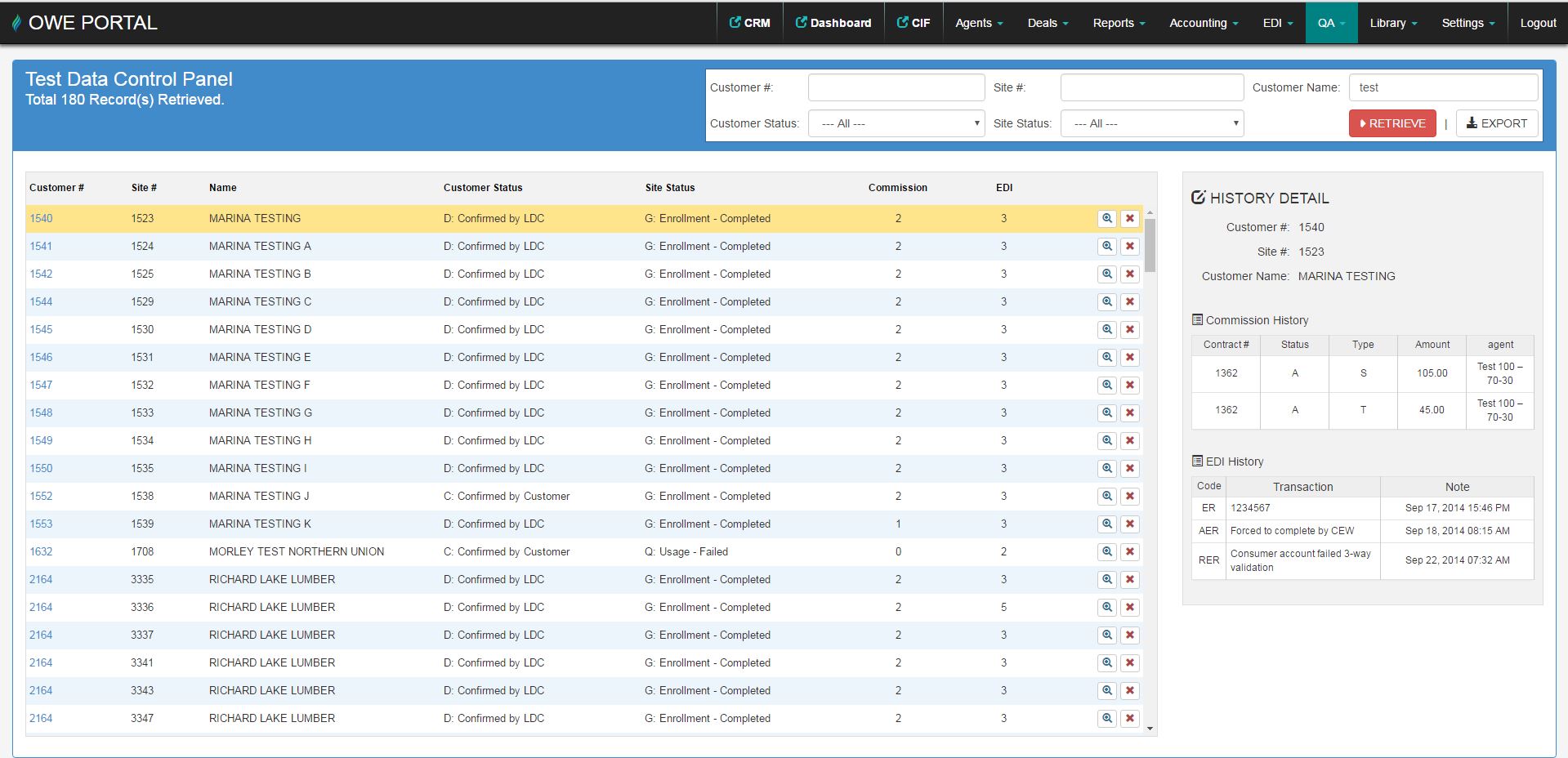
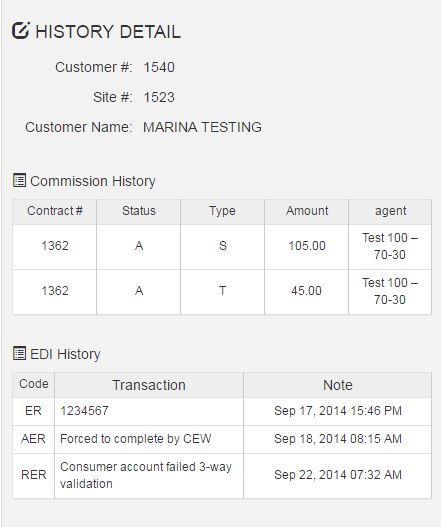
# Test Data Control Panel

Jun 14, 2016

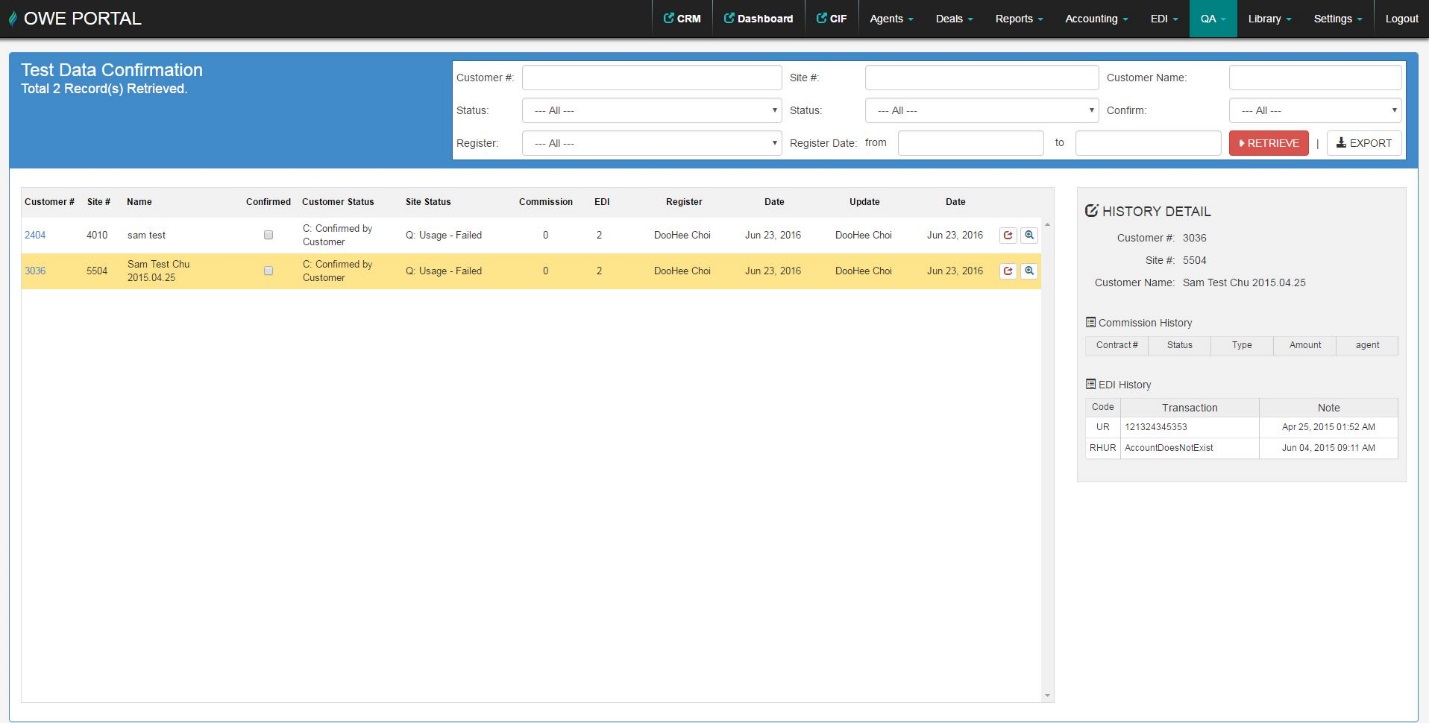
\*Test Data Control Panel



\*History Detail Page



\*Test Data Confirmation



**Database: CEW**

**Tables:**

**enroll\_customer\_site**: 등록되어 있는 site의 enroll\_site\_id, enroll\_customer\_id, enroll\_site\_status 상태, site에 관한 자세한 정보

**enroll\_site\_status:** site status에 관한 코드와 그 코드에 대한 설명

**enroll\_customer**: 등록되어 있는 enroll\_customer\_id, enroll\_customer\_status 상태, customer에 대한 자세한 정보

**enroll\_customer\_status**: customer status에 관한 코드와 그 코드에 대한 설명

**enroll\_customer\_site\_test**(is created in order to store data into Test Data Confirmation): Test Data Confirmation 페이지의 정보를 담는 Talbe.

**edi\_site\_download**: 중간 업체인 ‘Alexian’의 EDI 프로토콜로인해 두 테이블인 download와 upload로 나눈다. Site 별 EDI tracsaction log 정보를 담고 있다.

**edi\_site\_upload**: download 테이블 보다 먼저 정보가 발생한다. Transaction 코드 상태와 supplier\_id, account\_number 정보를 담고 있다.

**enroll\_customer\_site\_contract:** enroll\_contract\_id, contract\_status 등의 정보를 담고 있다.

**enroll\_customer\_site\_contract\_commission**: S(agent), T(telemarketer) 별 commission\_status와 commission\_amount 정보를 담고 있다.

**[login]**: login\_id와 id 별 상세 정보를 담고있다(password, phone number 등)

**Stored Procedures:**

**rpt\_enroll\_customer\_site\_test2**: Test Data Conrtol Panel page를 위해서 관련 테이블로 부터 정보를 display 해준다. 파라미터 별 검색과 validations.

**get\_enroll\_customer\_site\_test\_list2**: Test Data Control Panel page에서 user가 Test용 customer 나 site라고 판단 될 경우 C:\Users\DooHee.Choi\Desktop\3.JPG 버튼을 누르면 해당 row가 제거 된다. (Test Data Confirmation page 입장에서는 row가 추가 된다.) 바로 이 제거(추가) 된 data 들의 정보를 볼 수 있게 해주는 SP. Data들은 실제 enroll\_customer\_site\_test 테이블에 저장된다.

**get\_enroll\_customer**: enroll\_customer에 있는 data들을 보여주는 SP. Test Data Control Panel 에서는 C:\Users\DooHee.Choi\Desktop\1.JPG 버튼을 눌러, History Detail 페이지의 Customer #, Site #, Customer Name 을 display 하기 위한 SP.

**get\_enroll\_customer\_site\_contract\_commission\_list:** C:\Users\DooHee.Choi\Desktop\1.JPG **누른후,** History Detail 에서 commission History 부분(상세 페이지) display를 위한 SP.

**get\_edi\_site\_history3**: C:\Users\DooHee.Choi\Desktop\1.JPG 누른후, EDI History 부분(상세 페이지) display를 위한 SP.

**add\_enroll\_customer\_site\_test**: Test Data Control Panel에서 C:\Users\DooHee.Choi\Desktop\3.JPG 버튼을 눌러 해당 row 삭제시 ->@enroll\_site\_id, @register\_id 정보를 가져와서 Test Data Confirmation 에 (**enroll\_customer\_site\_test** 테이블에) insert 시킨다. add\_enroll\_customer\_site\_test SP, insert 문에서 모든 컬럼 value를 insert 시키지 않는 이유는 이 데이터들이 Snapshot으로 이용되지 않고 실시간으로 값이 update 되는 data로 이용되기 때문이다.

**del\_enroll\_customer\_site\_test**: Test Data Confirmation 에서 선택 된 row 즉, test 목적으로 생선된 customer 가 아니라고 판단되는 경우 취소를 하려고 할때, C:\Users\DooHee.Choi\Desktop\4.JPG 버튼을 눌러 취소 할 수 있는 SP. Enroll\_customer\_site\_test 테이블에서 삭제 된다.

**set\_enroll\_customer\_site\_test\_status\_toActive:** Test Data Confirmation 페이지 에서 Confirmed 컬럼 값을 체크표시 즉, Test 목적의 custormer 라 확신이 되는 경우 표시를 원할때, 사용되는 SP 이다. 체크시, 체크한 user의 ID 와 체크 당시의 날짜 및 시간이 표시된다.

**set\_enroll\_customer\_site\_test\_status\_toInactive**: Test Data Confirmation 페이지 에서 Confirmed 컬럼 값을 체크 해제시 사용되는 SP 이다. 시, 체크한 user의 ID 와 체크 당시의 날짜 및 시간이 표시된다. 체크 해제시, 체크 해제 한 user의 ID와 날짜가 표시된다.

Functions:

**fget\_edi\_site\_download\_count**: site의 EDI log(transaction의 유무와 양)를 카운트 하기 위한 function. edi\_site\_download 테이블로부터 정보를 가져온다.

**fget\_edi\_site\_upload\_count**: site 의 EDI log(transaction의 유무와 양)를 카운트 하기 위한 function. edi\_site\_upload 테이블로부터 정보를 가져온다.

**fget\_enroll\_customer\_site\_contract\_commission\_count**: commission의 기록을 count 하는 function.

\*Display information from

enroll\_customer\_site

enroll\_site\_status

enroll\_customer

enroll\_customer\_status

rpt\_enroll\_customer\_site\_test2

USE [CEW]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[rpt\_enroll\_customer\_site\_test2] Script Date: 06/23/2016 08:15:40 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: CHOI

-- Create date: 2016.06.17

-- Update date: 2016.06.20

-- Description:

--

-- Update: SKC

-- Update date: 2016.06.22

-- Description:

-- =============================================

ALTER PROCEDURE [dbo].[rpt\_enroll\_customer\_site\_test2]

@search\_string varchar(50) = null,

@enroll\_site\_id int = null,

@enroll\_customer\_id int = null,

@enroll\_customer\_status char(1) = null,

@enroll\_site\_status char(1) = null

AS

BEGIN

SET NOCOUNT ON;

if @search\_string is null and

@enroll\_site\_id is null and

@enroll\_customer\_id is null and

@enroll\_customer\_status is null and

@enroll\_site\_status is null

begin

raiserror('Invalid request!', 16, 1)

return -1

end

if @search\_string is not null

begin

if len(@search\_string ) < 3

begin

raiserror('Please enter at least 3 letters to complete this request!', 16, 1)

return -2

end

set @search\_string = '%' + @search\_string + '%'

end

select site\_name = a.name2,

a.enroll\_site\_id,

a.enroll\_site\_status,

enroll\_site\_status\_desc = a1.user\_friendly\_desc,

customer\_name = b.name2,

b.enroll\_customer\_id,

b.enroll\_customer\_status,

enroll\_customer\_status\_desc = b1.user\_friendly\_desc,

count\_edi\_log = dbo.fget\_edi\_site\_upload\_count(a.enroll\_site\_id) + dbo.fget\_edi\_site\_download\_count(a.enroll\_site\_id),

count\_commission\_log = dbo.fget\_enroll\_customer\_site\_contract\_commission\_count(a.enroll\_site\_id)

from enroll\_customer\_site a

inner join enroll\_site\_status a1 on a.enroll\_site\_status = a1.status\_code

inner join enroll\_customer b on a.enroll\_customer\_id = b.enroll\_customer\_id

inner join enroll\_customer\_status b1 on b.enroll\_customer\_status = b1.status\_code

where a.enroll\_site\_status <> 'C' and

(a.enroll\_site\_id not in (select enroll\_site\_id from enroll\_customer\_site\_test)) and

(@enroll\_customer\_id is null or a.enroll\_customer\_id = @enroll\_customer\_id) and

(@enroll\_site\_id is null or a.enroll\_site\_id = @enroll\_site\_id) and

(@enroll\_site\_status is null or a.enroll\_site\_status = @enroll\_site\_status) and

(@enroll\_customer\_status is null or b.enroll\_customer\_status = @enroll\_customer\_status) and

(

@search\_string is null or

a.name2 like @search\_string or

b.name2 like @search\_string

)

END

1. Customer #, Site #, Name, Customer Status, Site Status, Commission, EDI 정보 Display 를 위한 SP 작성. rpt\_enroll\_customer\_site\_test2

select site\_name = a.name2,

a.enroll\_site\_id,

a.enroll\_site\_status,

enroll\_site\_status\_desc = a1.user\_friendly\_desc,

customer\_name = b.name2,

b.enroll\_customer\_id,

b.enroll\_customer\_status,

enroll\_customer\_status\_desc = b1.user\_friendly\_desc

from enroll\_customer\_site a

inner join enroll\_site\_status a1 on a.enroll\_site\_status = a1.status\_code

inner join enroll\_customer b on a.enroll\_customer\_id = b.enroll\_customer\_id

inner join enroll\_customer\_status b1 on b.enroll\_customer\_status = b1.status\_code

Customer #: enroll\_customer Table(alias b)의 enroll\_customer\_id

Site #: enroll\_customer\_site Table(alias a)의 enroll\_site\_id

Name: enroll\_customer Table(alias b)의 name2

Customer Status: enroll\_customer\_status(alias b1)의 user\_friendly\_desc

Site Status: enroll\_site\_status(alias a1)의 user\_friendly\_desc

1. Customer #(ID), Site #(ID), Customer Status, Site Status, Customer Name 으로 검색, 단 5개 필드 중 한가지 값은 반드시 들어가도록 한다. 아무 값도 없는 상태에서 Retrieve를 했을때 모든 정보가 조회 되지 않도록.

if @search\_string is null and

@enroll\_site\_id is null and

@enroll\_customer\_id is null and

@enroll\_customer\_status is null and

@enroll\_site\_status is null

begin

raiserror('Invalid request!', 16, 1)

return -1

end

검색에 사용되는 파리미터 모두 null 조건을 걸어줘서, 만약 모두 null 값일때, riserror로 에러메세지가 뜨게 한다.

1. 검색 필드 Custoemr Name 에서 최소한 3글자 이상 입력해야 한다는 validation을 건다.

if @search\_string is not null

begin

if len(@search\_string ) < 3

begin

raiserror('Please enter at least 3 letters to complete this request!', 16, 1)

return -2

end

set @search\_string = '%' + @search\_string + '%'

end

검색 문장에 검색하고 싶은 단어나 문장이 포함되면 다 출력 되도록 set 을 한다.

1. site\_status 상태가 deleted 인 즉, 코드가 ‘C’ 인 site들은 제외 하고 display 한다. 왜냐하면 추후에 delete 작업이 있기때문이다.

where a.enroll\_site\_status <> 'C'

Test Data Confirmation 페이지에서 취소된 row는 다시 Test Data control Panel 에서 display 되야 하기 때문에 enroll\_customer\_site\_test에 없는 enroll\_site\_id, 즉 row만 보여주는 조건 not in을 건다.

(a.enroll\_site\_id not in (select enroll\_site\_id from enroll\_customer\_site\_test))

enroll\_customer\_site 테이블이나 enroll\_customer 테이블의 name2 컬럼 값에서 검색하는 @serarch\_string 과 같은 (like) 값을 display 하는 조건을 건다.

@search\_string is null or

a.name2 like @search\_string or

b.name2 like @search\_string

1. 파라미터를 통해 받는 값과 각 참조하고 있는 테이블의 컬럼값의 매칭

@enroll\_customer\_id is null or a.enroll\_customer\_id = @enroll\_customer\_id) and

@enroll\_site\_id is null or a.enroll\_site\_id = @enroll\_site\_id) and

@enroll\_site\_status is null or a.enroll\_site\_status = @enroll\_site\_status) and

@enroll\_customer\_status is null or b.enroll\_customer\_status = @enroll\_customer\_status)

1. EDI log count를 위한 function.

count\_edi\_log =

dbo.fget\_edi\_site\_upload\_count(a.enroll\_site\_id) + dbo.fget\_edi\_site\_download\_count(a.enroll\_site\_id),

fget\_edi\_site\_download\_count 와 fget\_edi\_site\_upload\_count 함수를 생성한다.

**fget\_edi\_site\_download\_count:**

ALTER FUNCTION [dbo].[fget\_edi\_site\_download\_count]

(

@site\_id int

)

RETURNS int

AS

BEGIN

DECLARE @ResultVar int = 0

select @ResultVar = count(transaction\_id)

from edi\_site\_download

where site\_id = @site\_id

RETURN @ResultVar

END

**fget\_edi\_site\_upload\_count:**

ALTER FUNCTION [dbo].[fget\_edi\_site\_upload\_count]

(

@site\_id int

)

RETURNS int

AS

BEGIN

DECLARE @ResultVar int = 0

select @ResultVar = count(transaction\_id)

from edi\_site\_upload

where site\_id = @site\_id

RETURN @ResultVar

END

1. Commission count를 위한 function.

count\_commission\_log = dbo.fget\_enroll\_customer\_site\_contract\_commission\_count(a.enroll\_site\_id)

fget\_enroll\_customer\_site\_contract\_commission\_count 함수를 생성한다.

ALTER FUNCTION [dbo].[fget\_enroll\_customer\_site\_contract\_commission\_count]

(

@enroll\_site\_id int

)

RETURNS int

AS

BEGIN

DECLARE @ResultVar int = 0

select @ResultVar = count(a.enroll\_contract\_id)

from enroll\_customer\_site\_contract a,

enroll\_customer\_site\_contract\_commission b

where a.enroll\_contract\_id = b.enroll\_contract\_id and

a.enroll\_site\_id = @enroll\_site\_id

RETURN @ResultVar

END

@ResultVar 변수를 선언하고 초기값 0을 준다. enroll\_contrac\_id 를 카운트 하고 commission amount 정보, commission type 정보를 얻기 위해, enroll\_customer\_site\_contract\_commission 테이블과 Join을 건다. 카운트 된 값은 @ResultVar 변수에 return 된다.

1. Test Data Control Panel 페이지에서 선택된 row의 C:\Users\DooHee.Choi\Desktop\7.JPG 버튼을 눌렀을때 Commission 과 EDI log history 상세정보 및 선택된 Customer #, Site #, Customer Name의 정보를 보여준다.

**get\_enroll\_customer:**

ALTER PROCEDURE [dbo].[get\_enroll\_customer]

@enroll\_customer\_id int

AS

BEGIN

SET NOCOUNT ON;

select \*

from enroll\_customer

where enroll\_customer\_id = @enroll\_customer\_id

END

enroll\_customer 테이블에서enroll\_customer\_id 별로 고객 정보를 보여준다. History Detail 페이지에서는 Customer ID, Site ID, Customer Name 이 사용된다.

**get\_enroll\_customer\_site\_contract\_commission\_list:**

ALTER PROCEDURE [dbo].[get\_enroll\_customer\_site\_contract\_commission\_list]

@enroll\_site\_id int

AS

BEGIN

SET NOCOUNT ON;

select a.enroll\_contract\_id,

b.commission\_status,

b.commission\_type,

b.commission\_amount,

c.full\_name

from enroll\_customer\_site\_contract a

inner join enroll\_customer\_site\_contract\_commission b on a.enroll\_contract\_id = b.enroll\_contract\_id

left outer join [login] c on b.login\_id = c.login\_id

where a.enroll\_site\_id = @enroll\_site\_id

END

enroll\_customer\_site\_contract 테이블을 기준으로 commission\_status,commission\_type,commission\_amount 값을 가져오기 위해 enroll\_customer\_site\_contract\_commission 테이블에 Join 을 건다.

enroll\_customer\_site\_contract 테이블의 해당 컬럼 값들을 다 조회 하면서 login 한 사람의 full name 을 가져오기 위해 login 테이블에서 full\_name 컬럼 값을 가져온다.

**get\_edi\_site\_history3:**

ALTER PROCEDURE [dbo].[get\_edi\_site\_history3]

@enroll\_site\_id int

AS

BEGIN

SET NOCOUNT ON;

select a.status\_code,

a.status\_text,

a.usage\_begin,

a.usage\_end,

a.actual\_usage,

a.contract\_date,

b.start\_date,

a.cancel\_date,

a.price\_date,

a.register\_date,

a.register\_time,

register\_datetime = cast(a.register\_date as datetime) + cast(a.register\_time as datetime)

into #get\_edi\_site\_history

from edi\_site\_download a

left outer join enroll\_customer\_site\_contract b on a.contract\_id = b.enroll\_contract\_id

where a.site\_id = @enroll\_site\_id

insert into #get\_edi\_site\_history

select status\_code = a.transaction\_code,

status\_text = a.account\_number,

usage\_begin = null,

usage\_end = null,

actual\_usage = null,

contract\_date = null,

--b.start\_date,

start\_date = null,

cancel\_date = null,

price\_date = null,

a.register\_date,

a.register\_time,

register\_datetime = cast(a.register\_date as datetime) + cast(a.register\_time as datetime)

from edi\_site\_upload a

left outer join enroll\_customer\_site\_contract b on a.contract\_id = b.enroll\_contract\_id

where a.site\_id = @enroll\_site\_id

select \*

from #get\_edi\_site\_history

--order by register\_date, register\_time

END

\*Temporary Table (Staging Table):

1. 여러 단계를 거쳐 동일한 데이터에 반복작업을 해결할 때

2. 복잡한 쿼리를 단순화 시킬때

3. 서버상의 Cursor 사용으로 인한 부하를 줄이고자 할 때

4. 로컬 임시 테이블은 다음과 같은 경우에 자동적으로 제거된다.

해당 세션의 연결이 끊어진 경우(LogOff)

저장 프로시저나 트리거에서 만들어졌을 경우

이 경우 해당 프로시저의 종료와 함께 임시 테이블이 제거된다.

5. 인덱스를 작성할 수 있다.

FK(외래키)를 제외한 나머지 제약을 지정할 수 있다.

테이블은 작성이 되지만 경고 메시지와 함께 FK 선언은 제외된다.

ALTER TABLE이 가능하다.

INSERT INTO, BULK INSERT 문과 함께 사용할 수 있다.