**1장.**

**데이터 모델링의 이해**

## 작업순서ERD

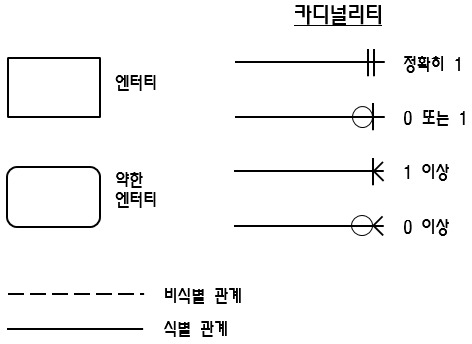
**데이터 모델링의 중요성 및 유의점**

* 중복 같: 은 시간 같은 데이터 제공
* 비유연성 사: 소한 업무변화에 데이터 모델이 수시

1. 엔터티 그림 엔2터.

4. 관계명 기술 관5계. 부

티 배치 엔터티3. 관계설정 의 참여도 기술 관계필6수. 여

로 변경되면 안됨 . 데이터 정의를 사용 프로세스와 분 리한다.

* 비일관성 데: 이터 간 상호 연관 관계에 대해 명확

히 정의해야 한다.

## 데이터 모델링

개념적, 논리적, 물리적 데이터 모델링

## 데이터 독립성 요소

외부 스키마 개: 개 사용자가 보는 개인적 스키D마B

개념 스키마 모: 든 사용자 관점을 통합한 전체 DB

내부 스키마 물: 리적 장치에서 데이터가 실제적 저장

## 데이터 독립성

논리적 독립성 개: 념스키마 변경 외,부스키마에 영향

X

물리적 독립성 내: 부스키마 변경 외,부 개념/ 스키마에

## 좋은 데이터 모델의 요소

1. 완전성 업: 무에 필요한 모든 데이터가 모델에 정 의
2. 중복배제 하: 나의 내에DB동일한 사실은 한번만 .
3. 업무규칙 많: 은 규칙을 사용자가 공유하도록 제공
4. 데이터 재사용 데: 이터가 독립적으로 설계돼야 함

영향X

1. 의사소통 업: 무규칙은 엔터티 서, 브타입 속, 성 관,

계 등의 형태로 최대한 자세히 표현

## 사상Ma: pping(

1. 통합성 동: 일한 데이터는 한 번만 정의 참조, 활

**)**상호 독립적인 개념을 연결시켜주는

다리 용

논리적 사상 외: 부 스키마 개념- 스키마 물리적 사상 개: 념 스키마 내부- 스키마

--------------------------------------------

------------

**엔터티**업: 무에 필요하고 유용한 정보를 저장하고 관

**데이터 모델링의 요소3** 어떤 것(Things) 성격(Attributes) 관계(Relationships)

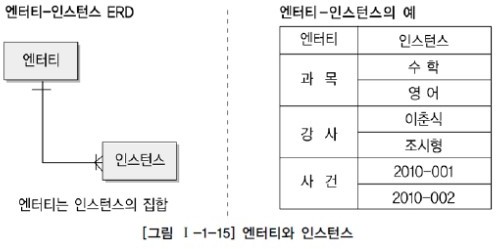
## 모델링의 특징

(현실세계) -> 추상화, 단순화, 정확화 모->델( )

## 데이터 모델 표기법

리하기 위한 집합적인 것

,보이지 않는 개념 포함

1976년 피터첸이 Entity Relationship Model 개발

* IE, Baker 기법이 많이 쓰임
* 엔터티,

관계, 속성으로 이뤄짐

## 엔터티의 특징

1. 반드시 해당 업무에서 필요하고 관리하고자 함
2. 유일한 식별자에 의해 식별 가능
3. 두 개 이상의 인스턴스의 집합
4. 업무 프로세스에 의해 이용되어야 함
5. 반드시 속성이 있어야 함
6. 다른 엔터티와 최소 개1 이상의 관계가 있어야 함

-> 통계성/코드성 엔터티는 관계 생략 가능

## 엔터티의 분류

**유무형에 따른 분류**유:형 개,념 사,건 엔터티

른 성능을 낼 수 있도록 원래 속성의 값을 계산 적을, 수록 좋음 합ex)

**도메인** 속: 성에 대한 데이터타입 크,기 제,약사항 지 정

## 속성의 명명

1. 해당업무에서 사용하는 이름 부여
2. 서술식 속성명은 사용 금지 약어 사용 금지3.

* 유형: 사물원리적 형태
* 개념: 조개직념적 정보

- 사건 업: 무 수행시 발생

물품 강사ex) ,

보험상품ex) ,

주문 청구ex)

1. ,구체적으로 명명하여 데이터 모델에서 유일성 확보

--------------------------------------------

----,------,--

미납

**발생시점에 따른 분류** 기: 본 키/ 중, 심 행, 위 엔터티

**관계**엔: 터티의 인스턴스 사이의 논리적인 연관성으 로서 존재의 형태로서나 행위로서 서로에게 연관성이 부여된 상태 관계 페어링의 집합,

- 기본 그:

업무에 원래 존재하는 정보 타 ,엔터티의

ex) 강사 가- 르친다 관(계

) 수- 강생

부모 역할 , 자신의 고유한 주식별자 가짐 사원ex) ,

부서

**패어링**엔:터티 안에 인스턴스가 개별적으로 관계를

- 중심 기: 본 엔터티로부터 발생 다른,

엔터티와의

가지는 것

관계로 많은 행위 엔터티 생성 계약 사고 주문ex) , ,

- 행위 개:

이2 상의 부모 엔터티로부터 발생 자주 ,

## UML(통합모델링언어)에서의 관계

바뀌거나 양이 증가 주e문x)목록 사원,변경이력

## 엔터티의 명명

현업업무에서 사용하는 용어 사용 약어 사용금지 단,

연관관계(실선) : 항상 이용하는 관계 소e속x)된다 의존관계(점선) : 상대 행위에 의해 발생하는 관계 주문한다.

,

.

ex)

수명사 사용 , 고유한 이름 사용

------------

,생성의미대로 부여

## 관계의 표기법

관계명 관계의 이름: 관계차수 : 1:1, 1:M, M:N

**속성**업:무에서 필요로 하는 인스턴스로 관리하고자 하는 의미상 분리되지 않는 최소의 데이터 단위

관계선택성(관계선택사양) : 필수관계,

선택관계

* 한 개의 엔터티는 개2 이상의 인스턴스 집합
* 한 개의 엔터티는 개2 이상의 속성을 가짐
* 한 개의 속성은 개1의 속성값을 가짐

## 관계 체크사항

* 1. 2개의 엔터티 사이에 관심있는 연관 규칙 O?
  2. 2개의 엔터티 사이에 정보의 조합 발생 O?
  3. 업무기술서, 장표에 관계연결에 대한 규칙 서술 O?

**구성 방식의 분류**일반 속성 : PK, FK,

**속성의 분류** 기: 본 설, 계 파, 생 속성

* 1. 업무기술서, O?

장표에 관계연결을 가능케 하는 동사

* 기본 업: 무로부터 추출한 모든 일반적인 속성
* 설계 업: 무를 규칙화하기 위해 새로 만들거나 변 형, 정의하는 속성 일ex련) 번호

--------------------------------------------

------------

**식별자**엔: 터티 내에서 인스턴스를 구분하는 구분자

* 파생 다: 른 속성에 영향을 받아 발생하는 속성 빠 , 식별자는 논리적 , Key는 물리적 데이터 모델링 단계

에 사용

**식별자의 특징**유:일성 최,소성 불,변성 존,재성

* 식별자 관계로만 설정 시 주식별자 증가로 오류 유 발
* 유일성 주: 식별자에 의해 모든 인스턴스들이 유일 **비식별자**부: 모 속성을 자식의 일반 속성으로 사용

하게 구분

* 최소성 주: 식별자를 구성하는 속성의 수는 유일성

1. 부모 없는 자식이 생성될 수 있는 경우
2. 부모와 자식의 생명주기가 다른 경우 별도(로 소

을 만족하는 최소의 수가 되어야 함 멸)

* 불변성 지: 정된 주식별자의 값은 자주 변하지 않아 3. 여러개의 엔터티가 하나의 엔터티로 통합되어 표현

야 함 . 변하면 이전 기록 말소됨

* 존재성 주: 식별자가 지정되면 반드시 값이 들어와

되었는데 각각의 엔터티가 별도의 관계를 가진 경우

1. 자식엔터티에 별도의 주식별자를 생성하는 것이 더

야 함 유리한 경우

1. SQL 문장이 길어져 복잡성 증가되는 것 방지

## 식별자 분류

**대표성여부**주: 식별자 보, 조식별자

* 약한 연결관계 표현 , 점선 표기
* 비식별자 관계로만 설정 시 부모 엔터티와 조인하
* 주 엔: 터티 내에서 각 어커런스를 구분할 수 있는 구분자, 타 엔터티와 참조관계를 연결 O

여 성능 저하

============================================

* 보조 구: 분자이나 대표성 X참, 조관계 연결 X ==

**스스로생성여부**내: 부식별자 외, 부식별자 내부 스: 스로 생성되는 식별자

**2장.**

**데이터 모델과 성능**

외부 타: 엔터티로부터 받아오는 식별자

**속성의 수**단:일식별자 복,합식별자 단일 하: 나의 속성으로 구성

복합 개: 2이상의 속성으로 구성

**성능 데이터 모델링**성능: 향DB상을 목적으로 설계

단계의 데이터 모델링 때부터 정규화 반정규화 테이, ,

블통합, 테이블분할, 조인구조, PK, FK 등 여러 가지

**대체 여부**본:질식별자 인,조식별자 본질 업: 무에 의해 만들어지는 식별자 인조 인: 위적으로 만든 식별자

## 주식별자 도출기준

1. 해당 업무에서 자주 이용되는 속성임
2. 명칭, 내역 등과 같이 이름으로 기술되는 것들은 X
3. 복합으로 주식별자로 구성할 경우 너무 많은 속성

X

성능과 관련된 사항이 데이터 모델링에 반영될 수 있 도록 하는 것

분석/설계 단계에서 데이터 모델에 성능을 고려한 데 이터 모델링을 수행할 경우 성능저하에 따른 재업무 비용을 최소화 할 수 있음.

데이터의 증가가 빠를수록 성능저하에 따른 성능개선 비용은 기하급수적으로 증가하게 된다.

## 성능 데이터 모델링 고려사항 순서

-> 너무 많으면 인조식별자 생성한다 .

1. 데이터 모델링을 할 때 정규화를 정확하게 수행
2. DB 용량산정을 수행한다 .

## 식별자 관계

**주식별자**자: 식의 주식별자로 부모의 주식별자 상속

1. DB에 발생되는 트랜잭션의 유형을 파악한다 .
2. 용량과 트랜잭션의 유형에 따라 반정규화를 수행

* 부모로부터 받은 식별자를 자식엔터티의 주식별자

1. 이력모델의 조정 , PK/FK조정 , 슈퍼 /서브타입 조정

로 이용하는 경우

* 강한 연결관계 표현 , 실선 표기

1. 성능관점에서 데이터 모델을 검증한다 .

**함수적 종속성**데:이터들이 어떤 기준 값에 의해 종 속되는 현상

인덱스를 조정함 조(회가 대부분일 때 클러스터링 적 용)

3. 파티셔닝 대: 량의 데이터는 의 성PK격에 따라 부

**정규화(-**:**)** 반복적인 데이터를 분리하고 각 데이터

가 종속된 테이블에 적절하게 배치되도록 하는 것

분적인 테이블로 분리할 수 있다.

물리적 저장공간 분리

파티셔닝 키에 의해

* 1차 정규화 같: 은 성격 내,용 컬럼이 연속될 때 컬 4. 캐시 응: 용 애플리케이션에서 로직을 구사하는 방

럼 제거 테이블 생성,

* 2차 정규화 복: 합PK키 구성일 때 부분적 함수 종 속 관계 테이블 분리
* 3차 정규화 가: 아PK닌 일반 컬럼에 의존하는 컬 럼 분리

법을 변경함으로써 성능을 향상시킬 수 있다.

## 반정규화 적용3. 테이블 반정규화3-1.

**테이블 병합 (1:1관계 , 1:M관계 , 슈퍼 /서브타입 )**

1. 1:1관계를 통합하여 성능향상

**반정규화(+**:**)** 정규화된 엔터티

, 속성

, 관계에 대해

1. 1:M관계를 통합하여 성능향상

시스템의 성능향상과 개발과 운영의 단순화를 위해 3. 슈퍼/서브 관계를 통합하여 성능향상

중복, 법

통합,

분리 등을 수행하는 데이터 모델링의 기

## 테이블분할 수직분할 수평분할( , )

조회 시 디스크 가I/많O거나 경로가 멀어 조인에 의 한 성능 저하를 막기 위해 수행

1. 수직분할 칼: 럼단위 테이블을 디스크 를 분I산/O 처리하기 위해 테이블을 로1:분1 리하여 성능향상
2. 수평분할 로: 우단위로 집중 발생되는 트랜잭션을

일반적으로 정규화시 입력 수/ 정 삭/ 제 성능이 향상되 며 반정규화시 조인 성능이 향상된다.

분석하여 디스크 및I/데O이터 접근의 효율성을 높

여 성능을 향상하기 위해 로우단위로 테이블을 쪼갬

## 반정규화 절차

**1.** 범**반**위**정**처**규**리**화**빈**대**도**상**수**조사**

( , 범위, 통계성)

## 테이블 추가

**중복1.**: 다른 업무이거나 서버가 다른 경우 동일한 테이블구조를 중복하여 원격조인을 제거하여 성능 향

1. 자주 사용되는 테이블에 접근하는 프로세스의 수가 상

많고 항상 일정한 범위만을 조회하는 경우

1. 테이블에 대량의 데이터가 있고 대량의 데이터 범 위를 자주 처리하는 경우에 처리범위를 일정하게 줄 이지 않으면 성능을 보장할 수 없는 경우

**통계2**:**.** 등SU을M미, 리AV수G행하여 계산해 둠 으로써 조회 시 성능을 향상

**이력3**:**.** 이력테이블 중에서 마스터 테이블에 존재하 는 레코드를 중복하여 이력테이블에 존재시켜 성능

1. 통계성 프로세스에 의해 통계 정보를 필요로 할 때 향상

별도의 통계테이블을 생성한다.

1. 테이블에 지나치게 많은 조인이 걸려 데이터를 조

**부분4**:**.** 하나의 테이블의 전체 칼럼 중 자주 이용하 는 집중화된 칼럼들이 있을 때 디스I/크O 를 줄이기

회하는 작업이 기술적으로 어려울 경우

위해 해당 칼럼들을 모아놓은 별도의 반정규화된 테

이블을 생성

**2.** 뷰**다른 방법유도 검토** ( , 클러스터링, 인덱스 조정 )

1. VIEW 사용 지: 나치게 많은 조인이 걸려 데이터를

조회하는 작업이 기술적으로 어려울 경우 를 사VIEW 용한다 가 성능향상. (VIEW X)

1. 클러스터링 대: 량의 데이터처리나 부분처리에 의

## 칼럼 반정규화3-2.

**중복칼럼 추가1.**:조인에 의해 처리할 때 성능저하 를 예방하기 위해 중복된 칼럼을 위치시킴

**파생칼럼 추가2.**: 트랜잭션이 처리되는 시점에 계산

해 성능이 저하되는 경우 클러스터링을 적용하거나 에 의해 발생되는 성능저하를 예방하기 위해 미리 값

을 계산하여 칼럼에 보관

## 이력테이블 칼럼추가

: 대량의 이력데이터를 처리

해시 알고리즘이 적용되어 테이블이 분리

할 때 불특정 날 조회나 최근 값을 조회할 때 나타날 수 있는 성능저하를 예방하기 위해 이력테이블에 기

## 테이블에 대한 수평 수직분할의 절차/

1. 데이터 모델링을 완성한다 .

능성 칼럼 (최근값 여부 , 시작과 종료일자 등 을)

추가함

1. DB 용량산정을 한다 .

## 4. 응용시스템 오작동을 위한 칼럼 추가

업무적으로:

3. 대량 데이터가 처리되는 테이블에 대해 트랜잭션

는 의미가 없지만 사용자의 실수로 원래 값으로 복구

하기 원하는 경우 이전 데이터를 임시적으로 중복하 여 보관하는 기법

**5. PK**단**에**일**의**안**한**에**컬**서**럼**특**추**정**가**값을: PK

별도로 조회하는 경우 성능 저하 발생할 수 있어 일 반속성으로 추가함

처리 패턴을 분석한다.

1. 칼럼 단위로 집중화된 처리가 발생하는지 로, 우 단 위로 집중화된 처리가 발생하는지 분석하여 집중화된 단위로 테이블을 분리하는 것을 검토한다.

* 컬럼 많음 분->리1:1
* 데이터 많음 파->티셔닝

## 3-3. 관계 반정규화 ; 무결성 유지

**중복관계 추가**데:이터를 처리하기 위한 여러 경로를

------------

**슈퍼 서브 타입 모델/**업: 무를 구성하는 데이터를 공

거쳐 조인이 가능하지만 이 때 발생할 수 있는 성능

저하를 예방하기 위해 추가적인 관계를 맺는 방법

통과 차이점의 특징을 고려하여 효과적 표현.

모델

논리적

------------

**로우 체이닝**로:우의 길이가 너무 길어서 데이터 블 록 하나에 데이터가 모두 저장되지 않고 두 개 이상 의 블록에 걸쳐 하나의 로우가 저장되어 있는 형태

* 슈퍼 타입 공: 통 부분
* 서브 타입 공: 통으로부터 상속받아 다른 엔터티와 차이가 있는 속성

## 슈퍼/서브 타입 데이터 모델의 변환기술

* 1. 개별로 발생되는 트랜잭션에 대해서는 개별 테이블

**로우 마이그레이션**데:이터블록에서 수정이 발생하면 수정된 데이터를 해당 데이터 블록에서 저장하지 못

로 구성 (OneToOne Type)

* 1. 슈퍼타입+서브타입에 대해 발생되는 트랜잭션에 대

하고 다른 블록의 빈 공간을 찾아 저장하는 방식

해서는 슈퍼 +서브타입 테이블로 구성

(Plus Type)

* 1. 전체를 하나로 묶어 트랜잭션이 발생할 때는 하나

로우 체이닝과 로우 마이그레이션이 발생하여 많은

블록에 데이터가 저장되면 메모D리B에서 디스크

I/O가 발생할 때 많은 가I/발O생하여 성능저하 발생 트랜잭션을 분석하여 적절하게 관계로 분리함으로1:1 써 성능향상이 가능하도록 해야 한다.

의 테이블로 구성 (Single Type, All in One Type)

**인덱스 특성을 고려한 성능향상PK/FK DB** 인덱스의 특징은 여러 개의 속성이 하나의 인덱스로 구성되어 있을 때 앞쪽에 위치한 속성의 값이 비교자 로 있어야 좋은 효율을 나타낸다.

## PK에 의해 테이블을 분할하는 방법 파( 티셔닝 )

앞쪽에 위치한 속성의 값이 가급‘=적’

아니면 최소한

1. **RANGE PARTITION**대:상 테이블이 날짜 또는 범위 가‘BE들T어WEE와야N’효‘<율>적’ 이다 .

숫자값으로 분리가 가능하고 각 영역별로 트랜잭션이 분리되는 경우 요금ex) \_0401

1. **LIST PARTITION**지: 점 사, 업소 등 핵심적인 코 드값으로 가PK구성되어 있고 대량의 데이터가 있는

--------------------------------------------

------------

## 분산 DB

1. 여러 곳으로 분산되어있는 를D하B 나의 가상 시스

테이블의 경우 고객 서울ex) \_ 템으로 사용할 수 있도록 한 DB

**3. HASH PARTITION**지:정된 조건H에A따SH라

2. 논리적으로 동일한 시스템에 속하지만 컴,

퓨터 네

트워크를 통해 물리적으로 분산되어 있는 데이터집합

**테이블 요약 분산4.**지:

역 간에 또는 서버 간에 데이

터가 비슷하지만 서로 다른 유형으로 존재하는 경우

## 분산 를DB만족하기 위한 가지 6투명성

**분석요약-**동:

일한 테이블 구조를 가지고 있으면서

## 분할 투명성 단(

**편화 )**하:나의 논리적 이

Relation

분산되어 있는 동일한 내용의 데이터를 이용하여 통

여러 단편으로 분할되어 각 사본이 여러 에 저장site

1. 사**위**용**치**하**투**려**명**는**성**데이: 터의 저장 장소 명

시 불필요 , 위치정보가 시스템 카탈로그에 유지

합된 데이터를 산출하는 방식 판매실적 지사, ex)

지사B

**-통**분산**합요**되어**약** 있: 는 다른 내용의 데이터를 이

**지역사상 투명성3.**:지역 와D물BM리S적 사이 DB

용하여 통합된 데이터를 산출하는 방ex식)

판매실적

의 보장Mapping

1. 객**중**체**복**가**투**여**명**러**성**에 :중D복B되어 있는지 알 필요가 없는 성질
2. 구**장**성**애**요**투**소**명**의**성**장애: 에 무관한 트랜잭션

stie

지사 제품 지사 제품A:C , B:D

## 분산 설DB계를 고려해야 하는 경우

1. 성능이 중요한 사이트

의 원자성 유지

1. 공통코드,

기준정보, 마스터 데이터의 성능향상

**6.** 다**병**수**행**트**투**랜**명**잭**성**션 동:

시 수행시 결과의

1. 실시간 동기화가 요구되지 않는 경우

, Near Real

일관성 유지 , TimeStamp, 분산 단2계 이용Locking

## 분산 장 단점DB -

**장점**지:역 자치성 신,뢰성 가용성 효용, 성 융통성 ,

빠른 응답속도 , 비용절감 , 각 지역 사용자 요구 수용

Time 특징을 가지고 있는 경우

1. 특정 서버에 부하가 집중되어 부하를 분산
2. 백업 사이트 구성하는 경우

= = = = = = = = = = = = = = = = = = = = = 2 과 목

=====================

**단점** 비: 용증가 오, 류의 잠재성 증대 설계,

관리의

# 장 기본1 . SQL

복잡성,

불규칙한 응답 속도

,통제의 어려움 데, 이터

무결성 위협

## 분산 적용 기법DB

**테이블 위치 분산1.** :설계된 테이블을 본사와 지사

**DB**특:정 기업이나 조직 또는 개인이 필요에 의해 데이터를 일정한 형태로 저장해 놓은 것을 의미한다.

**DBMS**효:율적인 데이터 관리 뿐만 아니라 예기치

단위로 분산 위치별 문서 필요, DB

못한 사건으로 인한 데이터의 손상을 피하고, 필요시

**테이블 분할 분산2.**:각각의 테이블을 쪼개어 분산

필요한 데이터를 복구하기 위한 강력한 기능의 SW

- 수평분할 로: 우 단위로 분리 지,사별로 다를때 중,

**SQL**관: 계형 에서D데B 이터 정의 조작

,제어를, 위

복X 해 사용하는 언어

- 수직분할 칼: 럼 단위로 분리 각 ,테이블에 동일

있어야 함PK .

**테이블 복제 분산3.** :동일한 테이블을 다른 지역이

* **DML** : SELECT, INSERT, UPDATE, DELETE
* **DDL** : CREATE, ALTER, DROP, RENAME
* **DCL** : GRANT, REVOKE
* **TCL** : COMMIT, ROLLBACK

나 서버에서 동시에 생성하여 관리하는 유형

**부분복제-**마: 스터 에서DB테이블의 일부의 내용만

**테이블**기: 본DB단위 데이터,를 저장하는 객체

다른 지역이나 서버에 위치

**광역복제-**:마스터 테이DB블의 내용을 각 지역이나 서버에 존재

* 가로 행= 로우= 튜플 인=스턴스= 세로 열 컬럼- = =

**정규화**데:이터의 정합성 확보와 데이터 입력 수정 / /

삭제시 발생할 수 있는 이상현상을 방지하기 위해 중

* 추가

: **ALTER TABLE** PLAYER **ADD**(ADDRESS

복 제거 VARCHAR2(80));

- 삭제 : **ALTER TABLE** PLAYER **DROP COLUMN**

**기본키**테:이블에 존재하는 각 행을 한 가지 의미로 ADDRESS;

특정할 수 있는 한 개 이상의 칼럼

* 수정

: **ALTER TABLE** TEAM\_TEMP **MODIFY**

**외부키**다: 른 테이블의 기본키로 사용되고 있는 관계 를 연결하는 칼럼

--------------------------------------------

------------

## [DDL]

**데이터 유형**

(ORIG\_YYYY VARCHAR2(8) **DEFAULT** '20020129' NOT NULL);

제약조건 삭제 : **DROP CONSTRAINT**조건명 ; 제약조건 추가 : **ADD CONSTRAINT**조건명 조건 칼럼명( );

테이블명 변경 : RENAME PLAYER TO PLAYER\_BACKUP;

**- CHAR(s)**고: 정 길이 문자열 정보 최대.

길이 만큼

테이블 삭제 : **DROP TABLE** PLAYER;

공간 채움 ‘AA’ = ‘AA ’

* **VARCHAR2(s)**가:변 길이 문자열 정보 할당.된 변수 값의 바이트만 적용 ‘AA’ != ‘AA ’
* **NUMBER**정: 수 실, 수 등 숫자 정보
* **DATE**날짜와 시각 정보 :

테이블 데이터 삭제 : **TRUNCATE TABLE** PLAYER; 컬럼명 변경 : **RENAME COLUMN** TEAM\_ID **TO** T\_ID;

--------------------------------------------

------------

## [DML]

* 테이블 명은 다른 테이블의 이름과 중복되면 안 된

다. DDL 명령어의 경우 실행시 하A지U만TO COMMIT

* 테이블 내의 칼럼명은 중복될 수 없다 . DML의 경우 을CO입M력M해IT야 한다 .
* 각 칼럼들은 로, 구분되고 로 끝; 난다 .
* 칼럼 뒤에 데이터 유형은 꼭 지정되어야 한다 .
* 테이블명과 칼럼명은 반드시 문자로 시작해야한다 .

- A-Z,a-z,0-9,\_,$,#만 사용 가능

## 제약조건; 데이터의 무결성 유지

1. PRIMARY KEY(기본키) : UNIQUE & NOT NULL
2. UNIQUE KEY(고유키) : 고유키 정의
3. NOT NULL : NULL 값 입력금지

**INSERT INTO** PLAYER (PLAYER) **VALUES** (‘PJS’);

**UPDATE** PLAYER **SET** BACK\_NO = 60;

**DELETE FROM** PLAYER;

**SELECT** PLAYER\_ID **FROM** PLAYER;

**SELECT** PLAYER **AS**선수명 “ ” **FROM** PLAYER;

DISTINCT : 중복 시 회1만 출력

## 와일드카드

1. CHECK :

입력 값 범위 제한

모든\* :

1. FOREIGN KEY(외래키) : NULL

능

가능,

여러속성가

모든% :

한 글자- :

## 테이블 생성

**CREATE TABLE** PLAYER **(**

PLAYER\_ID **CHAER(7) NOT NULL,**

PLAYER\_NAME **VARCHAR2(20) NOT NULL);**

## 테이블 구조 변경

**합성 연산자**

문자와 문자 연결 : ||

--------------------------------------------

------------

## [TCL]

**트랜잭션**밀: 접히 관련되어 분리될 수 없는 개 이상1 의 조DB작 논리. 적 연산단위

NULL 값과의 수치연산은 값N을UL리L 턴한다

NULL 값과의 비교연산은 거짓 (FALSE를)

.

리턴한다 .

**COMMIT**올: 바르게 반영된 데이터를 에 반D영B **ROLLBACK**트: 랜잭션 시작 이전의 상태로 되돌림 COMMIT 되지 않은 모든 트랜잭션을 롤백함 . **SAVEPOINT**저장 지점 :

## 트랜잭션의 특성

**연산자 우선순위**비:교()연->산N자OT-> ->AND->OR

**ROWNUM**원: 하는 만큼의 행을 가져올 때 사용

SELECT \* PLAYER\_NAME FROM PLAYER WHERE ROWNUM = 1;

--------------------------------------------

**원자성1**:**.** 트랜잭션에서 정의된 연산들은 모두 성공 적으로 실행되던지 아니면 전혀 실행되지 않아야 함

**일관성2.**: 트랜잭션 실행 전 내용DB이 잘못 되지 않으면 실행 후도 잘못 되지 않아야 함

------------

## 단일행 함수

1. SELECT, WHERE, ORDER BY

행에 개별적 조작2.

절에서 사용 가능

**고립성3.**: 트랜잭션 실행 도중 다른 트랜잭션의 영

향을 받아 잘못된 결과를 만들어서는 안된다.

**지속성4.**: 트랜잭션이 성공적으로 수행되면 의 DB

내용은 영구적으로 저장된다.

1. 여러 인자가 있어도 결과는 개1만 출력
2. 함수 인자에 상수 , 변수 , 표현식 사용 가능 함수 중첩 가능5.

## 문자형 함수

SAVEPOINT SVPT1;

LOWER :

문자열을 소문자로

ROLLBACK TO SVPT1; COMMIT;

UPPER : 문자열을 대문자로

ASCII : 문자의 값AS반CI환I

CHR : ASCII 값에 해당하는 문자 반환

------------

CONCAT :

문자열1, 2를 연결

## 연산자의 종류

* BETWEEN a AND b : a와 값b 사이에 있으면 됨
* IN (list) : 리스트에 있는 값중 어느 하나라도 일치

SUBSTR : 문자열 중 위m치에서 개의n 문자 반환

LENGTH : 문자열 길이를 숫자 값으로 반환

CONCAT(‘RDBMS’,‘ SQL’) -> ‘RDBMS SQL’

* IS NULL : NULL

값인 경우 은

(Oracle

SUBSTR(‘SQL Expert’,5,3) -> ‘Exp’

VARCHAR2

빈 문자열을 로NU판L단L )

LTRIM(‘xxxYYZZxYZ’,‘x’) -> ‘YYZZxYZ’

* IS NOT NULL : NULL 값이 아닌 경우
* NOT IN (list) : list의 값과 일치하지 않는다
* LIKE ‘비교문자열’ : 비교문자열과 형태가 일치

SELECT PLAYER\_NAME 선수명 FROM PLAYER WHERE TEAM\_ID = ‘K2’; -> 팀ID가 인K2사람

TRIM(‘x’ FROM ‘xxYYZZxYZxx’) -> ‘YYZZxYZ’

## 숫자형 함수

SIGN(숫자) : 숫자가 양수면 1 음수면 -1 0이면 반0환 MOD(숫자1, 숫자2) : 숫자1을 숫자 2로 나누어 나머지 반환

WHERE TEAM\_ID **IN**

(‘K2’,‘K7’); -> K2,K7인 사람

CEIL(숫자) : 크거나 같은 최소 정수 반환

WHERE HEIGHT **BETWEEN** 170 **AND** 180;

-> 키가 인17사0 람~ 180

WHERE POSITION **IS NULL**포; 지->션 없는 사람

FLOOR(숫자) : 작거나 같은 최대 정수 리턴

ROUND(38.5235,3) -> 38.524

TRUNC(38.5235,3) -> 38.523

## 날짜형 함수

SYSDATE : 현재날짜와 시각 출력

1. HAVING
2. HAVING

절에는 집계함수를 이용하여 조건 표시 o

절은 일반적으로 뒤G에RO위UP치BY

EXTRACT : 날짜에서 데이터 출력

TO\_NUMBER(TO\_CHAR(d,‘YYYY’)) :

출력

연도를 숫자로

## SEARCHED\_CASE\_EXPRESSION

CASE WHEN LOC = ‘a’ THEN ‘b’

## SIMPLE\_CASE\_EXPRESSION

1 = 하루, 1/24 = 1시간, 1/24/60 = 1분

**NULL** 은**관**또**련**는**함**공**수**백(N아U님LL 0 )

CASE LOC WHEN ‘a’ THEN ‘b’

= DECODE(LOC, 'a', 'b')

ELSE NULL이 생략되어 있음 .

--------------------------------------------

- NVL(식1,식2) :

식1의 값이 이N면UL식L

출력 2 .

------------

공집합을 바꿔주진 않음

* NULLIF(식1,식2) :

식1이 식 2와 같으면 을 NULL

1. SQL

## 특징ORDER BY

문장으로 조회된 데이터들을 다양한 목적에

아니면 식 을 출력1

* COALESCE(식1,식2) : NULL이 아닌 최초의 표현

맞게 특정한 칼럼을 기준으로 정렬하여 출력하는데 사용한다.

식, 모두 이NU면L반L 환

NULL

1. ORDER BY

절에 칼럼명 대신 명A이L나IAS칼럼

e.g. COALESCE(NULL, NULL, ‘abc’) -> ‘abc’ 순서를 나타내는 정수도 사용 가능하다.

--------------------------------------------

1. DEFAULT

값으로 오름차순

(ASC이) 적용되며

------------

DESC

옵션을 통해 내림차순으로 정렬이 가능하다 .

**다중행 집계 함수** 4. SQL 문장의 제일 마지막에 위치한다 .

1. 여러 행들의 그룹이 모여서 그룹당 단 하나의 결과

5. SELECT

절에서 정의하지 않은 칼럼 사용 가능

를 돌려주는 함수이다.

1. GROUP BY

절은 행들을 소그룹화 한다 .

Oracle에서는 을NU가L장L

큰 값으로 취급하며

SQL

1. SELECT, HAVING, ORDER BY

절에 사용 가능

Server에서는 을NU가L장L

작은 값으로 취급한다 .

* ALL : Default 옵션. 생략 가능
* DISTINCT : 같은 값을 하나의 데이터로 간주 옵션

## SELECT

**문장 실행 순서**

COUNT(\*) : NULL 포함 행의 수

COUNT(표현식) : NULL 제외 행의 수

SELECT ALIAS -> FROM -> WHERE -> GROUP BY -> HAVING -> SELECT -> ORDER BY

메모리에 모든 칼O럼RD올E리R 므B로Y에서

SUM, AVG : NULL 제외 합계 , 평균 연산

표준 편차STDDEV :

SELECT에 정의 안된 칼럼 써도 됨 .

분산VARIAN :

## SQL Server의

**WITH TIES**

MAX, MIN : 최대값, 최소값

## GROUP BY, HAVING 절의 특징

SELECT TOP(2) WITH TIES ENAME, SAL FROM EMP

ORDER BY SAL DESC;

1. GROUP BY

절을 통해 소그룹별 기준을 정한 후 ,

급여가 높은 명2을 내림차순으로 출력하는데 같, 은 급

SELECT

절에 집계 함수를 사용한다 .

여를 받는 사원은 같이 출력한다.

1. 집계 함수의 통계 정보는 값을NU가L진L 행을 제외하고 수행한다.

--------------------------------------------

------------

1. GROUP BY

절에서는 사AL용IA불S 가

**JOIN**두: 개 이상의 테이블들을 연결 또는 결합하여

1. 집계 함수는 절W에HE올RE수 없다 . 데이터를 출력하는 것

일반적으로 행들은 나PK값의 F연K관에 의해

JOIN

ALIAS는 처음 테이블 , 정렬은 마지막 테이블 기준

이 성립된다 . 어떤 경우에는 관계PK가, 없FK어도

논리적인 값들의 연관만으로 이 성립가능하다JOIN

**순수 관**. **계 연산자**관:계형 를 새D롭B게 구현

1. SELECT -> WHERE절로 구현

5가지 테이블을 하JO기IN위해서는 최소 번의 과정이 필요하다. (N-1)

4 JOIN

1. PROJECT -> SELECT절로 구현
2. NATRUAL JOIN -> 다양한 으JO로IN구현

사용4. DIVIDE -> x

**EQUI JOIN**개:의2테이블 간에 칼럼 값들이 서로 {a,x}{a,y}{a,z} divdie {x,z} = {a}

정확하게 일치하는 경우에 사용 계를 기반으로 한다.

,대부분 의P관K, FK

## 절 형태FROM JOIN

1. INNER JOIN

SELECT PLAYER.PLAYER\_NAME FROM PLAYER

위 처SQ럼L 컬럼명 앞에 테이블 명을 기술해줘야 함

**NON EQUI JOIN**개:의2테이블 간에 칼럼 값들이 서로 정확하게 일치하지 않는 경우에 사용

1. NATURAL JOIN

조건절3. USING

조건절4. ON

1. CROSS JOIN
2. OUTER JOIN

‘=’

연산자가 아닌 등BE연T산WE자EN사,용>, <=

**INNER JOIN**조: 건JO에IN서 동일한 값이 있는 행만 반환, USING이나 절ON을 필수적으로 사용

SELECT E.ENAME, E.JOB, E.SAL, S.GRADE FROM EMP E, SALGRADE S

WHERE E.SAL BETWEEN S.LOSAL AND S.HSAL;

**NATURAL JOIN**두: 테이블 간의 동일한 이름을 갖

는 모든 칼럼들에 대해 수행EQUI JOIN , NATURAL

위는 의E 의S값A을L 찾는 것이다.

의 과 범S위에서LOSAL HSAL

JOIN이 명시되면 추가로 서 조JO건IN을 정의할 수 없다

USING, ON, WHERE 절에

, SQL는S지ev원er

============================================ X

==

# 장 활용2 . SQL

## 조건절USING

같은 이름을 가진 칼럼들 중에서 원하는 칼럼에 대해

서만 선택적으로 을EQ할U수I J있OI다N 칼 , JOIN

**집합 연산자**두: 개 이상의 테이블에서 조인을 사용

하지 않고 연관된 데이터를 조회할 때 사용

럼에 대해서 나A테LI이AS블 이름과 같은 접두사를

붙일 수 없다 지원, SQL Server x

SELECT 절의 칼럼 수가 동일하고 절의SELEC동 T

일 위치에 존재하는 칼럼의 데이터 타입이 상호 호환 할 때 사용 가능

## 일반 집합 연산자

1. UNION : 합집합(중복 행 개1로 정) 렬

## 조건절ON

ON 조건절과 조W건H절ER을E 분리하여 이해가 쉬 우며, 칼럼명이 다르더라도 조JO건IN을 사용할 수 있 는 장점이 있다 , ALIAS나 테이블명 반드시 사용

1. UNION ALL : 합집합(중복 행도 표시 ) 정렬 X
2. INTERSECT : 교집합(중복 행 개1로 )

## CROSS JOIN = 카티시안 곱

양쪽 집합의 건M의\*N데이터 조합이 발생한다 .

1. MINUS : 차집합(중복 행 개1로 )
2. CROSS JOIN : 곱집합(PRODUCT)

## OUTER JOIN (LEFT, RIGHT, FULL)

JOIN

조건에서 동일한 값이 없는 행도 반환 가능하다 ,

을 때 동일 테이블 사이의 조인

. FROM절에 동일 테

USING이나 조O건N 절 반드시 사용해야 함

.식SQL

이블이 번2 이상 나타난다 반. 드시 테이블 별칭을 사

에서 안(+붙)

은 쪽으로 한다 JOIN .

용해야 함.

**- LEFT OUTER JOIN :** 먼저 표기된 좌측 테이블에

해당하는 데이터를 읽은 후

,나중 표기된 우측 테이블

**서브 쿼리**하:나의 문안에SQ포L 함되어 있는 또 다

에서 대JO상IN데이터를 읽어 온다 우측 값. 에서 같

은 값이 없는 경우 값으로 채운다NULL .

* **RIGHT OUTER JOIN :** 의LE반FT대OUTER JOIN
* **FULL OUTER JOIN :** 좌우측 테이블의 모든 데이

른 문SQL알려, 지지 않은 기준을 이용한 검색에 사 용.

* 서브쿼리를 괄호로 감싸서 사용한다 .
* 서브쿼리는 단일 행 또는 복수 행 비교 연산자와

터를 읽어 하JO여IN결과를 생성한다 중복 데.

삭제한다.

이터는

함께 사용 가능하다 . 단일 행 비교 연산자는 서브쿼리

의 결과가 반드시 건 1이하여야 하고 복수 행 비교 연산자는 결과 건수와 상관없다.

------------

계층형 질의 테: 이블에 계층형 데이터가 존재하는 경 우 데이터를 조회하기 위해 사용

* 서브쿼리에서는 를OR사D용ER하B지Y못한다 .
* SELECT, FROM, WHERE, HAVING, ORDER BY, INSERT-VALUES, UPDATE-SET 절에 사용 가능

**단일 행 비교 연산자**등 : =,<,>,<>

**START WITH**계: 층 구조 전개의 시작 위치 지정

## 다중 행 비교 연산자

: IN, ALL, ANY, SOME 등

**CONNECT BY**다: 음에 전개될 자식 데이터 지정 **스칼라 서브쿼리**한: 행 한,칼럼만을 반환하는 서브

**PRIOR**절:에C사ON용N되E며CT 현BY재 읽은 , 쿼리

칼럼을 지정한다 . PRIOR 자식 부=모 형태를 사용하

면 계층구조에서 부모 데이터에서 자식 데이터 부모( ->

## 동작 방식에 따른 서브쿼리 분류

자식)

방향으로 전개하는 순방향 전개를 한다 반. 대는

* 비연관 서브쿼리 서: 브쿼리가 메인쿼리 칼럼 안가

역방향 전개

**NOCYCLE**동: 일한 데이터가 전개되지 않음

**ORDER SIBLINGS BY**형: 제 노드간의 정렬 수행

짐. 메인쿼리에 값 제공 목적

* 연관 서브쿼리 서: 브쿼리가 메인쿼리 칼럼 가짐

**WHERE**모:든 전개를 수행한 후에 지정된 조건을 만족하는 데이터만 추출한다 필터링.( )

## 반환 데이터에 따른 서브쿼리 종류

* 단일행 서브쿼리 실: 행결과 건 이1 하
* 다중행 서브쿼리 실: 행결과 여러 건

**LEVEL**루: 트 데이터이면 그 하1위, 프 데이터까지 씩 증가1

데이터면 리 2,

* 다중컬럼 서브쿼리 실: 행결과 컬럼 여러 개

**CONNECT\_BY\_ISLEAF**해: 당 데이터가 리프 데이터 **인라인 뷰**절에:서 F사R용O되M 는 서브쿼리 .

면1,

그렇지 않으면 0

ORDER BY를 사용 가능

**CONNECT\_BY\_ISCYCLE**해:당 데이터가 조상이면

1, 아니면 옵0 션(CY사C용L했E 을 시만 사용 가능 ) **SYS\_CONNECT\_BY\_PATH**루:트 데이터부터 현재 전개할 데이터까지의 경로를 표시한다.

**CONNECT\_BY\_ROOT**현:재 전개할 데이터의 루트 데이터를 표시한다 단항 연산자이다. .

**셀프 조인**한: 테이블 내 두 칼럼이 연관 관계가 있

**뷰** 테: 이블은 실제로 데이터를 가지고 있는 반면 뷰 , 는 실제 데이터를 가지고 있지 않다 가상 테이블이라. 고도 함.

실행 시점에 재S작QL성하여 수행됨 .

## 뷰 사용 장점

**독립성1**:**.** 테이블 구조가 변경되어도 뷰를 사용하는

응용 플그램은 변경하지 않아도 된다.

**편리성2.**: 복잡한 질의를 뷰로 생성함으로써 관련 질의를 단순하게 작성할 수 있다.

**보안성3.**: 직원의 급여정보와 같이 숨기고 싶은 정 보가 존재할 때 사용

**CREATE VIEW** V\_PLAYER\_TEAM **AS SELECT** ...

할 수 있다 . e.g.같은 매니저를 두고 있는 사원들 중 최대 값

* **AVG**원:하는 조건에 맞는 데이터에 대한 통계 값

e.g. 같은 매니저 내에서 앞의 사번과 뒤의 사번의 평 균

ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING

(현재 행을 기준으로 파티션 내에서 앞의 건 현1 재,

; 행, 뒤의 건1 을 범위로 지정 )

**DROP VIEW** V\_PLAYER\_TEAM;

------------

**- COUNT**조: 건에 맞는 데이터에 대한 통계 값

ex)본인의 급여보다 이5하0 가 적거나 이하로15많0 은 급여를 받는 인원수

**ROLLUP**을 생: 성S하u기bto위ta해l 사용 ,

Grouping Columns의 수를 이N라고 했을 때

Level의 이Su생bt성ot된al다 인수 순서. 에 주의

N+1

**행 순서 관련 함수[ ] -** 지원SQL Server

* **FIRST\_VALUE** 파: 티션별 윈도우의 처음 값
* **LAST\_VALUE** 파: 티션별 윈도우의 마지막 값

**GROUPING** 집: 계 표시면

1아, 니면 0

* **LAG** 파: 티션별 윈도우에서 이전 몇 번째 행의 값
* **LEAD** 파:티션별 윈도우에서 이후 몇 번째 행의

**CUBE**결:합 가능한 모든 값에 대하여 다차원 집계 값

를 생성 , ROLLUP에 비해 시스템에 부하 심함 . 2^N

CUBE(A, B) = GROUPING SETS(A, B, (A,B ), ())

**GROUPING SETS**인: 수들에 대한 개별 집계를 구할 수 있다 . 다양한 소계 집합 생성 가능

------------

## 비율 관련 함수[ ]

* **RATIO\_TO\_REPORT**파: 티션 내 전체 에 대SUM 한 행별 칼럼 값의 백분율을 소수점으로 구할 수 있 다. >0, <=1
* **PERCENT\_RANK**파:티션별 윈도우에서 처음 값 을 마0,지막 값을 로 하여1 행의 순서별 백분율을

**윈도우 함수**행과:

행간의 관계를 정의하거나 행과

구한다. 0>=,<=1

행간을 비교 연산하는 함수,

## 순위 관련 함수[ ]

* **RANK** 동: 일한 값에 대해서는 동일한 순위를 부여

(1,2,2,4)

* **DENSE\_RANK** 동:일한 순위를 하나의 등수로 간 주(1,2,2,3)
* **ROW\_NUMBER** 동: 일한 값이라도 고유한 순위 부 여

(1,2,3,4)

* **CUME\_DIST**현: 재 행보다 작거나 같은 건수에 대

한 누적백분율을 구한다. >0, <=1

* **NTILE**파: 티션별 전체 건수를 인수 값으로 등분 N

한 결과를 구할 수 있다.

--------------------------------------------

------------

**[DCL]** 유: 저 생성하고 권한을 제어할 수 있는 명령 어

## OracLe과 의SQ사L 용S자erv아er키텍처 차이

**OracLe** 유: 저를 통해 에 접D속B 을 하는 형태 와 , ID

## 집계 관련 수함[ ]

- **SUM**파:티션별 윈도우의 합 구할 수 있다 .

e.g. 같은 매니저를 두고 있는 사원들의 월급 합

PW 방식으로 인스턴스에 접속을 하고 그에 해당하는

스키마에 오브젝트 생성 등의 권한을 부여받게 됨

**SQL Server**인: 스턴스에 접속하기 위해 로그인이라

- **MAX, MIN**파:티션별 윈도우의 최대 최소, 값을 구

는 것을 생성하게 되며

,인스턴스 내에 존재하는 다수

의 에DB연결하여 작업하기 위해 유저를 생성한 후

1. 변수,

상수 등을 선언하여 문S장QL간 값을 교환

로그인과 유저를 매핑해 주어야 한다. Windows 인증

1. IF, LOOP

등의 절차형 언어를 사용하여 절차적인

방식과 혼합 모드 방식이 존재함 프로그램이 가능하도록 한다.

1. DBMS 정의 에러나 사용자 정의 에러를 정의하여

**시스템 권한**사:용자가 문을S실Q행L 하기 위해 필 요한 적절한 권한

* **GRANT** 권한 부여:

사용할 수 있다.

1. PL/SQL은 에Or내ac장le되어 있으므로 호환성 굳
2. 응용 프로그램의 성능을 향상시킨다 .

* **REVOKE**권한 취소 :

**GRANT** CREATE USER **TO** SCOTT; CONN SCOTT/TIGER(ID/PW)

CREATE USER PJS IDENTIFIED BY KOREA7;

1. Block 단위로 처리 통->신량을 줄일 수 있다

**DECLARE** 절: 에BE서GI사N~용E될ND변수와 인수 에 대한 정의 및 데이터 타입 선언부

**BEGIN~END**개:발자가 처리하고자 하는 문과

.

SQL

**GRANT** CREATE SESSION **TO** PJS; 여러 가지 비교문 , 제어문을 이용 필요한 로직 처리

**GRANT** CREATE TABLE **TO** PJS;

**EXCEPTION**절:에B서EG실IN행~E되N는D 문

SQL

**REVOKE** CREATE TABLE **FROM** PJS;

모든 유저는 각각 자신이 생성한 테이블 외에 다른 유저의 테이블에 접근하려면 해당 테이블에 대한 오 브젝트 권한을 소유자로부터 부여받아야 한다.

이 실행될 때 에러가 발생하면 그 에러를 어떻게 처 리할지 정의하는 예외 처리부

**T-SQL**근: 본적으로 를 제SQ어L하S는er언ve어r

CREATE Procedure schema\_NAME.Procedure\_name

**ROLE**유: 저에게 알맞은 권한들을 한 번에 부여하기

**Trigger**특정:

한 테이블에

INSERT, UPDATE,

위해 사용하는 것

DELETE와 같은 문D이ML수행되었을 때 에,서D자B

**CREATE ROLE** LOGIN\_TABLE;

**GRANT** CREATE TABLE **TO** LOGIN\_TABLE;

## DROP USER PJS CASCADE;

동으로 동작하도록 작성된 프로그램,

아닌 자동 수행DB

## 프로시저와 트리거의 차이점

사용자 호출이

**CASCADE**하: 위 오브젝트까지 삭제

**프로시저** 절 :내에BEGIN~END COMMIT,

--------------------------------------------

------------

**절차형 SQL**문: 의SQ연L속적인 실행이나 조건에 따

른 분기처리를 이용하여 특정 기능을 수행하는 저장 모듈을 생성할 수 있다, Procedure, User Defined

ROLLBACK과 같은 트랜잭션 종료 명령어 사용가능 .

EXECUTE 명령어로 실행

**트리거** 절: B내E에GIN사~용EN불D 가 생성 후 자 .

동 실행.

Function, Trigger 등이 있음

**저장 모듈**문:장P을L/서SQ버L에 저장하여DB사

용자와 애플리케이션 사이에서 공유할 수 있도록 만 든 일종의 컴S포QL넌트 프로그램 독립적,으로 실행 되거나 다른 프로그램으로부터 실행될 수 있는 완전 한 실행 프로그램

## 특징PL/SQL

1. Block 구조로 되어있어 각 기능별로 모듈화 가능

- 끝 -