# The Relational Model and Relational Algebra

*Nothing is so practical as a good theory*

Kurt Lewin, 1945

# The relational model

- Overcame shortcomings of earlier database models
- Has a strong theoretical base
- Codd was the major developer

# Problems with other models

- Programmers worked at a low level of detail
- No commands for multiple record processing
- Little support for ad hoc querying by users

# Objectives of relational model research

- Data independence
  - Logical and physical models are separate
- Communicability
  - A simple model understood by programmers and users
- Set-processing
  - Increase programmer productivity

# Relational model concepts

- Data structures
- Integrity rules
- Operators

# Data structures

- Domain
  - A set of values all of the same data type
  - All the legal values of an attribute
  - Defines what comparisons are legal
  - Only attributes from the same domain should be compared

*The domain concept is rarely implemented*

# Data structures

- Relations
  - A table of n columns and m rows
- A relation's *cardinality* is its number of rows
- A relation's *degrees* is its number of columns
- A relational database is a collection of relations
  - No explicit linkages between tables

*Cardinality is easy to change but not degrees*

# Structures

- Primary key
  - A unique identifier of a row in a relation
  - Can be composite
- Candidate key
  - An attribute that could be a primary key
- Alternate key
  - A candidate key that is not selected as the primary key
- Foreign key
  - An attribute of a relation that is the primary key of a relation
  - Can be composite

# Integrity rules

- Entity integrity
  - *No component of the primary key of a relation can be null*
  - Each row in a relation is uniquely identified
- Referential integrity
  - *A database must not contain any unmatched foreign key values*
  - For every foreign key there is a corresponding primary key

# Operations

- Relational algebra has 8 operators
  - Restrict
  - Project
  - Product
  - Union
  - Intersect
  - Difference
  - Join
  - Divide

# Restrict

- Extracts rows from a single relation

| A | | | |
|---|---|---|---|
| W | X | Y | Z |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# Project

- Extracts columns from a single relation

| A | | | |
|---|---|---|---|
| W | X | Y | Z |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# Product

- Creates a new relation from all possible combinations of rows in two other relations

A

| V | W |
|---|---|
| v1 | w1 |
| v2 | w2 |
| v3 | w3 |

B

| X | Y | Z |
|---|---|---|
| x1 | y1 | z1 |
| x2 | y2 | z2 |

A TIMES B

| V | W | X | Y | Z |
|---|---|---|---|---|
| v1 | w1 | x1 | y1 | z1 |
| v1 | w1 | x2 | y2 | z2 |
| v2 | w2 | x1 | y1 | z1 |
| v2 | w2 | x2 | y2 | z2 |
| v3 | w3 | x1 | y1 | z1 |
| v3 | w3 | x2 | y2 | z2 |

# Union

- Creates a new relation containing rows appearing in one or both relations
- Duplicate rows are automatically eliminated
- Relations must be union compatible

| A | |
|---|---|
| X | Y |
| x1 | y1 |
| x2 | y2 |
| x3 | y3 |

| B | |
|---|---|
| X | Y |
| x2 | y2 |
| x4 | y4 |

| A UNION B | |
|---|---|
| X | Y |
| x1 | y1 |
| x2 | y2 |
| x3 | y3 |
| x4 | y4 |

# Intersect

- Creates a new relation containing rows appearing in both relations
- Relations must be union compatible

| A | |
|---|---|
| X | Y |
| x1 | y1 |
| x2 | y2 |
| x3 | y3 |

| B | |
|---|---|
| X | Y |
| x2 | y2 |
| x4 | y4 |

| A INTERSECT B | |
|---|---|
| X | Y |
| x2 | y2 |

# Difference

- Creates a relation containing rows in the first relation but not in the second
- Relations must be union compatible

| A | |
|---|---|
| X | Y |
| x1 | y1 |
| x2 | y2 |
| x3 | y3 |

| B | |
|---|---|
| X | Y |
| x2 | y2 |
| x4 | y4 |

| A MINUS B | |
|---|---|
| X | Y |
| x1 | y1 |
| x3 | y3 |

# Join

- Creates a new relation from all combinations of rows satisfying the join condition
- A join B where W = Z

| A | |
|---|---|
| V | W |
| v1 | wz1 |
| v2 | wz2 |
| v3 | wz3 |

| B | | |
|---|---|---|
| X | Y | Z |
| x1 | y1 | wz1 |
| x2 | y2 | wz3 |

| A EQUIJOIN B | | | | |
|---|---|---|---|---|
| V | W | X | Y | Z |
| v1 | wz1 | x1 | y1 | wz1 |
| v3 | wz3 | x2 | y2 | wz3 |

# Divide

- Is there a value in the X column of A (e.g., x1) that has a value in the Y column of A for every value of y in the Y column of B?

| A | |
|---|---|
| X | Y |
| x1 | y1 |
| x1 | y2 |
| x1 | y3 |
| x2 | y1 |
| x2 | y3 |

| B |
|---|
| Y |
| y1 |
| y2 |

| A DIVIDE B |
|---|
| X |
| x1 |

# A primitive set of operators

- Only five operators are required
  - Restrict
  - Project
  - Product
  - Union
  - Difference

# Relational algebra and SQL

- Relational algebra is a standard for judging a data retrieval language

| | Relational algebra | SQL |
|---|---|---|
| Restrict | A where condition | SELECT * FROM A WHERE condition |
| Project | A [X] | SELECT X FROM A |
| Product | A times B | SELECT * FROM A, B |
| Union | A union B | SELECT * FROM A UNION SELECT * FROM B |
| Difference | A minus B | SELECT * FROM A WHERE NOT EXISTS (SELECT * FROM B WHERE A.X = B.X AND A.Y = B.Y AND …) [1] |

1. Essentially, where all columns of A are equal to all columns of B

# A complete relational database

- A fully relational database supports
  - structures (domains and relations)
  - integrity rules
  - a manipulation language
- Most commercial systems are not fully relational because they do not support domains and integrity rules
  - Classified as relationally complete

# Codd's commandments

1. The information rule

    All data must appear to be stored as values in a table

2. The guaranteed access rule

    Every value in a database must be addressable by specifying its table name, column name, and the primary key of the row in which it is stored

3. Systematic treatment of null values

    There must be a distinct representation for unknown or inappropriate data

4. Active on-line catalog on the relational model

    There should be an on-line catalog that describes the relational model

# Codd’s commandments

5. The comprehensive data sublanguage rule

   There must be a relational language that supports data definition, data manipulation, security and integrity constraints, and transaction processing operations

6. The view updating rule

   The DBMS must be able to update any view that is theoretically updateable

7. High-level insert, update, and delete

   The system must support set-at-a-time operations

8. Physical data independence

   Changes to storage representation or access methods will not affect application programs

# Codd’s commandments

9. Logical data independence

   Information preserving changes to base tables will not affect application programs

10. Integrity independence

    Integrity constraints should be part of a database's definition rather than embedded within application programs

    It must be possible to change integrity constraints without affecting any existing application programs

11. Distribution independence

    Introduction of a distributed DBMS or redistributing existing distributed data should have no impact on existing applications

12. The nonsubversion rule

    It must not be possible to use a record-at-a-time interface to subvert security or integrity constraints

# Codd's Rule 0

- A relational DBMS must be able to manage databases entirely through its relational capacities
- A DBMS is either totally relational or it is not relational

# Key points

- The relational model is theoretically grounded and practically relevant
- Relational algebra is the foundation of SQL
- A relational DBMS should satisfy a range of requirements to be fully relational