1. Автоматизировать логирование времени последнего изменения в таблице products. Добавить в products соответствующую колонку и реализовать построчный триггер.

```
ALTER TABLE products

ADD COLUMN last_change timestamp;


CREATE OR REPLACE FUNCTION time_change() RETURNS trigger AS $$

BEGIN

        NEW.last_change = now();

        RETURN NEW;

END

$$ LANGUAGE plpgsql;


CREATE TRIGGER save_time BEFORE INSERT OR UPDATE ON products

        FOR EACH ROW EXECUTE PROCEDURE time_change();
```

2. Автоматизировать аудит операций в таблице order_details. Создайте отдельную таблицу для аудита, добавьте туда колонки для хранения наименования операций, имени пользователя и временного штампа. Реализуйте триггеры на утверждения.

```
CREATE TABLE order_details_audit(

        op char(1) NOT NULL,

        user_changed text NOT NULL,

        time_change timestamp,


        order_id smallint NOT NULL,

        product_id smallint NOT NULL,

        unit_price real NOT NULL,
```

```
        quantity smallint NOT NULL,
        discount real
);

CREATE OR REPLACE FUNCTION audit() RETURNS trigger AS $$
BEGIN
        IF TG_OP = 'INSERT' THEN
                INSERT INTO order_details_audit
                SELECT 'I', session_user, now(), nt.*
                FROM new_table nt;
        ELSEIF TG_OP = 'UPDATE' THEN
                INSERT INTO order_details_audit
                SELECT 'U', session_user, now(), nt.*
                FROM new_table nt;
        ELSEIF TG_OP = 'DELETE' THEN
                INSERT INTO order_details_audit
                SELECT 'D', session_user, now(), ot.*
                FROM old_table ot;
        END IF;
        RETURN NULL;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER order_details_audit_trigger_insert AFTER INSERT ON
order_details
REFERENCING NEW TABLE AS new_table
FOR EACH STATEMENT EXECUTE PROCEDURE audit();
```

```
CREATE TRIGGER order_details_audit_trigger_update AFTER UPDATE ON
order_details

REFERENCING NEW TABLE AS new_table

FOR EACH STATEMENT EXECUTE PROCEDURE audit();

CREATE TRIGGER order_details_audit_trigger_delete AFTER DELETE ON
order_details

REFERENCING OLD TABLE AS old_table

FOR EACH STATEMENT EXECUTE PROCEDURE audit();
```