Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №3 «Запросы на выборку и модификацию данных, представления и индексы в PostgreSQL»

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Фадеев Дмитрий Алексиевич

Факультет: ИКТ

Группа: К3239

Преподаватель: Говорова М.М.

Санкт-Петербург 2023

Оглавление

Цель работы 3
Практическое задание 3
Вариант 9. БД «Оптовая база» 3
Выполнение работы: 4
Создание схемы: 4
Создание таблицы: 4
Создание ограничений: 11
Заполнение рабочими данными: 17
Создание бэкапа: 17
Восстановление БД: 18
Вывод: 18

Цель работы

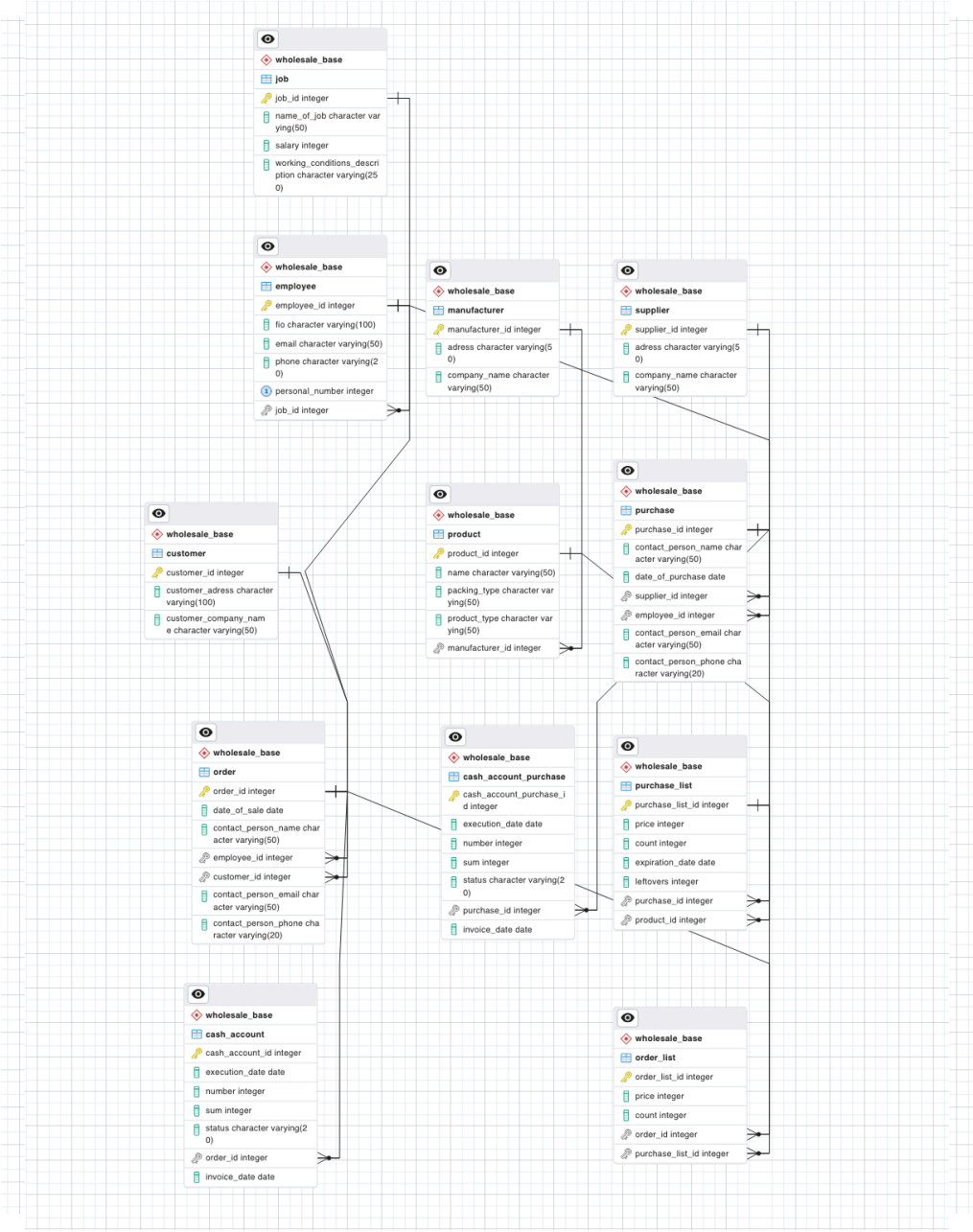
Овладеть практическими навыками создания таблиц базы данных Postrgresql 1X, заполнения рабочими данными, резервного копирования и восстановления БД.

Практическое задание

1. Создать базу данных с использованием pgAdmin 4
2. Создать схему в составе базы данных.
3. Создать таблицы базы данных.
4. Установить ограничения на данные: Primary Key, Unique, Check, Foreign Key.
5. Заполнить таблицы БД рабочими данными.
6. Создать резервную копию БД
7. Восстановить БД

Вариант 9. БД «Оптовая база»

ERD БД:



Выполнение работы:

Для описания работы я использовал листинг из бекапа БД в формате plain

Создание схемы:

```
CREATE SCHEMA wholesale_base;


ALTER SCHEMA wholesale_base OWNER TO pg_database_owner;
```

Создание таблицы:

```
CREATE TABLE wholesale_base.cash_account (
    cash_account_id integer NOT NULL,
    date date NOT NULL,
    number integer NOT NULL,
    sum integer NOT NULL,
    status character varying(20) NOT NULL,
    order_id integer NOT NULL
);


ALTER TABLE wholesale_base.cash_account OWNER TO pg_database_owner;


ALTER TABLE wholesale_base.cash_account ALTER COLUMN cash_account_id ADD GENERATED ALWAYS
AS IDENTITY (
    SEQUENCE NAME wholesale_base.cash_account_cash_account_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);



CREATE TABLE wholesale_base.cash_account_purchase (
    cash_account_purchase_id integer NOT NULL,
    date date NOT NULL,
```

```
    number integer NOT NULL,
    sum integer NOT NULL,
    status character varying(20) NOT NULL,
    purchase_id integer NOT NULL
);


ALTER TABLE wholesale_base.cash_account_purchase OWNER TO pg_database_owner;

ALTER TABLE wholesale_base.cash_account_purchase ALTER COLUMN cash_account_purchase_id ADD
GENERATED ALWAYS AS IDENTITY (
    SEQUENCE NAME wholesale_base.cash_account_purchase_cash_account_purchase_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);

CREATE TABLE wholesale_base.customer (
    customer_id integer NOT NULL,
    customer_adress character varying(100) NOT NULL,
    customer_company_name character varying(50) NOT NULL
);


ALTER TABLE wholesale_base.customer OWNER TO pg_database_owner;

ALTER TABLE wholesale_base.customer ALTER COLUMN customer_id ADD GENERATED ALWAYS AS
IDENTITY (
    SEQUENCE NAME wholesale_base.customer_customer_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
```

```
    CACHE 1
);



CREATE TABLE wholesale_base.employee (
    employee_id integer NOT NULL,
    fio character varying(100) NOT NULL,
    email character varying(50),
    phone character varying(20),
    personal_number integer NOT NULL,
    job_id integer NOT NULL
);



ALTER TABLE wholesale_base.employee OWNER TO pg_database_owner;



ALTER TABLE wholesale_base.employee ALTER COLUMN employee_id ADD GENERATED ALWAYS AS
IDENTITY (
    SEQUENCE NAME wholesale_base.employee_employee_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);


CREATE TABLE wholesale_base.job (
    job_id integer NOT NULL,
    name_of_job character varying(50) NOT NULL,
    salary integer NOT NULL,
    working_conditions_description character varying(250)
);
```

```
ALTER TABLE wholesale_base.job OWNER TO pg_database_owner;


ALTER TABLE wholesale_base.job ALTER COLUMN job_id ADD GENERATED ALWAYS AS IDENTITY (
    SEQUENCE NAME wholesale_base.job_job_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);



CREATE TABLE wholesale_base.manufacturer (
    manufacturer_id integer NOT NULL,
    adress character varying(50) NOT NULL,
    company_name character varying(50) NOT NULL
);



ALTER TABLE wholesale_base.manufacturer OWNER TO pg_database_owner;


ALTER TABLE wholesale_base.manufacturer ALTER COLUMN manufacturer_id ADD GENERATED ALWAYS
AS IDENTITY (
    SEQUENCE NAME wholesale_base.manufacturer_manufacturer_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);


CREATE TABLE wholesale_base."order" (
    order_id integer NOT NULL,
    date_of_sale date NOT NULL,
    contact_person character varying(50) NOT NULL,
```

```
    employee_id integer NOT NULL,
    customer_id integer NOT NULL
);


ALTER TABLE wholesale_base."order" OWNER TO pg_database_owner;


CREATE TABLE wholesale_base.order_list (
    order_list_id integer NOT NULL,
    price integer NOT NULL,
    count integer NOT NULL,
    order_id integer NOT NULL,
    purchase_list_id integer NOT NULL
);


ALTER TABLE wholesale_base.order_list OWNER TO pg_database_owner;


ALTER TABLE wholesale_base.order_list ALTER COLUMN order_list_id ADD GENERATED ALWAYS AS
IDENTITY (
    SEQUENCE NAME wholesale_base.order_list_order_list_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);


ALTER TABLE wholesale_base."order" ALTER COLUMN order_id ADD GENERATED ALWAYS AS
IDENTITY (
    SEQUENCE NAME wholesale_base.order_order_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
```

```
    CACHE 1
);

CREATE TABLE wholesale_base.product (
    product_id integer NOT NULL,
    name character varying(50) NOT NULL,
    expiration_date date NOT NULL,
    packing_type character varying(50) NOT NULL,
    product_type character varying(50) NOT NULL,
    manufacturer_id integer NOT NULL
);



ALTER TABLE wholesale_base.product OWNER TO pg_database_owner;

ALTER TABLE wholesale_base.product ALTER COLUMN product_id ADD GENERATED ALWAYS AS
IDENTITY (
    SEQUENCE NAME wholesale_base.product_product_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);

CREATE TABLE wholesale_base.purchase (
    purchase_id integer NOT NULL,
    contact_person character varying(50) NOT NULL,
    date date NOT NULL,
    supplier_id integer NOT NULL,
    employee_id integer NOT NULL
);

ALTER TABLE wholesale_base.purchase OWNER TO pg_database_owner;
```

```
CREATE TABLE wholesale_base.purchase_list (
    purchase_list_id integer NOT NULL,
    price integer NOT NULL,
    count integer NOT NULL,
    manufacture_date date NOT NULL,
    leftovers integer NOT NULL,
    purchase_id integer NOT NULL,
    product_id integer NOT NULL
);



ALTER TABLE wholesale_base.purchase_list OWNER TO pg_database_owner;


ALTER TABLE wholesale_base.purchase_list ALTER COLUMN purchase_list_id ADD GENERATED ALWAYS
AS IDENTITY (
    SEQUENCE NAME wholesale_base.purchase_list_purchase_list_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);


ALTER TABLE wholesale_base.purchase ALTER COLUMN purchase_id ADD GENERATED ALWAYS AS
IDENTITY (
    SEQUENCE NAME wholesale_base.purchase_purchase_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);


CREATE TABLE wholesale_base.supplier (
    supplier_id integer NOT NULL,
```

```
    adress character varying(50) NOT NULL,
    company_name character varying(50) NOT NULL
);



ALTER TABLE wholesale_base.supplier OWNER TO pg_database_owner;


ALTER TABLE wholesale_base.supplier ALTER COLUMN supplier_id ADD GENERATED ALWAYS AS
IDENTITY (
    SEQUENCE NAME wholesale_base.supplier_supplier_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1
);
```

Создание ограничений:

```
ALTER TABLE wholesale_base.supplier
    ADD CONSTRAINT adress_contains_english_letters_or_numbers_check CHECK
(((adress)::text ~* '^[A-Za-z0-9\s]+$'::text)) NOT VALID;

ALTER TABLE wholesale_base.manufacturer
    ADD CONSTRAINT adress_latin_letters_numbers_spaces_check CHECK (((adress)::text ~*
'^[A-Za-z0-9._\s]+$'::text)) NOT VALID;

ALTER TABLE ONLY wholesale_base.cash_account
    ADD CONSTRAINT cash_account_pkey PRIMARY KEY (cash_account_id);

ALTER TABLE ONLY wholesale_base.cash_account_purchase
    ADD CONSTRAINT cash_account_purchase_pkey PRIMARY KEY
(cash_account_purchase_id);

ALTER TABLE wholesale_base.manufacturer
```

```
    ADD CONSTRAINT company_name_latin_letters_numbers_spaces_check CHECK
(((company_name)::text ~* '^[A-Za-z0-9_.\s]+$'::text)) NOT VALID;

ALTER TABLE wholesale_base.supplier

    ADD CONSTRAINT company_name_letters_are_english_check CHECK
(((company_name)::text ~* '^[A-Za-z0-9_.\s]+$'::text)) NOT VALID;

ALTER TABLE wholesale_base."order"

    ADD CONSTRAINT contact_person_latin_letters_spaces_check CHECK
(((contact_person)::text ~* '^[A-Za-z\s]+$'::text)) NOT VALID;

ALTER TABLE wholesale_base.purchase

    ADD CONSTRAINT contact_person_latin_letters_spaces_check CHECK
(((contact_person)::text ~* '^[A-Za-z\s]+$'::text)) NOT VALID;

ALTER TABLE wholesale_base.customer

    ADD CONSTRAINT customer_adress_latin_letters_numbers_spaces_check CHECK
(((customer_adress)::text ~* '^[A-Za-z0-9\s]+$'::text)) NOT VALID;

ALTER TABLE wholesale_base.customer

    ADD CONSTRAINT customer_company_name_latin_letters_numbers_spaces_check
CHECK (((customer_company_name)::text ~* '^[A-Za-z0-9_.\s]+$'::text)) NOT VALID;

ALTER TABLE ONLY wholesale_base.customer

    ADD CONSTRAINT customer_pkey PRIMARY KEY (customer_id);

ALTER TABLE wholesale_base.employee

    ADD CONSTRAINT email_validation CHECK (((email)::text ~* '^[a-zA-Z0-9._%+-]+@[a-
zA-Z0-9.-]+\.[a-zA-Z]{2,}$'::text)) NOT VALID;

ALTER TABLE ONLY wholesale_base.employee

    ADD CONSTRAINT employee_pkey PRIMARY KEY (employee_id);

ALTER TABLE wholesale_base.employee

    ADD CONSTRAINT fio_latin_letters_spaces_check CHECK (((fio)::text ~* '^[A-Za-z\s]+
$'::text)) NOT VALID;

ALTER TABLE ONLY wholesale_base.job

    ADD CONSTRAINT job_pkey PRIMARY KEY (job_id);
```

```
ALTER TABLE wholesale_base.purchase_list

   ADD CONSTRAINT leftovers_lower_or_equal_count CHECK ((leftovers <= count)) NOT
VALID;

ALTER TABLE ONLY wholesale_base.manufacturer

   ADD CONSTRAINT manufacturer_pkey PRIMARY KEY (manufacturer_id);

ALTER TABLE wholesale_base.product

   ADD CONSTRAINT name_latin_letters_numbers_spaces_check CHECK (((name)::text ~*
'^[A-Za-z0-9_\s]+$'::text)) NOT VALID;

ALTER TABLE wholesale_base.job

   ADD CONSTRAINT name_of_job_latin_letters_spaces_check CHECK (((name_of_job)::text
~* '^[A-Za-z\s]+$'::text)) NOT VALID;

ALTER TABLE ONLY wholesale_base.order_list

   ADD CONSTRAINT order_list_pkey PRIMARY KEY (order_list_id);

ALTER TABLE ONLY wholesale_base."order"

   ADD CONSTRAINT order_pkey PRIMARY KEY (order_id);

ALTER TABLE wholesale_base.product

   ADD CONSTRAINT packing_type_choice_restriction CHECK (((packing_type)::text = ANY
((ARRAY['Block packaging'::character varying, 'Box packaging'::character varying, 'Pallet
packaging'::character varying, 'Container packaging'::character varying, 'Bag
packaging'::character varying])::text[]))) NOT VALID;

ALTER TABLE wholesale_base.employee

   ADD CONSTRAINT phone_validation CHECK (((phone)::text ~ '^[0-9]{10}$'::text)) NOT
VALID;

ALTER TABLE wholesale_base.purchase_list

   ADD CONSTRAINT positive_count_check CHECK ((count >= 0)) NOT VALID;

ALTER TABLE wholesale_base.order_list

   ADD CONSTRAINT positive_count_check CHECK ((count >= 0)) NOT VALID;

ALTER TABLE wholesale_base.purchase_list
```

```
    ADD CONSTRAINT positive_leftovers_check CHECK ((leftovers >= 0)) NOT VALID;

ALTER TABLE wholesale_base.cash_account

    ADD CONSTRAINT positive_number_check CHECK ((number > 0)) NOT VALID;

ALTER TABLE wholesale_base.cash_account_purchase

    ADD CONSTRAINT positive_number_check CHECK ((number > 0)) NOT VALID;

ALTER TABLE wholesale_base.employee

    ADD CONSTRAINT positive_presonal_number_check CHECK ((personal_number > 0))
NOT VALID;

ALTER TABLE wholesale_base.purchase_list

    ADD CONSTRAINT positive_price_check CHECK ((price >= 0)) NOT VALID;

ALTER TABLE wholesale_base.order_list

    ADD CONSTRAINT positive_price_check CHECK ((price >= 0)) NOT VALID;

ALTER TABLE wholesale_base.job

    ADD CONSTRAINT positive_salary_check CHECK ((salary > 0)) NOT VALID;

ALTER TABLE wholesale_base.cash_account

    ADD CONSTRAINT positive_sum_check CHECK ((sum > 0)) NOT VALID;

ALTER TABLE wholesale_base.cash_account_purchase

    ADD CONSTRAINT positive_sum_check CHECK ((sum > 0)) NOT VALID;

ALTER TABLE ONLY wholesale_base.product

    ADD CONSTRAINT product_pkey PRIMARY KEY (product_id);

ALTER TABLE wholesale_base.product

    ADD CONSTRAINT product_type_choice_restriction CHECK (((product_type)::text = ANY
((ARRAY['Food and Beverages'::character varying, 'Industrial materials and
equipment'::character varying, 'Technical and electronic goods'::character varying, 'Clothing and
textiles'::character varying, 'Medical goods and equipment'::character varying, 'Construction
materials'::character varying, 'Automotive parts and accessories'::character varying, 'Stationery
supplies'::character varying])::text[]))) NOT VALID;

ALTER TABLE ONLY wholesale_base.purchase_list

    ADD CONSTRAINT purchase_list_pkey PRIMARY KEY (purchase_list_id);
```

```
ALTER TABLE ONLY wholesale_base.purchase

    ADD CONSTRAINT purchase_pkey PRIMARY KEY (purchase_id);

ALTER TABLE wholesale_base.cash_account

    ADD CONSTRAINT status_choice_restriction CHECK ((((status)::text = 'awaiting
payment'::text) OR ((status)::text = 'paid'::text) OR ((status)::text = 'rejected'::text))) NOT
VALID;

ALTER TABLE wholesale_base.cash_account_purchase

    ADD CONSTRAINT status_choice_restriction CHECK ((((status)::text = 'awaiting
payment'::text) OR ((status)::text = 'paid'::text) OR ((status)::text = 'rejected'::text))) NOT
VALID;

ALTER TABLE ONLY wholesale_base.supplier

    ADD CONSTRAINT supplier_pkey PRIMARY KEY (supplier_id);

ALTER TABLE ONLY wholesale_base.employee

    ADD CONSTRAINT unique_personal_number UNIQUE (personal_number) INCLUDE
(personal_number);

CREATE TRIGGER check_if_leftovers_enough_before_insert BEFORE INSERT ON
wholesale_base.order_list FOR EACH ROW EXECUTE FUNCTION
wholesale_base.check_if_leftovers_enough();

CREATE TRIGGER subtract_count_from_leftovers AFTER INSERT ON
wholesale_base.order_list FOR EACH ROW EXECUTE FUNCTION
wholesale_base.subtract_from_leftovers();

ALTER TABLE ONLY wholesale_base."order"

    ADD CONSTRAINT fk_customer FOREIGN KEY (customer_id) REFERENCES
wholesale_base.customer(customer_id) NOT VALID;

ALTER TABLE ONLY wholesale_base."order"

    ADD CONSTRAINT fk_employee FOREIGN KEY (employee_id) REFERENCES
wholesale_base.employee(employee_id) NOT VALID;

ALTER TABLE ONLY wholesale_base.purchase

    ADD CONSTRAINT fk_employee FOREIGN KEY (employee_id) REFERENCES
wholesale_base.employee(employee_id) NOT VALID;

ALTER TABLE ONLY wholesale_base.employee
```

```
    ADD CONSTRAINT fk_job FOREIGN KEY (job_id) REFERENCES
wholesale_base.job(job_id) NOT VALID;

ALTER TABLE ONLY wholesale_base.product

    ADD CONSTRAINT fk_manufacturer FOREIGN KEY (manufacturer_id) REFERENCES
wholesale_base.manufacturer(manufacturer_id) NOT VALID;

ALTER TABLE ONLY wholesale_base.cash_account

    ADD CONSTRAINT fk_order FOREIGN KEY (order_id) REFERENCES
wholesale_base."order"(order_id) NOT VALID;

ALTER TABLE ONLY wholesale_base.purchase_list

    ADD CONSTRAINT fk_product FOREIGN KEY (product_id) REFERENCES
wholesale_base.product(product_id) NOT VALID;

ALTER TABLE ONLY wholesale_base.purchase_list

    ADD CONSTRAINT fk_purchase FOREIGN KEY (purchase_id) REFERENCES
wholesale_base.purchase(purchase_id) NOT VALID;

ALTER TABLE ONLY wholesale_base.cash_account_purchase

    ADD CONSTRAINT fk_purchase FOREIGN KEY (purchase_id) REFERENCES
wholesale_base.purchase(purchase_id) NOT VALID;

ALTER TABLE ONLY wholesale_base.purchase

    ADD CONSTRAINT fk_supplier FOREIGN KEY (supplier_id) REFERENCES
wholesale_base.supplier(supplier_id) NOT VALID;


ALTER TABLE ONLY wholesale_base.order_list

    ADD CONSTRAINT order_fkey FOREIGN KEY (order_id) REFERENCES
wholesale_base."order"(order_id) NOT VALID;


ALTER TABLE ONLY wholesale_base.order_list

    ADD CONSTRAINT purchase_list_fkey FOREIGN KEY (purchase_list_id) REFERENCES
wholesale_base.purchase_list(purchase_list_id) NOT VALID;
```

Заполнение рабочими данными:

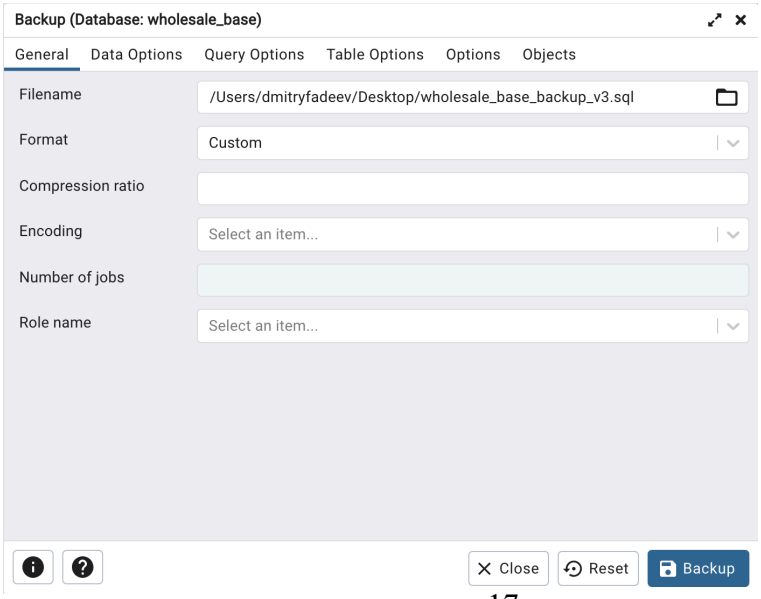
Заполнение было бы слишком огромное по листику, так что прикрепляю фото

	order_id [PK] integer	date_of_sale date	contact_person character varying (50)	employee_id integer	customer_id integer
1	1	2023-04-25	Samantha Brown	19	15
2	2	2023-05-01	Chris Evans	20	16
3	3	2023-05-14	Jennifer Lee	21	1
4	4	2023-06-20	Kevin Johnson	22	2
5	5	2023-07-30	Melissa Smith	23	3
6	6	2023-08-05	Christopher Davis	24	4
7	7	2023-08-25	Jessica White	19	5
8	8	2023-09-15	Andrew Martin	20	6
9	9	2023-09-28	Stephanie Thompson	21	7
10	10	2024-10-05	Patrick Harris	22	8
11	11	2023-10-24	Nicole Clark	23	9

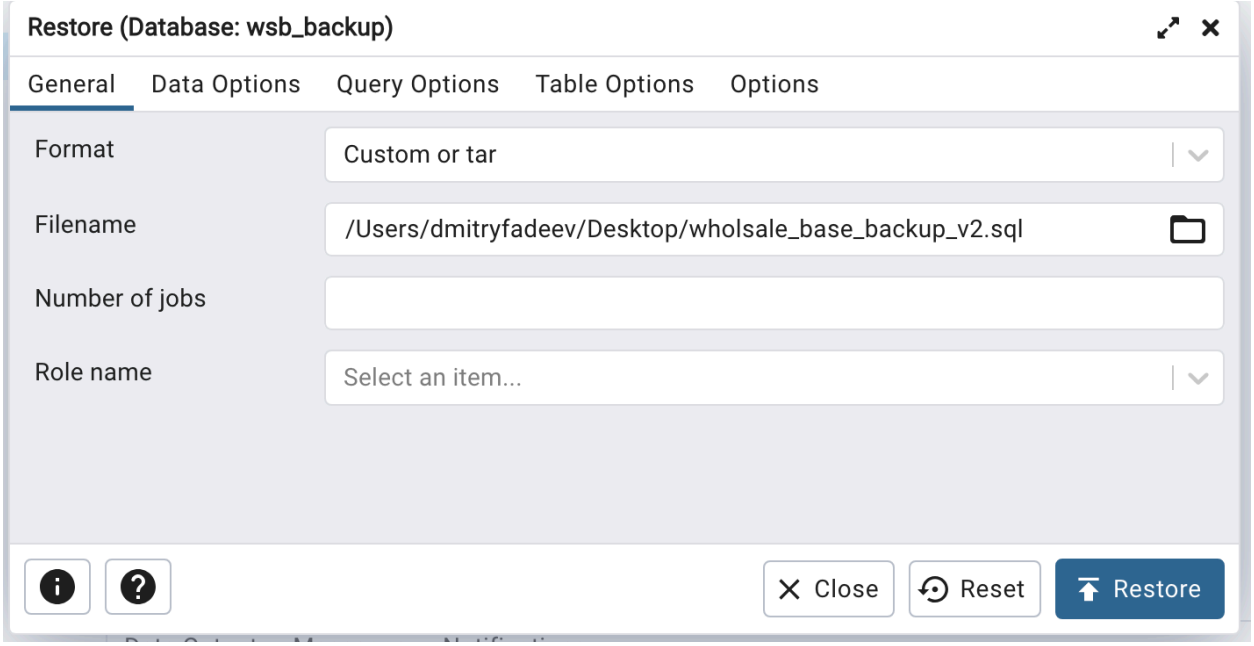
Total rows: 30 of 30 | Query complete 00:00:00.054

простейшего select запроса

Создание бэкапа:



Восстановление БД:



Вывод:

В данной лабораторной работе выполнены различные запросы к базе данных «Банк», используя соединения таблиц, подзапросы и др. Были выполнены запросы на создание представлений, а также на модификацию данных: вставка, изменение и удаление. Были выполнены запросы без индекса и созданы планы запросов через EXPLAIN, далее были созданы различные индексы для различных запросов, но ни в одном из них не получилось их использовать, так как выполнение запроса без индекса вероятнее происходило быстрее из-за маленького количества данных в таблицах.