in <module>
    T.add(101, "Maoz")
TypeError: add() takes 1 positional argument but 3 were given
>>> T.add(sid=101, name="Maoz")
>>> T.print()
Table Students:
+-----+
| sid | name |
+-----+
| 101 | Maoz |
+-----+
```

אופרטורים (כל אופרטור מחזיר מופע חדש של טבלה):

```
>>> A.difference(B).print()
Table A-B:
+-----+
| a | b |
+-----+
| 1 | 2 |
+-----+
```

```
>>> B.add(a=5, b=6)
>>> A.add(a=5, b=6)
>>> A.intersection(B).print()
Table intersection A∩B:
+-----+
| a | b |
+-----+
| 5 | 6 |
+-----+
```

```
>>> T.project("sid").print()
Table π(sid) Students:
+-----+
| sid |
+-----+
| 101 |
| 102 |
| 103 |
+-----+
```

```
>>> A = Table("A", "a", "b")
>>> B = Table("B", "a", "b")
>>> A.add(a=1, b=2)
>>> B.add(a=3, b=4)
>>> A.union(B).print()
Table A∪B:
+-----+
| a | b |
+-----+
| 1 | 2 |
| 3 | 4 |
+-----+
```

```
>>> T.print()
Table Students:
+-----+
| sid | name |
+-----+
| 101 | Maoz |
| 102 | Evyatar |
| 103 | Nerya |
+-----+

>>> R = T.rename("diar", "rhea")

>>> R.print()
Table ρ(diar, rhea) Students:
+-----+
| diar | rhea |
+-----+
| 101 | Maoz |
| 102 | Evyatar |
| 103 | Nerya |
+-----+
```

```
>>> R.natural_join(S).print()
Table R⋈S:
+-----+
| tid | sid | course | sname | year |
+-----+
| 20 | 202 | os | jones | 3 |
| 10 | 450 | calculus | adams | 1 |
| 20 | 202 | db | jones | 3 |
| 20 | 240 | db | white | 3 |
+-----+
```

```
>>> E.print()
Table E:
+-----+
| tid | sid | course | sname | year |
+-----+
| 20 | 202 | os | jones | 3 |
| 10 | 450 | calculus | adams | 1 |
| 20 | 202 | db | jones | 3 |
| 20 | 240 | db | white | 3 |
+-----+
```

```
>>> P.print()
Table P:
+-----+
| sname | year |
+-----+
| jones | 3 |
+-----+

>>> E.division(P).print()
Table E÷P:
+-----+
| tid | sid | course |
+-----+
| 20 | 202 | os |
| 20 | 202 | db |
+-----+
```

```
>>> R.cart_prod(S).print()
Table R×S:
+-----+
| tid | R.sid | course | S.sid | sname | year |
+-----+
| 20 | 202 | os | 240 | white | 3 |
| 20 | 202 | os | 202 | jones | 3 |
| 20 | 202 | os | 450 | adams | 1 |
| 10 | 450 | calculus | 240 | white | 3 |
| 10 | 450 | calculus | 202 | jones | 3 |
| 10 | 450 | calculus | 450 | adams | 1 |
| 20 | 202 | db | 240 | white | 3 |
| 20 | 202 | db | 202 | jones | 3 |
| 20 | 202 | db | 450 | adams | 1 |
| 20 | 240 | db | 240 | white | 3 |
| 20 | 240 | db | 202 | jones | 3 |
| 20 | 240 | db | 450 | adams | 1 |
+-----+
```

```
Table R:
+-----+
| tid | sid | course |
+-----+
| 20 | 202 | os |
| 10 | 450 | calculus |
| 20 | 202 | db |
| 20 | 240 | db |
+-----+

Table S:
+-----+
| sid | sname | year |
+-----+
| 240 | white | 3 |
| 202 | jones | 3 |
| 450 | adams | 1 |
+-----+
```

הערה: במכפלה קרטזית, אם יש שתי עמודות זהות בשם, כל אחת תקבל קידומת של שם הטבלה ממנה העמודה באה, כפי שניתן לראות בדוגמא. שינוי זה של השם הוא לכל דבר, ואם יתבצע על טבלה כזו project עם השם של העמודה בלי הקידומת (או אופרטור אחר שדורש שתהיה עמודה עם השם בלי הקידומת) האופרטור לא ייתן את התוצאה הרצויה. כמו כן, במכפלה קרטזית בין שתי טבלאות שלהן יש שם זהה, במקום קידומת של שמות הטבלאות, העמודות הזהות תקבלנה סיומת עם מספר 1,2 בהתאמה, כמובא למטה בדוגמא. גם שינויים אלה הם לכל דבר.

האופרטור natural_join יודע להתמודד עם זה, אז אל דאגה. אך לא לתת שמות לכתחילה לעמודות עם 1,2 בסוף, זה יכול לבלבל את האופרטור.

```
>>> A = Table("A", "col")
>>> A.cart_prod(A).print()
Table A×A:
+-----+
| col1 | col2 |
+-----+
```

4. שימוש במחלקה Condition:

לסוגי התנאים יש בנאים משלהם.

תנאי שמשווה עמודה לערך צריך לקבל את שם העמודה, האופרטור, ואת הערך.

תנאי שמשווה עמודה לעמודה צריך לקבל את שם העמודה, האופרטור, ואת שם העמודה השניה.

```
>>> c1 = cond_v("sid", ">", 230)
>>> c2 = cond_c("year1", "!=", "year2")
```

תנאי אב מקבל לוגיקה And או Or, ותנאי בן.

ניתן לצרף תנאי אב לתוך תנאי אב.

(המילים And וOr מוגדרות בתוך הקובץ Table.py).

```
>>> c3 = cond_p(And, c1, c2)
```

ניתן להדפיס תנאים כדי לראות את התצוגה היותר מוכרת שלהם.

```
>>> print(c1)
sid>230
>>> print(c2)
year1!=year2
>>> print(c3)
(sid>230^year1!=year2)
>>>
```

```
>>> c4 = cond_v("course", "=", "db")
>>> c5 = cond_p(Or, c3, c4)
>>> print(c5)
((sid>230^year1!=year2)^course=db)
```

האופרטורים select וjoin מקבלים תנאים:

```
>>> S.cond_join(R, cond_c("S.sid", "=", "R.sid")).print()
Table S^R(S.sid=R.sid)R:
+-----+-----+-----+-----+-----+
| S.sid | sname | year | tid | R.sid | course |
+-----+-----+-----+-----+-----+
| 240   | white | 3    | 20  | 240   | db      |
| 202   | jones | 3    | 20  | 202   | os      |
| 202   | jones | 3    | 20  | 202   | db      |
| 450   | adams | 1    | 10  | 450   | calculus|
```

```
>>> R.print()
Table R:
+-----+-----+-----+
| tid | sid | course |
+-----+-----+-----+
| 20  | 202 | os      |
| 10  | 450 | calculus|
| 20  | 202 | db      |
| 20  | 240 | db      |
+-----+-----+-----+

>>> c1 = cond_v("tid", "=", 10)
>>> R.select(c1).print()
Table σ(tid=10)R:
+-----+-----+-----+
| tid | sid | course |
+-----+-----+-----+
| 10  | 450 | calculus|
+-----+-----+-----+
```

אין לי נציג QA, אז ייתכן שיהיו באגים. אשמח מאוד, במידה ויימצא באג, אם הבאג ידווח אליי במיידית. רוב הסיכויים שזה פיצ'ר בכל אופן.

אני מודע לכך שקיימים כלים כאלה באינטרנט, אבל איני יודע להשתמש בהם, ולכן במקום ללמוד איך להשתמש בהם, עשיתי מה שכל אדם עם שכל בראשו היה עושה, וישבתי 6 שעות לכתוב את התוכנה הזו. אין דבר שאני אוהב יותר מקרדיט. טוב לי תורת פיך מאלפי זהב וכסף, וטוב לי קרדיט מתורת פיך. אז אם אהבתם, או שלא אהבתם אבל יש לכם סבלנות לעבוד עלי שאהבתם, תגידו לי.

שלכם תמיד, AviK,

יש עוד עמוד, לא לפספס.

דוגמא מהמצגת :

Find Pairs of Students who are in the Same Year (Version 3)

$$\pi_{sn1, sname}((\rho_{R(si1, sn1, y1)}(S)) \bowtie_{y1=year \wedge si1 \neq sid} S)$$

S	sid	sname	year
	240	white	3
	202	jones	3
	450	adams	1

→

sn1	sname
white	jones
jones	white

```
>>> S.print()
Table S:
+-----+-----+-----+
| sid | sname | year |
+-----+-----+-----+
| 240 | white | 3    |
| 202 | jones | 3    |
| 450 | adams | 1    |
+-----+-----+-----+

>>> c1 = cond_c("y1", "=", "year")
>>> c2 = cond_c("si1", "!=", "sid")
>>> cp = cond_p(And, c1, c2)
>>> S.rename("si1", "sn1", "y1").cond_join(S, cp).project("sn1", "sname").print()

Table π(sn1, sname)ρ(si1, sn1, y1)S ⋈ ((y1=year ∧ si1!=sid))S:
+-----+-----+
| sn1 | sname |
+-----+-----+
| white | jones |
| jones | white |
+-----+-----+
```