

**Комплект оценочных материалов по дисциплине
«Программирование и администрирование систем управления базами
данных»**

Задания закрытого типа

Задания закрытого типа на выбор правильного ответа

1. Выберите один правильный ответ.

Команда администрирования базы данных GRANT позволяет:

А) отключить определенные разрешения для объекта безопасности, и предотвратить наследование объектом разрешений через его членство в роли или группе

Б) установить определенные разрешения для объекта безопасности

В) управлять окружением SQL Server

Правильный ответ: Б

Компетенции (индикаторы): ПК-1, ПК-2

2. Выберите один правильный ответ.

Каким образом происходит поиск по дереву индекса?

А) начиная с корня дерева, выполняется поиск наименьшего значения ключа, меньшего или равного требуемому значению;

Б) начиная с корня дерева, выполняется поиск наибольшего значения ключа, меньшего или равного требуемому значению;

В) начиная с корня дерева, выполняется поиск наименьшего значения ключа, большего или равного требуемому значению;

Г) начиная с корня дерева, выполняется поиск наибольшего значения ключа, большего или равного требуемому значению;

Правильный ответ: В

Компетенции (индикаторы): ПК-1, ПК-2

3. Выберите один правильный ответ.

Что обеспечивает выполнение правил для вставок и обновления таблиц, содержащих внешний ключ и соответствующее ограничение первичного ключа?

А) Ссылочная целостность

Б) Первичный ключ

В) Декларативное ограничение целостности

Г) Внешний ключ

Правильный ответ: А

Компетенции (индикаторы): ПК-1, ПК-2

4. Выберите один правильный ответ.

Что такое хранимая процедура?

- А) Сохраненный набор команд SQL, созданный в редакторе запросов.
- Б) Системный объект базы данных, который выполняет заранее определенную задачу и может вызываться по имени без передачи параметров.
- В) Группа команд SQL, которая компилируется и выполняется как единое целое.

Правильный ответ: В

Компетенции (индикаторы): ПК-1, ПК-2

5. Выберите правильный ответ.

Как называется древовидная структура в SQL Server, в которую встроены индексы?

- А) дерево связей
- Б) сбалансированное дерево
- В) дерево индексации.

Правильный ответ: Б

Компетенции (индикаторы): ПК-1, ПК-2

6. Выберите один правильный ответ.

Чтобы повысить производительность запросов к столбцам с типом данных XML, столбцы XML можно:

- А) детерминировать
- Б) структурировать
- В) индексировать

Правильный ответ: В

Компетенции (индикаторы): ПК-1, ПК-2

7. Выберите правильный ответ.

Физическую целостность и живучесть транзакции обеспечивает:

- А) ядро базы данных SQL Server
- Б) идентификатор базы данных
- В) протокол WSDL

Правильный ответ: А

Компетенции (индикаторы): ПК-1, ПК-2

8. Выберите один правильный ответ.

1. Если в инструкциях ALTER TABLE и CREATE TABLE предложение CONSTRAINT не указывается, то:

- А) Database Engine преобразует ограничение в комментарий;
- Б) имя ограничению присваивается неявно компонентом Database Engine;

В) Database Engine выдает ошибку "недостаточно параметров";

Г) Database Engine игнорирует строку определения;

Правильный ответ: Б.

Компетенции (индикаторы): ПК-1, ПК-2

Задания закрытого типа на установление соответствия

1. Установите правильное соответствие. Каждому элементу левого столбца соответствует только один элемент правого столбца.

Целостность данных

Определение

- | | | |
|------------------------------------|---|--|
| 1) Целостность на уровне доменов | А | внешние ключи не должны быть несогласованными |
| 2) Целостность на уровне отношений | Б | отношение должно иметь по крайней мере один потенциальный ключ |
| 3) Целостность на уровне БД | В | типы используемых данных должны быть атомарными |

Правильный ответ:

1	2	3
В	Б	А

Компетенции (индикаторы): ПК-1, ПК-2

2. Установите правильное соответствие. Каждому элементу левого столбца соответствует только один элемент правого столбца.

Методы резервного копирования

Характеристика

- | | | |
|---|----|--|
| 1) Полное резервное копирование | А) | Копирует только изменения, сделанные с момента последнего полного бэкапа. |
| 2) Дифференциальное резервное копирование | Б) | Выполняется без остановки работы базы данных. |
| 3) Инкрементное резервное копирование | В) | Копирует изменения, сделанные с момента последнего дифференциального или полного бэкапа. |
| 4) Горячее резервное копирование | Г) | Копирует все данные базы данных. |

Правильный ответ:

1	2	3	4
Г	А	В	Б

Компетенции (индикаторы): ПК-1, ПК-2

3. Установите правильное соответствие. Каждому элементу левого столбца соответствует только один элемент правого столбца.

Элемент архитектуры СУБД

Назначение

- | | | |
|-------------------------|----|--|
| 1) Оптимизатор запросов | А) | Управляет правами доступа к данным и политиками безопасности. |
| 2) Диспетчер транзакций | Б) | Содержит информацию о всех выполненных изменениях в базе данных. |

- 3) Журнал транзакций В) Выбирает наиболее эффективный способ выполнения SQL-запросов.
- 4) Менеджер безопасности Г) Обеспечивает контроль параллельного выполнения транзакций и их откат.

Правильный ответ:

1	2	3	4
В	Г	Б	А

Компетенции (индикаторы): ПК-1, ПК-2

4. Установите правильное соответствие. Каждому элементу левого столбца соответствует только один элемент правого столбца.

Основные проблемы при параллельной работе с данными

Описание

- 1) Грязное чтение (Dirty Read) А) Одновременные транзакции изменяют одну и ту же строку, что приводит к потере обновлений.
- 2) Фантомное чтение (Phantom Read) Б) Одна транзакция читает изменения, сделанные другой транзакцией, которая еще не была зафиксирована.
- 3) Потерянное обновление (Lost Update) В) Повторный запрос к базе данных в одной транзакции возвращает разные результаты.
- 4) Несогласованное чтение (Non-Repeatable Read) Г) Одна транзакция видит строки, добавленные другой транзакцией после первого выполнения запроса.

Правильный ответ:

1	2	3	4
Б	Г	А	В

Компетенции (индикаторы): ПК-1, ПК-2

5. Установите правильное соответствие. Каждому элементу левого столбца соответствует только один элемент правого столбца.

Типы файлов в СУБД SQL Server

Назначение

- 1) MDF (Primary Data File) А) Хранит основные данные базы данных.
- 2) NDF (Secondary Data File) Б) Используется для хранения журнала транзакций.
- 3) LDF (Log File) В) Дополнительные файлы данных, распределяющие нагрузку.
- 4) TempDB Г) Временная база данных для промежуточных расчетов.

Правильный ответ:

1	2	3	4
А	В	Б	Г

Компетенции (индикаторы): ПК-1, ПК-2

6. Установите правильное соответствие. Каждому элементу левого столбца соответствует только один элемент правого столбца.

Уровни конфигурирования

Параметры

СУБД

- | | |
|-----------------------|--|
| 1) Аппаратный уровень | А) Управление схемами данных, таблицами и индексами. |
| 2) Системный уровень | Б) Настройка параметров работы сервера базы данных. |
| 3) Базовый уровень | В) Оптимизация CPU, RAM, дисковой подсистемы. |
| 4) Прикладной уровень | Г) Оптимизация SQL-запросов на уровне приложений. |

Правильный ответ:

1	2	3	4
В	Б	А	Г

Компетенции (индикаторы): ПК-1, ПК-2

Задания закрытого типа на установление правильной последовательности

1. Установите правильную последовательность выполнения предложений в операторе SELECT в MS SQL Server:

- А) SELECT (возврат результирующего датасета)
- Б) FROM (выбор таблицы)
- В) WHERE (фильтрация строк)
- Г) ORDER BY (сортировка)
- Д) GROUP BY (агрегирование данных)

Правильный ответ: Б, В, Д, А, Г

Компетенции (индикаторы): ПК-1, ПК-2

2. Установите правильную последовательность основных шагов, используемых SQL Server для обработки одиночной инструкции SELECT

А) Строится дерево запроса с описанием логических шагов, необходимых для преобразования исходных данных в формат, требуемый результирующему набору.

Б) Средство анализа просматривает инструкцию SELECT и разбивает ее на логические единицы, такие как ключевые слова, выражения, операторы и идентификаторы.

В) Реляционный механизм начинает реализовывать план выполнения. В ходе обработки шагов, требующих данных из базовых таблиц, реляционный

механизм запрашивает у подсистемы хранилища передачу данных из набора строк, указанных реляционным механизмом.

Г) Реляционный механизм преобразует данные, возвращенные подсистемой хранилища, в заданный для результирующего набора формат и возвращает результирующий набор клиенту.

Д) Оптимизатор запросов анализирует различные способы, с помощью которых можно обратиться к исходным таблицам. Затем он выбирает ряд шагов, которые возвращают результаты быстрее всего и используют меньше ресурсов. Дерево запроса обновляется для записи этого точного ряда шагов.

Правильный ответ: Б, А, Д, В, Г

Компетенции (индикаторы): ПК-1, ПК-2

3. Установите правильную последовательность действий для исправления ошибок в MS SQL с помощью Recovery Toolbox for SQL Server

А) Запустить программу Recovery Toolbox/

Б) Начать восстановление нажатием Start recovery.

В) Выбрать наиболее приемлемый способ, которым будут экспортироваться данные: сохранением на диск в качестве SQL-скрипта или выполнением SQL-скрипта в самой БД.

Г) Осуществить предварительный просмотр тех данных, которые в процессе выполнения программы могут быть подвергнуты извлечению из базы MS SQL сервер, которая подверглась повреждению.

Д) Из списка файлов выбрать файл *.mdf, который был поврежден.

Е) Произвести выборку информации, требующей восстановления и сохранения.

Правильный ответ: А, Д, Г, Е, Б

Компетенции (индикаторы): ПК-1, ПК-2

4. Установите правильную последовательность действий при администрировании Microsoft SQL Server.

А) Установка и первичная настройка сервера.

Б) Экспорт и импорт данных.

В) Обслуживание и оптимизация работы базы данных.

Г) Резервное копирование и восстановление базы данных.

Д) Конфигурация сервера.

Е) Мониторинг системы MS SQL Server.

Ж) Управление безопасностью.

Правильный ответ: А, Д, В, Г, Б, Ж, Е

Компетенции (индикаторы): ПК-1, ПК-2

5. Установите правильную последовательность при блокировке транзакции в MS SQL Server.

А) Начало транзакции.

Б) Блокировка таблицы.

- В) Установка уровня изоляции.
 - Г) Исполнение действий в хранимой процедуре.
 - Д) Завершение транзакции.
- Правильный ответ: А, В, Б, Г, Д
Компетенции (индикаторы): ПК-1, ПК-2

6. Установите правильную последовательность действий при восстановлении после сбоя в SQL Server.

- А) Определить целевую точку восстановления.
 - Б) Определить тип выполняемого восстановления.
 - В) Создать резервную копию заключительного фрагмента журнала базы данных.
 - Г) Произвести накат базы данных.
 - Д) Восстановить одну или несколько резервных копий данных.
- Правильный ответ: В, А, Б, Д, Г
Компетенции (индикаторы): ПК-1, ПК-2

Задания открытого типа

Задания открытого типа на дополнение

1. Напишите пропущенное слово.
_____ – это временное ограничение, накладываемое системой на использование тех или иных ресурсов.
Правильный ответ: Блокировка.
Компетенции (индикаторы): ПК-1, ПК-2
2. Напишите пропущенное слово.
Шаблоном для создания новых баз данных в MS SQL Server является база данных _____.
Правильный ответ: Model
Компетенции (индикаторы): ПК-1, ПК-2
3. Напишите пропущенное словосочетание.
_____ – это процесс конструирования общей информационной модели на основе отдельных моделей данных пользователей. При этом информационная модель является независимой от особенностей реально используемой СУБД и других физических условий.
Правильный ответ: Логическое моделирование
Компетенции (индикаторы): ПК-1, ПК-2
4. Напишите пропущенное слово.
База данных в MS SQL Server, хранящая информацию о пользователях, имеющих доступ к серверу, называется _____

Правильный ответ: Master

Компетенции (индикаторы): ПК-1, ПК-2

5. Напишите пропущенное словосочетание.

_____ – это ключевое свойство транзакций, гарантирующее их неделимость и выполнение полностью или не выполняющееся вовсе.

Правильный ответ: Атомарность

Компетенции (индикаторы): ПК-1, ПК-2

6. Напишите пропущенное слово.

Оператор _____ используется для удаления данных из таблицы.

Правильный ответ: Delete

Компетенции (индикаторы): ПК-1, ПК-2

7. Напишите пропущенное словосочетание.

_____ – это разновидность индекса в базе данных, который определяет физический порядок хранения строк в таблице.

Правильный ответ: Кластерный индекс

Компетенции (индикаторы): ПК-1, ПК-2

8. Напишите пропущенное слово

_____ – это способ защиты данных, который предлагает SQL Server при работе с конфиденциальной информацией.

Правильный ответ: Шифрование

Компетенции (индикаторы): ПК-1, ПК-2

9. Напишите пропущенное слово.

_____ – это процесс устранения избыточности данных и предотвращения аномалий обновления в реляционной базе данных называется

Правильный ответ: Нормализация

Компетенции (индикаторы): ПК-1, ПК-2

10. Напишите пропущенное слово.

_____ – это метод повышения производительности базы данных за счет автоматического хранения часто используемых данных.

Правильный ответ: Кэширование.

Компетенции (индикаторы): ПК-1, ПК-2

11. Напишите пропущенное словосочетание.

_____ – это процесс копирования данных базы данных для их защиты от потерь и сбоев

Правильный ответ: Резервное копирование

Компетенции (индикаторы): ПК-1, ПК-2

12. Напишите пропущенное слово.

_____ – это структуры данных, которые увеличивают скорость операций выборки в базе данных

Правильный ответ: Индексы

Компетенции (индикаторы): ПК-1, ПК-2

13. Напишите пропущенное слово.

_____ – это виртуальные таблицы, основанные на результатах SQL-запросов. Преимущества их использования включают упрощение сложных запросов, безопасность данных, скрывание сложности взаимодействия с таблицами

Правильный ответ: Представления

Компетенции (индикаторы): ПК-1, ПК-2

14. Напишите пропущенное слово.

_____ – это последовательности операций, которые выполняются как единое целое. Если одна из операций не выполняется, все операции могут быть отменены, что обеспечивает целостность данных.

Правильный ответ: Транзакции

Компетенции (индикаторы): ПК-1, ПК-2

15. Напишите пропущенное словосочетание.

_____ в MS SQL Server – это промежуточное хранилище данных, которое существует только на время сессии или до завершения выполнения запроса.

Правильный ответ: Временная таблица

Компетенции (индикаторы): ПК-1, ПК-2

16. Напишите пропущенное словосочетание.

_____ – асинхронный (в общем случае) процесс переноса изменений объектов исходной базы данных в базы, расположенные на других узлах распределенной системы.

Правильный ответ: Тиражирование данных.

Компетенции (индикаторы): ПК-1, ПК-2

Задания открытого типа с кратким свободным ответом

1. Напишите пропущенное слово.

_____ – это особый тип хранимой процедуры, которая автоматически выполняется при изменении таблицы с помощью операторов UPDATE, INSERT или DELETE

Правильный ответ: Триггер / Trigger

Компетенции (индикаторы): ПК-1, ПК-2

2. Напишите пропущенное слово (словосочетание).

_____ – механизм выполнения транзакций, когда результат совместного выполнения транзакций эквивалентен результату некоторого последовательного выполнения этих же транзакций

Правильный ответ: сериализация / сериализация транзакций/ метод сериализации транзакций.

Компетенции (индикаторы): ПК-1, ПК-2

3. Напишите пропущенное словосочетание.

_____ – это набор операторов T-SQL, которые SQL SERVER компилирует в единый план выполнения,

Правильный ответ: Хранимая процедура / stored procedure

Компетенции (индикаторы): ПК-1, ПК-2

4. Напишите пропущенное слово (словосочетание).

Запросы, которые содержат набор критериев для нахождения интересующих пользователя данных из одной или более таблиц, – это

_____ Правильный ответ: select / запросы select / запросы на выборку

Компетенции (индикаторы): ПК-1, ПК-2

5. Напишите пропущенное слово (словосочетание).

Устранение транзитивных зависимостей характерно при приведении к _____.

Правильный ответ: 3НФ / третьей нормальной форме / третьей НФ

Компетенции (индикаторы): ПК-1, ПК-2

6. Напишите пропущенное слово (словосочетание).

_____ – это индекс, представляющий собой двоичное дерево, в котором на нулевом уровне (уровне листьев) содержатся страницы актуальных данных таблицы, а физическое расположение информации в данном индексе логически упорядочено, называется

Правильный ответ: кластеризованный / кластеризованный индекс / кластеризующий индекс

Компетенции (индикаторы): ПК-1, ПК-2

7. Напишите пропущенное слово (словосочетание).

Операция согласования состояния базы данных в физических файлах с текущим состоянием системного буфера называется _____.

Правильный ответ: контрольная точка / контрольной точкой

Компетенции (индикаторы): ПК-1, ПК-2

8. Напишите пропущенное слово (словосочетание).

Запрос, который выбирает данные из таблицы со ссылкой на внешний запрос называется _____

Правильный ответ: коррелированный подзапрос /коррелированный /коррелированным /коррелированным подзапросом.

Компетенции (индикаторы): ПК-1, ПК-2

Задания открытого типа с развернутым ответом

1. Выполнить разбор скрипта T-SQL. Рассмотреть код, определить его назначение, дать детальные пояснения к алгоритму выполнения, перечислить выполняемые действия и использованные возможности T-SQL.

```
CREATE TRIGGER trg_example
ON SomeTable
AFTER INSERT
AS
BEGIN
    BEGIN TRY
        INSERT INTO AnotherTable (Column1, Column2)
        SELECT Column1, Column2
        FROM inserted;
    END TRY
    BEGIN CATCH
        INSERT INTO ErrorLog (ErrorMessage, ErrorDate, ErrorSeverity,
ErrorState)
        VALUES (ERROR_MESSAGE(), GETDATE(),
ERROR_SEVERITY(), ERROR_STATE());

        INSERT INTO ErrorLog (ErrorMessage, ErrorDate, ErrorSeverity,
ErrorState, SourceTable)
        VALUES (ERROR_MESSAGE(), GETDATE(),
ERROR_SEVERITY(), ERROR_STATE(), 'SomeTable');

    END CATCH
END
```

Время выполнения: 35 минут.

Ожидаемый результат.

Этот код создает триггер trg_example, который срабатывает после вставки (AFTER INSERT) в таблицу SomeTable.

Триггер выполняет вставку данных в другую таблицу (AnotherTable) и логирование ошибок в таблицу ErrorLog, если произошла ошибка.

Построчный разбор кода с комментариями:

```
CREATE TRIGGER trg_example
ON SomeTable
```

AFTER INSERT

Создание триггера с именем `trg_example`, который срабатывает после вставки (AFTER INSERT) в таблицу `SomeTable`.

Триггер AFTER INSERT выполняется только после успешного завершения операции INSERT, в отличие от INSTEAD OF INSERT, который может заменить выполнение операции.

AS

BEGIN

Начало тела триггера. Внутри BEGIN ... END определяются инструкции, которые выполняются при срабатывании триггера.

BEGIN TRY

Начало блока TRY. Этот блок используется для обработки ошибок в конструкции TRY...CATCH, что позволяет избежать прерывания выполнения триггера при возникновении ошибки.

INSERT INTO AnotherTable (Column1, Column2)

SELECT Column1, Column2

FROM inserted;

Основное действие триггера: вставка данных в таблицу `AnotherTable`.

`inserted` – это специальная таблица-псевдоним в триггерах, содержащая вставленные данные в `SomeTable`.

Что делает этот код?

1) выбирает значения `Column1` и `Column2` из `inserted` (то есть новые данные, добавленные в `SomeTable`);

2) записывает их в `AnotherTable`, что фактически дублирует или логирует вставленные данные.

END TRY

Завершение блока TRY. Если код в TRY выполняется без ошибок, блок CATCH пропускается.

BEGIN CATCH

Обработка ошибок, если во время вставки произошла ошибка.

Если попытка вставки (INSERT INTO AnotherTable) завершится с ошибкой, управление передается в CATCH.

INSERT INTO ErrorLog (ErrorMessage, ErrorDate, ErrorSeverity, ErrorState)

VALUES (ERROR_MESSAGE(), GETDATE(), ERROR_SEVERITY(), ERROR_STATE());

Запись информации об ошибке в таблицу `ErrorLog`.

ERROR_MESSAGE() – текст ошибки.

GETDATE() – текущее время ошибки.

ERROR_SEVERITY() – уровень критичности ошибки.

ERROR_STATE() – состояние ошибки (уникальный код ошибки внутри сеанса).

INSERT INTO ErrorLog (ErrorMessage, ErrorDate, ErrorSeverity, ErrorState, SourceTable)

VALUES (ERROR_MESSAGE(), GETDATE(),
ERROR_SEVERITY(), ERROR_STATE(), 'SomeTable');

Дополнительная запись в ErrorLog, но с указанием источника (SourceTable). Здесь добавляется информация о том, что ошибка произошла в SomeTable.

END CATCH

Завершение блока CATCH. Все действия, выполняемые при ошибке, должны быть внутри CATCH.

END

Завершение триггера.

Детальный разбор скрипта позволил определить действия, которые триггер выполняет:

- автоматически срабатывает при вставке данных в SomeTable;
- копирует (логирует) новые данные в AnotherTable;
- записывает сведения об ошибке в ErrorLog, если вставка в AnotherTable завершается с ошибкой;
- использует обработку ошибок TRY...CATCH, чтобы избежать аварийного завершения триггера.

Возможности T-SQL, которые использованы при написании кода:

- триггеры (CREATE TRIGGER) – автоматическое выполнение кода при изменении данных;
- временные таблицы inserted – доступ к данным, вставленным в SomeTable;
- обработка ошибок (TRY...CATCH) – предотвращает прерывание выполнения при ошибке;
- функции обработки ошибок (ERROR_MESSAGE(), ERROR_SEVERITY(), ERROR_STATE()) – логирование информации об ошибках.
- функция для работы с датами (GETDATE()) – для фиксации времени ошибки.

Этот триггер полезен для автоматического логирования операций и ошибок, помогая отслеживать корректность работы базы данных.

Критерии оценивания:

- ответ должен быть осмысленным, а не пересказом кода.
- особое внимание – объяснению INSERTED и TRY...CATCH.
- в ответе допускаются разные формулировки, но смысл должен быть верным.

В данном задании оценивается не только знание T-SQL, но и способность анализировать код и объяснять его ценность. Поэтому при проверке ответа следует обращать внимание не только на правильность разьяснения кода, но и на понимание принципов работы триггеров, обработки ошибок и временных таблиц в T-SQL. Если студент просто описывает код без анализа его смысла и пользы, ответ нельзя считать полным.

Компетенции (индикаторы): ПК-1, ПК-2

2. Выполнить разбор скрипта T-SQL. Рассмотреть код, определить его назначение, дать детальные пояснения к алгоритму выполнения, перечислить выполняемые действия и использованные возможности T-SQL.

```
CREATE TABLE #TempTable (  
    ID INT PRIMARY KEY,  
    Name NVARCHAR(100)  
);  
INSERT INTO #TempTable (ID, Name)  
VALUES (1, 'Alice'), (2, 'Bob');  
SELECT * FROM #TempTable;  
DECLARE @TableVariable TABLE (  
    ID INT PRIMARY KEY,  
    Name NVARCHAR(100)  
);  
INSERT INTO @TableVariable (ID, Name)  
VALUES (3, 'Charlie'), (4, 'David');  
SELECT * FROM @TableVariable;  
Время выполнения: 20 минут.  
Ожидаемый результат:
```

Этот скрипт демонстрирует различие между временными таблицами (#TempTable) и табличными переменными (@TableVariable) в SQL Server, создавая обе структуры, заполняя их данными и извлекая информацию.

Алгоритм выполнения кода

1) Создание временной таблицы (`#TempTable`)

```
CREATE TABLE #TempTable (  
    ID INT PRIMARY KEY,  
    Name NVARCHAR(100)  
);
```

Создается временная таблица `#TempTable`, которая существует только в рамках текущей сессии. Таблица содержит два столбца:

- `ID` (целочисленный, первичный ключ).
- `Name` (строковый тип `NVARCHAR(100)`).

2) Вставка данных во временную таблицу

```
INSERT INTO #TempTable (ID, Name)  
VALUES (1, 'Alice'), (2, 'Bob');
```

Заполняет `#TempTable` двумя записями (`Alice`, `Bob`).

3) Выборка данных из временной таблицы

```
SELECT * FROM #TempTable;
```

Извлекает все данные из `#TempTable`.

4) Создание переменной таблицы (`@TableVariable`)

```
DECLARE @TableVariable TABLE (  
    ID INT PRIMARY KEY,  
    Name NVARCHAR(100)
```

);

Определяется переменная таблица '@TableVariable', которая существует только в рамках текущего блока кода.

5) Вставка данных в переменную таблицу

```
INSERT INTO @TableVariable (ID, Name)
VALUES (3, 'Charlie'), (4, 'David');
```

Заполняет '@TableVariable' двумя записями ('Charlie', 'David').

6) Выборка данных из переменной таблицы

```
SELECT * FROM @TableVariable;
```

Извлекает все данные из '@TableVariable'.

Используемые возможности T-SQL:

- временные таблицы ('#TempTable') – хранят данные в 'tempdb' и могут использоваться в разных процедурах или сессиях.

- табличные переменные ('@TableVariable') – работают только внутри текущего блока кода и не могут передаваться между процедурами.

- PRIMARY KEY – устанавливает ограничение целостности данных.

- INSERT INTO ... VALUES – добавляет строки в таблицы.

- SELECT * FROM – извлекает данные.

Этот код демонстрирует различия между временными таблицами и переменными таблицами:

1. #TempTable (временная таблица)

- хранится во временной базе данных 'tempdb'.

- доступна в пределах сессии (или соединения).

- можно использовать в разных процедурах и транзакциях.

2. '@TableVariable' (табличная переменная)

- хранится в памяти, а не в 'tempdb'.

- существует только внутри текущего блока кода (например, внутри хранимой процедуры).

- не поддерживает статистику и индексы (кроме 'PRIMARY KEY'), что может снижать производительность при больших объемах данных.

Когда использовать:

#TempTable – если требуется передача данных между процедурами или многократное использование.

@TableVariable – если объем данных небольшой, и таблица используется только в рамках одного блока кода.

Критерии оценивания

- ответ должен быть осмысленным, а не пересказом кода.

- особое внимание – объяснению INSERTED и TRY...CATCH.

- проведен детальный разбор алгоритма выполнения;

- перечислены используемые возможности T-SQL

- особое внимание к различиям в области видимости и хранении временных таблиц и табличных переменных;

- корректность и структурированность ответа.

Оценивается не только знание T-SQL, но и способность анализировать код, понимать различия между типами таблиц и объяснять их практическое применение. В ответе допускаются разные формулировки, но логика должна быть верной – если объяснение правильное, но изложено своими словами, это допустимо.

Компетенции (индикаторы): ПК-1, ПК-2

3. Выполнить разбор скрипта T-SQL. Разобрать код, определить его назначение, дать детальные пояснения к алгоритму выполнения, перечислить выполняемые действия и использованные возможности T-SQL.

```
CREATE TABLE TargetTable (  
    ID INT PRIMARY KEY,  
    Name NVARCHAR(100),  
    CreatedAt DATETIME DEFAULT GETDATE()  
);  
CREATE TABLE #TempTable (  
    ID INT,  
    Name NVARCHAR(100)  
);  
INSERT INTO #TempTable (ID, Name)  
VALUES (1, 'Olga'), (2, 'Helena'), (3, 'Nic');  
INSERT INTO TargetTable (ID, Name)  
SELECT ID, Name FROM #TempTable;  
SELECT * FROM TargetTable;  
Время выполнения 25 минут.
```

Ожидаемый результат:

Разбор T-SQL-кода и пояснение используемых возможностей.

Данный T-SQL-код демонстрирует использование временных таблиц для промежуточного хранения данных и их последующий импорт в основную таблицу.

T-SQL-скрипт создает основную и временную таблицы, заполняет временную таблицу данными, импортирует их в основную таблицу, а затем выводит данные из нее.

Разбор кода построчно.

1) Создание основной таблицы (TargetTable)

```
CREATE TABLE TargetTable (  
    ID INT PRIMARY KEY,  
    Name NVARCHAR(100),  
    CreatedAt DATETIME DEFAULT GETDATE()  
);
```

Этот фрагмент кода:

- создает таблицу TargetTable, предназначенную для хранения данных.
- ID INT PRIMARY KEY – уникальный идентификатор записи, который

не может повторяться.

- Name NVARCHAR(100) – строковый столбец для хранения имен.
- CreatedAt DATETIME DEFAULT GETDATE() – автоматически проставляет дату и время вставки записи.

DEFAULT GETDATE() гарантирует, что каждая запись получает временную метку при вставке. Такой подход широко применяется в ETL-процессах, обработке данных и интеграции между различными хранилищами.

2) Создание временной таблицы (#TempTable)

```
CREATE TABLE #TempTable (
    ID INT,
    Name NVARCHAR(100)
);
```

Этот фрагмент кода:

- создает временную таблицу #TempTable, которая существует только в рамках текущей сессии;
- временные таблицы хранятся во временной БД tempdb и удаляются автоматически при завершении сессии;
- поля ID и Name аналогичны структуре основной таблицы.

Временные таблицы удобны для обработки временных данных и выполнения промежуточных вычислений.

3. Заполнение временной таблицы данными

```
INSERT INTO #TempTable (ID, Name)
VALUES (1, 'Olga'), (2, 'Helena'), (3, 'Nic');
```

Этот фрагмент кода:

- добавляет три записи в временную таблицу.
- оператор INSERT INTO ... VALUES используется для добавления данных по указанным столбцам.

4. Копирование данных из временной таблицы в основную

```
INSERT INTO TargetTable (ID, Name)
SELECT ID, Name FROM #TempTable;
```

Этот фрагмент кода:

- переносит данные из #TempTable в TargetTable.
- оператор INSERT INTO ... SELECT копирует данные пакетно, а не построчно.
- столбец CreatedAt в TargetTable заполняется автоматически, так как имеет значение по умолчанию (DEFAULT GETDATE()).

Использование INSERT INTO ... SELECT упрощает миграцию данных между таблицами

5. Вывод данных из TargetTable

```
SELECT * FROM TargetTable;
```

Этот фрагмент кода:

- выбирает все данные из TargetTable.
- показывает результат импорта из временной таблицы в основную.

Использованные возможности T-SQL:

1) Создание таблиц (CREATE TABLE)

Создает как постоянные (TargetTable), так и временные (#TempTable) таблицы. Временные таблицы удаляются автоматически после завершения сессии.

2) Ограничения (PRIMARY KEY)

Уникальный идентификатор ID, который исключает дублирование записей.

3) Заполнение значений по умолчанию (DEFAULT GETDATE()).

Позволяет автоматически проставлять дату и время создания записи.

4) Запрос на вставку данных (INSERT INTO ... VALUES)

Добавляет строки в таблицу по конкретным значениям.

5) Запрос на копирование данных (INSERT INTO ... SELECT)

Позволяет пакетно копировать данные из одной таблицы в другую.

Упрощает массовый импорт данных.

6) Выборка данных (SELECT * FROM)

Позволяет просматривать содержимое таблицы после выполнения всех операций.

Критерии оценивания:

- ответ должен быть осмысленным, а не пересказом кода.

- проведен детальный разбор алгоритма выполнения;

- перечислены используемые возможности T-SQL

- особое внимание к объяснению разницы между временной и постоянной таблицами.

различиям в области видимости и хранении временных таблиц и табличных переменных;

- корректность и структурированность ответа.

Оценивается не только знание T-SQL, но и способность анализировать код, понимать различия между типами таблиц и объяснять их практическое применение. В ответе допускаются разные формулировки, но логика должна быть верной – если объяснение правильное, но изложено своими словами, это допустимо.

Компетенции (индикаторы): ПК-1, ПК-2

4. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите ответ, используя четкие компактные формулировки.

Вопрос: объясните основные различия между моделями баз данных: реляционной, документоориентированной и графовой, поясните условия выбора определенной модели данных.

Время выполнения: 25 минут

Ожидаемый результат:

Базы данных различаются по способу хранения, организации и обработки данных. Три основные модели:

1) Реляционная модель:

- данные хранятся в виде таблиц (строки – записи, столбцы – атрибуты);

- обеспечивает целостность данных с помощью ключей и связей (PK, FK);

- использует язык SQL для запросов;

- примеры СУБД: MySQL, PostgreSQL, Microsoft SQL Server, Oracle.

2) Документо-ориентированная модель:

- данные представлены в виде JSON/BSON-документов, структурированных как вложенные объекты;

- гибкая схема хранения, подходящая для неструктурированных и полуструктурированных данных;

- использует NoSQL-подход, поддерживает горизонтальное масштабирование;

- Примеры СУБД: MongoDB, CouchDB, Firebase Firestore.

3) Графовая модель:

- основана на вершинах (объектах) и рёбрах (связях между объектами);

- оптимизирована для сложных взаимосвязей, где важны связи, а не просто данные;

- использует специализированные запросы (Cypher, Gremlin);

- примеры СУБД: Neo4j, ArangoDB, Amazon Neptune.

Вывод:

Выбор модели базы данных зависит от характера данных и задач.

1) Реляционные БД подходят для структурированных данных, где важны строгие связи и целостность (банковские системы, учетные системы, CRM). Они обеспечивают надежную обработку транзакций (ACID) и работают через SQL.

2) Документ-ориентированные БД эффективны для гибких и динамически изменяющихся данных, когда записи могут иметь разную структуру (например, данные пользователей в мобильных приложениях, интернет-магазины, логирование событий). Они удобны для работы с JSON-документами и горизонтально масштабируются.

3) Графовые БД оптимальны для анализа взаимосвязей между объектами (например, социальные сети, системы рекомендаций, блокчейн, управление сетями). Они работают быстрее традиционных БД при сложных многосвязных запросах.

Каждая модель имеет свои преимущества и области применения, и выбор подходящей технологии зависит от специфики проекта, объема данных и требований к скорости обработки запросов.

Критерии оценивания:

- корректное определение трех моделей баз данных;

- приведены особенности каждой модели;

- даны примеры СУБД для каждой модели.

Компетенции (индикаторы): ПК-1, ПК-2

5. Прочитайте текст задания. Продумайте логику и полноту ответа. Запишите ответ, используя четкие компактные формулировки.

Вопрос: В чем преимущества нормализации и почему нормализация базы данных важна при проектировании информационных систем?

Время выполнения: 30 минут.

Ожидаемый результат:

Нормализация базы данных – это процесс организации структуры данных с целью устранения дублирования, обеспечения целостности и повышения эффективности хранения и обработки данных.

Она достигается путем разделения данных на логически связанные таблицы и применения механизмов связи, таких как первичные (Primary Key) и внешние ключи (Foreign Key).

Основные преимущества нормализации.

1) Минимизация избыточности данных

- уменьшает повторение информации в таблицах;
- снижает объем хранимых данных, что важно для производительности и экономии памяти.

2) Обеспечение целостности данных

- гарантирует, что данные не противоречат друг другу;
- использование ключевых связей предотвращает потерю важной информации.

3) Исключение аномалий работы с данными, которые заключается в возможности ошибок при вставке, обновлении и удалении записей:

- аномалия вставки – нельзя добавить данные, пока не добавлена связанная запись;
- аномалия обновления – изменение одного значения требует корректировки нескольких записей;
- аномалия удаления – удаление одной записи может привести к потере другой связанной информации.

4) Оптимизация скорости запросов

- разделение данных по таблицам позволяет уменьшить нагрузку на систему;
- SQL-операторы (JOIN, индексы) работают быстрее, так как запросы обращаются к конкретным данным.

5) Масштабируемость и легкость поддержания структуры

- при добавлении новых типов данных не требуется менять всю структуру БД;
- легче вносить изменения и обеспечивать соответствие бизнес-требованиям.

При проектировании информационных систем нормализация базы данных очень важна. Нормализация - это обязательный этап проектирования информационных систем, обеспечивающий их надежность, удобство управления и высокую производительность.

Использование нормализации позволяет избежать проблем с обработкой данных, исключить избыточность и построить гибкую и эффективную структуру хранения информации.

Критерии оценивания:

- описана цель и необходимость нормализации;
- указаны основные преимущества;
- приведен понятный пример.

Если студент упоминает только определение, но не объясняет практическую пользу нормализации, ответ считается неполным.

Компетенции (индикаторы): ПК-1, ПК-2

Экспертное заключение

Представленный комплект оценочных материалов по дисциплине «Программирование и администрирование систем управления базами данных» соответствует требованиям ФГОС ВО.

Предлагаемые оценочные материалы адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 09.03.04 Программная инженерия.

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

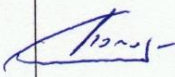
Разработанные и представленные для экспертизы оценочные материалы рекомендуются к использованию в процессе подготовки обучающихся по указанному направлению.

Председатель учебно-методической комиссии
института компьютерных систем
и информационных технологий



Ветрова Н.Н.

Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)
1.	Дополнен комплект оценочных материалов	протокол заседания кафедры компьютерных систем и сетей № <u>8</u> от <u>10.03.2025</u>	 С.В. Попов