

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.

Минск, 2017

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;

```

Script Output x

Task completed in 0.005 seconds

1 rows inserted.
committed.
1 rows inserted.
committed.
1 rows inserted.
committed.

108	90	Nikita	Pavets	nikitapavets@gmail.com	37
109	80	Nikita	Pavets	nikita@gmail.com	37
110	70	Nikita	Pavets	pavets@gmail.com	37

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;
```

Script Output x Query Result x
Task completed in 0.01 seconds
10 rows updated.
committed.

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	
1	100	Steven	King	90KINGS	5
2	101	Neena	Kochhar	90K0CHHARN	5
3	102	Lex	De Haan	90DE HAANL	5

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;
```

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	120	Treasury	(null)	1700
2	130	Corporate Tax	(null)	1700
3	140	Control And Credit	(null)	1700
4	150	Shareholder Services	(null)	1700
5	160	Benefits	(null)	1700
6	170	Manufacturing	(null)	1700
7	180	Construction	(null)	1700
8	190	Contracting	(null)	1700
9	200	Operations	(null)	1700
10	210	IT Support	(null)	1700
11	220	NOC	(null)	1700
12	230	IT Helpdesk	(null)	1700
13	240	Government Sales	(null)	1700
14	250	Retail Sales	(null)	1700
15	260	Recruiting	(null)	1700
16	270	Payroll	(null)	1700

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;

```

Script Output x

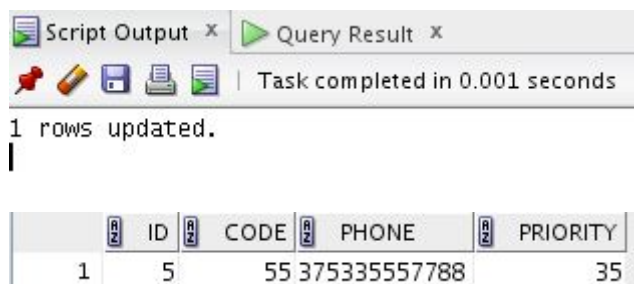
Task completed in 0.016 seconds

1 rows inserted.
committed.
1 rows inserted.
committed.
1 rows inserted.
committed.

	ID	FIRST_NAME	LAST_NAME	EMAIL	PASSWORD	CITY_ID
1	500	Pasha	Poh	poh@gmail.com	secret	1
2	490	Nikita	Pavets	pavets@gmail.com	secret	1
3	480	Nikita	Pavets	nikitapavets@gmail.com	secret	1
4	470	Nikita	Pavets	nikita@gmail.com	secret	1

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;
```



Script Output x Query Result x

Task completed in 0.001 seconds

1 rows updated.

	ID	CODE	PHONE	PRIORITY
1	5	55 375335557788		35

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;
```

	 ID	 CODE	 PHONE	 PRIORITY
1	1	11	375334785566	0