

### 1. Checked That All Required Tables in BL\_3NF Exist

I made sure that all needed tables are present in the BL\_3NF schema. I also created sequences (like category\_id\_seq, customer\_id\_seq) to generate unique IDs for new records in previous task

```
28  ✓ SELECT table_name
29     FROM information_schema.tables
30     WHERE table_schema = 'bl_3nf'
31     AND table_type = 'BASE TABLE'
32     ORDER BY table_name;
33
```

Data Output Messages Notifications

	table_name name	
1	ce_categories	
2	ce_customer_scd	
3	ce_dates	
4	ce_products	
5	ce_promotion	
6	ce_sales	
7	ce_stores	
8	ce_suppliers	

I gave BL\_CL the right to use the BL\_3NF schema and its tables. This includes permissions to insert, update, and select data from BL\_3NF.

```

54 -- Grant privileges to BL_CL for data cleansing
55 GRANT ALL PRIVILEGES ON SCHEMA BL_3NF TO BL_CL;
56 GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA BL_3NF TO BL_CL;
57 GRANT USAGE, SELECT ON ALL SEQUENCES IN SCHEMA BL_3NF TO BL_CL;

```

Data Output Messages Notifications

GRANT

The idea is to ensure data integrity and track procedure executions using a logging mechanism.

### Creating Logging Table

A logging table is created to keep track of the execution of data loading procedures. This table stores information about the procedure name, rows affected, log message, and timestamp.

```

5 -- Creating a centralized logging table in BL_CL schema
6 CREATE TABLE IF NOT EXISTS BL_CL.procedure_logs (
7     log_id SERIAL PRIMARY KEY,
8     log_timestamp TIMESTAMPTZ DEFAULT NOW(),
9     procedure_name TEXT,
10    rows_affected INT,
11    log_message TEXT,
12    error_message TEXT
13 );
14
15 -- Creating a logging function in BL_CL schema
16 CREATE OR REPLACE FUNCTION BL_CL.log_procedure_action(
17     proc_name TEXT,
18     rows INT,
19     message TEXT,
20     error_message TEXT DEFAULT NULL
21 ) RETURNS VOID AS $$
22 BEGIN
23     INSERT INTO BL_CL.procedure_logs (procedure_name, rows_affected, log_message, error_message)
24     VALUES (proc_name, rows, message, error_message);
25 END;
26 $$ LANGUAGE plpgsql;
27
28 SELECT table_name
29 FROM information_schema.tables
30 WHERE table_schema = 'bl_3nf'
31 AND table_type = 'BASE TABLE'
32 ORDER BY table_name;

```

This table is used to store log entries for each data loading procedure.

**log\_id:** This is like the entry number in our journal. It's a unique number that automatically increases with each new log entry.

**procedure\_name:** This tells the name of the procedure that is ran, so it known what operation the log entry is about.

**rows\_affected:** This shows how many rows of data were processed by the procedure. It's like noting down how many tasks were completed.

log\_message: This is a short description of what happened. It could be something like "Sales loaded successfully"

log\_timestamp: This records the exact time when the log entry was made. It helps know when each procedure was run.

And then next function writing entries into it. We call this function at the end of each data loading procedure to record what happened.

proc\_name is the name of the procedure where I am logging. Rows is the number of rows the procedure affected. Message is describing what the procedure did.

Calling procedures and selecting

The screenshot displays a database management interface. At the top, there is a toolbar with icons for various actions like save, delete, and refresh, followed by a tab labeled 'SQL'. Below this, a table of customer data is shown. The table has columns: customer\_id, customer\_age, customer\_gender, customer\_income, source\_id, source\_entity, and source\_system. The data is organized into 10 rows. Below the table, there is a SQL query window with two lines of code: 'CALL BL\_CL.load\_sales();' and 'SELECT \* FROM BL\_3NF.CE\_SALES LIMIT 10;'. Below the SQL window, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table of sales data. This table has columns: t\_id, customer\_id, product\_id, time, promotion\_id, unit\_price, quantity\_sold, source\_id, and source\_system. The data is organized into 10 rows.

	customer_id [PK] bigint	customer_age integer	customer_gender character varying (10)	customer_income integer	source_id character varying (100)	source_entity character varying (100)	source_system character varying (100)
1	-1	-1	n.a.	-1	-1	MANUAL	MANUAL
2	1	48	[null]	55646	2481	src_offline_sales	sa_offline_sales
3	2	67	[null]	37353	8801	src_offline_sales	sa_offline_sales
4	3	52	[null]	110265	5990	src_offline_sales	sa_offline_sales
5	4	47	[null]	64623	8277	src_offline_sales	sa_offline_sales
6	5	57	[null]	42209	7070	src_offline_sales	sa_offline_sales
7	6	65	[null]	28663	6096	src_offline_sales	sa_offline_sales
8	7	52	[null]	35138	3399	src_offline_sales	sa_offline_sales
9	8	38	[null]	76311	2685	src_offline_sales	sa_offline_sales
10	9	30	[null]	92700	8758	src_offline_sales	sa_offline_sales

```
20 CALL BL_CL.load_sales();
21 SELECT * FROM BL_3NF.CE_SALES LIMIT 10;
```

Data Output Messages Notifications

	t_id [PK] bigint	customer_id integer	product_id integer	time time without time zone	promotion_id integer	unit_price numeric (10,2)	quantity_sold integer	source_id character varying (100)	source_system character varying (100)
1	-1	-1	-1	00:00:00	-1	0.00	0	-1	MANUAL
2	249692	1674	296	20:45:00	[null]	1019.15	3	4866	src_offline_sales
3	249693	1674	275	20:45:00	[null]	1019.15	3	4866	src_offline_sales
4	249694	1674	248	20:45:00	[null]	1019.15	3	4866	src_offline_sales
5	249695	1674	216	20:45:00	[null]	1019.15	3	4866	src_offline_sales
6	249696	1674	208	20:45:00	[null]	1019.15	3	4866	src_offline_sales
7	249697	394	292	22:30:00	[null]	1327.64	1	4391	src_offline_sales
8	249698	394	278	22:30:00	[null]	1327.64	1	4391	src_offline_sales
9	249699	394	277	22:30:00	[null]	1327.64	1	4391	src_offline_sales
10	249700	394	262	22:30:00	[null]	1327.64	1	4391	src_offline_sales

```

18 SELECT * FROM BL_3NF.CE_SUPPLIERS LIMIT 10;
19
20 CALL BL_CL.load_sales();
21 SELECT * FROM BL_3NF.CE_SALES LIMIT 10;

```

Data Output Messages Notifications



	supplier_id [PK] bigint	source_id character varying (100)	source_entity character varying (100)	source_system character varying (100)
1	-1	-1	MANUAL	MANUAL
2	1	474	src_offline_sales	sa_offline_sales
3	2	135	src_offline_sales	sa_offline_sales
4	3	475	src_offline_sales	sa_offline_sales
5	4	118	src_offline_sales	sa_offline_sales
6	5	155	src_offline_sales	sa_offline_sales
7	6	169	src_offline_sales	sa_offline_sales
8	7	250	src_offline_sales	sa_offline_sales
9	8	248	src_offline_sales	sa_offline_sales
10	9	368	src_offline_sales	sa_offline_sales

```

15 SELECT * FROM BL_3NF.CE_DATES LIMIT 10;
16
17 CALL BL_CL.load_suppliers();

```

Data Output Messages Notifications

	transaction_date [PK] date	source_id [PK] character varying (100)	source_entity [PK] character varying (100)	source_system [PK] character varying (100)
1	1900-01-01	MANUAL	MANUAL	MANUAL
2	2024-08-10	8/10/2024	src_offline_sales	sa_offline_sales
3	2024-08-21	8/21/2024	src_offline_sales	sa_offline_sales
4	2024-03-05	3/5/2024	src_offline_sales	sa_offline_sales
5	2024-05-16	5/16/2024	src_offline_sales	sa_offline_sales
6	2024-06-30	6/30/2024	src_offline_sales	sa_offline_sales
7	2024-07-24	7/24/2024	src_offline_sales	sa_offline_sales
8	2024-09-02	9/2/2024	src_offline_sales	sa_offline_sales
9	2024-04-04	4/4/2024	src_offline_sales	sa_offline_sales
10	2024-07-25	7/25/2024	src_offline_sales	sa_offline_sales

```

12 SELECT * FROM BL_3NF.CE_STORES LIMIT 10;
13
14 CALL BL_CL.load_dates();
15 SELECT * FROM BL_3NF.CE_DATES LIMIT 10;
16
17 CALL BL_CL.load_suppliers();

```

Data Output Messages Notifications

	store_id [PK] bigint	store_location character varying (20)	source_id character varying (100)	source_entity character varying (100)	source_system character varying (100)
1	-1	n.a.	-1	MANUAL	MANUAL
2	203	Chicago	3	src_offline_sales	sa_offline_sales
3	204	Chicago	16	src_offline_sales	sa_offline_sales
4	205	Chicago	12	src_offline_sales	sa_offline_sales
5	206	New York	19	src_offline_sales	sa_offline_sales
6	207	Chicago	1	src_offline_sales	sa_offline_sales
7	208	Los Angeles	13	src_offline_sales	sa_offline_sales
8	209	Chicago	18	src_offline_sales	sa_offline_sales
9	210	Miami	7	src_offline_sales	sa_offline_sales
10	211	New York	10	src_offline_sales	sa_offline_sales

```
9 SELECT * FROM BL_3NF.CE_PRODUCTS LIMIT 10;
10
11 CALL BL_CL.load_stores();
```

Data Output Messages Notifications

     SQL

	product_id [PK] bigint	product_name character varying (15)	category_id integer	unit_price numeric (10,2)	source_id character varying (100)	source_entity character varying (100)	source_system character varying (100)
1	-1	n.a.	-1	0.00	-1	MANUAL	MANUAL
2	1	Smartphone	2	1327.64	338	src_offline_sales	sa_offline_sales
3	2	Fridge	2	1690.95	267	src_offline_sales	sa_offline_sales
4	3	Smartphone	2	796.42	339	src_offline_sales	sa_offline_sales
5	4	Fridge	1	956.81	367	src_offline_sales	sa_offline_sales
6	5	Laptop	2	620.58	248	src_offline_sales	sa_offline_sales
7	6	Washing Machine	1	169.00	499	src_offline_sales	sa_offline_sales
8	7	Camera	1	1703.25	140	src_offline_sales	sa_offline_sales
9	8	Laptop	1	440.90	576	src_offline_sales	sa_offline_sales
10	9	Smartphone	2	811.33	343	src_offline_sales	sa_offline_sales

Checking log tables:

	log_id [PK] integer	log_timestamp timestamp with time zone	procedure_name text	rows_affected integer	log_message text	error_message text
9	26	2025-02-20 18:18:39.983811+01	load_suppliers	0	Suppliers loaded successfully	[null]
10	27	2025-02-20 18:20:58.148323+01	load_sales	0	Sales loaded successfully	[null]
11	28	2025-02-20 20:28:30.603225+01	load_categories	0	Categories loaded successfully	[null]
12	29	2025-02-20 20:33:23.895898+01	load_categories	0	Categories loaded successfully	[null]
13	30	2025-02-20 20:33:34.833682+01	load_categories	0	Categories loaded successfully	[null]
14	31	2025-02-21 15:51:07.889897+01	load_customers	0	Customers loaded successfully	[null]
15	32	2025-02-21 15:52:42.733334+01	load_categories	0	Categories loaded successfully	[null]
16	33	2025-02-21 15:59:05.717875+01	load_customers	0	Customers loaded successfully	[null]
17	34	2025-02-21 15:59:10.734558+01	load_products	0	Products loaded successfully	[null]
18	35	2025-02-21 15:59:37.003082+01	load_products	0	Products loaded successfully	[null]
19	36	2025-02-21 15:59:54.652776+01	load_stores	0	Stores loaded successfully	[null]
20	37	2025-02-21 16:00:09.951179+01	load_dates	0	Dates loaded successfully	[null]
21	38	2025-02-21 16:00:25.451493+01	load_suppliers	0	Suppliers loaded successfully	[null]
22	39	2025-02-21 16:00:40.036199+01	load_sales	0	Sales loaded successfully	[null]
23	40	2025-02-21 16:01:11.120044+01	load_customers	0	Customers loaded successfully	[null]
24	41	2025-02-21 16:01:13.134744+01	load_products	0	Products loaded successfully	[null]
25	42	2025-02-21 16:01:16.166978+01	load_stores	0	Stores loaded successfully	[null]
26	43	2025-02-21 16:01:19.415491+01	load_dates	0	Dates loaded successfully	[null]
27	44	2025-02-21 16:01:21.232178+01	load_suppliers	0	Suppliers loaded successfully	[null]
28	45	2025-02-21 16:01:23.366549+01	load_dates	0	Dates loaded successfully	[null]
29	46	2025-02-21 16:01:26.062222+01	load_stores	0	Stores loaded successfully	[null]