

## TP13 - PostgreSQL Export Functions and Triggers

**Function and trigger to update timestamp .**

```
-- FUNCTION: update timestamp for delivery

CREATE OR REPLACE FUNCTION fn_update_delivery_timestamp()

RETURNS TRIGGER AS $$

BEGIN

    NEW.updated_at = CURRENT_TIMESTAMP;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_update_delivery_timestamp
BEFORE UPDATE ON "PostOffice_App_delivery"
FOR EACH ROW

EXECUTE FUNCTION fn_update_delivery_timestamp();
```

**Function and trigger to show in pgadmin4 logs that a delivery was created.**

```
-- FUNCTION: log delivery created

CREATE OR REPLACE FUNCTION fn_log_delivery_created()

RETURNS TRIGGER AS $$

BEGIN

    RAISE NOTICE 'Delivery % created with status %', NEW.id, NEW.status;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_delivery_created
AFTER INSERT ON "PostOffice_App_delivery"
FOR EACH ROW
```

## TP13 - PostgreSQL Export Functions and Triggers

```
EXECUTE FUNCTION fn_log_delivery_created();
```

**Function and trigger to show in pgadmin4 logs that a delivery status has changed.**

```
-- FUNCTION: log status change

CREATE OR REPLACE FUNCTION fn_log_delivery_status_change()
RETURNS TRIGGER AS $$

BEGIN

    IF NEW.status IS DISTINCT FROM OLD.status THEN

        RAISE NOTICE 'Delivery % changed status from % to %', NEW.id,
OLD.status, NEW.status;

    END IF;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_status_change
AFTER UPDATE ON "PostOffice_App_delivery"
FOR EACH ROW

EXECUTE FUNCTION fn_log_delivery_status_change();
```

## TP13 - PostgreSQL Export Functions and Triggers

**Function and trigger to show in pgadmin4 error in case of fields not being used.**

```
CREATE OR REPLACE FUNCTION fn_validate_delivery()

RETURNS TRIGGER AS $$

BEGIN

    IF NEW.status = 'Completed' AND NEW.delivery_date IS NULL THEN

        RAISE EXCEPTION 'Cannot mark delivery as Completed without
delivery_date';

    END IF;

    IF NEW.weight <= 0 THEN

        RAISE EXCEPTION 'Weight must be greater than 0';

    END IF;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_validate_delivery
BEFORE INSERT OR UPDATE ON "PostOffice_App_delivery"
FOR EACH ROW
EXECUTE FUNCTION fn_validate_delivery();
```

**Function and trigger to show in pgadmin4 error in case of fields not being used.**

```
CREATE OR REPLACE FUNCTION fn_validate_invoice()
```

## TP13 - PostgreSQL Export Functions and Triggers

```

RETURNS TRIGGER AS $$

BEGIN

    IF NEW.cost < 0 THEN

        RAISE EXCEPTION 'Invoice cost cannot be negative';

    END IF;

    IF NEW.quantity IS NOT NULL AND NEW.quantity <= 0 THEN

        RAISE EXCEPTION 'Invoice quantity must be greater than zero';

    END IF;

    IF NEW.paid = TRUE THEN

        NEW.invoice_status = 'Paid';

    END IF;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_validate_invoice
BEFORE INSERT OR UPDATE ON "PostOffice_App_invoice"
FOR EACH ROW
EXECUTE FUNCTION fn_validate_invoice();

```

**Function and trigger to show in pgadmin4 error in case of fields not being used.**

```
-- FUNCTION: validate_driver
```

## TP13 - PostgreSQL Export Functions and Triggers

```
CREATE OR REPLACE FUNCTION fn_validate_driver()

RETURNS TRIGGER AS $$

BEGIN

    IF NEW.license_expiry_date < CURRENT_DATE THEN

        RAISE EXCEPTION 'Driver license expired';

    END IF;

    IF NEW.driving_experience_years < 0 THEN

        RAISE EXCEPTION 'Driving experience cannot be negative';

    END IF;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_validate_driver
BEFORE INSERT OR UPDATE ON "PostOffice_App_employeedriver"
FOR EACH ROW
EXECUTE FUNCTION fn_validate_driver();
```

**Function and trigger to mark all deliveries of a route as completed when the route is completed.**

```
CREATE OR REPLACE FUNCTION fn_route_status()

RETURNS TRIGGER AS $$
```

## TP13 - PostgreSQL Export Functions and Triggers

```

BEGIN

    IF NEW.delivery_status = 'Completed' THEN

        UPDATE "PostOffice_App_delivery"

        SET status='Completed', delivery_date = CURRENT_DATE

        WHERE route_id = NEW.id;

    END IF;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_route_completed

AFTER UPDATE ON "PostOffice_App_route"

FOR EACH ROW

EXECUTE FUNCTION fn_route_status();

```

Export function for warehouses .

```

CREATE OR REPLACE FUNCTION export_warehouses_csv()

RETURNS TABLE(line TEXT)

LANGUAGE plpgsql AS $$

BEGIN

```

## TP13 - PostgreSQL Export Functions and Triggers

```
    RETURN QUERY

    SELECT CONCAT_WS(' ', '
        id,
        name,
        address,
        contact,
        po_schedule_open,
        po_schedule_close,
        maximum_storage_capacity
    )

    FROM "PostOffice_App_warehouse"

    ORDER BY id;

END;

$$;
```

Export function for vehicles

```
CREATE OR REPLACE FUNCTION export_vehicles_csv()

RETURNS TABLE(line TEXT)

LANGUAGE plpgsql AS $$

BEGIN

    RETURN QUERY
```

## TP13 - PostgreSQL Export Functions and Triggers

```
SELECT CONCAT_WS(' ',  
                 id,  
                 plate_number,  
                 brand,  
                 model,  
                 capacity,  
                 vehicle_status,  
                 year,  
                 fuel_type,  
                 last_maintenance_date,  
                 vehicle_type  
)  
  
FROM "PostOffice_App_vehicle"  
  
ORDER BY id;  
  
END;  
  
$$;
```

Export function for routes

```
CREATE OR REPLACE FUNCTION export_routes_csv()  
  
RETURNS TABLE(line TEXT)  
  
LANGUAGE plpgsql AS $$  
  
BEGIN  
  
    RETURN QUERY  
  
    SELECT CONCAT_WS(' ',  
                 id,  
                 route_name,  
                 route_start,  
                 route_end,  
                 route_type,  
                 route_length,  
                 route_time,  
                 route_fuel_type,  
                 route_maintenance_date,  
                 route_status  
)  
  
FROM "PostOffice_App_route"  
  
ORDER BY id;  
  
END;  
  
$$;
```

## TP13 - PostgreSQL Export Functions and Triggers

```

        id,
        description,
        delivery_status,
        vehicle_id,
        driver_id,
        origin_name,
        origin_address,
        origin_contact,
        destination_name,
        destination_address,
        destination_contact,
        delivery_date,
        delivery_start_time,
        delivery_end_time,
        kms_travelled,
        expected_duration,
        driver_notes
    )
    FROM "PostOffice_App_route"
    ORDER BY id;
END;
$$;

```

Export function for deliveries

```

CREATE OR REPLACE FUNCTION export_deliveries_csv()
RETURNS TABLE(line TEXT)
LANGUAGE plpgsql AS $$
```

## TP13 - PostgreSQL Export Functions and Triggers

```
BEGIN  
  
    RETURN QUERY  
  
    SELECT CONCAT_WS(' ',  
                     id,  
                     tracking_number,  
                     description,  
                     sender_name,  
                     sender_address,  
                     sender_phone,  
                     sender_email,  
                     recipient_name,  
                     recipient_address,  
                     recipient_phone,  
                     recipient_email,  
                     item_type,  
                     weight,  
                     dimensions,  
                     status,  
                     priority,  
                     registered_at,  
                     updated_at,  
                     in_transition,  
                     destination,  
                     delivery_date,  
                     driver_id,  
                     invoice_id,  
                     route_id,  
                     client_id
```

## TP13 - PostgreSQL Export Functions and Triggers

```
)  
  
FROM "PostOffice_App_delivery"  
  
ORDER BY id;  
  
END;  
  
$$;
```

Export function for invoices

```
CREATE OR REPLACE FUNCTION export_invoices_csv()  
  
RETURNS TABLE(line TEXT)  
  
LANGUAGE plpgsql AS $$  
  
BEGIN  
  
RETURN QUERY  
  
SELECT CONCAT_WS(' ',  
  
                id_invoice,  
  
                invoice_status,  
  
                invoice_type,  
  
                quantity,  
  
                invoice_datetime,  
  
                cost,  
  
                paid,  
  
                payment_method,  
  
                name,  
  
                address,  
  
                contact,  
  
                user_id  
  
)  
  
FROM "PostOffice_App_invoice"  
  
ORDER BY id_invoice;
```

## TP13 - PostgreSQL Export Functions and Triggers

```
END ;
```

```
$$;
```

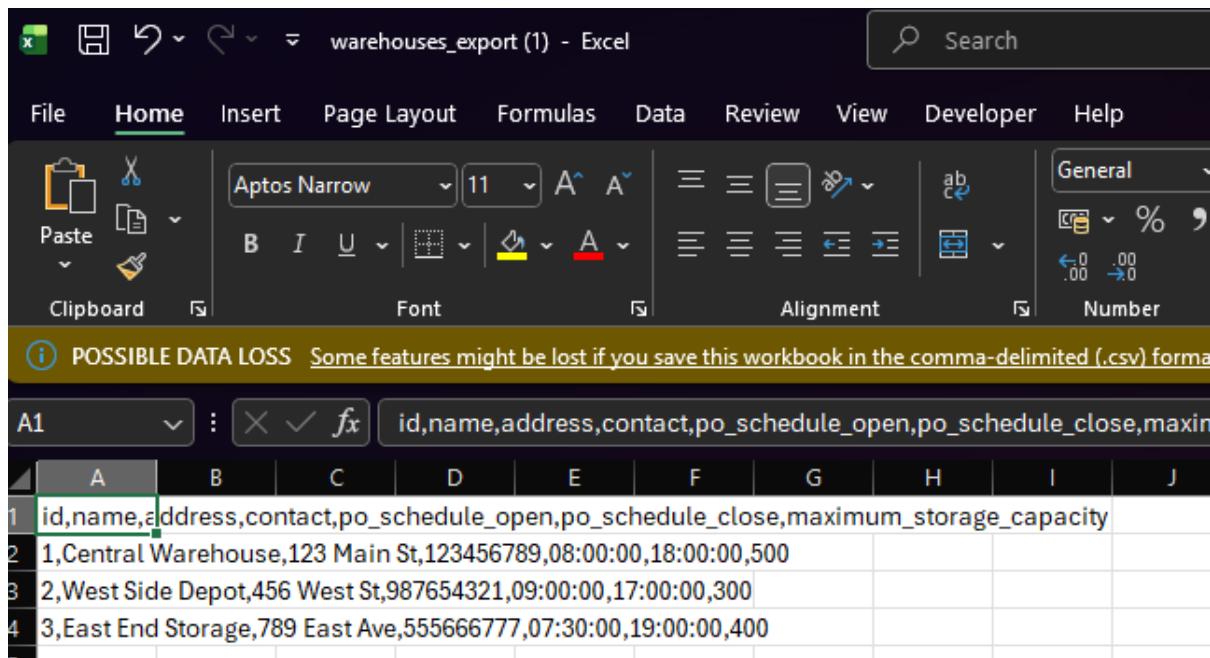
Name	Address	Contact	Open	Close	Capacity	Actions
Central Warehouse	123 Main St	123456789	8 a.m.	6 p.m.	500	<a href="#">Edit</a> <a href="#">Delete</a>
East End Storage	789 East Ave	555666777	7:30 a.m.	7 p.m.	400	<a href="#">Edit</a> <a href="#">Delete</a>
West Side Depot	456 West St	987654321	9 a.m.	5 p.m.	300	<a href="#">Edit</a> <a href="#">Delete</a>

Image 1 -Exemple of import and export of warehouses.

Name	Address	Contact	Open	Close	Capacity	Actions
Central Warehouse	123 Main St	123456789	8 a.m.	6 p.m.	500	<a href="#">Edit</a> <a href="#">Delete</a>
East End Storage	789 East Ave	555666777	7:30 a.m.	7 p.m.	400	<a href="#">Edit</a> <a href="#">Delete</a>
West Side Depot	456 West St	987654321	9 a.m.	5 p.m.	300	<a href="#">Edit</a> <a href="#">Delete</a>

Image 2- Export in Csv.

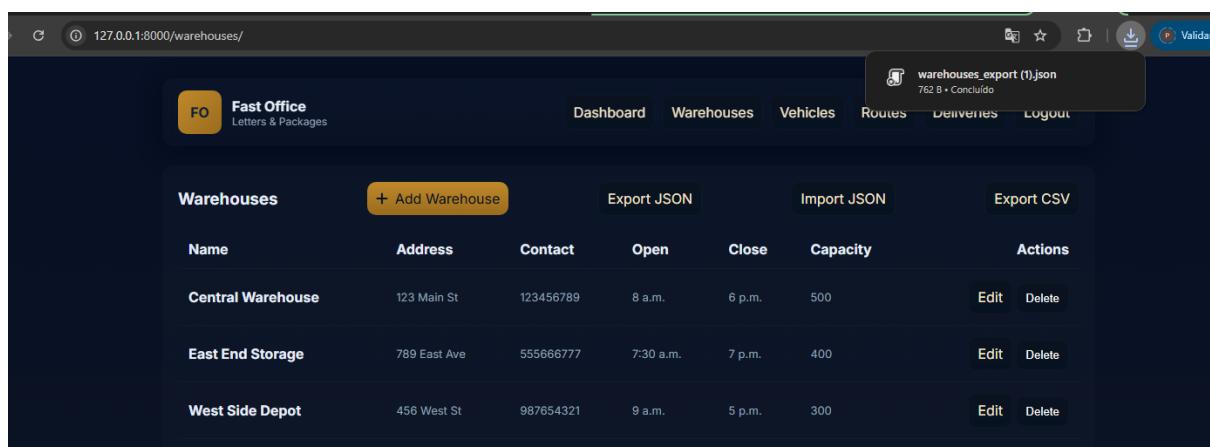
## TP13 - PostgreSQL Export Functions and Triggers



The screenshot shows a Microsoft Excel spreadsheet titled "warehouses\_export (1) - Excel". The ribbon menu is visible at the top, with "Home" selected. The formula bar shows the formula: `=id,name,address,contact,po_schedule_open,po_schedule_close,maximum_storage_capacity`. The data is displayed in a table format:

	A	B	C	D	E	F	G	H	I	J
1	<code>=id,name,address,contact,po_schedule_open,po_schedule_close,maximum_storage_capacity</code>									
2	1,Central Warehouse,123 Main St,123456789,08:00:00,18:00:00,500									
3	2,West Side Depot,456 West St,987654321,09:00:00,17:00:00,300									
4	3,East End Storage,789 East Ave,555666777,07:30:00,19:00:00,400									

Image 3 - Exported Csv



The screenshot shows a web application interface titled "Fast Office Letters & Packages". The URL in the address bar is `127.0.0.1:8000/warehouses/`. The page displays a table of warehouses with the following data:

Name	Address	Contact	Open	Close	Capacity	Actions
Central Warehouse	123 Main St	123456789	8 a.m.	6 p.m.	500	<a href="#">Edit</a> <a href="#">Delete</a>
East End Storage	789 East Ave	555666777	7:30 a.m.	7 p.m.	400	<a href="#">Edit</a> <a href="#">Delete</a>
West Side Depot	456 West St	987654321	9 a.m.	5 p.m.	300	<a href="#">Edit</a> <a href="#">Delete</a>

A tooltip above the table indicates a file download: `warehouses_export (1).json 762 B • Concluido`.

Image 4- Export in json.

## TP13 - PostgreSQL Export Functions and Triggers

```
[  
  {  
    "id": 1,  
    "name": "Central Warehouse",  
    "address": "123 Main St",  
    "contact": "123456789",  
    "po_schedule_open": "08:00:00",  
    "po_schedule_close": "18:00:00",  
    "maximum_storage_capacity": 500  
  },  
  {  
    "id": 3,  
    "name": "East End Storage",  
    "address": "789 East Ave",  
    "contact": "555666777",  
    "po_schedule_open": "07:30:00",  
    "po_schedule_close": "19:00:00",  
    "maximum_storage_capacity": 400  
  },  
  {  
    "id": 2,  
    "name": "West Side Depot",  
    "address": "456 West St",  
    "contact": "987654321",  
    "po_schedule_open": "09:00:00",  
    "po_schedule_close": "17:00:00",  
    "maximum_storage_capacity": 300  
  }  
]
```

Image 5- Exported json content

## TP13 - PostgreSQL Export Functions and Triggers

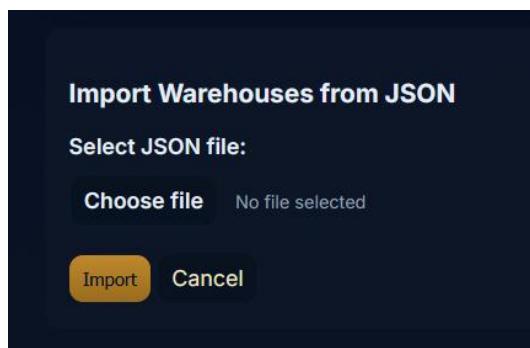


Image 6 - Import file selection page

```

imports_json > {} warehouses_import.json > ...
1  [
2    {
3      "name": "North Logistics Hub",
4      "address": "12 Arctic Road",
5      "contact": "111222333",
6      "po_schedule_open": "08:00:00",
7      "po_schedule_close": "18:00:00",
8      "maximum_storage_capacity": 700
9    },
10   {
11     "name": "South Distribution Point",
12     "address": "88 Sunrise Blvd",
13     "contact": "444555666",
14     "po_schedule_open": "07:00:00",
15     "po_schedule_close": "17:30:00",
16     "maximum_storage_capacity": 650
17   },
18   {
19     "name": "Harbor Storage Facility",
20     "address": "Dockyard Sector 5",
21     "contact": "999888777",
22     "po_schedule_open": "06:00:00",
23     "po_schedule_close": "20:00:00",
24     "maximum_storage_capacity": 900
25   }
26 ]
27

```



Image 7 - Import file selected

Warehouses	+ Add Warehouse	Export JSON		Import JSON		Export CSV	
Name	Address	Contact	Open	Close	Capacity	Actions	
Central Warehouse	123 Main St	123456789	8 a.m.	6 p.m.	500	Edit	Delete
East End Storage	789 East Ave	555666777	7:30 a.m.	7 p.m.	400	Edit	Delete
Harbor Storage Facility	Dockyard Sector 5	999888777	6 a.m.	8 p.m.	900	Edit	Delete
North Logistics Hub	12 Arctic Road	111222333	8 a.m.	6 p.m.	700	Edit	Delete
South Distribution Point	88 Sunrise Blvd	444555666	7 a.m.	5:30 p.m.	650	Edit	Delete
West Side Depot	456 West St	987654321	9 a.m.	5 p.m.	300	Edit	Delete

Image 8 - After clicking import

**TP13 - PostgreSQL Export Functions and Triggers**

Essay developed by:

Pedro Monteiro - estgv14366;  
Rodrigo Rolo - estgv18757;  
Diego Alonso - pv33986;  
David Gonzalez - pv33971;