CH 2

Database System Concepts and Architecture

Data Models - Categories

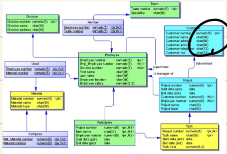
Data Model definition

: collection of concepts that describe the **structure of database**

- Structure
  - **element**, data types
  - groups of elements (**entity**, record, table)
  - **relationship** between groups
- Constraint : restrictions on valid data

Data Model Category

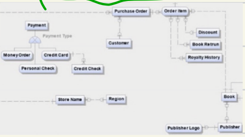
- Physical data model (low-level, internal) : stored data에 대한 detailed concepts



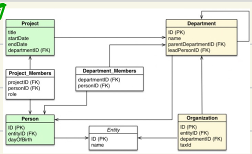
잘 보면 type, access, index 까지 나와있음

⇒ type 명시

- Conceptual data model (high-level, semantic) : user들의 perceive data하는 방법 concept
  - Entity (object) based model
  - ER-diagram ⇒ Entity 관계 명시



- Implementation data model (representational)
  - relational data model



⇒ Table 안에서 schema(description) 명시

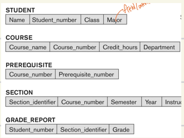
Implementation

Conceptual

Physical

Schema · State

Database Schema



STUDENT

Name	Student_number	Class	Major

field/attr

COURSE

Course_name	Course_number	Credit_hours	Department

PREREQUISITE

Course_number	Prerequisite_number

SECTION

Section_identifier	Course_number	Semester	Year	Instruc

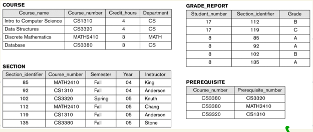
GRADE_REPORT

Student_number	Section_identifier	Grade

: description of database

- not expected to change frequently

Database State



COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

: actual data in database at particular moment in time

(database instance)

- change frequently (every time the db is updated)
- Valid state : satisfies structure·constraint of DB

Three-Schema Architecture

proposed to support some DBMS characteristics

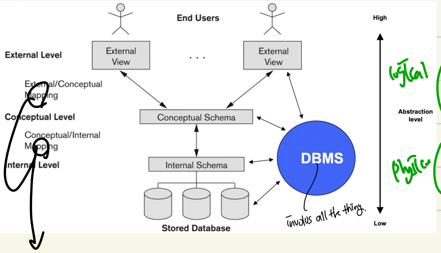
↓ Abstraction

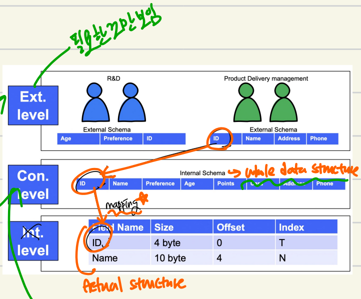
Purpose	: Insulation between application and physical database · define DBMS schemas at three levels
Internal schema	: describe physical storage structure and access paths (index) - Physical data model
Conceptual schema	: describe structure, constraints for whole database for users - Conceptual/implementation data model (ER-diagram) ER diagram relational data model (Relation-Attribute 표시)
External schema	: describe various user views - 보통 conceptual schema와 같은 data model 사용

External

Conceptual

Internal





Field Name	Size	Offset	Index
ID	4 byte	0	T
Name	10 byte	4	N

Mapping for transformation of requests and data

Data Independence

Independent from what? low-level change ↛ high level change

types	✓ • Logical data Independence ✓ • Physical data independence
Physical data Independence	: Internal schema — conceptual schema가 change에 서로 영향 X
Logical data Independence	: Conceptual schema — external schema가 change에 영향 안받음 예) conceptual schema 에서 data의 Attribute를 더 추가해도 external 에서는 영향 X (해당 Attribute가 필요없는 한!) • 즉, lower level change는 higher level에 영향 X

DBMs Language · Interface

DDL (Data Definition Language) : DBA conceptual schema 표현하기 위해서 사용

- define Interanl / External view

CREATE TABLE, DROP INDEX

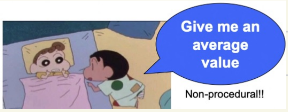
DML (Data Manipulation Language) : specify DB retrievals / updates

SELECT, INSERT, UPDATE, DELETE

type

· High-level / Non-procedure language

: declarative language



· Low-level / Procedure language

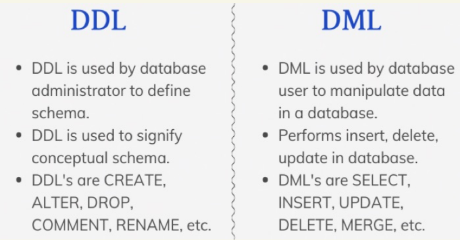
: tell system every details . . .

Add the values, then, divide that total by the number of value.
Procedural!!

DCL (Data Control Language) : Control the user access to DB

❑ GRANT (gives user's access privileges to DB)

❑ REVOKE (withdraw access privileges)



DDL	DML
• DDL is used by database administrator to define schema. • DDL is used to signify conceptual schema. • DDL's are CREATE, ALTER, DROP, COMMENT, RENAME, etc.	• DML is used by database user to manipulate data in a database. • Performs insert, delete, update in database. • DML's are SELECT, INSERT, UPDATE, DELETE, MERGE, etc.

Centralized Client-server DBMS Architecture

Centralized DBMS : Combines everything into single system

DBMS SW, HW, Programs, UI …

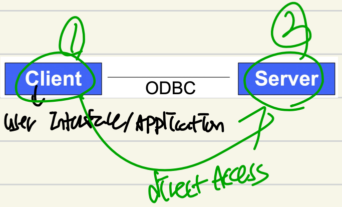
- User can connect through remote terminal
- All processing is done at centralized site

장점
- easy to Manage (only 1 site)

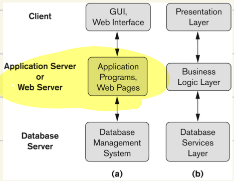
단점
- bottleneck from multi-user accessing
- productivity is limited … All the computing is done at one-side

specialized server : with specific functionalities

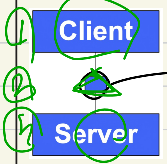
two-tier client/server Architecture : client ⊕ server



장점 · simple.



three-tier client/server Architecture : client ⊕ Application Server (web server) ⊕ server



database secrity ↑↑ : server request 전에 precheck!

- clients can't directly access DB server
- client = PC/Mobile device connected to Web