

Отчет по лабораторной работе № 6 «Разработка подсистемы ввода/вывода»			
дата	Оценка (max 5)	Бонус за сложность	подпись

Цели работы:

Отработка технологий ввода/вывода информации в информационных системах. Итоговая лабораторная работа систематизирует и обобщает все методы применения PL/SQL и PHP для реализации модулей информационных систем с использованием СУБД Oracle.

Задачи работы:

Подсистема ввода: Ввод данных скриптами, ввод данных в формы пользовательского интерфейса, ввод засекреченных данных, ввод данных из файлов, элементы ввода данных с внешних устройств (сканеров) и т.п.

Подсистема вывода: Форматированный вывод отчетов с помощью процедур в SQL+, форматированный вывод отчетов средствами пользовательского интерфейса, вывод отчетов в текстовые файлы, вывод форматированных отчетов в файлы разных форматов (*.pdf, *.xls и т.п.) и т.п.

Итог: Результат модуля АСУ отдела фирмы согласно варианту задания

Краткий конспект теоретической части (ответы на контрольные вопросы)

Принципы построения подсистемы ввода информации

Принципы построения подсистемы вывода информации

Описание функционала модуля АСУ фирмы по варианту задания

1. Подсистема ввода данных

```
1.1 1 SPool task_1.txt
Дл 2 INSERT INTO components (
СВ: 3     comp_cost
4     ,comp_qua
5     ,comp_type
6     ,comp_name
Ск 7     ,comp_id
8 ) VALUES ( 100
9             ,100
10            , 'Транзистор'
11            , '2n2222'
12            , comp_id_seq.NEXTVAL );
13 INSERT INTO components (
14     comp_cost
15     ,comp_qua
16     ,comp_type
17     ,comp_name
18     ,comp_id
19 ) VALUES ( 1200
20             ,10
21            , 'Транзистор'
22            , '2n3904'
23            , comp_id_seq.NEXTVAL );
24 INSERT INTO components (
25     comp_cost
26     ,comp_qua
27     ,comp_type
28     ,comp_name
29     ,comp_id
30 ) VALUES ( 1200
31             ,10
32            , 'ДУ'
33            , '1409д1А'
34            , comp_id_seq.NEXTVAL );
35 COMMIT;
36 SPool OFF;
```

скриптов

(отделу фирмы) приведите пример ввода данных в пример заполнения справочников)

```
SQL> SELECT * FROM components;
```

COMP_COST	COMP_QUA	COMP_TYPE	COMP_NAME	COMP_ID
1200	10	ДУ	1409д1А	3
100	100	Транзистор	2n2222	1
1200	10	Транзистор	2n3904	2

1.2. Ввод данных посредством интерактивных интерфейсных форм на РНР

Для модуля АСУ согласно варианту (отделу фирмы) приведите пример ввода данных в связанные таблицы через интерактивные РНР формы

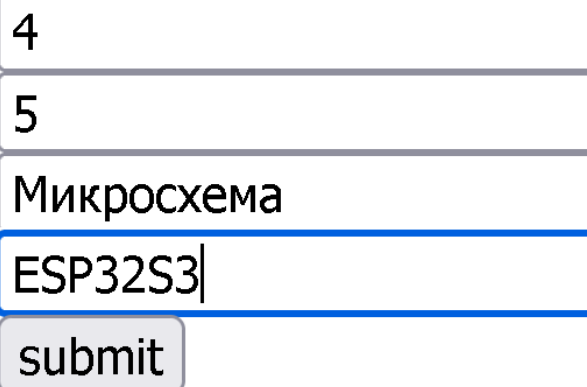
Скрипт:

```

1  <html>
2
3  <head>
4      <meta charset="utf-8">
5  </head>
6
7  <body>
8      <form TARGET="task2.php" METHOD="POST">
9          <input type="text" name="comp_cost" placeholder="cost" required /> <br>
10         <input type="text" name="comp_qua" placeholder="count" required /> <br>
11         <input type="text" name="comp_type" placeholder="type" required /> <br>
12         <input type="text" name="comp_name" placeholder="name" required /> <br>
13         <input type="submit" value="submit" />
14     </form>
15     <?php
16         if ($_SERVER['REQUEST_METHOD'] == 'POST') {
17             $conn = oci_connect("system", "oracle_password", "host.docker.internal:1521/FREE");
18             if (! $conn) {
19                 echo("Невозможно подключиться к базе: " . var_dump(oci_error()));
20                 die();
21             }
22
23             $query = "
24                 INSERT INTO components (
25                     comp_cost
26                     ,comp_qua
27                     ,comp_type
28                     ,comp_name
29                     ,comp_id
30                 ) VALUES (
31                     " . $_POST["comp_cost"] . "
32                     , " . $_POST["comp_qua"] . "
33                     , " . $_POST["comp_type"] . "
34                     , " . $_POST["comp_name"] . "
35                     ,comp_id_seq.NEXTVAL
36                 )
37             ";
38             $str = oci_parse($conn, $query);
39             oci_execute($str, OCI_DEFAULT);
40             oci_commit($conn);
41             oci_close($conn);
42         }
43     ?>
44
45 </body>
46
47 </html>

```

Скриншот интерактивной формы:



A screenshot of an interactive form. It consists of four vertically stacked input fields and a 'submit' button at the bottom. The first field contains the number '4', the second contains '5', the third contains 'Микросхема', and the fourth contains 'ESP32S3'. The 'submit' button is a light gray rounded rectangle.

1.3. Ввод засекреченных данных

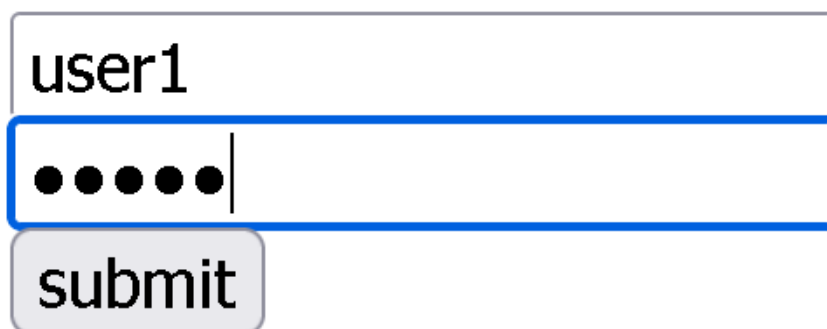
Для модуля АСУ согласно варианту (отделу фирмы) приведите пример ввода засекреченных данных (модуль авторизации)

Скрипт:



A series of 15 horizontal lines for writing a script.

Скриншот:



A screenshot of a login form. It has two input fields: the top one contains 'user1' and the bottom one contains five dots, indicating a password. Below the password field is a 'submit' button. The password field and the submit button are highlighted with blue borders.

```

1  <html>
2
3  <head>
4      <meta charset="utf-8">
5  </head>
6
7  <body>
8
9      <form TARGET="task_3.php" METHOD="POST">
10         <input type="login" name="login" placeholder="login" required /> <br>
11         <input type="password" name="password" placeholder="password" required /> <br>
12         <input type="submit" value="submit" />
13     </form>
14     <?
15         if ($_SERVER['REQUEST_METHOD'] == 'POST') {
16             $conn = oci_connect("system", "oracle_password", "host.docker.internal:1521/FREE");
17             if (!$conn) {
18                 echo("Невозможно подключиться к базе: " . var_dump(oci_error()));
19                 die();
20             }
21             $query = "
22             SELECT * FROM users WHERE LOGIN = '" . $_POST['login'] . "' AND PASSWORD = '" . $_POST['password'] . "'
23             ";
24             $str = oci_parse($conn, $query);
25             oci_execute($str, OCI_DEFAULT);
26             $row = oci_fetch($str);
27             if (oci_num_rows($str) >= 1) {
28                 echo("Пользователь найден");
29             } else {
30                 echo("Пользователь не найден");
31             }
32
33             oci_commit($conn);
34             oci_close($conn);
35         }
36     >
37 </body>
38
39 </html>

```

1.4. Ввод данных из файла

Для модуля АСУ согласно варианту (отделу фирмы) приведите пример ввода данных из текстового файла в БД.

Скрипт:

```
3 <head>
4   <meta charset="utf-8">
5 </head>
6
7 <body>
8   <form action="" target="task_4.php" method="POST" enctype="multipart/form-data">
9     <input type="file" name="file" required /> <br>
10    <input type="submit" value="submit" />
11  </form>
12  <?
13    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
14      $conn = oci_connect("system", "oracle_password", "host.docker.internal:1521/FREE");
15      if (!$conn) {
16        echo("Невозможно подключиться к базе: " . var_dump(oci_error()));
17        die();
18      }
19
20      if (($handle = fopen($_files['file']['tmp_name'], 'r')) !== false) {
21        $query = "BEGIN";
22        while (($data = fgetcsv($handle, 1000, ",")) !== false) {
23          $query .= "
24            INSERT INTO components (
25              comp_cost
26              ,comp_qua
27              ,comp_type
28              ,comp_name
29              ,comp_id
30            ) VALUES (
31              " . $data[0] . "
32              , " . $data[1] . "
33              , " . $data[2] . "
34              , " . $data[3] . "
35              ,comp_id_seq.NEXTVAL
36            );
37          ";
38        }
39        $query .= "END;";
40        fclose($handle);
41        $str = oci_parse($conn, $query);
42        oci_execute($str, OCI_DEFAULT);
43        oci_commit($conn);
44        oci_close($conn);
45      }
46    }
47  <?>
48 </body>
49
50 </html>
```

Обзор...

Файл не выбран.

submit

1.5 Ввод данных посредством внешних устройств

Для модуля АСУ согласно варианту (отделу фирмы) приведите пример ввода данных при помощи сканера штрих кодов (<http://oracle.iu4.bmstu.ru/grid/sem/sem8/index.php>).

Скрипт:

```
1  <html>
2
3  <head>
4  |   <meta charset="utf-8">
5  </head>
6
7  <body>
8  |   <form TARGET="task5.php" METHOD="POST">
9  |       <input type="barcode" name="barcode" placeholder="barcode EAN-13" required /><br>
10 |       <input type="submit" value="submit" />
11 |   </form>
12 |   <?php
13 |       if ($_SERVER['REQUEST_METHOD'] == 'POST') {
14 |           $conn = oci_connect("system", "oracle_password", "host.docker.internal:1521/FREE");
15 |           if (! $conn) {
16 |               echo("Невозможно подключиться к базе: " . var_dump(oci_error()));
17 |               die();
18 |           }
19 |
20 |           $query = "
21 |               INSERT INTO components_storage (
22 |                   comp_storage_id
23 |               ) VALUES (
24 |                   '' . $_POST['barcode'] . ''
25 |               )
26 |           ";
27 |           $str = oci_parse($conn, $query);
28 |           oci_execute($str, OCI_DEFAULT);
29 |           oci_commit($conn);
30 |           oci_close($conn);
31 |       }
32 |   ?>
33
34 </body>
35
36 </html>
```

barcode EAN-13

submit

```
SQL> SELECT * FROM components_storage;
```

COMP_STORAGE_ID

123456789

2. Подсистема вывода данных

Реализовать различными способами вывода информации получение отчетов по следующим задачам для схемы HR:

Задание 2.1: Решить задачу и организовать форматированный вывод отчетов с помощью процедур в SQL+:

Отобразить всех сотрудников, находящихся в подчинении вице-президента по фамилии Kochhar. В выходной таблице должно быть 2 столбца:

1. Фамилия сотрудника. Первая строка должна содержать Kochhar без отступа, вторая - сотрудника, непосредственно подчиняющегося Kochhar с отступом в 2, далее - сотрудники на третьем уровне подчинения с отступом в 4 и т.д. Для идентификации отступа использовать символ . (точка)
2. Второй столбец должен содержать фамилии сотрудников, отражающих иерархию подчинения, разделённых знаком /

Дополнительные требования к выполнению:

Пример фрагмента вывода:

Фамилия	Подчинение
Kochhar	/Kochhar
Baer	/Kochhar/Baer
Greenberg	/Kochhar/Greenberg
Chen	/Kochhar/Greenberg/Chen
.	
Urman	/Kochhar/Greenberg/Urman
Higgins	/Kochhar/Higgins
Gietz	/Kochhar/Higgins/Gietz
Mavris	/Kochhar/Mavris
Whalen	/Kochhar/Whalen

Фамилии подчинённых должны быть упорядочены в алфавитном порядке на КАЖДОМ уровне иерархии

Подсказки:

1. Для реализации отступов использовать функцию SQL lpad и псевдостолбец level
2. Для перечисления фамилий в иерархии во втором столбце использовать функцию Oracle 9i sys_connect_by_path.
3. Для сортировки на каждом уровне иерархии использовать конструкцию order siblings by.

Скрипт:

```
1  SPOOL task6.txt
2  COL "Фамилия" FORMAT A20
3  COL "Подчинение" FORMAT A30
4  SELECT lpad(
5      last_name
6      ,length(last_name) + 2 *(level - 1)
7      ,'. '
8  ) "Фамилия"
9      ,sys_connect_by_path(
10     last_name
11     , '/'
12 ) "Подчинение"
13 FROM hr.employees
14 CONNECT BY
15     PRIOR employee_id = manager_id
16 START WITH employee_id = 101
17 ORDER SIBLINGS BY last_name;
18 SPOOL OFF;
```

	Фамилия	Подчинение
1	Yang	/Yang
2	..Brown	/Yang/Brown
3	..Gruenberg	/Yang/Gruenberg
4Chen	/Yang/Gruenberg/Chen
5Faviet	/Yang/Gruenberg/Faviet
6Popp	/Yang/Gruenberg/Popp
7Sciarra	/Yang/Gruenberg/Sciarra
8Urman	/Yang/Gruenberg/Urman
9	..Higgins	/Yang/Higgins
10Gietz	/Yang/Higgins/Gietz
11	..Jacobs	/Yang/Jacobs
12	..Whalen	/Yang/Whalen

Задание 2.2: Решить задачу и организовать форматированный вывод отчетов средствами пользовательского интерфейса на РНР:

Для каждого данного сотрудника, имеющего оклад выше среднего в его отделе, отобразить всех сотрудников его отдела, имеющих оклад, больше чем у данного сотрудника.

Вывести данные для отделов с номерами 60 и 80.

Выходная таблица должна содержать следующие столбцы:

1. Фамилия данного сотрудника
2. Оклад данного сотрудника
3. Фамилия сотрудника, с большим окладом
4. Оклад сотрудника, с большим окладом
5. Отдел
6. Средний оклад по отделу (округлить до 2-х знаков после запятой)

Данные в выходной таблице должны быть упорядочены по номеру отдела, окладу данного сотрудника, фамилии данного сотрудника, окладу и фамилии сотрудника с большим чем у данного окладом.

Скрипт:

Вид отчета:

Фамилия	Оклад	Фамилия	Оклад	Отдел	Средний оклад
Miller	6000	James	9000	60	5760
Hall	9000	Bernstein	9500	80	8956
Hall	9000	Greene	9500	80	8956
Hall	9000	Sully	9500	80	8956
Hall	9000	Fox	9600	80	8956
Hall	9000	Bloom	10000	80	8956

```

1  <html>
2
3  <head>
4  |   <meta charset="utf-8">
5  </head>
6
7  <body>
8  |   <?php
9  |       $conn = oci_connect("system", "oracle_password", "host.docker.internal:1521/FREE");
10 |       if (! $conn) {
11 |           echo("Невозможно подключиться к базе: " . var_dump(oci_error()));
12 |           die();
13 |       }
14 |       $query = "
15 |           SELECT e1.last_name
16 |             ,e1.salary
17 |             ,e2.last_name
18 |             ,e2.salary
19 |             ,e1.department_id
20 |             ,g.avsas
21 |       FROM hr.employees e1
22 |            ,hr.employees e2
23 |            ,(
24 |              SELECT department_id
25 |                ,round(avg(salary)) avsas
26 |              FROM hr.employees
27 |             GROUP BY department_id
28 |            ) g
29 |      WHERE e1.department_id = e2.department_id
30 |            AND e1.department_id = g.department_id
31 |            AND e1.salary > g.avsas
32 |            AND e1.salary < e2.salary
33 |            AND e2.employee_id <> e1.employee_id
34 |            AND e1.department_id IN ( 60
35 |                                     ,80 )
36 |      ORDER BY e1.department_id
37 |             ,e1.salary
38 |             ,e1.last_name
39 |             ,e2.salary
40 |             ,e2.last_name
41 |      ";
42 |       $str = oci_parse($conn, $query);
43 |       oci_execute($str, OCI_DEFAULT);
44 |   >
45 |   <center>
46 |       <table border=1>
47 |           <tr>
48 |               <td>Фамилия</td>
49 |               <td>Оклад</td>
50 |               <td>Фамилия</td>
51 |               <td>Оклад</td>
52 |               <td>Отдел</td>
53 |               <td>Средний оклад</td>
54 |           </tr>
55 |           <?php
56 |               while (oci_fetch($str)) {
57 |                   echo("<tr>");
58 |                   echo("<td>" . oci_result($str, 1) . "</td>");
59 |                   echo("<td>" . oci_result($str, 2) . "</td>");
60 |                   echo("<td>" . oci_result($str, 3) . "</td>");
61 |                   echo("<td>" . oci_result($str, 4) . "</td>");
62 |                   echo("<td>" . oci_result($str, 5) . "</td>");
63 |                   echo("<td>" . oci_result($str, 6) . "</td>");
64 |                   echo("</tr>");
65 |               }
66 |           >
67 |       </table>
68 |   </center>
69 |   <?php
70 |       oci_commit($conn);
71 |       oci_close($conn);
72 |   >
73 </body>
74 </html>

```

Задание 2.3: Решить задачу и организовать вывод отчетов в текстовые файлы

Имеется: таблица с тремя столбцами: именем, фамилией и коэффициентом размножения, созданная и загруженная следующим образом:

```
drop table EMP_SELECTED;
create table EMP_SELECTED (
    First_name varchar2(20) not null
    , Last_name  varchar2(20) not null
    , N integer not null);
insert into EMP_SELECTED values('Ellen', 'ABEL', 3);
insert into EMP_SELECTED values('Matthew', 'WEISS', 5);
commit;
```

Требуется написать запрос, выводящий на печать таблицу, содержащую строки с именами и фамилиями двух сотрудников. Число строк для каждого сотрудника должно определяться коэффициентом размножения (столбец N исходной таблицы). То есть, должны быть 3 строки для сотрудника Ellen ABEL и 5 строк для Matthew WEISS. Строки должны быть объединены в группы и отсортированы по фамилии и имени. Кроме того, должны быть пронумерованы элементы внутри группы и присутствовать сквозная нумерация. Этот select должен работать для произвольного количества строк в исходной таблице EMP_SELECTED. В выходной таблице должны присутствовать следующие столбцы:

- сквозной номер строки по порядку;
- номер сотрудника в группе;
- имя (First_name);
- фамилия (Last_name).

Пример выходного отчёта:

Сквозной №	№ в группе	Имя	Фамилия
1	1	Ellen	ABEL
2	2	Ellen	ABEL
3	3	Ellen	ABEL
4	1	Matthew	WEISS
5	2	Matthew	WEISS
6	3	Matthew	WEISS
7	4	Matthew	WEISS
8	5	Matthew	WEISS

8 rows selected.

Подсказки:

1. Для оформления нумераций воспользоваться аналитической функцией row_number() OVER.
2. Для получения таблицы натуральных чисел воспользоваться конструкцией connect by Level, появившейся в Oracle 10g, например:

```
select Level from DUAL connect by Level <= 10;
```

выдаёт 10 натуральных чисел, начиная с 1.

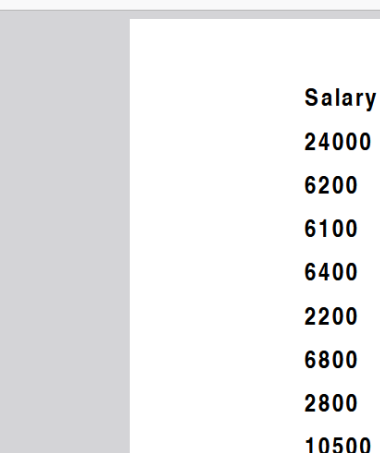
(Здесь DUAL стандартная таблица Oracle с одним столбцом и одной строкой)

```
select
    row_number() OVER(order by Last_name, First_name) "Сквозной №"
    , row_number() OVER(PARTITION BY Last_name, First_name Order by
                        Last_name, First_name) "№ в группе"
    , First_name "Имя"
    , Last_name  "Фамилия"
from EMP_SELECTED
    , (select /* Этот select выдаёт таблицу с MAX(N) строками и со значениями в
    столбце 1,2,..max(N) */
        level Lev
    from DUAL
        connect by Level <= (select max(N) from EMP_SELECTED))
where Lev <= N /* Хитрый join */
order by Last name, First name;
```

```

1
2   Сквозной # # в группе Имя                               Фамилия
3   ✓ -----
4   |         |         | 1      1 Eilen                      ABEL
5   |         |         | 2      2 Eilen                      ABEL
6   |         |         | 3      3 Eilen                      ABEL
7   |         |         | 4      1 Matthe(w)                WEISS
8   |         |         | 5      2 Matthe(w)                WEISS
9   |         |         | 6      3 Matthe(w)                WEISS
10  |         |         | 7      4 Matthe(w)                WEISS
11  |         |         | 8      5 Matthe(w)                WEISS
12
13      8 rows selected.
14

```



The screenshot shows a web browser window with the address bar displaying 'localhost/lab_6/file.pdf'. The main content area of the browser shows a PDF document. The PDF content is a table with two columns. The first column is labeled 'Salary' and the second column contains the following values: 24000, 6200, 6100, 6400, 2200, 6800, 2800, 10500, and 7200.

Salary
24000
6200
6100
6400
2200
6800
2800
10500
7200

```

1  <html>
2
3  <head>
4      <meta charset="UTF-8" />
5  </head>
6
7  <body>
8      <?php
9          require './fpdf/fpdf.php';
10         $conn = oci_connect("system", "oracle_password", "host.docker.internal:1521/FREE");
11         if (! $conn) {
12             echo("Невозможно подключиться к базе: " . var_dump(oci_error()));
13             die();
14         }
15         $query = "
16             SELECT salary
17             FROM (SELECT salary, MAX(hire_date) hire_date
18                   FROM (SELECT hire_date, salary
19                         FROM (SELECT hire_date, salary
20                               FROM HR.employees
21                               ORDER BY hire_date DESC, salary DESC)
22                               WHERE rownum <= 50)
23                         GROUP BY salary
24                         ORDER BY 2 DESC)
25             WHERE rownum<20
26         ";
27         $str = oci_parse($conn, $query);
28         oci_execute($str, OCI_DEFAULT);
29         $data = [];
30         $file = new FPDF();
31         $file->AddPage();
32         $file->SetFont('Arial', 'B', 16);
33         $file->Text(40, 20, 'Salary');
34         $margin = 20;
35         while (oci_fetch($str)) {
36             $margin += 10;
37             $file->Text(40, $margin, oci_result($str, 1));
38         }
39         $file->Output("F", "file.pdf");
40         echo " <a href='./file.pdf' download>Скачать файл<br>";
41         echo " <a href='./file.pdf'>Открыть файл";
42         oci_commit($conn);
43         oci_close($conn);
44     ?>
45 </body>
46
47 </html>

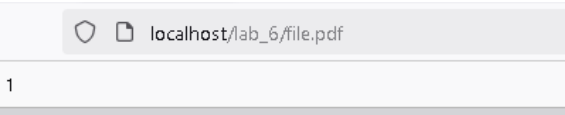
```

Задание 2.4а: Решить задачу 2.4. и организовать форматированный вывод отчетов посредством формирования различных штрихкодов:

Скрипт:

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper has a slight shadow on its right side, suggesting it's resting on a surface.

Вид отчета (комбинации штрихкодов):



[Скачать файл](#)
[Открыть файл](#)



```

1 <html>
2
3 <head>
4 |   <meta charset="UTF-8" />
5 </head>
6
7 <body>
8 |   <?php
9 |       require './barcode/fpdf/fpdf.php';
10 |       $conn = oci_connect("system", "oracle_password", "host.docker.internal:1521/FREE");
11 |       if (! $conn) {
12 |           echo("Невозможно подключиться к базе: " . var_dump(oci_error()));
13 |           die();
14 |       }
15 |       $query = "
16 |           SELECT salary
17 |           FROM (SELECT salary, MAX(hire_date) hire_date
18 |           FROM (SELECT hire_date, salary
19 |           FROM (SELECT hire_date, salary
20 |           FROM HR.employees
21 |           ORDER BY hire_date DESC, salary DESC)
22 |           WHERE rownum <= 50)
23 |           GROUP BY salary
24 |           ORDER BY 2 DESC)
25 |           WHERE rownum<20
26 |       ";
27 |       $str = oci_parse($conn, $query);
28 |       oci_execute($str, OCI_DEFAULT);
29 |       $data = [];
30 |       $file = new FPDF();
31 |       $file->AddFont('barcode', '', 'barcode.php');
32 |       $file->AddPage();
33 |       $file->SetFont('barcode', '', 25);
34 |       $margin = 20;
35 |       while (oci_fetch($str)) {
36 |           $margin += 10;
37 |           $file->Text(40, $margin, oci_result($str, 1));
38 |       }
39 |       $file->Output("F", "file.pdf");
40 |       echo " <a href='./file.pdf' download>Скачать файл<br>";
41 |       echo " <a href='./file.pdf'>Открыть файл";
42 |       oci_commit($conn);
43 |       oci_close($conn);
44 |   ?>
45 </body>
46
47 </html>

```

ИТОГО: Вклеить основные интерфейсные формы ввода/вывода разработанного модуля АСУ согласно варианту задания

 Файл не выбран.

[Скачать файл](#)
[Открыть файл](#)

Задания для самоконтроля

Задание 2.5: Решить задачу и организовать форматированный вывод отчетов с помощью процедур в SQL+:

Оператор отдела кадров при регистрации сотрудников с номерами 194 и 195 присвоила им данные друг друга. Необходимо исправить эту ошибку. Ситуация осложняется тем что у вас нет доступа к другим полям таблицы за исключением поля employee_id, являющегося первичным ключом. Необходимо решить задачу одним оператором SQL.

Проверка 1:

Вид проверки:

Корректность решения

Эталонный запрос:

Update employees set employee_id=Decode(employee_id,194,195,195,194) where employee_id in (194,-195)

Дополнительная информация:

Суть проверки – выполнение запроса

Select RowID,N from T2 where N in (5,-2)

До и после прогона представленного решения

Скрипт:

```
UPDATE hr.employees
SET
  employee_id = decode(
    employee_id
    ,194
    ,195
    ,195
    ,194
  )
WHERE employee_id IN ( 194
                      ,195 );
```

Вид отчета:

ID	NAME
195	Vance Jones
194	Samuel McLeod

2 rows updated.

ID	NAME
194	Vance Jones
195	Samuel McLeod

Задание 2.6: Решить задачу и организовать форматированный вывод отчетов с помощью процедур в SQL+:

Как известно, неделя у разных народов начинается с разных дней. Надо с помощью стандартных функций Oracle создать выражение, вычисляющее номер дня недели (начиная с понедельника), независимо от текущей версии Oracle, NLS-установок и кодировок.

Дополнительные требования к выполнению:

По возможности надо обойтись только стандартными функциями Oracle

Проверка 1:

Каноническое решение для текущей даты:

Select (Instr('MONTUEWEDTHUFRISATSUN', To_Char(SysDate, 'DY',
'NLS_DATE_LANGUAGE = AMERICAN')) + 2) / 3 from dual

Скрипт:

```
1  SPOOL task_12.txt
2  SELECT ( instr(
3      'MONTUEWEDTHUFRISATSUN'
4      ,to_char(
5          sysdate
6          , 'DY'
7          , 'NLS_DATE_LANGUAGE = AMERICAN'
8      )
9  ) + 2 ) / 3 "День от начала недели"
10 FROM dual;
11 SPOOL OFF;
```

Вид отчета:

```
- |
2  День от начала недели
3  ✓ -----
4  | | | | | 2
5
```

Задание 2.7: Решить задачу и организовать форматированный вывод отчетов с помощью процедур в SQL+:

Использование конструкции SQL rollup (задание № 1) Написать запрос, выдающий отчёт о суммарных выплатах сотрудникам, непосредственно подчиняющихся заданному руководителю по идентификаторам должностей (поле Job_id). Непосредственное подчинение предполагает подчинение на первом уровне. Иными словами, записи о сотрудниках, непосредственно подчиняющихся сотруднику с Employee_id, равным 101, содержат 101 в поле Manager_id.

Отчёт должен содержать группы строк. Каждая группа относится к данному руководителю и состоит из регулярных строк, отображающих суммарные выплаты и количество сотрудников на данной должности, непосредственно подчиняющихся этому руководителю. Группу должна завершать строка с итоговыми значениями суммарных выплат и количества сотрудников, для сотрудников, непосредственно подчиняющихся данному руководителю. Итоговая строка не должна содержать значение в поле идентификатора должности.

Кроме того, отчёт должен быть завершён строкой, представляющей общий итог и содержащей сумму выплат и количество сотрудников по всем упомянутым руководителям. В этой строке поля идентификаторов руководителя и должности должны быть пустыми.

Столбцы отчёта:

1. Идентификатор руководителя. Для строки, представляющей общий итог это поле должно быть пустым. Для остальных строк в этом поле представлен соответствующий идентификатор (Manager_id).

2. Идентификатор должности.

Для регулярных строк здесь должен присутствовать соответствующий идентификатор(Job_id).

Для итоговых строк и для строки общего итога – пустое значение.

3. Количество сотрудников.

Для регулярных строк – количество сотрудников на данной должности у данного руководителя.

Для итоговых строк – количество сотрудников, находящихся в непосредственном подчинении данного руководителя.

Для строки общего итога – общее количество сотрудников, находящихся в непосредственном подчинении у всех руководителей, представленных в отчёте.

Суммарные выплаты.

Для регулярных строк -суммарные выплаты для сотрудников, находящихся на данной должности у данного руководителя.

Для итоговых строк – суммарные выплаты всем сотрудникам, находящимся в непосредственном подчинении данного руководителя. Для строки общего итога – сумма выплат сотрудникам, находящимся в непосредственном подчинении у всех руководителей, представленных в отчёте.

Месячная суммарная выплата каждому сотруднику представляет собой оклад (столбец Salary) плюс комиссионные (столбец Commission_pct), представляющие указанную часть оклада (положительное число < 1).

Проверка 1:

Вид проверки: Название проверяемой правильности. Например: «проверка на данных имеющих неопределённые значения» БД: Код базы данных на котором проводится проверка. Для одной проверки используем HR. Код всех дополнительных баз, имеющих данные, отличные от HR согласуем впоследствии дополнительно и впишем сюда.

Скрипт:

```
1 SPOOL task_13.txt
2 SELECT manager_id "ID менеджера"
3       ,job_id "Должность"
4       ,COUNT(*) "Число обладателей"
5       ,SUM(salary *(1 + nvl(
6         commission_pct
7         ,0
8       ))) "Выплаты"
9 FROM hr.employees
10 GROUP BY ROLLUP(manager_id
11              ,job_id);
12
13 SPOOL OFF;
```

	ID менеджера	Должность	Число обладателей	Выплаты
58				
59				
60		124	8	23000
61		145	6	62775
62		146	6	67375
63		147	6	53310
64		148	6	61510
65		149	6	60570
66		201	1	6000
67		205	1	8300
68			108	789106
69				
70				

53 rows selected.

58	Руководитель	Должность	Сотрудников	Сумма
59	-----	-----	-----	-----
60Kevin Mourgositогг	Stock Manager	8	23000
61John Singhitогг	Sales Manager	6	62775
62Karen Partnersитогг	Sales Manager	6	67375
63Alberto Errazurizитогг	Sales Manager	6	53310
64Gerald Cambraultитогг	Sales Manager	6	61510
65Eleni Zlotkeyитогг	Sales Manager	6	60570
66Michael Martinezитогг	Marketing Manager	1	6000
67Shelley Higginsитогг	Accounting Manager	1	8300
68ОБЩИЙ	Итог	108	789106
69				
70	53 rows selected.			

```

1  SPOOL task_14.txt
2  SELECT decode(
3      gm + gj
4      ,0
5      ,(
6          SELECT first_name
7              || ' '
8              || last_name
9          FROM hr.employees
10         WHERE employee_id = m.manager_id
11      )
12      ,1
13      ,(
14          SELECT lpad(
15              first_name
16              || ' '
17              || last_name
18              || 'итоги'
19              ,30
20              ,'.'
21          )
22          FROM hr.employees
23          WHERE employee_id = m.manager_id
24      )
25      ,2
26      ,lpad(
27          'ОБЩИЙ'
28          ,25
29          ,'.'
30      )
31  ) "Руководитель"
32      ,decode(
33          gm + gj
34          ,0
35          ,(
36              SELECT job_title
37                  FROM hr.jobs
38              WHERE job_id = m.job_id
39          )
40          ,1
41          ,(
42              SELECT jobs.job_title
43                  FROM hr.employees
44                  ,hr.jobs
45              WHERE employee_id = m.manager_id
46                  AND employees.job_id = jobs.job_id
47          )
48          ,2
49          ,'Итого'
50  ) "Должность"
51      ,cnt "Сотрудников"
52      ,amount "Сумма"
53  FROM (
54      SELECT manager_id
55          ,job_id
56          ,COUNT(*) cnt
57          ,SUM(salary *(1 + nvl(
58              commission_pct
59              ,0
60          ))) amount
61          ,GROUPING(manager_id) gm
62          ,GROUPING(job_id) gj
63      FROM hr.employees
64      GROUP BY ROLLUP(manager_id
65          ,job_id)
66  ) m;
67  SPOOL OFF;

```

[Главная](#) > [Профиль](#)

Результат тестирования

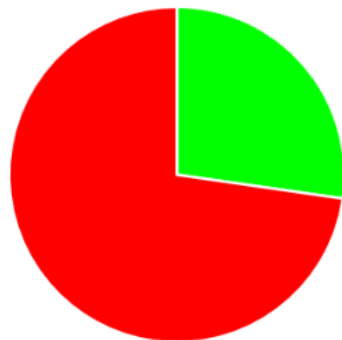
Тест: [Oracle 19c. Основы SQL](#)

Результат: Не сдан

Время: 22:26

Дата тестирования: 29.04.2025 22:28:00

■ Правильные ответы (3)
■ Неправильные ответы (8)



Обратите внимание! Мы не показываем ошибки, которые Вы сделали, потому что смысл теста – определить Ваш уровень знаний, а не дать список правильных ответов. За знаниями приходите на курсы!

[Oracle Database: Основы SQL](#)

По модулям

Операторы DML, DDL и DCL - **0%**

Оператор SELECT - **16%**

Специфические возможности SQL для СУБД Oracle - **100%**