

Post-class Exercises: Nested Queries

Schema and data

We will work on the **DineSafe** database, which is a collection of food safety inspections carried out between 2010 and 2012 in Toronto.

- (a) Find out the establishment that has received the highest amount of fine for a single infraction. Report the name and the address of the establishment, as well as the amount of the fine. If there are ties, report for each of them.

Solution:

```
SELECT name, addr, fine
FROM establishment, inspection, infraction
WHERE establishment.id = inspection.est_id
  AND inspection.id = infraction.insp_id
  AND fine = (
  SELECT MAX(fine)
  FROM infraction
);
```

Output:

name	addr	fine
FARM FRESH SUPERMARKET	4466 SHEPPARD AVE E	31250

(1 row)

- (b) Find out every establishment that has ever paid any fine because of some infraction, ordered by their names alphabetically.

Solution:

```
SELECT name, addr
FROM establishment
WHERE id IN (
  SELECT est_id
  FROM inspection, infraction
  WHERE inspection.id = infraction.insp_id
  AND fine > 0
)
ORDER BY name;
```

Output:

name	addr
786 HALAL RESTAURANT	1330 GERRARD ST E
ABC BAKERY AND COFFEE	3618 VICTORIA PARK AVE
AJI SAI SUSHI JAPANESE RESTAURANT	813 YONGE ST
AKIA KING CAFE	387 BROADVIEW AVE
AL LAGO RISTORANTE	3423 LAKE SHORE BLVD W

```

.....
YOGI NOODLES | 2301 BRIMLEY RD
ZAKOPANE DELI | 3061 LAKE SHORE BLVD W
ZAZA SOUTHERN ITALIAN TRADITIONAL EXPRESSO BAR | 75 YORKVILLE AVE
ZUPA'S RESTAURANT & DELI | 342 ADELAIDE ST W
(186 rows)

```

(c) Find out the establishment which a total amount of fine over \$3000 in the history.

Solution:

```

SELECT name, addr, totalFine
FROM establishment, (
  SELECT est_id, SUM(fine) as totalFine
  FROM inspection, infraction
  WHERE inspection.id = infraction.insp_id
  AND fine > 0
  GROUP BY est_id
) AS R1
WHERE id = est_id AND totalFine > 3000;

```

Output:

name	addr	totalfine
CHINA CITY SUPERMARKET	247 SPADINA AVE	3125
EDUARDAS BBQ POULTRY TAKE OUT AND FOOD STUFF	1546 DUPONT ST	3067.5
HO-LEE CHOW	3466 DUNDAS ST W	6095
FARM FRESH SUPERMARKET	4466 SHEPPARD AVE E	36250
MASHION BAKERY	345 SPADINA AVE	4330

(5 rows)

2. Fine the top 5 most popular type of establishment and the number of such establishment on Bloor Street West (Its address contains "BLOOR ST W").

Solution:

```

SELECT description, cnt
FROM est_type NATURAL JOIN (
  SELECT type_id AS id, count(*) as cnt
  FROM establishment
  WHERE addr LIKE '%BLOOR ST W%'
  GROUP BY type_id
  ORDER BY cnt DESC LIMIT 1
) AS R;

```

Output:

description	cnt
Restaurant	335
Food Store (Convenience / Variety)	96
Food Take Out	70

```

Supermarket          | 25
Food Court Vendor   | 17
(5 rows)

```

3. Find the top 10 establishment that has the most infractions in one single inspection. Report the name and the address of the establishment, as well as the number of infractions in that inspection. If there are ties, report for each of them.

Solution:

```

SELECT name, addr, cnt
FROM establishment, inspection, (
  SELECT insp_id, count(*) AS cnt
  FROM infraction
  GROUP BY insp_id
  ORDER BY cnt DESC LIMIT 10
) AS R
WHERE establishment.id = inspection.est_id
  AND inspection.id = R.insp_id;

```

Output:

name	addr	cnt
TRE MARI BAKERY	1311 ST CLAIR AVE W	24
B TRUST SUPERMARKET	1105 WILSON AVE	19
MARIKO JAPANESE RESTAURANT	551 BLOOR ST W	19
AMAYA BREAD BAR	3305 YONGE ST	18
PHILTHY MCNASTYS	130 EGLINTON AVE E	18
FROSHIGHA KABUL HALAL MEAT	1067 DANFORTH AVE	18
CUISINE OF INDIA CATERING	40 MAGNETIC DR	18
MI PHO NHA TRANG	1365 WILSON AVE	18
MANGIA MANGIA MANGIA RESTAURANT	4700 KEELE ST	17
GOLDEN PIZZA RESTAURANT	1201 BROADVIEW AVE	17

(10 rows)

4. In this query, we want to study if the minimal annual inspections are properly conducted. We will break this complicated query into some small steps, each of them involves using a nested query itself. You will find it handy to handle these complicated queries using views if possible.

- (a) Find out the establishment that was never inspected in year 2010. Report their names and addresses, ordered by their names alphabetically. Hint: for PostgreSQL, use `extract(year from date)` to extract the year from the date; for SQLite, need to use `STRFTIME('%Y', date)`.

Solution:

```

SELECT name, addr
FROM establishment
WHERE id NOT IN (
  SELECT est_id
  FROM inspection
  WHERE extract(year from date) = 2010
)
ORDER BY name;

```

Output:

name	addr
1 HOUR CAFETERIA	435 SPADINA AVE
1 PLUS 3 PIZZA & WINGS	1798 JANE ST
100 KM FOODS INC.	4478 CHESSWOOD DR
1000 VARIETY	1000 PAPE AVE
.....	
ZLD	1120 BELLAMY RD N
ZOBEL	1160 DANFORTH AVE
ZUCCA TRATTORIA	2150 YONGE ST
iBENTO	235 COLLEGE ST

(5250 rows)

- (b) For each establishment, find out the actual inspections happened in year 2010. Hint: don't forget to include those with zero inspection. Verify your result by checking whether the number of returned rows equals to the number of rows in the establishment relation.

Solution:

```

SELECT name, addr, cnt
FROM establishment INNER JOIN (
  (SELECT est_id, COUNT(*) AS cnt
   FROM inspection INNER JOIN establishment
     ON inspection.est_id = establishment.id
   WHERE extract(year from date) = 2010
   GROUP BY est_id
  ) UNION (
    SELECT id, 0 AS cnt
    FROM establishment
    WHERE id NOT IN (
      SELECT est_id
      FROM inspection
      WHERE extract(year from date) = 2010
    )
  )
) AS R
  ON establishment.id = R.est_id
ORDER BY name;

```

Output:

name	addr	cnt
(THE) HOLE IN THE WALL IN THE JUNCTION	2867 DUNDAS ST W	1
1 HOUR CAFETERIA	435 SPADINA AVE	0
1 PLUS 1 PIZZA & WINGS	361 OAKWOOD AVE	1
1 PLUS 2 PIZZA & WING	3260 DUNDAS ST W	1
.....		
ZYKA FINE FOODS & BBQ	2535 WARDEN AVE	2
ZYNG	730 YONGE ST	2
iBENTO	235 COLLEGE ST	0

(13921 rows)

- (c) Find out every establishment that has ever failed to meet the minimal number of required annual inspections in year 2010, ordered by name alphabetically. For each of them, report its name, address, minimal number of required annual inspections, and actual inspections conducted in 2010. Hint: start your work from the results in Q4(b).

Solution:

```
SELECT name, addr, minspec, cnt
FROM establishment, (
  (SELECT est_id, COUNT(*) AS cnt
   FROM inspection INNER JOIN establishment
    ON inspection.est_id = establishment.id
   WHERE extract(year from date) = 2010
   GROUP BY est_id
  ) UNION (
   SELECT id, 0 AS cnt
   FROM establishment
   WHERE id NOT IN (
     SELECT est_id
     FROM inspection
     WHERE extract(year from date) = 2010
   )
  ) AS R
WHERE minspec > cnt AND id = est_id
ORDER BY name;
```

Output:

name	addr	minspec	cnt
(THE) HOLE IN THE WALL IN THE JUNCTION	2867 DUNDAS ST W	2	1
1 HOUR CAFETERIA	435 SPADINA AVE	2	0
1 PLUS 1 PIZZA & WINGS	361 OAKWOOD AVE	2	1
1 PLUS 2 PIZZA & WING	3260 DUNDAS ST W	2	1
1 PLUS 3 PIZZA & WINGS	1798 JANE ST	2	0
.....			
ZUPA'S RESTAURANT & DELI	342 ADELAIDE ST W	2	1
ZYKA FINE FOODS & BBQ	2535 WARDEN AVE	3	2
ZYNG	730 YONGE ST	3	2
iBENTO	235 COLLEGE ST	2	0

(12112 rows)

5. According to DineSafe Inspection and Disclosure System, a “Conditional Pass” notice will be issued when one or more significant infractions are observed during an inspection, and a second re-inspection will be conducted within 24-48 hours. Find out the number of times that an establishment was issued a “Conditional Pass” notice at first inspection, but no re-inspection was conducted within 24-48 hours (holes in execution!). Hint: for PostgreSQL, use `extract(day from date)` to extract the day from the date; for SQLite, need to use `julianday(R2.date)`.

Solution:

```

SELECT count(*)
FROM inspection AS R1, est_status
WHERE status_id = est_status.id
  AND description = 'Conditional Pass'
  AND NOT EXISTS (
  SELECT *
  FROM inspection AS R2
  WHERE R2.est_id = R1.est_id
  AND extract(day from R2.date) - extract(day from R1.date) >= 1
  AND extract(day from R2.date) - extract(day from R1.date) <= 2
);

```

Output:

```

count
-----
1739
(1 row)

```

6. In the database, you may have found several establishment have the same name but in different locations. These are franchise stores or chain stores, e.g., McDonald's.

- (a) Find the name of the top 5 largest chains in the database. For each of them, report its name and the number of stores it owns in Toronto, ordered by the number in descending order.

Solution:

```

SELECT name, count(*) as stores
FROM establishment
GROUP BY name
HAVING count(*) > 1
ORDER BY stores DESC LIMIT 5;

```

Output:

name	stores
TIM HORTONS	203
SUBWAY	169
PIZZA PIZZA	107
SHOPPERS DRUG MART	73
SECOND CUP	66

(5 rows)

- (b) For each of the chain brand in (a), report the average minimal annual inspections per store, in the same order as in (a).

Solution:

```

SELECT name, SUM(minspec) * 1.0 / stores AS avgMinSpec
FROM establishment NATURAL JOIN(
  SELECT name, count(*) as stores

```

```

FROM establishment
GROUP BY name
HAVING count(*) > 1
ORDER BY stores DESC LIMIT 5
) AS R
GROUP BY name, stores
ORDER BY stores;

```

Output:

name	avgminspec
SECOND CUP	1.4696969696969697
SHOPPERS DRUG MART	1.0000000000000000
PIZZA PIZZA	2.0093457943925234
SUBWAY	2.0059171597633136
TIM HORTONS	1.9901477832512315

(5 rows)

- (c) Find out if any of the chain brand in (a), has received any inspections with a “Closed” status in the history. If yes, report the name, the number of “Closed” inspections for each of such chain brand.

Solution:

```

SELECT name, COUNT(*) AS cnt
FROM inspection INNER JOIN establishment
ON inspection.est_id = establishment.id
NATURAL JOIN (
SELECT name, count(*) as stores
FROM establishment
GROUP BY name
HAVING count(*) > 1
ORDER BY stores DESC LIMIT 5
) AS R
WHERE status_id = (
SELECT id
FROM est_status
WHERE description = 'Closed'
)
GROUP BY name, stores
ORDER BY stores;

```

Output:

name	cnt
SUBWAY	1

(1 row)

7. (a) Find the pass rate of “SUSHI WORLD” on COLLEGE ST. Report the total number of passed inspections, total number of previous inspections, and the pass rate in its history.

Solution:

```

SELECT name, addr, passes, specs, passes * 1.0 / specs AS rate
FROM (
  SELECT est_id, count(*) AS passes
  FROM inspection, est_status
  WHERE status_id = est_status.id
  AND description = 'Pass'
  GROUP BY est_id
) AS R1 NATURAL JOIN (
  SELECT est_id, count(*) AS specs
  FROM inspection
  GROUP BY est_id
) AS R2, establishment
WHERE est_id = id
  AND name = 'SUSHI WORLD'
  AND addr LIKE '%COLLEGE ST%';

```

Output:

name	addr	passes	specs	rate
SUSHI WORLD	281 COLLEGE ST	5	5	1.00000000000000000000

(1 row)

- (b) **A very hard one:** For each establishment, find its pass rate among all its previous inspections in the database. Report the name, address, total number of passed inspections, total number of previous inspections, and the pass rate. Order your results in an descending order of the pass rate, followed by the total number of inspections (check some of your favorite restaurants!). Verify that the number of rows in your results equals to the number of establishment in the database.

Solution: The follow query will only return those establishment with recorded passes.

```

SELECT name, addr, passes, specs, passes * 1.0 / specs AS rate
FROM (
  SELECT est_id, count(*) AS passes
  FROM inspection, est_status
  WHERE status_id = est_status.id
  AND description = 'Pass'
  GROUP BY est_id
) AS R1 NATURAL JOIN (
  SELECT est_id, count(*) AS specs
  FROM inspection
  GROUP BY est_id
) AS R2, establishment
WHERE est_id = id
ORDER BY rate DESC, specs DESC;

```

Need to union those with no recorded passes as well.

```

SELECT name, addr, passes, specs, passes * 1.0 / specs AS rate
FROM (
  (SELECT est_id, count(*) AS passes
  FROM inspection, est_status

```



```

WHERE status_id = est_status.id
AND description = 'Pass'
GROUP BY est_id
) UNION (
SELECT est_id, 0 AS passes
FROM inspection
WHERE est_id NOT IN (
SELECT est_id
FROM inspection, est_status
WHERE status_id = est_status.id
AND description = 'Pass'
))) AS R1 NATURAL JOIN (
SELECT est_id, count(*) AS specs
FROM inspection
GROUP BY est_id
) AS R2, establishment
WHERE est_id = id
ORDER BY rate DESC, specs DESC;

```

Output:

name	addr	passes	specs	rate
FRESH & WILD	69 SPADINA AVE	10	10	1.000
LITTLE INDIA RESTAURANT	255 QUEEN ST W	10	10	1.000
ZELLERS RESTAURANT	500 REXDALE BLVD	9	9	1.000
MANCHU WOK	1800 SHEPPARD AVE E	9	9	1.000
BOSTON PIZZA	16 LESLIE ST	8	8	1.000
.....				
HOT DOG CART	80 TORO RD	0	1	0.000
OAKDALE PARK MIDDLE SCHO...	315 GRANDRAVINE DR	0	1	0.000
OMM KALSOU	2175 LAWRENCE AVE E	0	1	0.000
BENTO	3501 YONGE ST	0	1	0.000

(13921 rows)

8. Which rule has been violated the most?

Solution:

```

SELECT rule.*, cnt
FROM rule, (
SELECT rule_id, count(*) AS cnt
FROM infraction
GROUP BY rule_id
) AS R1
WHERE id = rule_id AND NOT EXISTS (
SELECT rule_id, count(*)
FROM infraction
GROUP BY rule_id
HAVING count(*) > cnt
);

```

Output:

id	code	sec	subsec	par	subpar	cnt
5	0. Reg. 562/90	19				3358

(1 row)