

Теорминимум к экзамену по базам данных

Информационная система – программный комплекс, функции которого состоят в

- 1) поддержке надежного хранения информации в памяти компьютера,
- 2) выполнении специфических для данного приложения преобразований информации и/или вычислений,
- 3) предоставлении пользователям удобного и легко осваиваемого интерфейса.

Файл – именованная область внешней памяти, в которую можно записывать и из которой можно считывать данные.

Файловая система

- 1) программная система, управляющая файлами
- 2) архив файлов, хранящийся во внешней памяти.

Авторизацию доступа к файлам – по отношению к каждому зарегистрированному пользователю данной вычислительной системы для каждого существующего файла указываются действия, которые разрешены или запрещены данному пользователю (так называемый **мандатный** способ защиты – каждый пользователь имеет отдельный мандат для работы с каждым файлом или не имеет его).

Требования к системе управления базами данных (СУБД):

- многофайловая система с ключами доступа
- целостность данных (ограничения ссылочной целостности, общие ограничения целостности)
- язык запросов к БД на уровне пользователя
- транзакции
- журнализация
- многопользовательский режим
- независимый системный компонент

В **модели данных** описывается некоторый набор родовых понятий и признаков, которыми должны обладать все конкретные СУБД и управляемые ими базы данных, если они основываются на этой модели.

Ранние модели данных:

- модель данных инвертированных таблиц
- иерархическая модель данных
- сетевая модель данных

Реляционная модель данных – структурная часть, манипуляционная часть, целостная часть:

структурная часть – основные логические структуры данных, которые могут применяться на уровне пользователя при организации БД, соответствующих данной модели,

манипуляционная часть – спецификация одного или нескольких языков, предназначенных для написания запросов к БД,

целостная часть – специфицирует механизмы ограничений целостности, которые обязательно должны поддерживаться во всех реализациях СУБД, соответствующих данной модели.

Отношение (математике) над множествами $\{T_1, T_2, \dots, T_n\}$ называется подмножество декартова произведения этих множеств, т.е. некоторое множество кортежей $\{\{v_1, v_2, \dots, v_n\}\}$, где $v_i \in T_i$.

Схема БД в реляционной модели данных – это набор именованных заголовков отношений вида $H_i = \{ \langle A_i^1, T_i^1 \rangle, \langle A_i^2, T_i^2 \rangle, \dots, \langle A_i^{n_i}, T_i^{n_i} \rangle \}$. T_i называется доменом атрибута A_i .

Реляционная база данных в каждый момент времени представляет собой набор именованных отношений, каждое из которых обладает заголовком, таким как он определен в схеме БД, и телом. Имя отношения R_i совпадает с именем заголовка этого отношения HR_i .

Тело отношения V_{R_i} – это множество кортежей вида $\{ \langle A_i^1, T_i^1, v_i^1 \rangle, \langle A_i^2, T_i^2, v_i^2 \rangle, \dots, \langle A_i^{n_i}, T_i^{n_i}, v_i^{n_i} \rangle \}$, где $t_i^j \in T_i^j$. Во время жизни БД тела отношений могут изменяться, но все содержащиеся в них кортежи должны соответствовать заголовкам соответствующих отношений.

Первичный ключ – такое минимальное подмножество заголовка отношения, что в любом теле этого отношения, которое может появиться в базе данных, значение первичного ключа в любом кортеже этого тела является уникальным, т.е. отличается от значения первичного ключа в любом другом кортеже.

Внешним ключом отношения R_1 , ссылающимся на отношение R_2 , называется подмножество заголовка HR_1 , которое совпадает с первичным ключом отношения R_2 (с точностью до имен атрибутов).

Ограничение ссылочной целостности реляционной модели данных: в любом теле отношения R_1 , которое может появиться в базе данных, для «не пустого» значения внешнего ключа, ссылающегося на отношение R_2 , в любом кортеже

этого тела должен найтись кортеж в теле отношения R2, которое содержится в базе данных, с совпадающим значением первичного ключа.

Объектно-ориентированная модель данных: модель данных база данных – это набор объектов (контейнеров данных) произвольного типа.

Экстент объектного структурного типа – объект типа множества, элементами которого являются объекты данного атомарного типа. Поскольку такой объект создается неявно, его OID неизвестен, но зато у него имеется имя, явно задающееся в определении и совпадающее с именем атомарного объектного типа.

База данных в истинной реляционной модели – это набор долговременно хранимых именованных переменных отношений, каждая из которых определена на некотором типе отношений. В каждый момент времени каждая переменная отношения базы данных содержит некоторое значение отношения соответствующего типа.

Возможный ключ – это одно из подмножеств заголовка переменной отношения, обладающее свойствами первичного ключа.

Заголовком (или схемой) отношения r (H_r) называется конечное множество упорядоченных пар вида $\langle A, T \rangle$, где A называется именем атрибута, а T обозначает имя некоторого базового типа или ранее определенного домена. По определению требуется, чтобы все имена атрибутов в заголовке отношения были различны.

Кортежем tr , соответствующим заголовку H_r , называется множество упорядоченных триплетов вида $\langle A, T, v \rangle$, по одному такому триплету для каждого атрибута в H_r . Третий элемент – v – триплета $\langle A, T, v \rangle$ должен являться допустимым значением типа данных или домена T .

Телом V_r отношения r называется произвольное множество кортежей tr .

Значением V_r отношения r называется пара множеств H_r и V_r .

Переменной VAR_r называется именованный контейнер, который может содержать любое допустимое значение V_r .

Степенью, или «арностью», заголовка отношения, кортежа, соответствующего этому заголовку, тела отношения, значения отношения и переменной отношения является мощность заголовка отношения.

Схема реляционной базы данных – это набор пар $\langle \text{имя_}VAR_r, H_r \rangle$, включающий имена и заголовки всех переменных отношения, которые определены в базе данных.

Реляционная база данных – это набор пар $\langle VAR_r, H_r \rangle$ (конечно, каждая переменная отношения в любой момент времени содержит некоторое значение-отношение, в частности, пустое)

Первичным ключом переменной отношения является такое подмножество S множества атрибутов ее заголовка, что в любое время значение первичного ключа (составное, если в состав первичного ключа входит более одного атрибута) в любом кортеже тела отношения отличается от значения первичного ключа в любом другом кортеже тела этого отношения, а никакое собственное подмножество S этим свойством не обладает. В вырожденном случае, когда заголовок переменной отношения является пустым множеством, первичный ключ этой переменной отношения состоит из пустого подмножества заголовка.

Язык называется **реляционным**, если он обладает не меньшей выразительностью и мощностью, чем реляционная алгебра или реляционное исчисление.

Фундаментальные свойства отношений

- отсутствие кортежей-дубликатов, существование первичного и возможного ключа
- отсутствие упорядоченности кортежей
- отсутствие упорядоченности атрибутов
- атомарность значений атрибутов

Неопределенное значение:

Если a – это значение некоторого типа данных или NULL, op – любая двуместная «арифметическая» операция этого типа данных (например, +), а lop – операция сравнения значений этого типа (например, =), то по определению:

$a \text{ op } NULL = NULL$

$NULL \text{ op } a = NULL$

$a \text{ lop } NULL = \text{unknown}$

$NULL \text{ lop } a = \text{unknown}$

Unknown – третье значение логического, или булевского, типа, обладающее следующими свойствами:

$NOT \text{ unknown} = \text{unknown}$

$\text{true AND unknown} = \text{unknown}$

$\text{true OR unknown} = \text{true}$

$\text{false AND unknown} = \text{false}$

$\text{false OR unknown} = \text{unknown}$

Конкретный язык манипулирования реляционными БД называется **реляционно-полным**, если любой запрос, формулируемый с помощью одного выражения реляционной алгебры или одной формулы реляционного исчисления, может быть сформулирован с помощью одного оператора этого языка.

Операции реляционной алгебры Кодда:

теоретико-множественные операции:

- объединение (UNION),
- пересечение (INTERSECT),
- вычитание (MINUS),
- взятие расширенного декартова произведения (TIMES)

специальные реляционные операции:

- проекция (PROJECT),
- ограничение (WHERE),
- соединение (Θ -JOIN),
- деление (DIVIDE BY)

особая операция:

- присваивание
- переименование атрибутов (RENAME)

Операция реляционного деления (DIVIDE BY) – операция, имеющая два операнда – бинарное и унарное отношения. Результирующее отношение состоит из унарных кортежей, включающих значения первого атрибута кортежей первого операнда таких, что множество значений второго атрибута (при фиксированном значении первого атрибута) включает множество значений второго операнда.

Декартовым произведением множеств $A\{a\}$ и $B\{b\}$ является такое множество пар $C\{c_1, c_2\}$, что для каждого элемента c_1, c_2 множества C существуют такой элемент a множества A , что $c_1=a$, и такой элемент b множества B , что $c_2=b$.

Операция расширенного декартова произведения (алгебра Кодда). Пусть имеются два отношения $R_1\{a_1, a_2, \dots, a_n\}$ и $R_2\{b_1, b_2, \dots, b_m\}$. Тогда результатом операции R_1 TIMES R_2 является отношение $R\{a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m\}$, тело которого является множеством кортежей вида $\{ra_1, ra_2, \dots, ran, rb_1, rb_2, \dots, rbm\}$ таких, что $\{ra_1, ra_2, \dots, ran\}$ входит в тело R_1 , а $\{rb_1, rb_2, \dots, rbm\}$ входит в тело R_2 .

Проекция (алгебра Кодда). Результатом проекции отношения A на множество атрибутов $\{a_1, a_2, \dots, a_n\}$ (PROJECT $A \{a_1, a_2, \dots, a_n\}$) является отношение с заголовком, определяемым множеством атрибутов $\{a_1, a_2, \dots, a_n\}$, и с телом, состоящим из кортежей вида $\langle a_1:v_1, a_2:v_2, \dots, a_n:v_n \rangle$ таких, что в отношении A имеется кортеж, атрибут a_1 которого имеет значение v_1 , атрибут a_2 имеет значение v_2, \dots , атрибут a_n имеет значение v_n .

Операция соединения (алгебра Кодда). Результатом операции соединения A JOIN B WHERE $comp$ совместимых по взятию расширенного декартова произведения отношений A и B является отношение, получаемое путем выполнения операции ограничения по условию $comp$ расширенного декартова произведения отношений A и B (A JOIN B WHERE $comp$ (A TIMES B) WHERE $comp$).

Операция реляционного деления (алгебра Кодда). Пусть заданы два отношения – A с заголовком $\{a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m\}$ и B с заголовком $\{b_1, b_2, \dots, b_m\}$. Будем считать, что атрибут b_i отношения A и атрибут b_i отношения B ($i = 1, 2, \dots, m$) не только обладают одним и тем же именем, но и определены на одном и том же домене. Назовем множество атрибутов $\{a_j\}$ составным атрибутом a , а множество атрибутов $\{b_j\}$ – составным атрибутом b . После этого будем говорить о реляционном делении «бинарного» отношения $A\{a, b\}$ на унарное отношение $B\{b\}$.

По определению, результатом деления A на B (A DIVIDE BY B) является «унарное» отношение $C\{a\}$, тело которого состоит из кортежей v таких, что в теле отношения A содержатся кортежи v UNION w такие, что множество $\{w\}$ включает тело отношения B . Операция реляционного деления не является примитивной и выражается через операции декартова произведения, взятия разности и проекции. Мы покажем это в следующей лекции.

Операции алгебры A:

- операция реляционного отрицания <NOT> (дополнения)
- операция реляционной конъюнкции (или дизъюнкции)
- операция проекции <REMOVE> (удаления атрибута)

Операция <NOT>

Пусть s обозначает результат операции <NOT> r . Тогда:

$H_s = H_r$ (заголовок результата совпадает с заголовком операнда);

$B_s = \{ts : \text{exists } tr (tr \in B_r \text{ and } ts = tr)\}$ (в тело результата входят все кортежи, соответствующие заголовку и не входящие в тело операнда).

Операция <REMOVE>

Пусть s обозначает результат операции r $\langle \text{REMOVE} \rangle A$.

Для обеспечения возможности выполнения операции требуется, чтобы существовал некоторый тип (или домен) T такой, что $\langle A, T \rangle \text{ Hr}$ (т. е. в состав заголовка отношения r должен входить атрибут A). Тогда:

$H_s = H_r \text{ minus } \{ \langle A, T \rangle \}$, т. е. заголовок результата получается из заголовка операнда изъятием атрибута A ;

$B_s = \{ ts : \text{exists } tr \text{ exists } v (tr \text{ Br and } v \text{ T and } \langle A, T, v \rangle \text{ tr and } ts = tr \text{ minus } \{ \langle A, T, v \rangle \}) \}$, т. е. в тело результата входят все кортежи операнда, из которых удалено значение атрибута A .

Операция $\langle \text{AND} \rangle$

Пусть s обозначает результат операции $r_1 \langle \text{AND} \rangle r_2$.

Для обеспечения возможности выполнения операции требуется, чтобы если $\langle A, T_1 \rangle \text{ Hr}_1$ и $\langle A, T_2 \rangle \text{ Hr}_2$, то $T_1 = T_2$. (Другими словами, если в двух отношениях-операндах имеются одноименные атрибуты, то они должны быть определены на одном и том же типе (домене).) Тогда:

$H_s = H_{r_1} \text{ union } H_{r_2}$, т. е. заголовок результата получается путем объединения заголовков отношений-операндов;

$B_s = \{ ts : \text{exists } tr_1 \text{ exists } tr_2 ((tr_1 Br_1 \text{ and } tr_2 Br_2) \text{ and } ts = tr_1 \text{ union } tr_2) \}$;

Обратите внимание на то, что кортеж результата определяется как объединение кортежей операндов; поэтому:

- если схемы отношений-операндов имеют непустое пересечение, то операция $\langle \text{AND} \rangle$ работает как естественное соединение;
- если пересечение схем операндов пусто, то $\langle \text{AND} \rangle$ работает как расширенное декартово произведение;
- если схемы отношений полностью совпадают, то результатом операции является пересечение двух отношений-операндов.

Операция $\langle \text{OR} \rangle$

Пусть s обозначает результат операции $r_1 \langle \text{OR} \rangle r_2$.

Для обеспечения возможности выполнения операции требуется, чтобы если $\langle A, T_1 \rangle \text{ Hr}_1$ и $\langle A, T_2 \rangle \text{ Hr}_2$, то должно быть $T_1 = T_2$ (одноименные атрибуты должны быть определены на одном и том же типе). Тогда:

$H_s = H_{r_1} \text{ union } H_{r_2}$ (из схемы результата удаляются атрибуты-дубликаты);

$B_s = \{ ts : \text{exists } tr_1 \text{ exists } tr_2 ((tr_1 Br_1 \text{ or } tr_2 Br_2) \text{ and } ts = tr_1 \text{ union } tr_2) \}$;

При этом:

- если у операндов нет общих атрибутов, то в тело результирующего отношения входят все такие кортежи ts , которые являются объединением кортежей tr_1 и tr_2 , соответствующих заголовкам отношений-операндов, и хотя бы один из этих кортежей принадлежит телу одного из операндов;
- если у операндов имеются общие атрибуты, то в тело результирующего отношения входят все такие кортежи ts , которые являются объединением кортежей tr_1 и tr_2 , соответствующих заголовкам отношений-операндов, если хотя бы один из этих кортежей принадлежит телу одного из операндов, и значения общих атрибутов tr_1 и tr_2 совпадают;
- если же схемы отношений-операндов совпадают, то тело отношения-результата является объединением тел операндов.

Операция $\langle \text{RENAME} \rangle$

Пусть s обозначает результат операции $r \langle \text{RENAME} \rangle (A, B)$.

Для обеспечения возможности выполнения операции требуется, чтобы существовал некоторый тип T , такой, что $\langle A, T \rangle \text{ Hr}$, и чтобы не существовал такой тип T , что $\langle B, T \rangle \text{ Hr}$. (Другими словами, в схеме отношения r должен присутствовать атрибут A и не должен присутствовать атрибут B .) Тогда:

$H_s = (H_r \text{ minus } \{ \langle A, T \rangle \}) \text{ union } \{ \langle B, T \rangle \}$, т. е. в схеме результата B заменяет A ;

$B_s = \{ ts : \text{exists } tr \text{ exists } v (tr \text{ Br and } v \text{ T and } \langle A, T, v \rangle \text{ tr and } ts = (tr \text{ minus } \{ \langle A, T, v \rangle \}) \text{ union } \{ \langle B, T, v \rangle \}) \}$, т. е. в кортежах тела результата имя значений атрибута A меняется на B .

Реляционное исчисление – прикладная ветвь формального механизма исчисления предикатов первого порядка. В основе исчисления лежит понятие переменной с определенной для нее областью допустимых значений и понятие правильно построенной формулы, опирающейся на переменные, предикаты и кванторы. В зависимости от того, что является областью определения переменной, различают исчисление кортежей и исчисление доменов.

Простые условия – операции сравнения скалярных значений (значений атрибутов переменных или литерально заданных констант). По определению, простое сравнение является WFF, а WFF, заключенная в круглые скобки, представляет собой простое сравнение.

Сложные WFF – строятся с помощью логических связок NOT, AND, OR и IF ... THEN²⁶ с учетом обычных приоритетов операций (NOT > AND > OR) и возможности расстановки скобок.

EXISTS var (form) – формула принимает значение true в том и только в том случае, если в области определения переменной var найдется хотя бы одно значение (кортеж), для которого WFF form принимает значение true.

FORALL var (form) – формула принимает значение true, если для всех значений переменной var из ее области определения WFF form принимает значение true.

Свободные переменные – все переменные, входящие в WFF, при построении которой не использовались кванторы.

Связанная переменная – если имя переменной использовано сразу после квантора при построении WFF вида EXISTS var (form) или FORALL var (form), то в этой WFF и во всех WFF, построенных с ее участием, var является связанной переменной.

Целевой список – компонент, который определяет набор и имена атрибутов результирующего отношения. Целевой список строится из целевых элементов, каждый из которых может иметь следующий вид:

- var.attr, где var – имя свободной переменной соответствующей WFF, а attr – имя атрибута отношения, на котором определена переменная var;
- var, что эквивалентно наличию подсписка var.attr1, var.attr2, ..., var.attrn, где {attr1, attr2, ..., attrn} включает имена всех атрибутов определяющего отношения;
- new_name = var.attr; new_name – новое имя соответствующего атрибута результирующего отношения.

Выражением реляционного исчисления кортежей называется конструкция вида target_list WHERE WFF.

Условия членства. Если R – это n-арное отношение с атрибутами a_1, a_2, \dots, a_n , то условие членства имеет вид $R(a_{i1} : v_{i1}, a_{i2} : v_{i2}, \dots, a_{im} : v_{im})$ ($m \leq n$), где v_{ij} – это либо литерально задаваемая константа, либо имя доменной переменной. Условие членства принимает значение true в том и только в том случае, если в отношении R существует кортеж, содержащий указанные значения указанных атрибутов.

Функциональная зависимость (FD). Пусть задана переменная отношения R, и X и Y являются произвольными подмножествами заголовка R («составными» атрибутами). В значении переменной отношения R атрибут Y функционально зависит от атрибута X в том и только в том случае, если каждому значению X соответствует в точности одно значение Y. $R.X \rightarrow R.Y$. X является **детерминантом** (определителем) для Y, а Y является зависимым от X

$FD A \rightarrow B$ называется **тривиальной**, если $A \supseteq B$ (т. е. множество атрибутов A включает множество B или совпадает с множеством B).

Замыканием множества FD S является множество $FD S^+$, включающее все FD, логически выводимые из FD множества S.

$FD A \rightarrow C$ называется **транзитивной**, если существует такой атрибут B, что имеются функциональные зависимости $A \rightarrow B$ и $B \rightarrow C$ и отсутствует функциональная зависимость $C \rightarrow A$.

Аксиомы Армстронга. Пусть A, B и C являются (в общем случае, составными) атрибутами отношения R. Множества A, B и C могут иметь непустое пересечение. Для краткости будем обозначать через AB A UNION B. Тогда:

1. если $B \subseteq A$, то $A \rightarrow B$ (рефлексивность);
2. если $A \rightarrow B$, то $AC \rightarrow BC$ (пополнение);
3. если $A \rightarrow B$ и $B \rightarrow C$, то $A \rightarrow C$ (транзитивность).

Расширения Дейта

4. $A \rightarrow A$ (самодетерминированность)
5. если $A \rightarrow BC$, то $A \rightarrow B$ и $A \rightarrow C$ (декомпозиция)
6. если $A \rightarrow B$ и $A \rightarrow C$, то $A \rightarrow BC$ (объединение)
7. если $A \rightarrow B$ и $C \rightarrow D$, то $AC \rightarrow BD$ (композиция)
8. если $A \rightarrow BC$ и $B \rightarrow D$, то $A \rightarrow BCD$ (накопление)

Пусть заданы отношение R, множество Z атрибутов этого отношения (подмножество заголовка R, или составной атрибут R) и некоторое множество FD S, выполняемых для R. Тогда **замыканием Z над S** называется наибольшее множество Z^+ таких атрибутов Y отношения R, что $FD Z \rightarrow Y$ входит в S^+ .

Алгоритм нахождения Z^+ .

Рассмотрим множество T. На первом шаге оно равно Z.

На каждом следующем шаге рассматриваем все связи $A \rightarrow B$ из множества S, и если $A \subseteq T$, то $T = T \cup B$.

Алгоритм завершается, когда T не поменялось на очередном шаге.

Суперключом отношения R называется любое подмножество K заголовка R, включающее, по меньшей мере, хотя бы один возможный ключ R.

Множество FD S2 называется **покрытием множества FD S1**, если любая FD, выводимая из S1, выводится также из S2.

Два множества FD S_1 и S_2 называются **эквивалентными**, если каждое из них является покрытием другого, т. е. $S_1^+ = S_2^+$.

Множество FD S называется **минимальным** в том и только в том случае, когда удовлетворяет следующим свойствам:

1. правая часть любой FD из S является множеством из одного атрибута (простым атрибутом);
2. детерминант каждой FD из S обладает свойством минимальности; это означает, что удаление любого атрибута из детерминанта приводит к изменению замыкания S^+ , т. е. порождению множества FD, не эквивалентного S_3^0);
3. удаление любой FD из S приводит к изменению S^+ , т. е. порождению множества FD, не эквивалентного S .

Минимальным покрытием множества FD S называется любое минимальное множество FD S_1 , эквивалентное S .

Декомпозиция – разбиение путем проецирования.

Декомпозиции без потерь – такие декомпозиции отношения, которые обратимы, т. е. имеется возможность собрать исходное отношение из декомпозированных отношений без потери информации.

Теорема Хита.

Пусть задано отношение r $\{A, B, C\}$ (A, B и C , в общем случае, являются составными атрибутами) и выполняется FD $A \rightarrow B$. Тогда $r = (r \text{ PROJECT } \{A, B\}) \text{ NATURAL JOIN } (r \text{ PROJECT } \{A, C\})$.

Атрибут B **минимально зависит** от атрибута A , если выполняется минимальная слева FD $A \rightarrow B$.

Аномалии обновления – трудности, с которыми приходится сталкиваться при выполнении операций добавления кортежей в отношение (INSERT), удаления кортежей (DELETE) и модификации кортежей (UPDATE).

Неключевым атрибутом называется атрибут, не входящий ни в один возможный ключ

Вторая нормальная форма. Переменная отношения находится во второй нормальной форме (2NF) тогда и только тогда, когда она находится в первой нормальной форме, и каждый неключевой атрибут минимально функционально зависит от первичного ключа. (В определении предполагается, что у отношения имеется только один возможный ключ.)

Третья нормальная форма. Переменная отношения находится в третьей нормальной форме (3NF) в том и только в том случае, когда она находится во второй нормальной форме, и каждый неключевой атрибут нетранзитивно функционально зависит от первичного ключа. (В определении предполагается, что у отношения имеется только один возможный ключ.)

Независимые проекции отношения – такие проекции, которые могут обновляться независимо друг от друга.

Теорема Риссанена

Проекция r_1 и r_2 отношения r являются независимыми тогда и только тогда, когда:

- каждая FD в отношении r логически следует (выводится из аксиом Армстронга) из FD в r_1 и r_2 ;
- общие атрибуты r_1 и r_2 образуют возможный ключ хотя бы для одного из этих отношений.

Нормальная форма Бойса-Кодда

Переменная отношения находится в нормальной форме Бойса-Кодда (BCNF) в том и только в том случае, когда любая выполняемая для этой переменной отношения нетривиальная и минимальная FD имеет в качестве детерминанта некоторый возможный ключ данного отношения.

Многозначная зависимость (MVD). В переменной отношения R с атрибутами A, B, C (в общем случае, составными) имеется многозначная зависимость B от A ($A \twoheadrightarrow B$) в том и только в том случае, когда множество значений атрибута B , соответствующее паре значений атрибутов A и C , зависит от значения A и не зависит от значения C .

Лемма Фейджина

В отношении R $\{A, B, C\}$ выполняется MVD $A \twoheadrightarrow B$ в том и только в том случае, когда выполняется MVD $A \twoheadrightarrow C$.

Теорема Фейджина

Пусть имеется переменная отношения R с атрибутами A, B, C (в общем случае, составными). Отношение R декомпозируется без потерь на проекции $\{A, B\}$ и $\{A, C\}$ тогда и только тогда, когда для него выполняется MVD $A \twoheadrightarrow B \mid C$.

Четвертая нормальная форма. Переменная отношения r находится в четвертой нормальной форме (4NF) в том и только в том случае, когда она находится в BCNF, и все MVD r являются FD с детерминантами – возможными ключами отношения r .

n-декомпозируемое отношение – отношение, которое может быть декомпозировано без потерь на n проекций.

Тривиальная многозначная зависимость. В переменной отношения R с атрибутами (возможно, составными) A и B MVD $A \twoheadrightarrow B$ называется тривиальной, если либо $A \subseteq B$, либо $A \text{ UNION } B$ совпадает с заголовком отношения R .

Зависимость проекции/соединения. Пусть задана переменная отношения R , и A, B, \dots, Z являются произвольными подмножествами заголовка R (составными, перекрывающимися атрибутами). В переменной отношения R удовлетворяется зависимость проекции/соединения (Project-Join Dependency – PJD) $*(A, B, \dots, Z)$ тогда и только тогда, когда любое допустимое значение r переменной отношения R можно получить путем естественного соединения проекций этого значения на атрибуты A, B, \dots, Z .

Подразумеваемая зависимость проекции/соединения. В переменной отношения R PJD *(A, B, ..., Z) называется подразумеваемой возможными ключами в том и только в том случае, когда каждый составной атрибут A, B, ..., Z является суперключом R, т. е. включает хотя бы один возможный ключ R.

Тривиальная зависимость проекции/соединения. В переменной отношения R зависимость проекции/соединения *(A, B, ..., Z) называется тривиальной, если хотя бы один из составных атрибутов A, B, ..., Z совпадает с заголовком R.

Пятая нормальная форма. Переменная отношения R находится в пятой нормальной форме, или в нормальной форме проекции/соединения (5NF, или PJ/NF – Project-Join Normal Form) в том и только в том случае, когда каждая нетривиальная PJD в R подразумевается возможными ключами R.

Сущность (ER) – это реальный или представляемый объект, информация о котором должна сохраняться и быть доступной.

Связь (ER) – это графически изображаемая ассоциация, устанавливаемая между двумя типами сущностей

Атрибутом сущности (ER) является любая деталь, которая служит для уточнения, идентификации, классификации, числовой характеристики или выражения состояния сущности.

Подтипы и супертипы (ER)

Если у типа сущности A имеются подтипы B1, B2,..., Bn, то:

- любой экземпляр типа сущности B1, B2,..., Bn является экземпляром типа сущности A (включение);
- если a является экземпляром типа сущности A, то a является экземпляром некоторого подтипа сущности Bi (i = 1, 2, ..., n) (отсутствие собственных экземпляров у супертипа сущности);
- ни для каких подтипов Bi и Bj (i, j = 1, 2, ..., n) не существует экземпляра, типом которого одновременно являются типы сущности Bi и Bj (разъединенность подтипов).

Супертип – тип сущности, на основе которого определяются подтипы.

Простой тип – тип сущности, не являющийся подтипом и не имеющий подтипов.

Диаграммой классов (UML) – диаграмма, на которой показан набор классов (и некоторых других сущностей, не имеющих явного отношения к проектированию БД), а также связей между этими классами.

Классом (UML) называется именованное описание совокупности объектов с общими атрибутами, операциями, связями и семантикой. Графически класс изображается в виде прямоугольника. У каждого класса должно быть имя (текстовая строка), уникально отличающее его от всех других классов.

Атрибутом класса (UML) называется именованное свойство класса, описывающее множество значений, которые могут принимать экземпляры этого свойства. Класс может иметь любое число атрибутов (в частности, не иметь ни одного атрибута). Свойство, выражаемое атрибутом, является свойством моделируемой сущности, общим для всех объектов данного класса. Так что атрибут является абстракцией состояния объекта. Любой атрибут любого объекта класса должен иметь некоторое значение.

Операцией класса (UML) называется именованная услуга, которую можно запросить у любого объекта этого класса. Описание операции может также содержать ее сигнатуру, т. е. имена и типы всех параметров, а если операция является функцией, то и тип ее значения.

Зависимостью (UML) называют связь по применению, когда изменение в спецификации одного класса может повлиять на поведение другого класса, использующего первый класс.

Связью-обобщением (UML) называется связь между общей сущностью, называемой суперклассом, или родителем, и более специализированной разновидностью этой сущности, называемой подклассом, или потомком.

Полиморфизм по включению: объекты класса-потомка могут использоваться везде, где могут использоваться объекты класса-предка.

Ассоциацией (UML) называется структурная связь, показывающая, что объекты одного класса некоторым образом связаны с объектами другого или того же самого класса.

Кратностью (multiplicity) роли ассоциации называется характеристика, указывающая, сколько объектов класса с данной ролью может или должно участвовать в каждом экземпляре ассоциации.

Агрегатная ассоциация (UML) – ассоциация «часть-целое».

Композитные ассоциации (UML) – такие агрегатные ассоциации, в которых уничтожение «целого» приводит к уничтожению «части».

Инвариант класса (OCL) – условие, которому должны удовлетворять все объекты данного класса. Если говорить более точно, инвариант класса – это логическое выражение, вычисление которого должно давать true при создании любого объекта данного класса и сохранять истинное значение в течение всего времени существования этого объекта.

Транзакция (System R) – последовательность операторов языка, неделимая в смысле состояния базы данных.

Триггер (System R) – это каталогизированная операция модификации, для которой задано условие ее автоматического выполнения.

Представление (System R) – это каталогизированный именованный запрос на выборку данных (из одной или нескольких таблиц).

Индексы (System R). Каждый индекс определяется на одном или нескольких полях таблицы, значения которых составляют его ключ, и позволяет производить прямой поиск по ключу кортежей (их tid'ов) и последовательное сканирование таблицы по индексу, начиная с указанного ключа, в порядке возрастания или убывания значений ключа. Некоторые индексы при их создании могут обладать атрибутом уникальности. В таком индексе не допускаются дубликаты ключа.

Атомарная транзакция – неделимая с точки зрения воздействия на БД последовательность операторов манипулирования данными (чтения, удаления, вставки, модификации), такая, что либо результаты всех операторов, входящих в транзакцию, отображаются в состоянии базы данных, либо воздействие всех этих операторов полностью отсутствует.

Сериальный план выполнения транзакций.

Пусть в системе одновременно выполняется некоторое множество транзакций $S = \{T_1, T_2, \dots, T_n\}$. План (способ) выполнения набора транзакций S (в котором, вообще говоря, чередуются или реально параллельно выполняются операции разных транзакций) называется сериальным, если результат совместного выполнения транзакций эквивалентен результату некоторого последовательного выполнения этих же транзакций $(T_{i1}, T_{i2}, \dots, T_{in})$.

Сериализация транзакций – это механизм их выполнения по некоторому сериальному плану.

Граф ожидания транзакций – это ориентированный двудольный граф, в котором существует два типа вершин – вершины, соответствующие транзакциям (будем изображать их прямоугольниками), и вершины, соответствующие объектам блокировок (будем изображать их окружностями). В этом графе дуги соединяют только вершины-транзакции с вершинами-объектами. Дуга из вершины-транзакции к вершине-объекту существует в том и только в том случае, если для этой транзакции имеется удовлетворенная блокировка данного объекта. Дуга из вершины-объекта к вершине-транзакции существует тогда и только тогда, когда эта транзакция ожидает удовлетворения запроса блокировки данного объекта.

Мягкий сбой – восстановление после внезапной потери содержимого оперативной памяти.

Физически согласованное состояние внешней памяти базы данных – такое состояние БД, при котором наборы страниц всех объектов согласованы, т.е. соответствуют состоянию любого объекта либо после его изменения, либо до изменения.

Точки физической согласованности базы данных – моменты времени, в которые во внешней памяти содержатся согласованные результаты операций, завершившихся до соответствующего момента времени, и отсутствуют результаты операций, которые не завершились, а буфер журнала вытолкнут во внешнюю память.