БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра информатики

Факультет НиДО

Специальность ИиТП

Контрольная работа № 1

Практическая работа № 1,2

по дисциплине «МДиСУБД»

Выполнил студент: Дегтярев А.А.

группа 393551

Зачетная книжка № 902021-26

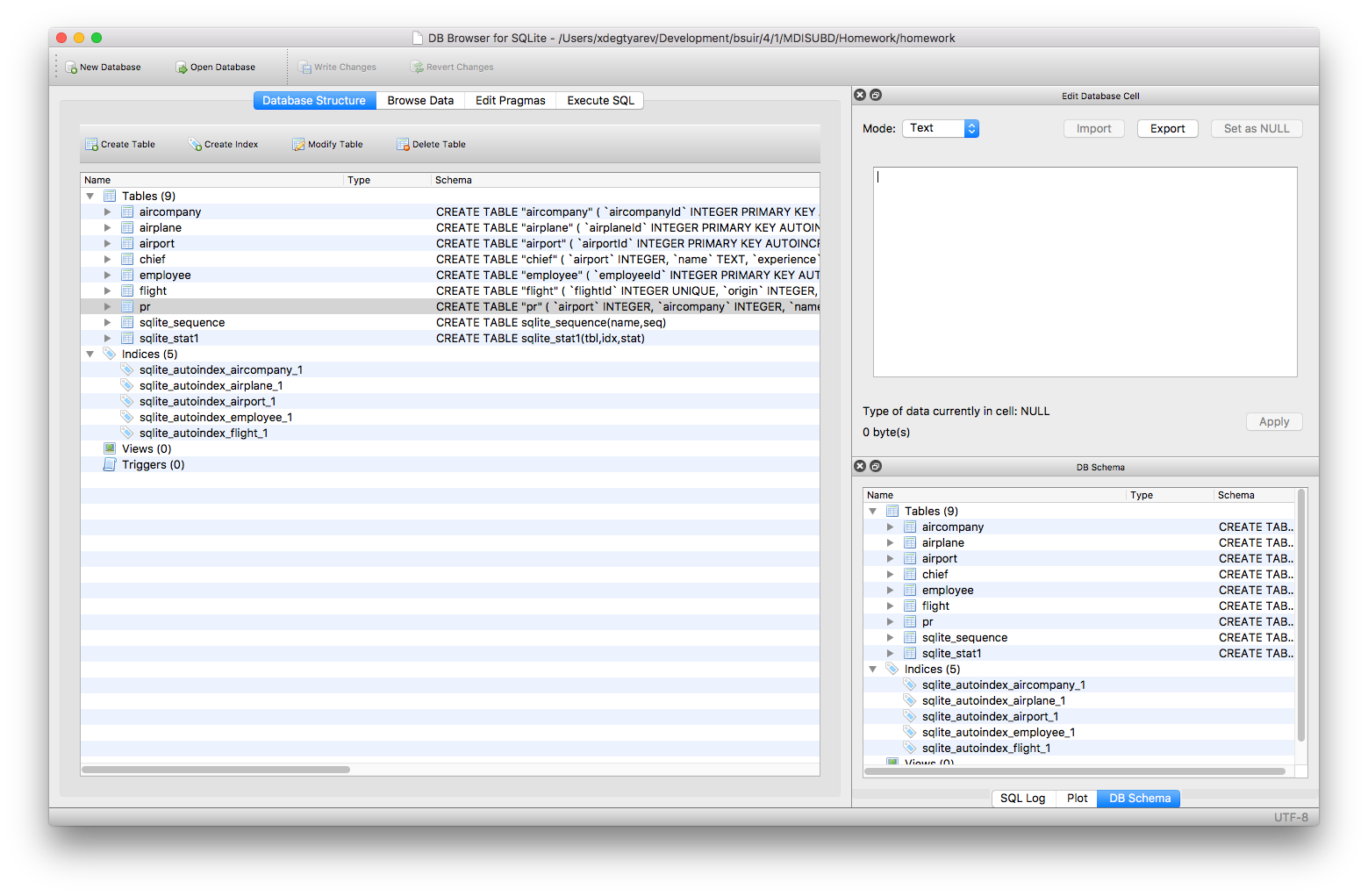
Минск 2017

*Во всех работах ниже я выполнял вариант №1 (26 mod 5 = 1)*

**Контрольная работа №1**

Вместо среды SQL\*Plus использовалась sqlite3 из терминала и с визуализатором DB Browser. sqlite – локальная реляционная бд, не требующая установления подключения к серверу. Доступна из одной библиотеки sqlite.lib

Использовалась база из первой части предмета МДиСУБД



**SQL: подзапросы, сложные запросы**

**Общая постановка задачи:**

Применить подзапросы в операторе SELECT для построения сложных выборок

|  |
| --- |
| 1. Вывести сотрудников, которые были приняты на работу после принятия на работу сотрудника, имя которого “Ivan Petrov”. |
| select \* from employee where hiredate > (select hireDate from employee where name like 'Petrov'); |
| 2. Найти должность сотрудника, у которой средняя заработная плата такая же, как и у всех сотрудников из таблицы Employee. |
| select job from employee where salary = (select Avg(salary) from employee); |
| 3. Вывести название представительства авиакомпании, в котором работают только лица женского пола и возрастом до 30 лет. |
| select name from pr where prId = (select pr from employee where sex like 'f' and age < 30); |
| 4. Вывести имя и номер социального страхования сотрудника , у которого заработная плата меньше средней заработной платы сотрудников таблицы Employee и возраст превышает средний возраст сотрудников. |
| select name, ssn from employee where salary < (select Avg(salary) from employee) and age > (select Avg(age) from employee); |
| 5. Вывести сотрудников, у которых заработная плата отличается от заработной платы любого из сотрудников должности “manager”. |
| select \* from employee where salary != (select salary from employee where job like 'manager'); |
| 6. Вывести имя и номер социального страхования сотрудников, которые были приняты на работу в один и тот же день и получают одинаковую заработную плату с сотрудником, который занимает должность “manager”. |
| select name, ssn from employee where hireDate = (select hireDate from employee where job like 'manager') and salary = (select salary from employee where job like 'manager'); |
| 7. Вывести имя, возраст, пол и дату приема на работу сотрудника, у которого заработная плата превышает среднюю заработную плату сотрудников таблицы Employee. |
| select name, age, sex, hireDate from employee where salary > (select Avg(salary) from employee); |

**Практическая работа №1**

**PL/SQL: переменные, анонимный блок**

**Цель:** Целью работы является изучение основных элементов программы на PL/SQL, приобретение практических навыков написания простейших программ, использующих выборку данных из таблиц БД.

**Общая постановка задачи:** Применить основные элементы программы на PL/SQL, написать простейшие программы, использующие выборку данных из таблиц БД.

|  |
| --- |
| 1. Создайте и выполните пустой анонимный блок. |
| DECLARE  BEGIN  END; |
| 1. Добавьте в блок команду вывода строки «This is my first string» |
| BEGIN  dbms\_output.put\_line('My first message');  END; |
| 1. Добавьте декларативную секцию и объявите переменную just\_a\_number числового типа, присвоив ей значение по умолчанию 10; объявите константу типа строка, присвоив ей значение “Hello world!”. |
| DECLARE  just\_a\_number integer := 10;  message constant varchar2(20) := 'Hello world!';  BEGIN  dbms\_output.put\_line(message || just\_a\_number);  END; |
| 1. Объявите следующие переменные: две переменные с типом столбца Name таблицы Employee, две переменные с типом столбца HireDate таблицы Employee , переменную-запись с типом строки таблицы Employee. |
| DECLARE  employee\_name employee.name%type;  other\_employee\_name employee.name%type;  employee\_hire\_date employee.hire%type;  other\_employee\_hire\_date employee.hire%type;  employee\_rec employee%rowtype;  BEGIN;  END; |
| 1. С помощью неявного курсора получите в переменные c типом столбца Name и HireDate таблицы Emloyee соответствующие значения из строки со значением поля ID равным 1 и 3, а в переменную с типом строки таблицы Employee — значение строки со значением поля ID равным 2. |
| DECLARE  employee\_name employee.name%type;  other\_employee\_name employee.name%type;  employee\_hire\_date employee.hire%type;  other\_employee\_hire\_date employee.hire%type;  employee\_rec employee%rowtype;  BEGIN  select name,hire into employee\_name, employee\_hire\_date from employee where employeeid = 1;  select name,hire into other\_employee\_name, other\_employee\_hire\_date from employee where employeeid = 3;  select \* into employee\_rec from employee where employeeid = 2;  END; |
| 1. Необходимо вывести следующую строку используя переменные из предыдущего задания :“Разница в трудоустройстве [имя\_1] и [имя\_2] составляет [дней]” Где вместо [имя\_1] требуется подставить соответствующее значение имени сотрудника, у которого ID равен 1; [имя\_2] следует подставить имя сотрудника, у которого ID равен 3. А вместо [дней] вычислить и вывести количество дней, на сколько один сотрудник работает больше или меньше другого. |
| DECLARE  employee\_name employee.name%type;  other\_employee\_name employee.name%type;  employee\_hire\_date employee.hire%type;  other\_employee\_hire\_date employee.hire%type;  employee\_rec employee%rowtype;  BEGIN  select name,hire into employee\_name, employee\_hire\_date from employee where employeeid = 1;  select name,hire into other\_employee\_name, other\_employee\_hire\_date from employee where employeeid = 3;  dbms\_output.put\_line('Diff in hire terms of ' || employee\_name || ' and ' || other\_employee\_name || ' is ' || (employee\_hire\_date - other\_employee\_hire\_date ) || ' days ');  END; |
| 1. Как осуществить действия пунктов 5-6 с помощью только SQL-запроса? |
| select name from employee where employeeid = 1;  select name from employee where employeeid = 3;  select hire - (select hire from employee where employeeid = 3) from employee where employeeid = 1; |
| 1. Требуется удалить сотрудников, у кого заработная плата составляет менее 100 у.е. Выполните внутри блока DELETE таблицы Employee. |
| DECLARE  minsalary int := 100;  BEGIN  delete from employee where salary<minsalary;  END; |
| 1. Добавьте команду, которая выводит количество строк, обработанных запросом из п.8. |
| select \* from employee;  insert into employee (employeeid,salary) values (10,90);  insert into employee (employeeid,salary) values (11,80);  select \* from employee;  DECLARE  opcount int := 0;  minsalary int := 100;  cursor low\_wage\_employees is  select employeeid from employee where salary<minsalary;  BEGIN  for n in low\_wage\_employees loop  opcount := opcount + 1;  delete from employee where employeeid = n.employeeid;  end loop;  dbms\_output.put\_line('removed ' || opcount || ' entries');  END; |

**Практическая работа №2**

**PL/SQL: курсоры, циклы, условия.**

**Цель:** Целью работы является изучение основных управляющих конструкций программы на PL/SQL, приобретение практических навыков написания программ, использующих ветвление и циклическую обработку и последовательное извлечение данных многострочных выборок с использованием явного курсора.

**Общая постановка задачи:** Применить основные управляющие конструкций программы на PL/SQL, написать программы, использующие ветвление и циклическую обработку и последовательное извлечение данных многострочных выборок с использованием явного курсора.

|  |
| --- |
| 1. Напишите анонимный блок, вычисляющий значение на основании значений строки, задаваемой значением первичного ключа, таблицы Employee в соответствии с приведенным алгоритмом: если работник проработал менее 10 лет, вывести сообщение “[Name] - young employee ” . Если же сотрудник проработал 10 и более лет, вывести сообщение “[Name] – old employee”. Значение первичного ключа запросите через подстановочную переменную. Выведите на экран полученную строку и стаж сотрудника в формате “[кол\_во\_дней]:[месяцев]:[лет]”. |
| select \* from employee;  DECLARE  employee\_id employee.employeeid%type := 1;  work\_exp\_arg constant int := 10;  employee\_work\_exp int;  employee\_hire\_date employee.hire%type;  employee\_name employee.name%type;  current\_d date:=CURRENT\_TIMESTAMP();  BEGIN  select name,hire into employee\_name, employee\_hire\_date from employee where employeeid = employee\_id;  employee\_work\_exp := current\_d-employee\_hire\_date;  IF( employee\_work\_exp < work\_exp\_arg ) THEN  dbms\_output.put\_line(employee\_name || ' young employee');  ELSE  dbms\_output.put\_line(employee\_name || ' old employee');  END IF;  dbms\_output.put\_line('DAYS:' || employee\_work\_exp || 'MONTHS: ' || employee\_work\_exp/31 || 'YEARS: ' || employee\_work\_exp/365);  END; |
| 1. Используя явный курсор, перепишите блок так, чтобы значение вычислялось последовательно для строк, в которых возраст сотрудника составляет подстановочную переменную. Используйте явное открытие/закрытие/выборку данных из курсора. |
| select \* from employee;  DECLARE  employee\_age employee.age%type := 22;  cursor employeeofspecificage is select name,age from employee where age = employee\_age;  e\_name employee.name%type;  e\_age employee.age%type;  BEGIN  open employeeofspecificage;  FETCH employeeofspecificage INTO e\_name, e\_age;  dbms\_output.put\_line('EMP:' || e\_name || ' age: ' || e\_age);  FETCH employeeofspecificage INTO e\_name, e\_age;  dbms\_output.put\_line('EMP:' || e\_name || ' age: ' || e\_age);  close employeeofspecificage;  END; |
| 1. Перепишите блок, используя курсорный цикл FOR. |
| select \* from employee;  DECLARE  employee\_age employee.age%type := 22;  cursor employeeofspecificage is select \* from employee where age = employee\_age;  BEGIN  for n in employeeofspecificage loop  dbms\_output.put\_line('EMP:' || n.name || ' age: ' || n.age);  end loop;  END; |