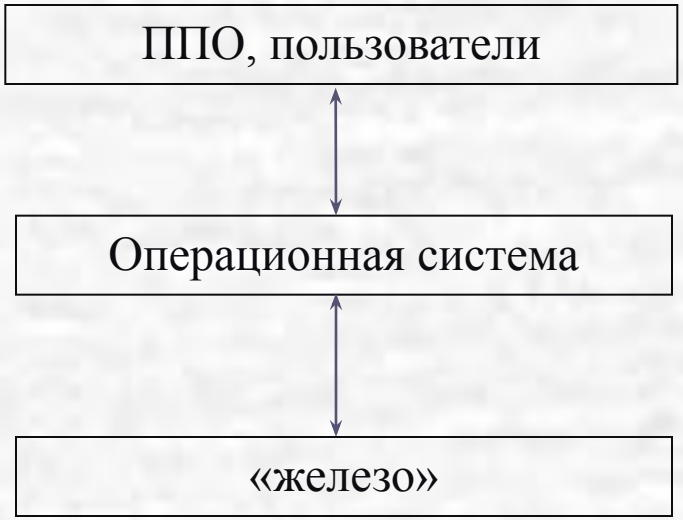
Реляционные системы управления базами данных

Основные концепции.
Особенности СУБД Oracle

Основные функции СУБД



- Обеспечение доступа ППО к базе данных
- Управление базой данных

СУБД

Программные составляющие СУБД включают в себя ядро и сервисные средства (утилиты).

- **Ядро СУБД** – это набор программных модулей, необходимый и достаточный для создания и поддержания БД, то есть универсальная часть, решающая стандартные задачи по информационному обслуживанию пользователей.
- **Сервисные программы** предоставляют пользователям ряд дополнительных возможностей и услуг, зависящих от описываемой предметной области и потребностей конкретного пользователя.

Системой управления базами данных называют программную систему, предназначенную для создания на ЭВМ общей базы данных для множества приложений, поддержания её в актуальном состоянии и обеспечения эффективного доступа пользователей к содержащимся в ней данным в рамках предоставленных им полномочий.

Классификация СУБД

По степени универсальности СУБД делят на два класса:

1. СУБД общего назначения (СУБД ОН)
2. специализированные СУБД (СпСУБД).

Специализированные СУБД создаются в тех случаях, когда ни одна из существующих СУБД общего назначения не может удовлетворительно решить задачи, стоящие перед разработчиками. Причин может быть несколько:

- не достигается требуемого быстродействия обработки данных;
- необходима работа СУБД в условиях жёстких аппаратных ограничений;
- требуется поддержка специфических функций обработки данных.

СпСУБД предназначены для решения конкретной задачи, а приемлемые параметры этого решения достигаются следующим образом:

1. за счёт знания особенностей конкретной предметной области,
2. путём сокращения функциональной полноты системы.

Классификация СУБД

По методам организации хранения и обработки данных СУБД делят на

- **Централизованные**
- **Распределённые**.

Первые работают с БД, которая физически хранится в одном месте (на одном компьютере). Это не означает, что пользователь может работать с БД только за этим же компьютером: доступ может быть удалённым (в режиме клиент–сервер). Большинство централизованных СУБД перекладывает задачу организации удалённого доступа к данным на сетевое обеспечение, выполняя только свои стандартные функции, которые усложняются за счёт одновременности доступа многих пользователей к данным.

По модели данных различают **иерархические**, **сетевые**, **реляционные**, **объектно-реляционные** и **объектно-ориентированные** СУБД.

Требования к реляционным СУБД (по Кодду)

1. **Явное представление данных** (The Information Rule). Информация должна быть представлена в виде данных, хранящихся в ячейках. Данные, хранящиеся в ячейках, должны быть атомарны. Порядок строк в реляционной таблице не должен влиять на смысл данных.
2. **Гарантированный доступ к данным** (Guaranteed Access Rule). К каждому элементу данных должен быть гарантирован доступ с помощью комбинации имени таблицы, первичного ключа строки и имени столбца.
3. **Полная обработка неизвестных значений** (Systematic Treatment of Null Values). Неизвестные значения (**NULL**), отличные от любого известного значения, должны поддерживаться для всех типов данных при выполнении любых операций.

Требования к реляционным СУБД (по Кодду)

4. **Доступ к словарю данных** в терминах реляционной модели (Dynamic On-Line Catalog Based on the Relational Model). **Словарь данных** должен сохраняться в форме реляционных таблиц, и СУБД должна поддерживать доступ к нему при помощи стандартных языковых средств.
5. **Полнота подмножества языка** (Comprehensive Data Sublanguage Rule). Система управления реляционными базами данных должна поддерживать **единственный язык запросов**, который позволяет выполнять все операции работы к данным:

- операции определения данных,
- операции манипулирования данными,
- управление доступом к данным,
- управление транзакциями.

Требования к реляционным СУБД (по Кодду)

6. Поддержка **обновляемых представлений** (View Updating Rule). Обновляемое представление должно поддерживать все операции манипулирования данными, которые поддерживают реляционные таблицы: операции выборки, вставки, модификации и удаления данных.
7. Наличие **высокоуровневых операций** управления данными (High-Level Insert, Update, and Delete). Операции вставки, модификации и удаления данных должны поддерживаться не только по отношению к одной строке реляционной таблицы, но по отношению к любому множеству строк.

Требования к реляционным СУБД (по Кодду)

8. **Физическая независимость данных** (Physical Data Independence). Приложения не должны зависеть от используемых способов хранения данных на носителях, от аппаратного обеспечения компьютеров, на которых находится реляционная база данных.
9. **Логическая независимость данных** (Logical Data Independence). Представление данных в приложении не должно зависеть от структуры реляционных таблиц.

Требования к реляционным СУБД (по Кодду)

10. **Независимость контроля целостности** (Integrity Independence). Вся информация, необходимая для поддержания целостности, должна находиться в словаре данных. СУБД должна выполнять проверку заданных **ограничений целостности** и автоматически поддерживать целостность данных.
11. **Независимость от распределенности** (Distribution Independence). База данных может быть распределенной, может находиться на нескольких компьютерах, и это не должно оказывать влияние на приложения.
12. **Согласование языковых уровней** (Non-Subversion Rule). Не должно быть иного средства доступа к данным, отличного от стандартного языка работы с данными. Если используется низкоуровневый язык доступа к данным, он не должен игнорировать правила безопасности и целостности, которые поддерживаются языком более высокого уровня.

Требования к составу и функциям СУБД

1. Хранение, извлечение и обновление данных.
2. Каталог (ССД), доступный конечным пользователям.

Обычно в системном каталоге хранятся следующие сведения:

3. имена, типы и размеры элементов данных;
4. имена связей;
5. накладываемые на данные ограничения поддержки целостности;
6. имена пользователей, которым предоставлено право доступа к данным;
7. внешняя, концептуальная и внутренняя схемы и отображения между ними;
8. статистические данные, например частота транзакций и счетчики обращений к объектам базы данных.

Преимущества наличия ССД

- Информация о данных может быть централизованно собрана и сохранена, что позволит контролировать доступ к этим данным.
- Можно определить смысл данных, что поможет другим пользователям понять их предназначение.
- Упрощается общение, так как имеются точные определения смысла данных.
- В системном каталоге также могут быть указаны один или несколько пользователей, которые являются владельцами данных или обладают правом доступа к ним.
- Благодаря централизованному хранению избыточность и противоречивость описания отдельных элементов данных могут быть легко обнаружены.
- Внесенные в базу данных изменения могут быть запротоколированы.
- Последствия любых изменений могут быть определены еще до их внесения, поскольку в системном каталоге зафиксированы все существующие элементы данных, установленные между ними связи, а также все их пользователи.
- Меры обеспечения безопасности могут быть дополнительно усилены.
- Появляются новые возможности организации поддержки целостности данных.
- Может выполняться аудит хранимой информации.

Системный словарь данных Oracle

Хранит всю информацию о структуре, информационных объектах и отношениях в конкретной базе данных. Словарь данных представляет собой набор таблиц и вспомогательных объектов (индексов, кластеров, синонимов, представлений, последовательностей), информация о которых также хранится в таблицах словаря.

Логически словарь данных разделяется на:

- ✔базовые таблицы;
- ✔представления базовых таблиц;
- ✔динамические таблицы и их представления.

Всего словарь данных включает более 100 базовых таблиц, которые расположены в табличном пространстве SYSTEM и нигде более. Их имена включают символ '$' (поэтому его не рекомендуется использовать в названиях небазовых объектов), например:

AUD$	– таблица audit-информации;	FILE$	– таблица файлов;
USER$	– таблица пользователей;	IND$	– таблица индексов;
OBJ$	– таблица объектов;	SEG$	– таблица сегментов;
SYN$	– таблица синонимов;	TAB$	– таблица таблиц;
TS$ – таблица табличных областей;		VIEW$	– таблица представлений.

Работа с системным словарём

Для получения информации из словаря данных пользователям предоставлены представления базовых таблиц. Они разбиты на три группы:

- DBA – представления, предназначенные пользователям, являющимися АБД, то есть которым присвоена роль DBA. По этим представлениям предоставляется наиболее полная информация из словаря данных;
- USER – представления, по которым каждый пользователь получает информацию о тех объектах, которыми владеет;
- ALL – представления, дающие каждому пользователю всю информацию об объектах, к которым ему разрешен доступ.

Например:

- DBA/ALL/USER_INDEXES – все/доступные/пользовательские индексы;
- DBA/ALL/USER_IND_COLUMNS – все/доступные/пользовательские колонки индексов;
- DBA/ALL/USER_OBJECTS – все/доступные/пользовательские объекты;
- DBA/ALL/USER_SYNONYMS – все/доступные/пользовательские синонимы;
- DBA/ALL/USER_TABLES – все/доступные/пользовательские таблицы;
- DBA/ALL/USER_TAB_COLUMNS – все/доступные/пользовательские колонки таблиц;
- DBA/ALL/USER_TAB_PRIVS – все/доступные/пользовательские привилегии на таблицы;
- DBA/ALL/USER_VIEWS – все/доступные/пользовательские представления.

Работа с системным словарём

Некоторые представления (по смыслу их применения) присутствуют только в одной или двух группах. Наиболее характерно это для DBA-представлений, например:

- DBA_DATA_FILES – данные о физических файлах базы и журналов;
- DBA/USER_FREE_SPACE – свободная память в табличных пространствах (вся и доступная конкретному пользователю);
- DBA_PROFILES – перечень вариантов "стоимости" системных ресурсов;
- DBA_ROLES – перечень определенных в базе данных ролей.

Примеры извлечения данных из ССД:

```
select table_name
    from user_tables;

select *
    from all_views;

select view_name
    from dba_views;
```

Работа с системным словарём

Важное значение имеет синоним DICT к представлению DICTIONARY. По нему выбираются имена таблиц, представлений, синонимов словаря данных с описаниями, если таковые есть в базе данных. Приведем небольшой фрагмент:

select * from dict;

ALL_CATALOG	Все таблицы, представления, синонимы, последовательности, доступные пользователю
ALL_DB_LINKS	Связи базы данных, доступные пользователю
DBA_OBJECTS	Все объекты в базе данных
DBA_ROLES	Все роли, которые существуют в БД
USER_EXTENTS	Экстенты, принадлежащие пользователю
USER_VIEWS	Определения представлений, принадлежащих пользователю
DUAL	Специальная таблица, содержащая один столбец DUMMY и одну строку
DICT	Синоним для DICTIONARY
TABS	Синоним для USER_TABLES

Работа с системным словарём

АБД открыт доступ к этим таблицам, но работать на этом уровне, за исключением случаев КРАЙНЕЙ необходимости, НИКОГДА НЕ рекомендуется:

- вся информация словаря данных доступна через представления базовых таблиц;
- данные в базовых таблицах представлены без дублирования по правилам внутри системной упорядоченности, без расшифровки;
- количество, названия, размеры столбцов таблиц сделаны без учета достаточной наглядности;
- случайная, намеренная или еще по какой-либо причине КОРРЕКТИРОВКА содержимого базовых таблиц (даже в очевидных случаях, например, хранение данных о давно удаленных табличных пространствах), как правило, приводит к **ПОВРЕЖДЕНИЮ словаря данных, то есть к ПОТЕРЕ всей базы данных.**
- Редчайшее исключение представляет AUD$ (таблица аудиторской информации), из которой следует периодически удалять ненужные записи, поскольку при включенном audit-режиме эта таблица быстро наполняется и может переполнить табличное пространство SYSTEM.

Требования к составу и функциям СУБД

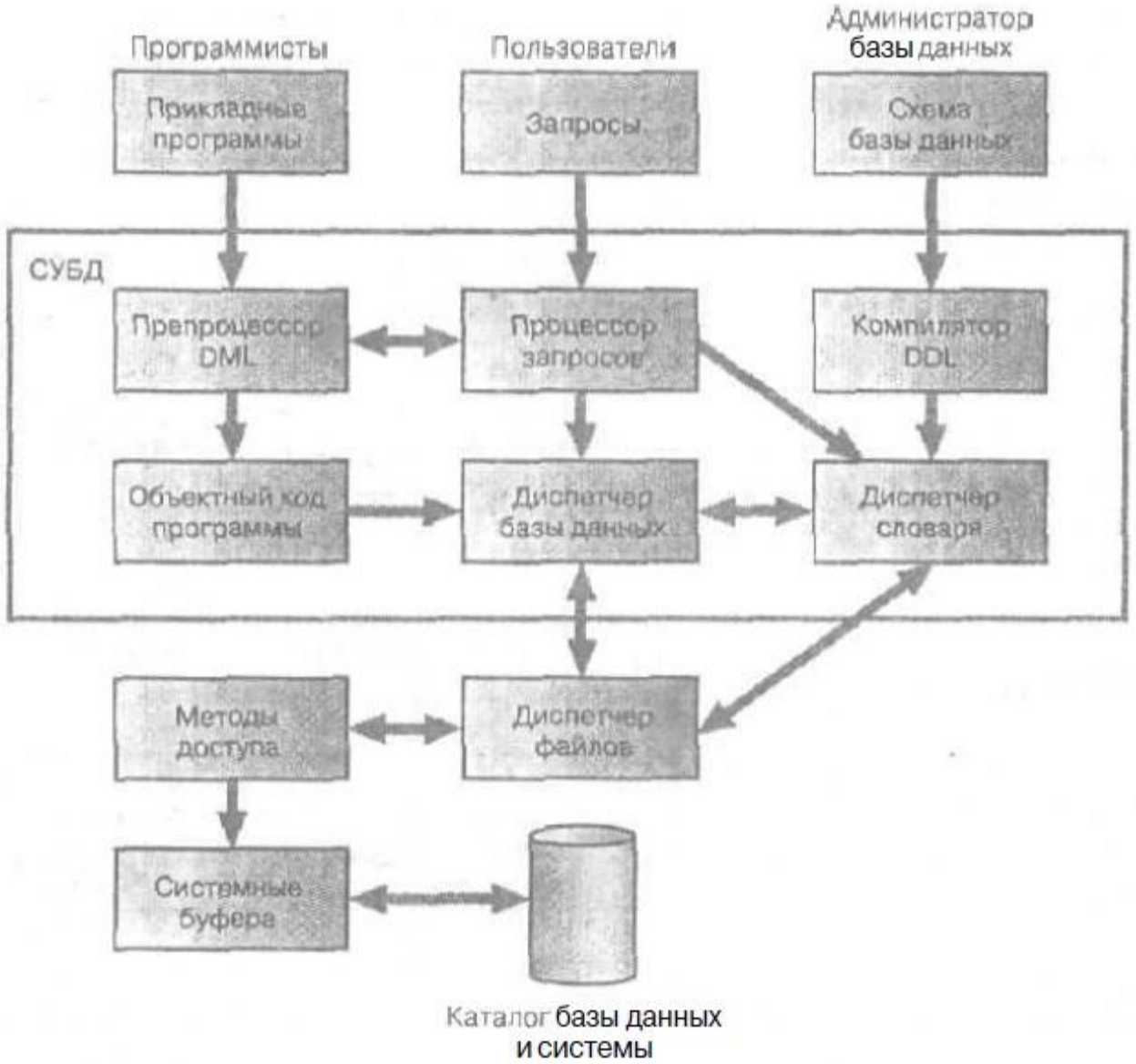
3. Поддержка транзакций.
4. Служба управления параллельной работой.
5. Службы восстановления.
6. Службы контроля доступа к данным.
7. Службы поддержки целостности данных.
8. Службы поддержки независимости от данных.
9. Вспомогательные службы.

Вспомогательные службы

Обычно предназначены для оказания помощи АБД в эффективном администрировании базы данных.

Некоторые примеры подобных утилит.

- Утилиты импортирования, предназначенные для загрузки базы данных из плоских файлов, а также утилиты экспортирования, которые служат для выгрузки базы данных в плоские файлы.
- Средства мониторинга, предназначенные для отслеживания характеристик функционирования и использования базы данных.
- Программы статистического анализа, позволяющие оценить производительность или степень использования базы данных.
- Инструменты реорганизации индексов, предназначенные для перестройки индексов в случае их переполнения.
- Инструменты сборки мусора и перераспределения памяти для физического устранения удаленных записей с запоминающих устройств, объединения освобожденного пространства и перераспределения памяти по мере необходимости.

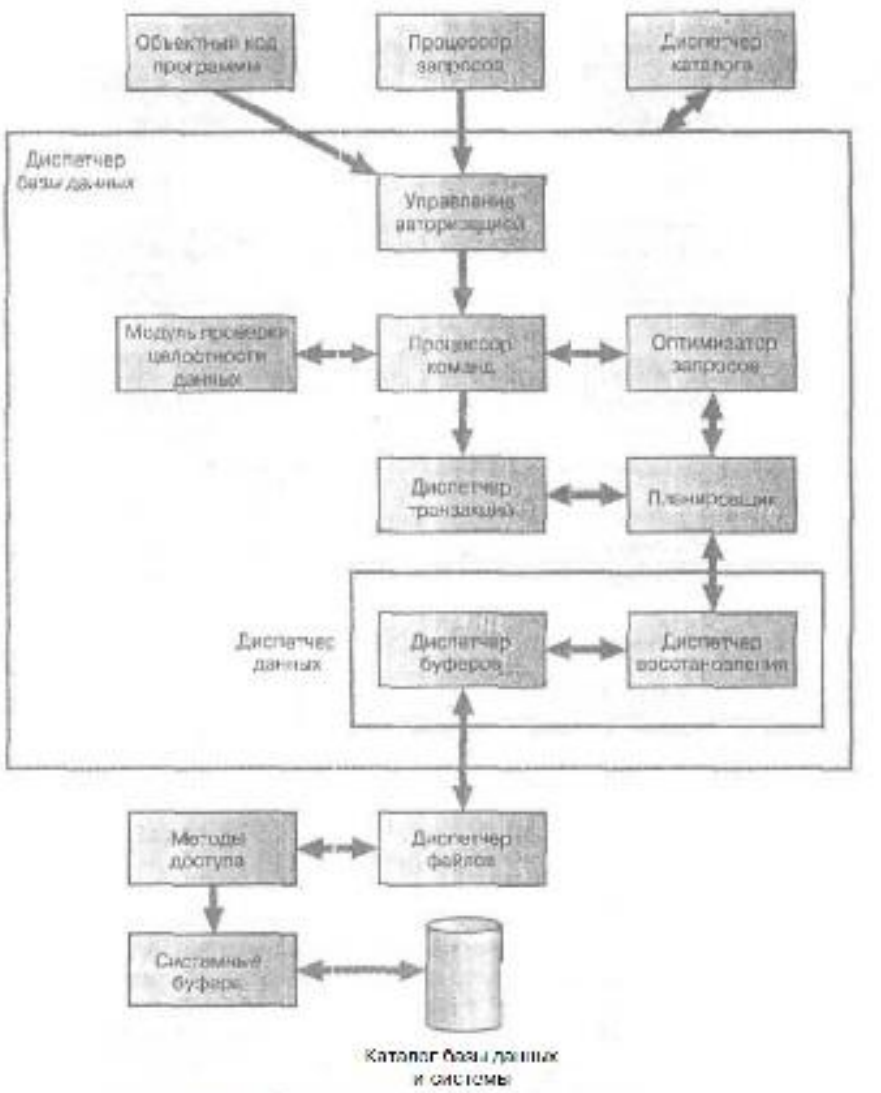


Основные компоненты типичной системы управления базами данных

Основные программные компоненты СУБД

- **Процессор запросов.** Преобразует запросы в последовательность низкоуровневых команд для диспетчера базы данных.
- **Диспетчер базы данных**. Принимает запросы и проверяет внешние и концептуальные схемы для определения тех концептуальных записей, которые необходимы для удовлетворения требований запроса. Затем вызывает диспетчер файлов для выполнения поступившего запроса.
- **Диспетчер файлов**. Манипулирует предназначенными для хранения данных файлами и отвечает за распределение доступного дискового пространства. Он создает и поддерживает список структур и индексов, определенных во внутренней схеме. Если используются хешированные файлы, то в его обязанности входит и вызов функций хеширования для генерации адресов записей.

Диспетчер базы данных



Компоненты диспетчера базы данных

Основные программные компоненты СУБД

- **Препроцессор языка DML**. Этот модуль преобразует внедренные в прикладные программы DML-операторы в вызовы стандартных функций базового языка. Для генерации соответствующего кода препроцессор языка DML должен взаимодействовать с процессором запросов.
- **Компилятор языка DDL**. Преобразует DDL-команды в набор таблиц, содержащих метаданные. Затем эти таблицы сохраняются в системном каталоге, а управляющая информация — в заголовках файлов с данными.
- **Диспетчер словаря**. Управляет доступом к системному каталогу и обеспечивает работу с ним. Системный каталог доступен большинству компонентов СУБД.

Основные программные компоненты СУБД

- **Модуль контроля прав доступа**. Этот модуль проверяет наличие у данного пользователя полномочий для выполнения затребованной операции.
- **Процессор команд**. После проверки полномочий пользователя для выполнения затребованной операции управление передается процессору команд.
- **Средства контроля целостности**. В случае операций, которые изменяют содержимое базы данных, средства контроля целостности выполняют проверку того, удовлетворяет ли затребованная операция всем установленным ограничениям поддержки целостности данных (например, требованиям, установленным для ключей).
- **Оптимизатор запросов**. Этот модуль определяет оптимальную стратегию выполнения запроса.

Основные программные компоненты СУБД

- **Диспетчер транзакций**. Осуществляет требуемую обработку операций, поступающих в процессе выполнения транзакций.
- **Планировщик**. Отвечает за бесконфликтное выполнение параллельных операций с базой данных. Он управляет относительным порядком выполнения операций, затребованных в отдельных транзакциях.
- **Диспетчер восстановления**. Гарантирует восстановление базы данных до непротиворечивого состояния при возникновении сбоев. В частности, он отвечает за фиксацию и отмену результатов выполнения транзакций.
- **Диспетчер буферов**. Отвечает за перенос данных между оперативной памятью и вторичным запоминающим устройством — например, жестким диском или магнитной лентой. Диспетчер восстановления и диспетчер буферов иногда (в совокупности) называют *диспетчером данных,* а сам диспетчер буферов — *диспетчером кэша.*

Основные объекты Oracle

- **База данных** (DATABASE) – объект, который находится на самом верхнем уровне физической организации базы данных Oracle находится объект, который так и называется: база данных (database). База данных состоит из словаря-справочника данных, собственно данных и различных вспомогательных объектов: файла параметров инициализации, управляющего файла, файла сегментов отката и двух файлов журнала транзакций. (Этот перечень может быть расширен, например, за счет копий управляющего файла). База данных может быть создана автоматически при инсталляции СУБД Oracle или вручную с помощью команды CREATE DATABASE.
- **Табличная область** (TABLESPACE) – область памяти, предназначенная для хранения всех объектов БД. Табличная область имеет имя и занимает один или более файлов операционной системы. Создается командой CREATE TABLESPACE. Иногда табличную область называют табличным пространством.

Основные объекты Oracle

- **Пользователь** (USER) – объект, обладающий возможностью создавать и использовать другие объекты Oracle, а также запрашивать выполнение функций сервера. К числу таких функций относятся организация сессии, изменение состояния сервера и базы данных, создание других объектов БД, запросы на выполнение операторов SQL и проч. В СУБД Oracle имя пользователя совпадает с именем схемы. Создается командой CREATE USER.

Каждый объект БД принадлежит тому пользователю, который его создал, и находится в его схеме. Полное имя любого объекта БД (кроме базы данных, табличных областей и пользователей) состоит из имени схемы, в которой он создан, и собственно имени объекта, например:

scott.emp

Здесь scott – имя пользователя (схемы), emp – имя объекта (таблицы "Сотрудники"), а точка – это т.н. квалифицированная ссылка, разделяющая уровни определения.

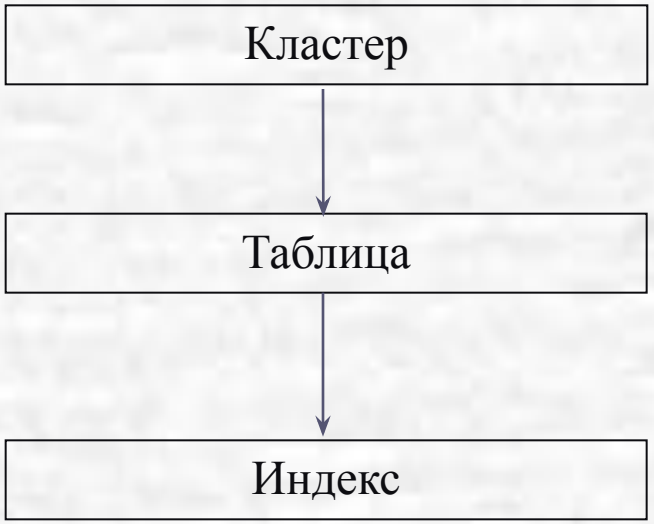
Основные объекты Oracle

- **Кластер** (CLUSTER) – объект, задающий способ совместного хранения данных нескольких таблиц, содержащих информацию, обычно обрабатываемую совместно. Кластеризация таблиц позволяет уменьшить время выполнения выборки. Создается командой CREATE CLUSTER. Включает таблицы с данными.
- **Таблица** (TABLE) является базовой структурой реляционной модели. Как известно, вся информация в базе данных хранится в таблицах. Таблицы состоят из множества поименованных столбцов или атрибутов. Множество значений столбца определено с помощью ограничений целостности, то есть поддерживается ограниченная концепция домена (множества допустимых значений). Таблица может быть пустой или состоять из одной или более строк значений атрибутов. Строки значений атрибутов таблицы называют также *записями* или *кортежами.* Создается командой CREATE TABLE, может быть создана в кластере.

Основные объекты Oracle

- **Индекс** (INDEX) – это объект базы данных, создаваемый для повышения производительности выборки данных. Индекс создается для столбца (столбцов) таблицы и обеспечивает более быстрый доступ к данным этой таблицы за счет упорядочения данных столбца (столбцов) по значению. Создается командой CREATE INDEX.

Кластеры, таблицы и индексы называются *объектами, занимающими память*, т.к. в них хранятся фактографические данные. Им при создании выделяется определенный объем памяти (один или несколько экстентов), который может быть увеличен при добавлении в них данных. Экстент (extent) – это непрерывная область памяти в табличном пространстве. Все экстенты, относящиеся к одному объекту, образуют сегмент (segment).



Основные объекты Oracle

- **Представление** (VIEW) – это поименованная, динамически поддерживаемая сервером выборка данных из одной или нескольких таблиц. В основе представления лежит оператор SELECT, который называется базовым запросом представления. Базовый запрос определяет видимые пользователем данные. Представление позволяет ограничить данные, которые пользователь может модифицировать. Данные в представлении не хранятся: сервер формирует представление каждый раз при обращении к нему (это называется *материализация представления*). Используя представления, администратор безопасности может ограничить доступную пользователям часть базы данных только теми данными, которые реально необходимы им для выполнения работы. Создается командой CREATE VIEW.
- **Последовательность** (SEQUENCE) – это объект, обеспечивающий генерацию уникальных номеров в условиях многопользовательского асинхронного доступа. Обычно элементы последовательности используются для вставки уникальных идентификационных номеров для элементов таблиц базы данных. Создается командой CREATE SEQUENCE.

Основные объекты Oracle

- **Синоним** (SYNONYM) – это альтернативное имя или псевдоним объекта Oracle, который позволяет пользователям базы данных иметь доступ к данному объекту. Синоним может быть частным и общим. Общий (public) синоним позволяет всем пользователям базы данных обращаться к соответствующему объекту по альтернативному имени. При этом имя схемы для обращения к объекту не надо указывать, даже если Вы подключились не как владелец объекта, а из другой схемы. Создается командой CREATE SYNONYM.
- **Роль** (ROLE) – именованная совокупность привилегий, которые могут быть предоставлены пользователям или другим ролям. Используется для эффективного управления разграничением доступа к данным. Oracle поддерживает несколько стандартных или предопределенных ролей (DBA, CONNECT, RESOURCE и др.). Создается командой CREATE ROLE.

Основные объекты Oracle

Специфичными для распределенных систем являются такие объекты Oracle как *снимок* и *связь базы данных*.

- **Снимок** (SNAPSHOT) – локальная копия таблицы удаленной базы данных, которая используется либо для тиражирования (копирования) всей или части таблицы, либо для тиражирования результата запроса данных из нескольких таблиц. Снимки могут быть *модифицируемыми* или *предназначенными только для чтения.* Снимки только для чтения возможно периодически обновлять, отражая изменения основной таблицы. Изменения, сделанные в модифицируемом снимке, распространяются на основную таблицу и другие копии. Создается командой CREATE SNAPSHOT.
- **Связь базы данных** (DATABASE LINK) – это объект базы данных, который позволяет обратиться к объектам удаленной базы данных. Имя связи базы данных можно рассматривать как ссылку на параметры механизма доступа к удаленной базе данных (имя узла, протокол и т. п.). Использование одного имени упрощает работу с объектами удаленной базы данных. Создается командой CREATE DATABASE LINK.

Основные объекты Oracle

Для программирования алгоритмов обработки данных, поддержки сложных правил целостности данных Oracle использует процедурные объекты:

- **Процедура** (PROCEDURE) – это подпрограмма на языке PL/SQL, предназначенная для решения конкретной задачи обработки данных. Создается командой CREATE PROCEDURE.
- **Функция** (FUNCTION) – это подпрограмма на языке PL/SQL, предназначенная для решения конкретной задачи и возвращающая конкретное значение. Создается командой CREATE FUNCTION.
- **Пакет** (PACKAGE) – это поименованный, структурированный набор переменных, процедур и функций, связанных единым функциональным замыслом. Пакет состоит из спецификации и тела пакета. Спецификация содержит описания внешних переменных, констант, типов и подпрограмм, а тело пакета – реализацию подпрограмм и описание внутренних переменных, констант и типов, которые доступны только внутри пакета. Спецификация пакета создается командой CREATE PACKAGE, а тело пакета – CREATE PACKAGE BODY.
- **Триггер** (TRIGGER) – это хранимая процедура, которая автоматически запускается тогда, когда происходит связанное с триггером событие. Обычно события связаны с выполнением операторов INSERT, UPDATE или DELETE в некоторой таблице. Создается командой CREATE TRIGGER.

Физическая структура базы данных Oracle

Параметры среды:

$ORACLE_HOME – имя домашней директории Oracle.

$ORACLE_SID – имя базы данных Oracle.

База данных Oracle включает:

- Управляющие файлы (ctrl1$ORACLE_SID.ctl, ctrl2$ORACLE_SID.ctl,..)
- Файл параметров запуска экземпляра init$ORACLE_SID.ora
- Файл параметров конфигурации базы config$ORACLE_SID.ora
- Журнальные файлы регистрации изменений (log1$ORACLE_SID.dbf, log2$ORACLE_SID.dbf,..)
- Системное табличное пространство (SYSTEM, system$ORACLE_SID.dbf)
- Временное табличное пространство (TEMP, temp$ORACLE_SID.dbf)
- Табличное пространство для данных пользователей (USER, user$ORACLE_SID.dbf)

Структуры оперативной памяти Oracle

- **SGA** – это память, используемая всеми процессами экземпляра. Существует всего одна SGA для экземпляра. Изменения, сделанные в элементах SGA для одного процесса, немедленно становятся доступными для всех процессов, функционирующих в составе этого экземпляра. Создаваемая при запуске экземпляра Oracle, SGA имеет фиксированный размер. Она существует до тех пор, пока экземпляр не будет завершен вручную, или случится перезагрузка операционной системы, или произойдет аварийное завершение (крах) собственно Oracle.

Основными внутренними структурами SGA являются:

- кеш буферов данных (Database Buffer Cache), то есть набор свободных, считанных и модифицированных блоков данных, в которых размещается информация из базы;
- буфер журнала транзакций (Redo Log Buffer);
- разделяемый (общий) буферный пул (Shared Buffer Pool).

Структуры оперативной памяти Oracle. SGA

Кеш буферов данных содержит два списка:

- список наименее используемых в данный момент блоков (LRU – least_recently_used), куда входят считанные с диска, но еще не модифицированные блоки, а также свободные буферы данных;
- список модифицированных (dirty – "грязный"), но еще не записанных на диск блоков.

Обратите внимание:

- обмен "диск-память" всегда производится блоками вне зависимости от их заполненности записями данных и от количества измененных при обработке записей;
- при обращении к данным Oracle сначала проверяет, имеются ли требуемые данные в кеше буферов, и, только если их нет, обращается к диску;
- считанные с диска блоки данных попадают в начало списка LRU. Если они затем модифицируются, то Oracle их переводит в список "грязных" блоков для последующей записи на диск;
- при недостатке в кеше свободных буферов для выполнения очередного запроса Oracle удаляет блоки с "хвоста" списка LRU, как наименее активно используемые, и на их место считывает с диска требуемые блоки данных.

Структуры оперативной памяти Oracle. SGA

- **Буфер журнала регистрации изменений** представляет собой циклически используемую память. В этот буфер поступают все изменения, происходящие в базе с пользовательскими, системными, служебными данными.

  Поскольку журнал регистрации изменений предназначен для восстановления состояния базы данных после аварийных ситуаций, записи журнала несут в себе "старое" и "новое" значения изменившихся элементов, в частности целиком записи данных после операций вставки их в базу или удаления из БД.

  Если обработка данных производится так интенсивно, что буфер журнала переполняется, то есть если процесс LGWR (процесс записи в журнал) не успевает переносить данные из буфера на диск, Oracle начинает сдерживать пользовательские процессы.
- **Разделяемый (общий) буферный пул** включает в себя:
- кеш словаря (Dictionary Cache): хранит в себе наиболее часто (в текущей работе) используемые сведения из системного словаря данных, а именно: названия таблиц и представлений, имена столбцов и типы данных, привилегии и роли пользователей, права доступа к объектам базы данных и др.
- разделяемую (общую) область SQL и PL/SQL (Shared SQL and PL/SQL), которая известна также как "библиотечный кеш" (library cache): включает в себя набор курсоров, то есть структур памяти, в которых хранятся результаты синтаксический разбора и планы выполнения SQL-предложений и блоков PL/SQL.

Структуры оперативной памяти Oracle. PGA

- **PGA** представляет собой область оперативной памяти, выделяемую для обеспечения функционирования отдельного процесса. Имеет место одна и только одна целиком выделяемая процессу и независимая от других процессов PGA для каждого процесса экземпляра. Размер PGA может динамически увеличиваться в процессе функционирования. PGA часто называют глобальной областью процесса (Process Global Area). Когда процесс Oracle нормально завершается, вся память PGA возвращается операционной системе.

PGA процесса Oracle-сервера включает в себя:

- область стека, содержащую переменные и служебную информацию о сеансе;
- частную SQL-область, которую иногда называют "Глобальной областью пользователя" (UGA – User Global Area), в которой производится синтаксический разбор SQL-предложений и блоков PL/SQL. Эта область физически располагается в SGA (вариант архитектуры MTS) или в PGA (архитектура с выделенными серверами). Важно то, что рекурсивные сессии не получают свои собственные UGA, а разделяют UGA породившей их сессии;
- необязательная область сортировки (размером sort_area_size), которая как временная память требуется для хранения промежуточных результатов сортировки данных. Если выделенной памяти недостаточно для проведения сортировки, процесс использует временный сегмент соответствующего табличного пространства.

Процессы экземпляра Oracle

- Набор работающих с базой данных фоновых процессов и порожденная при запуске экземпляра SGA (Системная Глобальная Область) составляют экземпляр Oracle. Все процессы экземпляра функционируют на едином программном ядре ($ORACLE_HOME/bin/oracle) СУБД.

Обычно процессы экземпляра определяют как фоновые (обслуживающие, вспомогательные, дополнительные) и серверные (содержательная обработка запросов).

Минимально необходимым для функционирования Oracle является набор из следующих четырех фоновых процессов:

- ora_pmon_<$ORACLE_SID> – процесс мониторинга внутреннего состояния системы
- ora_dbwr_<$ORACLE_SID> – процесс записи данных в базу данных Oracle
- ora_lgwr_<$ORACLE_SID> – процесс записи в журнал регистрации изменений
- ora_smon_<$ORACLE_SID> – процесс системного мониторинга.

Процессы экземпляра Oracle

1. pmon – фоновый процесс-монитор. Он следит:

- за состоянием процессов в системе (в частности, он отслеживает обращение к серверу со стороны пользователей (connect) и запускает сервер-процессы);
- обнаруживает аварийные ситуации и "мертвые" блокировки сервер-процессов;
- освобождает ресурсы, то есть снимает блокировки;
- завершает транзакции, удаляет процессы из списка активных;
- восстанавливает состояние (rollback – откат) базы данных после ненормальных ситуаций завершения пользовательских процессов.

2. dbwr – фоновый процесс записи блоков данных в базу из списка модифицированных блоков в SGA. dbwr "пробуждается" к работе, если:

- длина списка модифицированных блоков превысила пороговое значение;
- в списке свободных буферов не хватает памяти для чтения новых блоков;
- истек очередной 3-х секундный интервал времени;
- фоновый процесс записи в журнал lgwr сигнализирует о начале формирования очередной контрольной точки.

Процессы экземпляра Oracle

3. lgwr – фоновый процесс записи в журнал регистрации изменений в базе данных. Регистрация транзакций осуществляется следующим образом:

- по мере выполнения транзакции создаются небольшие записи, называемые элементами повтора (redo entries), в которых содержится информация, достаточная для воссоздания изменений, вносимых транзакцией.
- элементы повтора транзакции временно сохраняются в буфере журнала повтора.
- когда запрашивается завершение транзакции, процесс lgwr считывает необходимые элементы повтора из буфера журнала транзакций и записывает их в журнал транзакций базы данных. Транзакция считается завершенной, когда процесс lgwr запишет элемент повтора транзакции в журнал транзакций и сделает запись о ее завершении в журнале транзакций.

Данные из SGA-буфера переносятся на диск в следующих случаях:

- выполнена операция COMMIT фиксации изменений очередной транзакции;
- истек очередной 3-х секундный интервал времени;
- буфер журнала в SGA заполнен на одну треть своей емкости;
- процесс dbwr записал на диск очередную порцию модифицированных буферов.

Процессы экземпляра Oracle

4. smon – обязательный процесс системного мониторинга выполняет:

- автоматическое восстановление (roll forward – накат вперед) базы данных, если ее предыдущий запуск завершился ненормально или аварийно;
- освобождение временных сегментов от ненужных данных;
- объединение смежных свободных экстентов табличных пространств в непрерывные участки.

5. arch – необязательный фоновый процесс архивирования файлов оперативных журналов регистрации изменений в базе данных. Место копирования (диск, лента,...) определяется параметром log_archive_dest в файле init<$ORACLE_SID>.ora.

Если процесс arch не успел заархивировать очередной журнальный файл (например, переполнена файловая система и не хватает места, чтобы разместить файл-архив), а требуется на него переключение, Oracle приостанавливает функционирование, выполняя только транзакции, не связанные с ведением журнала.

Процессы экземпляра Oracle

6. **ckpt** – необязательный вспомогательный процесс записи контрольной точки в оперативный журнал фиксации изменений. Обычно контрольные точки записывает lgwr. Процесс ckpt (checkpoint_process = true в файле init<$ORACLE_SID>.ora) лишь освобождает lgwr от этой функции.
7. **reco** – (полу) обязательный процесс, ответственный за связи с удаленными базами данных. Процесс reco можно не запускать (в init<$ORACLE_SID>.ora параметр disributed_transaction = 0), но тогда экземпляр не сможет использовать ни одной "связи между базами данных".
8. **snpX** – от одного до десяти процессов автоматического обновления снапшотов локальной базы. Количество задается в init<$ORACLE_SID>.ora параметром snapshot_refresh_processes, а параметр snapshot_refresh_interval определяет регулярность их включения. Процессы snpX можно отнести к серверным, поскольку они, связываясь с другими (в частности с той же самой) базами данных, работают с пользовательской информацией в базе данных.

Процессы экземпляра Oracle

9. dbXX – дополнительные процессы записи в базу данных. Если узким местом производительности базы является ввод/вывод, а база физически размещается на нескольких дисках, рекомендуется запустить несколько дополнительных процессов записи (в среднем, по одному на каждый отдельный диск). Количество дополнительных dbXX определяется параметром db_writers.

10. dXXX – процессы диспетчеры в варианте архитектуры MTS с разделяемыми серверами. Количество функционирующих в данный момент диспетчеров зависит от напряженности работы Oracle, но не превышает заданного параметром mts_max_dispatchers числа. Каждый диспетчер обслуживает только конкретный сетевой или внутренний протокол. Например:

mts_dispatchers="tcp,1"

mts_dispatchers="ipc,1"

Процессы экземпляра Oracle

11. **sXXX** – процессы серверы в варианте архитектуры MTS с разделяемыми серверами. Количество функционирующих в данный момент серверов зависит от напряженности работы Oracle, но не превышает заданного параметром mts_max_servers числа. Стартуя, Oracle запускает несколько (mts_servers) сервер-процессов, а затем то мере возрастания или снижения нагрузки запускает или завершает дополнительные процессы.
12. **oracle<$ORACLE_SID>** – выделенный процесс сервера, индивидуально обслуживающий какой-то пользовательский (в частном случае и процесс snp) процесс, вполне возможно функционирующий на другой машине.
13. **locX** – от одного до десяти процессов блокировок, обеспечивающих взаимное управление ресурсами в среде параллельного сервера.

Архитектуры серверов Oracle

Однопользовательский вариант (пример среды – MS DOS) характеризуется тем, что:

- происходит объединение пользовательского процесса, процесса сервера и фоновых процессов в рамки одной задачи операционной системы;
- возможен запуск только одной базы данных и одного экземпляра Oracle;
- в распределенной базе данных не может функционировать в качестве сервера.

Многопользовательский вариант (пример среды – UNIX) характеризуется тем, что:

- происходит разделение пользовательских, серверных и фоновых процессов на отдельные задачи операционной системы;
- есть возможность запуска нескольких баз данных и экземпляров Oracle;
- возможно функционирование в качестве сервера в распределенной БД.

Архитектуры серверов Oracle

Однозадачный вариант (пример среды – NetWare) характеризуется тем, что:

- пользовательский процесс и процесс сервера образуют единую задачу операционной системы, называемую задачей пользователя;
- в каждый момент времени на сервере может выполняться только одна задача пользователя;
- возможен доступ многих пользователей через Net8 (SQL*Net) к базе данных.

Двухзадачный вариант (пример среды – UNIX) характеризуется тем, что:

- пользовательский процесс и процесс обслуживающего его сервера представляют собой полностью самостоятельные процессы операционной системы вплоть до того, что могут функционировать на разных машинах и платформах (архитектура "клиент-сервер");
- в каждый момент времени на сервере может функционировать несколько (много) пользовательских и серверных процессов;
- возможен доступ многих пользователей через Net8 (SQL*Net) к локальным базам данных и локальных пользователей к удаленным базам данных.

Архитектуры серверов Oracle

Однонитевая архитектура, или вариант с выделенными (Dedicated) серверами:

- жесткое закрепление за каждым пользовательским процессом процесса сервера, который выполняет его и только его запросы к базе данных.

Параллельный сервер (среда – кластерные системы, например, RM-1000):

- на каждом процессоре кластера функционирует свой экземпляр Oracle, включающий отдельную область SGA и набор системных процессов;
- каждый экземпляр ведет свои собственные журналы регистрации изменений;
- база данных и управляющие файлы являются общими для всех экземпляров;
- к каждому экземпляру возможно подключение многих пользователей;
- каждый экземпляр адресуем отдельно, и может самостоятельно работать как часть распределенной системы.

Архитектуры серверов Oracle

Многонитевая архитектура (MTS – Multi-Tread Server), вариант с разделяемыми серверами характеризуется:

- наличием процессов-диспетчеров, принимающих запросы от пользовательских процессов и возвращающих им результаты выполненных сервер-процессами запросов;
- наличием в SGA:
  - одной входной очереди для всех сервер-процессов, в которую диспетчеры помещают заявки на обслуживание от пользователей;
  - нескольких выходных очередей, закрепленных по одной за каждым процессом диспетчером, куда серверы помещают и откуда диспетчеры передают пользователям результаты выполнения запросов к базе данных;
- переносом в SGA экземпляра Oracle частных SQL-областей, ранее размещавшихся в PGA процессов серверов;
- динамическим изменением в зависимости от текущей нагрузки системы количества функционирующих диспетчеров и сервер-процессов;
- ни диспетчеры, ни серверы не закрепляются за какими-либо процессами пользователей: запросы обслуживаются по мере поступления;
- возможностью одновременного функционирования выделенных и разделяемых серверов.

Архитектура MTS