

Цель практического курса получить навыки проектирования и разработки хранилищ данных, витрин данных, а также навыки написания кода, обеспечивающего наполнение и перенос данных.

В рамках рекомендуется использовать средства SQL, но по согласованию допустимы и другие подходы. В своей работе студенты могут использовать и различные средства реализации ETL, это требует дополнительной договоренности с преподавателем.

Практика разделена на 6 работ. Каждая работа выполняется отдельно, с оформлением промежуточного отчета. Если задания конкретной работы кажутся студенту не корректными, или требующими внесения изменений, то он вправе изменить условия, предварительно согласовав изменения с преподавателем. Все изменения фиксируются у преподавателя в отдельном документе. Каждая работа содержит три составляющих: фактические материалы работы, демонстрация работы и защита отчета. Отчет включает в себя текст задания, теоретическое описание будущего решения, артефакты проектирования (диаграммы данных, диаграммы последовательностей и т.д.) с подробным объяснением, скрипты и код. Промежуточные отчеты выполняются в свободной форме. Итоговый отчет, включающий материалы по всем работам, выполняется как отчет о курсовой работе.

Каждая работа должна быть выполнена в определенный срок, и за каждую неделю просрочки снимается 20% баллов.

| Этап | Неделя | Баллы |
|---------|--------|-----------|
| 1 | 3-4 | 10 |
| 2 | 5 | 10 |
| 3 | 7 | 10 |
| 4 | 12 | 25 |
| 5 | 14 | 5 |
| 6 | 16 | 20 |
| экзамен | | 20 |
| | | 100 |

4 этап обязательен для выполнения (!)

По согласованию с преподавателем можно взять дополнительное задание – реализация программного модуля поддержки миграции данных. Дополнительное задание может быть оценено в 20 баллов.

В случае если студент принципиально не хочет использовать реляционные СУБД для реализации работ допустимо использовать NoSQL СУБД, рекомендовано MongoDB, с согласованием у преподавателя.

Лабораторная работа №1. Разработка базы данных «Филиал» как источника данных.

В данной лабораторной работе требуется разработать прототип базы данных филиала торгового предприятия. База данных будет использована в дальнейшем как источник данных для наполнения хранилища данных.

Разрабатываемый прототип должен обеспечить хранение информации о следующих основных сущностях:

1. Покупатель;
2. Товар;
3. Сделка.

Информацию о покупателях можно ограничить уникальным идентификатором, а так же названием покупателя для обеспечения удобства работы с информацией.

Сущность Товар должна содержать следующие минимальные поля: уникальный номер товара, название товара, цена товара по каталогу. Следует отметить что каждый товар принадлежит к одной или нескольким категориям. Таким образом следует создать дополнительную сущность Категория, и обеспечить связь многие ко многим с сущностью Товар.

Сущность Сделка должна отражать информацию о факте покупки товара покупателем. Требуется отметить, что покупатель за одну покупку может купить несколько наименований товара в разном количестве, а также информацию о цене, за которую был куплен товар (она может отличаться от заявленной цены, например с учетом скидки), а также общую сумму сделки.

Целесообразно сделать связь между сущностями Сделка и Товар через промежуточную сущность содержащую информацию о количестве купленного товара и цена. Сущность Сделка будет содержать итоговую сумму покупки, а также дополнительную информацию, например дату совершения сделки.

На основании имеющейся информации необходимо создать реляционную модель данных, и получить утвердить модель у преподавателя.

На основании имеющейся реляционной модели создать даталогическую модель, в соответствии с требованиями типов данных, используемых на конкретной версии SQL Server, и так же утвердить у преподавателя.

На основании данных моделей создать скрипт создания базы данных, с учетом следующих требований:

1. Каждая таблица должна иметь суррогатный целочисленный первичный ключ;
2. Каждая таблица, помимо атрибутов необходимых для хранения данных в соответствии с заданиями, должны содержать атрибут rowguid – уникальный идентификатор в рамках всей системы, и ModifiedDate - дата/время добавления или последней модификации строки;
3. Все связи должны быть реализованы как дополнительные изменения таблицы после их создания;
4. База данных должна содержать минимально необходимое количество ограничений и триггеров необходимых для стабильной работы;
5. Рекомендовано использовать английский язык для названий сущностей, атрибутов и других объектов БД.

Проверить работоспособность созданного скрипта и убедится в стабильности его работы.

Подготовить два комплекта операций вставки в базу данных, обеспечивающих наполнение базы необходимыми для тестирования данными. Объем данных должен быть минимум двадцать пять строк для базовых сущностей, и минимум пятьдесят строк для сущностей, реализующих связь.

Создать две базы данных по разработанному скрипту, и наполнить соответствующим набором данных. Созданные базы данных обозначить как «Филиал Запад» и «Филиал Восток». При создании скрипта наполнения данными рекомендуется использовать ИИ.

Выбор реляционной СУБД остается за студентом. СУБД, и полученный набор скриптов необходимо согласовать с преподавателем.

Лабораторная работа №2. Разработка модели и скрипта создания хранилища данных.

Хранилище данных должно реализовано с использованием реляционной СУБД, если иное не было оговорено заранее. Архитектура хранилища должна соответствовать одной из устоявшихся практик: Кимбалл, Инмон, Якорная модель, Data vault, или иная практика по согласованию с преподавателем. Студент обязан в отчете обосновать выбранную архитектуру. Наполнение будет производится в автоматизированном режиме из данных, поступающих из источников, разработанных в лабораторной работе №1. Для хранилища данных предусмотрена только операция добавления данных, операции модификации и удаления не допустимы.

Пример: Сформировать модель «звезда», в которой основной таблицей фактов будет продажа конкретного товара, измерениями будут товары, покупатели и т.д.

На основании имеющейся информации можно создать реляционную модель, утвердить ее у преподавателя. На основании утвержденной модели создать даталогическую модель, утвердить ее у преподавателя.

На основании созданных моделей создать скрипт создания хранилища данных, проверить его работоспособность. Создать тестовый (!) скрипт наполнения хранилища, проверить работоспособность.

Лабораторная работа №3. Создание витрины данных.

Витрина данных представляет собой специализированное хранилище, где данные консолидируются по определенному признаку. В данной работе требуется консолидировать результаты продаж по неделям. Это обеспечит уменьшение нагрузки на основное Хранилище данных.

Основными сущностями Витрины будут:

1. Недели;
2. Продажи.

Сущность недели хранит информацию о временных промежутках, за которые была произведена консолидация данных.

Сущность Продажи обеспечивает хранение информации о результатах продаж в Филиале в соответствующую неделю.

Лабораторная работа №4. Создания процедуры наполнения Хранилища данных.

Цель работы: создать хранимую процедуру, или скрипт, обеспечивающую автоматизированный перенос данных из Филиалов в Хранилище. Процедура должна обеспечить добавление вновь поступившей информации из Филиала Запад, и Филиала Восток в центральное хранилище данных. Процедура должна исключить дублирование информации. Разработчик должен продумать механизм обеспечивающий минимализацию нагрузки на продуктивные сервера филиалов.

В процедуре можно использовать операции множественной вставки.

Лабораторная работа №5. Создание Процедуры наполнения Витрины данных.

Цель работы: разработать скрипт, или хранимую процедуру, которая обеспечит наполнение Витрины данных на основании детальных данных, имеющихся в Хранилище. Витрина данных наполняется данными за определенный период, например за пять недель выбранных пользователем.

Лабораторная работа №6. Создание скрипта восстановления утраченных филиалом данных на основании детальных данных Хранилища.

Цель работы: Создание скрипта, обеспечивающего обратный перенос очищенных данных из Хранилища в выбранный Филиал.

Скрипт можно оформить как набор SQL команд, или как хранимую процедуру. Скрипт должен переносить данные за выбранный промежуток времени из Хранилища в Филиал. Скрипт должен работать исходя из того, что данные в Филиале полностью отсутствуют.