گزارش کار
درس مبانی رایانش امن

بهار ۱۳۹۹

دانشگاه یزد

استاد درس

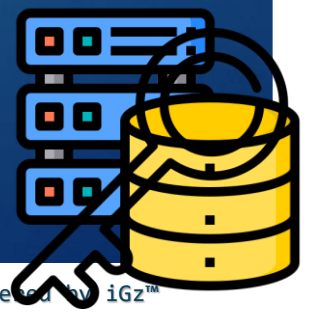
دکتر ادیب نیا

ارائه دهندگان

علیرضا اکرمی ابرقویی

امیرحسین محمدی

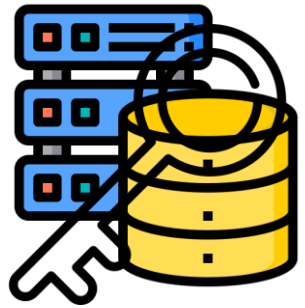
ایمان قشقایی زاده



تعریف پروژه

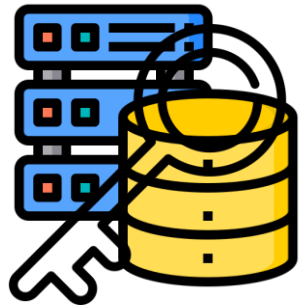
- برنامه‌ای جهت رمزنگاری پایگاه داده؛ مشخصات جدول، ستون ها و باقی موارد اختیاری.

- به وسیله‌ی این برنامه می توان جدول ابرداده را به دست آورده و با فرض درخواست کاربر به کلاینت کوئری تصحیح شده به جدول ابرداده ارسال می‌شود. نتیجه‌ی دریافتی از سرور رمزگشایی شده و دوباره گوئری کاربر روی آن اجرا می‌شود. تا نتیجه‌ی دقیق به وی نمایش داده شود.



گام اول

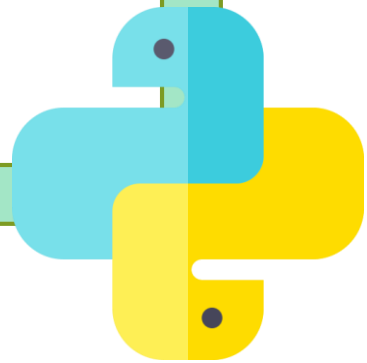
- در گام اول نیازمندی های نرم‌افزاری موجود را بررسی کرده و در صورت نیاز اقدام به تهیه و نصب آن ها می کنیم.
  - آخرین ویرایش از زبان پایتون و پکیج های مورد نیاز از جمله :
    - IPython-sql
    - Sqlite3
    - Prettey_table
    - Cryptography
  - نصب نرم‌افزارهای Anaconda و NoteBook Jupyter به واسطه‌ی آن



اتصال ژوپیتر به پایگاه داده

با استفاده از قطعه کد مقابل پایگاه داده ی ایجـاد شـده بازیـابی مـی شـود و امکـان نوشتن دستورات در محیط ژوپیتـر فـراهم می شود.

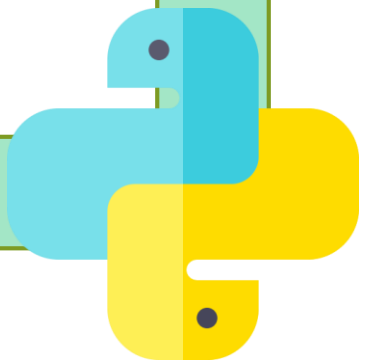
- %load_ext sql
- %sql sqlite:///test.db
- %reload_ext sql



ایجاد جداول

ایجاد جدول اصلی، در واقع این جدول حاوی داده های بدون رمز و بدون دسته بندی است.

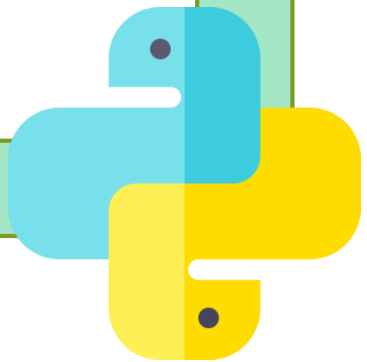
```
%%sql
create table raw_data(
sid int PRIMARY KEY,
nid int,
age int,
grade int
)
```



وارد کردن داده‌ها

با استفاده از دستور مقابل داده های مورد نیاز و بدون هیچ گونه رمز گذاری در جدول اصلی درج می کنیم.

```
%%sql
insert into raw_data values(35,350,30,20)
```



نمایش جدول ایجاد شده

جدول ایجاد شده پس از اینکه داده های مورد نیاز را در آن درج کردیم.

- %%sql
- SELECT *
- FROM raw_data

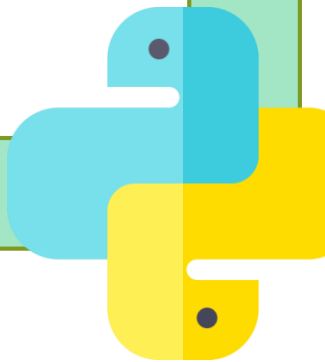
Out put

sid	nid	age	grade
5	55	12	14
15	150	21	18
25	250	25	19
35	350	30	20

بازیابی هر رکورد های از جدول جهت رمزگذاری

جهت ایجاد جدول ابر داده نیازمندیم تا ابتدا از جدول اصلی هر رکورد را به صورت کامل فراخوانی کنیم و پس از اعمال رمز گذاری در جدول ابر داده درج کنیم.

```
import sqlite3
conn=sqlite3.connect('mydb.db')
c=conn.cursor()
query=("""SELECT *
FROM raw_data""")
y=c.execute(query)
print(y)
```



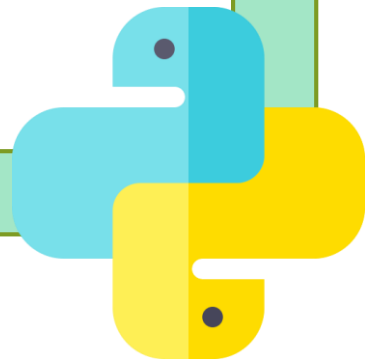
Out put

```
<sqlite3.Cursor object at 0x000002410448CD50>
```

ایجاد جدول ابر داده

هر رکورد را به صورت یک tuple از جدول اصلی استحراج کردیم تا در مرحله با استافده از این داده ها جدول ابر داده را ایجاد کنیم.

```
l=[]
for row in c.execute(query):
    l.append(str(row))
    print(row)
```



Out put

```
(5, 55, 12, 14)
(15, 150, 21, 18)
(25, 250, 25, 19)
(35, 350, 30, 20)
```

دسته بندی داده ها
جهت ساخت جدول ابر داده

برای دسته بندی داده ها برای شناسه sid دسته هارا ده تایی انتخاب کردیم به این صورت که ۰ تا ۹ دسته ۱، ۱۰ تا ۱۹ دسته ۲ و الی آخر....

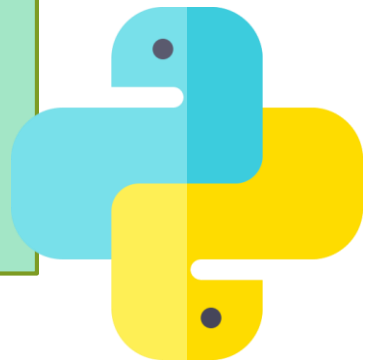
برای شناسه nid از ۰ تا ۹۹ دسته ۱، از ۱۰۰ تا ۲۹۹ دسته ۲ ، و از ۳۰۰ تا ۴۰۰ دسته ۳

برای شناسه age از ۰ تا ۹ دسته ۱، ۱۰ تا ۱۹ دسته ۲، از ۲۰ تا ۲۹ دسته ۳، و از ۳۰ تا ۴۰ دسته ۴

برای grade ۰ تا ۴ دسته ۱، ۵ تا ۹ دسته ۲، ۱۰ تا ۱۴ دسته ۳، ۱۵ تا ۲۰ دسته ۴

جدول اصلی

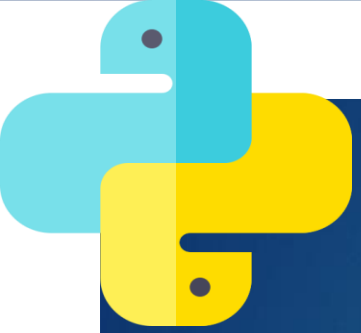
sid	nid	age	grade
5	55	12	14
15	150	21	18
25	250	25	19
35	350	30	20



Out put

lsid	lnid	lage	lgrade
1	1	2	3
2	2	3	4
3	2	3	4
4	3	4	4

جدول ابر داده



استفاده از کتابخانه‌ی Cryptography و شی داده‌ای fernet

Fernet also uses 128-bit AES in CBC mode and PKCS7 padding, with HMAC using SHA256 for authentication.

رمز گذری داره های اصلی وایجاد جدول ابر داده

جهت ایجاد جدول ابرداده علاوه بر دسته بندی داده ها و ایندکس گذاری به رمز گذاری هر رکورد از جدول اصلی و دخیره سازی در جدول ابرداده

با استفاده از کتابخانه **cryptography** و ماژول **Fernet** و هم چنین دو تابع write_key() جهت ایجاد کلید و load_key ذخیر سازی آن و تابع جهت بازیابی کلید استفاده می شود.

```
from cryptography.fernet import Fernet
def write_key():
    """
    Generates a key and save it into a file
    """
    key = Fernet.generate_key()
    with open("key.key", "wb") as key_file:
        key_file.write(key)
def load_key():
    """
    Loads the key from the current directory named `key.key`
    """
    return open("key.key", "rb").read()
```

Out put

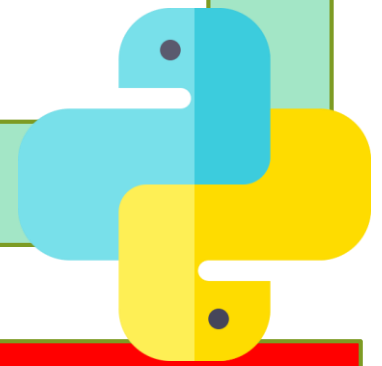
key.key

فایل محتوی کلید داده های رمز شده

رمزگذاری هر رکورد
با استفده از کلید تولید
شده

با استفاده از تابع load_key کلید تولید شده را بازیابی میکنیم و با استفاده از آن هر رکورد جدول را مطابق کد مقابل و تابع encrypt را رمز می کنیم و خروجی زیر به عنوان مقابل تولید می شود.

```
write_key()
key = load_key()
message = f"{l[3]}".encode()
f = Fernet(key)
encrypted = f.encrypt(message)
print(encrypted)
decrypted_encrypted = f.decrypt(encrypted)
print(decrypted_encrypted)
```



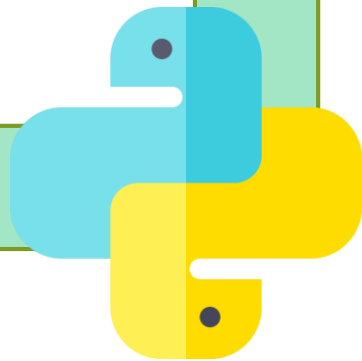
Out put

```
b'gAAAAABesKAKO75HvKWoYQIdgxRvCU8xlulWaqM4RwHgDo4XTE8JMB_u4GDnnEIUwwnpYrRvUuDdL8KDE9fmOyWEb0Oq7FqQIkzXivg0LZX8v44iWyD
DRVE='
b'(35, 350, 30, 20)'
```

ایجاد جدول ابر داده

با استفاده از کد مقابل جدول مورد نظر جهت ذخیر سازی داد های رمز شده و دست بندی شده ایجاد می شود.

```
%%sql
create table encrypt_data(
    encrypt varchar(200) PRIMARY KEY,
    Isid INTEGER,
    Inid INTEGER,
    Iage INTEGER,
    Igrade INTEGER
)
```



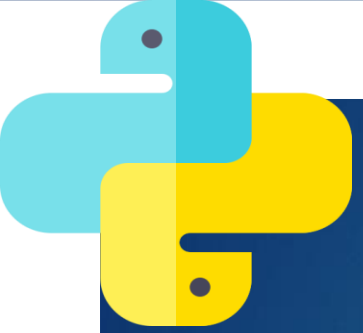
درج اطلاعات در جدول ابر داده

با استفاده از دستور مقابل داده های مورد نیاز و بدون هیچ گونه رمز گذاری در جدول اصلی درج می کنیم.

```
%%sql
INSERT INTO encrypt_data1 values('gAAAAABesJCZ5NUCMbx930MGjMpPdtrWIuC0tZBB05y7WitJkGKeQCpCMRRHWsN8X3-5C0zV0Kvc0OJnLWPmg6JRdNztDopx0TLAb3ZvCUMtJsD0mh2w8sA=',4,3,4,4)
```

Out put

encrypt	lsid	lnid	lage	lgrade
gAAAAABesloc-C7nW4-BuSQjUhcd2DX_UhES2Nxv9Yjv1gP8NTFTHuqeLFuTn7FVgLkF2LABhh05pagHBYSrizwFkQmWN6dY9Q==	1	1	2	3
gAAAAABesl-hJYTe-rsBKimkpj3tHLbQBH9xT-aBF3rLLuj0wSJ6jTTpqLloKN0wUGa5NQheTbXx6J8GKnyxGH3Wsd00hvH3KKDKM4wgy7prwk67Y5zbHpY=	2	2	3	4
gAAAAABesJAD2FX8_6XaE0d_pJaXlOj31xSSMDvOnvv9splcyCZlSqRfp2pfxU_LW1QogPKyH50wcbkClRDBlhnDDZ7uTyj8TRyuE71qPa-P4Urde0Fu690=	3	2	3	4
AABesJCZ5NUCMbx930MGjMpPdtrWIuC0tZBB05y7WitJkGKeQCpCMRRHWsN8X3-5C0zV0Kvc0OJnLWPmg6JRdNztDopx0TLAb3ZvCUMtJsD0mh2w8sA=	4	3	4	4



ایجاد اپلیکیشنی، جهت ارسال نتایج کوئری های کاربران با توجه به محدودتی هایی که اعمال می کنند. در این برنامه ابتدا کوئری کابران به کوئری مناسب جهت بازیابی داده های رمزگذاری شده از جدول ابر داده تبدیل می شود و داده های بازیابی شده را از فرم رمز شده خارج می کند و پس از اعمال فیلترینگ نتیجه را به کاربر باز میگرداند. در اسلاید بعد نمونه ای از تعامل با این برنامه نشان داده می شود.

Our program

```
**********
welcom to SarvSearch
# menu
1 : search StudentID
2 : search NationalID
3 : search Age
4 : search Grade
**********
please Select a number from 1 to 4 :2
enter Inid: 150
your Query Result is:
+-----+-----+-----+-------+
| sid | nid | age | grade |
+-----+-----+-----+-------+
|  15 | 150 |  21 |   18  |
+-----+-----+-----+-------+


More details:

===>>>Recovered records from BigData:
["('gAAAAABesLVlHD2WTlmz0UKs7TGJ2UMDDVla6ShCz7xaNse42YVPFKhq96Jau0psKrDvxTsN3dKYgVyGAubCzIzz24XSwyS77RX35TX6RMRUtA4zE
Lme8lU=',)", "('gAAAAABesLV6Zia5TvmWG4CbT-F79ZHcRkdrZBTCMbVabGz1DGHlaQ0IgOYYpn4gzRTAjjs-s-zCjVxwTt7Noam4pXpNHdrSkf5gt
6lQCbOYjPu8N8A_oBg=',)"]

===>>>encrypted data is:b'gAAAAABesLVlHD2WTlmz0UKs7TGJ2UMDDVla6ShCz7xaNse42YVPFKhq96Jau0psKrDvxTsN3dKYgVyGAubCzIzz24X
SwyS77RX35TX6RMRUtA4zELme8lU='

*****The generated key is==>>> b'CiWj2kb9hg3SXUBVYXrfDcbXe28Iy1E5a5SFHyCPq5M='******
```

در ابتدا منویی به کاربر نمایش داده می شود که با استفاده از آن می تواند ار بین شناسه های جدول اصلی اونی که میخواهد بر اساس آن محدودیتی در جدول اصلی اعمال کند را انتخاب کند.

پس از این مرحله کاربر مقداری که به دنبال آن است را وارد می کند. و پس از این مرحله ابتدا ورودی کاربر به پردازش شده و به شکل مقادیر متناسب جهت جستجو در جدول ابرداده تبدیل کنیم. پس از این مرحله در جدول ابر داده داده های متناسب را بازیابی می کنیم و پس از رمز گشایی و فیلترینگ به کاربر نمایش می دهیم.

داده های بازیابی از جدول ابر داده
پیش از انجام فیلترینگ

پس از نمایش کوئری مور نظر کاربر اطلاعاتی مثل عنوان کلید مورد استفاده جهت رمز کردن داده ها و داده ی مورد نظر که از بین داده های بازیابی شده از جدول ابر داده فیلتر شده و به شکل رمز شده به کاربر نمایش داده می شود.

اندر وادی کار گروهی در ساعات مختلف شبانه روز، همراه با تلاشی که زمان را از یاد ما برد.

تهران - ابرکوه - شیراز

بهار ۱۳۹۹

"هیچ گنجی در نزد تلاش‌گران، پنهان نخواهد ماند."

ا.ق. اردی بهشت

- علیـرضا اکرمی ابرقـویی
- امـیـرحسـین محـمـدی
- ایمـان قـشـــقایی زاده

- گزارش کار درس مبانی رایانش امن
  - استاد درس

    دکتر ادیب نیا
  - دانشگاه یزد
  - بهار ۱۳۹۹