БД - совокупность данных, отображающих состояние объектов в рассматриваемой предметной области. Сущность - часть реального мира, обладающая характерными свойствами, отличающих ее от других сущностей и позволяющих ее идентифицировать. Объект - нечто существующее 1. Основные определения теории баз данных и различимое. Атрибут - столбец. Домен - совокупность атрибутов. Связь - показывает как экземпляры сущности взаимодействуют между Теорию БД в 2-х словах не опишешь (см. в гугле database 1. Централизованная (один ПК По технологии: все в 1 файле). 2. Распределенная (2+ ПК) 2. Классификация баз данных 1. Файл-сервер - файл хранится на сервере и выдается без обработки. По способу доступа: 2. Клиент-сервер - обработка данных производится на сервере. СУБД - совокупность языковых и программных средств, предназначенных для создания, ведения и использования БД. Иерархические 3. Основные свойства СУБД Реляционные По модели: и баз данных Сетевые БД делятся: Объектно-ориентированные По содержимому По объему Представление БД в виде древовидной Между объектами существуют связи, каждый структуры, состоящей из объектов (данных) объект может включать в себя несколько различных уровней. объектов более низкого уровня. Обязателен один предок, не более. Неэффективна. Недостатки: 4. Иерархическая модель данных Громоздкость. Сбалансированными. Деревья бывают: Несбалансированными. Двоичными (не более 2-х ветвей для одного узла. Состоит из служащих Является служащим Отдел Служащие Руководитель Логическая модель данных, представляющая их Имеет руководителя сетевыми структурами типов записей и связанные отношениями мощности один-к-одному или один-ко-многим. Пример. 5. Сетевая модель данных Быстродействие по сравнению Преимущества: с иерарх. бд. Гибкость - множество связей. Жесткость - поменяешь одну связь, придется перестраивать Недостатки: всю бд. Сложность - сложная структура памяти. Логическая модель данных, прикладная теория построения баз данных. «Реляционный» означает, что теория основана на математическом понятии отношение (relation). На реляционной модели данных строятся реляционные базы данных. Домен - совокупность допустимых значений. 6. Реляционная модель данных Кортеж - таблица, строки. Основные понятия реляционной БД: Кардинальность - количество строк в таблице. Атрибут - поле, столбец таблицы. Степень отношения - количество полей. БД, в которой данные моделируются в виде объектов, их атрибутов, методов и Используются тогда, когда требуется высокопроизводительная обработка данных, классов. имеющих сложную структуру. 7. Объектно-ориентированная В БД хранятся объекты, а не модель данных голые данные как в реляционных В настоящее время для ООБД нет стандартизированного языка запросов и нет математического аппарата, как в реляц. БД (реляц. отношения). Отображение объектов предметной области в абстрактные объекты модели данных таким образом, чтобы это При проектирование БД отображение не противоречило семантике предметной решаются 2 основные области, и было по возможности лучшим (эффективным, проблемы: удобным и т.д.). Обеспечение эффективного выполнения запросов к базе данных, т.е. рациональное расположение данных во внешней памяти, создание полезных дополнительных структур 8. Анализ предметной области при проектировании баз (например, индексов) с учетом особенностей конкретной СУБД. данных Проблема проектирования реляционной базы данных состоит в обоснованном принятии решений о том, из каких отношений (таблиц) должна состоять БД и какие атрибуты (характеристики и свойства) должны быть у этих отношений. Уяснить и указать назначение В ходе анализа предметной области необходимо: базы данных; Определить и выделить первоначальный набор сущностей и атрибутов предметной Данная модель ориентирована на обработку фактографической информации. Данная модель используется при высокоуровневом проектировании баз данных. Во время проектирования баз данных происходит преобразование ER-модели в конкретную схему базы данных на основе выбранной модели данных (реляционной, Моделирование предметной области в этом случае базируется на использовании графических диаграмм - ER диаграмм. объектной, сетевой или др.). Связи между объектами и наборами свойств, и таким образом определить сами ER модель должна определить объекты и взаимосвязи между ними, т.е. установить объекты. связи следующих 2х видов: 9. Модель данных сущность-связь, ER-диаграмма Связи между объектами, задающие характер и функциональную природу их взаимозависимости Сущность - часть реального мира, обладающая характерными свойствами, отличающими ее от других сущностей и позволяющих ее идентифицировать. Существует 3 основных понятия. Связь - показывает как экземпляры сущности взаимодействуют между собой. Атрибут - наименование столбца, единичный домен 1. Один к одному - каждому экземпляру сущности А соответствует один экземпляр Связь - показывает как экземпляры 10. Виды связей модели сущности взаимодействуют между сущности В. Существует 3 вида связей: данных сущность-связь собой. ᆛ 2. Один ко многим - 1А ко многим В. 3. Многие ко многим - много A, много B, все дела. В БД среди множества атрибутов должна быть такая Если ключ содержит 1 атрибут - он совокупность, которая отличает один экземпляр Ключ - минимальный набор атрибутов называется атомарным или элементарным, однозначно идентифицирующий набор сущности от другого, такая совокупность называется если несколько то составным. 11. Ключевые атрибуты ключом. сущностей. Сущность может иметь несколько ключей, их называют возможными. Среди возможных выбирают 1 - первичный. В учебнике на стр. 98+ может быть что-то полезное по этому вопросу. Ранг (арность отношений) - число столбцов в таблице. 1. Объединение. 2. Пересечение 3. Разность 4 12. Отношения и их свойства 4. Декартово произведение Операции над отношениями: 5. Деление 6. Проекция 7. Селекция 8. Соединение 9. Естественное соединение Возвращает отношение, содержащее все кортежи, которые принадлежат либо одному отношению, либо обоим. Бинарная операция для выполнения которой отношения A и В должны иметь одинаковый 13. Операция объединения отношений перечень атрибутов. Проще говоря, соединяются 2 таблицы. Повторяющиеся элементы в результате объединяются и не повторяются. Результатом является отношение с теми же атрибутами что A и B, которое содержит Бинарная операция для выполнения которой отношения А и В должны иметь одинаковый кортежи принадлежащие и А и В. 14. Операция пересечения отношений перечень атрибутов. В результате в отношении будут только повторяющиеся кортежи (строки) Возвращает отношение, содержащее все Бинарная операция для выполнения которой кортежи, которые принадлежат 1-ому из 2-ух отношения А и В должны иметь один перечень заданных отношений и не принадлежат 2-ому. 15. Операция разность отношений атрибутов. Напр., в отношении А∖В возвращаем кортежи, которые есть в А, но нет В. Результатом будет являться отношение в котором содержаться атрибуты и А и В, т.е. количество атрибутов равно сумме атрибутов А и В. Количество же кортежей определяется произведением числа кортежей А и В, т.е. конечное отношение будет содержать все варианты комбинации кортежей обоих Бинарная операция для произвольных отношений A и B. Т.е. с разным количеством атрибутов и кортежей (столбцов и строк) 16. Декартово произведение отношений отношений. Например, отношение А содержит 2 атрибута (столбца) и 4 кортежа (строки). Отношение В - 3 атрибута и 3 кортежа. Результатом АхВ будет отношение имеющее 5 атрибутов и 12 кортежей. Возвращает отношение, содержащее все кортежи из первого унарного отношения, которые содержатся также в бинарном отношении и соответствуют всем кортежам во втором унарном отношении. Деление для 2-ух заданных унарных отношений и 1-го бинарного. Например, делим основную таблицу на какой-то определенный атрибут (или несколько атрибутов) и получаем тиблицу с перечисленными Т.е. из таблицы А берем значение строк, для которых присутствуют все комбинации значений из таблицы В. кортежами. Хороший пример по ссылке Результатом является отношение, содержащее все кортежи заданного отношения, которые остались в этом отношении после исключения их него некоторых Унарная операция, т.е. требуется только одно атрибутов. 18. Операция проекции Например, отношение имеющее 5 атрибутов 4 кортежа. Из него исключаются 2 атрибута. В итоге получается отношение с 3 атрибутами и теми же 4 отношение. кортежами. Вопросы к экзаменам Фактически, мы выкидываем некоторые столбцы. Унарная операция, т.е. требуется только одно Возвращает отношение, содержащее все кортежи из заданного отношения, которые 19. Операция селекции отношение. удовлетворяют указанным условиям. Обычный запрос. Возвращает отношение, содержащее все возможные кортежи, которые представляют собой комбинацию атрибутов двух кортежей, принадлежащих двум заданным, при условии, что в этих двух комбинированных кортежах присутствуют одинаковые значения в одном или нескольких общих для исходных отношений атрибутах. Сначала просто "складываем" 2 таблицы в одну, как при умножении. После должна 20. Операция соединения отношений следовать выборка по одному атрибуту. Обязательно должны присутствовать одинаковые записи в обоих отношениях. Пример: http://migku.wikidot.com/gos-db-16#toc14 Естественным соединением называется соединение по эквивалентности двух отношений R и S, выполненное по всем общим атрибутам, из результатов которого исключается по одному экземпляру каждого общего атрибута. 21. Операция естественного соединения отношений Складываем 2 отношения по одному схожему атрибуту. Обязательно должны присутствовать одинаковые Базы данных атрибуты. 22. Функциональные зависимости Зависимость атрибутов между между атрибутами собой в отношениях. http://www.intuit.ru/department/database/rdbdev/5/ 1. Аксиома рефлексивности. Если В является подмножеством A, то B функционально зависит от A. Eë доказывать не надо, это настолько элементарно, что доказывать не следует. 23. Аксиомы Армстронга Фактически являются правилами вывода. 1 2. Аксиома пополнения. Если есть функц. зав В от А, то существует функциональность зависим АС от ВС. 3. Аксиома транзитивности. Если есть зависимость В от А и С от В, то существует функц. зав С от А. При проектировании баз данных упор в первую очередь делается на достоверность и непротиворечивость хранимых данных, причем эти свойства не должны утрачиваться в процессе работы с данными, т.е. после многочисленных изменений, удалений и дополнений данных по отношению к первоначальному состоянию БД Целью концепции нормализации, созданной в 70х годах, является ликвидация избыточности данных, недопущении Для поддержания БД в устойчивом состоянии используется 24. Нормализация схем аномалий обновления, т.е. сложности, которые возникают при ряд механизмов, которые получили обобщенное название отношений (общие положения) операции удаления, вставки, изменения. средств поддержки целостности. Это является некими ограничениями, которым должна соответствовать БД в процессе создания. В целом суть этих ограничений весьма проста: каждый факт, хранимый в БД, должен храниться один-единственный раз, поскольку дублирование может привести к несогласованности между копиями одной и той же информации. Следует избегать любых неоднозначностей, а также избыточности хранимой информации. Например, если в поле хранится список идентификаторов, разделенных запятыми, то это нарушение данного Первая нормальная форма – любое поле любой записи хранит только одно определения и база не находится в первой нормальной значение. Например, у нас есть запись с полями (Идентификатор, Название CD-Диска, Название группы), где ключом является поле «Идентификатор». При этом, Вторая нормальная форма – БД находится в первой нормальной форме и любое «Идентификатора» но и от поля «Название CD-Диска». Поэтому такая БД не находится во второй нормальной форме. Вторая нормальная форма – БД находится в Третья нормальная форма – БД находится во Например, у нас в записи хранятся код региона и его название (помимо самих полей с информацией о CD-диске). Понятно, второй нормальной форме и нет неключевых полей зависящих от значения других что название региона зависит от кода, и наоборот, поэтому неключевых полей. такая БД не будет находиться в третьей нормальной фоме. Нормальная форма Бойса-Кодда – БД находится в третьей нормальной форме и в любой таблице поля составных ключей не зависят друг от друга. Нетрудно догадаться, что эта нормальная форма является своеобразным расширением третьей формы. Только в третьей форме требовалась независимость неключевых полей, в этой форме дополнительно требуется независимость 25. Нормальные формы Например, у нас есть список записей, в которых помимо прочих полей, есть поля «Идентификатор фирмы» и «Название фирмы» являющиеся составным ключом. Понятно, что они зависят друг от друга. Поэтому данная зависимость является показателем того, что БД не находится в форме ключевых полей. Бойса-Кодда. Например, у нас есть таблица, в которой хранится информация о закупках оборудования – идентификатор записи, название поставщика и название поставляемого оборудования. Четвёртая нормальная форма – БД находится в Поскольку разные поставщики могут поставлять одинаковое оборудование, и наоборот – одно и то же оборудование может поставляться разными поставщиками, то мы имеем две группы независимых данных – поставщики и оборудование. Они повторяющиеся и независимые. Следовательно такая таблица не находится в четвёртой нормальной форме нормальной форме Бойса-Кодда и ни одна таблица не содержит повторяющихся независимых групп данных. Пятая нормальная форма – если БД находится в четвертой нормальной форме, избыточность данных полностью не устранена и разбивка таблицы если и возможна, то только на 3 и более частей. Надо заметить, что реальная необходимость в приведении к пятой форме возникает чрезвычайно редко. Декомпозиция — научный метод, использующий структуру задачи и позволяющий заменить решение одной большой задачи решением серии меньших задач, пусть и взаимосвязанных, но более простых. Декомпозиция отношений проводится, чтобы исключить избыточное дублирование в 26. Декомпозиция схем отношений отношениях. Выделяют два типа декомпозиций Без потерь отношений: С потерями Т.е. в моём понимании декомпозиции это разбиение одного отношения на более мелкие в ходе нормализации. Организации данных - определение и изменение структуры представления данных; установка отношений. Обработки данных -Изначально создавался как добавление/удаление/обновление Сейчас используется для: инструмент для выборки. данных. Управления доступом - ограничения возможностей пользователя; координирование совместного использования. Ну, и конечно же про выборку не забываем. 27. Общие положения об SQL. Типы данных SQL является единственным стандартным языком для работ стандартным языком для работы с реляционными БД. 1. Целые числа (INT). 1. Десятичные числа Фиксированное кол-во знаков 2. Дробные: (NUMERIC, DECIMAL). после запятой. 2. Числа с плавающей запятой (FLOAT). Типы данных: ─ 4. Строки (TEXT, CHAR). 5. Денежные величины (MONEY, CURRENCY). 6. Дата, время (DATE, TIME). 7. Булевы величины (BIT, YESNO). CREATE TABLE <название таблицы> (<имя поля1> <тип данных>, <uмя поля2> <тип данных>); 28. SQL. Создание и удаление таблиц. Таблица с находящимися в ней строками, не может быть удалена. DROP TABLE <название таблицы>; Удалить все строки из таблицы можно следующим DELETE FROM <название таблицы>; способом: ALTER TABLE <имя таблицы> 1. Добавление столбца ADD COLUMN <имя столбца> <тип данных>; ALTER TABLE <имя таблицы> DROP COLUMN <имя столбца> <тип данных>; 2. Удаление столбца Изменения производятся с помощью команды ALTER ALTER TABLE <имя таблицы> Модификация невозможна 29. SQL. Изменение структуры таблицы 3. Модификация столбца ALTER COLUMN <имя столбца> <тип данных>; если имеются какие-то ключи. Основные режимы: ALTER TABLE <имя таблицы>
ADD CONSTRAINT <имя связи>
FOREIGN KEY <имя столбца в другой таблице> 4. Изменение, добавление и удаление ключей. REFERENCES <имя связываемой таблицы> ' ALTER TABLE <имя таблицы> DROP CONSTRAINT <имя удаляемой связи> PRIMARY - первичный ключ CREATE INDEX <uma> ON <uma DISALLOW NULL - запрет заполнения индекса нулевым таблицы> (<имя поля>); WITH {тип Типы индекса - значением IGNORE NULL - возможность заполнения индекса нулевым 30. SQL. Создание и удаление индексов значением. DROP INDEX <имя> ON <имя таблицы> // возможно удаление только тогда, когда индекс является обычным Для удаления первичных ключей и уникальных индексов используется ALTER TABLE с предложением DROP CONSTRAINT: ALTER TABLE <имя таблицы> DROP CONSTRAINT <имя индекса> ALTER TABLE <имя подчинённой таблицы> ADD CONSTRAINT <имя взаимосвязи> FOREIGN KEY (имя поля из подчинённой таблицы) REFERENCES <имя главной таблицы> (имя поля из 31. SQL. Установление связей главной таблицы) между таблицами Запрос на выборку из одной таблицы: SELECT < названия_нужных_полей > FROM < название_таблицы > WHERE < условие_выборки > 32. SQL. Запросы на выборку Запрос на выборку из нескольких таблиц: SELECT * FROM <перечисление таблиц необходимые для выполнения запроса> WHERE <условие выборки> 33. SQL. Запросы на удаление DELETE <имя таблицы> Без WHERE удалятся все WHERE <условие отбора> строки в таблице. UPDATE <имя таблицы> SET <имя столбца> = <выражение> WHERE <условие отбора> 34. SQL. Запросы на изменение T.e. если VALUES (4, 5, qwe), то 4 запишется в первый столбец, 5 - во второй, qwe - в INSERT <имя таблицы> VALUES (<список значений>) третий. 35. SQL. Запросы на добавление INSERT INTO <имя таблицы> (<столбец1>, <столбец2>) VALUES (<значение для столбца1>, <значение для столбца2>) SELECT <имя столбца> FROM <имя таблицы> GROUP BY <название столбца для группировки> 36. SQL. Группировка и SELECT *
FROM <имя таблицы>
ORDER BY <имя упорядочиваемого столбца> упорядочивание данных По возрастанию: ORDER BY <имя упорядочиваемого столбца> ASC По убыванию: ORDER BY <имя упорядочиваемого столбца> DESK SELECT * FROM <имя таблицы> WHERE <условие отбора> 37. SQL. Сортировка данных ORDER BY <имя упорядочиваемого столбца> // Для понимания - Бизнес-логика — в разработке информационных систем — совокупность правил, принципов, Триггер - особый тип процедуры, который автоматически выполняется при изменении таблицы с помощью операторов UPDATE, INSERT или DELETE. Триггеры применяются для зависимостей поведения объектов предметной области (области человеческой деятельности, которую система поддерживает). обеспечения целостности данных и реализации сложной бизнес-логики. UPDATE Соответственно существует три вида триггеров: - INSERT DELETE 38. Понятие о триггерах Триггер запускается сервером автоматически при попытке изменения данных в таблице, с которой он связан. Все производимые им модификации данных рассматриваются как выполняемые в транзакции, в которой выполнено действие, вызвавшее срабатывание триггера. Соответственно, в случае обнаружения // Транзакция — минимальная логически ошибки или нарушения целостности данных может произойти откат этой осмысленная операция, которая имеет смысл и может быть совершена только полностью. CREATE TRIGGER <имя триггера> ON <имя таблицы> FOR [INSERT],[UPDATE],[DELETE] AS <SQL-операторы>

mindmeister

Notes

1) Фактически являются правилами вывода.

А, В и С - атрибуты