

#부록 워크샵가이드

제2장 WORKSHOP 엔티타입(ENTITY TYPE)의 선정 - 가이드

ITSoft(주)학원의 시나리오에서 엔티타입을 선정하는 예는 다음과 같다.

모델링과 데이터베이스의 설계 및 구축을 포함한 IT기술을 전문적으로 강의하는 우리 학원에는 강사가 10명이 있으며 각각의 강사는 자신이 담당하는 과목이 한과목인 강사도 있고 두 과목이상 강의하는 강사도 있다. 우리 학원에 개설된 과목은 총 17개의 과목이며 강의실은 6개가 존재하므로 강의 계획을 잘 수립하여야 한다. 강의를 끝나면 그 주에 강의했던 내용에 대해 수강생으로부터 설문서를 받아 다음 강의에 참고 자료로 이용한다. 매 주 1회씩 강사회의를 하여 서로의 강의내용과 수강생에 대한 이야기를 주고받는다. 과목에 따라서는 선수과목을 듣지 않으면 수강할 수 없는 과목도 존재한다.

① 업무기술서에서 명사를 구분한다.

모델링, 데이터베이스의 설계 및 구축, IT기술, 강의, 학원, 강사, 담당하는 과목, 과목, 학원, 강의실, 강의 계획, 수강생, 설문서, 참고 자료, 강사회의, 강의내용, 이야기, 선수과목, 수강

② 개념이 불분명한 것, 광범위한 것은 제거한다.

모델링, 데이터베이스의 설계 및 구축, IT기술, 강의, 학원, 강사, 담당하는 과목, 과목, 학원, 강의실, 강의 계획, 수강생, 설문서, 참고 자료, 강사회의, 강의내용, 이야기, 선수과목, 수강

③ 엔티타입의 특성인 것은 제거한다.

강의, 강사, 담당하는 과목, 과목, 강의실, 강의 계획, 수강생, 설문서, 참고 자료, 강사회의, 강의내용, 이야기, 선수과목, 수강

④ 업무 프로세스에 해당되는 명사는 제거한다.

강의, 강사, 담당하는 과목, 과목, 강의실, 강의 계획, 수강생, 참고 자료, 강사회의, 이야기, 선수과목, 수강

㉔ 중복되는 명사는 제거한다. (없음)

강의, 강사, 담당하는 과목, 과목, 강의실, 강의 계획, 수강생, 참고 자료, 선수과목, 수강

㉕ 누락된 엔티타입 정보를 유추한다.

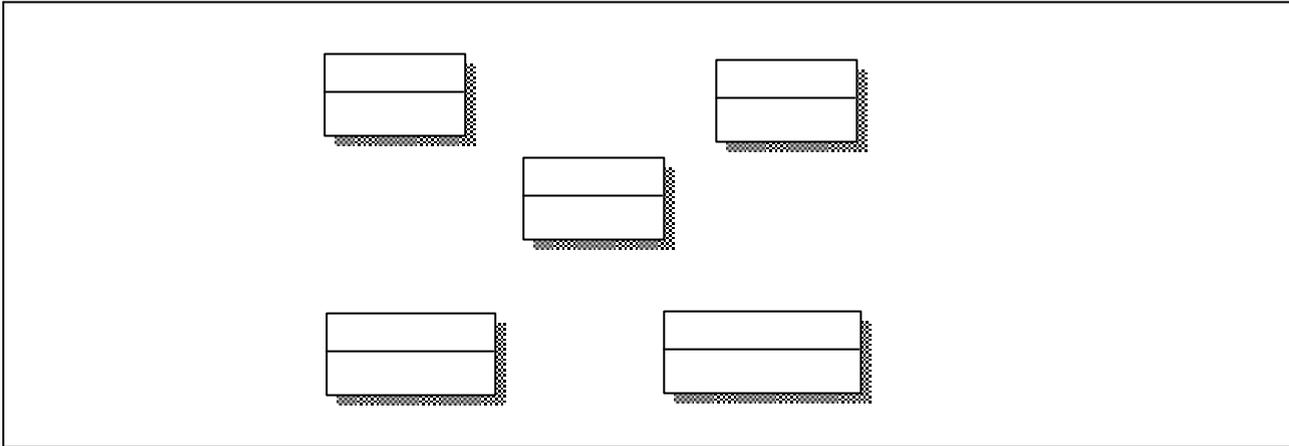
(인터뷰를 통해 강사이외에 행정적으로 강의를 주관하는 사람이 있을 경우)

강의담당자를 추가함

최종 엔티타입 대상은 강의, 강사, 담당하는 과목, 과목, 강의실, 강의 계획, 수강생, 참고 자료, 선수과목, 수강, 강의담당자이다.

제2장 - WORKSHOP 관계정의 - 가이드

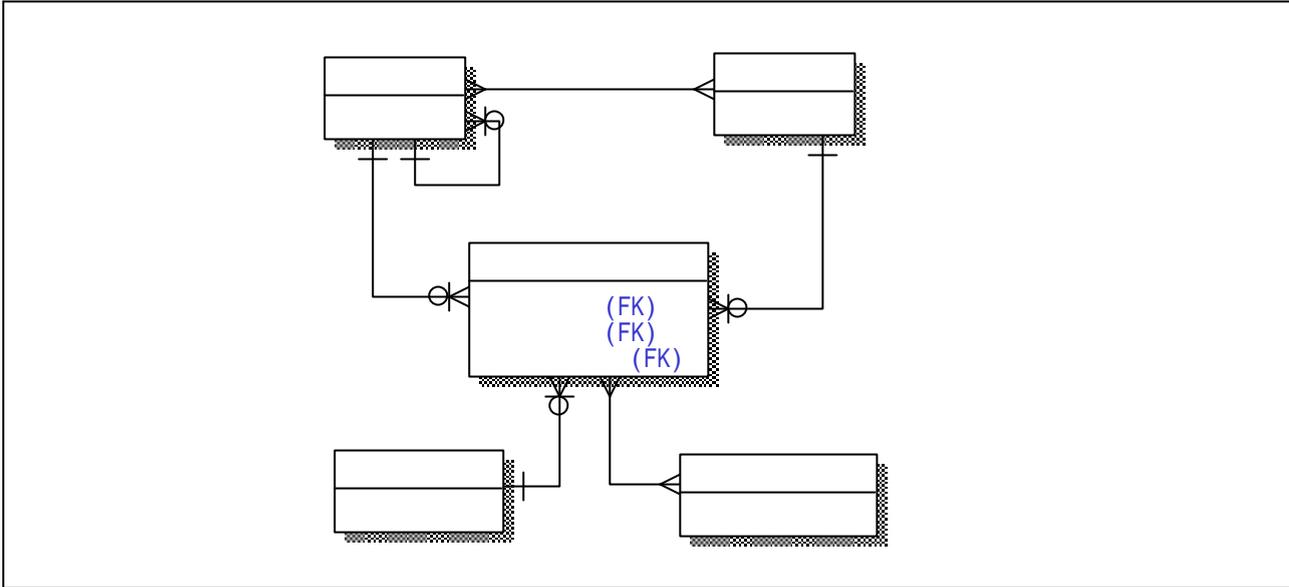
1. 엔티타입 선정방법에 의해 업무 적인 의미가 있는 명사로부터 엔티타입을 선정한다.



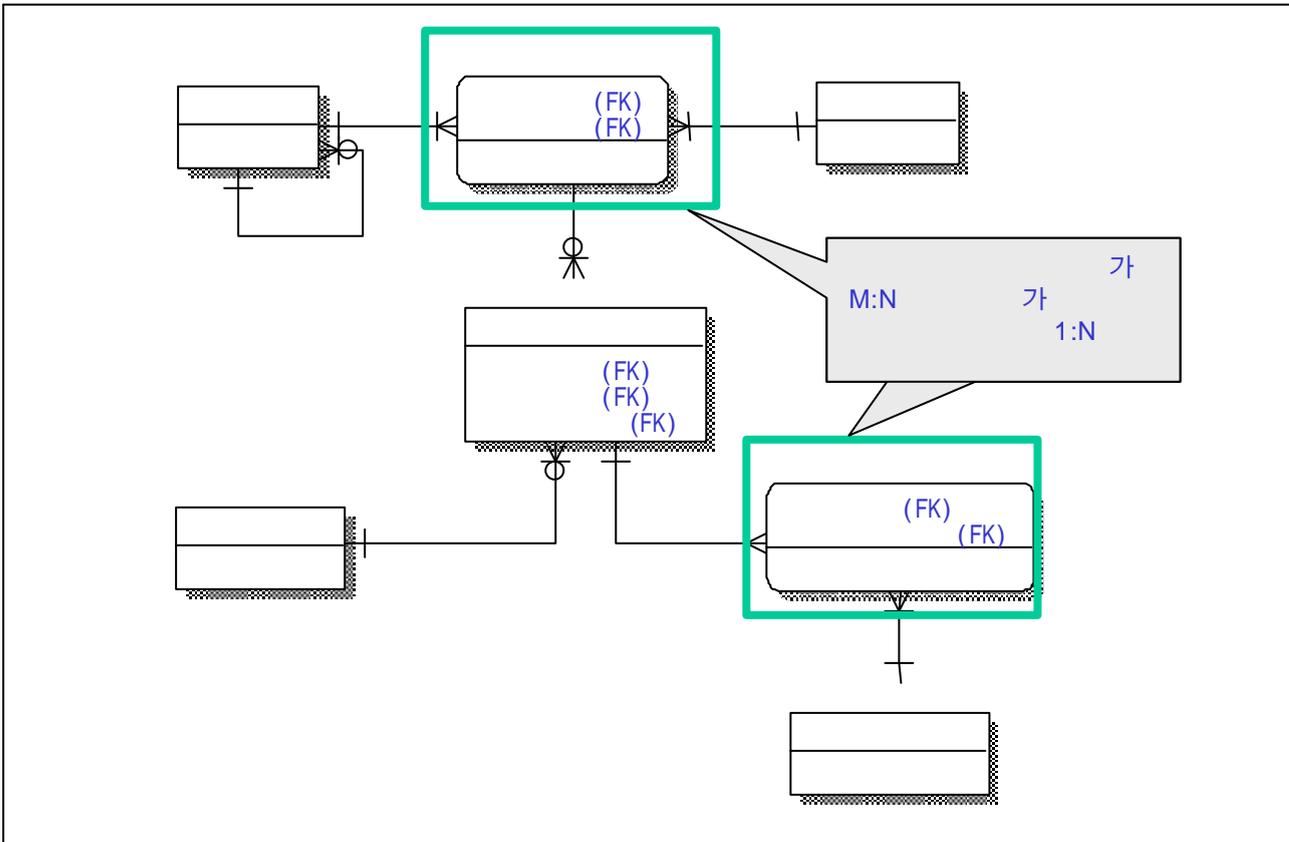
2. 선정된 엔티타입간 관계를 조사하여 관계정의서를 작성한다.

기준 엔티타입	관계형태(방향, 참여도, 참여방법)	참여 방법	관련 엔티타입
강사	각각의 강사는 한 과목, 여러 과목을 담당할 수 있다. 각 과목은 담당하는 여러 강사가 담당할 수 있다.	필수 선택	과목
강의	각각의 강의는 한 강의실에서 강의를 한다. 각각의 강의실은 여러번 강의를 이루어질 수 있다..	필수 선택	강의실
강의	각각의 강의는 한 과목을 강의를 한다. 각각 과목은 여러 번 강의를 이루어질 수 있다..	필수 선택	과목
강의	각각의 강의에는 한 강사가 강의한다. 한 강사는 여러 강의를 강의할 수 있다.	필수 선택	강사
강의	각각의 강의에는 여러 수강생이 수강한다. 한 명의 수강생은 여러 강의를 수강할 수 있다.	필수 선택	수강생
과목	각각의 과목에는 여러 과목을 선수과목이 있을 수 있다. 각 선수과목은 반드시 과목에 등록되어 있다.	선택 필수	과목

3. 관계정의서를 이용하여 ERD의 관계를 완성한다.

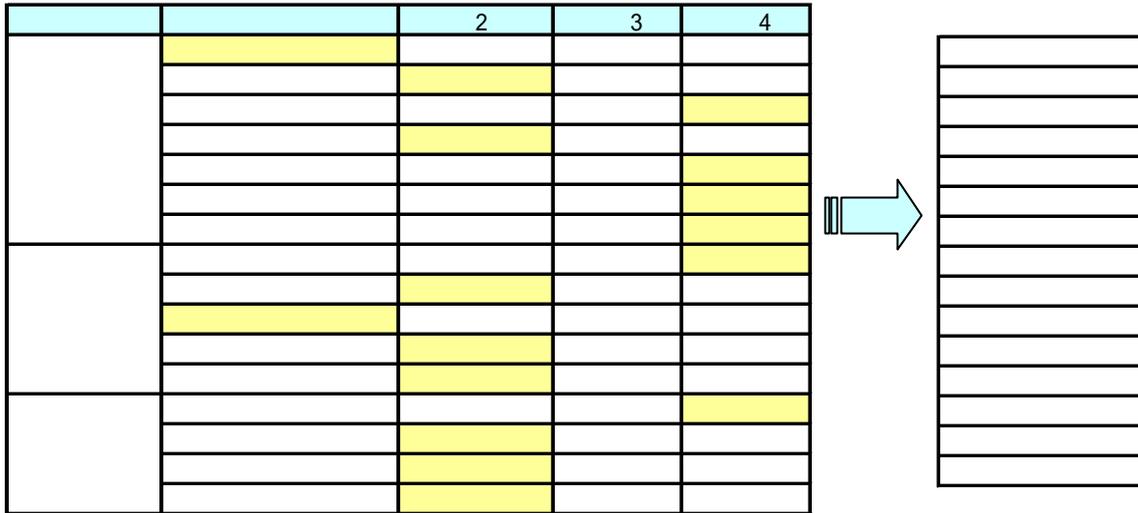


** 추가 고려사항 M:N 관계를 해소한다.



제2장 WORKSHOP 도메인정의 - 가이드

1.2.3. 첫 번째 데이터모델의 모든 속성을 나열하고 속성의 접미어를 분리하여 구분하도록 한다.



4. 분리된 접미어를 비슷한 것끼리 묶어 그룹을 만들어 이름을 부여한다.

	(N, 7)
	(N, 10)
	(V, 40)
	(V, 80)
	(V, 500)
	(V, 100)
ID	

5. 각 도메인별로 데이터타입과 길이를 지정한다(도메인정의를 만든다).

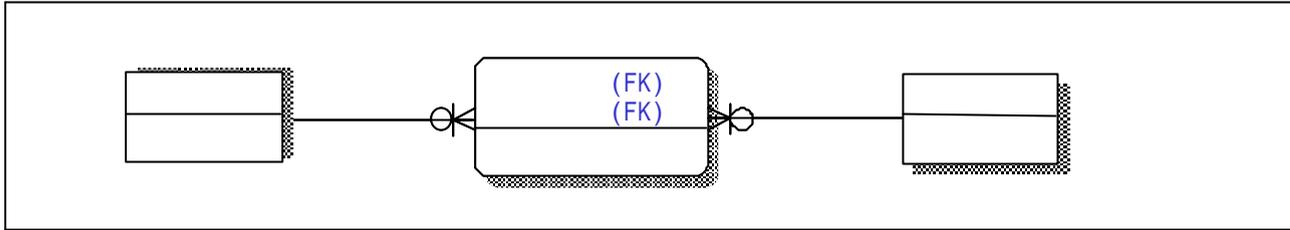
		VARCHAR2(10)	YYYYMM+SEQ(7)
		VARCHAR2(20)	
		VARCHAR2(13)	'-'
		VARCHAR2(18)	'-'
		CHAR(7)	'-'
		VARCHAR2(10)	
		VARCHAR2(14)	YYYYMMDD+SEQ(7)
		VARCHAR2(8)	YYYYMMDD
		CHAR(6)	YYYYMM
	(N,7)	NUMBER(7)	
	(N,10)	NUMBER(10)	
		VARCHAR2(120)	
	(V,40)	VARCHAR2(40)	
	(V,80)	VARCHAR2(80)	
	(V,500)	VARCHAR2(500)	
	(V,100)	VARCHAR2(100)	
ID		VARCHAR2(40)	
		VARCHAR2(200)	

6. 각 엔티티타입의 속성에 도메인을 할당한다.

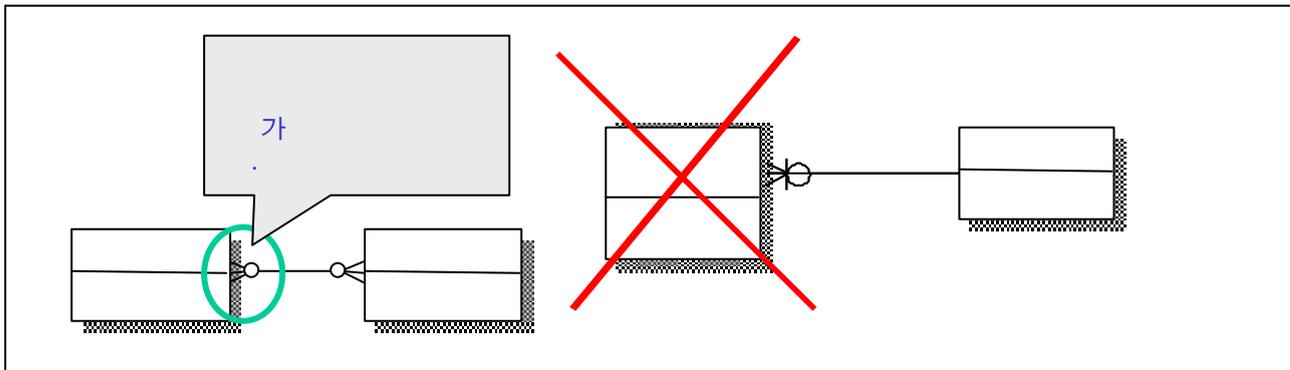
			VARCHAR2(10)
		(V,40)	VARCHAR2(40)
			VARCHAR2(13)
			VARCHAR2(120)
			VARCHAR2(18)
			CHAR(7)
			VARCHAR2(40)
			VARCHAR2(10)
		(V,40)	VARCHAR2(40)
			VARCHAR2(20)
			VARCHAR2(200)
			CHAR(6)
			VARCHAR2(14)
		(V,100)	VARCHAR2(100)
			VARCHAR2(8)
		(N,10)	NUMBER(10)

제3장 WORKSHOP M:N(MANY TO MANY) 관계해소 방법 - 가이드

① 관계 엔티타입을 분리하는 방법으로 M:N 관계를 해소한다.



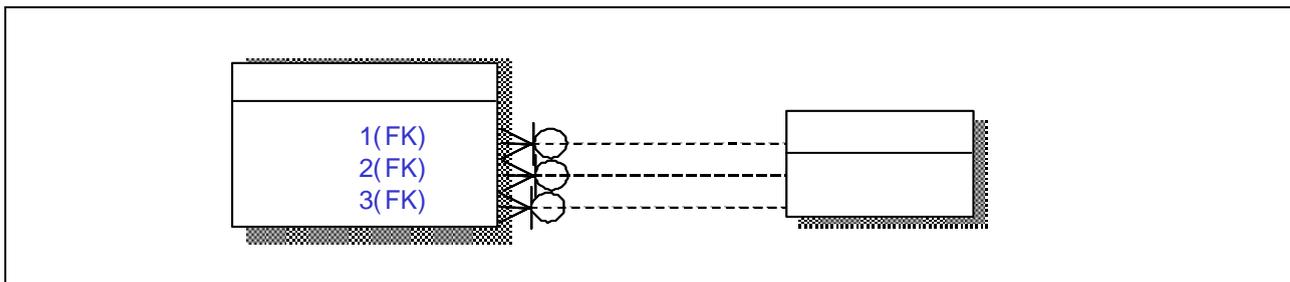
② 주식별자를 정의함으로써 문제의 학생과 강의의 M:N관계를 해결하면 안된다. 양쪽 모두 선택관계(OPTIONAL RELATIONSHIP)이므로 주식별자를 정의하여 엔티타입을 통합할 경우 반드시 강의를 수강한 학생만이 학생 엔티타입에 와야 하는 RULE이 있어야 한다. 그러나 현재 관계가 선택관계이므로 강의를 듣지 않은 학생도 독립적으로 존재할 수 있는 모습이다. 물론 강의번호를 NON-IDENTIFYING 속성으로 강의와 연결하여 주식별자를 학번 + 순번으로 하면 엔티타입 구성이 가능하지만 여러 개의 속성이 수강한 강의 수 만큼 반복적으로 나타나므로 많은 속성이 중복이 발생하여 관리(CREATE, UPDATE, DELETE)가 복잡해진다.



③ 속성을 추가하여 M:N 관계를 해소한다.

최대 세 번까지 수강할 수 있으므로 학생 엔티타입에 속성정보로 포함시켜 M:N관계를 해소 할 수 있다.

최대 세 번까지 수강할 수 있으므로 학생 엔티타입에 속성정보로 포함시켜 M:N관계를 해소 할 수 있다.

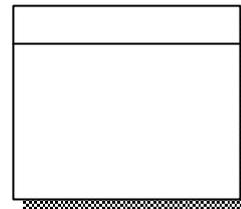


제3장 WORKSHOP 1:1(ONE TO ONE) 관계해소 방법 - 가이드

1. 양쪽모두 관계가 필수 관계라는 것은 두 엔티타입 모두 생사(LIFE CYCLE)를 같이 하는 하나의 엔티타입이라는 이야기이다. 그러므로 1번 시나리오의 장애발생과 장애조치는 하나의 엔티타입으로 통합되어야 한다.

하나의 장애발생에 대해서 한번의 장애조치사항이 이루어진다. 한번의 장애조치사항은 반드시

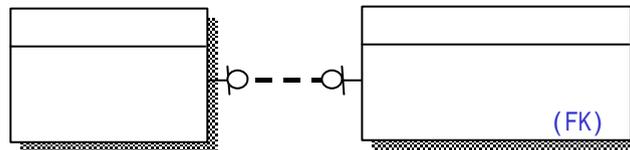
하나의 장애발생에 대해서 이루어진다.



2. 2번 시나리오는 두 엔티티 사이의 관계가 1:1이면서 양쪽모두 선택관계에 해당되는 데이터모델의 예이다.

하나의 장애발생에 대해서 한번의 장애조치사항이 이루어질 수 있거나 조치가 안 될 수도 있다. 한번의 장애조치사항은 하나의 장애발생에 대해서 이루어질 수도 있고 장애발생 내용 없이

조치될 수 있다.



일반적으로, 양쪽모두 선택관계 이면 관계를 잘못된 데이터모델로 관계를 삭제하는 경우가 많은데, 오히려 양쪽 모두 선택관계 라고 하여 반드시 관계가 없어지는 것은 아니라는 것을 기억해야 할 것이다. 2번 시나리오처럼 해당 장애발생에 대해서 조치할 때는 반드시 장애발생번호가 장애조치에 포함되어야 하는 것이다. 이러한 유형은 실제 프로젝트를 진행하면 많이 접하는 데이터모델이지만 관계를 잘못 설정한 경우가 많이 발견되는 유형이다.

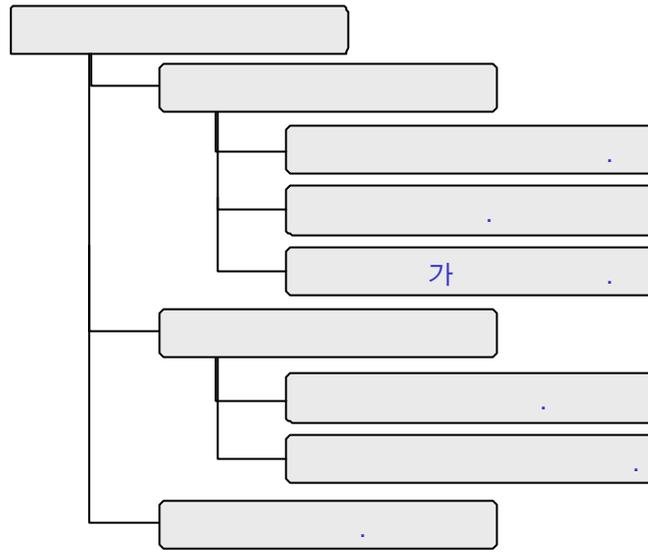
위의 모델로 DDL을 생성하면 장애발생번호는 속성의 NULL/NOT NULL 옵션 중 NULL로 지정이 되고

```
ALTER TABLE 장애조치
ADD ( FOREIGN KEY (장애발생번호)
REFERENCES 장애발생 );
```

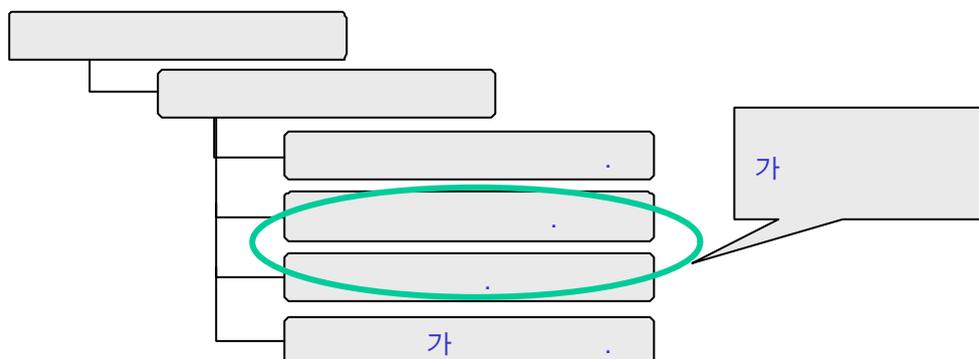
위의 문장을 이용하여 FOREIGN KEY를 생성 가능한 유형이 되는 것이다.

제4장 WORKSHOP 업무기능분해도 작성방법 - 가이드

책마을(주)의 예상되는 업무기능분해도는 다음과 같다.



크게 주문접수와 대여기간연기 그리고 반납이 상위 업무기능에 해당되고 그에 따른 ELEMENTARY PROCESS를 도출하였다. 책을 반납하는 기능은 더 이상 하위 ELEMENTARY PROCESS가 도출될 수 없으므로 상위 기능이 ELEMENTARY PROCESS가 되었다.



위의 예에서는 책 존재를 확인한다는 책을 대여할 때 항상 동시에 수행되어야 하는 업무기능이므로 두개의 ELEMENTARY PROCESS로 분리한 것은 부적당하다고 할 수 있다.

제5장 WORKSHOP ELEMENT PROCESS 와 ENTITY TYPE의 CRUD MATRIX - 가이드

예상되는 CRUD MATRIX는 다음과 같다.

CRUD MATRIX

ELEMENRARY PROCESS					
		R	C		
			R	C	R
	가		U		R
		R		R	C
				U	
			U	U	U

고객관리와 제품관리에 대해서는 시나리오에서 언급하지 않았으므로 데이터가 READ는 되고 있지만 생성되는 ELEMENTARY PROCESS는 발견되지 않는다. 만약 모든 업무가 다 기술되었다고 하면 반드시 생성에 해당되는 "C"가 포함되어 있어야 할 것이다.

제12장 WORKSHOP 엔티타입 정의 - 가이드

엔티타입 정의

엔티타입을 구분하는 6가지 원칙을 적용하여 시나리오에서 엔티타입을 도출한다. 맨 처음에는 엔티타입의 대상이 되는 명사형을 찾는 작업을 한다.

출판사 거래 시나리오를 이용하여 엔티타입을 도출하는 방법은 다음과 같다.

출판사 거래

- > 출판사와는 별도의 **계약**을 맺고 있으며 **계약**을 맺지 않은 출판사와는 **거래**를 하지 않는다.
- > 한 번 **계약**을 맺었다가 **해지** 할 수 있으며 다시 **계약**을 맺을 수도 있다. 이 때 **이전 계약** 상태에 대한 정보도 필요하다. **계약의 갱신**은 여러 번 가능하다.
- > **계약서**에는 **사업자번호**, **출판사명**, **계약일자**, **최저가 비율**이 포함된다.
- > **출판사**별로 **원가**에 대한 **최저가 비율**을 지정하여 비율에 의해 산정된 **최저금액**이하로는 **도서를 판매**할 수 없다.
- > **계약수정**에 의해 **최저가 비율**이 변경가능하다.
- > **출판사정보**는 **사업자번호**, **출판사명**, **출판사전화번호**, **출판사주소**, **거래은행코드**, **계좌번호**를 관리한다.
- > **출판사**에서 **도서**를 공급받으면 **도서**를 각각의 **목록**별로 분류하여 **참고**에 등록하고 **재고 현황**을 관리한다.
- > 관리하고자 하는 **도서의 정보**는 **도서번호**, **도서명**, **도서구분**, **저자**, **출판사**, **발행일자**, **정가**, **적용할인율** 이다. **적용할인율**은 **출판사별 최저가 비율**을 초과할 수 없다.
- > **출판사**에서 **신간서적**이 **발간**되면 **공급**받아서 **인터넷**에 **신간광고**를 한다.
- > **신간의 기준**은 **발행일**로부터 3개월 이내의 **서적**이다.
- > **인터넷**으로부터 **접수**되는 **도서주문**을 **접수**받아 **재고현황**을 파악하고 만약 **재고**가 남아있지 않을 경우는 **책**을 제조한 **출판사**에 연락하여 **책**을 **공급**받는다.
- > **출판사**에는 매달 **도서**가 매각되는 **현황**을 **보고서**로 **출력**하여 **송부**한다.

시나리오에서 명사를 이용하여 엔티타입을 도출한다.

- ① 엔티타입 대상 명사구분 - 출판사, 계약, 거래, 해지, 이전 계약, 계약의 갱신, 가능, 계약서, 사업자번호, 출판사명, 계약일자, 최저가 비율, 원가, 최저금액, 도서, 판매, 계약수정, 출판사정보, 사업자번호, 출판사전화번호, 출판사주소, 거래은행코드, 계좌번호, 공급, 목록, 참고, 등록, 재고현황, 도서의 정보, 도서번호, 도서명, 도서구분, 저자, 출판사, 발행일자, 정가, 적용할인율, 출판사, 최저가 비율, 신간서적, 발간, 신간광고, 신간의 기준, 발행일, 서적, 인터넷, 접수, 도서주문, 재고현황, 재고, 책, 매각, 현황, 보고서, 출력, 송부

② 개념이 불분명한 것, 광범위한 것 제거 - 출판사, 계약, 거래, 해지, 이전 계약, **계약의 갱신, 가능**, 계약서, 사업자번호, 출판사명, 계약일자, 최저가 비율, 원가, 최저금액, 도서, 판매, 계약수정, 출판사정보, 사업자번호, 출판사전화번호, 출판사주소, 거래은행코드, 계좌번호, 공급, 목록, 참고, 등록, 재고현황, 도서의 정보, 도서번호, 도서명, 도서구분, 저자, 출판사, 발행일자, 정가, 적용할인율, 최저가 비율, 신간서적, 발간, 신간광고, **신간의 기준**, 발행일, 서적, **인터넷**, 접수, 도서주문, 재고현황, 재고, 책, 매각, **현황**, 보고서, 출력, 송부

③ 엔티티타입 특성이나 속성제거 - 출판사, 계약, 거래, 해지, 이전 계약, 계약서, **사업자번호, 출판사명, 계약일자, 최저가 비율, 원가, 최저금액**, 도서, 판매, 계약수정, 출판사정보, **사업자번호, 출판사전화번호, 출판사주소, 거래은행코드, 계좌번호**, 공급, 목록, 참고, 등록, 재고현황, 도서의 정보, **도서번호, 도서명, 도서구분, 저자, 발행일자, 정가, 적용할인율**, 출판사, **최저가 비율**, 신간서적, 발간, 신간광고, **발행일, 서적**, 접수, 도서주문, 재고현황, 재고, 책, 매각, 보고서, 출력, 송부

④ 포괄적인 업무프로세스 제거 - 출판사, 계약, **거래, 해지**, 이전 계약, 계약서, 도서, **판매, 계약수정**, 출판사정보, 공급, 목록, 참고, 등록, 재고현황, 도서의 정보, 신간서적, **발간, 신간광고**, 접수, 도서주문, **재고현황**, 재고, 책, **매각, 보고서, 출력, 송부**

⑤ 중복되는 명사 중 한 개만 남기고 나머지 제거 - **출판사, 계약, 이전 계약, 계약서, 도서, 출판사정보**, 공급, 목록, 참고, **재고현황, 도서의 정보, 신간서적, 접수, 도서주문, 재고, 책**

* 접수와 주문은 업무적인 행위가 동일하게 발생하여 하나로 통합한다.

* 도서와 도서주문, 주문은 회원등록 및 주문에서 이미 도출되었으므로 중복으로 제거한다.

⑥ 누락된 엔티티타입 유추 - 출판사에서 공급한 상세 도서 내용을 관리해야 하므로 **공급도서를 유추한다.**

출판사, 계약, 공급, 목록, 참고, 재고

* 참고에 대한 관리는 고객과 협의하여 관리하지 않기로 하였다고 가정한다.

* 목록도 일반적으로 코드로만 관리하기로 하였다.

최종 엔티티타입 대상 : 출판사, 계약, 공급, 공급도서, 재고

엔티티정의서를 작성한다.

배송관리 시나리오를 이용하여 엔티티타입을 도출하는 방법은 다음과 같다.

배송관리

> 주문된 도서에 대해서는 **택배**를 이용하여 **직접 배송**하게 하는 방법과 **지하철**을 이용한 **간접 배송**하게 하는 방법을 이용한다.

엔티티타입명	엔티티타입 설명	동의어/유의어	엔티티타입 구분	관련 속성	비고
출판사	북사반과 거래하는 출판사의 정보를 관리	출판사정보	기본	사업자번호 출판사명 출판사전화번호 출판사주소 거래은행코드 계좌번호	
계약	출판사와 거래를 하기 위해 계약관계를 맺음	공급계약 계약서 이전계약	중심	사업자번호, 계약일자 최저가 비율	계약업 체로 변경
공급	출판사에서 도서를 공급한 기본내용	납품	중심	공급일자	
공급도서	출판사에서 도서를 공급한 상세 도서내용	공급도서목록	중심	도서번호 수량	
재고	출판사에서 공급한 남아있는 도서의 정보		행위	재고량	

- > 주문이 발생한 단위로 **배송**이 이루어지며 주문된 **개별도서별**로는 **배송**이 되지 않는다.
- > **배송요청서**에는 **배송일시**와 **배송지주소**, **배송지전화번호**와 **직접배송인지** **위탁배송인지**를 구분하는 내용이 있다.
- > **직접배송**을 위해 **택배업체**를 미리 등록하여 가능한 **택배업체**에 **용역**을 맡긴다.
- > **배송업체정보**는 **사업자번호**, **사업자명**, **배송업체전화번호**, **배송업체주소**가 있다.
- > **지하철 배송**을 위해 **잡지를 파는 상점** (이하 **지하철상점**이라 함)을 미리 등록하여 **책을 구매**한 **회원**들이 **찾아갈** 수 있도록 한다.
- > **배송업체**가 **배송**을 하는 **일자가 공휴일일** 경우 **특별 배송조건**에 따라 **용역비**를 추가한다. 예를 들어 **택배업체**가 **기본용역비** 건당 2000 원인데 1월1일에 **배송**을 할 경우 **용역비 30%**를 추가하여 2600원을 **계산**한다.
- > 관리하고자 하는 **지하철상점 정보**로는 **사업자번호**, **사업자명**, **지하철상점전화번호**, **지하철역**, **역내위치**가 있다.
- > 한 달간의 **배송실적**에 따라 매달30일에 **배송 용역비**를 **정산**한다. **정산방법**에 따라 **일괄 정산**이 있을 수 있고 **분납**할 수 있다. **최고 3회**까지 **분납**이 가능하다.

- ① 엔티티타입 대상 명사구분 - 주문, 도서, 택배, 직접 배송, 지하철, 간접 배송, 개별도서, 배송요청서, 배송일시, 배송지주소, 배송지전화번호, 위탁배송, 택배업체, 등록, 용역, 배송업체정보, 사업자번호, 사업자명, 배송업체전화번호, 배송업체주소, 지하철 배송, 잡지를 파는 상점, 지하철상점, 책, 구매, 회원, 배송업체, 일자, 공휴일, 특별 배송조건, 용역비, 지하철상점 정보, 지하철상점전화번호, 지하철역, 역내위치, 배송실적, 배송 용역비, 정산, 일괄정산, 분납
- ② 개념이 불분명한 것, 광범위한 것 제거 - 주문, 도서, 택배, 직접 배송, **지하철**, 간접 배송, **개별도서**, 배송요청서, 배송일시, 배송지주소, 배송지전화번호, 위탁배송, 등록, 택배업체, 용역, 배송업체정보, 사업자번호, 사업자명, 배송업체전화번호, 배송업체주소, 지하철 배송, **잡지를 파는 상점**, 지하철상점, 책, 구매, 회원, 배송업체, **일자**, **공휴일**, 특별 배송조건, 용역비, 지하철상점 정보, 지하철상점전화번호, 지하철역, 역내위치, 배송실적, 배송 용역비, 정산, 일괄정산, 분납
- ③ 엔티티타입 특성이나 속성제거 - 주문, 도서, 택배, 직접 배송, 간접 배송, 개별도서, 배송요청서, **배송일시**, **배송지주소**, **배송지전화번호**, 위탁배송, 택배업체, 등록, 용역, 배송업체정보, **사업자번호**, **사업자명**, **배송업체전화번호**, **배송업체주소**, 지하철 배송, 지하철상점, 책, 구매, 회원, 배송업체, **특별 배송조건**, **용역비**, 지하철상점 정보, **지하철상점전화번호**, **지하철역**, **역내위치**, 배송실적, **배송 용역비**, 정산, 일괄정산, 분납
- ④ 포괄적인 업무프로세스 제거 - 주문, 도서, 택배, 직접 배송, 간접 배송, 개별도서, 배송요청서, 위탁배송, 택배업체, 등록, **용역**, 배송업체정보, 지하철 배송, 지하철상점, 책, 구매, 회원, 배송업체, 지하철상점 정보, 배송실적, 정산, 일괄정산, 분납
- ⑤ 중복되는 명사 중 한 개만 남기고 나머지 제거 - 주문, 도서, **택배**, **직접 배송**, **간접 배송**, **개별도서**, **배송요청서**, **위탁배송**, **택배업체**, **배송업체정보**, **지하철 배송**, **지하철상점**, **책**, **구매**, **회원**, **배송업체**, **지하철상점 정보**, **배송실적**, **정산**, **일괄정산**, **분납**

- * 주문, 도서, 회원은 회원관리에서 도출한 엔티티타입으로 중복되어 제거한다.
- * 구매는 주문과 동일한 업무적인 행위가 발생된다.

⑥ 누락된 엔티티타입 유추 - 없음

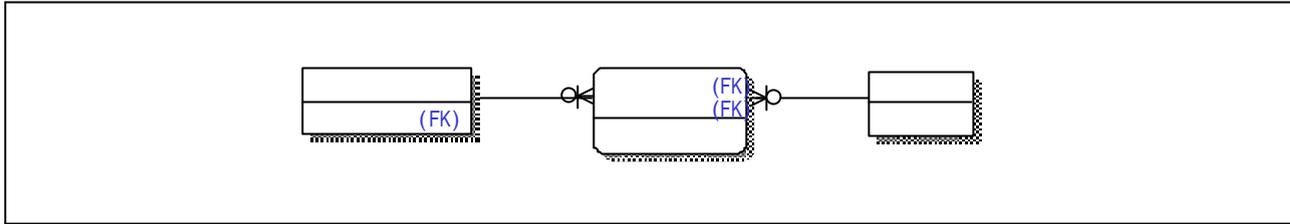
택배업체, 배송, 지하철상점, 정산, 분납

최종 엔티티타입 대상 : 택배업체, 배송, 지하철상점, 정산, 분납

엔티티정의서를 작성한다.

관계 정의

공급과 공급도서간의 관계가 **M:N** 관계이다.
 엔티타입을 추가하여 관계를 해소는 다음과 같다.



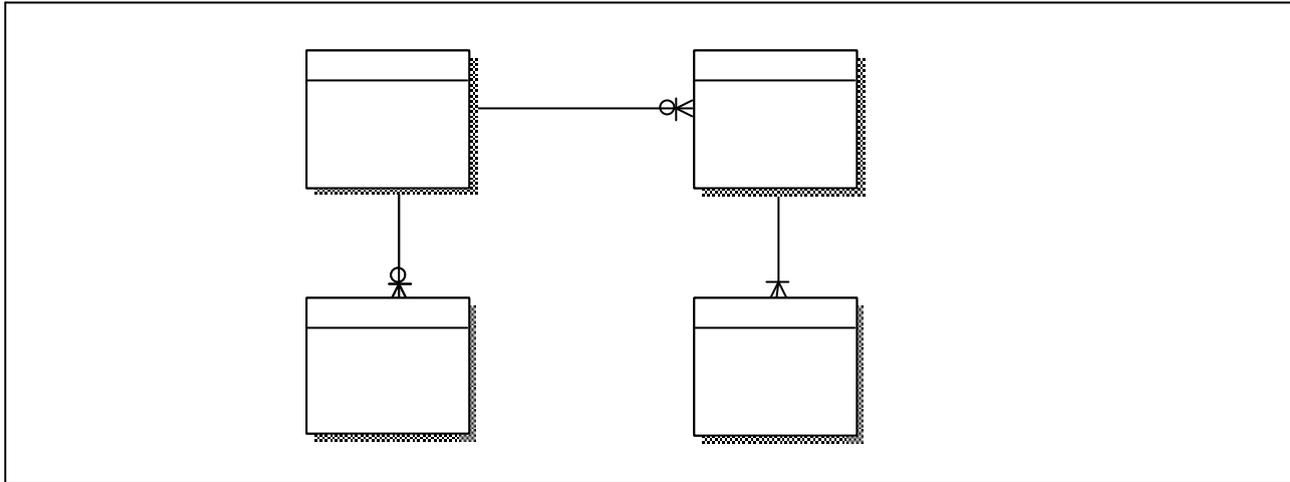
회원등록 및 주문관리에 대해 관계 정의서는 다음과 같다.

엔티타입명	엔티타입 설명	동의어/유의어	엔티타입 구분	관련 속성
배송	인터넷을 통해 주문된 도서를 배송하는 정보를 관리	직접배송 간접배송 배송요청서	중심	배송일시 배송지주소 배송지전화번호 배송구분배송
택배업체	주문된 회원에게 직접 배송하는 택배업체 관리	직접배송업체	기본	사업자번호 사업자명 배송업체전화번호 배송업체주소
지하철상점	주문된 서적을 회원이 찾아갈 수 있는 지하철 상점정보	간접배송업체	기본	사업자번호 사업자명 지하철상점전화번호 지하철역 역내위치
정산	배송을 한 용역비를 정산한 정보	일괄정산	행위	정산일자 정산금액
분납	용역비를 분납한 정보	분할정산	행위	분납금액

(엔티타입은 도서, 회원, 주문, 주문목록, 체크도서 이다.)

기준 엔티타입	관계 형태(방향, 참여도, 참여방법)	참여 방법	관련 엔티타입
회원	각각의 회원은 여러 번의 주문을 할 수 있다. 가 주문은 한 회원에 의해 주문될 수 있다.	선택 필수	주문
회원	각각의 회원은 여러 번 도서에 여러 번 체크할 수 있다. 각각의 체크도서는 한 회원에 의해 체크된다.	선택 필수	체크도서
주문	각각의 주문에는 여러 개의 주문목록을 포함한다. 각각의 주문목록은 하나의 주문내용에 포함된다.	필수 필수	주문목록

회원등록 및 주문관리에 대한 ERD는 다음과 같다.

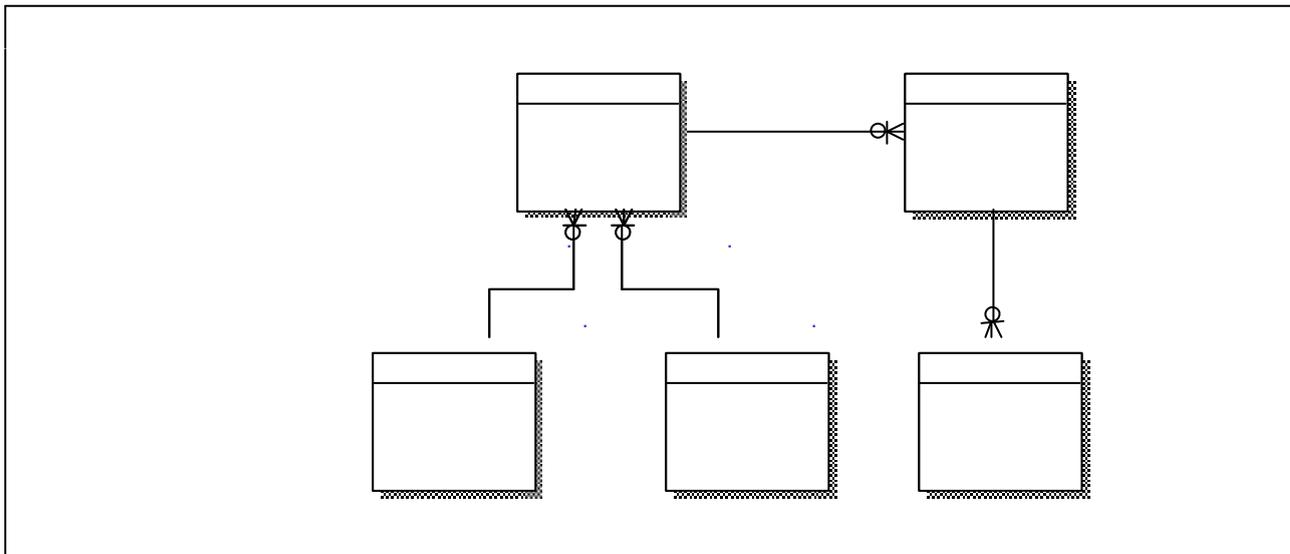


배송관리에 대해 관계 정의서는 다음과 같다.

(엔티타입은 택배업체, 배송, 지하철상점, 정산, 분납이다.)

기준 엔티타입	관계 형태(방향, 참여도, 참여방법)	참여 방법	관련 엔티타입
택배업체	각각의 택배업체는 여러 번 배송할 수 있다.	선택	배송
	각각의 배송은 하나의 배송업체에 의해 배송된다.	필수	
지하철상점	각각의 지하철상점은 여러 번 배송할 수 있다.	선택	배송
	각각의 배송은 하나의 지하철상점에 의해 배송된다.	필수	
배송	각각의 배송은 여러 번 정산될 수 있다.	선택	정산
	각각의 정산내용에는 한번의 배송내용을 정산한다.	필수	
정산	각각의 정산내용은 여러 번 분납할 수 있다.	선택	분납
	각각의 분납내용에는 한번의 정산내용이 분납된다.	필수	

배송관리에 대한 ERD는 다음과 같다.



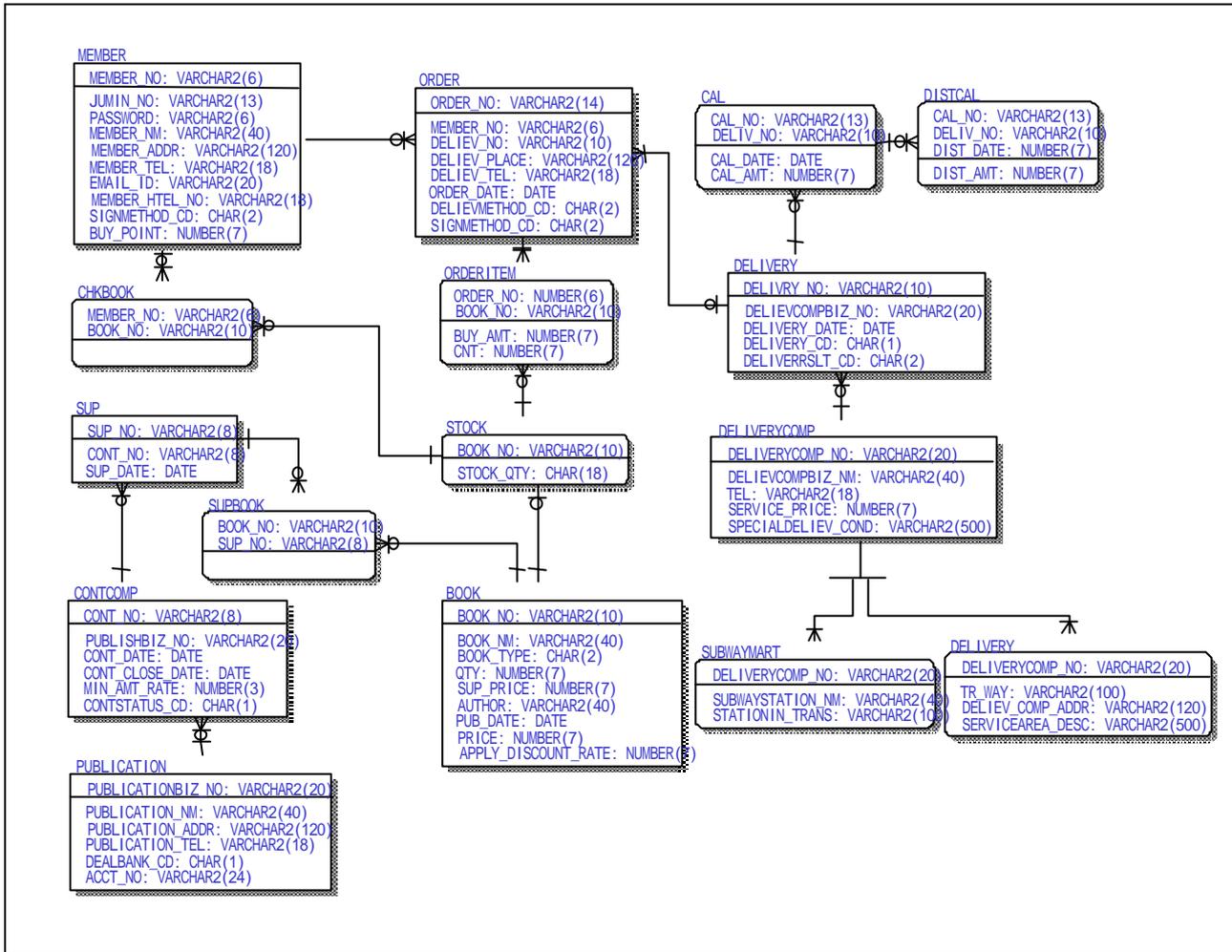
이 외에도 회원 및 주문관리, 배송관리, 출판사거래관리간의 상호 엔티타입 간의 관계를 도출하고 이를 관계정의서에 기술하고 ERD에 표현할 수 있다. 이 후에 전체적인 ERD에서 참조하기를 바란다.

용어사전 정의

용어사전을 정의한 내용은 다음과 같다.

가	AMT	, ~가	가
	CNT	~	
	DEAL		
	RSLT		
	SIGN	, 가	/
	CONT		
	ACCT		
	SUP		
	BUY		
	TYPE		
	AMT	가	
	IN		
	DESC		
가	PRICE		가
	BOOK		
	NM		
	PUB		
	METHOD		
	DEL LEV		
	DEL LEV PLACE		
	NO		
	RATE		
	BIZ		
	STATUS		
	SERVICE		
	WAY		
	QTY		
	COMP		
	STATION		
	SERVICE		
	TR		
	TRANS		
	BANK		
	DATE		
	DATE		
	STOCK		
	AUTHOR		
	APPLY		
	EMAIL ID		ID
	TEL		
가	PRICE	가	가
	CAI		
	COND		
	ORDER		
	JUM INNO		
	ADDR		
	AREA		
	SUBWAY		
가	MIN AMT		가 가
	PUBLICATION		
	CD		
	DELEV		
	SPECIAL		
	POINT		가
	DC RATE	DC	
	MEMBER		
	HTEL		

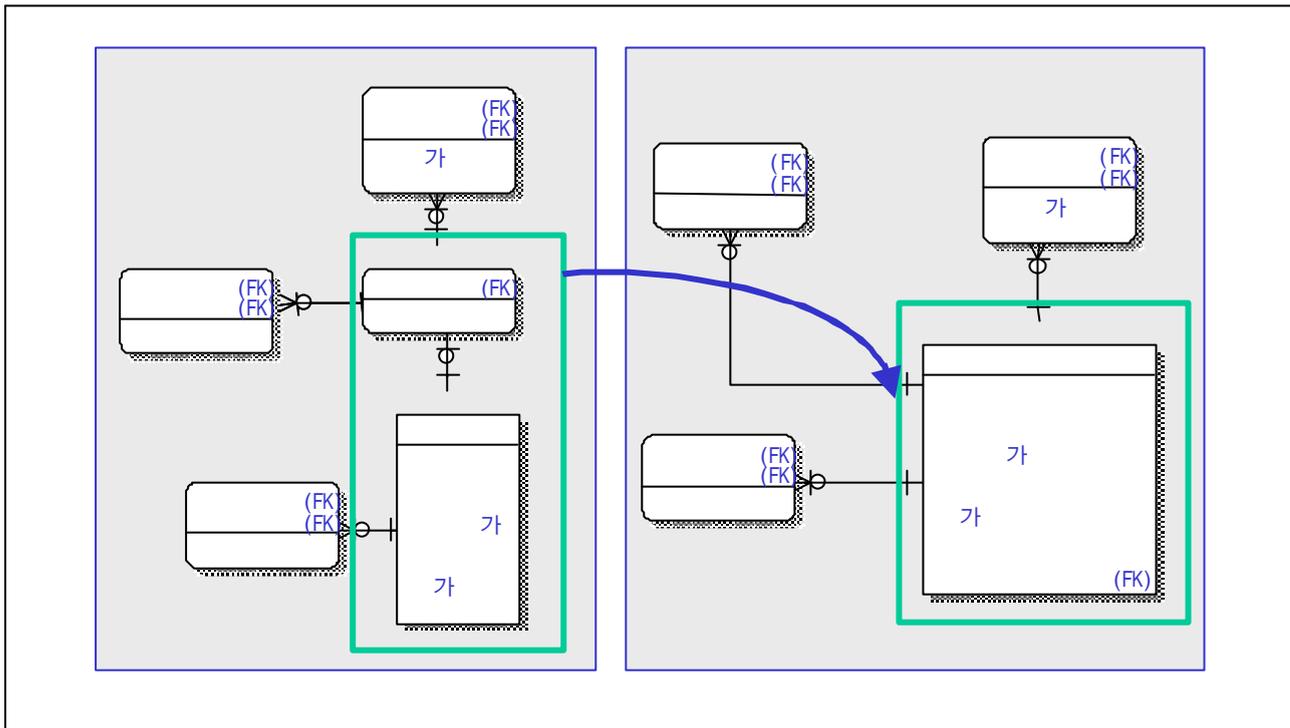
용어사전과 도메인정의서를 이용하여 물리명과 데이터타입이 표시된 ERD는 다음과 같다.



엔티타입 통합

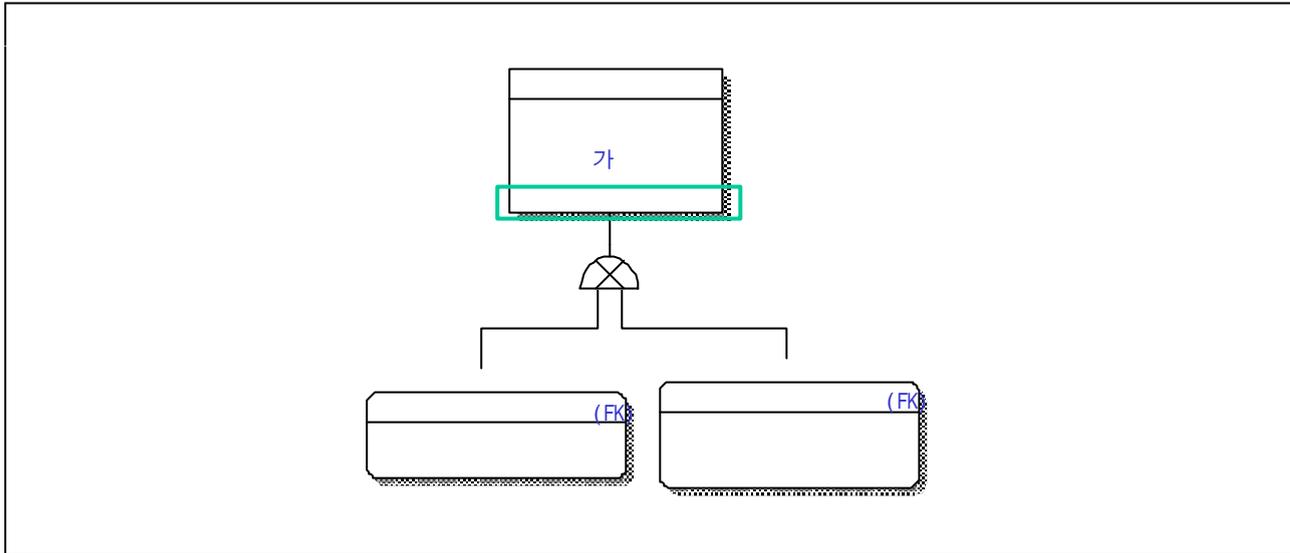
설계된 ERD에서 통합할 수 있는 엔티타입과 관계는 어떤 것이 있는지 조사하고 통합하도록 한다.

예에서는 재고와 도서가 1:1 관계이므로 통합의 대상이 된다. 다음은 재고와 도서 엔티타입을 통합한 모델이다.



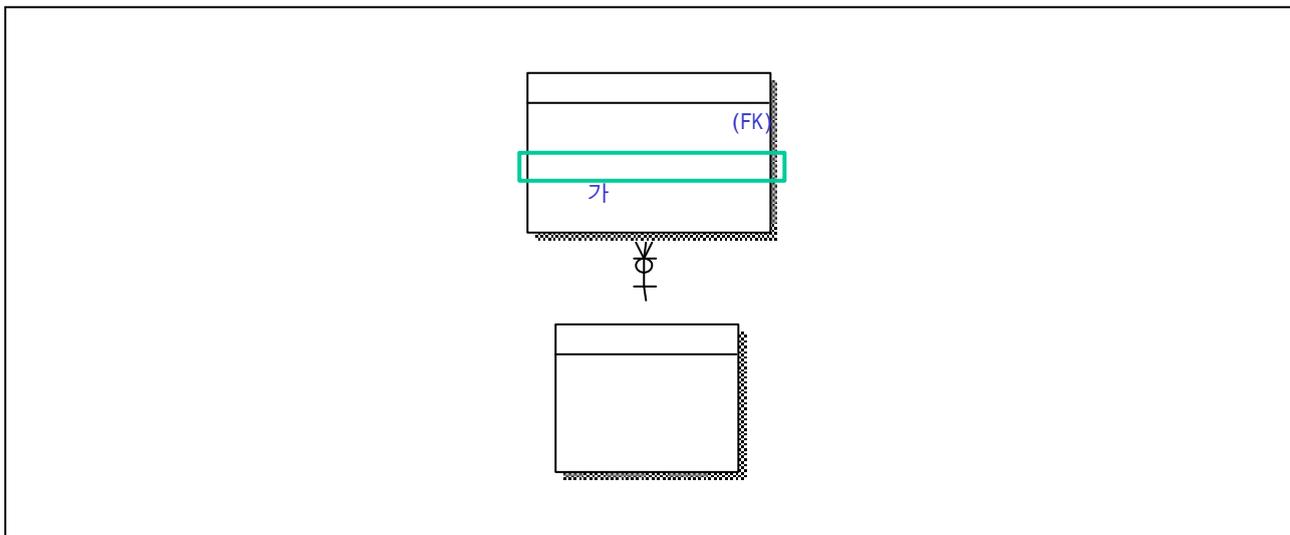
데이터모델 검토- 슈퍼타입/서브타입 모델검토

예에서는 슈퍼타입과 서브타입으로 나누어지기 위한 구분자가 누락된 경우이다. 즉 배송업체 엔티타입에 지하철상점과 택배업체를 구분하기 위한 배송업체구분코드가 누락되어 있다.



데이터모델 검토- 이력 모델검토

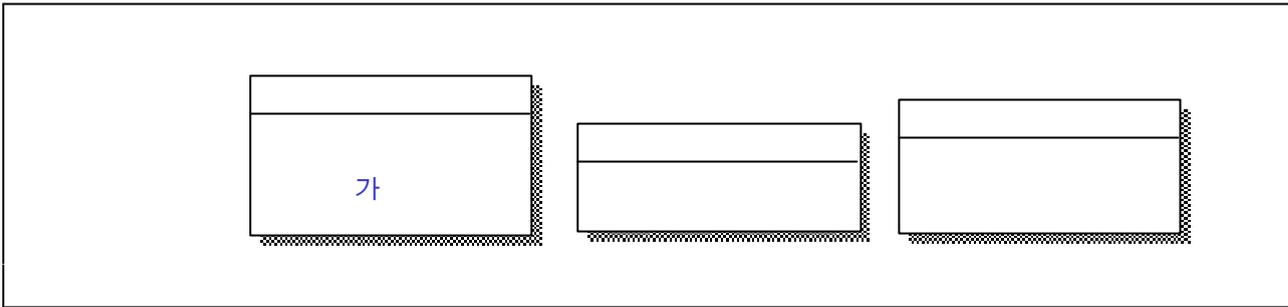
진행이력의 경우 이력의 시작과 종료를 나타내는 속성이 필요하므로 계약업체 엔티타입에 계약종료일자를 추가한다.



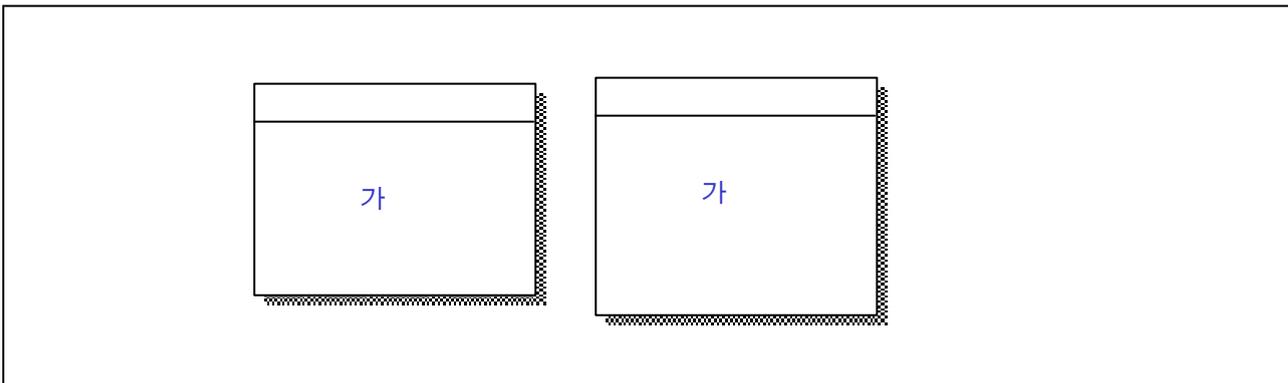
ERD를 관계형테이블로 전환

수퍼타입과 서브타입으로 나누어져 있는 배송업체와 지하철상점, 택배업체를 3가지의 경우의 수의 테이블로 전환한다.

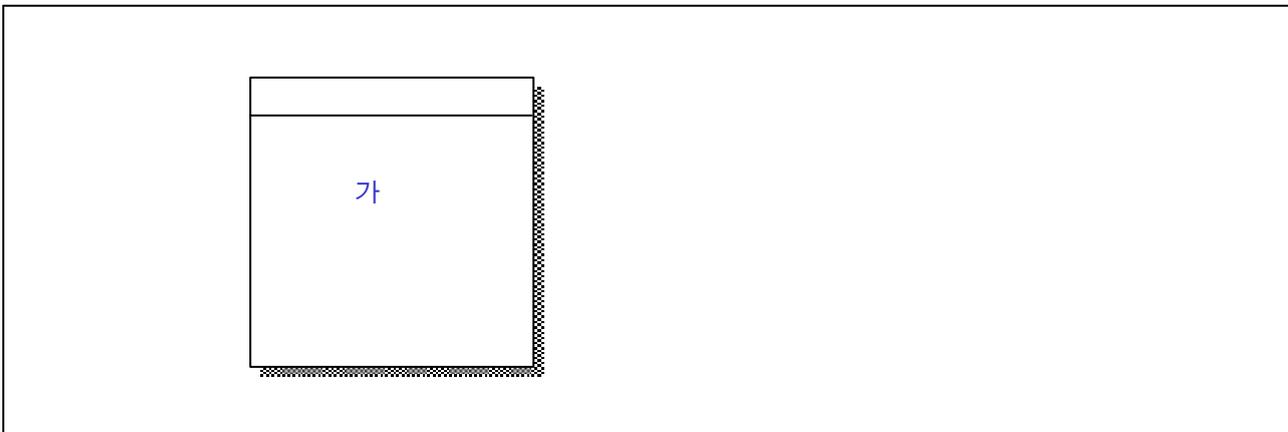
1. 각각의 엔티타입을 개별 테이블로 전환



2. 수퍼타입과 서브타입을 통합하여 전환

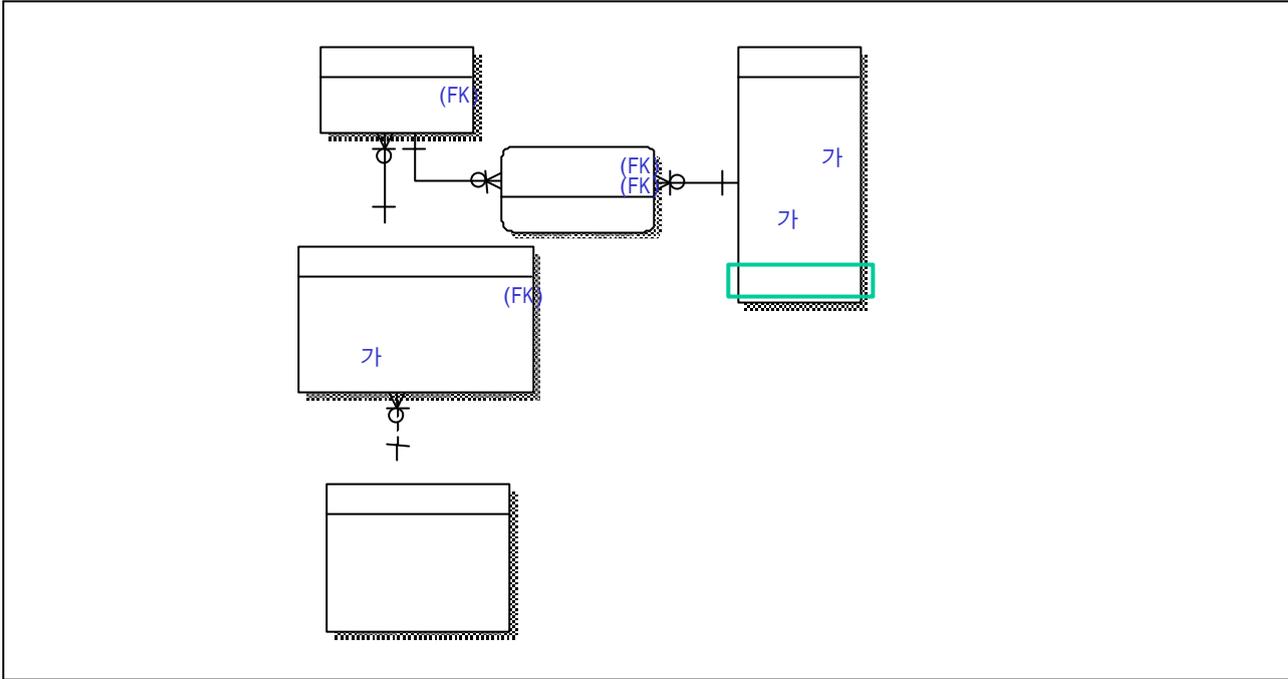


3. 하나의 테이블로 전환



반정규화

다음은 속성의 반정규화를 적용한 모델이다.



반정규화를 적용한 모델에서 도서번호가 = '2002070101'인 정보들을 도서번호, 출판사명, 도서명, 도서구분, 정가, 할인을 정보를 조회하는 SQL문장이다.

```
SELECT 도서번호, 출판사명, 도서명, 도서구분, 정가, 적용할인율
FROM 도서
WHERE A.도서번호 = '2002070101'
```

테이블정의서

물리 테이블의 관계를 이용하여 테이블정의서를 작성한다.

							2002	7	1
ORDERITEM									
						NULL		KEY	
1	가	ORDER_NO		VARCHAR2	6	NOT NULL		PK/FK	
2		BOOK_NO		VARCHAR2	10	NOT NULL		PK/FK	
		BUY_AMT		NUMBER	7	NOT NULL			
		CNT		NUMBER	7	NOT NULL			

							2002	7	1
CHKBOOK									
						NULL		KEY	
1		MEMBER_NO		VARCHAR2	6	NOT NULL		PK/FK	
2		BOOK_NO		VARCHAR2	10	NOT NULL		PK/FK	

							2002	7	1
BOOK									
						NULL		KEY	
1	가	BOOK_NO		VARCHAR2	10	NOT NULL		PK	
2		BOOK_NM	(V,40)	VARCHAR2	40	NOT NULL			
3		BOOK_TYPE		CHAR	2	NULL			
4	가	QTY		NUMBER	7	NOT NULL			
5		SUP_PRICE		NUMBER	7	NOT NULL			
6	가	AUTHOR	(V,40)	VARCHAR2	40	NULL			
7		PUB_DATE		DATE		NOT NULL			
8		PRICE		NUMBER	7	NOT NULL			
9		APPLY_DISCOUNT_RATE		NUMBER	7	NULL			
10		STOCK_QTY		NUMBER	7	NOT NULL			
11		PUBLICATIONBIZ_NO		NUMBER	4	NOT NULL		FK	

						2002 7 1		
SUP								
						NULL		KEY
1	SUP_NO			VARHCHAR2	8	NOT NULL		PK/FK
2	CONT_NO			VARHCHAR2	8	NOT NULL		
3	SUP_DATE			DATE	-			

						2002 7 1		
SUPBOOK								
						NULL		KEY
1		BOOK_NO		VARCHAR2	10	NOT NULL		PK/FK PK/FK
2		SUP_NO		VARCHAR2	8	NOT NULL		

						2002 7 1		
CONTCOMP								
						NULL		KEY
1	CONT_NO			VARHCHAR2	8	NOT NULL		PK FK
2	PUBLISHBIZ_NO			VARHCHAR2	20	NOT NULL		
3	CONT_DATE			DATE	-	NULL		
4	CONT_CLOSE_DATE			DATE	-	NULL		
5	MIN_AMT_RATE	가		NUMBER	3	NULL		
6	CONSTATUS_CD			CHAR	1	NULL		

						2002 7 1		
PUBLICATION								
						NULL		KEY
1		PUBLICATIONBIZ_NO		VARCHAR2	20	NOT NULL		PK
2		PUBLICATION_NM	(V,40)	VARCHAR2	40	NOT NULL		
3		PUBLICATION_ADDR		VARCHAR2	120			
4		PUBLICATION_TEL		VARCHAR2	18			
5		PUBLICATION_TEL		CHAR	1			
6		ACCT_NO		VARCHAR2	24			

						2002 7 1	
DELIVERY							
						NULL	KEY
1	DELIVERY_NO			VARHCHAR2	10	NOT NULL	PK
2	DELIVCOMPBIZ_NO			VARHCHAR2	20	NOT NULL	
3	DELIVERY_DATE			DATE	-	NULL	
4	DELIVERY_CD			CHAR	1	NULL	
5	DELIVERYRSLT_CD			CHAR	2		

						2002 7 1	
DELIVERYCOMP							
						NULL	KEY
1	가	DELIVERYCOMP_NO		VARHCHAR2	20	NOT NULL	PK
2		DELIVCOMPBIZ_NM	(V,40)	VARHCHAR2	40	NOT NULL	
3		TEL		VARHCHAR2	18	NULL	
4		SERVICE_PRICE	1	NUMBER	7	NULL	
5		SPECIALDELIEV_COND	(V,500)	VARHCHAR2	500	NULL	
6		DELIEVCOMPTYPE_CD		CHAR	1	NULL	
7		TR_WAY	(V,100)	VARHCHAR2	100	NULL	
8		DELIEV_COMP_ADDR	1	VARHCHAR2	120	NULL	
9		SERVICEAREA_DESC	(V,500)	VARHCHAR2	500	NULL	
10		SUBWAYSTATION_NM	(V,40)	VARHCHAR2	40	NULL	
11		STATIONIN_TRANS	(V,100)	VARHCHAR2	100	NULL	

						2002 7 1	
CAL							
						NULL	KEY
1	CAL_NO			VARHCHAR2	13	NOT NULL	PK/FK PK/FK
2	DELIV_NO			VARHCHAR2	10	NOT NULL	
3	CAL_DATE			DATE	-	NOT NULL	
4	CAL_AMT			NUMBER	7	NOT NULL	

인덱스정의서

기본적인 인덱스에 대한 인덱스정의서는 다음과 같다.

	MEMBER	PK_MEMBER	MEMBER_NO	VARCHAR2(6)	ISINDEX01	UNIQUE	ASC	PK
		I_MEMBER01	JUMIN_NO	VARCHAR2(13)	ISINDEX01	UNIQUE	ASC	INDEX
	ORDER	PK_ORDER	ORDER_NO	VARCHAR2(14)	ISINDEX01	UNIQUE	ASC	PK
		I_ORDER01	MEMBER_NO	VARCHAR2(13)	ISINDEX01	NON-UNIQUE	ASC	FK
		I_ORDER02	DELIEV_NO	VARCHAR2(10)	ISINDEX01	NON-UNIQUE	ASC	FK
	ORDERITEM	PK_ITEM	ORDER_NO	NUMBER(6)	ISINDEX01	UNIQUE	ASC	PK
			BOOK_NO	VARCHAR2(10)				
		I_ORDER01	BOOK_NO	VARCHAR2(10)	ISINDEX01	NON-UNIQUE	ASC	FK
	BOOK	PK_BOOK	BOOK_NO	VARCHAR2(10)	ISINDEX01	UNIQUE	ASC	PK
		I_BOOK01	PUBLICATIONBIZ_NO	NUMBER(4)	ISINDEX01	NON-UNIQUE	ASC	INDEX
	CHKBOOK	PK_CHKBOOK	MEMBER_NO	VARCHAR2(6)	ISINDEX01	UNIQUE	ASC	PK
			BOOK_NO	VARCHAR2(10)				
	SUP	PK_SUP	SUP_NO	VARCHAR2(8)	ISINDEX01	UNIQUE	ASC	FK
	SUPBOOK	PK_SUPBOOK	BOOK_NO	VARCHAR2(10)	ISINDEX01	UNIQUE	ASC	PK
			SUP_NO	VARCHAR2(8)				
	CONTCOMP	PK_CONTCOMP	CONT_NO	VARCHAR2(8)	ISINDEX01	UNIQUE	ASC	PK
		I_CONTCOMP01	PUBLISHBIZ_NO	CHAR(1)	ISINDEX01	UNIQUE	ASC	FK
	PUBLICATION	PK_PUBLICATION	PUBLICATIONBIZ_NO	VARCHAR2(20)	ISINDEX01	UNIQUE	ASC	PK
	DELIVERY	PK_DELIVERY	DELIVERY_NO	VARCHAR2(10)	ISINDEX01	UNIQUE	ASC	PK
		I_DELIVERY01	DELIEVCOMPBIZ_NO	VARCHAR2(20)	ISINDEX01	UNIQUE	ASC	FK
	DELIVERYCOMP	PK_DELIVERYCOMP	DELIVERYCOMP_NO	VARCHAR2(20)	ISINDEX01	UNIQUE	ASC	PK
	CAL	PK_CAL	CAL_NO	VARCHAR2(13)	ISINDEX01	UNIQUE	ASC	PK
			DELIV_NO	VARCHAR2(10)				

뷰 정의서

주문과 주문목록에 조회가 대부분이 동시에 조회가 되어 다음과 같이 뷰를 생성한다.

뷰명	테이블명	필드명	데이터형식
V_ORDERITEM		ORDER	ORDER_NO MEMBER_NO DELIEV_NO DELIEV_PLACE DELIEV_TEL ORDER_DATE DELIEVMETHOD_CD SIGNMETHOD_CD VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 VARCHAR2 DATE CHAR CHAR
		ORDERITEM	ORDER_NO BOOK_NO BUY_AMT CNT VARCHAR2 VARCHAR2 NUMBER NUMBER

테이블스페이스 용량산정서

테이블 용량에 따라 테이블스페이스 용량을 산정한 예이다.

테이블명	용량	테이블스페이스명	총용량	파일명
CAL	30M	TS001	150M	DF001.DBF
CHKBOOK	5M			
CONTCOMP	5M			
DELIVERY	30M			
DELIVERYCOMP	20M			
MEMBER	30M	TS002	200M	DF002.DBF
ORDER	50M			
ORDERITEM	50M			
PUBLICATION	5M			
SUP	5M			
BOOK	10M			

인덱스스페이스 용량산정서

인덱스 용량을 산정한 예이다.

		CAL	5M	IS001	100M	DF003.DBF
		CHKBOOK	5M			
		CONTCOMP	5M			
		DELIVERY	30M			
		DELIVERYCOMP	10M			
		MEMBER	30M	IS002	200M	DF004.DBF
		ORDER	60M			
		ORDERITEM	30M			
		PUBLICATION	10M			
		SUP	10M			
	BOOK	20M				

데이터파일 용량산정서

테이블 스페이스 용량산정에 따라 데이터파일의 용량산정 예이다.

C:	C:\ORADATA\DB1	10G	3G	30%	DF001.DBF	200M
					DF002.DBF	300M
D:	D:\ORADATA\DB2	10G	6.5G	65%	DF003.DBF	200M
					DF004.DBF	300M

데이터베이스 환경 설계서

데이터베이스 생성시 반영해야 할 각종 환경에 대한 값을 설계한 내용이다.

DB_NAME		SABAN
SID		SABAN
		SABAN01/SABAN01
SGA		20M(20%)
MAX DATAFILES		20
CHARACTER SET/NATIONAL CHARACTER SET		KO16KSC5601/KO16KSC5601
DB_BLOCK_SIZE		8K
ROLLBACK_SEGMENTS		R01, R02, R03, R04, R05
DB_FILES		100
DB_FILE_MULTIBLOCK_READ_COUNT		8
DB_BLOCK_BUFFERS		1500(SGA 60%)
SHARED_POOL_SIZE		200000
LOG_BUFFER		50000
LOG_CHECKPOINT_INTERVAL		10000
PROCESSES		20
SESSIONS		10
DML_LOCKS		60
SEQUENCE_CACHE_ENTRIES		10
SEQUENCE_CACHE_HASH_BUCKETS		7
OPEN_CURSORS		100
ENQUEUE_RESOURCES		200
OPTIMIZER_MODE		RULE
SORT_AREA_SIZE		100000
SORT_AREA_RETAINED_SIZE		100000
LOG_SIMULTANEOUS_COPIES		4
LOG_SMALL_ENTRY_MAX_SIZE		80
SYSTEM TABLESPACE (SYSTEM)		SYSTEMSABAN.DBF(100M)
TEMPORARY TABLESPACE (TEMP)		TEMPSABAN.DBF(30M)
RBS TABLESPACE (RBS)		RBSSABAN(50M)
BIG RBS TABLESPACE (BIGRBS)		BIG_ROLL(30M)
TOOLS		TOOLSSABAN(50M)
USERS		USERSSABAN(50M)
		CONTROL1BONSA.CTL(3M) CONTROL2BONSA.CTL(3M)
		REDO11BONSA.LOG REDO12BONSA.LOG REDO21BONSA.LOG REDO22BONSA.LOG
DB		-
Init.ora		initBONSA.ora
Config.ora		configBONSA.ora

데이터베이스 조회 테스트

데이터베이스 기본 내용 조회

데이터베이스이름, 생성일자, 백업모드를 조회

```
PROMPT *****
PROMPT  V$DATABASE
PROMPT *****

COL NAME FORMAT A8
COL CHECK_CHNG FORMAT 9,999,999,999,999
COL ARCH_CHNG FORMAT 9,999,999,999,999

SELECT NAME,CREATED,LOG_MODE,CHECKPOINT_CHANGE# CHECK_CHNG,
        ARCHIVE_CHANGE# ARCH_CHNG
FROM V$DATABASE;
```

데이터베이스 SGA 크기를 조회

```
COLUMN NAME          FORMAT A20 HEADING 'SGA SEGMENT'
COLUMN VALUE         FORMAT 99,999,990 HEADING 'SIZE|(BYTES)'
COLUMN KVAL          FORMAT 99,990.9  HEADING 'SIZE|(KB)'
BREAK ON REPORT
COMPUTE SUM OF VALUE KVAL ON REPORT

SELECT NAME,VALUE,ROUND(VALUE/1024,1) KVAL
FROM V$SGA;
```

데이터베이스 전체 파라미터 조회

```
SHOW PARAMETER;
```

세션 및 트랜잭션 조회

데이터베이스에 접속되어 있는 OS 사용자 및 프로그램 조회

```

SET LINESIZE 110

COL SID FOR 9999
COL SERIAL# FOR 9999999
COL STATUS FOR A8
COL OSUSER FORMAT A15
COL USERNAME FORMAT A12
COL PROGRAM FORMAT A14
COL LOGON_TIME FORMAT A16

SELECT SID, SERIAL#, STATUS, TADDR, OSUSER, USERNAME, PROGRAM,
TO_CHAR(LOGON_TIME,'YYYY/MM/DD HH:MI') LOGON_TIME
FROM V$SESSION
WHERE TYPE != 'BACKGROUND'
AND STATUS = 'ACTIVE'
/
    
```

작업 중인 데이터베이스 트랜잭션 조회

```

COL SID          FOR 99999
COL SERIAL#      FOR 999999
COL OSUSER       FOR A9
COL TERMINAL     FOR A15
COL DBUSER       FOR A12
COL TRANSACTION FOR A8
COL SESSION      FOR A8
COL ROLLBACK     FOR A10
COL START_TIME  FOR A17

SELECT S.SID, S.SERIAL#, S.STATUS "SESSION", S.OSUSER, S.TERMINAL,
S.USERNAME "DBUSER", R.SEGMENT_NAME "ROLLBACK",T.STATUS
"TRANSACTION",S.TADDR,
T.USED_UBLK,T.USED_UREC,T.START_TIME
FROM V$SESSION S, V$TRANSACTION T, DBA_ROLLBACK_SEGS R
WHERE S.TADDR = T.ADDR
AND T.XIDUSN = R.SEGMENT_ID
/
    
```

실행된 SQL 문장 조회

```

SET VERIFY OFF LINESIZE 120

ACCEPT PGM_PROCS CHAR DEFAULT '0' PROMPT ENTER PROGRAM PROCESS ID
:

COL TODAY FORMAT A20 NEW_VALUE CURR_TIME
COL TODAY NOPRINT
SELECT TO_CHAR(SYSDATE,'YYYY/MM/DD HH24:MI:SS' ) TODAY FROM DUAL;

PROMPT -----
PROMPT == PROCESS/SQL STATUS ==
PROMPT => DATE : [ &CURR_TIME ]
PROMPT -----

COL PGM_PS    FORMAT A9
COL SID       FORMAT 999
COL SERIAL#   FORMAT 99999
COL MACHINE   FORMAT A10
COL DB_OBJECT FORMAT A22
COL SQL_TEXT  FORMAT A40
COL ROW_X     FORMAT 99,999

BREAK ON PGM_PS ON PROGRAM ON SID
SELECT B.PROCESS AS PGM_PS ,
       B.SID, SUBSTR(MACHINE,1,10) AS MACHINE,    ROWS_PROCESSED AS
ROW_X,
       C.SQL_TEXT AS SQL_TEXT
FROM V$SESSION B, V$SQL C
WHERE ( B.SQL_ADDRESS = C.ADDRESS OR
       B.PREV_SQL_ADDR = C.ADDRESS )
AND B.USERNAME IS NOT NULL
AND C.SQL_TEXT NOT LIKE 'ALTER SESSION SET %'
AND C.SQL_TEXT NOT LIKE 'SELECT VALUE$ FROM PROPS$ %'
AND B.PROCESS = DECODE(&PGM_PROCS,'0',B.PROCESS,&PGM_PROCS)
ORDER BY MACHINE ;

```

테이블 스페이스 및 데이터파일 조회

데이터파일과 컨트롤파일 로그파일을 조회

```
SET VERIFY OFF LINESIZE 120 PAGESIZE 100
COL TODAY FORMAT A20 NEW_VALUE CURR_TIME
COL TODAY NOPRINT
SELECT TO_CHAR(SYSDATE,'YYYY/MM/DD HH24:MI:SS' ) TODAY FROM DUAL;

PROMPT =====
PROMPT == CONTROL AND LOG FILE AND DATA FILE ==
PROMPT => DATE : [ &CURR_TIME ]
PROMPT =====

COL DB FORMAT A5
COL LOG_DB FORMAT A10
COL PHY_DB FORMAT A50
COL SZ_KB FORMAT 9,999,999
SELECT 'DATA' AS DB, TABLESPACE_NAME AS LOG_DB,
       FILE_NAME AS PHY_DB, BYTES/1024 AS SZ_KB
FROM DBA_DATA_FILES
UNION ALL
SELECT 'LOG ' AS DB, TO_CHAR(V1.GROUP#) AS LOG_DB,
       V1.MEMBER AS PHY_DB, V2.BYTES/1024 AS SZ_KB
FROM V$LOGFILE V1, V$LOG V2
WHERE V1.GROUP# = V2.GROUP#
UNION ALL
SELECT 'CTL ' AS DB, 'CONTROL' AS LOG_DB,
       NAME AS PHY_DB , 2 AS SZ_KB
FROM V$CONTROLFILE;
```

데이터파일이 차지하고 있는 디스크의 전체 크기와 현재 사용되는 크기를 조회

```
SET VERIFY OFF LINESIZE 120 PAGESIZE 100
COL TODAY FORMAT A20 NEW_VALUE CURR_TIME
COL TODAY NOPRINT
SELECT TO_CHAR(SYSDATE,'YYYY/MM/DD HH24:MI:SS' ) TODAY FROM DUAL;

PROMPT =====
PROMPT == FREE SPACE USAGE PCT% FOR EACH TABLESPACE AND DATAFILE
PROMPT => DATE : [ &CURR_TIME ]
PROMPT =====

COL TABLESPACE_NAME FORMAT A10
COL FILE_NAME FORMAT A30
COL FREE_KB FORMAT 999,999,999
COL TOT_KB FORMAT 999,999,999
COL USAGE_P FORMAT 999.9
SELECT A.TABLESPACE_NAME, A.FILE_NAME,
       SUM(A.BYTES)/1024 AS TOT_KB,
       NVL(SUM(SZ_KB),0) AS FREE_KB,
       NVL(( 1. - SUM(SZ_KB)/(SUM(A.BYTES)/1024.) ) * 100,100) AS USAGE_P
FROM DBA_DATA_FILES A,
     ( SELECT FILE_ID , SUM(BYTES)/1024. AS SZ_KB
       FROM DBA_FREE_SPACE
       GROUP BY FILE_ID ) B
WHERE A.FILE_ID = B.FILE_ID (+ )
GROUP BY A.TABLESPACE_NAME,A.FILE_NAME;
```

데이터파일별 물리적인 I/O 현황을 조회

```
PROMPT *****
PROMPT *                CONTENTS : 1. DATAFILE별 PHYSICAL I/O 현황
*
PROMPT *****
PROMPT
COLUMN T0 FORMAT A10 HEADING 'TABLESPACE'
COLUMN T1 FORMAT A17 HEADING 'FILENAME'
COLUMN T2 FORMAT 9,999,999 HEADING 'PHYS_READ'
COLUMN T3 FORMAT 99.9 HEADING '% READ'
COLUMN T4 FORMAT 999,999 HEADING 'PHYS_WRITES'
COLUMN T5 FORMAT 99.9 HEADING '% WRITE'
COLUMN T6 FORMAT 99,999 HEADING 'SIZE(MB)'

SELECT F.TABLESPACE_NAME T0,F.FILE_NAME T1,
       X.PHYRDS T2,
       TO_CHAR(TRUNC((PHYRDS/TOT_RDS)*100,2),99.99)||' %' T3,
       X.PHYWRTS T4,
       TO_CHAR(TRUNC((PHYWRTS/TOT_WRTS)*100,2),99.99)||'% ' T5,
       ROUND((I.BYTES/1024)/1024) T6
FROM V$FILESTAT X, DBA_DATA_FILES F, V$DATAFILE I,
     (SELECT SUM(PHYRDS) TOT_RDS,SUM(PHYWRTS) TOT_WRTS FROM
V$FILESTAT) TOT_IO
WHERE X.FILE# = F.FILE_ID
      AND F.FILE_ID = I.FILE#
ORDER BY F.TABLESPACE_NAME;
```

잠금(LOCK) 정보조회

잠금 발생 유형 조회

```
COLUMN USERNAME FORMAT A10
COLUMN SID FORMAT 999
COLUMN LOCK_TYPE FORMAT A25
COLUMN MODE_HELD FORMAT A11
COLUMN MODE_REQUESTED FORMAT A10
COLUMN LOCK_ID1 FORMAT A8
COLUMN LOCK_ID2 FORMAT A8
COL   BLOCKING_OTHERS FOR A13

SELECT A.SID,
       DECODE(A.TYPE,
MR', 'MEDIA RECOVERY',
RT', 'REDO THREAD',
UN', 'USER NAME',
TX', 'TRANSACTION',
TM', 'DML',
UL', 'PL/SQL USER LOCK',
DX', 'DISTRIBUTED XACTION',
CF', 'CONTROL FILE',
IS', 'INSTANCE STATE',
FS', 'FILE SET',
IR', 'INSTANCE RECOVERY',
ST', 'DISK SPACE TRANSACTION',
IR', 'INSTANCE RECOVERY',
ST', 'DISK SPACE TRANSACTION',
TS', 'TEMP SEGMENT',
IV', 'LIBRARY CACHE INVALIDATION',
LS', 'LOG START OR SWITCH',
RW', 'ROW WAIT',
SQ', 'SEQUENCE NUMBER',
TE', 'EXTEND TABLE',
TT', 'TEMP TABLE',
A.TYPE) LOCK_TYPE,
       DECODE(A.LMODE,
0, 'NONE',           /* MON LOCK EQUIVALENT */
1, 'NULL',          /* N */
```

```

2, 'ROW-S (SS)',      /* L */
3, 'ROW-X (SX)',     /* R */
3, 'ROW-X (SX)',     /* R */
4, 'SHARE',         /* S */
5, 'S/ROW-X (SSX)', /* C */
6, 'EXCLUSIVE',     /* X */
TO_CHAR(A.LMODE)) MODE_HELD,
DECODE(A.REQUEST,
0, 'NONE',          /* MON LOCK EQUIVALENT */
1, 'NULL',         /* N */
2, 'ROW-S (SS)',   /* L */
3, 'ROW-X (SX)',   /* R */
4, 'SHARE',        /* S */
5, 'S/ROW-X (SSX)', /* C */
6, 'EXCLUSIVE',    /* X */
TO_CHAR(A.REQUEST)) MODE_REQUESTED,
TO_CHAR(A.ID1) LOCK_ID1, TO_CHAR(A.ID2) LOCK_ID2,
DECODE(BLOCK,0, NOT BLOCKING', /* NOT BLOCKING ANY OTHER
PROCESSES */
1, BLOCKING', /* THIS LOCK BLOCKS OTHER PROCESSES */
2, 'GLOBAL', /* THIS LOCK IS GLOBAL, SO WE CAN'T
TELL */
TO_CHAR(BLOCK)) BLOCKING_OTHERS
FROM V$LOCK A
WHERE (ID1,ID2) IN
(SELECT B.ID1, B.ID2 FROM V$LOCK B WHERE B.ID1=A.ID1 AND
B.ID2=A.ID2 AND REQUEST > 0);

```

롤백세그먼트 조회

롤백세그먼트 현황 조회

```

COL RBS_SEG FORMAT A10
COL TBS_NM FORMAT A10
COL STATUS FORMAT A10
COL INI_KB FORMAT 9,999,999
COL NXT_KB FORMAT 9,999,999
COL TOT_KB FORMAT 9,999,999
COL XACTS FORMAT 99999
COL EXTENTS FORMAT 9999
COL USN FOR 99999

SELECT A.SEGMENT_NAME RBS_SEG, C.USN, B.STATUS, C.XACTS,
       ROUND(A.INITIAL_EXTENT/1024,0) INI_KB,
       ROUND(A.NEXT_EXTENT/1024,0) NXT_KB,
       A.EXTENTS, ROUND(A.BYTES/1024,0) TOT_KB, A.MAX_EXTENTS,
       A.TABLESPACE_NAME TBS_NM
FROM DBA_SEGMENTS A, DBA_ROLLBACK_SEGS B, V$ROLLSTAT C
WHERE A.SEGMENT_NAME = B.SEGMENT_NAME
AND B.SEGMENT_ID = C.USN(+)
AND A.SEGMENT_TYPE = 'ROLLBACK';
    
```

롤백세그먼트 경합내용 조회

```

COLUMN T0 FORMAT A10 HEADING 'ROLLBACK'
COLUMN T1 FORMAT 9,999,999 HEADING 'TRANS|TBL_GETS'
COLUMN T2 FORMAT 9,999,999 HEADING 'TRANS|TBL_WAITS'
COLUMN T3 FORMAT 999,999 HEADING 'MISS RATIO'
COLUMN T4 FORMAT A10 HEADING 'SIZE(KB)'
COLUMN T5 FORMAT 999,999 HEADING 'SHRINKS'
COLUMN T6 FORMAT 999,999 HEADING 'EXTENDS'
SELECT NAME T0,
       GETS T1,
       WAITS T2,
       TO_CHAR(TRUNC(WAITS/GETS*100, 2),099.99)||'%' T3,
       TO_CHAR(ROUND(RSSIZE/1024)) T4,
       SHRINKS T5,
       EXTENDS T6
FROM V$ROLLSTAT, V$ROLLNAME
WHERE V$ROLLSTAT.USN = V$ROLLNAME.USN;
    
```

데이터베이스 오브젝트 조회

테이블에 따른 스토리지 및 테이블스페이스 조회

```
COL SEGMENT_NAME FORMAT A15
COL TABLESPACE_NAME FORMAT A10

SELECT SEGMENT_NAME,ROUND(INITIAL_EXTENT/1024,0) INI_KB,
ROUND(NEXT_EXTENT/1024,0) NXT_KB, ROUND(BYTES/1024,0) TOT_KB,
PCT_INCREASE PCT, EXTENTS, TABLESPACE_NAME FROM DBA_SEGMENTS
WHERE SEGMENT_TYPE = 'TABLE' AND EXTENTS >= NVL(&EXTENT_NUM,1)
/
```

테이블에 걸쳐있는 인덱스를 조회

```
SET HEAD OFF
SET FEEDBACK OFF
SET LINESIZE 200
SPOOL INDEX.LST

SELECT 'SET HEAD OFF' FROM DUAL;
SELECT 'SET FEEDBACK OFF' FROM DUAL;

SELECT
  A.TABLE_NAME,B.INDEX_NAME,C.PARTITION_NAME,C.TABLESPACE_NAME,
  ROUND(C.BYTES/1024/1024,0),C.EXTENTS,
  ROUND(C.INITIAL_EXTENT/1024/1024,0),ROUND(C.NEXT_EXTENT/1024/1024,0)
FROM USER_TABLES A,USER_INDEXES B,USER_SEGMENTS C
WHERE A.TABLE_NAME=B.TABLE_NAME(+ )
  AND B.INDEX_NAME=C.SEGMENT_NAME(+ )
ORDER BY TABLE_NAME,INDEX_NAME,PARTITION_NAME;

SELECT 'SPOOL OFF' FROM DUAL;
SELECT 'SET HEAD ON' FROM DUAL;
SELECT 'SET FEEDBACK ON' FROM DUAL;

SPOOL OFF
SET HEAD ON
SET FEEDBACK ON
```

인덱스에 대한 컬럼 조회

```
BREAK ON OWNER ON TABLE_NAME ON INDEX_NAME
COL OWNER FORMAT A15
COL INDEX_NAME FORMAT A20
COL COLUMN_NAME FORMAT A20
COL P FORMAT 999

SELECT TABLE_NAME, INDEX_NAME, COLUMN_POSITION P, COLUMN_NAME
FROM USER_IND_COLUMNS
ORDER BY TABLE_NAME, INDEX_NAME, COLUMN_POSITION;
```

테이블에 걸려있는 FOREIGN KEY 조회

```
PROMPT *****
PROMPT FK COLUMN
PROMPT *****

BREAK ON TABLE_NAME ON CONSTRAINT_NAME ON R_CONSTRAINT_NAME ON
STATUS
COL TABLE_NAME FOR A15
COL CONSTRAINT_NAME FOR A20
COL R_CONSTRAINT_NAME FOR A15
COL COLUMN_NAME FOR A25
COL STATUS FOR A10

SELECT  A.TABLE_NAME,  A.CONSTRAINT_NAME,  A.R_CONSTRAINT_NAME,
A.STATUS,
        B.POSITION, B.COLUMN_NAME
FROM DBA_CONSTRAINTS A, DBA_CONS_COLUMNS B
WHERE A.CONSTRAINT_TYPE='R'
      AND A.CONSTRAINT_NAME = B.CONSTRAINT_NAME
      AND A.TABLE_NAME = UPPER('&TABLE_NAME')
ORDER BY CONSTRAINT_NAME, POSITION ;
```

제약사항을 조회

```

BREAK ON OWNER
COL OWNER FORMAT A10
COL TABLE_NAME FORMAT A15
COL CONSTRAINT_NAME FORMAT A20
COL TYPE FORMAT A5
COL REF_TABLE FORMAT A15

SELECT A.OWNER, A.TABLE_NAME, A.CONSTRAINT_NAME,
       DECODE(A.CONSTRAINT_TYPE,R,'FK',
              'P','PK',
              'C','CHECK',
              NONE) TYPE,
       RTRIM(A.R_OWNER) || '.' || RTRIM(B.TABLE_NAME) REF_TABLE, A.STATUS
FROM   DBA_CONSTRAINTS A, DBA_CONSTRAINTS B
WHERE  A.CONSTRAINT_TYPE = 'R' AND
       A.TABLE_NAME LIKE UPPER('&FK_TABLE_NAME') || '%' AND
       B.TABLE_NAME LIKE UPPER('&PK_TABLE_NAME') || '%' AND
       A.R_CONSTRAINT_NAME = B.CONSTRAINT_NAME;
    
```

오브젝트 EXTENT 조회

```

SPOOL EXTENT.LST
COL SEGMENT_NAME FORMAT A17
COL TABLESPACE FORMAT A12
COL TYPE FORMAT A6
COL EXT FORMAT 99999

SELECT SEGMENT_NAME,SEGMENT_TYPE "TYPE",
       TABLESPACE_NAME "TABLESPACE",
       ROUND(BYTES/1024) "SZ(KB)",
       EXTENTS "EXT", ROUND(INITIAL_EXTENT/1024) "INI_EXT(KB)",
       ROUND(NEXT_EXTENT/1024) "NXT_EXT(KB)"
FROM   USER_SEGMENTS
WHERE  SEGMENT_TYPE IN ('TABLE','INDEX')
ORDER BY SEGMENT_TYPE,TABLESPACE_NAME, SEGMENT_NAME;
SPOOL OFF
    
```

뷰 조회

```
SPOOL VIEW.LST
BREAK ON OWNER
COL OWNER FORMAT A15
COL OBJECT_NAME FORMAT A30
SELECT OWNER, OBJECT_NAME, OBJECT_TYPE, STATUS FROM DBA_OBJECTS
WHERE OBJECT_TYPE = 'VIEW'
/
SPOOL OFF
```