

Реляционная модель данных

Базы данных
Виноградова М.В.
МГТУ им.Н.Э. Баумана

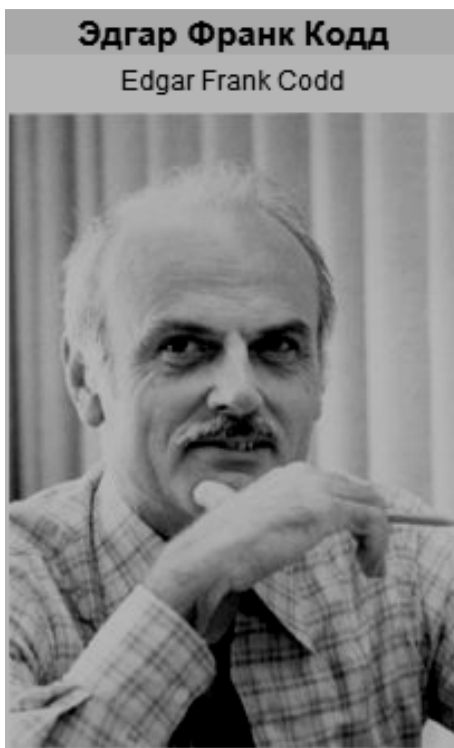
Реляционный подход

- Реляционный подход в настоящее время представляет наиболее развитую идеологию построения БД и БЗ.
- Реляционная модель данных была предложена в начале 70-х годов американским ученым Э.Ф. Коддом, за что в 1981 г. он был удостоен премии Тьюринга Американской ассоциации по вычислительной технике.

Преимущества реляционного подхода

- Теоретическое доказательство свойств БД (нормальные формы)
- Нормализация структуры БД (отсутствие аномалий)
- Абстрактный теоретический язык запросов
- Оптимизация запросов к БД (эквивалентность выражений)

Автор – Э.Ф.Кодд



- Родился в Портланде (Дорсет) в Англии 23 августа 1923г.
- Обучался математике и химии в Оксфордском университете (Exeter College).
- Получил докторскую степень по информатике и вычислительной технике в Университете Мичигана.

Работы Кодда

- В 60-х — 70-х годах работал над теорией хранения данных.
- В 1970 издал работу которая считается первой работой по реляционной модели данных
- **Codd E.F. A relational model of data for large shared data banks //Comm. ACM, v. 13, no. 6, 1970, p. 377-387.**
- **Есть русский пер.: Е.Ф. Кодд. Реляционная модель данных для больших совместно используемых банков данных //СУБД, 1/95, с. 145-160.**

- Одна из нормальных форм названа в его честь (Нормальная форма Бойса — Кодда).
- В начале 80-х началось активное использование в СУБД реляционной модели. Борясь с поставщиками СУБД, чьи продукты неправильно реализуют реляционную технологию, Кодд опубликовал «**12 правил Кодда**», описывающие, что должна содержать реляционная СУБД.

- Кодд ввёл в оборот термин OLAP и написал 12 законов аналитической обработки данных.
- В 1976 Кодд получил почетное звание IBM Fellow. В 1981 он получил премию Тьюринга. В 2002 журнал Forbes поместил реляционную модель данных в список **важнейших инноваций последних 85 лет.**
- Э. Кодд был дном из первых специалистов, который формально определил понятие модели данных. В статье «Модели данных в управлении базами данных»:
- **Codd E.F. “Data Models in Database Management Proc. Workshop in Data Abstraction, Databases, and Conceptual Modelling (Michael L. and N. Zilles, eds.), Pinge Park, Colo. (June 1980): ACM SIGART Newsletter No. 74 (January 1981); ACM SIGMOD Record 11(2), February 1981; ACM SIGPLAN Notices 16(1), January 1981.**

Домен

- В основе реляционной модели лежит понятие «отношение» (**relation**), представляющее собой подмножество декартова произведения доменов.
- **Домен** – некоторое множество элементов - множество допустимых значений, которые может принимать одно из свойств объекта
- (например, множество целых чисел, или множество простых чисел, или множество имен и т.п.).

Пример трех доменов:

$$D1 = \{A, B\}$$

$$D2 = \{C, D\}$$

$$D3 = \{4, 5, 7\}$$

Описание доменов $D1 \dots Dk$ в общем виде:

$$D_1 = \{d_{1_1}, d_{1_2}, \dots, d_{1_n}, \dots, d_{1_m}\};$$

$$D_2 = \{d_{2_1}, d_{2_2}, \dots, d_{2_n}, \dots, d_{2_m}\};$$

...

$$D_k = \{d_{k_1}, d_{k_2}, \dots, d_{k_n}, \dots, d_{k_m}\}.$$

Декартово произведение доменов

- Декартовым произведением доменов D_1, D_2, \dots, D_k является выражение вида

$$D = D_1 \times D_2 \times \dots \times D_k,$$

- Это множество всех наборов значений (кортежей), состоящих из k элементов – по одному из каждого домена D_i :

$$(d_{1_i}, d_{2_i}, \dots, d_{k_i}).$$

- декартово произведение позволяет получить все возможные комбинации элементов исходных множеств рассматриваемых доменов:

$$D = \{(A, D, 4), (A, D, 5), (A, D, 7), (A, C, 4), \\ (A, C, 5), (A, C, 7), (B, D, 4), (B, D, 5), \\ (B, D, 7), (B, C, 4), (B, C, 5), (B, C, 7)\}$$

Отношение на доменах

- Отношением R на множествах D_1, D_2, \dots, D_k называется подмножество декартова произведения $D_1 \times D_2 \times \dots \times D_k$.
- Отношение R , определенное на множествах D_1, D_2, \dots, D_k (причем не обязательно, чтобы эти множества были различными), есть некоторое множество наборов арности k .
- Элементами отношения являются кортежи:

$$\langle d_{1,i}, d_{2,i}, \dots, d_{k,i} \rangle,$$

таких, что $d_{1,i}$ принадлежит D_1 , $d_{2,i}$ принадлежит D_2 и т.д.:

$$R \subseteq D_1 \times D_2 \times \dots \times D_k$$

Примеры отношений

$$R_0 = \theta, \quad R_0 \subseteq D_1 \times D_2 \times D_3$$

$$R_1 = \{(A, C, 4)\}, \quad R_1 \subseteq D_1 \times D_2 \times D_3$$

$$R_2 = \{(A, C, 4), (B, C, 7)\}, \quad R_2 \subseteq D_1 \times D_2 \times D_3$$

.....

$$R_{12} = \{(A, C, 4), (A, C, 5), (A, C, 7), (A, D, 4), \\ (A, D, 5), (A, D, 7), (B, C, 4), (B, C, 5), (B, C, 7), \\ (B, D, 4), (B, D, 5), (B, D, 7)\}, \quad R_{12} \subseteq D_1 \times D_2 \times D_3$$

Арность кортежа

- Арность кортежа определяет арность отношения.
- Отношения арности 1 часто называют унарным,
- арности 2 – бинарным,
- арности 3 – тернарным, арности k – k -арным.
- Поскольку отношение есть множество, в нем не должны встречаться одинаковые кортежи и, кроме того, порядок кортежей в отношении несуществен

От отношений к таблицам

- В ряде случаев отношение удобно представлять как таблицу, где каждая строка – это кортеж, а каждый столбец соответствует одному и тому же компоненту декартова произведения
- (т.е. в нем могут появляться только элементы из соответствующего домена).
- Таблица, представляющая k -арное отношение R , обладает следующими свойствами:
- каждая строка представляет собой кортеж из k значений, принадлежащих k столбцам;
- порядок столбцов фиксирован $(1, 2, \dots, k)$;
- порядок строк безразличен;
- любые две строки различаются хотя бы одним элементом;
- строки и столбцы таблицы могут обрабатываться в любой последовательности, определяемой применяемыми операциями обработки.

Отношение - Relation

Отношение – основная структура в реляционной модели данных.
Представляется двумерной таблицей.

Отношение состоит из **кортежей** (строк) и **атрибутов** (столбцов, колонок).
Пересечение строки и столбца - **поле**.



Домен (тип данных) – область допустимых значений, которые могут принимать атрибуты. **Поле** – для хранения значения данных.

Именованние столбцов

- Для обращения к столбцу используется его номер.
- Для нумерации столбцов используется индекс – 1, 2, 3, 4, ... (натуральный ряд чисел).
- Это удобно для обсуждения операций реляционной алгебры.
- Однако не всегда целесообразно из-за фиксированного порядка столбцов в отношении (в целом ряде практических приложений возникает необходимость перестановки столбцов в отношениях в любом порядке).
- Переходом от номеров к именам.
- Присвоение столбцам отношений имен делает их порядок в отношении несущественным (т.е. столбец определяется по его имени, а не по порядковому номеру).
- При таком подходе оказываются эквивалентными отношения (например, два отношения, отличающиеся порядком столбцов), которые не были бы эквивалентными при традиционном определении.

Схема отношения

- Столбцы отношения назовем атрибутами и присвоим им имена.
- Каждому атрибуту (имени атрибута) соответствует свой домен (мн-во допустимых значений).
- Список имен атрибутов отношения называется схемой отношения.
- Если отношение называется R и его схема имеет атрибуты A_1, A_2, \dots, A_k , то схема отношения обозначается следующим образом:
 $R(A_1, A_2, \dots, A_k)$.
- Существует аналогия между схемой отношения и структурой таблицы, между кортежем и записью, между отношением и таблицей.

Схема базы данных

- Реляционная БД содержит конечное множество экземпляров отношений. Схему реляционной базы данных можно представить как набор схем отношений:

$$P = \left\{ \begin{array}{l} R_1(A_{11}, A_{12}, \dots, A_{1n_1}); \\ R_2(A_{21}, A_{22}, \dots, A_{2n_2}); \\ \dots \dots \dots \\ R_m(A_{m1}, A_{m2}, \dots, A_{m_{n_m}}). \end{array} \right.$$

Интерпретация отношений

Для представления данных отношение используется двояко:

- для представления набора объектов или сущностей (набор объектов представляет собой группу подобных объектов);
- для представления связей между наборами объектов.

Для представления набора объектов

- атрибуты интерпретируются столбцами отношения,
- множество допустимых значений атрибута – соответствующим доменом.
- Каждый кортеж отношения выполняет роль описания отдельного объекта из набора,
- само отношение – весь набор объектов.

Представления набора объектов

Пусть имеется:

1) набор объектов E

$$E = \{e_1, e_2, \dots, e_i, \dots, e_n\};$$

2) множество атрибутов, описывающих набор

$$A = \{A_1, A_2, \dots, A_j, \dots, A_m\};$$

3) множество значений по каждому атрибуту (которые этот атрибут может принимать):

$$K_1 = \{k_{11}, k_{12}, \dots, k_{1n}\} \text{ по атрибуту } A_1;$$

$$K_2 = \{k_{21}, k_{22}, \dots, k_{2n}\} \text{ по атрибуту } A_2;$$

$$\dots$$
$$K_m = \{k_{m1}, k_{m2}, \dots, k_{mn}\} \text{ по атрибуту } A_m.$$

Интерпретация

При выполнении интерпретации объявляем:

1) 1-й столбец отношения соответствует атрибуту A_1 ; 2-й столбец отношения – атрибуту A_2 ; ...; m -й столбец отношения – атрибуту A_m .

2) каждому атрибуту соответствует домен:

$$D_1 = K_1, D_2 = K_2, \dots, D_m = K_m;$$

3) Отношение описывает набор объектов E :

$$R \subseteq D_1 \times D_2 \times \dots \times D_m = K_1 \times K_2 \times \dots \times K_m$$

Пояснение

- В этом отношении R будет l кортежей – по числу объектов в наборе E .
- Каждый кортеж ri отношения R описывает отдельный объект ei из набора объектов E .
- Если атрибут Ai (или совокупность атрибутов $\{A1, A2, \dots, Az\}$) является ключом, то значение в столбце i (или совокупность значений из столбцов $1, 2, \dots, z$) некоторой строки отношения R однозначно идентифицирует эту строку (кортеж) в данном отношении.
- Таким образом, по значению ключа всегда найдем в отношении кортеж, описывающий интересующий нас объект.
- Например, нас интересует описание объекта Aj . Известно, что значение ключа Aj для этого объекта равно $kj3$. Находим в отношении R строку, у которой в столбце j находится значение $kj3$. Это и будет искомое описание объекта eh .

Представление связей объектов

- Для представления связей между наборами объектов (отношениями) можно также использовать отношение (набор объектов-связей):

$$E_1, E_2, \dots, E_k.$$

- где: E_k - k -ый набор объектов:

$$E_1 = \{e_{11}, e_{12}, \dots, e_{1n_1}, \dots, e_{1n_1}\},$$

$$E_2 = \{e_{21}, e_{22}, \dots, e_{2n_2}, \dots, e_{2n_2}\},$$

$$E_k = \{e_{k1}, e_{k2}, \dots, e_{kn_k}, \dots, e_{kn_k}\}.$$

Конкретный экземпляр связи рассматривается между экземплярами объектов:

$$\{e_{1i_1}, e_{2i_2}, \dots, e_{ki_k}\}$$

$$\begin{cases} e_{1i_1} \in E_1 \\ e_{2i_2} \in E_2 \\ \vdots \\ e_{ki_k} \in E_k \end{cases}$$

Пояснение

- Для построения кортежей ri в отношении R используют идентифицирующие атрибуты объектов (от каждого E_k его идентифицирующий атрибут – A_{ik}):

$R (\underline{A_{i1}}, \underline{A_{i2}}, \dots, \underline{A_{ik}})$.

- Кортеж ri в отношении R в этом случае представляет список экземпляров объектов, участвующих в i – ом экземпляре связи.
- После составного $\underline{A_{i1}}, \underline{A_{i2}}, \dots, \underline{A_{ik}}$ первичного ключа отношения R могут следовать описательные атрибуты связи, если они есть в инфологической модели предметной области.
- Однако использовать такой большой первичный ключ нецелесообразно. Поэтому на практике к отношению добавляют более простой первичный ключ – идентификатор номера строки в таблице (порядковый номер):

$R (\underline{ID}, \underline{A_{i1}}, \underline{A_{i2}}, \dots, \underline{A_{ik}})$.

- Наличие кортежа ri в отношении R указывает, что объекты $e_{1i1}, e_{2i2}, \dots, e_{kik}$ ассоциируют между собой с помощью связи, представляемой отношением R .

Операции над отношениями

- Для получения данных из отношений необходим язык, выполняющий операции над отношениями. Прежде всего запросы на чтение данных.
- Были разработаны три абстрактных теоретических языка:
 - **1) реляционная алгебра;**
 - **2) реляционное исчисление с переменными-кортежами;**
 - **3) реляционное исчисление с переменными-доменами.**
- Языки запросов первого типа – алгебраические языки – позволяют выражать запросы средствами специализированных операторов, применяемых к отношениям.
- Языки второго и третьего типов – языки исчисления – позволяют выражать запросы путем спецификации предиката, которому должны удовлетворять требуемые кортежи (или домены).
- Все эти языки предложил Е.Ф. Кодд. По своей выразительности все три языка оказались эквивалентны между собой.
- Реальные языки обеспечивают не только функции соответствующего теоретического языка или их комбинации, но и реализуют некоторые дополнительные операции (арифметические операции, команды присваивания, печати и т.п.).