



210811 BXM 실습 및 소프트웨어 아키텍처

🕒 생성일	@2021년 8월 11일 오후 1:09
▼ 성함	나한주
🔗 속성	
▼ 수업 유형	이론 및 실습
🕒 수정일	@2021년 8월 11일 오후 9:51
👤 작성자	현동빈

오전 : BXM 실습

오후 : 소프트웨어 아키텍처

오전 : BXM 실습

- 서비스 계층 코드 작성

```
20 @BxmService("DpstStaffInfoSrchSvc")|
21 @BxmCategory(logicalName = "수신 스테프 정보 조회 서비스", description = "수신 스테프 정보 조회용 서비스")
22 public class DpstStaffInfoSrchSvc {
23     private final Logger logger = LoggerFactory.getLogger(this.getClass());
24
25
26
27     //Biz.getStaff()호출
28
29
30
31
32
```

DpstStaffInfoSrchBiz.java

```
22 * @since 2021.08.09
23 * @version 1.0.0
24 * @author NB-21070707
25 */
26 @BxmBean
27 @BxmCategory(logicalName = "수신 스테프 정보 조회 비즈니스", description = "수신 스테프 정보 조회용 비즈니스")
28
```

- Service 클래스로 작성 ⇒ @BxmService (하위 : Bean 클래스 작성 ⇒ @BxmBean)

```

    } @BxmService("DpstStaffInfoSrchSvc")
    } @BxmCategory(logicalName = "수신스타
    ) public class DpstStaffInfoSrchSvc

```

- @BxmService("DpstStaffInfoSrcSvc") & public class DpstStaffInfoSrchSsvc

```

4 @BxmCategory(logicalName = "수신스태프정보조회 서비스", description = "수신스태프정보조회 서비스")
0 public class DpstStaffInfoSrchSvc {
1     private final Logger logger = LoggerFactory.getLogger(this.getClass());

    @BxmServiceOperation("getStaff")
    @TransactionalOperation
    @CbbSvcInfo(srvCd = "STAFFSVC11", srvNm = "Get Staff Info", srvcAbvtnNm = "getStaff")
    @BxmCategory(description = "")
    public DpstStaffInfoSrchSvcOutDto getStaff(DpstStaffInfoSrchSvcInDto inDto) throws BizApplicationException {

        //Biz.getStaff()호출
        return null;
    }
}

```

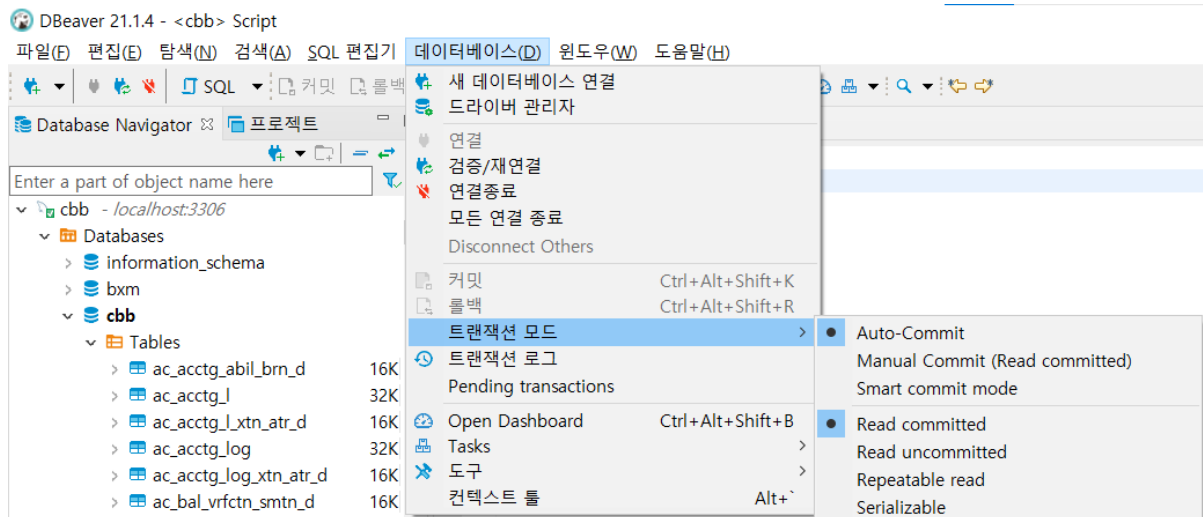
- @BxmServiceOperation("getStaff") & public DpstStaffInfoSrchSvcOutDto getStaff~

```

    @BxmServiceOperation("getStaff")
    @TransactionalOperation
    @CbbSvcInfo(srvCd = "STAFFSVC11", srvNm = "Get Staff Info", srvcAbvtnNm = "getStaff")
    @BxmCategory(description = "")
    public DpstStaffInfoSrchSvcOutDto getStaff(DpstStaffInfoSrchSvcInDto inDto) throws BizApplicationException {

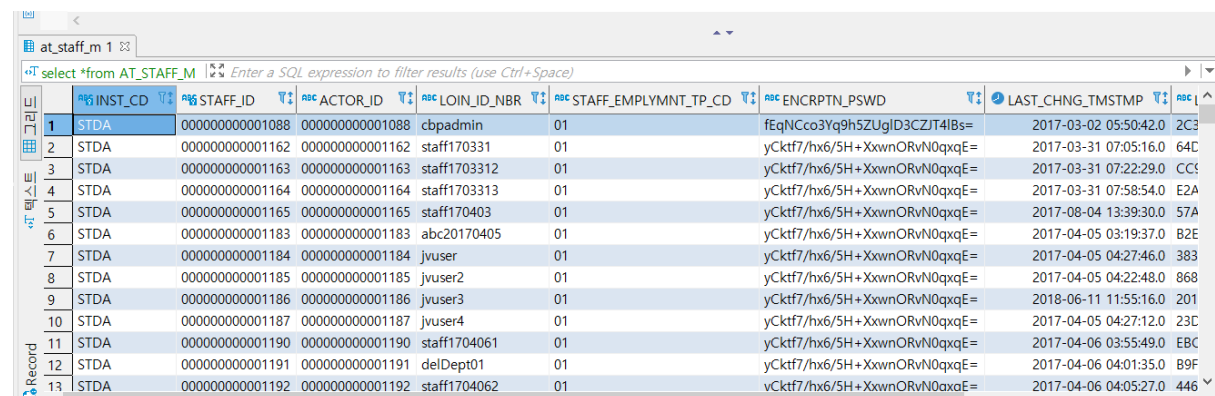
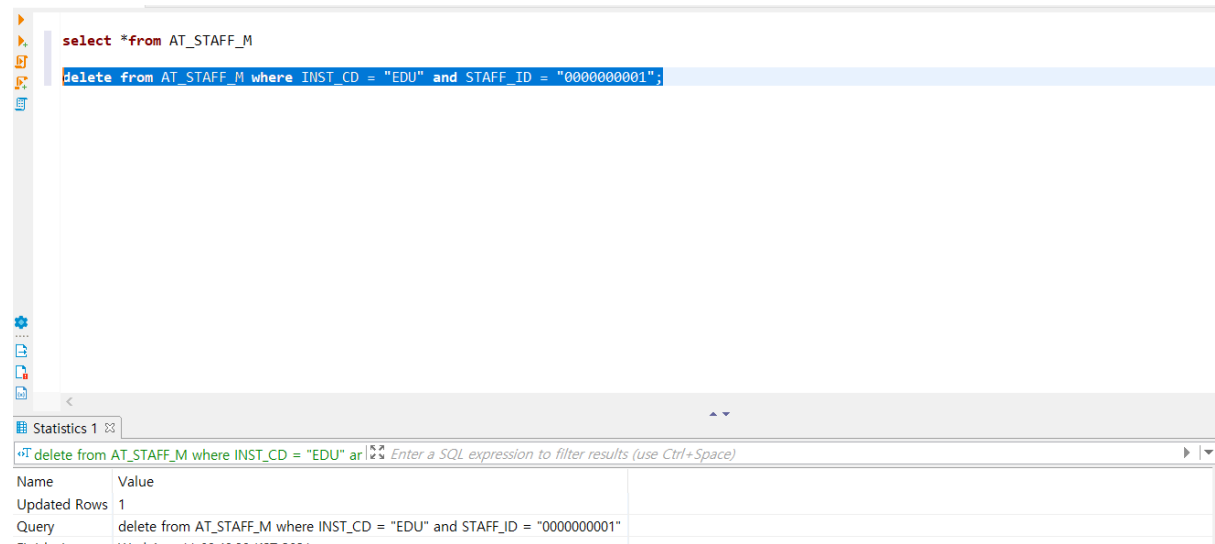
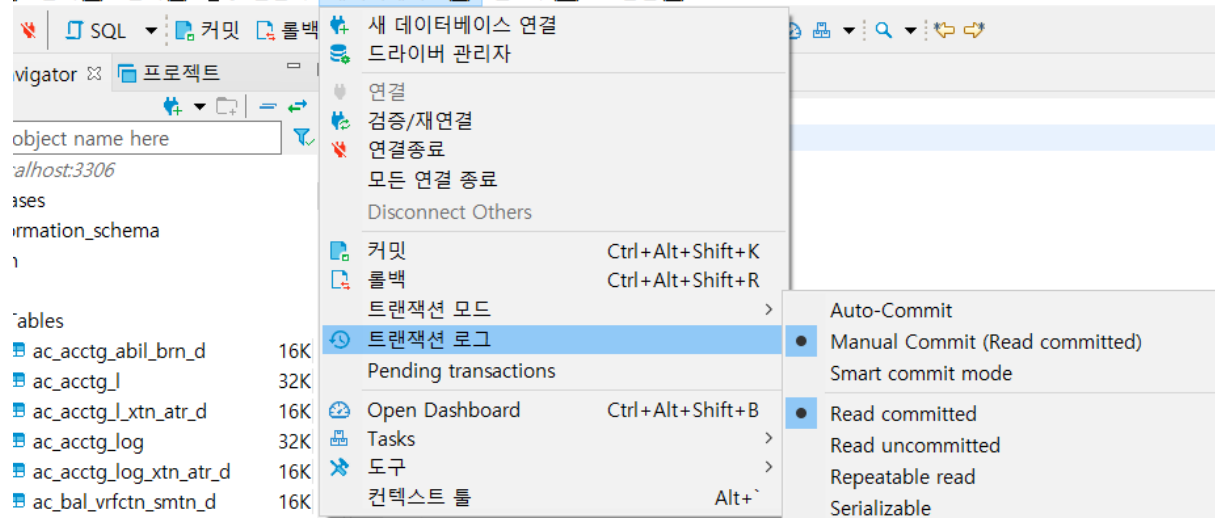
```

- @TransactionalOperation
- CbbSvcInfo(SrvCd = "STAFFSVC11"~)



1.4 - <cbb> Script

) 탐색(N) 검색(A) SQL 편집기 데이터베이스(D) 윈도우(W) 도움말(H)



*<cbb> Script

```

select *from AT_STAFF_M;

delete from AT_STAFF_M where INST_CD = "EDU" and STAFF_ID = "0000000001";

rollback;

```

at_staff_m 1

select *from AT_STAFF_M

	INST_CD	STAFF_ID	ACTOR_ID	LOIN_ID_NBR	STAFF_EMPLOYMNT_TP_CD	ENCRPTN_PSWD	LAST_CHNG_TMSTMP	
1	EDU	0000000001	0000000001	JungTeller	01	fEqNCco3Yq9h5ZUgID3CZJT4Bs=	2018-10-11 07:50:20.0	201
2	STDA	00000000001088	00000000001088	cbpadmin	01	fEqNCco3Yq9h5ZUgID3CZJT4Bs=	2017-03-02 05:50:42.0	2C3
3	STDA	000000000001162	000000000001162	staff170331	01	yCktf7/hx6/5H+XxwnORvN0qxqE=	2017-03-31 07:05:16.0	64C
4	STDA	000000000001163	000000000001163	staff1703312	01	yCktf7/hx6/5H+XxwnORvN0qxqE=	2017-03-31 07:22:29.0	CC3
5	STDA	000000000001164	000000000001164	staff1703313	01	yCktf7/hx6/5H+XxwnORvN0qxqE=	2017-03-31 07:58:54.0	F2A

```

select *from AT_STAFF_M;

delete from AT_STAFF_M where INST_CD = "EDU" and STAFF_ID = "0000000001";

rollback;

```

SQL의 트랜잭션 처리 매커니즘

```
begin Tran ~ commit or rollback
```

• Transaction 과정(mysql)

```

31 /
32 @BxmService("DpstStaffInfoSrhcSvc")
33 @BxmCategory(logicalName = "수신스테프정보조회 서비스", description = "수신스테프정보조회용 서비스")
34 public class DpstStaffInfoSrhcSvc {
35
36
37 @BxmServiceOperation("getStaff")
38 @TransactionalOperation
39 @CbbSrcInfo(srvCd = "STAFFSVC01", srvNm = "Get Staff Info", srvAbrvtnNm = "getStaff")
40 @BxmCategory(description = "Get Staff Info Service")
41 public DpstStaffInfoSrhcSvcOutDto getStaff(DpstStaffInfoSrhcSvcInDto inDto) throws BizApplicationException {
42 //CBP = 베이스컴포넌트(CBB) + 서비스컴포넌트
43 //srvCd 광장히 중요(별 다섯개)
44
45 //Biz.getStaff()호출
46 DpstStaffInfoSrhcBizInDto BizInDto = new DpstStaffInfoSrhcBizInDto();
47 BizInDto.setStaffId(inDto.getStaffId());
48
49 DpstStaffInfoSrhcBizOutDto BizOutDto = _dpstStaffInfoSrhcDso().getStaff(BizInDto);
50
51 return _setOutDto(BizOutDto);
52 }

```

```

54 private DpstStaffInfoSrchrBiz _dpstStaffInfoSrchrDso() {
55
56     return CbbApplicationContext.getBean(DpstStaffInfoSrchrBiz.class);
57 }
58
59
60 private DpstStaffInfoSrchrSvcOutDto _setOutDto(DpstStaffInfoSrchrBizOutDto BizOutDto) {
61
62     DpstStaffInfoSrchrSvcOutDto outDto = new DpstStaffInfoSrchrSvcOutDto();
63
64     outDto.setInstCd(BizOutDto.getInstCd());
65     outDto.setStaffId(BizOutDto.getStaffId());
66     outDto.setActorId(BizOutDto.getActorId());
67     outDto.setLoinIdNbr(BizOutDto.getLoinIdNbr());
68     outDto.setStaffEmplymntTpCd(BizOutDto.getStaffEmplymntTpCd());
69     outDto.setEnchrptnPswd(BizOutDto.getEnchrptnPswd());
70     outDto.setLastChngTmstmp(BizOutDto.getLastChngTmstmp());
71     outDto.setLastChngGuid(BizOutDto.getLastChngGuid());
72
73     return outDto;
74 }
75

```

- 직원 호출 서비스 클래스 메서드 작성

오후 : 소프트웨어 아키텍처

▼ 컴퓨터 시스템

- 거래처리시스템(TPS)
- 정보시스템
- 경영정보시스템(MIS)

▼ 시스템 통합 프로젝트(SI)

- 시스템 구축을 전문으로 하는 기업에게 위탁하여 원하는 시스템을 만드는 프로젝트로 다양한 역할의 사람들이 참여하며, 이들은 다양한 H/W 및 S/W들을 통합하여 시스템을 구축

▼ 아키텍처?

- 시스템을 이해하기 위한 다양한 관점에서 각 컴포넌트가 무엇이며 어떻게 상호작용하는지, 정보가 어떻게 교환이 이루어지는가를 설명
- 비즈니스 아키텍처
- 시스템/소프트웨어 아키텍처
- 네트워크 아키텍처
- 애플리케이션 아키텍처
- 서비스 지향 아키텍처

- 마이크로 서비스 아키텍처

▼ 아키텍트

- 요구사항, 업무 도메인, 기술, 개발과정 등을 이해하고 사용자가 원하는 시스템을 만들기 위해 개발자부터 경영진까지 다수의 이해관계자들과 소통하면 시스템을 구축하도록 가이드하는 사람
- 시스템/소프트웨어 아키텍트
- 비즈니스 아키텍트
- 데이터 아키텍트
- 애플리케이션 아키텍트
- 솔루션 아키텍트
- 기타 다양한 아키텍트...
- 아키텍트는 이해관계자들과 지속적으로 소통하며 원하는 시스템이 만들어지도록 조율해야 함

▼ 아키텍트처럼 생각하기

- Usecase Diagram : 시스템과 사용자의 상호작용을 다이어그램으로 표현한 것으로 사용자의 관점에서 시스템의 서비스 혹은 기능 및 그와 관련한 외부요소를 보여주는 것
- Usecase Scenario
- 분석 클래스 다이어그램
- DDD : 비즈니스 도메인을 기준으로 애플리케이션을 나누어 설계 및 개발하는 방법

▼ 아키텍처 결정하기

- 아키텍처 결정 사항 : 아키텍처 상 중요한 요구 사항을 해결하는 디자인 결정