

СОДЕРЖАНИЕ

1	Теория проектирования реляционных баз данных: функциональные зависимости, нормальные формы	2
1.1	Функциональные зависимости	3
1.2	Нормальные формы, основанные на функциональных зависимостях	5
2	Реляционная модель данных: структурная, целостная, манипуляционная части. Реляционная алгебра. Исчисление кортежей	9
2.1	Структурная часть реляционной модели	9
2.2	Целостная часть реляционной модели	10
2.3	Манипуляционная часть реляционной модели	11

1 Теория проектирования реляционных баз данных: функциональные зависимости, нормальные формы

Классический подход к проектированию реляционных баз данных заключается в том, что сначала предметная область представляется в виде одного или нескольких отношений, а далее осуществляется процесс *нормализации* схем отношений.

Каждой нормальной форме соответствует некоторый определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору ограничений.

В теории реляционных баз данных обычно выделяется следующая последовательность нормальных форм:

- первая нормальная форма (1НФ или 1NF);
- вторая нормальная форма (2НФ или 2NF);
- третья нормальная форма (3НФ или 3NF);
- нормальная форма Бойса-Кодда (НФБК или BCNF);
- четвертая нормальная форма (4НФ или 4NF);
- пятая нормальная форма, или нормальная форма проекции-соединения (5НФ или 5NF или PJ/NF).

Основные свойства нормальных форм такие:

- каждая следующая нормальная форма в некотором смысле лучше предыдущей;
- при переходе к следующей нормальной форме свойства предыдущих нормальных свойств сохраняются.

Процесс проектирования реляционной базы данных на основе метода нормализации преследует две основные цели:

- избежать избыточности хранения данных;
- устранить аномалии обновления отношений.

В основе метода нормализации лежит *декомпозиция* отношения, находящегося в предыдущей нормальной форме, в два или более отношения, удовлетворяющих требованиям следующей нормальной формы. Считаются правильными такие декомпозиции отношения, которые обратимы, т. е. имеется возможность собрать исходное отношение из декомпозированных отношений без потери информации.

Наиболее важные на практике нормальные формы отношений основываются на фундаментальном в теории реляционных баз данных понятии функциональной зависимости.

1.1 Функциональные зависимости

Определение 1. Пусть R — это отношение, а X и Y — произвольные подмножества множества атрибутов отношения R . Тогда Y **функционально зависимо** от X , что в символическом виде записывается как $X \rightarrow Y \Leftrightarrow \forall$ значение множества X связано в точности с одним значением множества Y .

Левая и правая стороны ФЗ будут называться **детерминантом** и **зависимой частью** соответственно.

Определение 2. Пусть R является переменной-отношением, а X и Y — произвольными подмножествами множества атрибутов переменной-отношения R . Тогда $X \rightarrow Y \Leftrightarrow \forall$ **допустимого значения отношения** $R \forall$ значение X связано в точности с одним значением Y .

Определение 2а. Пусть $R(A_1, A_2, \dots, A_n)$ — схема отношения. Функциональная зависимость, обозначаемая $X \rightarrow Y$ между двумя наборами атрибутов X и Y , которые являются подмножествами R определяет ограничение на возможность существования кортежа в некотором отношении r . Ограничение означает, что для любых двух кортежей t_1 и t_2 в r , для которых имеет место $t_1[X] = t_2[X]$, также имеет место $t_1[Y] = t_2[Y]$.

- Если ограничение на схеме отношения R утверждает, что не может быть более одного кортежа со значением атрибутов X в любом отношении экземпляре отношения r , то X является потенциальным ключом R . Это означает, что $X \rightarrow Y$ для любого подмножества атрибутов Y из R . Если X является потенциальным ключом R , то $X \rightarrow R$.
- Если $X \rightarrow Y$ в R , это не означает, что $Y \rightarrow X$ в R .

Функциональная зависимость является семантическим свойством, т. е. свойством значения атрибутов.

Определение 3. ФЗ $(X \rightarrow Y)$ тривиальная $\Leftrightarrow Y \subseteq X$.

Определение 4. Множество всех ФЗ, которые задаются данным множеством ФЗ S , называется замыканием S и обозначается символом S^+ .

Пусть в перечисленных ниже правилах A , B и C – произвольные подмножества множества атрибутов заданной переменной-отношения R , а символическая запись AB означает $\{ A, B \}$. Тогда правила вывода определяются следующим образом:

- 1) Правило *рефлексивности*: $(B \subseteq A) \Rightarrow (A \rightarrow B)$.
- 2) Правило *дополнения*: $(A \rightarrow B) \Rightarrow AC \rightarrow BC$.
- 3) Правило *транзитивности*: $(A \rightarrow B) \text{ и } (B \rightarrow C) \Rightarrow (A \rightarrow C)$.
- 4) Правило *самоопределения*: $A \rightarrow A$.
- 5) Правило *декомпозиции*: $(A \rightarrow BC) \Rightarrow (A \rightarrow B) \text{ и } (A \rightarrow C)$.
- 6) Правило *объединения*: $(A \rightarrow B) \text{ и } (A \rightarrow C) \Rightarrow (A \rightarrow BC)$.
- 7) Правило *композиции*: $(A \rightarrow B) \text{ и } (C \rightarrow D) \Rightarrow (AC \rightarrow BD)$.
- 8) *Общая теорема объединения*: $(A \rightarrow B) \text{ и } (C \rightarrow D) \Rightarrow (A(C - B) \rightarrow BD)$.

Определение 5. *Суперключ* переменной-отношения R — это множество атрибутов переменной-отношения R , которое содержит в виде подмножества (но не обязательно собственного подмножества), по крайней мере, один потенциальный ключ.

Определение 6. Два множества ФЗ S_1 и S_2 *эквивалентны* \Leftrightarrow они *являются покрытиями* друг для друга, т. е. $S_1^+ = S_2^+$.

Каждое множество ФЗ эквивалентно, по крайней мере, одному *неприводимому* множеству.

Определение 7. Множество ФЗ является *неприводимым* \Leftrightarrow оно обладает всеми перечисленными ниже свойствами.

- Каждая ФЗ этого множества имеет одноэлементную правую часть.
- Ни одна ФЗ множества не может быть устранена без изменения замыкания этого множества.
- Ни один атрибут не может быть устранен из левой части любой ФЗ данного множества без изменения замыкания множества.

Если I является неприводимым множеством, которое эквивалентно множеству S , то проверка выполнения ФЗ из множества I автоматически обеспечит выполнение ФЗ из множества S .

1.2 Нормальные формы, основанные на функциональных зависимостях

Определение 1. Переменная отношения находится в первой нормальной форме (1НФ) тогда и только тогда, когда в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов.

Определение 2. Функциональная зависимость $R.X \rightarrow R.Y$ называется полной, если атрибут Y не зависит функционально от любого точного подмножества X .

Определение 3. Функциональная зависимость $R.X \rightarrow R.Y$ называется транзитивной, если существует такой атрибут Z , что имеются функциональные

зависимости $R.X \rightarrow R.Z$ и $R.Z \rightarrow R.Y$ и отсутствует функциональная зависимость $R.Z \rightarrow R.X$. (При отсутствии последнего требования мы имели бы «неинтересные» транзитивные зависимости в любом отношении, обладающем несколькими ключами.)

Определение 4. Неключевым атрибутом называется любой атрибут отношения, не входящий в состав потенциального ключа (в частности, первичного).

Определение 5. Два или более атрибута взаимно независимы, если ни один из этих атрибутов не является функционально зависимым от других.

Вторая нормальная форма

Определение 6. (В этом определении предполагается, что единственным ключом отношения является первичный ключ.) Отношение R находится во второй нормальной форме (2НФ) в том и только в том случае, когда оно находится в 1НФ, и каждый неключевой атрибут полностью зависит от первичного ключа.

Если допустить наличие нескольких ключей, то определение 6 примет следующий вид:

Определение 6а. Отношение R находится во второй нормальной форме (2NF) в том и только в том случае, когда оно находится в 1НФ, и каждый неключевой атрибут полностью зависит от каждого ключа R .

Третья нормальная форма

Определение 7. (Снова определение дается в предположении существования единственного ключа.) Отношение R находится в третьей нормальной форме (3НФ) в том и только в том случае, если оно находится в 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Если отказаться от того ограничения, что отношение обладает единственным ключом, то определение 3NF примет следующую форму:

Определение 7а. Отношение R находится в третьей нормальной форме (3НФ) в том и только в том случае, если оно находится в 2НФ, и каждый неключевой атрибут не является транзитивно зависимым от какого-либо ключа R .

Нормальная форма Бойса-Кодда

Определение ЗНФ неадекватно при выполнении следующих условий:

- 1) переменная-отношение имеет два (или более) потенциальных ключа;
- 2) эти потенциальные ключи являются составными;
- 3) два или более потенциальных ключей перекрываются (т. е. имеют по крайней мере один общий атрибут).

Поэтому впоследствии исходное определение ЗНФ было заменено более строгим определением нормальной формы Бойса-Кодда (НФБК).

Определение 8. Детерминант - любой атрибут (или группа атрибутов), от которого полностью функционально зависит некоторый другой атрибут.

Определение 9. Отношение R находится в нормальной форме Бойса-Кодда (НФБК) в том и только в том случае, если каждый детерминант является потенциальным ключом.

Многозначные зависимости и четвертая нормальная форма

Формальное определение МЗ. Пусть A, B и C являются произвольными подмножествами множества атрибутов переменной-отношения R. Тогда подмножество B **многозначно зависит** от подмножества A, что символически выражается записью $A \twoheadrightarrow B \Leftrightarrow$ множество значений B, соответствующее заданной паре (значение A, значение C) переменной-отношения R, зависит от A, но не зависит от C.

Определение 4НФ. Переменная-отношение R находится в четвертой нормальной форме (4НФ) тогда и только тогда, когда в случае существования таких подмножеств A и B атрибутов этой переменной-отношения R, для которых выполняется нетривиальная МЗ $A \twoheadrightarrow B$, все атрибуты переменной-отношения R также функционально зависят от атрибута A.

Зависимости соединения и пятая нормальная форма

Определение *зависимости соединения. Пусть R является переменной-отношением, а A, B, \dots, Z - произвольными под-множествами множества ее атрибутов. Переменная-отношение R удовлетворяет зависимости соединения $* A, B, \dots, Z$ (читается «звездочка A, B, \dots, Z ») тогда и только тогда, когда любое допустимое значение переменной-отношения R эквивалентно соединению ее проекций по подмножествам атрибутов A, B, \dots, Z .

Определение 5НФ. Переменная-отношение R находится в пятой нормальной форме (5НФ), которую иногда иначе называют проекционно-соединительной нормальной формой (ПСНФ), тогда и только тогда, когда каждая нетривиальная ЗС в переменной-отношении R подразумевается ее потенциальными ключами.

2 Реляционная модель данных: структурная, целостная, манипуляционная части. Реляционная алгебра. Исчисление кортежей

Реляционная модель состоит из трех частей: *структурной, целостностной и манипуляционной*.

2.1 Структурная часть реляционной модели

Структурная часть описывает, из каких объектов состоит реляционная модель. Основной структурой данных, используемой в реляционной модели, являются нормализованные «n-арные» отношения.

Основными понятиями структурной части реляционной модели являются:

- 1) **тип данных**;
- 2) **домен** — рассматривается как подмножество значений некоторого типа данных, имеющих определенный смысл;
- 3) **атрибут отношения** — пара вида $\langle \text{имя_атрибута}, \text{имя_домена} \rangle$. Имена атрибутов должны быть уникальны в пределах отношения. Часто имена атрибутов отношения совпадают с именами соответствующих доменов;
- 4) **схема отношения** — именованное множество упорядоченных пар $\langle \text{имя_атрибута}, \text{имя_домена} \rangle$. *Степенью* или «*арностью*» схемы отношения является мощность этого множества;
- 5) **схема базы данных** — множество именованных схем отношений;
- 6) **кортеж** — множество упорядоченных пар $\langle \text{имя_атрибута}, \text{значение_атрибута} \rangle$, которое содержит одно вхождение каждого имени атрибута, принадлежащего схеме отношения. Значение атрибута должно быть допустимым значением домена, на котором определен данный атрибут. *Степень* или «*арность*» кортежа совпадает с «*арностью*» соответствующей схемы отношения. В реляционной модели каждый кортеж любого

отношения должен отличаться от любого другого кортежа этого отношения (т.е. любое отношение должно обладать уникальным ключом);

- 7) **отношение** определенное на множестве из n доменов (не обязательно различных), содержит две части: заголовок (схему отношения) и тело (множество из m кортежей). Значения n и m называются соответственно *степенью* и *кардинальностью* отношения;
- 8) **потенциальный ключ** — непустое подмножество множества атрибутов схемы отношения, обладающее свойствами *уникальности* (в отношении нет двух различных кортежей с одинаковыми значениями потенциального ключа) и *неизбыточности* (никакое из собственных подмножеств множества потенциального ключа не обладает свойством уникальности);
- 9) **первичный ключ** — один из выбранных потенциальных ключей. Остальные потенциальные ключи называются **альтернативными**;
- 10) **реляционная база данных** — набор отношений, имена которых совпадают с именами схем отношений в схеме базы данных;

2.2 Целостная часть реляционной модели

Фиксируются 2 базовых требования целостности, которые должны выполняться для любых отношений в любых реляционных базах данных — *целостность сущностей* и *ссылочная целостность*.

Поддержание целостности сущностей обеспечивается средствами СУБД. Это осуществляется с помощью двух ограничений:

- при добавлении записей в таблицу проверяется уникальность их первичных ключей;
- не допускается изменение значений атрибутов, входящих в первичный ключ.

Требования для ссылочной целостности (также на СУБД): для каждого значения внешнего ключа, появляющегося в дочернем отношении, в родительском отношении должен найтись кортеж с таким же значением первичного ключа.

2.3 Манипуляционная часть реляционной модели

Описывает два эквивалентных способа манипулирования реляционными данными — *реляционную алгебру* и *реляционное исчисление*. Реляционная алгебра в явном виде предоставляет набор операций, а реляционное исчисление представляет систему обозначений для определения требуемого отношения в терминах данных отношений.

Реляционная алгебра

Реляционная алгебра является основным компонентом реляционной модели и состоит из 8-ми операторов, составляющих две группы по 4 оператора:

- 1) **Традиционные операции над множествами:** объединение (UNION), пересечение (INTERSECT), разность (MINUS) и декартово произведение (TIMES). Все операции модифицированы, с учетом того, что их операндами являются отношения, а не произвольные множества;
- 2) **Специальные реляционные операции:** ограничение (WHERE), проекция (PROJECT), соединение (JOIN) и деление (DIVIDE BY).

Результат выполнения любой операции реляционной алгебры над отношениями также является отношением — свойство **реляционной замкнутости**.

Если рассматривать свойство реляционной замкнутости строго, то каждая реляционная операция должна быть определена таким образом, чтобы выдавать результат с надлежащим типом отношения (в частности, с соответствующим набором атрибутов или заголовком). Для достижения этой цели вводится новый оператор переименование (RENAME), предназначенный для переименования атрибутов в определенном отношении.

Упрощенный синтаксис выражений:

- `реляционное_выражение ::=`
`унарное_выражение | бинарное_выражение;`
- `унарное_выражение ::=` `переименование | ограничение | проекция;`
- `переименование ::=` `терм RENAME имя_атрибута AS имя_атрибута`
- `терм ::=` `имя_отношения | (реляционное_выражение)`
- `ограничение ::=` `терм WHERE логическое_выражение`
- `проекция ::=` `терм | терм [список_имен_атрибутов]`
- `бинарное_выражение ::=` `проекция | бинарная_операция |`
`реляционное_выражение`
- `бинарная_операция ::=` `UNION | INTERSECT | MINUS | TIMES | JOIN |`
`DIVIDEBY`

Замечания:

- 1) реляционные операторы UNION, INTERSECT и MINUS требуют, чтобы отношения были совместимыми по типу, т. е. имели идентичные заголовки.
- 2) реляционные операторы UNION, INTERSECT, TIMES и JOIN ассоциативны и коммутативны.
- 3) если отношения A и B не имеют общих атрибутов, то операция соединения A JOIN B эквивалентна операции A TIMES B, т. е. в таком случае соединение вырождается в декартово произведение. Такое соединение называют естественным.
- 4) другой допустимый синтаксис для синтаксической категории переименования таков: `(терм RENAME список_переименований)`. Здесь каждый из элементов списка переименований представляет собой выражение `имя_атрибута AS имя_атрибута`.

- 5) Несмотря на большие возможности, предоставляемые операторами реляционной алгебры, существует несколько типов запросов, которые нельзя выразить этими средствами. Для таких случаев необходимо использовать процедурные расширения реляционных языков.

Дополнительные операторы реляционной алгебры: **SEMIJOIN** (полусоединение), **SEMIMINUS** (полувычитание), **EXTEND** (расширение), **SUMMARIZE** (обобщение), **TCLOSE** (транзитивное замыкание), **TRANSFORM**, **ROLLUP**, **CUBE**.

Синтаксис этих операторов выглядит следующим образом.

- <полусоединение> ::= <реляционное выражение> SEMIJOIN <реляционное выражение>;
- <полувычитание> ::= <реляционное выражение> SEMIMINUS <реляционное выражение>;
- <расширение> ::= EXTEND <реляционное выражение> ADD<список добавляемых расширений>
- <добавляемое расширение> ::= <выражение> AS <имя атрибута>
- <обобщение> ::= SUMMARIZE <реляционное выражение> PER <реляционное выражение> ADD <список добавляемых обобщений>
- <добавляемое обобщение> ::= <тип обобщения> [(<скалярное выражение>)] AS <имя атрибута>
- <тип обобщения> ::= COUNT | SUM | AVG | MAX | MIN | ALL | ANY | COUNTD | SUMD | AVGD
- <транзитивное замыкание> ::= TCLOSE <реляционное выражение>

Реляционные сравнения

Реляционная алгебра в том виде, в котором она была изначально определена, не поддерживает прямого сравнения двух отношений (например, проверки

их равенства или того, является ли одно из них подмножеством другого). Это упущение легко исправляется следующим образом. Сначала определяется новый вид условия - реляционное сравнение - со следующим синтаксисом.

```
<реляционное выражение> <отношение> <реляционное выражение>  
<отношение> ::= > -- Собственное супермножество, >=  
-- Супермножество, < -- Собственное подмножество, <= -- Подмножество,  
= -- Равно, <> -- Не равно
```

Исчисление кортежей

Реляционное исчисление является альтернативой реляционной алгебре. Внешне два подхода очень отличаются – исчисление описательное, а алгебра предписывающая, но на более низком уровне они представляют собой одно и то же, поскольку любые выражения исчисления могут быть преобразованы в семантически эквивалентные выражения в алгебре и наоборот.

Выражение исчисления кортежей содержит заключенный в скобки список целевых элементов и выражение WHERE, содержащее формулу WFF («правильно построенную формулу»). Такая формула WFF составляется из кванторов (EXISTS и FORALL), свободных и связанных переменных, литералов, операторов сравнения, логических (булевых) операторов и скобок. Каждая свободная переменная, которая встречается в формуле WFF, должна быть также перечислена в списке целевых элементов.

Синтаксис исчисления кортежей

- объявление-кортежной-переменной ::= RANGE
OF переменная IS список-областей;
- область ::= отношение | реляционное-выражение;
- реляционное-выражение ::= (список-целевых-элементов) [WHERE wff]
- целевой-элемент ::= переменная | переменная.атрибут [AS атрибут]

- `wff ::= условие | NOT wff | условие AND wff | условие OR wff |
IF условие THEN wff | EXISTS переменная (wff) |
FORALL переменная (wff) | (wff)`
- `условие ::= (wff) | компаранд операция-отношения компаранд`

Если `f` и `g` - формулы WFF, то выражение логической импликации `IF f THEN g` также будет формулой WFF с семантикой, идентичной семантике формулы `(NOT f) OR g`

Вычислительные возможности исчисления кортежей

`aggregate_function (expression [, attribute])` где

- `aggregate_function` — это COUNT, SUM, AVG, MAX или MIN (возможны, конечно, и некоторые другие функции);
- `expression` — это выражение исчисления кортежей (вычисляющее отношение);
- `attribute` — это такой атрибут результирующего отношения, по которому подсчитывается итог.