

Формат проведения занятий

1. Семинары с докладами. Доклады должны быть посвящены обзору типа СУБД, соответствующего определенному варианту курсовой работы. Всего – 5 докладов за семестр (суммарно). Доклад выполняется командой из 2х человек (команда может не соответствовать таковой при написании курсовой, вклад каждого – 50%). Доклад должен быть подготовлен самостоятельно, не допускается цитирование статей с Википедии, обзоров с Хабра, доклад не может быть зачитан с листа, длительность доклада – 10 минут, тезисы доклада и иллюстрации должны быть отправлены преподавателю не позднее 3 дней до доклада. Доклад -> +1 балл к итоговой оценке себе или одногруппнику.
2. Защита обязательных и зачетных заданий курсовой работы.

Курсовая работа

Использование объектных и NoSQL (not only SQL) баз данных.

Цель курсовой работы:

Получение знаний, умений и навыков, необходимых для создания баз данных, основанных на принципах, отличных от классической реляционной (и объектно-реляционной) модели, в произвольной предметной области. Применение этих навыков и знаний для реализации БД по выбранной исполнителями предметной области.

Команда:

1-2 человека

Варианты:

Вариант	Тип	Возможные СУБД (могут быть выбраны другие)
1	Document-oriented	MongoDB + WiredTiger
2	Column-oriented	Apache Cassandra
3	Key/Value	Redis, Berkeley DB

* - достаточно для получения оценки «удовлетворительно»

*, **, *** - достаточно для получения оценки «хорошо»

*, **, ***, ****, ***** - обязательно для получения оценки «отлично»

4	Graph	Neo4j, AllegroGraph
5	Объектная	Cache

Алгоритм выбора варианта:

1. Складываются цифры номера группы, прибавляется номер по списку в группе первого студента, прибавляется номер по списку в группе второго студента.
2. От полученной суммы берется остаток от деления на 5.
3. Если получилось значение «0», то «5».

Обязательные задания:

1. Представить предметную область, для которой производится разработка БД.
2. Сформировать модель данных (или ее фрагмент, в случае если формирование полной модели недоступно, или описание, предусмотренной стандартами выбранной СУБД, если модель в принципе не может быть построена).
3. Обеспечить заполнение БД наборами данных аналогичными тем, что были внесены в курсовой работе, выполненной в предыдущем семестре.
4. Реализовать, средствами СУБД или иначе (в случае если СУБД не предоставляет такой возможности), программные модули (процедуры, функции, классы, методы, ...) для изменения сведений в разработанной БД (выполнение операций вставки, изменения и удаления).
5. «Защитить» разработанные программные решения.

Зачетные задания:

1. * **ДАНО:** база данных должна содержать набор данных, эквивалентный следующему (в случае реляционной СУБД): стержневая сущность, содержащая более 1000000 (одного миллиона) строк (со столбцами различных типов: строковых, числовых, даты), для этой стержневой сущности существуют 3 или более характеристические сущности, содержащие, в свою очередь, минимум 1000000 (один миллион) строк каждая, со значениями, актуальными на определенные даты (например, номер вашей учебной группы в текущий момент времени). Объем данных может быть уменьшен в случае, если

* - достаточно для получения оценки «удовлетворительно»

*, **, *** - достаточно для получения оценки «хорошо»

*, **, ***, ****, ***** - обязательно для получения оценки «отлично»

используемое вами аппаратное обеспечение не может в приемлемые сроки (обосновывается при защите или ранее) обеспечить обработку объемов, указанных в задании.

ЗАДАНИЕ:

- a. Обеспечить получение набора данных, эквивалентного, в случае реляционной модели и RDBMS Oracle, 1 (одному) SQL-запросу, каждая строка результата которого должна содержать строку из стержневой сущности и, объединенные агрегационными функциями (например, LISTAGG), данные из характеристических сущностей.
 - b. Обеспечить поиск по строковым атрибутам (на маску *"^.*поисковыйзапрос.*\$"*, с возможностью использования неправильной раскладки клавиатуры, например, *"fps lfyys["* = "базы данных").
 - c. Обосновать: увеличение, уменьшение или примерное равенство времени получения наборов данных из пп. 1.a и 1.b, при одинаковых входных параметрах, относительно времени получения тех же наборов при использовании реляционной СУБД (например, RDBMS Oracle), при условии равенства мощностей обрабатываемых множеств.
2. **** ДАНО:** наборы данных и структуры из задания №1.
ЗАДАНИЕ: реализовать алгоритмы Левенштейна, Джаро-Винклера и еще один любой другой алгоритм нечеткого поиска. Применить результаты задания №2 для решения заданий №№ 1.b и 1.c.
3. ***** ДАНО:** наборы данных и структуры из задания №1, алгоритмы из задания №2 (отличные от Левенштейна и Джаро-Винклера).
ЗАДАНИЕ: создать кластер, содержащий 3 узла (для этого могут быть использованы 2 виртуальных машины и host-машина). Использовать полученный кластер для применения возможностей параллельной обработки в решении задания №2.
4. ****** ДАНО:** наборы данных и структуры из задания №1, алгоритм, реализованный вами в задании №2. СУБД выбирается по принципу "ваш_вариант + 1", при этом, если ваш вариант – 5, то в этом задании используется СУБД из варианта 1. Предметная область остается вашей.

* - достаточно для получения оценки «удовлетворительно»

*, **, *** - достаточно для получения оценки «хорошо»

*, **, ***, ****, ***** - обязательно для получения оценки «отлично»

ЗАДАНИЕ: см. задание №3, со следующим допущением мощности всех множеств могут быть уменьшены в 10-100 раз.

Требования к БД:

1. Структура данных в БД должна соответствовать выбранной предметной области
2. Тип используемой СУБД должен соответствовать варианту (в зачетных заданиях 1-3)
3. Должна быть обеспечена корректность и целостность (если СУБД позволяет) сведений, содержащихся в БД
4. В случае, если используемая СУБД позволяет, то обеспечить поддержку транзакций

Артефакты:

1. Титульный лист
2. Описание предметной области
3. Модель базы данных (если возможно, иначе – описание в соответствии со стандартами СУБД, соответствующей вашему варианту в обязательном задании и зачетных заданиях 1-3)
4. Примеры CRUD-кода (для обязательного задания)
5. Описание программных модулей в формате комментариев к коду (для зачетных заданий)

Порядок выполнения курсовой работы:

1. Выполнение и защита обязательных заданий
2. Зачетное задание 1
3. Зачетное задание 2
4. Зачетное задание 3
5. Зачетное задание 4

* - достаточно для получения оценки «удовлетворительно»
*, **, *** - достаточно для получения оценки «хорошо»
*, **, ***, ****, ***** - обязательно для получения оценки «отлично»

Возможные источники (составлен 02.2016):

1. Документация вашей СУБД (!!!!!)
2. Лекции
3. Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement (Eric Redmond, Jim R. Wilson)
4. NoSQL Databases (Christof Strauch)
5. Programmer to Programmer - Tiwari S. - Professional NoSQL
6. <https://open.sap.com/> - курсы по HANA
7. MOOC'и на <http://coursera.org>, или edX, или <https://stepic.org> или др.
8. Видеокурсы NoSQL for SQL Professionals (<http://www.lynda.com/NoSQL-tutorials/NoSQL-SQL-Professionals/368756-2.html>)
9. ...

* - достаточно для получения оценки «удовлетворительно»
*, **, *** - достаточно для получения оценки «хорошо»
*, **, ***, *****, ***** - обязательно для получения оценки «отлично»