**템플릿의 역할** – HTML, 보여주기 위한 것을 저장

**모델의 역할** – 디비와 연결다리

**뷰의 역할 안나옴 – 애매모호하기 때문에**

**url연결 태그**

a 태그 = form 태그의 method=”GET”

form 태그의 method=”POST”

**render에서 값을 HTML로 넘기는 방법**

render(request, “~~.html”, { “ “: ( ), ... }) 어떻게 파라미터가 이루어지는가

**HTML에서 POST로 온 것을 View 받는 방법 – GET 안나옴 – 이름 헷갈리게 안나옴**

**render vs redirect**

a 태그와 같은 역할 = redirect – 확장자가 안붙는다.

**model 관련**

hit나 title 등의 어트리뷰트에 어떤 타입을 쓰는 것이 좋을까.

- 내용 ( textfield ), 조회수 ( integerfield ), 타이틀 ( Charfield )

- textfield = 길이 제한 X / charfield = 길이제한 설정해주어야

**SQL**

DDL – 데이터 생성어 **( 테이블 생성 )**

create 생성

alter 변경 타입포함

drop 삭제

rename 이름 변경

truncate 비우기

DML – 데이터 조작어 **( 테이블’s 데이터 조작 )**

insert ex) insert into MEMBER( ID, NAME, AGE) VALUES(“a”, “b”, 34)

update ex) UPDATE MEMBER SET NAME=’b1’, age=66 WHERE ID=’a’

delete ex) delete FROM MEMBER WHERE ID=”a”

select ex) select \* FROM MEMBER WHERE ID LIKE ‘a%’ ORDER BY NO (DESC | ASC)

- ‘%’ || ‘a’ || ‘%’ => a가 포함된 (내림차순 | 오름차순)

- ‘a’ || ‘%’ => a로 시작하는

- ‘%’ || ‘a’ => a가 끝나는

**↓ 시험에는 안나오나 기억은 하자 ↓**

DCL – 데이터 제어어

grant 권한부여

revoke 권한삭제

TCL – 트랜젝션 제어어

commit 적용

rollback 되돌리기

savepoint 세이브 포인트

**bulk\_create & bulk\_update**

**dateframe – df.to\_html(classes=true)**

        # 1. QuerySet >> list 변경

        # SELECT NO, NAME, KOR FORM MEMBER\_TABLE2

        rows = list(Table2.objects.all().values("no","name", "kor"))

        print(rows) # >> [{ }, { }, ...  ] dict의 list

        # 2. list >> dataframe 변경

        df = pd.DataFrame(rows)

        # 3. dataframe >> list 변경

        rows1 = df.values.tolist()

        print(rows) # >> [( ), ( ), ...  ]list의 list

        return render(request, "member/dataframe.html", {"df\_table": df.to\_html, "list":rows})

<body>

    {% for row in list %}

      {{row.no}},

      {{row.name}},

      {{row.kor}}

      <br />

    {% endfor %}

    <hr/>

    <h2>데이터프레임 테이블</h2>

    {% autoescape off %}

    {{df\_table}}

    {% endautoescape %}

</body>