

V1:

```
20 INSERT INTO v_active_employees
21 (first_name, last_name, email, phone, hire_date, department_id, position_id, manager_id, active, created_at)
22 VALUES
23 ('Alice','Smith','alice.smith@example.com', '+972500000001', CURRENT_DATE - INTERVAL '30 days',
24 (SELECT department_id FROM department WHERE name='IT'),
25 (SELECT position_id FROM position WHERE title='Developer'),
26 NULL, TRUE, now())
27 RETURNING *;
28 SELECT * FROM v_active_employees WHERE email = 'alice.smith@example.com';
29
30
```

Data Output Messages Notifications

SQL

Showing rows: 1 to 1

Page No: 1 of 1

employee_id	first_name	last_name	email	phone	hire_date	department_id	position_id	manager_id
integer	character varying (100)	character varying (100)	character varying (255)	character varying (50)	date	integer	integer	integer
999989749	Alice	Smith	alice.smith@example.com	+972500000001	2025-09-16	8	31	

Total rows: 1 Query complete 00:00:00.297 CRLF Ln 27, Col 13

Query Query History

SQL

Showing rows: 1 to 1

Page No: 1 of 1

employee_id	first_name	last_name	email	phone	address	birth_date	hire_date	termination_date
[PK] integer	character varying (100)	character varying (100)	character varying (255)	character varying (50)	character varying (255)	date	date	date
1	999989748	Alice	Smith	alice.smith@example.com	+972500000001	[null]	[null]	2025-09-16

Total rows: 1 Query complete 00:00:00.156 CRLF Ln 1, Col 8

Update:

```
1 UPDATE v_active_employees
2 SET phone = '+972500000002'
3 WHERE employee_id = (SELECT employee_id FROM employee WHERE email = 'alice.smith@example.com')
4 RETURNING *;
5
6 -- 5) Show updated rows
7 SELECT * FROM v_active_employees WHERE email = 'alice.smith@example.com';
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

	employee_id integer	first_name character varying (100)	last_name character varying (100)	email character varying (255)	phone character varying (50)	hire_date date	department_id integer	position_id integer	manager integer
1	999989749	Alice	Smith	alice.smith@example.com	+972500000002	2025-09-16	8	31	

Total rows: 1 Query complete 00:00:00.208 CRLF Ln 7, Col 8

Query Query History

```
1 SELECT 'After update: base' AS src, * FROM employee WHERE email = 'alice.smith@example.com';
2
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

	src text	employee_id [PK] integer	first_name character varying (100)	last_name character varying (100)	email character varying (255)	phone character varying (50)	address character varying (255)	birth_date date
1	After update: base	999989749	Alice	Smith	alice.smith@example.com	+972500000002	[null]	[null]

Total rows: 1 Query complete 00:00:00.163 CRLF Ln 2, Col 1

Query

Query History

```

1 DO $$
2 BEGIN
3     -- Attempt to set active = FALSE via the view; this should fail due to WITH CHECK OPTION.
4     UPDATE v_active_employees
5     SET active = FALSE
6     WHERE email = 'alice.smith@example.com';
7 --EXCEPTION WHEN OTHERS THEN
8     --RAISE NOTICE 'EXPECTED ERROR (v_active_employees CHECK OPTION): %', SQLERRM;
9 END$$;
10

```

Data Output

Messages

Notifications

ERROR: new row violates check option for view "v_active_employees"

Failing row contains (999989749, Alice, Smith, alice.smith@example.com, +972500000002, null, null, 2025-09-16, null, f, 8, 31, null, null, null, 2025-10-16 01:46:11.299488).

```

1 SELECT * FROM employee WHERE email = 'alice.smith@example.com';
2
3

```

Data Output

Messages

Notifications

+

📄

▼

🗑️

▼

🗑️

📄

📄

📄

📄

📄

SQL

Showing rows: 1 to 1

Page No: 1

of 1

⏪

⏴

⏵

⏩

	phone (255) character varying (50)	address character varying (255)	birth_date date	hire_date date	termination_date date	active boolean	department_id integer	position_id integer	manager_id integer	employee_id integer
1	nple.com	+972500000002	[null]	[null]	2025-09-16	[null]	true	8	31	[null]

Total rows: 1

Query complete 00:00:00.190

CRLF

Ln 1, Col 8

Delete:

```
1 DELETE FROM v_active_employees
2 WHERE employee_id = (SELECT employee_id FROM employee WHERE email = 'alice.smith@example.com')
3 RETURNING *;
4
5 -- 6) Confirm deletion in base table
6 SELECT 'After delete: base (should be empty)' AS note, * FROM employee WHERE email = 'alice.smith@example.com';
7
8
```

Data Output Messages Notifications

note	employee_id	first_name	last_name	email	phone	address	birth_date	hire_date
text	[PK] integer	character varying (100)	character varying (100)	character varying (255)	character varying (50)	character varying (255)	date	date

Total rows: 0 Query complete 00:00:00.219 CRLF Ln 8, Col 1

V2 v_oncall_shifts_valid

Insert:

```
1
2 SELECT * FROM v_oncall_shifts_valid
3 WHERE employee_id = (SELECT employee_id FROM employee WHERE email='oncall.user@example.com');
4
```

Data Output Messages Notifications

shift_id	employee_id	day_of_week	start_time	end_time	escalation_order	created_at
integer	integer	integer	time without time zone	time without time zone	integer	timestamp without time zone
299977	999989740	3	09:00:00	17:00:00	1	2025-10-16 01:57:29.169497

Total rows: 1 Query complete 00:00:00.163 CRLF

```

1 SELECT * FROM oncall_shift
2 WHERE employee_id = (SELECT employee_id FROM employee WHERE email='oncall.user@example.com');
3

```

Data Output Messages Notifications

	shift_id [PK] integer	employee_id integer	day_of_week integer	start_time time without time zone	end_time time without time zone	escalation_order integer	created_at timestamp without time zone
1	299977	999989740	3	09:00:00	17:00:00	1	2025-10-16 01:57:29.169497

Update

```

1 -- 3) Update the oncall shift (valid): change start_time earlier
2 UPDATE v_oncall_shifts_valid
3 SET start_time = TIME '08:00:00'
4 WHERE shift_id = (
5     SELECT shift_id FROM oncall_shift WHERE employee_id = (SELECT employee_id FROM employee WHERE email='oncall.
6 )
7 RETURNING *;
8
9 SELECT * FROM v_oncall_shifts_valid
10 WHERE employee_id = (SELECT employee_id FROM employee WHERE email='oncall.user@example.com');
11

```

Data Output Messages Notifications

	shift_id integer	employee_id integer	day_of_week integer	start_time time without time zone	end_time time without time zone	escalation_order integer	created_at timestamp without time zone
1	299977	999989740	3	08:00:00	17:00:00	1	2025-10-16 01:57:29.169497

1	SELECT * FROM oncall_shift
2	WHERE employee_id = (SELECT employee_id FROM employee WHERE email='oncall.user@example.com');
3	

Data Output	Messages	Notifications
-------------	----------	---------------

Showing rows: 1 to 1	Page No: 1	of 1
----------------------	------------	------

src	shift_id	employee_id	day_of_week	start_time	end_time	escalation_order	created_at
After update: base	299977	999989740	3	08:00:00	17:00:00	1	2025-10-16 01:57:29.169497

Total rows: 1	Query complete 00:00:00.178	CRLF	Ln 3, Col 1
---------------	-----------------------------	------	-------------

Violations

1	DO \$\$
2	BEGIN
3	-- Attempt to insert an invalid day_of_week = 8 (out of 1..7) -> should be blocked by WITH CHECK OPTION
4	INSERT INTO v_oncall_shifts_valid (employee_id, day_of_week, start_time, end_time, escalation_order, created_at)
5	VALUES ((SELECT employee_id FROM employee WHERE email='oncall.user@example.com'), 8, TIME '09:00:00', TIME '10:00:00', 1,
6	END\$\$;
7	

Data Output	Messages	Notifications
-------------	----------	---------------

ERROR: new row for relation "oncall_shift" violates check constraint "chk_oncall_day_range"

Failing row contains (299978, 999989740, 8, 09:00:00, 10:00:00, 1, 2025-10-16 02:01:01.394975).

SQL state: 23514

Detail: Failing row contains (299978, 999989740, 8, 09:00:00, 10:00:00, 1, 2025-10-16 02:01:01.394975).

Context: SQL statement "INSERT INTO v_oncall_shifts_valid (employee_id, day_of_week, start_time, end_time, escalation_order, created_at) VALUES ((SELECT employee_id FROM employee WHERE email='oncall.user@example.com'), 8, TIME '09:00:00', TIME '10:00:00', 1, now())"

PL/pgSQL function inline_code_block line 4 at SQL statement

Total rows: 1	Query complete 00:00:00.115	CRLF	Ln 7, Col 1
---------------	-----------------------------	------	-------------

```
Data Output Messages Notifications
ERROR: new row for relation "oncall_shift" violates check constraint "chk_shift_order"
Failing row contains (299977, 999989740, 3, 18:00:00, 17:00:00, 1, 2025-10-16 01:57:29.169497).

SQL state: 23514
Detail: Failing row contains (299977, 999989740, 3, 18:00:00, 17:00:00, 1, 2025-10-16 01:57:29.169497).
Context: SQL statement "UPDATE v_oncall_shifts_valid
        SET start_time = TIME '18:00:00', end_time = TIME '17:00:00'
        WHERE employee_id = (SELECT employee_id FROM employee WHERE email='oncall.user@example.com')"
PL/pgSQL function inline_code_block line 4 at SQL statement

Total rows: 1    Query complete 00:00:00.107    CRLF    Ln 7, Col 7
```

```

1 DELETE FROM v_oncall_shifts_valid
2 WHERE shift_id = (
3     SELECT shift_id FROM oncall_shift WHERE employee_id = (SELECT employee_id FROM employee WHERE email='oncall.user@example
4 )
5 RETURNING *;
6
7 -- 6) Confirm deletion
8 SELECT 'After delete: base oncall_shift (should be empty)' AS note, * FROM oncall_shift
9 WHERE employee_id = (SELECT employee_id FROM employee WHERE email='oncall.user@example.com');
10

```

Data Output

Messages

Notifications

≡

+

▼

▼

▼

▼

▼

SQL

note

text

🔒

shift_id

[PK] integer

✎

employee_id

integer

✎

day_of_week

integer

✎

start_time

time without time zone

✎

end_time

time without time zone

✎

escalation_order

integer

✎

created_at

timestamp without time zone

✎

Total rows: 0

Query complete 00:00:00.218

CRLF

Ln 10, Col 1

insert

```

7      INSERT INTO v_current_licenses
8      (employee_id, license_name, issued_date, expiry_date, created_at)
9      VALUES (
10     (SELECT employee_id FROM employee WHERE email='charlie.license@example.com'),
11     'Senior Programmer',
12     (CURRENT_DATE - INTERVAL '365 days'),
13     (CURRENT_DATE + INTERVAL '365 days'), -- >= CURRENT_DATE
14     now()
15     )
16     RETURNING *
17 )
18 SELECT * FROM v_current_licenses
19 WHERE employee_id = (SELECT employee_id FROM employee WHERE email='charlie.license@example.com');

```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

	license_id integer	employee_id integer	employee text	license_name character varying (200)	issued_date date	expiry_date date	created_at timestamp without time zone
1	99862	999989741	Charlie License	Senior Programmer	2024-10-16	2026-10-16	2025-10-16 02:09:15.176139

Total rows: 1 Query complete 00:00:00.279 CRLF Ln 1, Col 1

```

1 SELECT * FROM employee_license
2 WHERE employee_id = (SELECT employee_id FROM employee WHERE email='charlie.license@example.com');
3

```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

	license_id [PK] integer	employee_id integer	license_name character varying (200)	issued_date date	expiry_date date	notes text	created_at timestamp without time zone
1	99862	999989741	Senior Programmer	2024-10-16	2026-10-16	[null]	2025-10-16 02:09:15.176139

Total rows: 1 Query complete 00:00:00.171 CRLF Ln 1, Col 8

Update


```

1  -- 3) Update the license via the view (extend expiry_date)
2  UPDATE v_current_licenses
3  SET expiry_date = (CURRENT_DATE + INTERVAL '730 days')
4  WHERE license_id = (
5    SELECT license_id FROM employee_license WHERE employee_id = (SELECT employee_id FROM employee WHERE email='charlie.license@example.com')
6  )
7  RETURNING *;
8
9  -- 5) Show after update
10 SELECT * FROM v_current_licenses
11 WHERE employee_id = (SELECT employee_id FROM employee WHERE email='charlie.license@example.com');
12

```

Data Output Messages Notifications							
Showing rows: 1 to 1 Page No: 1 of 1							
	license_id	employee_id	employee	license_name	issued_date	expiry_date	created_at
	integer	integer	text	character varying (200)	date	date	timestamp without time zone
1	99862	999989741	Charlie License	Senior Programmer	2024-10-16	2027-10-16	2025-10-16 02:09:15.176139

Total rows: 1 Query complete 00:00:00.201 CRLF Ln 10, Col 7

```

1  SELECT * FROM employee_license
2  WHERE employee_id = (SELECT employee_id FROM employee WHERE email='charlie.license@example.com');
3

```

Data Output Messages Notifications							
Showing rows: 1 to 1 Page No: 1 of 1							
	license_id	employee_id	license_name	issued_date	expiry_date	notes	created_at
	[PK] integer	integer	character varying (200)	date	date	text	timestamp without time zone
1	99862	999989741	Senior Programmer	2024-10-16	2027-10-16	[null]	2025-10-16 02:09:15.176139

Total rows: 1 Query complete 00:00:00.212 CRLF Ln 1, Col 7

Trigger Violations:

```

1 DO $$
2 BEGIN
3     -- 1) Try inserting a license with expiry_date < CURRENT_DATE (trigger should raise BEFORE underlying insert)
4     INSERT INTO v_current_licenses (employee_id, license_name, issued_date, expiry_date, created_at)
5     VALUES ((SELECT employee_id FROM employee WHERE email='charlie.license@example.com'),
6             'Expired Cert', CURRENT_DATE - INTERVAL '400 days', CURRENT_DATE - INTERVAL '1 day', now());
7 END$$;
8

```

Data Output Messages Notifications

ERROR: Cannot insert: expiry_date must be >= CURRENT_DATE
 CONTEXT: PL/pgSQL function trg_v_current_licenses_iud() line 6 at RAISE
 SQL statement "INSERT INTO v_current_licenses (employee_id, license_name, issued_date, expiry_date, created_at)
 VALUES ((SELECT employee_id FROM employee WHERE email='charlie.license@example.com'),
 'Expired Cert', CURRENT_DATE - INTERVAL '400 days', CURRENT_DATE - INTERVAL '1 day', now())"
 PL/pgSQL function inline_code_block line 4 at SQL statement

SQL state: P0001

Total rows: 1	Query complete 00:00:00.107	CRLF	Ln 8, Col 1
---------------	-----------------------------	------	-------------

```

1 DO $$
2 BEGIN
3     -- 2) Try inserting a license for non-existent employee (trigger should raise)
4     INSERT INTO v_current_licenses (employee_id, license_name, issued_date, expiry_date, created_at)
5     VALUES (-99999, 'Another Cert', CURRENT_DATE, CURRENT_DATE + INTERVAL '365 days', now());
6 END$$;
7

```

Data Output Messages Notifications

ERROR: Cannot insert: employee -99999 not found
 CONTEXT: PL/pgSQL function trg_v_current_licenses_iud() line 11 at RAISE
 SQL statement "INSERT INTO v_current_licenses (employee_id, license_name, issued_date, expiry_date, created_at)
 VALUES (-99999, 'Another Cert', CURRENT_DATE, CURRENT_DATE + INTERVAL '365 days', now())"
 PL/pgSQL function inline_code_block line 4 at SQL statement

SQL state: P0001

Total rows: 1	Query complete 00:00:00.129	CRLF	Ln 7, Col 1
---------------	-----------------------------	------	-------------

Delete

```

1  -- 4) Delete the license via the view
2  DELETE FROM v_current_licenses
3  WHERE license_id = (
4      SELECT license_id FROM employee_license WHERE employee_id = (SELECT employee_id FROM employee WHERE email='charlie.license
5  )
6  RETURNING *;
7
8  -- 6) Confirm deletion
9  SELECT * FROM employee_license
10 WHERE employee_id = (SELECT employee_id FROM employee WHERE email='charlie.license@example.com');
11

```

Data Output Messages Notifications

SQL

license_id [PK] integer	employee_id integer	license_name character varying (200)	issued_date date	expiry_date date	notes text	created_at timestamp without time zone
----------------------------	------------------------	---	---------------------	---------------------	---------------	---

Total rows: 0 Query complete 00:00:00.149

CRLF Ln 9, Col 8

VIEW 4: v_employee_overview

Insert

```

1  WITH ins AS (
2      INSERT INTO v_employee_overview (full_name, department, position)
3      VALUES ('Bobby B Jones', 'IT', 'Developer') -- IT & Developer exist earlier
4      RETURNING *
5  )
6  SELECT 'From view v_employee_overview' AS src, * FROM v_employee_overview
7  WHERE full_name = 'Bobby B Jones';
8

```

Data Output Messages Notifications

SQL

Showing rows: 1 to 1

Page No: 1

of 1

employee_id integer	full_name text	department character varying (200)	position character varying (200)	payments_count bigint
1	999989756 Bobby B Jones	IT	Developer	0

Total rows: 1 Query complete 00:00:00.159

CRLF Ln 2, Col 7

```

2 SELECT * FROM employee
3 WHERE first_name = 'Bobby' AND last_name = 'B Jones';
4

```

Data Output Messages Notifications								
Showing rows: 1 to 1 Page No: 1 of 1								
	employee_id [PK] integer	first_name character varying (100)	last_name character varying (100)	email character varying (255)	phone character varying (50)	address character varying (255)	birth_date date	hire_date date
1	999989756	Bobby	B Jones	bobbybjones@example.com		[null]	[null]	2025-10-16

Total rows: 1 Query complete 00:00:00.214 CRLF Ln 3, Col 48

Update

```

1 UPDATE v_employee_overview
2 SET full_name = 'Ozymanthus Jones'
3 WHERE employee_id = (
4     SELECT employee_id FROM employee WHERE first_name='Bobby' AND last_name='B Jones' LIMIT 1
5 )
6 RETURNING *;
7
8 -- 5) Show after update
9 SELECT * FROM v_employee_overview
10 WHERE employee_id = (SELECT employee_id FROM employee WHERE first_name='Ozymanthus' AND last_name='Jones');
11

```

Data Output Messages Notifications					
Showing rows: 1 to 1 Page No: 1 of 1					
	employee_id integer	full_name text	department character varying (200)	position character varying (200)	payments_count bigint
1	999989756	Ozymanthus Jones	IT	Developer	0

Total rows: 1 Query complete 00:00:00.227 CRLF Ln 10, Col 84

1	SELECT 'After update: base' AS src, * FROM employee
2	WHERE first_name='Ozymanthus' AND last_name='Jones';
3	

Data Output	Messages	Notifications
-------------	----------	---------------

Showing rows: 1 to 1	Page No: 1	of 1
----------------------	------------	------

src	employee_id	first_name	last_name	email	phone	address	birth_d
text	[PK] integer	character varying (100)	character varying (100)	character varying (255)	character varying (50)	character varying (255)	date
1	After update: base	999989756	Ozymanthus	Jones	bobbybjones@example.com	[null]	[null]

Total rows: 1	Query complete 00:00:00.160	CRLF	Ln 2, Col 26
---------------	-----------------------------	------	--------------

Trigger violations

1	DO \$\$
2	BEGIN
3	-- 1) Attempt to insert with a non-existent department -> trigger should RAISE EXCEPTION
4	INSERT INTO v_employee_overview (full_name, department, position)
5	VALUES ('No Dept', 'Nonexistent Dept', 'Developer');
6	END\$\$;
7	

Data Output	Messages	Notifications
-------------	----------	---------------

ERROR: Cannot insert: department "Nonexistent Dept" not found
CONTEXT: PL/pgSQL function trg_v_employee_overview_iud() line 30 at RAISE
SQL statement "INSERT INTO v_employee_overview (full_name, department, position)
VALUES ('No Dept', 'Nonexistent Dept', 'Developer')"
PL/pgSQL function inline_code_block line 4 at SQL statement
SQL state: P0001

Total rows: 1	Query complete 00:00:00.113	CRLF	Ln
---------------	-----------------------------	------	----

```

1 DO $$
2 BEGIN
3   -- 2) Attempt to insert with NULL/empty full_name -> trigger should RAISE EXCEPTION
4   INSERT INTO v_employee_overview (full_name, department, position)
5   VALUES ('', 'IT', 'Developer');
6 END$$;

```

Data Output Messages Notifications

ERROR: Cannot insert: full_name is required
 CONTEXT: PL/pgSQL function trg_v_employee_overview_iud() line 14 at RAISE
 SQL statement "INSERT INTO v_employee_overview (full_name, department, position)
 VALUES ('', 'IT', 'Developer')"
 PL/pgSQL function inline_code_block line 4 at SQL statement

SQL state: P0001

Total rows: Query complete 00:00:00.087

CRLF

Delete

```

1 -- 4) Delete the employee via the view
2 DELETE FROM v_employee_overview
3 WHERE employee_id = (SELECT employee_id FROM employee WHERE first_name='Ozymanthus' AND last_name='Jones' LIMIT 1)
4 RETURNING *;
5
6 -- 6) Confirm deletion
7 SELECT * FROM employee
8 WHERE first_name='Ozymanthus' AND last_name='Jones';
9

```

Data Output Messages Notifications

employee_id	first_name	last_name	email	phone	address	birth_date	hire_date				
[PK] integer	character varying (100)	character varying (100)	character varying (255)	character varying (50)	character varying (255)	date	date				

Total rows: 0 Query complete 00:00:00.413

CRLF Ln 8, Col 30