Тест №1 Рубежный

Выполнил:

Кислюк Игорь Витальевич К4120

1. Определение реляционной базы данных.

Реляционная база данных – это совокупность взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа.

1. Среды разработки баз данных: MySQL Workbench, Oracle Data Modeler, MyDB studio, PHPMyAdmin
2. Основные правила создания SQL инструкций.

SQL – запросы создаются программистами из последовательности SQL – **инструкций**.

**Инструкции SQL** состоят из предложений (SELECT, FROM, WHERE и т.д.)

Пример SQL:  
SELECT Рабочий.КодРабочего FROM ТаблицаРабочих;

1. Арифметические операторы SQL.

Сложение, вычитание, умножение, деление.

1. Псевдонимы столбцов.

Для имен столбцов можно создать псевдонимы, чтобы облегчить работу с ними, подсчеты и суммирования значений. Псевдоним должен соответствовать всем соглашениям об именах для SQL.

Псевдоним столбца:

1. Переименовывает заголовок столбца;
2. Полезен с вычислениями;
3. Следует сразу за именем столбца;
4. Требует двойных кавычек.

6. Оператор конкатенации строк.

В Oracle SQL предусмотрен оператор конкатенации – две вертикальные черты и предусмотрена встроенная функция CONCAT, которая также производит конкатенацию и работает на любых платформах.

Пример SQL:

SELECT name || ' ' || age   FROM students;

7. Приоритет операторов:

1. Побитовое НЕ;
2. \* (умножение), / (деление), % (остаток от деления);
3. + (Положительное), - (отрицательное), + (сложение), (+ объединение), - (вычитание), & (побитовое и), ^ (побитовое исключающее или), | (Побитовый оператор или);
4. =, >, <, > =, < = <>,! =,! >,! < (операторы сравнения);
5. NOT;
6. ALL, ANY, BETWEEN, IN, LIKE, OR, SOME;
7. = (присваивание).

8. Классификация одностроковых функций и их назначение.

Строковые литералы в Оracle – это последовательность из нуля, одного или более символов, заключенных в одинарные кавычки.

Примерами являются:

CONCAT(strl, str2), REPLACE(str, search\_str, [,replace\_str]).

9. Символьные функции.

Функции LOWER(str), UPPER(str), INITCAP(str), LTRIM(str [,set]) и RTRIM(str [,set]), CHR(N).

10. Числовые функции.

В Oracle предусмотрен ряд встроенных функций для работы с числами:

1. Функция ABS(n) - возвращает абсолютное значение числа.
2. Функция CEIL (n) - возвращает наименьшее целое, большее или равное пе­реданному в качестве параметра числу n.
3. Функция FLOOR(n) возвращает наибольшее целое, меньшее или равное переданному в качестве параметра числу n.
4. Функция TRUNC(n [,m]) возвращает число n, усеченное до m знаков после десятичной точки. Параметр m может не указываться — в этом случае n усекается до целого.
5. Функция ROUND(n [,m]) возвращает число n, округленное до m знаков после десятичной точки по правилам математического округления. Параметр m может не указываться — в этом случае n округляется до целого.
6. Функция SIGN(n) определяет знак числа. Если n положительное, то функ­ция возвращает 1. Если отрицательное — возвращается -1. Если равно нулю, то возвращается 0.
7. Функция MOD(n, m) возвращает от деления n на m.
8. Функция POWER(n, m) возводит число n в степень m. Степень может быть дробной и отрицательной, что существенно расширяет возможности дан­ной функции.

11. Функции управления датами.

Для хранения даты и времени в Oracle предусмотрен специальный тип DATE. С физической точки зрения это дробное число, целая часть которого хранит количество дней с некоторой базовой даты, а дробная — время. Это позволяет совершать над датами арифметические операции — сложение и вычитание.

Функции SYSDATE, ADD\_MONTHS(d, x), LAST \_DAY(d), MONTHS\_BETWEEN(dl, d2), TRUN С (d[,mask ]), ROUND(d[,mask]).

12. Функции преобразования типов данных.

CAST конвертирует (преобразует) один тип данных в другой.

13. Условный оператор.

1. [IF-THEN-ELSE](http://oracleplsql.ru/if-then-else.html)
2. [CASE](http://oracleplsql.ru/case.html)
3. [GOTO](http://oracleplsql.ru/goto.html)

14. Изменение данных в таблице. Примеры.

Оператор **UPDATE** используется для обновления существующих записей в таблице в базе данных Oracle. Пример:

UPDATE workers

SET worker\_phone = '123456'

WHERE worker\_name = 'Worker 1';

15. Различие между использованием DDL и DML инструкций.

Data Definition Language (DDL) (язык описания данных) — это семейство компьютерных языков, используемых в компьютерных программах для описания структуры баз данных.

Data Manipulation Language (DML) (язык управления (манипулирования) данными) — это семейство компьютерных языков, используемых в компьютерных программах или пользователями баз данных для получения, вставки, удаления или изменения данных в базах данных.

Триггеры DML создаются для инструкций INSERT, UPDATE и DELETE, чтобы обеспечить соблюдение бизнес-правил и целостность данных таблиц или представлений при их изменении.

Триггеры DDL применяются к инструкциям CREATE, ALTER, DROP и некоторым другим инструкциям DDL, а также к хранимым процедурам, которые выполняют схожие с DDL операции. Используют их для выполнения административных задач и гарантии соблюдения бизнес-правил, влияющих на базы данных. Действие этих триггеров распространяется на все команды одного типа во всей базе данных или на всем сервере.

16. Типы данных БД Oracle 11g. Классификация и описание типов данных.

Классификация:

1. Символьные: char(размер), nchar(размер), nvarchar2(размер), varchar2(размер), long, raw
2. Числовые типы данных: number(точность,масштаб), numeric(точность,масштаб), decimal(точность,масштаб), PLS\_INTEGE
3. Дата/время типы данных: date
4. Большие объекты (LOB) типы данных: bfile, blob, clob, nclob
5. Rowid тип данных: rowed
6. Булевы (BOOLEAN) типы данных: BOOLEAN

17. Понятие ограничения целостности данных.

Целостность БД не гарантирует достоверности (истинности) содержащейся в ней информации, но обеспечивает по крайней мере правдоподобность этой информации, отвергая заведомо невозможные значения.

18. Создание групп данных.

GroupBy используется для вычисления определенного группового значения, основанного на столбцах, перечисленных в этом операторе. Проще говоря, выборка разбивается на группы. Группы состоят из записей отличающихся друг от друга значениями в столбцах перечисленных в конструкции group by.

19. Подзапросы. Типы подзапросов.

В Oracle подзапрос — это запрос в запросе. Вы можете создавать подзапросы внутри ваших операторов. Эти подзапросы могут находиться в предложении WHERE, в предложении FROM, или SELECT. Типы:

1. Однострочные подзапросы: Запросы, которые возвращают только одну строку из внутреннего оператора SELECT
2. Многострочные подзапросы: Запросы, которые возвращают больше чем одну строку из внутреннего оператора SELECT

20. Назначение и классификация распределенных систем.

Распределенной считают такую систему, в которой функционирует более одного сервера БД. Это применяется для уменьшения нагрузки на сервер и обеспечения работы территориально удаленных подразделений. Различная сложность создания, модификации, сопровождения, интеграции с другими системами позволяют разделить ИС на классы: малых, средних и крупных распределенных систем. Часто при определении распределенной системы во главу угла ставят разделение ее функций между несколькими компьютерами. При таком подходе распределенной является любая вычислительная система, где обработка данных разделена между двумя и более компьютерами.

21. Архитектура и основные компоненты распределенных систем.

Архитектура информационной системы - концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов информационной системы.

Компоненты информационной системы по выполняемым функциям можно разделить на три слоя: слой представления, слой бизнес-логики и слой доступа к данным. Слой представления - все, что связано с взаимодействием с пользователем: нажатие кнопок, движение мыши, отрисовка изображения, вывод результатов поиска и т.д. Бизнес логика - правила, алгоритмы реакции приложения на действия пользователя или на внутренние события, правила обработки данных. Слой доступа к данным - хранение, выборка, модификация и удаление данных, связанных с решаемой приложением прикладной задачей.

Характеристики распределенных систем:

1. От пользователей скрыты различия между компьютерами и способы связи между ними. То же самое относится и к внешней организации распределенных систем.
2. Пользователи и приложения единообразно работают в распределенных системах, независимо от того, где и когда происходит их взаимодействие.

Распределенные системы должны также относительно легко поддаваться расширению, или масштабированию. Эта характеристика является прямым следствием наличия независимых компьютеров, но в то же время не указывает, каким образом эти компьютеры на самом деле объединяются в единую систему. Распределенные системы обычно существуют постоянно, однако некоторые их части могут временно выходить из строя. Пользователи и приложения не должны уведомляться о том, что части системы заменены или починены, или, что добавлены новые для поддержки дополнительных пользователей.

Архитектура: Клиент-серверная архитектура, Переходная к трехслойной архитектуре, Трехуровневая клиент-серверная архитектура и т.д.

22. Определение функциональных и нефункциональных требований

Анализ требований — часть процесса разработки программного обеспечения, включающая в себя сбор требований к программному обеспечению (ПО), их систематизацию, выявление взаимосвязей, а также документирование. Функциональные требования объясняют, что должно быть сделано. Они идентифицируют задачи или действия, которые должны быть выполнены. Функциональные требования определяют действия, которые система должна быть способной выполнить, связь входа/выхода в поведении системы. Нефункциональные требования – требования, определяющие свойства, которые система должна демонстрировать, или ограничения, которые она должна соблюдать, не относящиеся к поведению системы. Например, производительность, удобство сопровождения, расширяемость, надежность, факторы эксплуатации.

23. Средства анализа бизнес-процессов.

Бизнес-модель – это формализованное описание бизнес-процессов. Средства: ARIS, BPwin, Rational Rose.

24. Определение сущностей и необходимых атрибутов

Сущность – объект, выделяемый пользователь в предметной области.

Атрибут – это поименованная характеристика, значимая с точки зрения пользователя.

25. Определение атрибутов, которые будут являться уникальными идентификаторами для каждой сущности.

Ключ – минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся.