

통계 데이터베이스 실습

# 데이터베이스 기초

# 학습 목표

- 데이터베이스의 광범위한 활용에 대해 고려
- 데이터베이스 개발역사에 대한 간단한 고찰
- 기본적인 데이터베이스 개념 학습
- 보편적인 데이터베이스 관리 소프트웨어의 소개
- 정규화를 통한 효과적인 데이터 조직화
- 데이터베이스 설계 과정 고찰
- 데이터들간의 관계의 이해
- 구조적 질의어(SQL)에 대한 이해
- 자주 사용되는 SQL 명령어 학습

# 데이터베이스에 대해 알아야 하는 이유

- 데이터는 사용되려면 조직화되어야 함
- 능률적인 컴퓨터 과학자는 데이터베이스 설계를 숙지해야함
- 정규화 : 다단계 데이터베이스 설계과정
- 구조적 질의어(SQL) : 데이터의 저장, 변경, 검색을 위한 인터페이스

# 데이터베이스 응용

## □ 데이터베이스

- 논리적 관계로부터 구축된 데이터 구조
- 질의를 통한 데이터 조작의 지원

## □ 데이터베이스 응용은 매우 넓음

- 범위 : 인간 유전자 응용부터 우주 정거장 계획까지

## □ 데이터베이스는 일상생활과 컴퓨터과학 분야에서 모두 중요함

# 데이터베이스 관리시스템의 간략한 역사

## □ 1970 – 1975

- IBM 직원인 E.F. Codd 와 C.J. Date의 작업
  - 데이터베이스 구조를 위한 이론적 모델을 창조
  - 모델은 데이터베이스 설계의 기초가 됨
- 데이터를 정렬하고 조직하는 소프트웨어
  - IBM의 System R과 UC-Berkeley의 Ingres
  - 구조적 질의어(SQL)를 채택
  - SQL은 데이터베이스 표준이 됨
  - PC들을 위한 데이터베이스 관리시스템(DBMS)
- Martin-Marietta의 Wayne Ratliff가 Vulcan을 개발

# 데이터베이스 관리시스템의 간략한 역사(계속)

## □ 1980 – 현재

- Vulcan을 dBASE II로 개명 (dBase I은 없음)
- dBASE II는 다른 회사들을 자극
  - Paradox, Microsoft Access, 또는 FoxPro 등등
- 데이터베이스가 업무에 필수적이 됨
  - 전사적 의사 결정
  - 시스템 : 재고관리에서 고객지원까지

# 데이터베이스 관리시스템 기초

## □ DBMS의 6가지 기본 기능

- 데이터베이스 보안 관리
- 여러 사용자의 데이터베이스 접근 관리
- 데이터베이스 백업 및 복구 관리
- 데이터 무결성 보장
- 데이터베이스에 대한 최종사용자 인터페이스 제공
- 데이터베이스로부터 정보의 추출을 위한 질의어 제공

# 데이터베이스 개념

## □ 데이터베이스 기본요소

- 데이터베이스 : 하나 이상의 테이블(개체) 집합
- 테이블 : 행과 열로 구분됨(스프레드시트)
- 행(레코드 또는 튜플) : 열의 집합
- 열(필드 또는 어트리뷰트)
  - 특정 정보를 표현함
  - 가능한 열 값의 집합을 도메인이라 함
- 인덱스(순서) : 정보의 접근을 용이하게 함



[그림 8-1] 데이터베이스 테이블은 행과 열로 구성된다.

Song_Name	Artist_Code	Album_Num	Track_Num	Album_Name	Artist_Name	Media_Type	Genre_Code
American Soldier	TK	911	3	Shock'n Y'all	Toby Keith	MP3	CW
Beer For My Horses	TK	826	5	Unleashed	Toby Keith	MP3	CW
Big Time	PG	10	14	Shaking the Tree	Peter Gabriel	CD	POP
Come As You Are	NIRV	24	3	Nevermind	Nirvana	MP3	NW
Lithium	NIRV	24	5	Nevermind	Nirvana	MP3	NW
One Tree Hill	U2	146	9	The Joshua Tree	U2	CD	NW
Only a Lad	OB	78	16	Best O Boingo	Oingo Boingo	CD	NW
Sledgehammer	PG	10	3	Shaking the Tree	Peter Gabriel	CD	POP
Sloop John B	BB	45	7	Pet Sounds	The Beach Boys	Vinyl	CLA
The Thrill is Gone	BBK	98	8	The Best of B.B. King	B.B. King	MP3	BLU
With or Without You	U2	146	3	The Joshua Tree	U2	CD	NW
*		0	0				

Record: 1 of 11

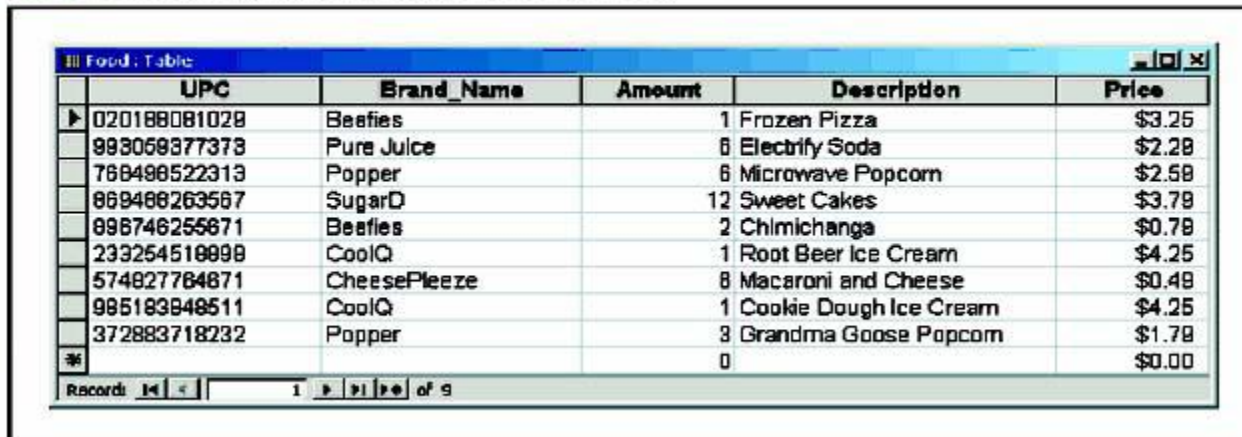
# 인덱스

- 인덱스 : 특정 열(들)에 따라 레코드를 조직화하는 데이터 구조
- 예 : 음악 데이터베이스, 전화번호부
- 주요 장점
  - 유연성 : 서로 다른 여러 열에 대하여 정렬 가능
  - 탐색 및 검색의 속도 향상
- 주요 단점
  - 추가적인 저장공간 필요
  - 갱신의 속도 저하

# 인덱스(계속)

## □ 인덱스의 예 : 식료품가게 쇼핑

[그림 8-2] 모든 일상생활에서 데이터베이스 개념을 사용한다.



UPC	Brand Name	Amount	Description	Price
020188081028	Beefies	1	Frozen Pizza	\$3.25
993059377373	Pure Juice	6	Electrify Soda	\$2.28
768498522313	Popper	6	Microwave Popcom	\$2.58
869486263567	SugarD	12	Sweet Cakes	\$3.78
896746255671	Beefies	2	Chimichanga	\$0.78
233254518098	CoolQ	1	Root Beer Ice Cream	\$4.25
574827784871	CheesePleeze	8	Macaroni and Cheese	\$0.48
985183848511	CoolQ	1	Cookie Dough Ice Cream	\$4.25
372883718232	Popper	3	Grandma Goose Popcom	\$1.78
*		0		\$0.00

Records: 1 of 9

# 인덱스(계속)

- 데이터베이스 내에서 정보는 순차적으로 보관됨
- 키 : 정렬 순서의 결정을 위해 사용되는 열(들)
  - UPC 열을 키로서 식품 항목을 정렬
  - Brand\_Name과 Description으로 식품 항목을 정렬
- 데이터를 조작하거나 참조하기 위한 수단
  - 보고서, 양식, 테이블, 저 수준 화일 I/O, 소스코드

[그림 8-3] UPC 열을 키로 사용하여 정렬된 데이터베이스 레코드

Food_by_UPC : Table					
	UPC	Brand_Name	Amount	Description	Price
▶	020188081029	Beefies	1	Frozen Pizza	\$3.25
	233254518898	CoolQ	1	Root Beer Ice Cream	\$4.25
	372883718232	Popper	3	Grandma Goose Popcorn	\$1.79
	574827764671	CheesePleeze	6	Macaroni and Cheese	\$0.49
	768498522313	Popper	6	Microwave Popcorn	\$2.59
	869488263567	SugarD	12	Sweet Cakes	\$3.79
	896746255671	Beefies	2	Chimichanga	\$0.79
	985183848511	CoolQ	1	Cookie Dough Ice Cream	\$4.25
	983059377373	Pure Juice	6	Electrify Soda	\$2.29
*			0		\$0.00

Record: 14 | 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 of 9

[그림 8-4] Brand\_Name과 Description으로 정렬된 데이터베이스 레코드

Food_by_Brand_Desc: Table					
	UPC	Brand_Name	Amount	Description	Price
▶	896746255671	Beefies	2	Chimichanga	\$0.79
	020188081029	Beefies	1	Frozen Pizza	\$3.25
	574827764671	CheesePleeze	6	Macaroni and Cheese	\$0.49
	985183848511	CoolQ	1	Cookie Dough Ice Cream	\$4.25
	233254518888	CoolQ	1	Root Beer Ice Cream	\$4.25
	372883718232	Popper	3	Grandma Goose Popcorn	\$1.79
	768498522313	Popper	6	Microwave Popcorn	\$2.59
	993059377373	Pure Juice	6	Electrify Soda	\$2.29
	869488263567	SugarD	12	Sweet Cakes	\$3.79
*			0		\$0.00

Record: 1 of 9

# 정규화

## □ 정규화

- 데이터베이스 설계를 위한 일련의 표준 규칙
- 프로세스 : 정규화라는 순서적 단계
  - 5가지의 정규형이 존재
  - 제 3정규형으로 충분함
- 해결되는 세가지 데이터베이스 설계 문제
  - 특정 실 세계 항목의 표현
  - 데이터의 중복(반복)
  - 제외되거나 일관성 없는 정보

# 정규화를 위한 준비: 열의 수집

- 관련된 모든 필드(열 또는 애트리뷰트)의 리스트를 만듦
  - 필드의 출처 : 최종사용자 보고서; 예를 들면 노래재고
- 필드들을 열 리스트에 기록
- 사용자가 지정한 입력 양식을 검토
- 보고서의 각 필드를 테이블의 열로 변환



[그림 8-5] 테이블 열들이 강조 표시된 최종사용자 보고서

## Songs

<i>Artist Name</i>	<i>Song Name</i>	<i>Track Num</i>	<i>Album Name</i>	<i>Media Type</i>	<i>Genre Code</i>
B.B. King	The Thrill is Gone	8	The Best of B.B. King	MP3	BLU
Nirvana	Come As You Are	3	Nevermind	MP3	NW
Nirvana	Lithium	5	Nevermind	MP3	NW
Oingo Boingo	Only a Lad	16	Best O Boingo	CD	NW
Peter Gabriel	Big Time	14	Shaking the Tree	CD	POP
Peter Gabriel	Sledgehammer	3	Shaking the Tree	CD	POP
The Beach Boys	Sloop John B	7	Pet Sounds	Vinyl	CLA
Toby Keith	American Soldier	3	Shock'n Y'all	MP3	CW
Toby Keith	Beer For My Horses	5	Unleashed	MP3	CW
U2	One Tree Hill	9	The Joshua Tree	CD	NW
U2	With or Without You	3	The Joshua Tree	CD	NW

## 정규화를 위한 준비 : 열의 수집(계속)

- 보고서에 있는 필드들을 열 리스트로 조정
- 열들의 테이블을 생성
  - 연관된 필드들을 조합
  - 연관된 정보를 논리적으로 그룹화
  - 예 : 예술가 및 노래 화일에 대한 정보
- 물리적 음악 데이터베이스 생성을 위해  
데이터 수집

The image shows a software window titled "Songs" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains a form with the following fields and values:

Song_Name	Sledgehammer
Track_Num	3
Album_Num	10
Album_Name	Shaking the Tree
Media_Type	CD
Artist_Code	PG
Artist_Name	Peter Gabriel
Genre_Code	POP

At the bottom of the window, there is a record navigation bar. It includes the text "Record:" followed by navigation icons (back, forward, first, last, search), a text box containing the number "1", and the text "of 11".

# 제 1정규형

- 비정규화 테이블 : 둘 이상의 값을 가진 행/열의 교차점 존재
- 제1정규형(1NF) : 중복의 제거
  - 중복된 열에 대해 새로운 레코드를 생성
  - 레코드의 모든 열들이 하나의 값을 갖도록 빈 공간을 채움
  - 중복이 있는 열들 : the Album\_Num, Album\_Name, Artist\_Code, Artist\_Name, Media\_Type, and Genre\_Code
- 아직 남아 있는 중복에 대해서는 나중에 제거

# 제 2정규형

## □ 다음 단계

- 테이블에 기본키를 지정
- 테이블 내에서 함수 종속을 식별

□ 기본키(PK) : 테이블 내에서 한 행을 유일하게 식별하는 하나의 열, 또는 여러 열의 조합(복합)

- 예 : 학번 또는 Artist\_Code

## 제 2정규형(계속)

- 결정자 : 같은 행에 있는 다른 열(들)에 할당된 값을 결정하는데 사용되는 열(들)
  - 예 : Artist\_Code의 결정자는 Artist\_Code임
- 함수 종속
  - 결정자와 이것이 결정하는 열들
  - 첫 번째 열의 각 값이 두 번째 열의 단일 값과 부합됨
  - 예 : Artist\_Name은 Artist\_Code에 함수 종속됨

# 제 2 정규형(계속)

## □ 제 2정규형(2NF)

- 제 1정규형이고

- 기본키가 아닌 열들이 기본키에 함수 종속됨

## □ 제 2정규형의 생성

- 기본키에 종속되지 않은 열들을 식별

- 이런 열들을 제거하여 새로운 테이블로 이동

- 자동 제 2정규형 : 복합키를 가지지 않은 테이블

## □ 제 2정규형의 주요 장점 : 디스크 공간의 절약

그리 <1>의 <2>에 기본키에 주소저이지 앞으 역으 제거하여 새르으 테이블으 생성하다

## 제2정규형 데이터

### Songs 테이블

Artist_Code	Song_Name	Album_Num	Track_Num	Media_Type	Genre_Code
BB	Sloop John B	45	7	Vinyl	CLA
BBK	The Thrill is Gone	98	8	MP3	BLU
NIRV	Come As You Are	24	3	MP3	NW
NIRV	Lithium	24	5	MP3	NW
OB	Only a Lad	78	16	CD	NW
PG	Big Time	10	14	CD	POP
PG	Sledgehammer	10	3	CD	POP
TK	American Soldier	911	3	MP3	CW
TK	Beer For My Horses	826	5	MP3	CW
U2	One Tree Hill	146	9	CD	NW
U2	With or Without You	146	3	CD	NW

### Artists 테이블

Artist_Code	Artist_Name
BB	The Beach Boys
BBK	B.B. King
NIRV	Nirvana
OB	Oingo Boingo
PG	Peter Gabriel
TK	Toby Keith
U2	U2

### Album 테이블

Album_Num	Album_Name
10	Shaking the Tree
24	Nevermind
45	Pet Sounds
78	Best O Boingo
98	The Best of B.B. King
146	The Joshua Tree
826	Unleashed
911	Shock'n Y'all



# 제 3정규형

## □ 제 3정규형(3NF)

### □ 이행적 함수 종속을 제거

□ 이행적 함수 종속 : 열이 기본키의 일부가 아닌 다른 열에 함수 종속

□ 예 : Genre\_Desc는 Genre\_Code에 종속됨

### □ 키가 아닌 각 필드는 기본키에 대한 사실이어야 함

[그림 8-11] Genre\_Desc 열이 추가된 Songs 테이블

## Songs 테이블

Artist_Code	Song_Name	Album_Num	Track_Num	Media_Type	Genre_Code	Genre_Desc
BB	Sloop John B	45	7	Vinyl	CLA	Classic Rock
BBK	The Thrill is Gone	98	8	MP3	BLU	Blues
NIRV	Come As You Are	24	3	MP3	NW	NewWave
NIRV	Lithium	24	5	MP3	NW	NewWave
OB	Only a Lad	78	16	CD	NW	NewWave
PG	Big Time	10	14	CD	POP	Popular
PG	Sledgehammer	10	3	CD	POP	Popular
TK	American Soldier	911	3	MP3	CW	Country Western
TK	Beer For My Horses	826	5	MP3	CW	Country Western
U2	One Tree Hill	146	9	CD	NW	NewWave
U2	With or Without You	146	3	CD	NW	NewWave

# 제 3정규형(계속)

## □ 3NF의 생성

- 이행적 함수 종속을 제거
- 제거된 열들을 새로운 테이블로 배치

## □ 3NF의 주요 장점 : 디스크 공간의 절약

## □ 3NF 레벨까지 다음의 새로운 테이블들이 생성됨

- Genre, Artists, Album

[그림 8-12] 3NF인 Songs와 Genre테이블

제3정규형 데이터

Songs 테이블

Artist_Code	Song_Name	Album_Num	Track_Num	Media_Type	Genre_Code
BB	Sloop John B	45	7	Vinyl	CLA
BBK	The Thrill is Gone	98	8	MP3	BLU
NIRV	Come As You Are	24	3	MP3	NW
NIRV	Lithium	24	5	MP3	NW
OB	Only a Lad	78	16	CD	NW
PG	Big Time	10	14	CD	POP
PG	Sledgehammer	10	3	CD	POP
TK	American Soldier	911	3	MP3	CW
TK	Beer For My Horses	826	5	MP3	CW
U2	One Tree Hill	146	9	CD	NW
U2	With or Without You	146	3	CD	NW

Genre 테이블

Genre_Code	Genre_Desc
BLU	Blues
CLA	Classic Rock
CW	Country Western
NW	New Wave
POP	Popular

[그림 8-13] 중복의 제거는 저장 공간을 절약한다.

### 제3정규형 데이터

#### Songs 테이블

Artist_Code	Song_Name	Album_Num	Track_Num	Media_Type	Genre_Code
BB	Sloop John B	45	7	Vinyl	CLA
BBK	The Thrill is Gone	98	8	MP3	BLU
NIRV	Come As You Are	24	3	MP3	NW
NIRV	Lithium	24	5	MP3	NW
OB	Only a Lad	78	16	CD	NW
PG	Big Time	10	14	CD	POP
PG	Sledgehammer	10	3	CD	POP
TK	American Soldier	911	3	MP3	CW
TK	Beer For My Horses	826	5	MP3	CW
U2	One Tree Hill	146	9	CD	NW
U2	With or Without You	146	3	CD	NW

#### Genre 테이블

Genre_Code	Genre_Desc
BLU	Blues
CLA	Classic Rock
CW	Country/Western
NW	New/Wave
POP	Popular

#### Artists 테이블

Artist_Code	Artist_Name
BB	The Beach Boys
BBK	B.B. King
NIRV	Nirvana
OB	Oingo Boingo
PG	Peter Gabriel
TK	Toby Keith
U2	U2

#### Album 테이블

Album_Num	Album_Name
10	Shaking the Tree
24	Nevermind
45	Pet Sounds
78	Best O Boingo
98	The Best of B.B. King
146	The Joshua Tree
826	Unleashed
911	Shock'n Y'all

# 데이터베이스 설계 과정

- 정규화 데이터베이스 설계를 위한 6단계
- 예 : 성적관리 시스템 개발

# 단계 1 — 조사와 정의

- 모델링 하고자 하는 정보에 대해 조사 및 연구
- 데이터베이스의 목적과 용도를 정의
- 최종사용자가 사용하는 문서들을 활용
- 설계과정에서 최종사용자를 참여시킴
- 강의 계획서를 기반으로 한 성적 관리 시스템

## 단계 2 – 마스터 열 리스트의 작성

- 정보를 위한 필드 리스트 생성
- 필드 속성은 다음과 같은 항목 포함 가능
  - 필드 명
  - 데이터 타입 (char, varchar, number, date, 등등.)
  - 길이
  - 소수점 밑의 자리 수(존재시)
- 필드를 위한 사용자 문서를 검토
  - 양식과 보고서들은 필드들의 근원이 됨
  - 필드의 예: 학번, 성명(First Name, Last Name)



# 단계 3 — 테이블 생성

- 정의된 열들을 논리적으로 그룹 지어 테이블로 만듦
  - 설계과정의 핵심
  - 정규화 규칙에 크게 의존적임
- 데이터베이스 설계의 주요 규칙 : 1NF – 3NF
- 3NF 형태의 테이블은 잘 정의된 것임
- 데이터베이스의 정규화는 벽장 정리와 유사

[그림 8-14] 성적 관리 시스템을 위해 생성된 테이블

The image displays four database tables from a grade management system. Each table is shown in a separate window with a standard grid interface and navigation controls.

**Students Table:**

Student_ID	First_Name	Last_Name	E_Mail	Grade_Level
223212578	Squidward	Tentacles	Squidward@wsu.edu	So
543768893	Yosemite	Sam	Yosemite@wsu.edu	Ju
984367183	Bugs	Bunny	Bugs@wsu.edu	Sr

**Grades Table:**

Course_ID	Student_ID	Homework_Ave	Quiz_Ave	Test_Ave	Final_Exam	Final_Grade	Letter_Grade
1829388133	111111111	70	85	80	72	87	D+
1829388133	223212578	80	85	90	86	85	B
1829388133	984367183	98	92	94	98	95	A
2731288812	543768893	84	82	86	78	83	B-
3712883828	223212578	100	90	88	92	93	A-
3712883828	984367183	100	98	98	96	98	A
4311223112	111111111	60	70	82	80	73	C-
4311223112	223212578	82	84	78	90	84	B

**Grade Levels Table:**

Grade_Level	Grade_Desc
Fr	Freshman
Ju	Junior
So	Sophomore
Sr	Senior

**Courses Table:**

Course_ID	Course_Desc
1829388133	Underwater Programming
2731288812	Intro to Computer Science
3712883828	Software Engineering for Cartoons
4311223112	Database Design in a Toon Environment

# 단계 4 – 관계에 대한 작업

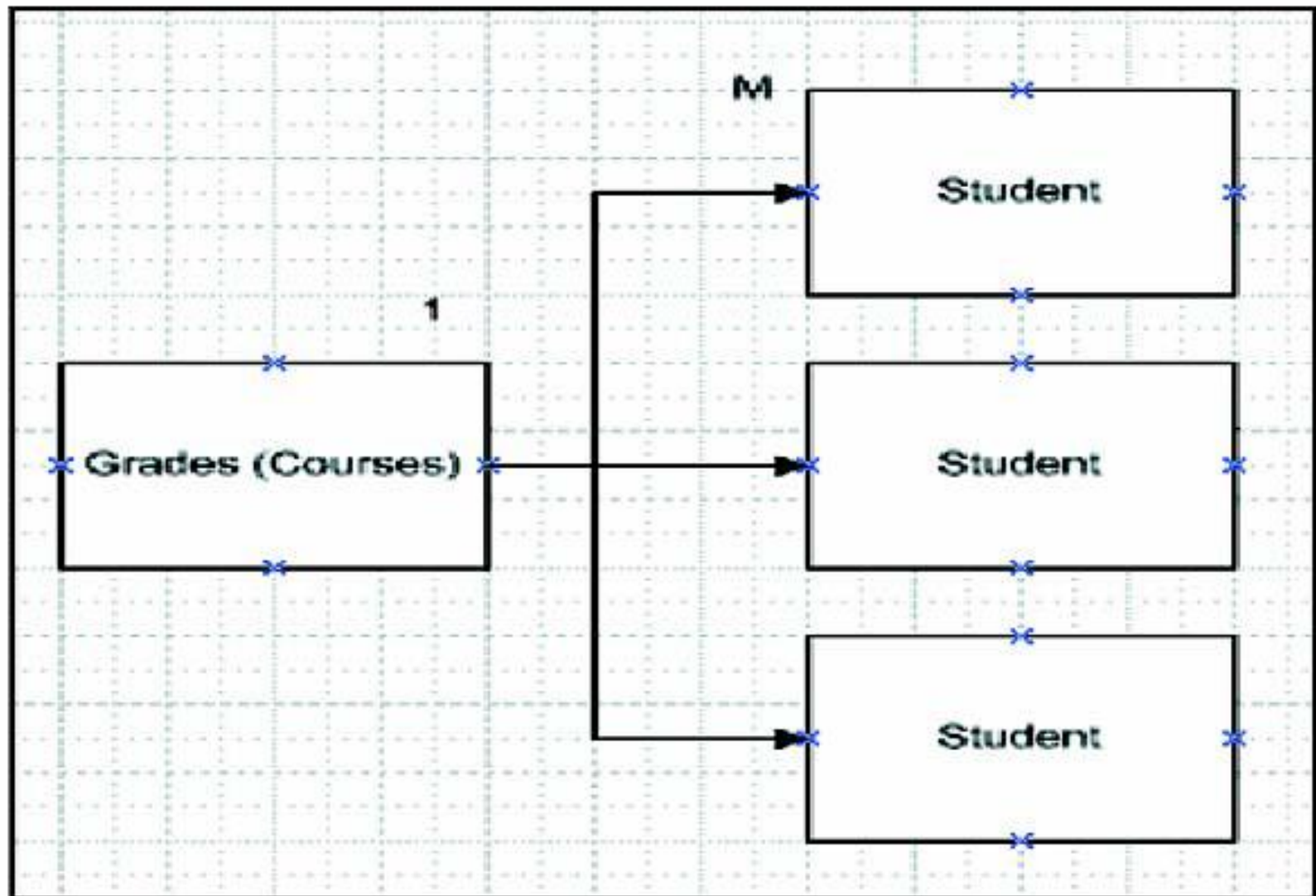
- 관계 : 테이블간의 관련성을 정의
- 이 장에서 논의되는 두 유형의 관계
  - 일 대 다 (1:M)
  - 일 대 일 (1:1)
- 각 테이블에서 기본키와 외래키를 정의
  - 기본키(PK): 이전에 설명하였던 결정자
  - 외래키(FK): 한 테이블 내의 열이 다른 테이블에서 PK임
  - 후속 절에서 PK와 FK의 기능을 기술함

# 단계 4 – 관계에 대한 작업(계속)

## □ 일 대 다 (1:M)

- 가장 보편적인 관계
- 테이블 A의 각 레코드가 테이블 B의 여러 레코드와 관련됨을 나타냄
- “다” 테이블의 FK열(들)이 “일” 테이블의 PK를 참조해야 함
- 예 : Grades 테이블과 Student 테이블

[그림 8-15] Grade의 Student에 대한 관계는 일대다(1:M)이다

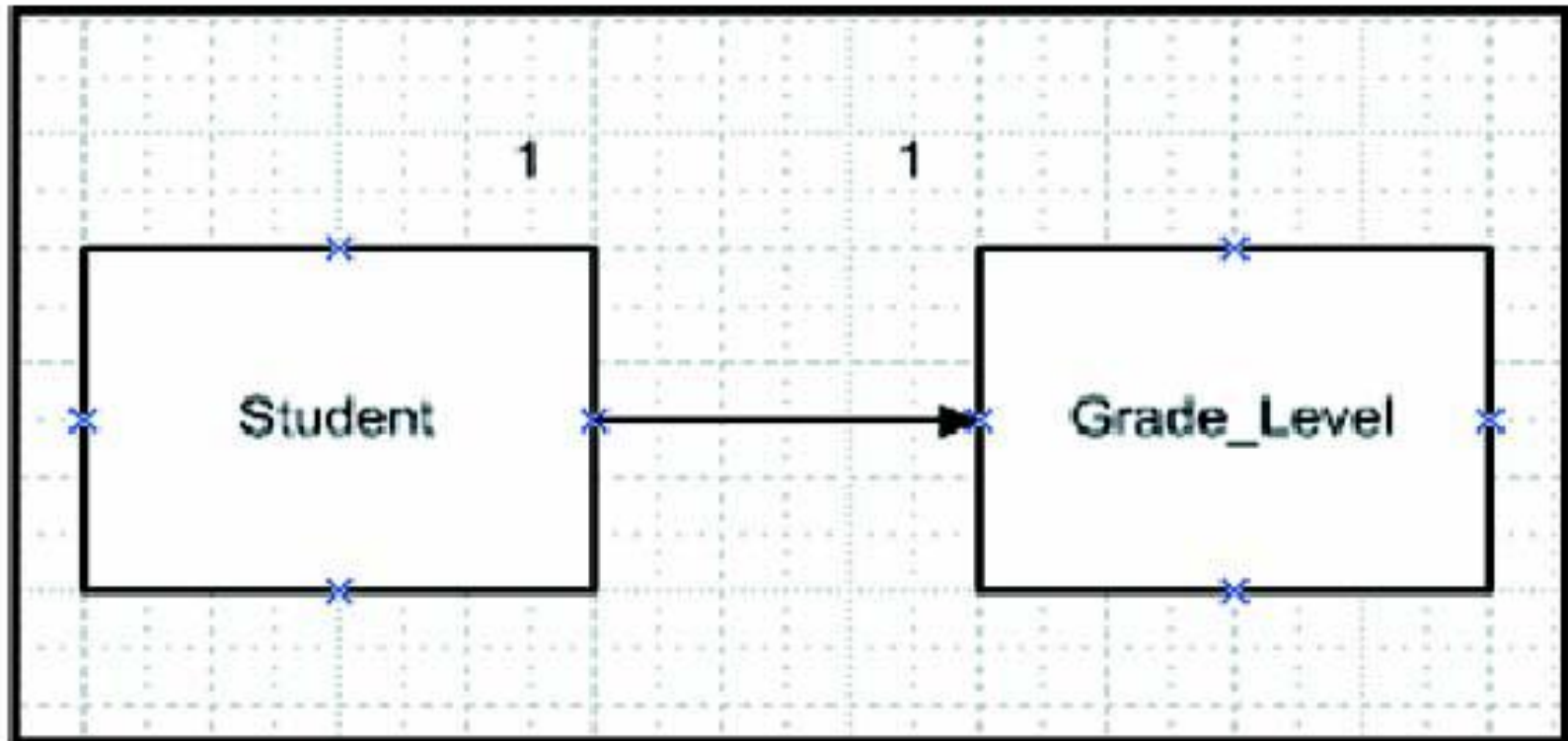


# 단계 4 – 관계에 대한 작업(계속)

## □ 일 대 일 (1:1)

- 테이블 A에 있는 각 레코드가 테이블 B에 있는 오직 하나의 레코드와 부합될 수 있음을 나타냄
- 1 : 1 관계에 있는 테이블들의 결합 여부 고려
- 1 : 1이 때로는 적절함 : 각 개별 학생은 한 학년에 속함 (Student 테이블과 Grade Level 테이블)
- “일” 테이블의 FK열(들)이 다른 “일” 테이블에 있는 PK열(들)을 참조하는 값을 가짐

[그림 8-16] Student의 Grade Level에 대한 관계는 일대일(1:1)이다



# 단계 5 - 설계의 분석

## □ 완료된 작업을 분석

- 설계 오류를 탐색하고, 필요한 경우 테이블을 개선
- 정규형(이상적으로 3NF까지)을 따름
- 위반사항이 있을 경우 교정

## □ ER 모델

- 개체와 관계로 구성된 시각적 다이어그램
- 개체는 데이터베이스 테이블을 나타냄
- 관계는 테이블이 서로 관련되는 방식을 나타냄
- 카디널리티 : 개체 사이의 수치적인 관련성을 나타냄



# 단계 5 - 설계의 분석(계속)

## □ 카디널리티의 유형(과 표기법)

□ 0..1, 0:1 (0 대 1)

□ 0..M, 0:N, 0..\*, 0..n (0 대 다)

□ 1..1, 1:1 (1 대 1)

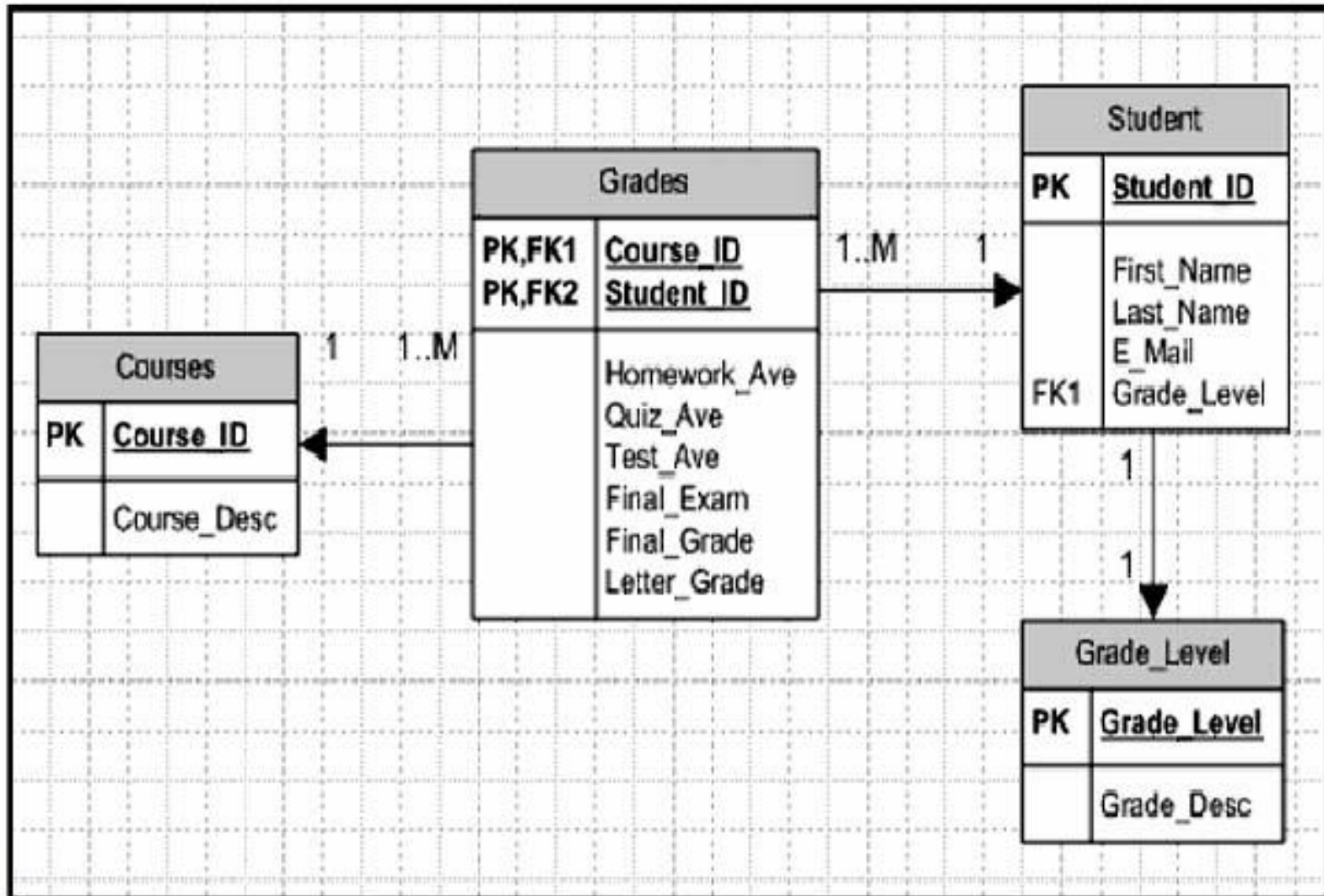
□ 1..M, 1:M, 1:N, 1..\*, 1..n (1 대 다)

□ M..1, M:1, N:1, \*..1, n..1 (다 대 1)

□ M..M, M:M, N:N, \*..\*, n..n (다 대 다)

□ 예 : 성적 관리시스템을 위한 ER모델

[그림 8-17] Visio로 작성된 성적 관리 시스템의 ER모델



# 단계 6 - 재평가

- 데이터베이스 성능을 재평가
  - 데이터베이스가 모든 보고서 및 서식 요구사항들을 만족시키는지 확인
  - 최종사용자를 포함
  - 사용되는 각 테이블과 필드들을 설명
  - 필드들이 사용자의 요구조건에 따라 정의되었는지 확인
- SQL 명령들로 데이터 구조를 조작

# 구조적 질의어 (SQL)

- 구조적 질의어(SQL)의 기능
  - 데이터 조작
  - 데이터 정의
  - 데이터 관리
- 서로 상이한 다수의 SQL “방언”들이 존재
- SQL 명령은 대문자(전통적임)이거나 소문자일 수 있음

# 구조적 질의어(SQL) (계속)

- SQL은 다음의 장점을 제공
  - 훈련시간 감소(영어에 기반한 구문구조)
  - 응용의 이식성을 높임(SQL의 표준화)
  - 전송되는 데이터 양 감소
  - 응용의 속도 증가
- 다음 절은 기본적인 SQL 문장을 보여줌
  - 테이블 생성
  - 데이터 행의 추가(삽입)
  - 특정 정보를 추출하기 위해 테이블에 대한 질의

[그림 8-18] SQL 문의 예 및 결과

## SQL 문

```
SELECT Song_Name, Artist_Code, Album_Num, Track_Num, Media_Type  
FROM Songs  
WHERE Genre_Code = "NW";
```

## 결과:

	Song_Name	Artist_Code	Album_Num	Track_Num	Media_Type
	With or Without You	U2	146	3	CD
	One Tree Hill	U2	146	9	CD
	Only a Lad	OB	78	16	CD
	Lithium	NIRV	24	5	MP3
▶	Come As You Are	NIRV	24	3	MP3
*			0	0	

Record: 5 of 5

# CREATE TABLE 문

□ CREATE TABLE 문: 새로운 테이블 생성

□ 구문 :

□ CREATE TABLE 테이블\_명

□ ( 열\_이름 데이터\_타입 [NULL | NOT NULL]

□ [, 열\_이름 데이터\_타입 [NULL | NOT NULL] ... );

□ NULL/NOT NULL

□ 선택적 특성은 데이터가 요구되는지를 나타냄

# CREATE TABLE 문(계속)

□ 다음 SQL문은 Songs라는 테이블을 생성함

```
CREATE TABLE Songs  
(Song_Name char (50) NOT NULL,  
Album_Num number NOT NULL,  
Artist_Code char (5) NOT NULL,  
Track_Num number NULL,  
Media_Type char (5) NULL,  
Genre_Code char (5) NOT NULL,  
);
```



# INSERT 문

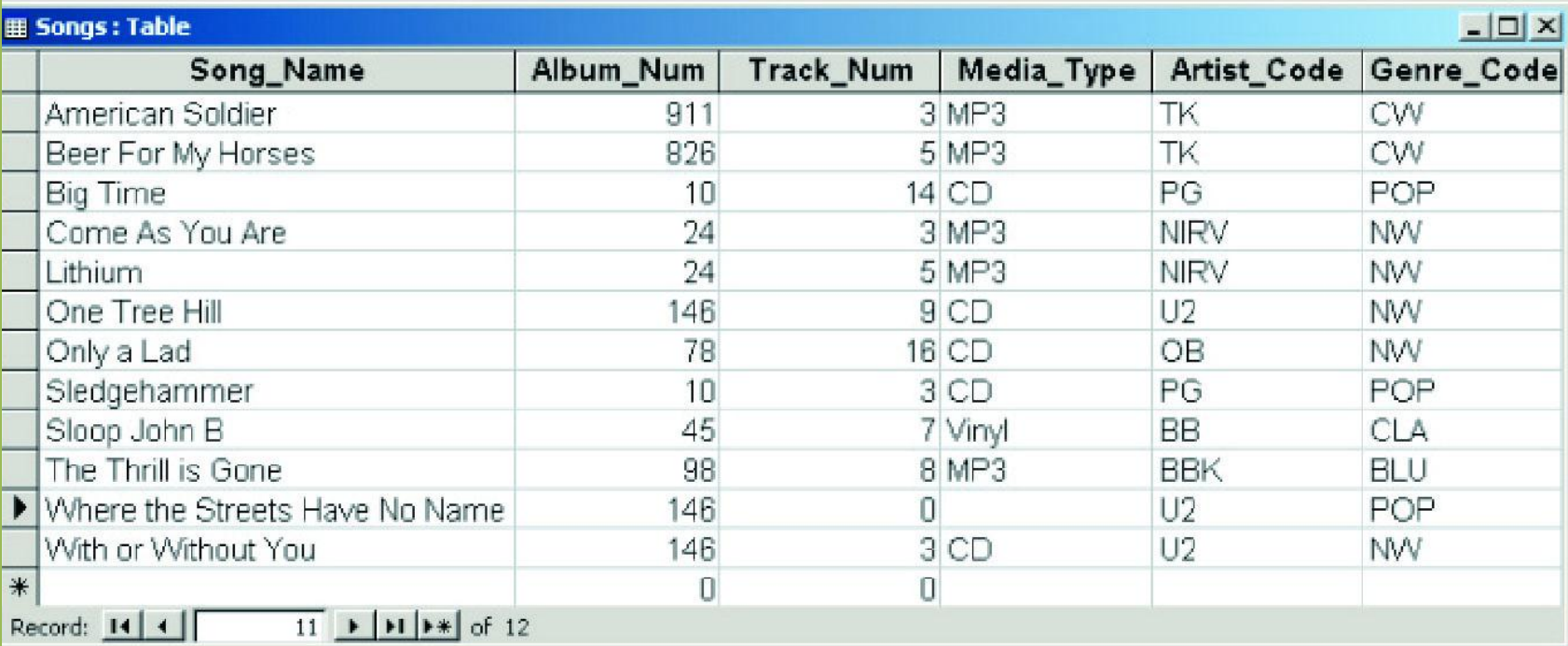
- INSERT 문 : 새로운 데이터 행을 추가
- 구문 :
- INSERT INTO 테이블\_명 [(열1,  
열2, ...)]
  - VALUES (상수1, 상수2, ...)
- INSERT 문은 테이블 명을 필요로 함
- 중괄호 ([..])는 선택적인 열을 규정함
- 가독성을 위해 별도의 행에 열을 표기

그림 8-19] Songs 테이블에 레코드를 추가하는 SQL INSERT 문과 그 결과

## SQL 문

```
INSERT INTO Songs (Song_Name, Album_Num, Artist_Code, Genre_Code)
VALUES ("Where the Streets Have No Name", 146, "U2", "POP");
```

## 결과:



Song_Name	Album_Num	Track_Num	Media_Type	Artist_Code	Genre_Code
American Soldier	911	3	MP3	TK	CW
Beer For My Horses	826	5	MP3	TK	CW
Big Time	10	14	CD	PG	POP
Come As You Are	24	3	MP3	NIRV	NW
Lithium	24	5	MP3	NIRV	NW
One Tree Hill	146	9	CD	U2	NW
Only a Lad	78	16	CD	OB	NW
Sledgehammer	10	3	CD	PG	POP
Sloop John B	45	7	Vinyl	BB	CLA
The Thrill is Gone	98	8	MP3	BBK	BLU
▶ Where the Streets Have No Name	146	0		U2	POP
With or Without You	146	3	CD	U2	NW
*	0	0			

Record: 11 of 12

# SELECT 문

□ SELECT 문 : 하나 또는 그 이상의 테이블로부터 데이터 검색

□ 구문 :

□ SELECT [DISTINCT] 열\_리스트

□ FROM 테이블\_참조

□ [WHERE 탐색\_조건]

□ [ORDER BY 순서\_리스트]

□ 명시된 순서는 검색/출력의 순서를 결정함

[그리 8-20] Song Name Media Type Track Num 바꿀하기 의하 SQL SELECT문

### SQL 문

```
SELECT Song_Name, Media_Type, Track_Num  
FROM Songs;
```

결과:



Song_Name	Media_Type	Track_Num
American Soldier	MP3	3
Beer For My Horses	MP3	5
Big Time	CD	14
Come As You Are	MP3	3
Lithium	MP3	5
One Tree Hill	CD	9
Only a Lad	CD	16
Sledgehammer	CD	3
Sloop John B	Vinyl	7
The Thrill is Gone	MP3	8
Where the Streets Have No Name		
With or Without You	CD	3
		0

# WHERE 절

## □ WHERE 절

- 데이터 검색을 위한 추가적인 기준을 명세
- 검색 필드 리스트에 조건 필드를 포함

## □ AND 와 OR 키워드

- 다중 탐색 기준의 명세를 가능하게 함
- AND는 모든 기준이 만족되어야 함을 나타냄
- OR는 하나의 조건만 만족되어도 됨을 나타냄

[그림 8-21] WHERE 절을 사용한 SQL SELECT 문과 그 결과

### SQL 문

```
SELECT Song_Name, Track_Num  
FROM Songs  
WHERE Media_Type="CD";
```

### 결과:



	Song_Name	Track_Num
	With or Without You	3
	One Tree Hill	9
	Only a Lad	16
	Sledgehammer	3
▶	Big Time	14
*		0

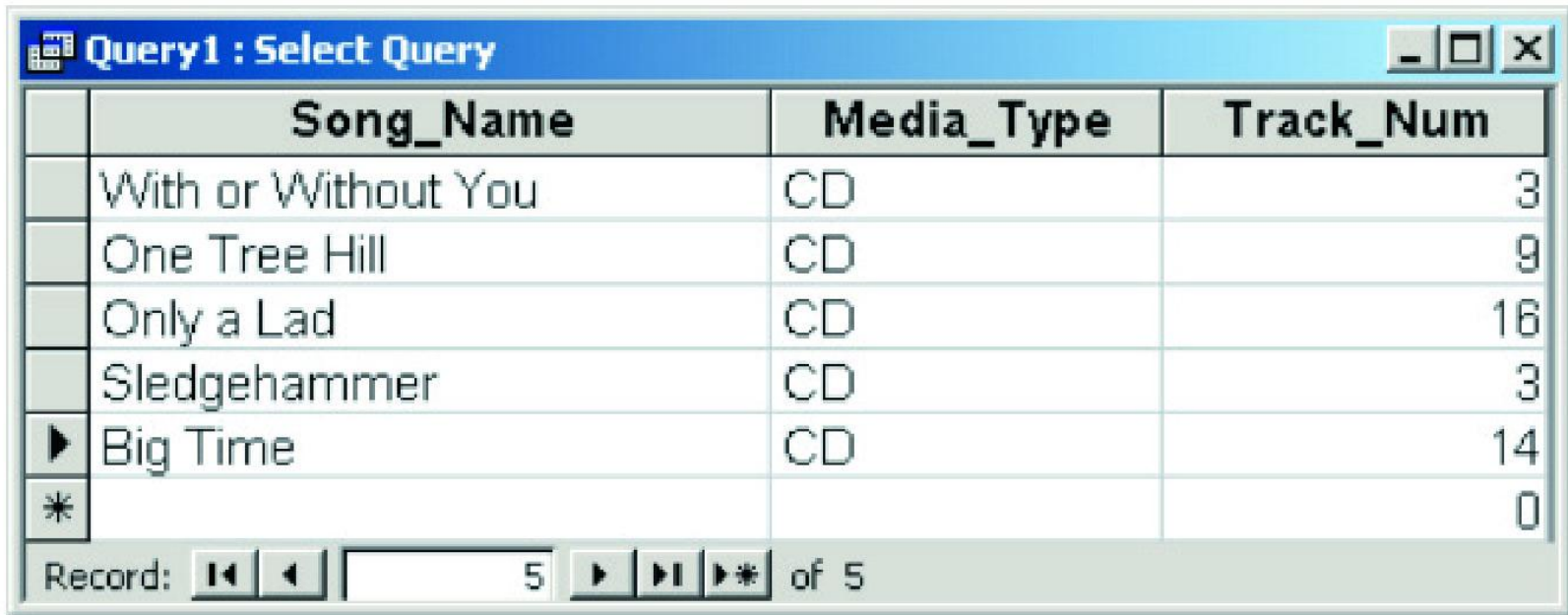
Record: 5 of 5

[그림 8-22] WHERE절을 사용한 보다 세부적인 SQL SELECT 문

## SQL 문

```
SELECT Song_Name, Media_Type, Track_Num  
FROM Songs  
WHERE Media_Type = "CD";
```

결과:



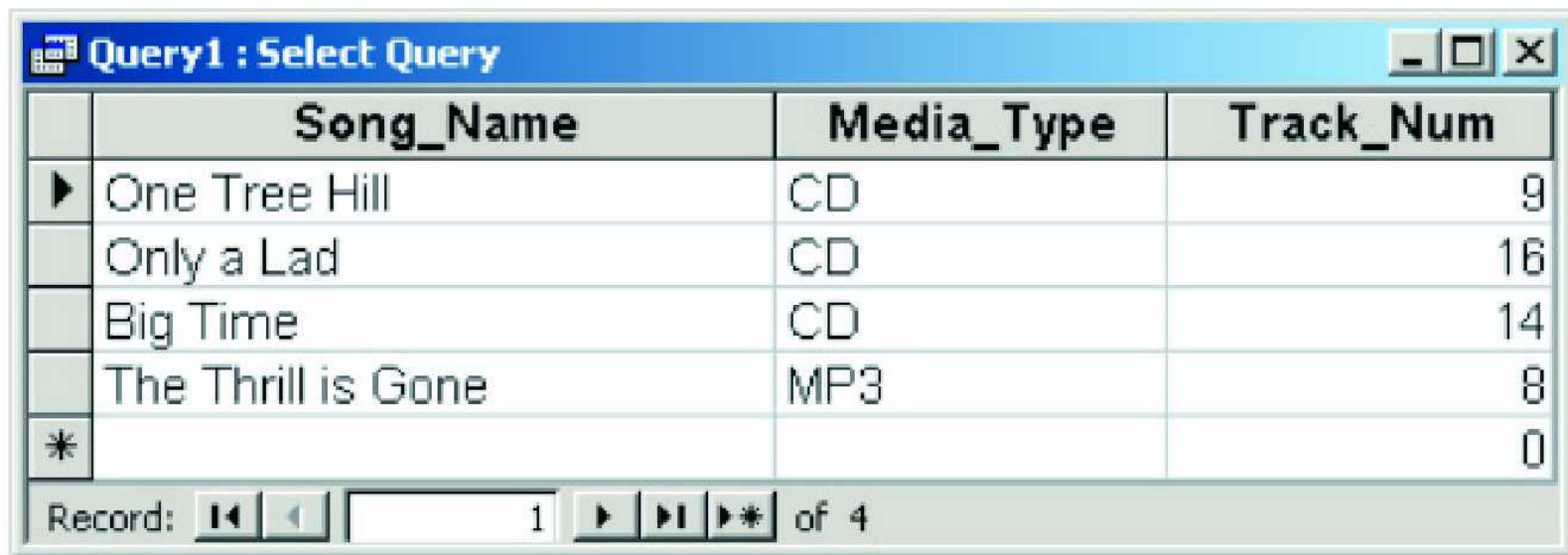
	Song_Name	Media_Type	Track_Num
	With or Without You	CD	3
	One Tree Hill	CD	9
	Only a Lad	CD	16
	Sledgehammer	CD	3
▶	Big Time	CD	14
*			0

Record: 5 of 5

[그림 8-23] OR에 대해서 AND를 가진 WHERE절을 사용하는 SQL SELECT 문  
SQL 문

```
SELECT Song_Name, Media_Type, Track_Num  
FROM Songs  
WHERE(Media_Type = "CD" OR Media_Type = "MP3")  
AND  
Track_Num > 6 ;
```

결과:



	Song_Name	Media_Type	Track_Num
▶	One Tree Hill	CD	9
	Only a Lad	CD	16
	Big Time	CD	14
	The Thrill is Gone	MP3	8
*			0

Record: 1 of 4



# ORDER BY 절

## □ ORDER BY 절

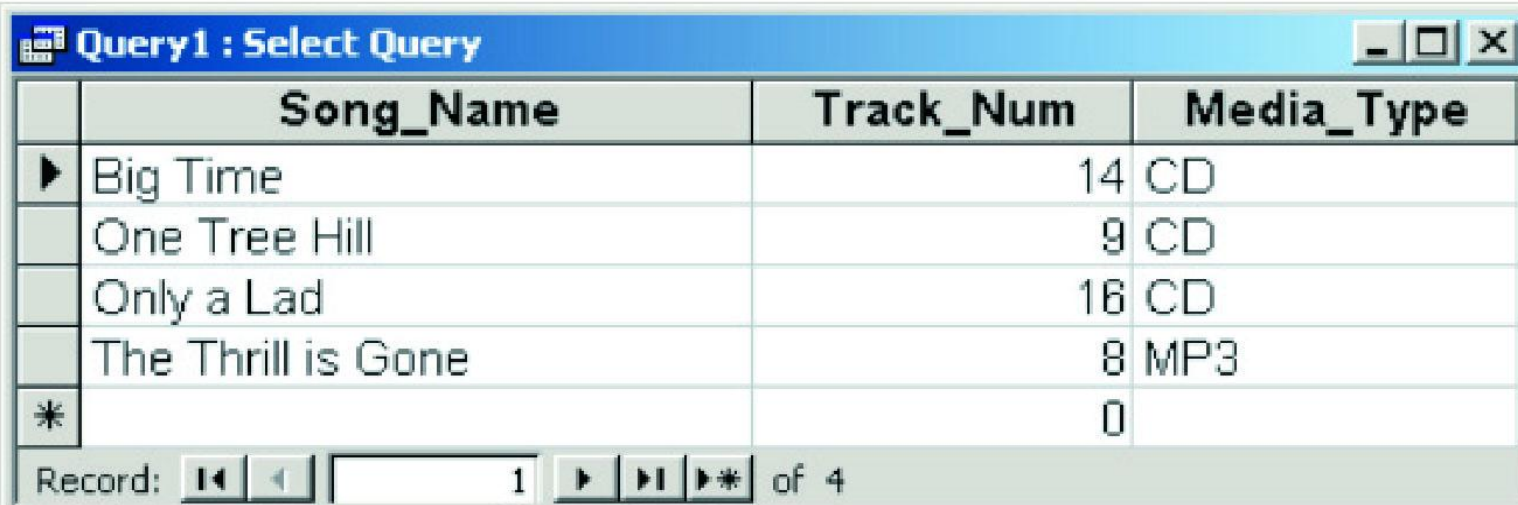
- 데이터가 반환되는 방법의 변경을 가능케 함
- 출력이 더욱 의미 있도록 해줌
- 기본적으로 데이터는 순차적으로 반환됨
- 열 이름(들)에 따라서 순서를 명세 할 수 있음
- ORDER BY는 데이터를 오름차순(기본적임) 또는 내림차순으로 반환함

그림 8-24] ORDER BY적용 사용하 SQL SELECT 문과 그 결과

## SQL 문

```
SELECT Song_Name, Track_Num, Media_Type  
FROM Songs  
WHERE (Media_Type = "CD" OR Media_Type = "MP3")  
AND  
Track_Num > 6  
ORDER BY Song_Name, Track_Num;
```

## 결과:



	Song_Name	Track_Num	Media_Type
▶	Big Time	14	CD
	One Tree Hill	9	CD
	Only a Lad	16	CD
	The Thrill is Gone	8	MP3
*		0	

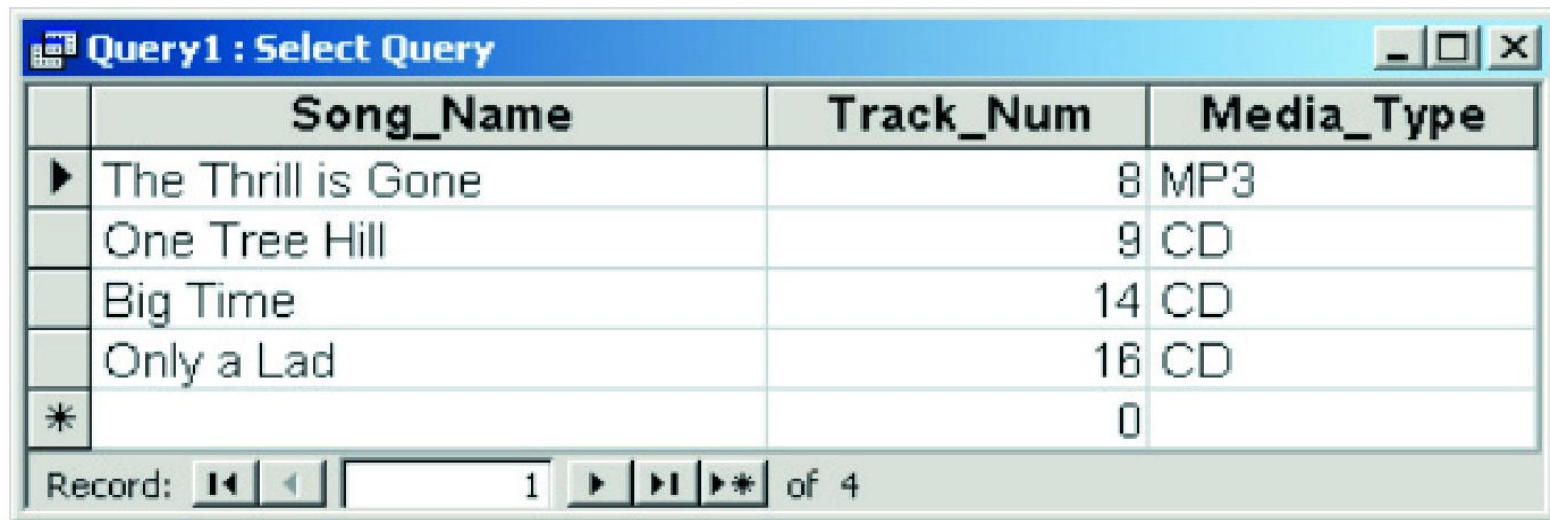
Record: 1 of 4

[그림 8-25] 디폴트 오름차순 선택을 포함한 ORDER BY절을 사용한 SQL SELECT 문

SQL 문

```
SELECT Song_Name, Track_Num, Media_Type  
FROM Songs  
WHERE (Media_Type = "CD" OR Media_Type = "MP3")  
AND  
Track_Num > 6  
ORDER BY Track_Num, Song_Name;
```

결과:



	Song_Name	Track_Num	Media_Type
▶	The Thrill is Gone	8	MP3
	One Tree Hill	9	CD
	Big Time	14	CD
	Only a Lad	16	CD
*		0	

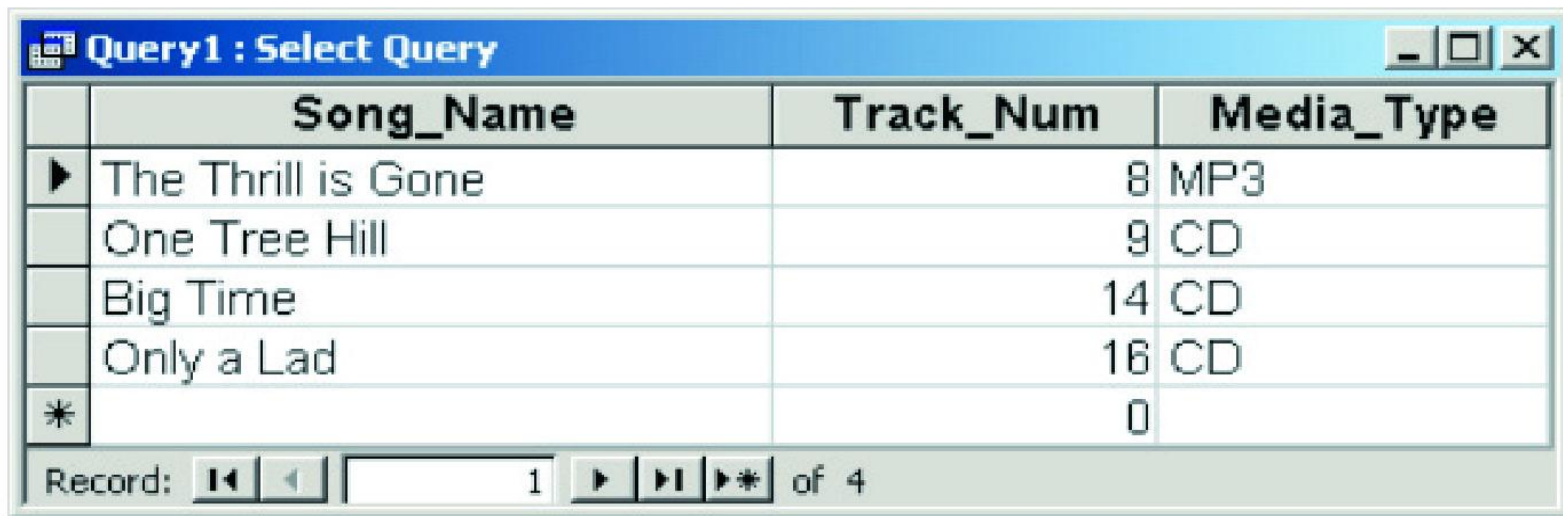
Record: 1 of 4

[그림 8-26] DESC 선택을 포함한 ORDER BY 절을 사용한 SQL SELECT 문

### SQL 문

```
SELECT Song_Name, Track_Num, Media_Type  
FROM Songs  
WHERE (Media_Type = "CD" OR Media_Type = "MP3")  
AND  
Track_Num > 6  
ORDER BY Track_Num DESC, Song_Name;
```

### 결과:



	Song_Name	Track_Num	Media_Type
▶	The Thrill is Gone	8	MP3
	One Tree Hill	9	CD
	Big Time	14	CD
	Only a Lad	16	CD
*		0	

Record: 1 of 4

# ORDER BY 절(계속)

- SELECT 문에 더 많은 선택사항의 명세가 가능함
- 데이터베이스 내의 데이터를 유지, 정의, 관리 하기 위해 더 많은 SQL 문이 사용됨

# 요약

- 데이터베이스 : 논리적으로 연관된 레코드의 집합
- DBMS : 데이터베이스의 설계, 유지, 인터페이스를 위해 사용되는 소프트웨어
- 인덱스 : 기본적으로 순차적인 데이터의 순서를 바꾸는 화일
- 정규화 : 데이터 중복을 제거하는 과정

# 요약(계속)

- 5개의 정규형으로 데이터가 정규화됨
- 처음 세 개의 정규형이 가장 중요함
- 기본키 : 테이블 엔트리를 유일하게 식별함
- 외래키 : 다른 테이블에서 기본키
- 개체 관계 모델 : 테이블과 관계에 대한 시각적 다이어그램

# 요약(계속)

- 1:M 과 1:1 표기법은 카디널리티를 나타냄
- 6단계 데이터베이스 설계과정
- 구조적 질의어(SQL) : 데이터를 조작, 정의 및 관리
- 기본적 SQL 문: CREATE TABLE, INSERT, SELECT