

## سوالات CW:

1.

PostgreSQL را روی سیستم عامل خود (Windows / Linux) نصب کنید.  
پس از نصب، سرویس PostgreSQL را اجرا و مطمئن شوید که فعال است.  
با استفاده از ابزار psql یا ابزار گرافیکی مثل pgAdmin وارد سیستم شوید (pgAdmin را نصب کنید بر روی سیستم خود)  
نیاز نیست دیتابیس یا کوئری بنویسید صرفاً نصب کافیست

### تفاوت DBMS و RDBMS چیست؟

جدول زیر را بررسی کنید و مشخص کنید در کدام نرمال فرمها نقض دارد:

StudentID	StudentName	Course	Teacher	TeacherPhone
1	Ali	Math	Karimi	09120001
1	Ali	Physics	Ahmadi	09124444
2	Sara	Math	Karimi	09120001

این جدول از نظر 1NF چه مشکلی دارد؟

چرا 2NF را نقض می کند؟

چرا 3NF را نقض می کند؟

جدول را تا 3NF نرمال کنید.

اگر هر کدام از سوالات بالا رو بلد نبودید صرفاً درباره ی آنها سرچ کنید.

2.

در این تمرین باید یک برنامه پایتون طراحی کنید که فعالیت های ورزشی افراد را ثبت کرده و قبل از ثبت فعالیت، وضعیت آب و هوا را بررسی کند. اگر شرایط هوا نامساعد باشد (مثلاً بارانی)، برنامه باید هشدار مناسب نمایش دهد و آن را در لاگ ثبت کند.

## مدیریت ورزشکاران و فعالیت‌ها

- افزودن ورزشکار با اطلاعات پایه:
  - نام
  - سن
  - شهر (برای بررسی وضعیت هوا الزامی است)
- ثبت فعالیت روزانه شامل:
  - نوع فعالیت (مثلاً دویدن، دوچرخه‌سواری، پیاده‌روی و ...)
  - مسافت یا مدت زمان فعالیت
  - بررسی وضعیت هوا برای تاریخ مشخص با استفاده از `datetime.strptime`
- نمایش فعالیت‌های یک ورزشکار و همچنین مجموع کل فعالیت‌های او (یعنی ممکن است چندین فعالیت انجام داده باشد)

## هشدار وضعیت هوا

- برنامه باید با استفاده از API سایت **OpenWeatherMap** وضعیت هوا را برای شهر ورزشکار دریافت کند.  
لینک مستندات:
- <https://openweathermap.org/current>
- اگر ورزشکار قصد دویدن (**running**) دارد و وضعیت هوا بارانی (**rain**) است:
  - پیام هشدار باید چاپ شود و هشدار باید در فایل لاگ ذخیره شود.
- در صورت بروز خطاهای API مثل مشکل شبکه، API Key اشتباه و ...، پیام مناسب با سطح **ERROR** ثبت شود.

سطح های لاگ:

INFO برای عملیات موفق

WARNING برای شرایط غیر عادی مثل فعالیت در هوای نامساعد

ERROR برای خطاهای API یا ورودی نامعتبر

- داشتن یک فایل جداگانه برای لاگ ها به نام `app.log`

ساختار کد باید قابل فهم، تمیز و قابل اجرا باشد و خروجی مد نظر باید در نهایت این قابلیت ها را داشته باشد:

- امکان افزودن ورزشکار
- امکان ثبت فعالیت برای ورزشکار
- بررسی وضعیت آبوهوا قبل از ثبت
- ثبت هشدار یا خطا در لاگ
- امکان نمایش لیست فعالیت ها و مجموع آن ها

3.

برای هر تابع حداقل ۲ تست بنویسید.

تست اول: حالت عادی باشد (Normal Case)

تست دوم: (Edge Case) باشد

فیلتر کاربران بر اساس سن

یک لیست از دیکشنری کاربران را گرفته و کاربرانی که سنشان

در بازه min\_age تا max\_age است برمی گرداند.

مثال: `users = [ {"name": "Ali", "age": 20}, ... ]`

```
def filter_users_by_age(users, min_age, max_age):
```

```
    if min_age > max_age:
```

```
        raise ValueError("min_age cannot be greater than max_age")
```

```
    return [u for u in users if min_age <= u.get("age", -1) <= max_age]
```

محاسبه معدل با دو رقم اعشار

لیستی از نمرات را گرفته و معدل را با دو رقم اعشار برمی گرداند.

نمره ها باید بین 0 و 20 باشند.

```
def calculate_gpa(scores):
    if not scores:
        return 0.0
    for s in scores:
        if not (0 <= s <= 20):
            raise ValueError("Invalid score value")

    avg = sum(scores) / len(scores)
    return round(avg, 2)
```

بررسی پسورد قوی:

```
def is_strong_password(password):
    if len(password) < 8:
        return False

    has_upper = any(c.isupper() for c in password)
    has_lower = any(c.islower() for c in password)
    has_digit = any(c.isdigit() for c in password)

    return has_upper and has_lower and has_digit
```

محاسبه مجموع قیمت سبد خرید

مثال: `cart = [ {"name": "milk", "price": 10, "qty": 2}, ... ]`

مجموع کل قیمت را همراه با مالیات محاسبه می‌کند.

```
def calculate_cart_total(cart, tax=0.0):
    if any(item["price"] < 0 or item["qty"] < 0 for item in cart):
        raise ValueError("Price and quantity must be non-negative")

    subtotal = sum(item["price"] * item["qty"] for item in cart)
    total = subtotal + (subtotal * tax)
    return round(total, 2)
```

### تبدیل رشته تاریخ به datetime

```
from datetime import datetime

def parse_date(date_str):
    return datetime.strptime(date_str, "%Y-%m-%d")
```

- حالت Edge Case : اگر تاریخ با فرمت اشتباه بدهید باید ValueError بدهد

### شمارش تعداد تکرار یک کلمه در متن

```
def count_word(text, word):
    if not isinstance(text, str) or not isinstance(word, str):
        raise TypeError("Inputs must be strings")

    words = text.lower().split()
    return words.count(word.lower())
```