select to\_char(sysdate+3, 'DAY')

from dual

;

/\*1 Написать запрос, выводящий всю информацию о департаментах. Упорядочить по коду департамента.\*/

select d.department\_id,

d.department\_name,

d.manager\_id,

d.location\_id

from departments d

order by d.department\_id

;

/\*2 Написать запрос, выбирающий ID, имя+фамилию (в виде одного столбца через пробел)

и адрес электронной почты всех клиентов. (Использовать конкатенацию строк и переименование

столбца с именем и фамилией на «NAME»). Упорядочить по коду клиента.\*/

select

c.customer\_id,

c.cust\_first\_name||' '|| c.cust\_last\_name as NAME,

c.cust\_email

from

customers c

order by

c.customer\_id

;

/\*3 Написать запрос, выводящий сотрудников, зарплата которых за год

лежит в диапазоне от 100 до 200 тыс. дол., упорядочив их по занимаемой

должности, зарплате (от большей к меньшей) и фамилии. Выбранные данные должны включать фамилию, имя, должность (код должности),

email, телефон, зарплату за месяц за вычетом налогов. Будем считать, что у нас прогрессивная

шкала налогообложения: с зарплаты за год от 100 до 150 тыс. дол. налог составляет 30%, выше – 35%.

Результат округлить до целого дол. Обязательно использовать between и case.\*/

select e.last\_name,

e.first\_name,

e.job\_id,

e.email,

e.phone\_number,

case

when e.salary\*12 between 100000 and 150000 then

salary\*0.7

when e.salary\*12 between 150000 and 200000 then

salary\*0.65

END salary

from employees e

where e.salary\*12 between 100000 and 200000

order by e.job\_id,salary DESC, e.last\_name

;

/\*4 Выбрать страны с идентификаторами DE, IT или RU. Переименовать столбцы на «Код страны», «Название страны».

Упорядочить по названию страны.\*/

select с.country\_id as "Код",

с.country\_name as "Название страны"

from countries с

where c.country\_id = 'DE' or

c.country\_id = 'IT' or

c.country\_id = 'RU'

order by c.country\_name

;

/\*5 Выбрать имя+фамилия сотрудников, у которых в фамилии вторая буква «a» (латинская),

а в имени присутствует буква «d» (не важно, в каком регистре). Упорядочить по имени.

Использовать оператор like и функции приведения к нужному регистру.\*/

select e.first\_name||' '||e.last\_name as emp\_name

from employees e

where e.last\_name like '\_a%' and

lower (e.first\_name) like '%d%'

order by e.first\_name

;

/\*6 Выбрать сотрудников у которых фамилия или имя короче 5 символов.

Упорядочить записи по суммарной длине фамилии и имени, затем по длине фамилии, затем просто по фамилии,

затем просто по имени.\*/

select \*

from employees e

where length(e.first\_name) < 5 or

length(e.last\_name) < 5

order by length(e.first\_name || e.last\_name),

length(e.last\_name),

e.last\_name,

e.last\_name

;

/\*7 Выбрать должности в порядке их «выгодности» (средней зарплаты, за среднюю взять среднее-арифметическое минимальной

и максимальной зарплат). Более «выгодные» должности должны быть первыми, в случае одинаковой зарплаты упорядочить по

коду должности. Вывести столбцы код должности, название должности, средняя зарплата после налогов, округленная до сотен.

Считаем шкалу налогообложения плоской – 18%.\*/

select j.job\_id,

j.job\_title, round((((j.min\_salary+j.max\_salary)/2)\*0.82),-2) as avg\_salary

from jobs j

order by avg\_salary desc,

j.job\_id

;

/\*8 Будем считать, что все клиенты делятся на категории A, B, C. Категория A – клиенты с

кредитным лимитом >= 3500, B >= 1000, C – все остальные. Вывести всех клиентов, упорядочив их по категории в

обратном порядке (сначала клиенты категории A), затем по фамилии. Вывести столбцы фамилия, имя, категория, комментарий.

В комментарии для клиентов категории A должно быть строка «Внимание, VIP-клиенты», для остальных клиентов комментарий должен

остаться пустым (NULL).\*/

select c.cust\_last\_name,

c.cust\_first\_name,

case

when c.credit\_limit >= 3500 then

'A'

when c.credit\_limit >= 1000 then

'B'

else

'C'

END categories,

case

when c.credit\_limit >= 3500 then

'Внимание, VIP-клиент'

else

null

END comments

from customers c

order by categories,

c.cust\_last\_name

;

/\*9 Вывести месяцы (их название на русском), в которые были заказы в 1998 году. Месяцы не должны повторяться и должны

быть упорядочены. Использовать группировку по функции extract от даты для исключения дублирования месяцев и decode для

выбора названия месяца по его номеру. Подзапросы не использовать.\*/

select decode(extract(month from order\_date),

1, 'январь',

2, 'февраль',

3, 'март',

4, 'апрель',

5, 'май',

6, 'июнь',

7, 'июль',

8, 'август',

9, 'сентябрь',

10, 'октябрь',

11, 'ноябрь',

12, 'декабрь')

month

from orders o

where extract(year from order\_date) = 1998

group by extract(month from order\_date)

;

/\*10 Написать предыдущий запрос, используя для получения названия месяца функцию to\_char (указать для функции

nls\_date\_language 3-м параметром). Вместо группировки использовать distinct, подзапросы не использовать.\*/

select distinct to\_char(order\_date,'month') as month

from orders o

where to\_char(order\_date, 'yy') = 98

;

/\*\*11 Написать запрос, выводящий все даты текущего месяца. Текущий месяц должен браться из sysdate.

Второй столбец должен содержать комментарий в виде строки «Выходной» для суббот и воскресений.

Для определения дня недели воспользоваться функций to\_char. Для выбора чисел от 1 до 31 можно воспользоваться

псевдостолбцом rownum, выбирая данные из любой таблицы, где количество строк более 30.

\*\*/ ----

select trunc(sysdate, 'mm')+rownum-1 as dt,to\_char(trunc(sysdate, 'mm')+rownum-1, 'day'),

case

when to\_char(trunc(sysdate, 'mm')+rownum-1, 'day') in('воскресенье', 'суббота ') then 'выходной'

else null

end comments

from orders

where trunc(sysdate, 'mm')+rownum-1<add\_months(trunc(sysdate, 'mm'), 1)

;/\*\*12 Выбрать всех сотрудников (код сотрудника, фамилия+имя через пробел, код должности, зарплата, комиссия - %),

которые получают комиссию от заказов. Воспользоваться конструкцией is not null.Упорядочить сотрудников по проценту

комиссии (от большего к меньшему), затем по коду сотрудника.\*\*/

select e.employee\_id,

e.last\_name||

' '||

e.first\_name as

emp\_name,

e.job\_id,

e.salary,

e.commission\_pct

from employees e

where commission\_pct is not null

order by e.commission\_pct desc,

e.employee\_id

;

/\*\*13 Получить статистику по сумме продаж за 1995-2000 годы в разрезе кварталов (1 квартал – январь-март и т.д.).

В выборке должно быть 6 столбцов – год, сумма продаж за 1-ый, 2-ой, 3-ий и 4-ый квартала, а также общая сумма продаж за год.

Упорядочить по году. Воспользоваться группировкой по году, а также суммированием по выражению с case или decode, которое

будут отделять продажи за нужный квартал.\*\*/

select extract(year from o.order\_date) as year,

sum(case

when extract(month from o.order\_date) between 1 and 3

then o.order\_total

end) as quart1\_sum1,

sum(case

when extract(month from o.order\_date) between 4 and 6

then o.order\_total

end) as quart1\_sum2,

sum(case

when extract(month from o.order\_date) between 7 and 9

then o.order\_total

end) as quart1\_sum3,

sum(case

when extract(month from o.order\_date) between 10 and 12

then o.order\_total

end) as quart1\_sum4,

sum(case

when extract(month from o.order\_date) between 1 and 12

then o.order\_total

end) as year\_sum

from orders o

where extract(year from order\_date) between 1995 and 2000

GROUP BY extract(year from o.order\_date)

order by year

;

/\*\*14 Выбрать из таблицы товаров всю оперативную память. Считать таковой любой товар для которого в

названии указан размер в MB или GB (в любом регистре), название товара не начинается с HD, а также в первых 30

символах описания товара не встречаются слова disk, drive и hard. Вывести столбцы: код товара, название товара, гарантия,

цена (по прайсу – LIST\_PRICE), url в каталоге. В поле гарантия должно быть выведено целое число – количество месяцев гарантии

(учесть, что гарантия может быть год и более). Упорядочить по размеру памяти (от большего к меньшему), затем по цене

(от меньшей к большей). Размер для упорядочивания извлечь из названия товара по шаблону NN MB/GB (не забыть при этом

сконвертировать GB в мегабайты) c помощью regexp\_replace. Like не использовать, вместо него использовать regexp\_like с

явным указанием, что регистр букв следует игнорировать.\*\*/

/\*\*14\*\*/

select p.product\_id,

p.product\_name,

case

when extract(year from p.warranty\_period) = 0 then extract(month from p.warranty\_period)

when extract(year from p.warranty\_period) = 1 then extract(month from p.warranty\_period)+12

end warranty\_period,

p.list\_price,

p.catalog\_url,

p.product\_description

from product\_information p

where regexp\_like(p.product\_name, 'mb|gb', 'i') and

not regexp\_like(p.product\_name, '^hd', 'i') and

not regexp\_like(substr(p.product\_description, 1, 30), 'disk|drive|hard', 'i')

order by list\_price

;

/\*\*15 Вывести целое количество минут, оставшихся до окончания занятий. Время окончания занятия в запросе должно

быть задано в виде строки, например «21:30». Явного указания текущей даты в запросе быть не должно. Можно воспользоваться

комбинацией функций to\_char/to\_date.\*\*/

select round(to\_number(

to\_date(to\_char(to\_date('11-14 21:50:00', 'MM-DD HH24:MI:SS'),'MM-DD HH24:MI:SS'),'MM-DD HH24:MI:SS')-

to\_date(to\_char(sysdate, 'MM-DD HH24:MI:SS'), 'MM-DD HH24:MI:SS'),'9999.999999999999999999999999999999999999999999')

\*24\*60,0) as minutes

from dual

;

/\*1 Выбрать клиентов, у которых были заказы в июле 1999 года. Упорядочить по коду клиента.

Использовать внутреннее соединение (inner join) и distinct.\*/

select distinct \*

from customers c

inner join orders o on

c.customer\_id = o.customer\_id

where date'1999-07-01' <= o.order\_date and o.order\_date < date'1999-08-01'

order by c.customer\_id

;

/\* 2. Выбрать всех клиентов и сумму их заказов за 2000 год, упорядочив их по сумме

заказов (клиенты, у которых вообще не было заказов за 2000 год, вывести в конце), затем по ID заказчика.

Вывести поля: код заказчика, имя заказчика (фамилия + имя через пробел), сумма заказов за 2000 год.

Использовать внешнее соединение (left join) таблицы заказчиков с подзапросом для выбора суммы товаров (по таблице заказов)

по клиентам за 2000 год (подзапрос с группировкой).\*/

select c.customer\_id,

c.cust\_first\_name||' '||c.cust\_last\_name,

total\_orders

from customers c

left join

(select

customer\_id,

sum(order\_total) total\_orders

from orders o

where to\_char(order\_date, 'yyyy') = 2000

GROUP BY customer\_id) o on

c.customer\_id = o.customer\_id

order by total\_orders desc nulls last,

c.customer\_id

;

/\*3 Выбрать сотрудников, которые работают на первой своей должности (нет записей в истории).

Использовать внешнее соединение (какое конкретно?) с таблицей истории, а затем отбор записей из таблицы

сотрудников таких, для которых не «подцепилось» строк из таблицы истории. Упорядочить отобранных сотрудников

по дате приема на работу (в обратном порядке, затем по коду сотрудника (в обычном порядке).\*/

select \*

from employees e

left join job\_history j on

e.employee\_id = j.employee\_id

where j.start\_date is null

order by e.hire\_date desc,

e.employee\_id desc

;

/\*4 Выбрать все склады, упорядочив их по количеству номенклатуры товаров, представленных в них.

Вывести поля: код склада, название склада, количество различных товаров на складе. Упорядочить по количеству

номенклатуры товаров на складе (от большего количества к меньшему), затем по коду склада (в обычном порядке).

Склады, для которых нет информации о товарах на складе, вывести в конце. Подзапросы не использовать.\*/

select w.warehouse\_id,

w.warehouse\_name,

count(p.product\_id) as product\_count

from warehouses w

join inventories i on

w.warehouse\_id = i.warehouse\_id

join product\_information p on

i.product\_id = p.product\_id

group by w.warehouse\_id,

w.warehouse\_name

order by product\_count desc,

w.warehouse\_id

;

/\*5 Выбрать сотрудников, которые работают в США. Упорядочить по коду сотрудника.\*/

select \*

from employees e

join departments d on

e.department\_id = d.department\_id

join locations l on

d.location\_id = l.location\_id

where l.country\_id = 'US'

;

/\*6 Выбрать все товары и их описание на русском языке. Вывести поля: код товара, название товара, цена товара в каталоге

(LIST\_PRICE), описание товара на русском языке. Если описания товара на русском языке нет, в поле описания вывести «Нет описания»,

воспользовавшись функцией nvl или выражением case (в учебной базе данных для всех товаров есть описания на русском языке, однако запрос

должен быть написан в предположении, что описания на русском языке может и не быть; для проверки запроса можно указать код

несуществующего языка и проверить, появилось ли в поле описания соответствующий комментарий). Упорядочить по коду категории товара,

затем по коду товара. \*/

select p.product\_id,

p.product\_name,

p.list\_price,

d.translated\_description as ru\_description

from product\_information p

join product\_descriptions d on

p.product\_id = d.product\_id

where language\_id = 'RU'

order by p.list\_price desc nulls last,

p.product\_id

;

/\*7 Выбрать товары, которые никогда не продавались. Вывести поля: код товара, название товара,

цена товара в каталоге (LIST\_PRICE), название товара на русском языке (запрос должен быть написан в предположении,

что описания товара на русском языке может и не быть). Упорядочить по цене товара в обратном порядке (товары, для которых

не указана цена, вывести в конце), затем по коду товара.\*/

select p.product\_id,

p.product\_name,

p.list\_price,

p.product\_status

from product\_information p

left join order\_items oi on

p.product\_id = oi.product\_id

left join orders o on

oi.order\_id = o.order\_id

where oi.order\_id is null

order by p.list\_price DESC nulls last,

p.product\_id

;

/\*8 Выбрать клиентов, у которых есть заказы на сумму больше, чем

в 2 раза превышающую среднюю цену заказа. Вывести поля: код клиента,

название клиента (фамилия + имя через пробел), количество таких заказов, максимальная

сумма заказа. Упорядочить по количеству таких заказов в обратном порядке, затем по коду клиента.\*/

select c.customer\_id,

c.cust\_last\_name || ' ' || c.cust\_first\_name as cust\_name,

count(o.order\_id) as large\_sum\_orders\_count,

max(o.order\_total) as max\_order\_sum

from customers c

inner join orders o on

o.customer\_id = c.customer\_id

where o.order\_total > 2 \*

(select avg(order\_total)

from orders)

group by c.customer\_id,

c.cust\_last\_name || ' ' || c.cust\_first\_name

order by large\_sum\_orders\_count desc,

c.customer\_id

;

/\*9 Упорядочить клиентов по сумме заказов за 2000 год. Вывести поля: код клиента,

имя клиента (фамилия + имя через пробел), сумма заказов за 2000 год. Упорядочить данные

по сумме заказов за 2000 год в обратном порядке, затем по коду клиента. Клиенты, у которых

не было заказов в 2000, вывести в конце.\*/

select c.customer\_id,

c.cust\_last\_name || ' ' || c.cust\_first\_name as cust\_name,

sum(o.order\_total) as orders\_sum

from customers c

left join orders o on

o.customer\_id = c.customer\_id and

date'2000-01-01' <= o.order\_date and o.order\_date < date'2001-01-01'

group by c.customer\_id,

c.cust\_last\_name || ' ' || c.cust\_first\_name

order by orders\_sum desc nulls last,

c.customer\_id

;

/\*10 Переписать предыдущий запрос так, чтобы не выводить клиентов, у которых вообще не было заказов.\*/

select c.customer\_id,

c.cust\_last\_name || ' ' || c.cust\_first\_name as cust\_name,

sum(o.order\_total) as orders\_sum

from customers c

join orders o on

o.customer\_id = c.customer\_id and

date'2000-01-01' <= o.order\_date and o.order\_date < date'2001-01-01'

group by c.customer\_id,

c.cust\_last\_name || ' ' || c.cust\_first\_name

order by orders\_sum desc nulls last,

c.customer\_id

; c.customer\_id

;

/\*11! Каждому менеджеру по продажам сопоставить последний его заказ.

Менеджера по продажам считаем сотрудников, код должности которых: «SA\_MAN» и «SA\_REP».

Вывести поля: код менеджера, имя менеджера (фамилия + имя через пробел), код клиента, имя клиента

(фамилия + имя через пробел), дата заказа, сумма заказа, количество различных позиций в заказе.

Упорядочить данные по дате заказа в обратном порядке, затем по сумме заказа в обратном порядке, затем по коду сотрудника.

Тех менеджеров, у которых нет заказов, вывести в конце.\*/

select e.employee\_id,

e.last\_name || ' ' || e.first\_name as manager\_name,

c.customer\_id,

c.cust\_last\_name || ' ' || c.cust\_first\_name as cust\_name,

o.order\_date,

o.order\_id

from employees e

inner join (select sales\_rep\_id ,

max(order\_date) as max\_date

from orders

group by sales\_rep\_id

) ord on

e.employee\_id = ord.sales\_rep\_id

inner join orders o on

ord.max\_date = o.order\_date

inner join customers c on

o.customer\_id = c.customer\_id

where job\_id in ('SA\_MAN','SA\_REP')

;

select \* from order\_items;

select \* from employees;

select \* from orders;

select \* from customers;

/\*12 Проверить, были ли заказы, в которых товары поставлялись со скидкой. Считаем, что скидка была,

если сумма заказа меньше суммы стоимости всех позиций в заказе, если цены товаров смотреть в каталоге (прайсе).

Если такие заказы были, то вывести максимальный процент скидки среди всех таких заказов, округленный до 2 знаков после

запятой.\*/

select max(round((oi.real\_price - o.order\_total) / oi.real\_price \* 100,2)) as max\_discount\_percent

from orders o

join (select sum(p.list\_price \* oi.quantity) as real\_price,

oi.order\_id

from order\_items oi

inner join product\_information p on

p.product\_id = oi.product\_id

group by oi.order\_id

) oi on

oi.order\_id = o.order\_id

;

/\*13! Выбрать товары, которые есть только на одном складе. Вывести поля: код товара, название товара,

цена товара по каталогу (LIST\_PRICE), код и название склада, на котором есть данный товар, страна, в которой

находится данный склад. Упорядочить данные по названию стране, затем по коду склада, затем по названию товара.\*/

select p.product\_id,

p.product\_name,

p.list\_price,

w.warehouse\_id,

w.warehouse\_name,

c.country\_name

from product\_information p

inner join (select i.product\_id,

count (i.warehouse\_id) as warehouse\_count,

min(i.warehouse\_id) as warehouse\_id

from inventories i

group by i.product\_id) pr on

pr.product\_id = p.product\_id and

pr.warehouse\_count = 1

inner join warehouses w on

pr.warehouse\_id = w.warehouse\_id

inner join locations l on

w.location\_id = l.location\_id

inner join countries c on

l.country\_id = c.country\_id

order by c.country\_name,

w.warehouse\_id,

p.product\_name

;

/\*14 Для всех стран вывести количество клиентов, которые находятся в данной стране.

Вывести поля: код страны, название страны, количество клиентов. Для стран, в которых нет

клиентов, в качестве количества клиентов вывести 0. Упорядочить по количеству клиентов в

обратном порядке, затем по названию страны.\*/

select cn.country\_id,

cn.country\_name,

nvl(cs.customers\_count, 0) as customers\_count

from countries cn

left join (select cs.cust\_address\_country\_id,

count(cs.customer\_id) as customers\_count

from customers cs

group by cs.cust\_address\_country\_id

) c on

cs.cust\_address\_country\_id = cn.country\_id

order by customers\_count desc,

cn.country\_name

;

/\*15! Для каждого клиента выбрать минимальный интервал (количество дней) между его заказами.

Интервал между заказами считать как разницу в днях между датами 2-х заказов без учета времени заказа.

Вывести поля: код клиента, имя клиента (фамилия + имя через пробел), даты заказов с минимальным интервалом

(время не отбрасывать), интервал в днях между этими заказами. Если у клиента заказов нет или заказ один за

всю историю, то таких клиентов не выводить. Упорядочить по коду клиента.\*/

select c.customer\_id,

c.cust\_last\_name || ' ' || c.cust\_first\_name as cust\_name,

o3.min\_date

from orders o

inner join (select o1.customer\_id,

min(o2.order\_date - o1.order\_date) as min\_date

from orders o1

join orders o2 on

o2.customer\_id = o1.customer\_id and

o2.order\_date > o1.order\_date

group by o1.customer\_id

) o3 on o.customer\_id = o3.customer\_id

inner join customers c on o3.customer\_id = c.customer\_id

group by c.customer\_id,

c.cust\_last\_name || ' ' || c.cust\_first\_name,

o3.min\_date

order by c.customer\_id

;

/\*

1. Выбрать с помощью иерархического запроса сотрудников 3-его уровня иерархии (т.е.

таких, у которых непосредственный начальник напрямую подчиняется руководителю

организации). Упорядочить по коду сотрудника

\*/

select e.\*, level

from employees e

where level = 3

start with e.manager\_id is null

connect by prior e.employee\_id = e.manager\_id

order by e.employee\_id

;

/\*

2. Для каждого сотрудника выбрать всех его начальников по иерархии. Вывести поля: код

сотрудника, имя сотрудника (фамилия + имя через пробел), код начальника, имя

начальника (фамилия + имя через пробел), кол-во промежуточных начальников между

сотрудником и начальником из данной строки выборки. Если у какого-то сотрудника

есть несколько начальников, то для данного сотрудника в выборке должно быть

несколько строк с разными начальниками. Упорядочить по коду сотрудника, затем по

уровню начальника (первый – непосредственный начальник, последний – руководитель

организации).

\*/

select connect\_by\_root(e.employee\_id) as emp\_id,

connect\_by\_root(e.first\_name || ' ' || e.last\_name) as emp\_name,

e.employee\_id as man\_id,

e.first\_name || ' ' || e.last\_name as man\_name,

level - 2 as between\_man\_count

from employees e

where level > 1

connect by e.employee\_id = prior(e.manager\_id)

order by emp\_id,

level

;

/\*

3. Для каждого сотрудника посчитать количество его подчиненных, как непосредственных,

так и по иерархии. Вывести поля: код сотрудника, имя сотрудника (фамилия + имя через

пробел), общее кол-во подчиненных.

\*/

select e.employee\_id,

e.first\_name || ' ' || e.last\_name,

count(e.employee\_id) as employee\_count

from employees e

where level > 1

connect by e.employee\_id = prior e.manager\_id

group by e.employee\_id,

e.first\_name,

e.last\_name

;

select \* from employees

order by manager\_id

;

/\*

3. Для каждого сотрудника посчитать количество его подчиненных, как непосредственных,

так и по иерархии. Вывести поля: код сотрудника, имя сотрудника (фамилия + имя через

пробел), общее кол-во подчиненных.

\*/

select e.employee\_id,

e.first\_name || ' ' || e.last\_name,

count(e.employee\_id) as emp\_count

from employees e

where level > 1

connect by e.employee\_id = prior(e.manager\_id)

group by e.employee\_id,

e.first\_name,

e.last\_name

order by emp\_count desc

;

/\*

4. Для каждого заказчика выбрать в виде строки через запятую даты его заказов. Для

конкатенации дат заказов использовать sys\_connect\_by\_path (иерархический запрос). Для

отбора «последних» строк использовать connect\_by\_isleaf

\*/

select o.customer\_id,

substr(sys\_connect\_by\_path(to\_char(o.order\_date2, 'DD.MM.YYYY'), ', '), 3) as order\_dates

from (

select o.customer\_id,

lead(o.order\_date) over (

partition by o.customer\_id

order by o.order\_date

) as order\_date1,

o.order\_date as order\_date2

from orders o

group by o.customer\_id,

o.order\_date

) o

where connect\_by\_isleaf = 1

start with o.order\_date1 is null

connect by o.customer\_id = prior(o.customer\_id) and

o.order\_date1 = prior(o.order\_date2)

;

/\*1. В анонимном PL/SQL блоке распечатать все пифагоровы числа, меньшие 25

(для печати использовать пакет dbms\_output, процедуру put\_line).\*/

declare

c\_max\_num int := 25;

begin

for i in 1..c\_max\_num loop

for j in i..c\_max\_num loop

for k in j..c\_max\_num loop

if i\*i + j\*j = k\*k then

dbms\_output.put\_line(i || ' ' || j || ' ' || k);

end if;

end loop;

end loop;

end loop;

end;

/

/\*2. Переделать предыдущий пример, чтобы для определения, что 3 числа пифагоровы использовалась функция.\*/

declare

c\_max\_num int := 25;

begin

for i in 1..c\_max\_num loop

for j in i..c\_max\_num loop

for k in j..c\_max\_num loop

if first\_package.f\_check\_numbers(i, j, k) then

dbms\_output.put\_line(i || ' ' || j || ' ' || k);

end if;

end loop;

end loop;

end loop;

end;

/

/\*3. Написать хранимую процедуру, которой передается ID сотрудника и которая увеличивает ему зарплату на 10%,

если в 2000 году у сотрудника были продажи. Использовать выборку количества заказов за 2000 год в переменную.

А затем, если переменная больше 0, выполнить update данных.\*/

declare

v\_order\_count int;

v\_salary employees.salary%type;

v\_emp\_id employees.employee\_id%type := 158;

begin

select e.salary

into v\_salary

from employees e

where e.employee\_id = v\_emp\_id;

dbms\_output.put\_line(v\_salary);

first\_package.pr\_increase\_salary(v\_emp\_id);

select e.salary

into v\_salary

from employees e

where e.employee\_id = v\_emp\_id;

dbms\_output.put\_line(v\_salary);

end;

/