**Database System**

**Lecture2**

**IMS (Hierarchical Model)**

- Example Hierarchy

Segment Structure

IMS / DL/1 Operations

- Example PL/1 Program #1

- Example PL/1 Program #2

- What’s Bad About IMS/PL1?

Logical Data Independence

Schemas Change for Many Reasons

Study break #1

Questions

Solution

**CODASYL (Conference/Committee on Data Systems Languages)**

Example CODASYL Network

Example: Find Cages Joe Keeps

Codasyl Problems

Relational Principles

**Relational Data Model**

Zoo Tables

Zoo Tables (last lecture)

Relational Algebra

Join as Cross Product

Relational Identities

Push Down Example

Join Ordering Example

Study Break #2

Questions

Solution

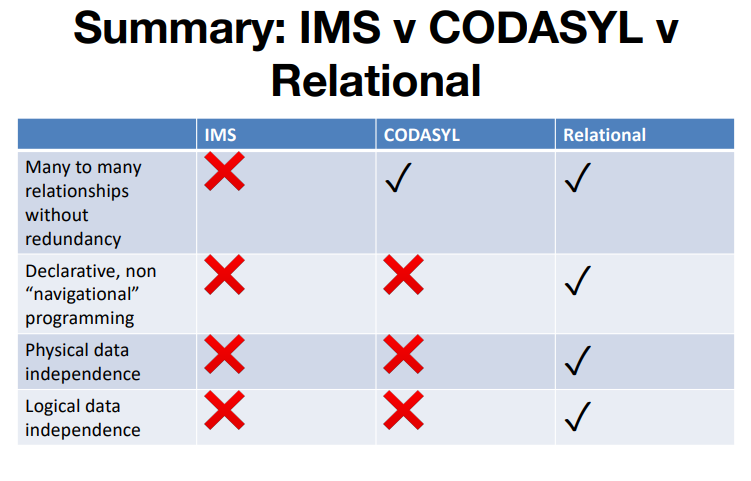
**IMS vs CODASYL vs Relational**

Physical Independence

Logical Data Independence

Key Idea: View

Views Example



**Lecture3**

Aliases and Ambiguity

Aggregation

Left Join?

**Self Join**

Question: Keepers who keep bears and giraffes

Need to build two tables, Bear keepers and Giraffe keepers, and intersect them.

SELECT bear\_keepers.name

FROM keepers AS bear\_keepers

JOIN keeps AS bear\_keeps ON bear\_keepers.id = bear\_keeps.kid

JOIN cages AS bear\_cages ON bear\_keeps.cageno = bear\_cages.no

JOIN animals AS bear\_animals ON bear\_animals.acageno = bear\_cages.no

JOIN keepers AS giraffe\_keepers

JOIN keeps AS giraffe\_keeps ON giraffe\_keepers.id = giraffe \_keeps.kid

JOIN cages AS giraffe\_cages ON giraffe \_keeps.cageno = giraffe \_cages.no

JOIN animals AS giraffe\_animals ON giraffe \_animals.acageno = giraffe \_cages.no

WHERE bear\_animals.species = ‘Bear’

AND giraffe\_animals.species = ‘Giraffe’

AND giraffe\_keepers.id = bear\_keepers.id

But, 7-way join

**Nested Queries**

SELECT bear\_keepers.name

FROM (

SELECT id, keepers.name FROM

keepers JOIN keeps ON id = kid

JOIN cages ON cageno = no

JOIN animals ON acageno = no

WHERE species = ‘Bear’

) AS bear\_keepers

JOIN (

SELECT id, keepers.name FROM

keepers JOIN keeps ON id = kid

JOIN cages ON cageno = no

JOIN animals ON acageno = no

WHERE species = ‘Giraffe’

) AS giraffe\_keepers

ON giraffe\_keepers.id = bear\_keepers.id

**Simplify with Common Table Expression (CTEs)**

WITH bear\_keepers AS (

SELECT id, keepers.name FROM

keepers JOIN keeps ON id = kid

JOIN cages ON cageno = no

JOIN animals ON acageno = no

WHERE species = ‘Bear’

),

giraffe\_keepers AS (

SELECT id, keepers.name FROM

keepers JOIN keeps ON id = kid

JOIN cages ON cageno = no

JOIN animals ON acageno = no

WHERE species = ‘Giraffe’

)

SELECT bear\_keepers.name

FROM bear\_keepers JOIN giraffe\_keepers

ON giraffe\_keepers.id = bear\_keepers.id

SQL can get complex

Study Break

Solution

※ 결과값에서 중복값 제거하려면 SELECT 할때 SELECT DISTINCT

**Recursive Queries**

Recursive Example

**Window Functions**

Other Window Functions

Examples

Study Break

Soln

What is SimpleDB?

Module Diagram

Database

Catalog

BufferPool

HeapFile (Implements DbFile)

HeapPage (Implements Page)

SeqScan (Implements Dblterator)

**Lecture 4**

Entity Relationship Modeling Already Saw with Zoo

**ER Diagrams**

More ER Modeling

Converting to Relations

Study Break

Solution

Hobbies Example

Hobby DB, Attempt 1

Types of Anomalies

Hobby DB Attempt 2

**Normalization**

Schema From ER Diagram

Why Does Redundancy Arise?

Functional Dependencies

FDs are a Property of the Application, Not the Data

Boyce-Codd Normal Form (BCNF)

BCNFify

BCNFify Example for Hobbies

Account, Client, Office

A Dilemma

BCNF vs 3NF

Study Break

Solution