Ответы на экзаменационные вопросы по курсу “Базы данных” для направления ИСИТ (2025)

1. Основные понятия теории баз данных: база данных, система управления базами данных, основные требования к информации в БД, модели данных, логическая схема БД, основная терминология реляционных баз данных.

База данных (БД) — это совокупность данных, организованных по определённым правилам, которые обеспечивают их сохранность, целостность и доступность.

Система управления базами данных (СУБД) — это программное обеспечение, обеспечивающее управление базами данных, включая создание, модификацию, удаление данных, а также управление пользователями и правами доступа.

К основным требованиям к информации в БД относятся:

* Надёжность (устойчивость к сбоям),
* Целостность (отсутствие противоречий),
* Согласованность (логическая непротиворечивость),
* Доступность (возможность быстрого и удобного получения данных),
* Безопасность (защита от несанкционированного доступа).

Модели данных:

* Иерархическая: данные представлены в виде древовидной структуры.
* Сетевая: данные организованы в виде графа.
* Реляционная: данные представлены в виде таблиц (наиболее распространённая модель).
* Объектно-ориентированная: расширяет реляционную модель за счёт хранения объектов и их поведения.

Логическая схема — это описание структуры данных, включая таблицы, поля, типы данных и связи между таблицами, независимое от физической реализации. Основная терминология реляционных БД: таблица, строка (запись), столбец (поле), первичный ключ, внешний ключ, уникальность, индекс и т.д.

1. Язык SQL:

SQL (Structured Query Language) — язык структурированных запросов, используемый для взаимодействия с реляционными базами данных. Он включает следующие подъязыки:

* DDL (Data Definition Language): создание и изменение объектов (CREATE, ALTER, DROP);
* DML (Data Manipulation Language): работа с данными (SELECT, INSERT, UPDATE, DELETE);
* DCL (Data Control Language): управление доступом (GRANT, REVOKE);
* TCL (Transaction Control Language): управление транзакциями (COMMIT, ROLLBACK, SAVEPOINT).

1. Системные базы данных в SQL Server:

СУБД SQL Server содержит несколько системных баз данных:

* master — содержит информацию о конфигурации сервера, учетных записях, базах данных;
* model — шаблон для всех новых баз данных;
* msdb — отвечает за задания, планировщик, резервное копирование;
* tempdb — временная база данных, используется для хранения промежуточных данных, временных таблиц и курсоров.

Эти базы являются критически важными для работы SQL Server, особенно master и tempdb.

1. Структура файла базы данных в SQL Server. Файловые группы:

Каждая база данных состоит как минимум из одного основного файла данных (с расширением .mdf) и одного файла журнала транзакций (с расширением .ldf). Вторичные файлы (.ndf) могут использоваться для распределения данных.

Файловые группы позволяют логически объединять файлы данных, например:

* PRIMARY — основная файловая группа (по умолчанию);
* USER\_DEFINED — пользовательские группы для размещения таблиц, индексов и др.

Это даёт гибкость в управлении производительностью и масштабируемостью.

1. Нормализация таблиц базы данных. Нормальные формы таблиц.

Нормализация — это процесс упорядочивания данных в базе данных с целью устранения избыточности и повышения целостности данных.

Основные нормальные формы:

* Первая нормальная форма (1НФ): все поля содержат только атомарные значения (неделимые).
* Вторая нормальная форма (2НФ): таблица находится в 1НФ и все неключевые атрибуты полностью функционально зависят от первичного ключа.
* Третья нормальная форма (3НФ): таблица находится во 2НФ и не содержит транзитивных зависимостей.
* Бойс-Кодда нормальная форма (BCNF): усиленная 3НФ.

1. Таблицы и их типы данных Microsoft SQL Server

В SQL Server существуют следующие основные типы данных:

* Числовые: int, bigint, smallint, decimal, numeric, float;
* Строковые: char(n), varchar(n),
* Дата и время: date, datetime, time;
* Булевы: bit;
* Двоичные: binary, varbinary
* Универсальные идентификаторы: uniqueidentifier (GUID).

Выбор типа данных зависит от объема хранимой информации и требований к производительности.

1. Подзапросы. Конструкции IN, EXISTS, ALL, ANY, SOME:

Подзапрос — это запрос, вложенный в другой запрос. Бывают:

* Скалярный (возвращает одно значение),
* Многозначный (список значений),
* Коррелированный (связан с внешним запросом).

Операторы:

* IN: WHERE id IN (SELECT id FROM …);
* EXISTS: проверяет наличие строк в подзапросе;
* ALL: сравнение со всеми значениями подзапроса;
* ANY / SOME: сравнение хотя бы с одним значением из подзапроса.

1. Группировка данных с использованием CUBE, ROLLUP:

Оператор GROUP BY используется для агрегации данных. Расширения:

* ROLLUP — иерархическая агрегация (например, по годам, кварталам);
* CUBE — полная агрегация по всем комбинациям группировок.

Пример: SELECT department, year, SUM(sales) FROM sales GROUP BY CUBE(department, year)

1. Операторы UNION (ALL), INTERSECT, EXCEPT:

* UNION: объединение результатов двух запросов, без дубликатов;
* UNION ALL: то же, но сохраняет дубликаты;
* INTERSECT: возвращает пересечение;
* EXCEPT: возвращает строки, которые есть в первом запросе, но нет во втором.

Пример: SELECT name FROM students UNION SELECT name FROM teachers

1. Операторы соединения таблиц:

Операторы JOIN позволяют объединять строки из двух и более таблиц:

* INNER JOIN: строки, у которых совпадают значения ключей;
* LEFT JOIN: все строки из левой таблицы и совпадающие из правой;
* RIGHT JOIN: наоборот;
* FULL JOIN: объединение всех строк;
* CROSS JOIN: декартово произведение.

Пример: SELECT s.name, c.name FROM students s INNER JOIN courses c ON s.course\_id = c.id

1. Язык T-SQL. Пакеты. Объявление переменных. Операторы присвоения. Оператор цикла while.

T-SQL (Transact-SQL) — это расширение SQL от Microsoft, добавляющее конструкции процедурного программирования. В отличие от Oracle, в T-SQL отсутствуют пакеты, но логика может быть организована через процедуры и функции.

Переменные объявляются с помощью оператора DECLARE. Примеры:

DECLARE @Counter INT;  
SET @Counter = 5;

Также можно использовать SELECT для присваивания:

SELECT @Counter = COUNT(\*) FROM Employees;

Цикл WHILE используется для выполнения повторяющихся операций:

WHILE @Counter > 0  
BEGIN  
 PRINT @Counter;  
 SET @Counter = @Counter - 1;  
END

1. Язык T-SQL. Операторы PRINT, IF-ELSE, CASE. Операторы BEGIN-END, WAITFOR и RETURN:

* PRINT используется для вывода сообщений:

PRINT 'Начало обработки';

* IF-ELSE используется для ветвления логики:

IF @Age >= 18  
 PRINT 'Взрослый';  
ELSE  
 PRINT 'Несовершеннолетний';

* CASE позволяет реализовать множественные условия:

SELECT Name,  
 CASE Gender  
 WHEN 'M' THEN 'Мужчина'  
 WHEN 'F' THEN 'Женщина'  
 ELSE 'Неизвестно'  
 END AS GenderDesc  
FROM People;

* BEGIN-END группируют блоки кода.
* WAITFOR задерживает выполнение:

WAITFOR DELAY '00:00:10';

* RETURN завершает выполнение процедуры и возвращает код:

RETURN 1;

1. Язык T-SQL. Обработка ошибок в конструкциях TRY-CATCH. Функция RAISERROR:

Обработка ошибок осуществляется с помощью блоков TRY…CATCH:

BEGIN TRY  
 -- потенциально ошибочная операция  
 UPDATE Accounts SET Balance = Balance - 100 WHERE ID = 1;  
END TRY  
BEGIN CATCH  
 PRINT 'Ошибка: ' + ERROR\_MESSAGE();  
END CATCH;

Функция RAISERROR позволяет создавать пользовательские ошибки:

RAISERROR ('Недостаточно средств', 16, 1);

Уровень ошибки 16 означает ошибку, вызванную пользователем, 1 — состояние ошибки.

1. Локальные и глобальные временные таблицы в SQL Server:

* Локальная временная таблица (начинается с #): доступна только текущей сессии.
* Глобальная временная таблица (##): доступна всем сессиям, пока существует хотя бы одно подключение к ней.

Примеры:

CREATE TABLE #TempTable (ID INT);  
INSERT INTO #TempTable VALUES (1);

Таблицы автоматически удаляются при завершении сессии или сервера.

1. Курсоры в SQL Server. Объявление курсора. Общая схема работы с курсором: declare, open, fetch, close, deallocate. Типы курсоров.

Курсоры позволяют обрабатывать строки по одной. Используются, когда невозможно использовать наборные операции. Схема работы:

DECLARE cursor\_name CURSOR FOR  
SELECT Name FROM Employees;  
  
OPEN cursor\_name;  
FETCH NEXT FROM cursor\_name INTO @Name;  
WHILE @@FETCH\_STATUS = 0  
BEGIN  
 PRINT @Name;  
 FETCH NEXT FROM cursor\_name INTO @Name;  
END  
  
CLOSE cursor\_name;  
DEALLOCATE cursor\_name;

Типы курсоров: STATIC, DYNAMIC, FORWARD\_ONLY, FAST\_FORWARD. Выбор зависит от требований к производительности и обновляемости данных.

1. Хранимые процедуры и функции T-SQL. Создание хранимых процедур. Передача параметров. Входные и выходные параметры.

Хранимые процедуры — это предварительно скомпилированные блоки SQL-кода, хранящиеся в БД. Их можно вызывать с параметрами. Пример создания:

CREATE PROCEDURE GetEmployeeById  
 @ID INT,  
 @Name NVARCHAR(100) OUTPUT  
AS  
BEGIN  
 SELECT @Name = Name FROM Employees WHERE ID = @ID;  
END

Вызов процедуры:

DECLARE @Name NVARCHAR(100);  
EXEC GetEmployeeById 1, @Name OUTPUT;  
PRINT @Name;

1. Хранимые процедуры и функции T-SQL. Виды функций. Создание функций. Передача параметров.

SQL Server поддерживает:

* Скалярные функции (возвращают одно значение),
* Многострочные табличные функции,
* Встроенные табличные функции.

Пример скалярной функции:

CREATE FUNCTION GetBonus (@Salary DECIMAL(10,2))  
RETURNS DECIMAL(10,2)  
AS  
BEGIN  
 RETURN @Salary \* 0.1;  
END

Использование:

SELECT dbo.GetBonus(Salary) FROM Employees;

1. Индексы. Назначение и применение индексов. Виды индексов. Применение различных видов индексов. Оптимизация запросов.

Индексы — это структуры, ускоряющие доступ к данным. Основные типы:

* Кластерный (данные отсортированы по индексу);
* Некластерный (хранятся отдельно от данных);
* Уникальный (гарантирует уникальность);
* Полнотекстовый (поиск по тексту).

Создание:

CREATE INDEX IX\_Employees\_Name ON Employees(Name);

Индексы оптимизируют выполнение WHERE, JOIN, ORDER BY. Однако их избыток замедляет INSERT/UPDATE.

1. План запроса. Этапы обработки select запроса. Понятие стоимости запроса. Понятия селективности и плотности. Индексы. Реорганизация, перестроение, включение и отключение индексов.

План запроса — пошаговое описание того, как SQL Server выполнит SELECT-запрос. Этапы:

* Анализ,
* Оптимизация (выбор плана),
* Исполнение (доступ к данным).

Селективность — доля строк, соответствующих условию. Чем выше селективность (меньше строк), тем эффективнее индекс. Плотность — мера уникальности значений в столбце.

SQL Server может реорганизовать (REORGANIZE) или перестроить (REBUILD) индекс.

1. Триггеры. Типы триггеров. Создание after-триггера.

Триггер — это процедура, автоматически выполняемая при изменении таблицы. AFTER-триггеры срабатывают после выполнения INSERT, UPDATE, DELETE:

CREATE TRIGGER trgAfterInsert ON Employees  
AFTER INSERT  
AS  
BEGIN  
 PRINT 'Сотрудник добавлен';  
END

Триггеры полезны для проверки правил и ведения журналов.

1. Триггеры. Создание и назначение instead of-триггеров. Таблицы inserted, deleted.

INSTEAD OF-триггеры — это триггеры, которые выполняются вместо стандартной операции INSERT, UPDATE или DELETE. Они используются, например, в представлениях, которые не поддерживают прямые обновления, или когда требуется полная замена действия.

Пример:

CREATE TRIGGER trgInsteadOfDelete  
ON Employees  
INSTEAD OF DELETE  
AS  
BEGIN  
 PRINT 'Удаление запрещено политикой безопасности';  
END;

Внутри триггера доступны виртуальные таблицы:

* inserted — содержит новые строки (при INSERT, UPDATE);
* deleted — содержит удаляемые или старые строки (при DELETE, UPDATE).

Они позволяют анализировать изменения и откатывать или переопределять действия.

1. Транзакции. Свойства ACID. Транзакции. Уровни изолированности транзакций. Функция trancount.

Транзакции позволяют выполнять несколько операций над БД как одну логическую единицу. Свойства ACID:

* Атомарность (Atomicity): изменения либо все фиксируются, либо откатываются.
* Согласованность (Consistency): переход от одного корректного состояния к другому.
* Изолированность (Isolation): параллельные транзакции не мешают друг другу.
* Долговечность (Durability): зафиксированные изменения не теряются.

Пример:

BEGIN TRAN;  
UPDATE Accounts SET Balance = Balance - 100 WHERE ID = 1;  
UPDATE Accounts SET Balance = Balance + 100 WHERE ID = 2;  
COMMIT;

@@TRANCOUNT — возвращает количество активных транзакций:

SELECT @@TRANCOUNT;

1. Операторы TCL. Привилегии. Роли. Назначение привилегий.

TCL (Transaction Control Language) содержит:

* COMMIT — подтверждает изменения;
* ROLLBACK — отменяет изменения;
* SAVEPOINT — устанавливает точку возврата внутри транзакции.

Привилегии — это разрешения на действия с объектами:

GRANT SELECT, UPDATE ON Employees TO User1;  
REVOKE UPDATE ON Employees FROM User1;

Роли — это группировки привилегий, применяемые к пользователям:

CREATE ROLE hr\_manager;  
GRANT SELECT, INSERT TO hr\_manager;  
EXEC sp\_addrolemember 'hr\_manager', 'Alice';

1. Дистрибутивы СУБД Oracle. Установка СУБД Oracle 12с на Windows. Global Database Name и SID.

Oracle 12c поставляется как сетевой дистрибутив или ISO-образ. Установка включает:

* выбор типа установки (одинарная база, CDB);
* настройку пути и учетных записей;
* указание Global Database Name (например, orcl.local);
* SID (System Identifier) — локальный идентификатор экземпляра (например, ORCL).

SID используется в файлах listener.ora и tnsnames.ora, а также при подключении.

1. Основные системные пользователи. Основные специальные привилегии. Роль DBA.

Основные пользователи Oracle:

* SYS — владелец системных объектов (имеет доступ ко всему);
* SYSTEM — административный пользователь (создание объектов, пользователей);

Привилегии:

* CREATE SESSION — вход в БД;
* CREATE TABLE, VIEW, PROCEDURE — создание объектов.

Роль DBA включает все системные привилегии. Назначается так:

GRANT DBA TO user\_name;

1. Понятия базы данных и экземпляра базы данных

База данных Oracle — это набор физических файлов данных. Экземпляр — набор процессов и SGA (Shared Global Area), управляющий этими файлами.

На одном сервере может быть несколько экземпляров и баз. В CDB — один экземпляр обслуживает несколько PDB.

1. Запуск и останов экземпляра базы данных Oracle

Управление осуществляется через SQL\*Plus:

STARTUP; -- запуск  
SHUTDOWN IMMEDIATE; -- завершение

Стадии:

* NOMOUNT — чтение pfile/spfile;
* MOUNT — открытие control file;
* OPEN — подключение к файлам данных.

1. Словарь базы данных: назначение, применение, основные представления

Словарь — это система служебных представлений Oracle, описывающих объекты базы данных.

Категории:

* USER\_ — информация о текущем пользователе;
* ALL\_ — доступные объекты других пользователей;
* DBA\_ — админская информация (при наличии прав DBA).

Примеры:

* USER\_TABLES, ALL\_OBJECTS, DBA\_USERS.

1. Мультиарендная архитектура Oracle Multitenant

Multitenant в Oracle позволяет запускать несколько логически изолированных PDB в одном экземпляре CDB.

Преимущества:

* Централизованное управление;
* Независимые резервные копии и миграции;
* Эффективное использование ресурсов.

Команда подключения к PDB:

ALTER SESSION SET CONTAINER = pdb1;

1. Файлы экземпляра Oracle. Файл параметров, управляющие файлы, файлы паролей, файлы трассировки.

* PFILE/SPFILE — файлы параметров запуска (настройки экземпляра);
* Control files — определяют структуру БД, подключаемые файлы;
* Password file — используется для удалённой аутентификации администраторов;
* Alert log — журнал событий;
* Trace files — технические логи, создаваемые при ошибках.

Пример строки в spfile:

\*.control\_files='/u01/app/oracle/oradata/orcl/control01.ctl'

1. Файлы базы данных Oracle. Файлы данных, журналы, архивы

Oracle использует несколько типов файлов для хранения и обслуживания базы данных:

* Файлы данных (Datafiles) — содержат таблицы, индексы, и другую пользовательскую информацию;
* Журналы (Redo Log Files) — хранят все изменения данных, обеспечивая возможность восстановления после сбоев;
* Архивы (Archived Redo Logs) — резервные копии redo-журналов, создаются при включённой архивации.

Пример:

/data/oracle/oradata/orcl/users01.dbf — файл данных  
/data/oracle/oradata/orcl/redo01.log — файл журнала  
/data/oracle/archive/redo01\_123.arc — архивный журнал

1. Абстрактная модель Oracle. Логическая структура внешней памяти

Логическая структура включает в себя:

* Табличные пространства (Tablespaces),
* Сегменты (Segments) — структуры для хранения таблиц, индексов и т.д.,
* Экстенты (Extents) — единицы хранения внутри сегмента,
* Блоки данных (Blocks) — минимальная единица ввода-вывода.

Эта структура позволяет гибко управлять хранением данных.

1. Абстрактная модель Oracle. Физическая структура внешней памяти:

Физическая структура состоит из:

* Файлов данных (Datafiles),
* Контрольных файлов (Control files),
* Журналов повторения (Redo log files).

Связь с логической структурой обеспечивается через табличные пространства, которые логически связаны с файлами данных.

1. Абстрактная модель Oracle. Структура SGA

SGA (System Global Area) — это общая область памяти, используемая всеми процессами экземпляра. Основные компоненты:

* Shared Pool — хранит кэш запросов и данные словаря;
* Database Buffer Cache — кэширует блоки данных;
* Redo Log Buffer — буфер для redo-записей;
* Large Pool, Java Pool — дополнительные пулы для специализированных задач.

1. Абстрактная модель Oracle. Серверные процессы Oracle

Серверные процессы (Server Processes) — обрабатывают SQL-запросы от пользователей, взаимодействуют с SGA и файлами данных.

Их роль:

* Выполнение команд SELECT, INSERT и др.;
* Управление буферизацией;
* Ведение логов.

Могут быть выделенными (Dedicated) или разделяемыми (Shared Server).

1. Абстрактная модель Oracle. Фоновые процессы Oracle

Некоторые ключевые фоновые процессы:

* DBWn — записывает изменённые блоки из кэша на диск;
* LGWR — записывает redo-записи из буфера в redo-журналы;
* CKPT — отвечает за контрольные точки;
* SMON — очищает неиспользуемые пространства и восстанавливает сессии;
* PMON — завершает аварийно завершённые процессы пользователей.

Каждый из них выполняет критически важные функции для обеспечения работоспособности базы.

1. Процесс-слушатель Oracle и его основные параметры

Listener — это процесс, принимающий подключения к экземпляру базы данных Oracle. Настраивается через файл listener.ora. Основные параметры:

* PORT (обычно 1521),
* HOST — имя сервера,
* SERVICE\_NAME — имя базы данных.

Пример:

LISTENER =  
 (DESCRIPTION\_LIST =  
 (DESCRIPTION =  
 (ADDRESS = (PROTOCOL = TCP)(HOST = myhost)(PORT = 1521))  
 )  
 )

1. Сетевые настройки Oracle. Установление соединения по сети

Соединение между клиентом и сервером Oracle осуществляется через Oracle Net.

Файлы конфигурации:

* tnsnames.ora — содержит алиасы и параметры подключения;
* sqlnet.ora — настройки сети и протоколов;
* listener.ora — настройки слушателя.

Пример записи в tnsnames.ora:

ORCL =  
 (DESCRIPTION =  
 (ADDRESS = (PROTOCOL = TCP)(HOST = myhost)(PORT = 1521))  
 (CONNECT\_DATA =  
 (SERVER = DEDICATED)  
 (SERVICE\_NAME = orcl.local)  
 )  
 )

1. Табличные пространства СУБД Oracle и их основные параметры

Табличное пространство — логическая структура хранения данных. Основные типы:

* SYSTEM — содержит системные объекты;
* USERS — для пользовательских таблиц;
* TEMP — используется для временных операций;
* UNDO — откат транзакций.

Параметры:

* AUTOEXTEND — автоматическое расширение файла;
* MAXSIZE — максимальный размер.

Пример:

CREATE TABLESPACE myspace DATAFILE 'myspace01.dbf' SIZE 100M AUTOEXTEND ON NEXT 10M;

1. Роли и привилегии СУБД Oracle и их основные параметры

Привилегии делятся на:

* Системные (CREATE TABLE, ALTER USER),
* Объектные (SELECT, INSERT на конкретные объекты).

Роли позволяют объединять привилегии:

CREATE ROLE analyst;  
GRANT SELECT ON employees TO analyst;  
GRANT analyst TO user1;

Для безопасности можно задать пароль на роль:

CREATE ROLE secure\_role IDENTIFIED BY pass123;

1. Пользователь СУБД Oracle и его основные параметры

Пользователь в Oracle — это сущность, представляющая учетную запись, которая используется для доступа к базе данных. Основные параметры пользователя:

* **Имя пользователя**: уникальное имя, идентифицирующее учетную запись.
* **Пароль**: используется для аутентификации.
* **Роль**: набор привилегий, назначенных пользователю.
* **Профиль**: задает ограничения на ресурсы и правила безопасности.
* **Привилегии**: права доступа к объектам базы данных.
* **Табличное пространство**: место хранения данных, создаваемых пользователем.

1. Профиль безопасности СУБД Oracle и его основные параметры

Профиль безопасности Oracle предназначен для управления ограничениями на использование ресурсов и соблюдения правил безопасности. Основные параметры:

* **LIMIT**: ограничивает использование ресурсов (например, количество сессий, время CPU).
* **PASSWORD\_LIFE\_TIME**: срок действия пароля.
* **PASSWORD\_REUSE\_TIME**: время, через которое можно повторно использовать старый пароль.
* **FAILED\_LOGIN\_ATTEMPTS**: число неудачных попыток входа до блокировки.
* **PASSWORD\_VERIFY\_FUNCTION**: функция проверки сложности пароля.

1. Таблица в СУБД Oracle и ее основные параметры. Типы данных базы данных. Ограничения целостности в таблицах

Таблица — это объект базы данных, используемый для хранения структурированных данных. Основные параметры:

* **Имя таблицы**.
* **Колонки** (их имена, типы данных, ограничения).
* **Табличное пространство**.
* **Ограничения целостности**:
  + **NOT NULL**: поле не может быть пустым.
  + **UNIQUE**: значение должно быть уникальным.
  + **PRIMARY KEY**: уникальный идентификатор строки.
  + **FOREIGN KEY**: ссылка на другую таблицу.
  + **CHECK**: проверка условий.

Типы данных:

* Числовые: NUMBER, INTEGER.
* Символьные: CHAR, VARCHAR2.
* Дата/время: DATE, TIMESTAMP.
* Двоичные: RAW, BLOB.

1. Временные таблицы СУБД Oracle

Временные таблицы предназначены для хранения данных, которые действительны только в рамках одной сессии или транзакции. Типы:

* **ON COMMIT DELETE ROWS**: данные удаляются после фиксации транзакции.
* **ON COMMIT PRESERVE ROWS**: данные сохраняются до окончания сессии.

1. Индексы базы данных СУБД Oracle. Виды и особенности применения индексов

Индекс — это структура, ускоряющая доступ к данным. Виды:

* **B-tree индексы**: стандартный тип.
* **Bitmap индексы**: эффективны для столбцов с небольшим количеством уникальных значений.
* **Unique индексы**: обеспечивают уникальность значений.
* **Function-based индексы**: основаны на функциях.
* **Composite индексы**: включают несколько колонок.

1. Последовательность СУБД Oracle и ее параметры

Последовательность — это объект для генерации уникальных числовых значений. Параметры:

* **START WITH**: начальное значение.
* **INCREMENT BY**: шаг увеличения.
* **MAXVALUE/MINVALUE**: максимальное/минимальное значение.
* **CYCLE/NOCYCLE**: перезапуск последовательности при достижении максимума.

1. Кластер и его параметры

Кластер — это группа таблиц, которые хранят данные совместно, чтобы минимизировать доступ к диску. Параметры:

* **Cluster key**: общий ключ.
* **Табличное пространство**.
* **Размер блока**.

1. Представление в СУБД Oracle и его параметры

Представление — это виртуальная таблица, которая формируется на основании SQL-запроса. Параметры:

* **WITH CHECK OPTION**: ограничивает изменения данных.
* **READ ONLY**: запрет на модификацию.

1. Материализованное представление и его параметры

Материализованное представление — это копия данных, которая обновляется периодически. Параметры:

* **REFRESH ON DEMAND** или **REFRESH ON COMMIT**.
* **FAST** или **COMPLETE**: стратегии обновления.

1. Частные и публичные синонимы СУБД Oracle

Синонимы — это альтернативные имена для объектов.

* **Частные**: доступны только владельцу.
* **Публичные**: доступны всем пользователям.

1. Структура программы языка PL/SQL. Анонимные и именованные блоки

Программа состоит из:

* **DECLARE**: объявление переменных.
* **BEGIN**: основной блок.
* **EXCEPTION**: обработка ошибок.
* **END**: завершение блока.  
  Анонимные блоки не имеют имени. Именованные блоки могут быть вызваны повторно.

1. Типы данных, основные операции, константы языка PL/SQL. Псевдостолбцы

PL/SQL поддерживает:

* **Типы данных**:
  + Скаляры: NUMBER, VARCHAR2, BOOLEAN.
  + Композиционные: записи, таблицы.
  + Референции: указатели на объекты.
* **Основные операции**: математические, строковые, сравнения.
* **Псевдостолбцы**: ROWNUM, SYSDATE, NEXTVAL.

1. Поддержка национальных языков в СУБД Oracle. Наборы символов. Байтовая и символьная семантика символов

* **Наборы символов**: Unicode, AL32UTF8.
* **Семантика**:
  + Байтовая: измеряет данные в байтах.
  + Символьная: измеряет данные в символах.
* **Национальные настройки**: NLS-параметры.

1. Связанные объявления переменных: инструкция %TYPE, инструкция %ROWTYPE

* %TYPE: для переменных с типом данных столбца.
* %ROWTYPE: для переменных, представляющих всю строку таблицы.

1. Локальные процедуры и функции языка PL/SQL

Локальные процедуры и функции определяются внутри блока и доступны только в рамках этого блока.

1. Использование записей в PL/SQL. Вложенные записи

Записи — это композиционные типы данных. Вложенные записи позволяют создавать записи внутри других записей.

1. Операторы управления, операторы цикла языка PL/SQL

* **Управление**: IF, CASE.
* **Циклы**: FOR, WHILE, LOOP.

1. Курсоры. Виды курсоров. Схемы обработки курсора. Атрибуты курсора. Курсоры с параметрами. Динамические курсоры

* **Виды**: явные, неявные, динамические.
* **Атрибуты**: %FOUND, %NOTFOUND, %ROWCOUNT.

1. Курсоры. Курсорные переменные. Курсорные подзапросы. Использование конструкции CURRENT OF в курсорах

* **Курсорные переменные**: указывают на результат запроса.
* **CURRENT OF**: используется для обновления строк.

1. Обработка исключений в PL/SQL, стандартные исключения, генерация и обработка исключения

* **Исключения**: NO\_DATA\_FOUND, TOO\_MANY\_ROWS.
* **Обработка**: блок EXCEPTION.

1. Принцип распространения исключений в PL/SQL. Инструкция RAISE\_APPLICATION\_ERROR

Исключения распространяются вверх по стеку вызовов. RAISE\_APPLICATION\_ERROR создает пользовательские ошибки.

1. Встроенные функции языка PL/SQL. Функции работы с датами, текстом и числами

* Даты: SYSDATE, ADD\_MONTHS.
* Текст: SUBSTR, INSTR.
* Числа: ROUND, TRUNC.

1. Коллекции. Массивы переменной длины. Вложенные таблицы. Ассоциативные массивы

* **Массивы**:
  + Вложенные таблицы.
  + Ассоциативные массивы (ключ-значение).

1. Процедурные объекты. Локальные процедуры и функции

Процедурные объекты:

* **Процедура (PROCEDURE)** — именованный блок PL/SQL, выполняющий действия, не возвращает значение.
* **Функция (FUNCTION)** — возвращает значение.
* **Пакет (PACKAGE)** — набор процедур, функций и переменных.
* **Триггер (TRIGGER)** — автоматически выполняемый код при событиях в базе.

1. Процедурные объекты. Хранимые процедуры. Вызов процедур. Входные и выходные параметры, позиционный и параметрический форматы передачи фактических параметров. Значения параметров по умолчанию

Хранимые процедуры — это именованные блоки, хранящиеся в базе. Поддерживаются входные/выходные параметры.

1. Процедурные объекты. Хранимые функции. Параметры функции. Вызов функций. Понятие детерминированной функции. Понятие pipeline функции. Значения параметров по умолчанию

* **Хранимые функции** возвращают значение.
* **Детерминированные функции**: всегда возвращают один и тот же результат для одинаковых входных данных.
* **Pipeline функции** возвращают результат поэлементно.

1. Процедурные объекты. Пакеты. Спецификация и реализация пакета

Пакеты — это группы процедур/функций. Состоят из спецификации и тела.

1. Процедурные объекты. Триггеры. Виды триггеров. Классификация, порядок выполнения и предикаты триггеров. Триггеры замещения. Привилегии для создания триггеров. Включение/отключение триггеров. Псевдозаписи old и new.

Процедурные объекты:

**Процедура (PROCEDURE)** — именованный блок PL/SQL, выполняющий действия, не возвращает значение.

* **Функция (FUNCTION)** — возвращает значение.
* **Пакет (PACKAGE)** — набор процедур, функций и переменных.
* **Триггер (TRIGGER)** — автоматически выполняемый код при событиях в базе.
* **Виды**: до/после операций, для каждой строки.
* **Псевдозаписи**: :OLD, :NEW.

1. Секционирование таблиц. Виды секционирования

* Секционирование (partitioning) — это метод разделения большой таблицы на более мелкие части (секции, партиции) для повышения производительности, удобства управления и масштабируемости.
* **Виды**:
  + Диапазон (range).
  + Список (list).
  + Хэш (hash).

1. Транзакции в СУБД Oracle. Виды транзакций. Понятие автономной транзакции.

* **Транзакции**: логическая единица работы, состоящая из SQL-операций.

**Виды транзакций в Oracle**

1. **Простая транзакция**  
   Обычная транзакция, состоящая из последовательности SQL-операторов, которая завершается COMMIT или ROLLBACK.
2. **Вложенные транзакции**  
   Oracle поддерживает Д-ослабленные транзакции, где вложенная транзакция разделяет ресурсы главной и откатывается вместе с ней, если главная откатывается.

* **Автономная транзакция**: независимая транзакция внутри другой.