

---

# Chapter# 4

## **An Introduction to Relational Databases**

# Relational systems

---

- Relational systems are based on a theory called Relational model of data.
- Definition of Relational model
  - Open-ended collection of scalar types.
  - Relation type generator.
  - Facilities for defining relation variables.
  - Relational assignment.
  - An open-ended collection of relational operators.

- 
- The three aspects of Relational model of data are:
    1. Structural aspect.
    2. Integrity aspect.
    3. Manipulative aspect.

# Structural aspect.

---

**DEPT:**

DEPT#	DNAME	BUDGET
D1	MBA	10M
D2	EDUCATION	8M
D3	COMPUTER	14M
D4	ELECTRICAL	12M

---

**EMP:**

<b>EMP#</b>	<b>ENAME</b>	<b>DEPT#</b>	<b>SALARY</b>
E1	JOSEPH	D1	57K
E2	LARRY	D2	45K
E3	JOHN	D3	72K
E4	CATHY	D4	60K
E5	ROSE	D5	40K

# Manipulative aspect:

---

**RESTRICT:**

DEPTS WHERE BUDGET > 10M;

**RESULT:**

<b>DEPT#</b>	<b>DNAME</b>	<b>BUDGET</b>
D3	COMPUTER	14M
D4	ELECTRICAL	12M

---

## JOIN:DEPTS AND EMPS OVER DEPT#;

DEPT#	DNAME	BUDGET	EMP#	ENAME	SALARY
D1	MBA	10M	E1	JOSEPH	57K
D1	MBA	10M	E2	LARRY	45K
D2	EDUCATION	8M	E3	JOHN	72K
D3	COMPUTER	14M	E4	CATHY	60K
D4	ELECTRICAL	12M	E5	ROSE	40K

---

PROJECT:DEPTS OVER DEPT#,BUDGET;

**RESULT:**

<b>DEPT#</b>	<b>BUDGET</b>
D1	10M
D2	8M
D3	14M
D4	12M

---

**DELETE:** DELETE EMP WHERE EMP#='E4';

**RESULT:**

EMP#	ENAME	DEPT#	SALARY
E1	JOSEPH	D1	57K
E2	LARRY	D1	45K
E3	JOHN	D2	72K
E5	ROSE	D4	40K

# Remarks on the previous examples

---

- The result of each of the three operations is another table.
- Operators derive tables from tables.
- Closure property.
- Output from one operation can become input to another.
- Nested expressions.

# Intermediate results.

---

- Example: restriction of join.
- Materialized evaluation: Intermediate results are fully evaluated.
- Pipelined evaluation: Intermediate results are not fully materialized.

# Set processing.

---

- Relational operators are set-at-a-time: operands and results are whole tables. Tables contains set of rows.
- Non-relational systems are row-at-a-time.
- Relational systems have set processing capability.

# Logical vs. Physical structure.

---

- Tables are the logical structure in the relational systems.
- At the physical level the data data can be in any format.
- There must be a mapping from physical to logical.
- Tables are the abstraction of the stored data at the physical level.

# Information principle.

---

- The entire information content of the database is presented in one and only one way (tables).
- There is no pointer connecting one table to another.
- If there is a relation (in E/R notation) between two tables it will be implicit.
  - Example employee working at a department.
  - In non-relational systems such information is represented using a pointer.
  - At the physical level of the relational model pointers may be used.

# Integrity constraints.

---

- Examples:
  - Employee salary in the range 25k to 95k.
  - Department budget in the range 1M to 15M
- Primary keys
  - DEPT# in DEPT table
- Foreign keys.
  - DEPT# in EMP table.

# Relations and Relvars.

---

- DEPT and EMP are relational variables (Relvars)
- Values of Relvars are relation values
- Example:
  - DELETE EMP WHERE EMP#='E4'.
  - The old relation value of EMP has been replaced by an entirely new relation value.

- 
- The delete operation is a relational assignment
    - $EMP := EMP \text{ MINUS } (EMP \text{ Where } EMP\# = 'E4')$ .
  - Insert and Update operations are relational assignments.
  - Relation variable(Relvar) VS. Relation value.
  - The term Relvar is not in common use.

# What relations means?

---

- The heading of a relation is a **‘predicate’**.
- The body of the relation is a **‘true proposition’**
- Example:
  - **Predicate** : Department **DEPT#** is named **DNAME** and has a budget **BUDGET**.
  - **True proposition**: Department **D1** is named **MBA** and has a **BUDGET** of **10M**

---

<b>EMP#:: EMP#</b>	<b>ENAME:: ENAME</b>	<b>DEPT#:: DEPT#</b>	<b>SALARY:: MONEY</b>
E1	JOSEPH	D1	57K
E2	LARRY	D1	45K
E3	JOHN	D2	72K
E4	CATHY	D3	60K
E5	ROSE	D4	40K

# Types and Relations.

---

- The relational model includes an open-ended set of data types(system defined plus user defined).
- Types are sets of things we can talk about.
- Relations are sets of things we say about the things we can talk about.

# Optimization

---

- Operations such as restrict,project,join are set-level.
- Relational languages are non-procedural.
- Users specify what they want(not how).
- Users request is satisfied automatically by the system.
- Automatic navigation
- Manual navigation.

# Example of automatic vs. manual navigation

MOVE 'S4' TO S# IN S

FIND CALC S

ACCEPT S-SP-ADDR FROM S-SP  
CURRENCY

FIND LAST SP WITHIN S-SP

While SP found PERFORM

ACCEPT S-SP-ADDR FROM S-SP

FIND OWNER WITHIN P-SP

GET P

IF P# IN P < 'P3'

Leave loop

END-IF

FIND PRIOR SP WITHIN S-SP

END PERFORM

MOVE 'P3' TO P# IN P

FIND CALC P

ACCEPT P-SP-ADDR FROM P-SP  
CURRENCY

FIND LAST SP WITHIN P-SP

While SP found PERFORM

ACCEPT P-SP-  
ADDR FROM P-SP

FIND OWNER  
WITHIN S-SP

GET S

IF S# IN S < 'S4'

Leave loop

END-IF

FIND PRIOR SP  
WITHIN P-SP

END PERFORM

MOVE 1000 TO QTY IN SP

FIND DB-KEY IS S-SP-  
ADDR

FIND DB-KEY IS P-SP-  
ADDR

STORE SP

CONNECT SP TO S-SP

CONNECT SP TO P-SP

- **INSERT INTO SP (S#, P#, QTY) VALUES ('S4', 'P3', 1000);**

- 
- Automatic navigation is performed by the optimizer
    - Evaluation strategy
      - Use a physical scan until the required data is found.
      - Use an index on EMP#.

- 
- Manual navigation is performed by the user.
    - Evaluation strategy
      - Which relvars are referenced in the request
      - What indexes exist.
      - How selective those indexes are.
      - How the data is physically clustered on the disk.
      - What relational operators are involved.

# Catalog

---

- Various schemas (external, conceptual, internal)
- All mappings (external/conceptual, conceptual/internal)
- Metadata: data regarding data

<b>Table</b>	<b>TABNAME</b>	<b>COLCOUNT</b>	<b>ROWCOUNT</b>	⋮ ⋮
	DEPT	3	5	⋮ ⋮
	EMP	4	4	⋮ ⋮

---

**COLUMN**

<b>TABNAME</b>	<b>CLONAME</b>	<b>.....</b>
DEPT	DEPT#	
DEPT	DNAME	
DEPT	BUDGET	
EMP	EMP#	
EMP	ENAME	

- 
- The catalog consists of Relvars (System Relvars)
  - Catalog contains metadata for:
    - Relvars
    - Indexes
    - Users
    - Integrity constraints
    - Security constraints

- 
- Examples on system relvars
    - (Column where tabname='DEPT#'){colname}
    - (Column where colname = 'EMP'){tabname}
    - ((table join column) where colcount<5){tabname,colname}

# Base relvars and views

---

- The original relvars are called base relvars (such as DEPT and EMP).
- The relations corresponding to base relvars are called base relations.
- Relations that can be obtained from base relations using some relational expression are called derived relations.
- Creation of base relvar
  - ```
CREATE TABLE EMP (EMP# VARCHAR2(10)
                    ENAME VARCHAR2(20)
                    DEPT# VARCHAR2(10)
                    SALARY VARCHAR2(4));
```

- 
- **VIEW (DERIEVED RELVAR):** is a relvar whose value is a derived relation.
    - **CREATE VIEW TOPEMP AS (EMP WHERE SALARY>40K)**  
**{EMP#,ENAME,SALARY}**

**RESULT:**

| EMP# | ENAME  | DEPT# | SALARY |
|------|--------|-------|--------|
| E1   | JOSEPH | D1    | 57K    |
| E2   | LARRY  | D1    | 45K    |
| E3   | JOHN   | D2    | 72K    |
| E4   | CATHY  | D3    | 60K    |
| E5   | ROSE   | D4    | 40K    |

- 
- TOPEMP is remembered in the catalog and it is not evaluated.
  - For the user it looks like a new relvar. It is just a view.
  - There is no separate copy of TOPEMP.
  - Changes in EMP are instantaneously reflected in TOPEMP and vice versa.

# TRANSACTIONS

---

- A logical unity of work that involves several database operations
  - Operations
    - Begin transaction
    - Commit
    - Rollback

- 
- Example:

```
BEGIN TRANSACTION
```

```
UPDATE account A
```

```
UPDATE account B
```

```
IF everything worked fine
```

```
    then COMMIT
```

```
Else ROLLBACK
```

```
ENDIF
```

# Transaction characteristics

---

- Atomic: Execute in their spell or not execute at all.
- Durable: Once COMMIT is executed, updates to the database are guaranteed.
- Isolated: Updates made by the transaction T1 are not made visible to transaction T2 until T1 executes COMMIT.
- Serializable: The execution of a set of concurrent transactions produces the same result as executing the transactions in some serial order.

# Summary

---

- Relational database
- Relational variables (Relvars)
- Relational system
- Operations: restrict, project, join
- Set-level operations
- Closure property
- Nested relational expressions

# Summary (Cont.)

---

- Relational assignment
- Primary keys
- Foreign keys
- Information Principle
- Predicate
- True proposition
- Automatic navigation

# Summary (Cont.)

---

- Physical data independence
- Catalog
- Base relvar, derived relvar
- Base relation, Derived relation
- Transaction
- Atomic, durable, isolated operation