Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования «Российский химико-технологический университет имени Д.И. Менделеева»

Кафедра информационных компьютерных технологий

Отчёт по курсовой работе по дисциплине «Базы данных» на тему:

Отдел продаж фирмы

Исполнитель: гр. КС-33
ФИО студента: Копылов М.Д.
ФИО руководителя: Семёнов Г.Н.

Москва, 2022 г.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ
ПО ДИСЦИПЛИНЕ «БАЗЫ ДАННЫХ

Студенту Копылову М.Д. группы КС-33

Тема курсовой работы

Отдел продаж фирмы

Этапы разработки работы

1._____
2._____
3._____
4._____
5._____

Основные разделы	Удельный вес раздела работы	Срок выполнения
Этап 1		
Этап 2		
Этап 3		
Этап 4		
Этап 5		

Дата выдачи задания _____20__г. _____
Дата сдача работы _____20__г. _____
Дата защиты _____ 20__ г._____
Руководитель работы Семёнов Г.Н._____

Оглавление

Введение

Проектирование баз данных – тяжёлый интеллектуальный труд, потому что для создания полноценной информационной структуры нужно учесть очень много факторов: осмысленность атрибутов, дубликаты, отсутствие элементов, нормализация и др.

В данной работе я постарался учесть все эти факторы и не только, вот что у меня получилось.

Техническое задание

Спроектировать небольшую реляционную базу данных для предметной области, заданной набором ключевых слов. Нормализация таблиц (не менее трёх) должна быть сделана до 3 нормальной формы **3NF**.

Реализовать базу данных, в которой должны быть:

1. Клиенты: название компании, ФИО контактного лица, адрес выставления счѐта, адрес доставки, телефон.
2. Заказы: дата получения заказа, тип заказа, общая стоимость, скидка, товар, их изготовители, модели, клиент, срок завершения, стоимость доставки.
3. Ресурсы: ФИО, отделы и телефоны исполнителей, число рабочих часов для выполнения заказа, ставка зарплаты, материалы, их количество и стоимость, наличие материалов на складе.

Инфологическая модель

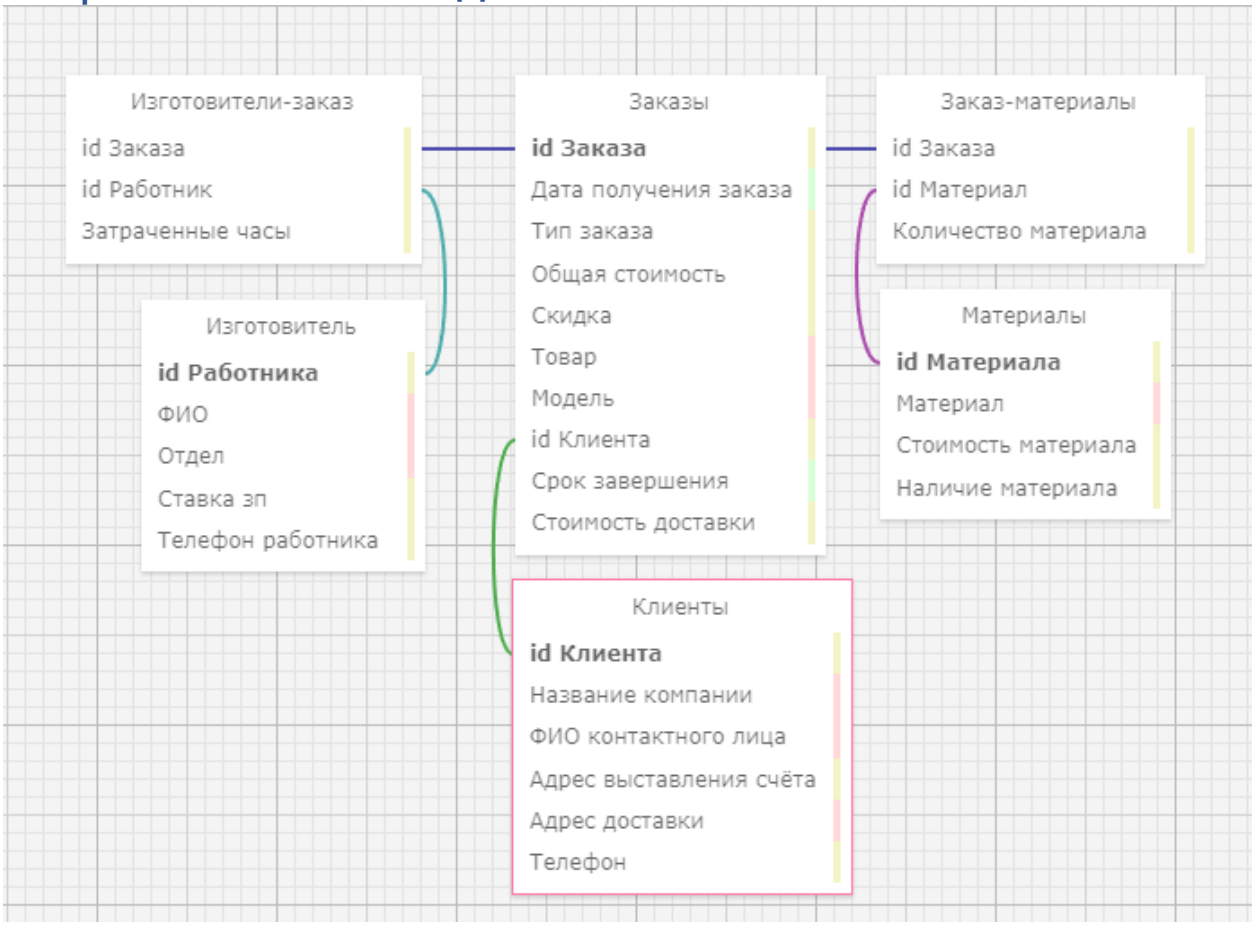

Рисунок 1. Инфологическая модель

Даталогическая модель

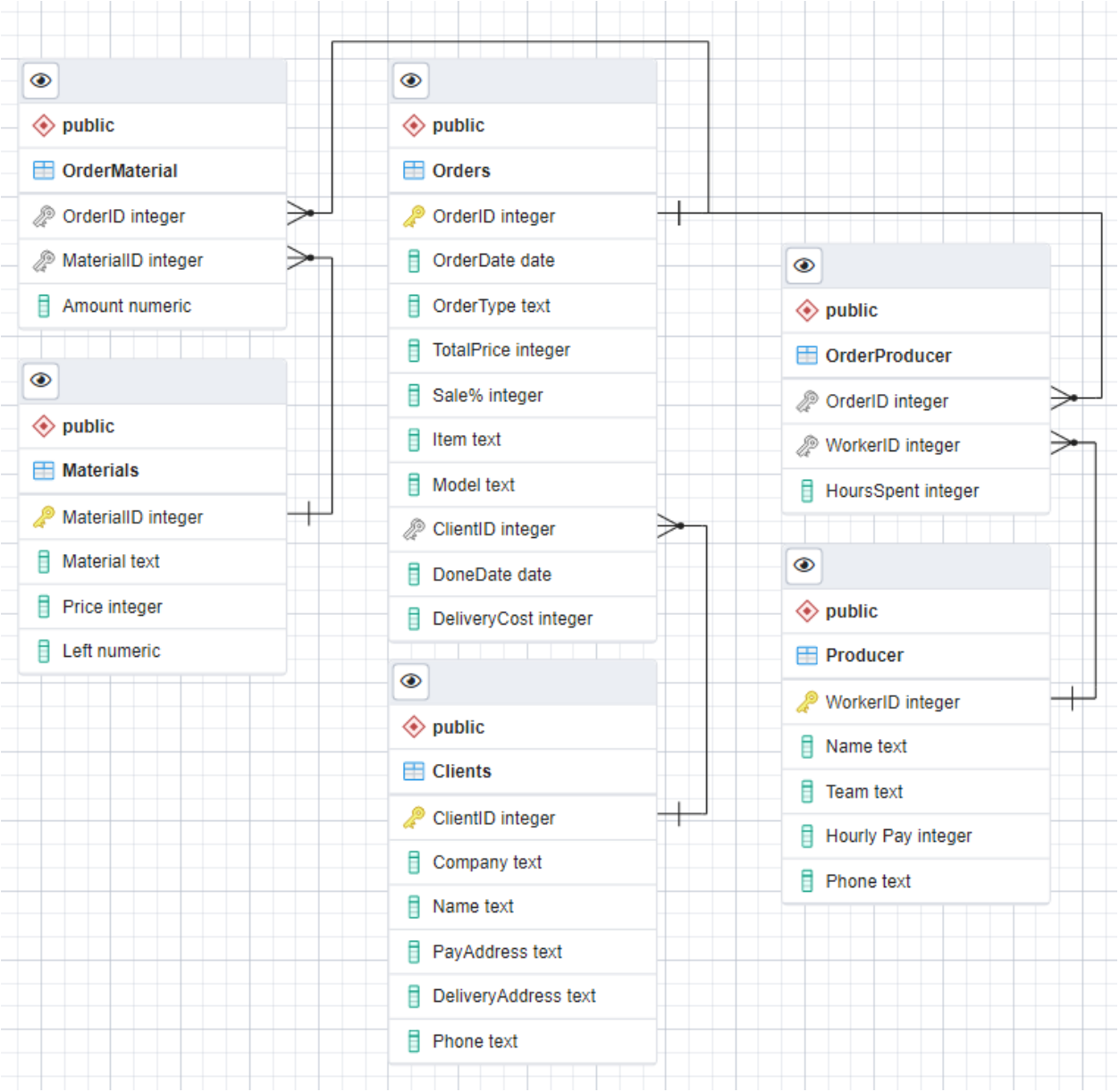

Рисунок 2. Даталогическая модель

Изготовитель:

```
CREATE TABLE public."Producer"
(
    "WorkerID" integer NOT NULL,
    "Name" text NOT NULL,
    "Team" text NOT NULL,
    "Hourly Pay" numeric NOT NULL,
```

```
    "Phone" text NOT NULL,
    PRIMARY KEY ("WorkerID")
);

ALTER TABLE IF EXISTS public."Producer"
    OWNER to postgres;

INSERT INTO "Producer" VALUES
(1,'Копылов Михаил Дмитриевич','Разработка',400,'+79206159976'),
(2,'Немирко Глеб Алексеевич','Разработка',369,'+79690005535'),
(3,'Стариков Артем Борисович','Менеджмет',420,'+78005553535')
```

	WorkerID [PK] integer	Name text	Team text	Hourly Pay integer	Phone text
1	1	Копылов Михаил Дмитриевич	Разработка	400	+79206159976
2	2	Немирко Глеб Алексеевич	Разработка	369	+79690005535
3	3	Стариков Артем Борисович	Менеджмет	420	+78005553535

Рисунок 3. Изготовитель

Клиенты:

```
CREATE TABLE public."Clients"
(
    "ClientID" integer NOT NULL,
    "Company" text NOT NULL,
    "Name" text NOT NULL,
    "PayAddress" text NOT NULL,
    "DeliveryAddress" text NOT NULL,
    "Phone" text NOT NULL,
    PRIMARY KEY ("ClientID ")
);
ALTER TABLE IF EXISTS public." Clients "
    OWNER to postgres;
```

INSERT INTO "Clients" VALUES

(1, 'Рога и копыта', 'Зубов Дмитрий Владимирович', 'Москва, Кутузовский проспект, 1', 'Москва, Кутузовский проспект, 1', '+79990000451'),

(2, 'Озон', 'Иванов Иван Иванович', 'Москва, Лубянка, 1', 'Москва, Лубянка, 2', '+79307776655')

	ClientID [PK] integer	Company text	Name text	PayAddress text	DeliveryAddress text	Phone text
1	1	Рога и копыта	Зубов Дмитрий Владимирович	Москва, Кутузовский проспект, 1	Москва, Кутузовский проспект, 1	+79990000451
2	2	Озон	Иванов Иван Иванович	Москва, Лубянка, 1	Москва, Лубянка, 2	+79307776655

Рисунок 4. Клиенты

Заказы:

```
CREATE TABLE public."Orders"
(
    "OrderID" integer NOT NULL,
    "OrderDate" date NOT NULL,
    "OrderType" text NOT NULL,
    "TotalPrice" integer NOT NULL,
    "Sale%" integer NOT NULL,
    "Item" text NOT NULL,
    "Model" text NOT NULL,
    "ClientID" integer NOT NULL,
    "DoneDate" date,
    "DeliveryCost" integer NOT NULL,
    PRIMARY KEY ("OrderID")
);

ALTER TABLE IF EXISTS public."Orders"
    OWNER to postgres;

ALTER TABLE IF EXISTS public."Orders"
    ADD CONSTRAINT "ClientID" FOREIGN KEY ("ClientID")
    REFERENCES public."Clients" ("ClientID") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID;

INSERT INTO "Orders" VALUES
(1, '2022-01-01', 'Обычный', 100000, 0, 'Сайт', 'Лендинг', 1, '2022-02-02', 0),
(2, '2022-03-01', 'Срочный', 50000, 0, 'Сайт', 'Регистрация', 1, '2022-03-07', 500),
(3, '2022-03-10', 'Обычный', 100000, 0, 'Программа', 'Калькулятор', 2, '2022-04-15', 1000),
```

(4, '2022-04-01', 'Обычный', 150000, 30, 'Сайт', 'Лендинг', 2, '2022-05-01', 0),
(5, '2022-06-01', 'Срочный', 70000, 10, 'Программа', 'Дефрагментатор', 1, '2022-06-15', 0);

	OrderID [PK] integer	OrderDate date	OrderType text	TotalPrice integer	Sale% integer	Item text	Model text	ClientID integer	DoneDate date	DeliveryCost integer
1	1	2022-01-01	Обычный	100000	0	Сайт	Лендинг	1	2022-02-02	0
2	2	2022-03-01	Срочный	50000	0	Сайт	Регистрация	1	2022-03-07	500
3	3	2022-03-10	Обычный	100000	0	Програм...	Калькулятор	2	2022-04-15	1000
4	4	2022-04-01	Обычный	150000	30	Сайт	Лендинг	2	2022-05-01	0
5	5	2022-06-01	Срочный	70000	10	Програм...	Дефрагментатор	1	2022-06-15	0

Рисунок 5. Заказы

Заказы-Производитель

```
CREATE TABLE public."OrderProducer"
(
    "OrderID" integer NOT NULL,
    "WorkerID" integer NOT NULL,
    "HoursSpent" integer NOT NULL,
    CONSTRAINT "OrderID" FOREIGN KEY ("OrderID")
        REFERENCES public."Orders" ("OrderID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "WorkerID" FOREIGN KEY ("WorkerID")
        REFERENCES public."Producer" ("WorkerID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

ALTER TABLE IF EXISTS public."OrderProducer"
    OWNER to postgres;

INSERT INTO "OrderProducer" VALUES
(1, 1, 20),
(1, 2, 30),
(1, 3, 20),
(2, 1, 100),
(3, 2, 50),
(3, 3, 10),
(4, 1, 30),
(4, 2, 30),
(5, 1, 80),
(5, 2, 80),
```

(5, 3, 20)

	OrderID integer	WorkerID integer	HoursSpent integer
1	1	1	20
2	1	2	30
3	1	3	20
4	2	1	100
5	3	2	50
6	3	3	10
7	4	1	30
8	4	2	30
9	5	1	80
10	5	2	80
11	5	3	20

Рисунок 6. Заказы-Подрядчик

Материалы:

```
CREATE TABLE public."Материалы"
(
    "MaterialID" integer NOT NULL,
    "Material" text NOT NULL,
    "Price" integer NOT NULL,
    "Left" numeric NOT NULL,
    PRIMARY KEY ("MaterialID")
);

ALTER TABLE IF EXISTS public."Материалы"
    OWNER to postgres;

INSERT INTO "Materials" VALUES
(1, 'ИнтернетГБ', 3, 1000.0),
(2, 'КофеЛитр', 300, 10.3),
(3, 'МаффинШт', 100, 5),
(4, 'ЭлектричествоКВт', 6, 1000.0),
(5, 'КлавиатураРабочая', 3000, 2)
```

	MaterialID [PK] integer	Material text	Price integer	Left numeric
1	1	ИнтернетГБ	3	1000.0
2	2	КофеЛитр	300	10.3
3	3	МаффинШт	100	5
4	4	ЭлектричествоКВт	6	1000.0
5	5	КлавиатураРабочая	3000	2

Рисунок 7. Материалы

Заказ-Материал:

```
CREATE TABLE public."OrderMaterial"
(
    "OrderID" integer,
    "MaterialID" integer,
    "Amount" numeric(),
    CONSTRAINT "OrderID" FOREIGN KEY ("OrderID")
        REFERENCES public."Orders" ("OrderID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "MaterialID" FOREIGN KEY ("MaterialID")
        REFERENCES public."Materials" ("MaterialID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);

ALTER TABLE IF EXISTS public."OrderMaterial"
    OWNER to postgres;
INSERT INTO "OrderMaterial" VALUES
(1, 1, 10),
```

(1, 2, 0.9),

(1, 4, 10),

(2, 1, 30),

(2, 2, 1.5),

(2, 3, 2),

(2, 4, 15),

(3, 2, 3),

(3, 3, 5),

(3, 4, 50),

(4, 1, 42),

(4, 2, 2.2),

(4, 3, 4),

(4, 4, 48.1),

(4, 5, 1),

(5, 1, 31.4),

(5, 3, 1),

(5, 4, 100)

	OrderID integer	MaterialID integer	Amount numeric
1	1	1	10
2	1	2	0.9
3	1	4	10
4	2	1	30
5	2	2	1.5
6	2	3	2
7	2	4	15
8	3	2	3
9	3	3	5
10	3	4	50
11	4	1	42
12	4	2	2.2
13	4	3	4
14	4	4	48.1
15	4	5	1
16	5	1	31.4
17	5	3	1
18	5	4	100

Рисунок 8. Заказ-Материал

Результаты запросов

Запрос 1, 6. Выбор значений заданных атрибутов из более чем 2х таблиц с сортировкой, уместное использование представлений

Создать представление заказа, показав его параметры и необходимые материалы без учета труда рабочих.

```
CREATE VIEW "OrderWithMaterials" AS
SELECT "Orders".*, "OrderMaterial"."MaterialID", "OrderMaterial"."Amount",
"Materials"."Material", "Materials"."Price" FROM
("Orders" JOIN "OrderMaterial" ON "OrderMaterial"."OrderID" =
"Orders"."OrderID"
JOIN "Materials" ON "OrderMaterial"."MaterialID" = "Materials"."MaterialID")
ORDER BY "Orders"."OrderID"
```

	OrderID integer	OrderDate date	OrderType text	TotalPrice integer	Sale% integer	Item text	Model text	ClientID integer	DoneDate date	DeliveryCost integer	MaterialID integer	Amount numeric	Material text	Price integer
1	1	2022-01-01	Обычный	100000	0	Сайт	Лендинг	1	2022-02-02	0	1	10	ИнтернетГБ	3
2	1	2022-01-01	Обычный	100000	0	Сайт	Лендинг	1	2022-02-02	0	2	0.9	КофеЛитр	300
3	1	2022-01-01	Обычный	100000	0	Сайт	Лендинг	1	2022-02-02	0	4	10	ЭлектричествоКВт	6
4	2	2022-03-01	Срочный	50000	0	Сайт	Регистрация	1	2022-03-07	500	1	30	ИнтернетГБ	3
5	2	2022-03-01	Срочный	50000	0	Сайт	Регистрация	1	2022-03-07	500	2	1.5	КофеЛитр	300
6	2	2022-03-01	Срочный	50000	0	Сайт	Регистрация	1	2022-03-07	500	3	2	МаффинШт	100
7	2	2022-03-01	Срочный	50000	0	Сайт	Регистрация	1	2022-03-07	500	4	15	ЭлектричествоКВт	6
8	3	2022-03-10	Обычный	100000	0	Программа	Калькулятор	2	2022-04-15	1000	2	3	КофеЛитр	300
9	3	2022-03-10	Обычный	100000	0	Программа	Калькулятор	2	2022-04-15	1000	3	5	МаффинШт	100
10	3	2022-03-10	Обычный	100000	0	Программа	Калькулятор	2	2022-04-15	1000	4	50	ЭлектричествоКВт	6
11	4	2022-04-01	Обычный	150000	30	Сайт	Лендинг	2	2022-05-01	0	1	42	ИнтернетГБ	3
12	4	2022-04-01	Обычный	150000	30	Сайт	Лендинг	2	2022-05-01	0	2	2.2	КофеЛитр	300
13	4	2022-04-01	Обычный	150000	30	Сайт	Лендинг	2	2022-05-01	0	3	4	МаффинШт	100
14	4	2022-04-01	Обычный	150000	30	Сайт	Лендинг	2	2022-05-01	0	4	48.1	ЭлектричествоКВт	6
15	4	2022-04-01	Обычный	150000	30	Сайт	Лендинг	2	2022-05-01	0	5	1	КлавиатураРабоч...	3000
16	5	2022-06-01	Срочный	70000	10	Программа	Дефрагментатор	1	2022-06-15	0	1	31.4	ИнтернетГБ	3
17	5	2022-06-01	Срочный	70000	10	Программа	Дефрагментатор	1	2022-06-15	0	3	1	МаффинШт	100
18	5	2022-06-01	Срочный	70000	10	Программа	Дефрагментатор	1	2022-06-15	0	4	100	ЭлектричествоКВт	6

Рисунок 9. Результат выбора нескольких таблиц с группировкой

Запрос 2 Использование условий WHERE, отличных от (=, <, >)

Показать заказы, на исполнение которых ушло от 1 до 2 литров кофе

```
SELECT * FROM
(SELECT * FROM "OrderWithMaterials" WHERE "MaterialID" = 2) As
"Myview"
WHERE "Myview"."Amount" BETWEEN 1 AND 2
```

	OrderID integer	OrderDate date	OrderType text	TotalPrice integer	Sale% integer	Item text	Model text	ClientID integer	DoneDate date	DeliveryCost integer	MaterialID integer	Amount numeric	Material text	Price integer
1	2	2022-03-01	Срочный	50000	0	Сайт	Регистрация	1	2022-03-07	500	2	1.5	КофеЛитр	300

Рисунок 10. Использование WHERE

Запрос 3 Запрос с использованием подзапросов. Уместное использование однострочных функций

Показать ID заказа, его предмет и модель, если стоимость заказать меньше средней

```
SELECT DISTINCT "OrderID","Item","Model" FROM "OrderWithMaterials"
WHERE "TotalPrice" < (SELECT AVG("TotalPrice") FROM
"OrderWithMaterials")
```

	OrderID integer	Item text	Model text
1	2	Сайт	Регистрация
2	5	Программа	Дефрагментатор

Рисунок 11. Подзапросы

Запрос 4 Вычисление групповой функции (GROUP BY) (с HAVING или с несколькими таблицами - 2 балла)

Показать ID заказа и количество разных материалов, которые были потрачены при изготовлении.

```
SELECT "OrderID", COUNT("MaterialID") AS "Distinct materials" FROM
"OrderWithMaterials"
GROUP BY "OrderID"
HAVING COUNT("MaterialID") > 3
```

	OrderID integer	Distinct materials bigint
1	2	4
2	4	5

Рисунок 12. Запрос GROUP BY

Запрос 5 Запрос с использованием коррелированных подзапросов

Вывести всю информацию о заказах, в которых работники сломали клавиатуру (т.е. клавиатура в материалах)

```
SELECT * FROM "OrderWithMaterials" b
WHERE 5 IN (SELECT "MaterialID"
                FROM "OrderWithMaterials" d
                WHERE b."OrderID" = d."OrderID")
```

	OrderID integer	OrderDate date	OrderType text	TotalPrice integer	Sale% integer	Item text	Model text	ClientID integer	DoneDate date	DeliveryCost integer	MaterialID integer	Amount numeric	Material text	Price integer
1	4	2022-04-01	Обычный	150000	30	Сайт	Лендинг	2	2022-05-01	0	1	42	Интернет...	3
2	4	2022-04-01	Обычный	150000	30	Сайт	Лендинг	2	2022-05-01	0	2	2.2	КофеЛитр	300
3	4	2022-04-01	Обычный	150000	30	Сайт	Лендинг	2	2022-05-01	0	3	4	Маффин...	100
4	4	2022-04-01	Обычный	150000	30	Сайт	Лендинг	2	2022-05-01	0	4	48.1	Электрич...	6
5	4	2022-04-01	Обычный	150000	30	Сайт	Лендинг	2	2022-05-01	0	5	1	Клавиату...	3000

Рисунок 13. Коррелирующий подзапрос

Запрос 7. Запрос с использованием какой-нибудь теоретико-множественной операции (объединение, пересечение, разность) над результатами выборок.

Показать информацию по обычным заказам, кроме тех, в которых нужно было сделать программу.

```
SELECT * FROM "OrderWithMaterials"
WHERE "OrderType" = 'Обычный'
EXCEPT
SELECT * FROM "OrderWithMaterials"
WHERE "Item" = 'Программа'
ORDER BY "OrderID"
```

	OrderID integer	OrderDate date	OrderType text	TotalPrice integer	Sale% integer	Item text	Model text	ClientID integer	DoneDate date	DeliveryCost integer	MaterialID integer	Amount numeric	Material text	Price integer
1	1	2022-01-01	Обычный	100000	0	Сайт	Лендинг	1	2022-02-02	0	2	0.9	КофеЛитр	300
2	1	2022-01-01	Обычный	100000	0	Сайт	Лендинг	1	2022-02-02	0	4	10	ЭлектричествоКВт	6
3	1	2022-01-01	Обычный	100000	0	Сайт	Лендинг	1	2022-02-02	0	1	10	ИнтернетГБ	3
4	4	2022-04-01	Обычный	150000	30	Сайт	Лендинг	2	2022-05-01	0	3	4	МаффинШт	100
5	4	2022-04-01	Обычный	150000	30	Сайт	Лендинг	2	2022-05-01	0	5	1	КлавиатураРабоч...	3000
6	4	2022-04-01	Обычный	150000	30	Сайт	Лендинг	2	2022-05-01	0	4	48.1	ЭлектричествоКВт	6
7	4	2022-04-01	Обычный	150000	30	Сайт	Лендинг	2	2022-05-01	0	2	2.2	КофеЛитр	300
8	4	2022-04-01	Обычный	150000	30	Сайт	Лендинг	2	2022-05-01	0	1	42	ИнтернетГБ	3

Рисунок 14. Запрос Разность

Запрос 8. Создать осмысленные триггеры (по 1 баллу за каждый) и хранимые процедуры с параметром(параметрами) для получения информации, которую невозможно получить SQL-запросом (по 2 балла за каждую)

Триггер: Ограничим цену TotalPrice – должна быть больше 0

```
CREATE FUNCTION TotalPriceCheck() RETURNS TRIGGER AS $$
BEGIN

IF NEW."TotalPrice" <= 0 THEN
RAISE EXCEPTION 'Total price cannnot be 0 or less';
END IF;
RETURN NEW;

END;
```

$$ LANGUAGE plpgsql;
CREATE TRIGGER TotalPriceCheck BEFORE INSERT OR UPDATE ON "Orders"
FOR EACH ROW EXECUTE PROCEDURE TotalPriceCheck();

```
1  INSERT INTO "Orders" VALUES
2  (6, '2022-02-04', 'Обычный', 0, 0, 'Сайт', 'Регистрация', 2, '2022-02-06', 0)
```

Data output | Сообщения | Notifications

```
ERROR: ОШИБКА:  Total price cannnot be 0 or less
CONTEXT:  функция PL/pgSQL totalpricecheck(), строка 5, оператор RAISE
```

Рисунок 15. Триггер

Процедура: узнаем сколько денег мы приобрели/потеряли бы на заказе, будь у него другая скидка

CREATE PROCEDURE money_spent(IN order_id integer, IN new_sale integer, OUT expenses numeric) AS $$

DECLARE

total_price integer;

sale integer;

BEGIN

SELECT "TotalPrice", "Sale%"

INTO total_price, sale

FROM "OrderWithMaterials"

WHERE "OrderID" = order_id;

expenses := (total_price*sale/100) - (total_price * new_sale/100);

END;

$$ LANGUAGE plpgsql;

CALL money_spent(1, 10, NULL);

	expenses numeric
1	-10000

Рисунок 16. Хранимая процедура

Вывод

Была разработана реляционная база данных для отдела продаж фирмы в третьей нормальной форме. К ней построены инфологическая и даталогическая схемы, написаны разнообразные запросы, написан триггер и хранимая процедура.

Список литературы

1. https://ondras.zarovi.cz/sql/demo/
2. https://habr.com/ru/post/254773/
3. https://www.postgresql.org/docs/current/sql-createtrigger.html
4. https://learn.microsoft.com/ru-RU/sql/t-sql/language-elements/set-operators-except-and-intersect-transact-sql?view=azure-sqldw-latest