**Тесты для нового сотрудника отдела автоматизации**

**Знание SQL**

Описание базы: (*См. Файл к тестам.xls, Лист 4 «Тестовая база данных»)*

Даны 3 таблицы:

**Customer** (Клиенты)

**OrderHeader** (Заказы)

**OrderLine** (СтрокиЗаказа)

Описание таблицы **Customer**:

*CustomerNo* (Клиент Но.) (Тип поля String)

*CustomerName* (Имя Клиента) (Тип поля String)

*Address* (Адрес Клиента) (Тип поля String)

*SalesPerson* (Менеджер) (Тип поля String)

Описание таблицы **OrderHeader**

*OrderNo* (Заказ Но.) (Тип поле Integer)

*CustomerNo* (Клиент Но.) (Тип поле String)

*OrderDate* (Дата заказа) (Тип поле Date)

*DeliveryTerms* (Условия доставки) (Тип поле String)

*OrderSum* (Сумма заказ) (Тип поле Real)

Описание таблицы **OrderLine**

*OrderNo* (Заказ Но.) (Тип поле Integer)

*LineNo* (Строка Но.) (Тип поле Integer)

*ItemNo* (Товар Но.) (Тип поля String)

*ItemText* (Описание Товара) (Тип поля String)

*NumberOf* (Кол-во ) (Тип поле Real)

*UnitPrice* (Цена единицы) (Тип поле Real)

create database if not exists SQLtask;

use SQLtask;

create table if not exists Customer

(

CustomerNo varchar(20) primary key,

CustomerName varchar(50) not null,

Address varchar(50) not null,

SalesPerson varchar(50) not null

);

create table if not exists OrderHeader

(

OrderNo int not null,

CustomerNo varchar(20) not null,

OrderDate date not null,

DeliveryTerms varchar(20) not null,

OrderSum real not null,

primary key (OrderNo),

constraint OrderHeader\_Customer\_fk

foreign key (CustomerNo) references Customer(CustomerNo) on delete cascade

);

create table if not exists OrderLine

(

OrderNo int not null,

LineNo int not null,

ItemNo varchar(20) not null,

ItemText varchar(20) not null,

NumberOf real not null,

UnitPrice real not null,

primary key (OrderNo,LineNo),

constraint OrderLine\_OrderHeader\_fk

foreign key (OrderNo) references OrderHeader(OrderNo) on delete cascade

);

insert Customer(CustomerNo, CustomerName, Address, SalesPerson)

values ('000001', 'ООО ВолгаЭнергоСнаб', 'ул. Коммунальная, 33', 'Петров'),

('000002', 'ЗАО ДАНОН ВОЛГА', 'ул. Коммунальная, 26А', 'Иванов'),

('000003', 'ООО СТИМ-ЛИЗИНГ', 'ул. Индустриальная, 1В', 'Петров'),

('000004', 'ООО Химзавод', 'ул.Новозаводская 2А', 'Иванов');

insert OrderHeader(OrderNo,CustomerNo,OrderDate,DeliveryTerms,OrderSum)

values (10001,"000002",'2009-01-03',"Самовывоз",1500),

(10002,"000003",'2009-02-03',"Доставка",5700),

(10003,"000001",'2009-03-03',"Доставка",1000),

(10004,"000002",'2009-10-03',"Самовывоз",2400);

insert OrderLine(OrderNo,LineNo,ItemNo,ItemText,NumberOf,UnitPrice)

values (10001,1,"Item1","Лампа",100,5),

(10001,2,"Item2","Вилка",6,100),

(10001,3,"Item2","Вилка",1,100),

(10001,4,"Item3","Выключатель",5,60),

(10002,1,"Item3","Выключатель",12,60),

(10002,2,"Item4","Трансформатор",1,5000),

(10003,1,"Item1","Лампа",200,5),

(10004,1,"Item5","Счетчик",2,1200);

**Задания:**

1. Из скольких Таблиц состоит данная База Данных?

База данных состоит из 3 таблиц: **Customer** (Клиенты), **OrderHeader** (Заказы), **OrderLine** (СтрокиЗаказа), **НО** их должно быть 4, поля *ItemText* и *UnitPrice* должны быть в отдельной таблице **Item** (Товар), а не в **OrderLine** (см. 3-ю нормальную форму), также в таблице **Item** должна быть копия поля *ItemNo* в качестве первичного ключа.

1. Из скольких полей состоит таблица **OrderLine**?

Таблица **OrderLine** состоит из 6 полей, где поля *ItemText* и *UnitPrice неуместны(см. 1-й вопрос)*

1. Сколько записей насчитывается в таблице **Customer**?

В таблице **Customer** 4 записи.

1. Какие первичные ключи, по-вашему, должны иметь таблицы **Customer**, **OrderHeader**, **OrderLine**?

Первичные ключи: для **Customer -** *CustomerNo*, для **OrderHeader -** *OrderNo*, для **OrderLine -** *OrderNo*+*LineNo*.

1. Расскажите своими словами, как вы понимаете 1-нормальную форму и вторую нормальную форму таблицы?

**1-я нормальная форма** - это таблица **без повторяющихся столбцов или наборов значений**. То есть допустим таблица людей содержит поле “электронная почта”, и туда клепают штук по 5 электронных почт в одну ячейку таблицы - это нарушение 1-й нормальной формы, так как столбец “электронная почта” содержит набор значений, и этот столб можно вынести в отдельную таблицу. Или допустим таблица людей содержит в том числе поля “20 сентября”,”калории1”,”21 сентября”,”калории2” - это повторяющиеся столбцы, тоже нарушение 1-й нормальной формы, тут тоже стоит вынести отдельную таблицу с полями “HumanID”(идентификатор человека),”Date”(дата),”Calories”(калории). Иногда имеет смысл допустить избыточность информации и вместо вынесения столбцов в отдельную таблицу просто добавить строки чтобы исключить повторение столбцов.

**2-я нормальная форма** - это когда все столбцы таблицы зависят от всех столбцов ключа, а не только от части ключа. Например, будь в таблице **OrderLine** столбец OrderSum, то он бы нарушал 2-ю нормальную форму, так как зависел бы только от номера заказа, но не от строки заказа. А первичный ключ этой таблицы состоит и из номера и из строки заказа.

1. В каком отношении находятся таблицы **OrderHeader** и **OrderLine**:
   1. Один к одному (1 - 1)
   2. Один ко многим (1 - n)
   3. Многое ко многим (n - n)

Таблицы **OrderHeader** и **OrderLine** находятся в отношении Один ко многим (1 - n), ведь заказ (OrderHeader) может включать в себя множество SKU, то есть единиц товаров (OrderLine).

1. Напишите sql-запрос, который вернет:
   1. Все данные из таблицы Клиенты;

SELECT \* FROM Customer;

* 1. Все данные из таблицы Клиенты, упорядоченные по полю *SalesPerson* (Менеджер);

SELECT \* FROM Customer

ORDER BY SalesPerson;

* 1. Значения из столбца *DeliveryTerms* (Условия доставки) из таблицы Заказы?

SELECT DeliveryTerms FROM OrderHeader;

* 1. УНИКАЛЬНЫЕ Значения из столбца *DeliveryTerms* (Условия доставки) из таблицы Заказы?

SELECT DISTINCT DeliveryTerms FROM OrderHeader;

* 1. Все записи из таблицы Заказы по клиенту «000002»;

SELECT \* FROM OrderHeader

WHERE CustomerNo = 000002;

* 1. Все записи из таблицы Заказы по клиенту «000002» и на сумму >2000;

SELECT \* FROM OrderHeader

WHERE CustomerNo = 000002 and OrderSum>2000;

* 1. Номера клиентов, в названии которых СОДЕРЖИТСЯ слово «Волга» (с учетом того, что поле *CustomerName* является Case-Sensitive);

SELECT CustomerNo FROM Customer

WHERE LOCATE('волга', LOWER(CustomerName)) != 0;

* 1. Все номера заказов с менеджерами по продаже, созданные до 10 марта 2009 года;

SELECT O.OrderNo, C.SalesPerson

FROM OrderHeader AS O, Customer AS C

WHERE O.OrderDate < '2009-03-10'

AND C.CustomerNo = O.CustomerNo;

* 1. Все столбцы из таблицы **OrderLine** и еще один новый столбец, которые будет произведением Кол-ва на Цену. Столбец нужно назвать *LineSum* (сумма по строке);

SELECT \*, NumberOf \* UnitPrice FROM OrderLine AS LineSum;

* 1. Кол-во записей в таблице **OrderLine**;

SELECT COUNT(\*) FROM OrderLine;

* 1. Кол-во записей в таблице **OrderLine**, в которых товар равен «Item1»;

SELECT COUNT(\*) FROM OrderLine

WHERE ItemNo = 'item1';

* 1. Номера клиентов и кол-во строк размещенных заказов(OrderLine, а не кол-во самих заказов). Возвращать только номера тех клиентов, которые разместили хотя бы один заказ;

SELECT OH.CustomerNO, COUNT(OL.LineNo) AS CountLines

FROM OrderHeader AS OH, OrderLine AS OL

WHERE OH.OrderNo = OL.OrderNo

GROUP BY OH.CustomerNO;

* 1. Номер клиента, Название клиента, общую сумму всех заказов клиента. Даже если клиент не разместил ни одного заказа, sql-запрос должен вернуть 0.

SELECT C.CustomerNo, C.CustomerName, IFNULL(SUM(OH.OrderSum),0) AS TotalSum

FROM Customer AS C

LEFT OUTER JOIN OrderHeader AS OH

ON C.CustomerNo = OH.CustomerNo

GROUP BY C.CustomerNo;

* 1. Список номеров клиентов (CustomerNo) (уникальных значений), которые заказывали товар Item3. (применить конструкцию EXISTS);

SELECT DISTINCT CustomerNo FROM OrderHeader

WHERE EXISTS

(SELECT \* FROM OrderLine

WHERE OrderLine.OrderNo = OrderHeader.OrderNo

AND OrderLine.ItemNo = 'Item3');

* 1. Список номеров товаров (ItemNo) (уникальных значений), сумма заказов по которым всего составила >=1000. (применить Group By and Having)

SELECT DISTINCT OL.ItemNo, SUM(OH.OrderSum)

FROM OrderLine AS OL, OrderHeader AS OH

WHERE OH.OrderNo = OL.OrderNo

GROUP BY OL.ItemNo

HAVING SUM(OH.OrderSum)>=1000;

* 1. Сколько записей вернет sql:
     1. select OrderNo from OrderHeader

where exists(select 1 from OrderLine

where OrderLine. OrderNo=OrderHeader. OrderNo

and OrderLine.ItemText=’Вилка’)

Вернёт 1 запись, где вилка есть в заказе. Такой запрос применим на практике.

* + 1. select OrderNo from OrderHeader

where exists(select 1 from OrderLine

where OrderLine.ItemText=’Вилка’)

Вернёт все 4 заказа, так как условие “Есть позиция вилка в заказах” выполнено для одного из заказов. Проблема в отсутствии связи между таблицами, поэтому запрос теряет свой смысл.

Какой из sql не имеет смысла, хотя и корректен, а какой применим на практике?

1. Напишите sql, который :
   1. Удалит все записи из таблицы **Customer**

truncate table Customer;

* 1. Обновит поле *ItemText* на «Трансформатор понижающий» в таблице **OrderLine** по заказу 10002 во второй строке заказа;

UPDATE OrderLine

SET ItemText = "Трансформатор понижающий"

WHERE OrderNo = 10002 AND LineNo = 2;

* 1. Удалить таблицу **OrderHeader** из базы данных;

DROP TABLE OrderHeader;

**Знания по программированию**

1. Кусок программы:

………

i:=1

str1:=”Integer Variable”

str2:=””

varLength:= Length(str1)

Do While i<= varLength

str2:= str2+SubString(str1, Length(str1),1)

str1:= SubString(str1, 1, Length(str1)-1)

i:=i+1

End While

Print (str2)

Print (str1)

Вопросы:

* 1. Что, по-вашему, делает программа?

Программа циклично стирает символы с конца строки str1, пока не сотрёт её полностью.

* 1. Какое значение будет иметь str1 после выполнения программы?

Пустая строка.

* 1. Какое значение будет иметь str2 после выполнения программы?

Пустая строка.

* 1. Сколько раз выполнится цикл?

16

1. Дан двумерный массив (матрица) размером N x N. Напишите абстрактную программу (можно схематично), которая:
   1. Сначала заполняет элементы массива числами равными сумме их индексов (a(i,j)=i+j)

for i in range(N):

for j in range(N):

array[i][j]=i+j;

* 1. Считает сумму элементов диагонали матрицы.

sum = 0;

for i in range(N):

sum = sum + array[i][i];

1. Дана матрица, состоящая из N-строк и M-столбцов. Все элементы матрицы целые числа. Напишите абстрактную программу (можно схематично), которая считает, сколько среди элементов матрицы элементов, которые без остатка делятся на 3. (Функция «Остаток от целочисленного деления» X MOD Y)

devideCount=0;

for i in range(N):

for j in range(M):

if (array[i][j] MOD 3 == 0) devideCount = devideCount + 1;

1. Дана строка «This Is A String». Напишите абстрактную программу (можно схематично), которая считает количество пробелов в строке.

myString = ‘This Is A String’;

spaceCount = 0;

for i in range(len(myString)):

if (myString[i] == ‘ ‘) spaceCount = spaceCount + 1;

1. Дан массив, состоящий из N элементов. Напишите абстрактную программу (можно схематично), которая считает сумму элементов с четными индексами.

sum = 0;

for i in range(math.floor(N/2)-1):

sum = sum + array[2\*i];

**Знания по техническому и бизнес английскому**

1. Переведите с английского на русский следующие предложения:

The accounts receivable module supports work routines relating to customer. These include maintaining customer information cards and customer discount agreements, reconciling open entries.;

In this window you can create, change and delete information.

Is there a limit to the number of arguments that you can have in one DECODE statement? I'm getting an error, "ORA-00939: too many arguments for function". Yes, the maximum number of components that you can have in a DECODE function is 255. This includes the expression, search, and result arguments.

The necessary data cannot be extracted using this sql-query as the WHERE part of the query is incorrect.

Модуль расчёта с клиентами поддерживает рабочие процедуры, связанные с клиентом. К ним относятся хранение информационных карт клиентов, соглашений о скидках, сверка открытых записей;

В этом окне вы можете создавать, изменять и удалять информацию.

Существует ли ограничение на количество аргументов, которые можно иметь в одном операторе DECODE? Я получаю сообщение об ошибке "ORA-00939: слишком много аргументов для функции". Да, максимальное количество аргументов, которое может быть в функции DECODE, равно 255. Это включает в себя аргументы выражения, поиска и результата.

Необходимые данные не могут быть извлечены с помощью этого sql-запроса, так как условная(WHERE) часть запроса некорректна.

1. Переведите с русского на английский следующие предложения:
   1. Данные заказы должны быть отгружены сегодня и доставлены до клиента к концу этой недели.

These orders should be loaded today and delivered to the customer till the end of this week.

* 1. Данная таблица имеет 5 столбцов. Все столбцы имеют целочисленный тип.

This table has 5 columns. All columns are of an integer type.

* 1. Записи из данной таблицы должны быть перенесены в другую.

Records from this table must be transferred to another table.

* 1. Данная программа выдает ошибку, потому что одна из переменных не инициализирована.

This program throws an error because one of the variables is not initialized.