

## 1. Rules (Declarative Constraints): Safe Prescriptions

## 2. Active Databases (E–C–A Trigger): Bill Totals That Stay Correct

```
90
91 -- Test
92 INSERT INTO bill VALUES (1,0);
93 INSERT INTO bill_item VALUES (1,100,now()), (1,200,now());
94 UPDATE bill_item SET amount=300 WHERE amount=100;
95 DELETE FROM bill_item WHERE amount=200;
96 SELECT * FROM bill;
97 SELECT * FROM bill_audit;
98
```

Data Output Messages Notifications

	bill_id integer	old_total numeric (12,2)	new_total numeric (12,2)	changed_at timestamp without time zone
1	1	[null]	300.00	2025-10-30 23:08:54.81414
2	1	[null]	500.00	2025-10-30 23:08:54.81414
3	1	[null]	300.00	2025-10-30 23:08:54.81414

## 3. Deductive Databases (Recursive WITH): Referral/Supervision Chain

```

116 -- Recursive query
117 WITH RECURSIVE supers(emp, sup, hops, path) AS (
118     SELECT employee, supervisor, 1, employee::TEXT || '>' || supervisor
119     FROM staff_supervisor
120     UNION ALL
121     SELECT s.employee, t.sup, hops + 1, path || '>' || t.sup
122     FROM staff_supervisor s
123     JOIN supers t ON s.supervisor = t.emp
124     WHERE POSITION(t.sup IN path) = 0 -- prevent cycles
125 )
126 SELECT emp, sup AS top_supervisor, MAX(hops) AS hops
127 FROM supers
128 GROUP BY emp, sup
129 ORDER BY emp;
130

```

Data Output Messages Notifications

SQL

	emp character varying (50)	top_supervisor character varying (50)	hops integer
1	Alice	Bob	1
2	Bob	Carol	1
3	Carol	Diana	1
4	Eve	Bob	1
5	Frank	Eve	1

## 4. Knowledge Bases (Triples & Ontology): Infectious-Disease Roll-Up

```
152 -- Compute transitive closure of isa
153 WITH RECURSIVE isa(child, ancestor) AS (
154     SELECT s, o FROM triple WHERE p = 'isa'
155     UNION
156     SELECT t.s, i.ancestor
157     FROM triple t
158     JOIN isa i ON t.o = i.child
159     WHERE t.p = 'isa'
160 ),
161 infectious_patients AS (
162     SELECT DISTINCT t.s
163     FROM triple t
164     JOIN isa i ON t.o = i.child
165     WHERE t.p = 'hasDiagnosis'
166           AND i.ancestor = 'InfectiousDisease'
167 )
168 SELECT s AS patient_id FROM infectious_patients;
169
```

Data Output Messages Notifications

	patient_id character varying (100) 🔒
1	Patient1
2	Patient2

## 5. Spatial Databases (PostGIS): Radius & Nearest-3

Spatial Database task for Oracle using SDO\_GEOMETRY.

--The issues in the starter are:

--Wrong SRID – should be 4326 (WGS84) instead of 3857. Lat/Lon order swapped – Oracle expects (X=longitude, Y=latitude).

--distance units missing – need 'unit=KM'.

```

-- placeholder :AMB_POINT – must define the ambulance location as a
-- SDO_GEOMETRY point.

--Corrected Query

-- Define ambulance location (example coordinates)

VAR AMB_POINT SDO_GEOMETRY;

EXEC :AMB_POINT := SDO_GEOMETRY(2001, 4326, SDO_POINT_TYPE(-73.935242,
40.730610, NULL), NULL, NULL); -- Example: New York City

-- Find nearest hospital within 10 km

SELECT HOSPITAL_ID, NAME, ADDRESS,

        SDO_NN_DISTANCE(1) AS DISTANCE_KM

FROM HOSPITAL

WHERE SDO_NN(LOCATION, :AMB_POINT, 'sdo_num_res=1') = 'TRUE'

AND SDO_WITHIN_DISTANCE(LOCATION, :AMB_POINT, 'distance=10 unit=KM') =
'TRUE';

ORDER BY DISTANCE_KM;

-- Limit to 1 nearest hospital

FETCH FIRST 1 ROW ONLY;

```