Relations : 3
n-ary Relations and Their Applications

N Geetha

AM & CS

PSG Tech

n-ary relations

We can have relation between more than just 2 sets

A binary relation involves 2 sets and can be described by a set of pairs
A ternary relation involves 3 sets and can be described by a set of triples
...

An n-ary relation involves n sets and can be described by a set of n-tuples

Relations are used to represent computer databases

Also used to represent constraints!

n-ary Relations

Let $A_1, A_2, \cdots, A_n$ be sets

An $n-ary$ relation is a subset of the cartesian product $A_1 \times A_2 \times \cdots \times A_n$

The sets $A_1, A_2, \cdots, A_n$ are the *domains* of the relation

The degree of the relation is n

An example

Let R be the relation on $N \times N \times N$ consisting of triples (a, b, c) such that $a < b < c$

Note: N is the set of natural numbers {0,1,2,3,...}

$$R = \{(0,1,2), (0,1,3), \cdots, (0,2,3), (0,2,4), \cdots, (1,2,3), \cdots\}$$

$$(2,4,3) \notin R$$

The relation has degree 3

The domains of the relation are the set of natural numbers

Note: R could be considered as an extensional representation of the ternary relation a<b<c, assuming domains are finite and really quite small

An example

Let R be the relation on $N \times Z \times N \times Z$ consisting of 4-tuples (a, b, c, d) such that $(a + b \neq c + d) \wedge (a + b + c + d = 0)$

Note: N is the set of natural numbers {0,1,2,3,...}
Z is the set of integers {...,-2,-1,0,1,2,...}

$(0,-1,1,0) \in R$
$(5,-11,3,3) \in R$
$(6,6,3,9) \notin R$

The relation has degree 4

Note: R could be considered as an extensional representation of the ternary relation a<b<c, assuming domains are finite and really quite small

Relational databases

Database is made up of records.
Typical operations on a database are

- find records that satisfy a given criteria
- delete records
- add records
- update records

Some everyday databases
- student records
- health records
- tax information
- telephone directories
- banking records
- ...

Databases ***may*** be represented using the relational model

Sign in / create account
article | discussion | edit this page | history

Relational database

From Wikipedia, the free encyclopedia

A **relational database** is a database that conforms to the relational model. The term refers to the data, and the structure of that data. The software used to create a relational database is called the Relational Database Management System (RDBMS), but sometimes that software is mistakenly called the relational database.

The term was originally defined and coined by E.F. Codd.[1] Codd's definition is now not the only usage of the term, as many modern DBMS manufacturers have adopted a more relaxed usage of the term.

Contents [hide]

WIKIPEDIA
The Free Encyclopedia
navigation
- Main Page
- Community Portal
- Featured articles
- Current events
- Recent changes
- Random article
- Help
- Contact Wikipedia
- Donations

search
Go Search

toolbox
- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link
- Cite this article

in other languages

Edgar F. Codd - Wikipedia, the free encyclopedia - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites

Address http://en.wikipedia.org/wiki/E.F._Codd Go Links

WIKIPEDIA
The Free Encyclopedia

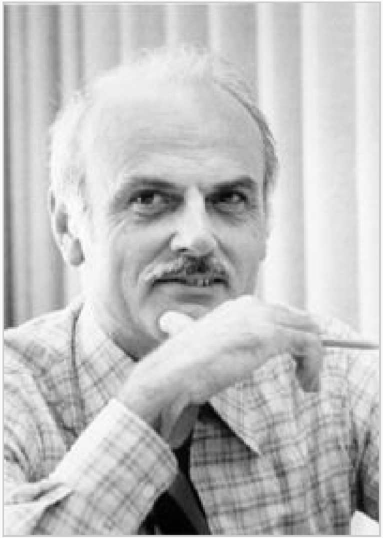
Sign in / create account

article | discussion | edit this page | history

Your continued donations keep Wikipedia running!

Edgar F. Codd

From Wikipedia, the free encyclopedia
(Redirected from E.F. Codd)



Edgar 'Ted' Codd

Edgar Frank "Ted" Codd (August 23, 1923 – April 18, 2003) was a British computer scientist who made seminal contributions to the theory of relational databases. While working for IBM, he created the relational model for database management. He made other valuable contributions to computer science, but the relational model, a very influential general theory of data management, remains his most memorable achievement.

Biography

[edit]

Edgar Frank Codd was born at Portland, Dorset, in England. He studied mathematics and chemistry at Exeter College, Oxford, before serving as a pilot in the Royal Air Force during the Second World War. In 1948, he moved to New York to work for IBM as a mathematical programmer. In 1953, angered by Senator Joseph McCarthy, Codd moved to Ottawa, Canada. A decade later he returned to the USA and received his doctorate in computer science from the University of Michigan in Ann Arbor. Two years later he moved to San Jose, California to work at IBM's Almaden Research Center.

In the 1960s and 1970s he worked out his theories of data arrangement, issuing his paper "A Relational Model of Data for Large Shared Data Banks" in 1970, after an internal IBM paper one year earlier. To his disappointment, IBM proved slow to exploit his suggestions until commercial rivals started implementing them.

Initially, IBM refused to implement the relational model in order to preserve revenue from IMS/DB. Codd then showed IBM customers the potential of the implementation of its model, and they in turn pressured IBM. Then IBM included in its Future System project a System R subproject — but put in charge of it developers who were not thoroughly familiar with Codd's ideas, and isolated the team from Codd. As a

navigation
- Main Page
- Community Portal
- Featured articles
- Current events
- Recent changes
- Random article
- Help
- Contact Wikipedia
- Donations

search

Go Search

toolbox
- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link
- Cite this article

in other languages

Internet

start nAryRelations Algorithmic Fo... Edgar F. Cod... mySlides 12:28

Database made up of *records*, they are *n-tuples*, made up of *fields*

Student record might look as follows

(name,IDNo,Major,Gpa)

Gpa is an attribute

(Jones,200401986,Arts,4.9)
(Lee,200408972,Science,3.6)
(Kuhns,200501728,Humanities,5.0)
(Moore,200308327,Science,5.5)

relations (in relDB) also called *tables*

Name	*IDNo*	*Dept*	*GPA*
Ackermann	231455	Computer Science	3.88
Adams	888323	Physics	3.45
Chou	102147	Computer Science	3.49
Goodfriend	453876	Mathematics	3.49
Rao	678543	Mathematics	3.90
Stevens	786576	Psychology	2.99

Attributes: name, ID No, Dept and GPA

Example from the book

Name	*IDNo*	*Dept*	*GPA*
Ackermann	231455	Computer Science	3.88
Adams	888323	Physics	3.45
Chou	102147	Computer Science	3.49
Goodfriend	453876	Mathematics	3.49
Rao	678543	Mathematics	3.90
Stevens	786576	Psychology	2.99

primary key:
An attribute/domain/column is a primary key when the value of this attribute uniquely defines tuples
i.e. no two tuples have the same value for that attribute
In a database, a primary key should remain unique even if new records are added.

Name cannot be a primary key, neither can Dept or GPA.
IDNo is a primary key

Name	*IDNo*	*Dept*	*GPA*
Ackermann	231455	Computer Science	3.88
Adams	888323	Physics	3.45
Chou	102147	Computer Science	3.49
Goodfriend	453876	Mathematics	3.49
Rao	678543	Mathematics	3.90
Stevens	786576	Psychology	2.99

The current collection of n-tuples (records) in the relation (table) is called ***the extension of the relation***

The permanent aspects of the relation (table) such as the attribute names is called ***the intension of the relation***

Name	*IDNo*	*Dept*	*GPA*
Ackermann	231455	Computer Science	3.88
Adams	888323	Physics	3.45
Chou	102147	Computer Science	3.49
Goodfriend	453876	Mathematics	3.49
Rao	678543	Mathematics	3.90
Stevens	786576	Psychology	2.99

A ***composite key*** is a combination of attributes that uniquely define tuples

Combinations of domains can also uniquely identify n-tuples in an n-ary relation.

When the values of a **set of domains** determine an n-tuple in a relation, the **Cartesian product** of these domains is called a **composite key**.

Name	*IDNo*	*Dept*	*GPA*
Ackermann	231455	Computer Science	3.88
Adams	888323	Physics	3.45
Chou	102147	Computer Science	3.49
Goodfriend	453876	Mathematics	3.49
Rao	678543	Mathematics	3.90
Stevens	786576	Psychology	2.99

Let R be an $n-ary$ relation and C a condition that elements in R must satisfy. The selection operator S_c maps R to the new $n-ary$ relation of all $n-tuples$ from R that satisfy the condition C

Let R be an $n-ary$ relation and C a condition that elements in R must satisfy. The selection operator S_c maps R to the new $n-ary$ relation of all $n-tuples$ from R that satisfy the condition C

Name	*IDNo*	*Dept*	*GPA*
Ackermann	231455	Computer Science	3.88
Adams	888323	Physics	3.45
Chou	102147	Computer Science	3.49
Goodfriend	453876	Mathematics	3.49
Rao	678543	Mathematics	3.90
Stevens	786576	Psychology	2.99

Apply the selection operator S_c
where C is the condition GPA > 3.45

Name	*IDNo*	*Dept*	*GPA*
Ackermann	231455	Computer Science	3.88
Chou	102147	Computer Science	3.49
Goodfriend	453876	Mathematics	3.49
Rao	678543	Mathematics	3.90

Name	*IDNo*	*Dept*	*GPA*
Ackermann	231455	Computer Science	3.88
Adams	888323	Physics	3.45
Chou	102147	Computer Science	3.49
Goodfriend	453876	Mathematics	3.49
Rao	678543	Mathematics	3.90
Stevens	786576	Psychology	2.99

The *projection* $P_{i_1 i_2 \cdots i_m}$ where $i_1 < i_2 < \cdots < i_m$ maps the $n-tuple$ $(a_1, a_2, \cdots, a_n)$ to the $m-tuple$ $(a_{i_1}, a_{i_2}, \cdots, a_{i_m})$ where $m \le n$

It strips out specific columns

The *projection* $P_{i_1 i_2 \cdots i_m}$ where $i_1 < i_2 < \cdots < i_m$ maps the $n-tuple$ $(a_1, a_2, \cdots, a_n)$ to the $m-tuple$ $(a_{i_1}, a_{i_2}, \cdots, a_{i_m})$ where $m \leq n$.

A projection $P_{i_1 i_2 \cdots i_m}$ keeps the m components $a_{i_1}, a_{i_2}, \cdots, a_{i_m}$ of an n-tuple and deletes its (n – m) other components.

Name	*IDNo*	*Dept*	*GPA*
Ackermann	231455	Computer Science	3.88
Adams	888323	Physics	3.45
Chou	102147	Computer Science	3.49
Goodfriend	453876	Mathematics	3.49
Rao	678543	Mathematics	3.90
Stevens	786576	Psychology	2.99

Apply the projection $P_{1,4}$

Name			*GPA*
Ackermann			3.88
Adams			3.45
Chou			3.49
Goodfriend			3.49
Rao			3.90
Stevens			2.99

Projection

•In some cases, applying a projection to an entire table may not only result in fewer columns, but also in **fewer rows**.

•Why is that?

•Some records may only have differed in those fields that were deleted, so they become **identical**, and there is no need to list identical records more than once.

Eg: 2 : Identical rows are not listed

Table 5.3

Course Number	Course Title	Professor	Section Letter
MA 111	Calculus I	P. Z. Chinn	A
MA 111	Calculus I	V. Larney	B
MA 112	Calculus II	J. Kinney	A
MA 112	Calculus II	A. Schmidt	B
MA 112	Calculus II	R. Mines	C
MA 113	Calculus III	J. Kinney	A

Table 5.4

Course Number	Professor	Section Letter
MA 111	P. Z. Chinn	A
MA 111	V. Larney	B
MA 112	J. Kinney	A
MA 112	A. Schmidt	B
MA 112	R. Mines	C
MA 113	J. Kinney	A

Table 5.5

Course Number	Course Title
MA 111	Calculus I
MA 112	Calculus II
MA 113	Calculus III

Join

•Can use the **join** operation to combine two tables into one if they share some identical fields.

•**Definition:** Let R be a relation of degree m and S a relation of degree n. The **join** $J_p(R, S)$, where $p \leq m$ and $p \leq n$, is a relation of degree $m + n - p$ that consists of all

$(m + n - p)$-tuples

$$(a_1, a_2, \ldots, a_{m-p}, c_1, c_2, \ldots, c_p, b_1, b_2, \ldots, b_{n-p}),$$

where

the m-tuple $(a_1, a_2, \ldots, a_{m-p}, c_1, c_2, \ldots, c_p)$ belongs to R

and

the n-tuple $(c_1, c_2, \ldots, c_p, b_1, b_2, \ldots, b_{n-p})$ belongs to S.

Lecturer	*Dept*	*Course*
Cruz	Zoology	335
Cruz	Zoology	412
Faber	Psychology	501
Faber	Psychology	617
Grammer	Physics	544
Grammer	Physics	551
Rosen	Computer Science	518
Rosen	Mathematics	575

Dept	*Course*	*Room*	*Time*
Computer Science	518	N521	14.00
Mathematics	575	N502	15.00
Mathematics	611	N521	16.00
Physics	544	B505	16.00
Psychology	501	A100	15.00
Psychology	617	A110	11.00
Zoology	335	A100	09.00
Zoology	412	A100	08.00

The join operator $J_p(R,S)$ where R and S are $m-ary$ and $n-ary$ relations respectively and $p \leq m$ and $p \leq n$ delivers a new relation of degree $m+n-p$ such that the first $m-p$ attributes come R and the last $n-p$ attributes come from S where the overlapping p attributes match (see Rosen p.534 Defn 4)

Joins two tables/relations together, matching up on specific attributes

Lecturer	*Dept*	*Course*
Cruz	Zoology	335
Cruz	Zoology	412
Faber	Psychology	501
Faber	Psychology	617
Grammer	Physics	544
Grammer	Physics	551
Rosen	Computer Science	518
Rosen	Mathematics	575

Relation R

Dept	*Course*	*Room*	*Time*
Computer Science	518	N521	14.00
Mathematics	575	N502	15.00
Mathematics	611	N521	16.00
Physics	544	B505	16.00
Psychology	501	A100	15.00
Psychology	617	A110	11.00
Zoology	335	A100	09.00
Zoology	412	A100	08.00

Relation S

$$J_2(R, S)$$

Lecturer	*Dept*	*Course*	*Room*	*Time*
Cruz	Zoology	335	A100	09.00
Cruz	Zoology	412	A100	08.00
Faber	Psychology	501	A100	15.00
Faber	Psychology	617	A110	11.00
Grammer	Physics	544	B505	16.00
Rosen	Computer Science	518	N521	14.00
Rosen	Mathematics	575	N502	15.00