

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. Ввод данных посредством SQL скриптов

Для модуля АСУ согласно варианту (отделу фирмы) приведите пример ввода данных в связанные таблицы SQL скриптами (пример заполнения справочников)

Скрипт:

[illegible]

Проверка:

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:

[illegible]

Скриншот интерактивной формы:

<p>1.3. Ввод засекреченных данных</p> <p>Для модуля АСУ согласно варианту (отделу фирмы) приведите пример ввода засекреченных данных (модуль авторизации)</p>
--

[illegible]

Скриншот:

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

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

[illegible]

Проверка:

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>).

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>).

[illegible]

Проверка:

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:

Вид отчета:

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:

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 таблице). 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;
```


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.

Скрипт:

Задание 2.4: Решить задачу и организовать вывод форматированных отчетов в файлы разных форматов (*.pdf, *.xls и т.п.)

Из таблицы EMPLOYEES надо выбрать не более 19-ти различных зарплат (столбец SALARY) среди самых новых сотрудников (столбец HIRE_DATE), анализируя не более 50 сотрудников. Желательно осуществить выборку одним SQL-запросом

```
Select Salary from
(Select Salary,Max(Hire_Date) Hire_Date from
(Select Hire_Date,Salary from
(Select Hire_Date,Salary from Employees order by Hire_Date Desc,Salary desc)
where RowNum<=50)
Group by Salary
order by 2 desc)
Where RowNum<20
```

Результат:

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

Скрипт:

Вид отчета (комбинации штрихкодов):

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

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:

Вид отчета:

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:

Вид отчета:

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:

Вид отчета:

Задание 2.8: Решить задачу и организовать форматированный вывод отчетов с помощью процедур в SQL+:

Написать запрос, выдающий отчёт о суммарных выплатах сотрудникам, непосредственно подчиняющихся руководителю (задаётся полное имя) по названиям должностей (поле JOBS.Job_Title).

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.

Кроме того, отчёт должен быть завершён строкой, представляющей общий итог и содержащей количество сотрудников и сумму выплат и по всем упомянутым руководителям.

Столбцы отчёта:

1. Полное имя руководителя.

Итоговые строки в этом поле должны содержать полное имя руководителя (First_name, пробел, Last_name из таблицы EMPLOYEES) с отступом (несколько точек) за которым следует текстовая константа “итоги:”.

Для строки, представляющей общий итог, это поле должно содержать текстовую константу “О Б Щ И Й” с отступом, представленным несколькими точками.

Для остальных (регулярных) строк в этом поле должно быть полное имя руководителя (First_name, пробел, Last_name) без отступа.

2. Название должности.

Для итоговых строк это поле должно содержать название должности руководителя (поле Job_Title из таблицы JOBS).

и для строки общего итога – пустое значение. а в поле названия должностей – текстовая константа “И Т О Г”.

Для регулярных строк здесь должно быть название должности, которую занимают сотрудники с представленными в следующих полях количеством и суммарными выплатами.

3. Количество сотрудников.

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

Для итоговых строк – количество сотрудников, находящихся данного руководителя.

Для регулярных строк – количество сотрудников на данной должности в непосредственном подчинении данного руководителя.

Суммарные выплаты.

Для строки общего итога – сумма выплат сотрудникам, находящимся в непосредственном подчинении у всех руководителей, представленных в отчёте.

Для итоговых строк – суммарные выплаты всем сотрудникам, находящимся в непосредственном подчинении данного руководителя. Для регулярных строк -суммарные месячные выплаты для сотрудников, находящихся на данной должности у данного руководителя.

Скрипт:

Вид отчета: