Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра вычислительных машин систем и сетей

Дисциплина: Базы данных

**ОТЧЕТ**

по лабораторной работе №6

(вариант 2)

Студент: гр. 450504

Павец Н. К.

Руководитель: старший преподаватель

Калабухов Е. В.

**Проект**

Интернет-магазин компьютерной техники

**Сущности**

* City
* Country
* Brand
* Notebook
* Customer
* Image
* Provider

**Схема HR**

Задание 1: В отдел “IT” добавить трех новых сотрудников. Для задания значения поля employee\_id создать и использовать последовательность (объект sequence), настроенную на генерацию значений от наименьшего номера сотрудника с шагом -10. Значение поля job\_id задать как “PU\_CLERK”, значение salary должно быть задано максимальной реально выплачиваемой зарплатой в этой должности, остальные значения требуется задать самостоятельно.

Описание создания последовательности:

*DROP SEQUENCE employees\_seq;*

*DECLARE*

*e\_id NUMBER;*

*BEGIN*

*SELECT MIN(employee\_id) INTO e\_id FROM employees;*

*EXECUTE immediate '*

*CREATE SEQUENCE employees\_seq*

*INCREMENT BY -10*

*START WITH ' || e\_id || '*

*MAXVALUE 999999999*

*MINVALUE 1*

*NOCACHE*

*NOCYCLE';*

*END;*

Вставка данных:

*INSERT INTO employees e (*

*employee\_id,*

*first\_name,*

*last\_name,*

*email,*

*phone\_number,*

*hire\_date,*

*job\_id,*

*salary,*

*commission\_pct,*

*manager\_id,*

*department\_id*

*)*

*VALUES (*

*employees\_seq.nextval,*

*'Nikita',*

*'Pavets',*

*'nikitapavets@gmail.com',*

*'375335556677',*

*TO\_DATE('2017/12/06 01:46:59', 'yyyy/mm/dd hh24:mi:ss'),*

*'PU\_CLERK',*

*(*

*SELECT MAX(salary)*

*FROM employees*

*WHERE job\_id = 'PU\_CLERK'*

*),*

*0.3,*

*100,*

*(*

*SELECT department\_id*

*FROM departments*

*WHERE department\_name= 'IT'*

*)*

*);*

*COMMIT;*

*INSERT INTO employees e (*

*employee\_id,*

*first\_name,*

*last\_name,*

*email,*

*phone\_number,*

*hire\_date,*

*job\_id,*

*salary,*

*commission\_pct,*

*manager\_id,*

*department\_id*

*)*

*VALUES (*

*employees\_seq.nextval,*

*'Nikita',*

*'Pavets',*

*'nikita@gmail.com',*

*'375335556677',*

*TO\_DATE('2017/12/06 01:46:59', 'yyyy/mm/dd hh24:mi:ss'),*

*'PU\_CLERK',*

*(*

*SELECT MAX(salary)*

*FROM employees*

*WHERE job\_id = 'PU\_CLERK'*

*),*

*0.3,*

*100,*

*(*

*SELECT department\_id*

*FROM departments*

*WHERE department\_name= 'IT'*

*)*

*);*

*COMMIT;*

*INSERT INTO employees e (*

*employee\_id,*

*first\_name,*

*last\_name,*

*email,*

*phone\_number,*

*hire\_date,*

*job\_id,*

*salary,*

*commission\_pct,*

*manager\_id,*

*department\_id*

*)*

*VALUES (*

*employees\_seq.nextval,*

*'Nikita',*

*'Pavets',*

*'pavets@gmail.com',*

*'375335556677',*

*TO\_DATE('2017/12/06 01:46:59', 'yyyy/mm/dd hh24:mi:ss'),*

*'PU\_CLERK',*

*(*

*SELECT MAX(salary)*

*FROM employees*

*WHERE job\_id = 'PU\_CLERK'*

*),*

*0.3,*

*100,*

*(*

*SELECT department\_id*

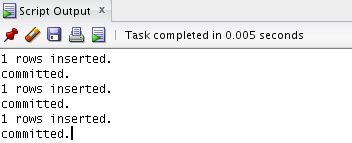
*FROM departments*

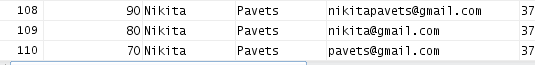
*WHERE department\_name= 'IT'*

*)*

*);*

*COMMIT;*





Задание 2: Сотрудникам всех отделов с числом сотрудников менее 4 требуется заполнить поле email в формате NLF (все буквы в верхнем регистре), где N – номер отдела, L –фамилия, F – первая буква имени.

*UPDATE employees*

*SET email = UPPER(CONCAT(*

*CONCAT(department\_id, last\_name),*

*SUBSTR(first\_name, 0, 1)*

*))*

*WHERE department\_id IN (*

*SELECT department\_id*

*FROM (*

*SELECT COUNT(\*) as department\_size, department\_id*

*FROM employees*

*INNER JOIN departments using (department\_id)*

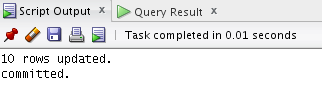
*GROUP BY department\_id*

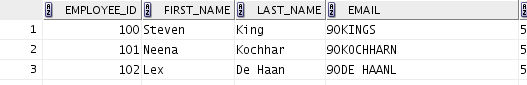
*)*

*WHERE department\_size < 4*

*);*

*COMMIT;*





Задание 3: Создать в схеме дополнительную пустую таблицу DEL\_DEPS для хранения данных о должностях. Из таблицы отделов DEPARTMENTS удалить все отделы, которые не имеют сотрудников. В новую таблицу DEL\_DEPS должны записаться все эти удаленные отделы. Для реализации последнего требования желательно создать соответствующий триггер (объект trigger).

Создание таблицы DEL\_DEPS:

CREATE TABLE DEL\_DEPS

(

DEPARTMENT\_ID NUMBER(4,0) NOT NULL,

DEPARTMENT\_NAME VARCHAR2(30 BYTE) NOT NULL,

MANAGER\_ID NUMBER(6,0),

LOCATION\_ID NUMBER(4,0),

CONSTRAINT DEL\_DEPS\_ID\_PK

PRIMARY KEY (DEPARTMENT\_ID),

CONSTRAINT DEL\_DEPT\_LOC\_FK

FOREIGN KEY (LOCATION\_ID)

REFERENCES LOCATIONS,

CONSTRAINT DEL\_DEPT\_MGR\_FK

FOREIGN KEY (MANAGER\_ID)

REFERENCES EMPLOYEES

);

Создание триггера:

CREATE OR REPLACE TRIGGER delete\_department

AFTER DELETE ON departments

FOR EACH ROW

BEGIN

INSERT INTO del\_deps

(

department\_id,

department\_name,

manager\_id,

location\_id

)

VALUES

(

:old.department\_id,

:old.department\_name,

:old.manager\_id,

:old.location\_id

);

END;

Выполнение удаления:

DELETE FROM departments

WHERE department\_id IN

(

SELECT department\_id

FROM departments

LEFT JOIN employees USING (department\_id)

WHERE NOT department\_id IN

(

SELECT department\_id

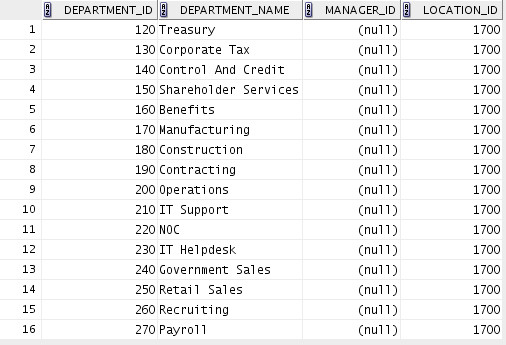
FROM departments

INNER JOIN employees USING (department\_id)

)

);

COMMIT;



**Своя Схема**

Задание 1: Добавить трех новых пользователей. Для задания значения поля id создать и использовать последовательность (объект sequence), настроенную на генерацию значений от наименьшего номера пользователя с шагом -10. Значение city\_id должно быть задано id города.

Описание создания последовательности:

*DROP SEQUENCE customers\_seq;*

*DECLARE*

*e\_id NUMBER;*

*BEGIN*

*SELECT MIN(id) INTO e\_id FROM customers;*

*EXECUTE immediate '*

*CREATE SEQUENCE customers\_seq*

*INCREMENT BY -10*

*START WITH ' || e\_id || '*

*MAXVALUE 999999999*

*MINVALUE 1*

*NOCACHE*

*NOCYCLE';*

*END;*

Вставка данных:

*INSERT INTO customers (*

*id,*

*first\_name,*

*last\_name,*

*email,*

*password,*

*city\_id*

*)*

*VALUES (*

*customers\_seq.nextval,*

*'Nikita',*

*'Pavets',*

*'nikitapavets@gmail.com',*

*'secret',*

*(*

*SELECT MAX(id)*

*FROM cities*

*)*

*);*

*COMMIT;*

*INSERT INTO customers (*

*id,*

*first\_name,*

*last\_name,*

*email,*

*password,*

*city\_id*

*)*

*VALUES (*

*customers\_seq.nextval,*

*'Nikita',*

*'Pavets',*

*'nikita@gmail.com',*

*'secret',*

*(*

*SELECT MAX(id)*

*FROM cities*

*)*

*);*

*COMMIT;*

*INSERT INTO customers (*

*id,*

*first\_name,*

*last\_name,*

*email,*

*password,*

*city\_id*

*)*

*VALUES (*

*customers\_seq.nextval,*

*'Nikita',*

*'Pavets',*

*'pavets@gmail.com',*

*'secret',*

*(*

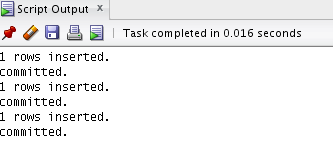
*SELECT MAX(id)*

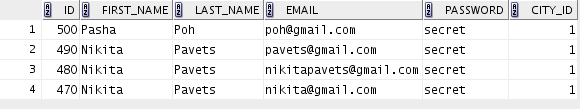
*FROM cities*

*)*

*);*

*COMMIT;*





Задание 2: Поставщикам с числом поставок более 5 требуется заполнить поле priority в формате 10 + общее число поставок.

*UPDATE providers*

*SET priority = 10 + (*

*SELECT SUM(notebooks\_size)*

*FROM (*

*SELECT COUNT(\*) as notebooks\_size, p.id as provider\_id*

*FROM notebooks n*

*INNER JOIN providers p ON p.id = n.provider\_id*

*GROUP BY n.provider\_id*

*)*

*WHERE notebooks\_size > 5*

*)*

*WHERE provider\_id IN (*

*SELECT provider\_id*

*FROM (*

*SELECT COUNT(\*) as notebooks\_size, p.id as provider\_id*

*FROM notebooks n*

*INNER JOIN providers p ON p.id = n.provider\_id*

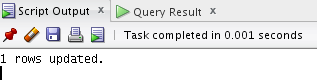
*GROUP BY n.provider\_id*

*)*

*WHERE notebooks\_size > 5*

*);*

*COMMIT;*





Задание 3: Создать в схеме дополнительную пустую таблицу DEL\_PROVIDERS для хранения данных о поставщиках. Из таблицы поставщиков PROVIDERS удалить всех поставщиков, которые не поставляли товар. В новую таблицу DEL\_PROVIDERS должны записаться все эти удаленные поставщики. Для реализации последнего требования желательно создать соответствующий триггер (объект trigger).

Создание таблицы DEL\_PROVIDERS:

CREATE TABLE DEL\_PROVIDERS

(

ID NUMBER(9) NOT NULL,

CODE NUMBER(9) NOT NULL,

PHONE VARCHAR2(15) NOT NULL,

PRIORITY NUMBER(3) NOT NULL,

CONSTRAINT DEL\_PROVIDER\_CODE\_UK UNIQUE (CODE),

CONSTRAINT DEL\_PROVIDER\_ID\_PK PRIMARY KEY (ID)

);

Создание триггера:

CREATE OR REPLACE TRIGGER delete\_provider

AFTER DELETE ON providers

FOR EACH ROW

BEGIN

INSERT INTO del\_providers

(

id,

code,

phone,

priority

)

VALUES

(

:old.id,

:old.code,

:old.phone,

:old.priority

)

END;

Выполнение удаления:

DELETE FROM providers

WHERE provider\_id IN

(

SELECT p.id as provider\_id

FROM providers p

LEFT JOIN customers c ON p.id = c.provider\_id

WHERE NOT provider\_id IN

(

SELECT p.id as provider\_id

FROM providers

INNER JOIN customers c ON p.id = c.provider\_id

)

);

COMMIT;

****