

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра защиты информации



Новосибирский
государственный
технический университет
НЭТИ

По лабораторной работе №3
по дисциплине: «Безопасность систем баз данных»
на тему «Работа с выводом данных»

Выполнил:
студент гр. АБ-120, АВТФ
Плешков К.А.

«__» _____ 2023 г.

(подпись)

Проверил:
ассистент кафедры ЗИ
Питько Я.А.

«__» _____ 20__ г.

(подпись)

Новосибирск
2023

Цель работы: получить практические навыки создания сводных таблиц на основе уже существующих, а также конструирования комплексных запросов для получения желаемой выборки данных.

Задачи:

1. Создать ветку от master(main) с названием “lab3”. Все изменения в рамках данной лабы производить в этой ветке.
2. Дополнить скрипт инициализации необходимой составной таблицей в соответствии со своим вариантом. В этом случае необходимо использовать конструкцию **CREATE TABLE AS**.
3. Продемонстрировать созданную составную таблицу в отчёте.
4. При необходимости, дополнить недостающими сущностями существующую структуру БД.
5. Убедиться, что данные в таблицах позволяют продемонстрировать работу SELECT-запросов. Если это не так - скорректировать данные.
6. Продемонстрировать работу запросов SELECT в отчёте.
7. После окончания работы, создать merge request ветки с лабораторной работой в master(main), добавить преподавателя в reviewer. Не сливать без аппрува от преподавателя!
8. В Диспее прикрепить ссылку на репозиторий и на отчёт в *.pdf формате.

Вариант – Курьерская служба.

Составная таблица: Список заказов по пунктам и работников, которые обрабатывали заказы.

SELECT-запросы:

Вывести все пункты, обслуживающие крупногабаритные грузы и с эффективной площадью хранения больше 400.

Вывести всех работников определённого пункта и определённой должности.

Вывести всех пеших-курьеров старше 30 лет

Ход работы:

Перед созданием составной таблицы, нужно связать таблицы: Employees, Points и Orders. Свяжем их через таблицу Orders, добавив данные строки:

```
employee_id INT NOT NULL,  
point_id INT NOT NULL,  
FOREIGN KEY (employee_id) REFERENCES Employees (employee_id),  
FOREIGN KEY (point_id) REFERENCES Points (point_id)
```

После этого мы можем создать составную таблицу. Для этого воспользуемся конструкцией CREATETABLE AS.

```
--Составная таблица  
CREATE TABLE Orders_by_points_and_employees(order_id, employee_full_name, point_address) AS  
SELECT o.order_id, e.full_name, p.address FROM Orders o  
JOIN Employees e ON o.employee_id = e.employee_id  
JOIN Points p ON o.point_id = p.point_id;
```

Рисунок 1 – Создание составной таблицы.

Добавим ограничения на таблицу:

```
--Ограничения на составную таблицу  
ALTER TABLE Orders_by_points_and_employees ADD PRIMARY KEY (order_id);  
ALTER TABLE Orders_by_points_and_employees ADD FOREIGN KEY (order_id) references Orders on delete cascade;
```

Рисунок 2 – Ограничения таблицы.

Посмотрим в pgAdmin на полученную таблицу:

	order_id [PK] integer	employee_full_name character varying (60)	point_address character varying (255)
1	1	Анна Сидорова	Московская улица, 1
2	2	Анна Сидорова	Ленинградский проспект, 2
3	3	Александр Иванов	Сибирский тракт, 3
4	4	Анна Сидорова	Сибирский тракт, 3
5	5	Марина Смирнова	Невский проспект, 5
6	6	Екатерина Козлова	Сибирский тракт, 3
7	7	Игорь Павлов	Московская улица, 1
8	8	Александр Иванов	Московская улица, 1
9	9	Екатерина Козлова	Ленинградский проспект, 2
10	10	Марина Смирнова	Сибирский тракт, 3

Рисунок 3 – Составная таблица.

Составим SELECT-запросы и посмотрим на вывод:

1. Вывести все пункты, обслуживающие крупногабаритные грузы и с эффективной площадью хранения больше 400.

sodbms-postgres-1	point_id	address	is_oversized_cargo_service	phone_number	effective_storage_area
sodbms-postgres-1	5	Невский проспект, 5	t	56789012345	1200.8
sodbms-postgres-1	8	Казанская улица, 8	t	89012345678	1600.9
sodbms-postgres-1	9	Омская улица, 9	t	90123456789	1100.4
sodbms-postgres-1	(3 rows)				

```
178
179 --Вывести все пункты, обслуживающие крупногабаритные грузы и с эффективной площадью хранения больше 400.
180 SELECT p.point_id, p.address, p.is_oversized_cargo_service, pn.phone_number, p.effective_storage_area
181 FROM Points p
182 JOIN PhoneNumbers pn ON p.phone_number_id = pn.phone_number_id
183 WHERE is_oversized_cargo_service = true AND effective_storage_area > 400;
```

Рисунок 4 – Первый SELECT-запрос.

2. Вывести всех работников определённого пункта и определённой должности.

sodbms-postgres-1	employee_id	full_name	location	position
sodbms-postgres-1	1	Анна Сидорова	Москва	Менеджер
sodbms-postgres-1	10	Татьяна Николаева	Москва	Менеджер
sodbms-postgres-1	(2 rows)			

```
--Вывести всех работников определённого пункта и определённой должности.
SELECT e.employee_id, e.full_name, e.location, p.position
FROM Employees e
JOIN Positions p ON e.position_id = p.position_id
WHERE e.location = 'Москва' AND p.position = 'Менеджер';
```

Рисунок 5 – Второй SELECT-запрос.

3. Вывести всех пеших-курьеров старше 30 лет

sodbms-postgres-1	courier_id	full_name	birth_date	age	method_of_travel	phone_number
sodbms-postgres-1	4	Мария Ковалева	1992-07-20	31	Пеший	45678901234
sodbms-postgres-1	7	Дмитрий Соколов	1987-12-05	35	Пеший	78901234567
sodbms-postgres-1	(2 rows)					

```
--Вывести всех пеших-курьеров старше 30 лет.
SELECT c.courier_id, c.full_name, c.birth_date, date_part('year', AGE(c.birth_date)) AS age, t.method_of_travel, pn.phone_number
FROM Couriers c
JOIN TravelMethods t ON c.method_of_travel_id = t.method_of_travel_id
JOIN PhoneNumbers pn ON c.phone_number_id = pn.phone_number_id
WHERE t.method_of_travel = 'Пеший' AND date_part('year', AGE(c.birth_date)) > 30;
```

Рисунок 6 – Третий SELECT-запрос.

Вывод: проделав данную лабораторную работу, были получены практические навыки создания составных таблиц с помощью конструкции `CREATE TABLE AS` и составления `SELECT`-запросов.

ПРИЛОЖЕНИЕ

Листинг Pleshkov.sql:

```
CREATE TABLE TravelMethods (  
    method_of_travel_id SERIAL PRIMARY KEY,  
    method_of_travel VARCHAR(30) NOT NULL  
);  
  
CREATE TABLE Positions (  
    position_id SERIAL PRIMARY KEY,  
    position VARCHAR(60) NOT NULL  
);  
  
CREATE TABLE PhoneNumbers (  
    phone_number_id SERIAL PRIMARY KEY,  
    phone_number VARCHAR(11) NOT NULL  
);  
  
CREATE TABLE Couriers (  
    courier_id SERIAL PRIMARY KEY,  
    full_name VARCHAR(60) NOT NULL,  
    birth_date DATE NOT NULL,  
    method_of_travel_id INT NOT NULL,  
    phone_number_id INT NOT NULL,  
    FOREIGN KEY (method_of_travel_id) REFERENCES TravelMethods  
(method_of_travel_id),  
    FOREIGN KEY (phone_number_id) REFERENCES PhoneNumbers (phone_number_id)  
);  
  
CREATE TABLE Employees (  
    employee_id SERIAL PRIMARY KEY,  
    full_name VARCHAR(60) NOT NULL,  
    location VARCHAR(80) NOT NULL,  
    position_id INT NOT NULL,  
    FOREIGN KEY (position_id) REFERENCES Positions (position_id)  
);  
  
CREATE TABLE Points (  
    point_id SERIAL PRIMARY KEY,  
    address VARCHAR(255) NOT NULL,  
    is_oversized_cargo_service BOOLEAN NOT NULL,  
    phone_number_id INT NOT NULL,  
    effective_storage_area DECIMAL(8,1) NOT NULL,  
    FOREIGN KEY (phone_number_id) REFERENCES PhoneNumbers (phone_number_id)  
);  
  
CREATE TABLE Orders (  
    order_id SERIAL PRIMARY KEY,  
    order_date DATE NOT NULL,  
    distance_to_addressee DECIMAL(8,1) NOT NULL,  
    employee_id INT NOT NULL,  
    point_id INT NOT NULL,  
    FOREIGN KEY (employee_id) REFERENCES Employees (employee_id),  
    FOREIGN KEY (point_id) REFERENCES Points (point_id)  
);  
  
INSERT INTO TravelMethods (method_of_travel)  
VALUES  
    ('Автомобиль'),  
    ('Самолет'),
```

```

('Пеший');

INSERT INTO Positions (position)
VALUES
    ('Менеджер'),
    ('Продавец'),
    ('Администратор');

INSERT INTO PhoneNumbers (phone_number)
VALUES
    ('12345678901'),
    ('23456789012'),
    ('34567890123'),
    ('45678901234'),
    ('56789012345'),
    ('67890123456'),
    ('78901234567'),
    ('89012345678'),
    ('90123456789'),
    ('01234567890');

INSERT INTO Couriers (full_name, birth_date, method_of_travel_id,
phone_number_id)
VALUES
    ('Иван Иванов', '1990-01-01', 1, 1),
    ('Алексей Петров', '1992-05-15', 2, 2),
    ('Елена Смирнова', '1985-11-30', 1, 3),
    ('Мария Ковалева', '1992-07-20', 3, 4),
    ('Андрей Васильев', '1988-03-10', 2, 5),
    ('Ольга Николаева', '1991-09-25', 1, 6),
    ('Дмитрий Соколов', '1987-12-05', 3, 7),
    ('Сергей Морозов', '1994-04-18', 2, 8),
    ('Наталья Волкова', '1989-08-12', 1, 9),
    ('Павел Лебедев', '1995-02-28', 3, 10);

INSERT INTO Employees (full_name, location, position_id)
VALUES
    ('Анна Сидорова', 'Москва', 1),
    ('Александр Иванов', 'Санкт-Петербург', 2),
    ('Екатерина Козлова', 'Екатеринбург', 3),
    ('Игорь Павлов', 'Новосибирск', 1),
    ('Марина Смирнова', 'Казань', 2),
    ('Артем Петров', 'Владивосток', 3),
    ('Ольга Морозова', 'Ростов-на-Дону', 1),
    ('Денис Васильев', 'Самара', 2),
    ('Николай Ковалев', 'Омск', 3),
    ('Татьяна Николаева', 'Москва', 1);

INSERT INTO Points (address, is_oversized_cargo_service, phone_number_id,
effective_storage_area)
VALUES
    ('Московская улица, 1', false, 1, 400.5),
    ('Ленинградский проспект, 2', true, 2, 150.2),
    ('Сибирский тракт, 3', false, 3, 800.7),
    ('Красная площадь, 4', true, 4, 200.1),
    ('Невский проспект, 5', true, 5, 1200.8),
    ('Уральская улица, 6', true, 6, 180.3),
    ('Волгоградский проспект, 7', false, 7, 90.6),
    ('Казанская улица, 8', true, 8, 1600.9),
    ('Омская улица, 9', true, 9, 1100.4),
    ('Тихоокеанская улица, 10', true, 10, 140.0);

INSERT INTO Orders (order_date, distance_to_addressee, employee_id, point_id)

```

```

VALUES
('2023-01-01', 10.5, 1, 1),
('2023-02-15', 5.2, 1, 2),
('2023-03-30', 8.7, 2, 3),
('2023-04-10', 12.3, 1, 3),
('2023-05-20', 6.8, 5, 5),
('2023-06-05', 9.1, 3, 3),
('2023-07-18', 7.4, 4, 1),
('2023-08-25', 11.2, 2, 1),
('2023-09-10', 4.9, 3, 2),
('2023-10-28', 14.6, 5, 3);

-- Задание 1 добавляем столбец полный ли рабочий день у курьера
ALTER TABLE Couriers
ADD COLUMN is_full_time_working BOOLEAN;

-- Задание 2 меняем должность работников
UPDATE Employees
SET location = 'Казань'
WHERE location = 'Ленск';

-- Задание 3 удаляем столбец
ALTER TABLE Couriers
DROP COLUMN is_full_time_working;

-- Задание 4 удаляем пункт по адресу
DELETE FROM Points
WHERE address = 'Тихоокеанская улица, 10';

--Ограничения

--Ограничение на дату рождения
ALTER TABLE Couriers
ADD CONSTRAINT check_birth_date CHECK (birth_date >= '1900-01-01' AND
birth_date <= CURRENT_DATE);

--Ограничение на дату заказа
ALTER TABLE Orders
ADD CONSTRAINT order_date CHECK (order_date <= CURRENT_DATE);

--Ограничение на расстояние до адресата
ALTER TABLE Orders
ADD CONSTRAINT distance_to_addressee CHECK (distance_to_addressee > 0);

--Ограничение на эффективную площадь хранения
ALTER TABLE Points
ADD CONSTRAINT effective_storage_area CHECK (effective_storage_area > 0);

--Составная таблица
CREATE TABLE Orders_by_points_and_employees(order_id, employee_full_name,
point_address) AS
SELECT o.order_id, e.full_name, p.address FROM Orders o
JOIN Employees e ON o.employee_id = e.employee_id
JOIN Points p ON o.point_id = p.point_id;

--Ограничения на составную таблицу
ALTER TABLE Orders_by_points_and_employees ADD PRIMARY KEY (order_id);
ALTER TABLE Orders_by_points_and_employees ADD FOREIGN KEY (order_id)
references Orders on delete cascade;

```



```

--Вывести все пункты, обслуживающие крупногабаритные грузы и с эффективной
площадью хранения больше 400.
SELECT p.point_id, p.address, p.is_oversized_cargo_service, pn.phone_number,
p.effective_storage_area
FROM Points p
JOIN PhoneNumbers pn ON p.phone_number_id = pn.phone_number_id
WHERE is_oversized_cargo_service = true AND effective_storage_area > 400;

--Вывести всех работников определённого пункта и определённой должности.
SELECT e.employee_id, e.full_name, e.location, p.position
FROM Employees e
JOIN Positions p ON e.position_id = p.position_id
WHERE e.location = 'Москва' AND p.position = 'Менеджер';

--Вывести всех пеших-курьеров старше 30 лет.
SELECT c.courier_id, c.full_name, c.birth_date, date_part('year',
AGE(c.birth_date)) AS age, t.method_of_travel, pn.phone_number
FROM Couriers c
JOIN TravelMethods t ON c.method_of_travel_id = t.method_of_travel_id
JOIN PhoneNumbers pn ON c.phone_number_id = pn.phone_number_id
WHERE t.method_of_travel = 'Пеший' AND date_part('year', AGE(c.birth_date))
> 30;

```