

sql trigger.sql make-database.ipynb U queries.sql M ... Results 2 X Results 3 Results 4 Results 5 Results 8 Results 9 Results 10 ...

sql-project > queries.sql

```
1 USE hospital;
2 -- 1.
3 -- Find the physicians (ssn) who have most prescribed drugs which can
4 -- (due to possible adverse interaction with a previously prescribed
5 WITH alert_rankings AS (
6     SELECT physician_id, DENSE_RANK() OVER(
7         ORDER BY COUNT(*) DESC
8     ) as alert_rank
9     FROM alerts
10    GROUP BY physician_id
11 )
12
13 SELECT physician_id
14 FROM alert_rankings
15 WHERE alert_rank = 1;
16
17 -- 2.
18 -- Find the physicians (ssn) who have prescribed two drugs to the same
19
20 SELECT DISTINCT p1.physician_id
21 FROM prescriptions as p1,
22      prescriptions as p2
23 WHERE p1.physician_id = p2.physician_id
24 AND p1.drug_name != p2.drug_name
25 AND p1.patient_id = p2.patient_id
26 AND p2.drug_name IN (
27     SELECT drug_name_2
28     FROM adverse_interactions
29     WHERE drug_name = p1.drug_name
30 );
31
32 -- 3.
33 -- Find the physicians who have prescribed most drugs supplied by con
34 WITH drugxo_fills AS (
35     SELECT physician_id, COUNT(*) as cnt
36     FROM pharmacy_fills as pf, contracts as con,
37          companies as comp, prescriptions as p
38     WHERE con.company_id = comp.id
39     AND pf.pharmacy_id = con.pharmacy_id
40     AND comp.name LIKE "%DRUGXO%"
41     AND pf.prescription_id = p.id
42     GROUP BY p.physician_id
43 )
44
45 SELECT physician_id, DENSE_RANK() OVER (
```

physician\_id

156-28-1945
571-13-9020

data-1050

main\* ↻ ⌂ 0 ▲ 0 localhost hospital Go Live



.sql

trigger.sql

make-database.ipynb

queries.sql M

Results 2

Results 3

Results 4

Results 5

Results 8

Results 9

Results

```

sql-project > queries.sql
1 USE hospital;
2 -- 1.
3 -- Find the physicians (ssn) who have most prescribed drugs which can
4 -- (due to possible adverse interaction with a previously prescribed
5 WITH alert_rankings AS (
6     SELECT physician_id, DENSE_RANK() OVER(
7         ORDER BY COUNT(*) DESC
8     ) as alert_rank
9     FROM alerts
10    GROUP BY physician_id
11 )
12
13 SELECT physician_id
14 FROM alert_rankings
15 WHERE alert_rank = 1;
16
17 -- 2.
18 -- Find the physicians (ssn) who have prescribed two drugs to the same
19
20 SELECT DISTINCT p1.physician_id
21 FROM prescriptions as p1,
22      prescriptions as p2
23 WHERE p1.physician_id = p2.physician_id
24 AND p1.drug_name != p2.drug_name
25 AND p1.patient_id = p2.patient_id
26 AND p2.drug_name IN (
27     SELECT drug_name_2
28     FROM adverse_interactions
29     WHERE drug_name = p1.drug_name
30 );
31
32 -- 3.
33 -- Find the physicians who have prescribed most drugs supplied by con
34 WITH drugxo_fills AS (
35     SELECT physician_id, COUNT(*) as cnt
36     FROM pharmacy_fills as pf, contracts as con,
37          companies as comp, prescriptions as p
38     WHERE con.company_id = comp.id
39     AND pf.pharmacy_id = con.pharmacy_id
40     AND comp.name LIKE "%DRUGXO%"
41     AND pf.prescription_id = p.id
42     GROUP BY p.physician_id
43 )
44
45 SELECT physician_id, DENSE_RANK() OVER (

```

physician_id
156-28-1945
571-13-9020
614-57-6885



sql trigger.sql make-database.ipynb U queries.sql M ...
Results 2 Results 3 Results 4 X Results 5 Results 8 Results 9 Results ...

sql-project > queries.sql

```

1 USE hospital;
2 -- 1.
3 -- Find the physicians (ssn) who have most prescribed drugs which can
4 -- (due to possible adverse interaction with a previously prescribed
5 WITH alert_rankings AS (
6     SELECT physician_id, DENSE_RANK() OVER(
7         ORDER BY COUNT(*) DESC
8     ) as alert_rank
9     FROM alerts
10    GROUP BY physician_id
11 )
12
13 SELECT physician_id
14 FROM alert_rankings
15 WHERE alert_rank = 1;
16
17 -- 2.
18 -- Find the physicians (ssn) who have prescribed two drugs to the same
19
20 SELECT DISTINCT p1.physician_id
21 FROM prescriptions as p1,
22      prescriptions as p2
23 WHERE p1.physician_id = p2.physician_id
24 AND p1.drug_name != p2.drug_name
25 AND p1.patient_id = p2.patient_id
26 AND p2.drug_name IN (
27     SELECT drug_name_2
28     FROM adverse_interactions
29     WHERE drug_name = p1.drug_name
30 );
31
32 -- 3.
33 -- Find the physicians who have prescribed most drugs supplied by companies
34 WITH drugxo_fills AS (
35     SELECT physician_id, COUNT(*) as cnt
36     FROM pharmacy_fills as pf, contracts as con,
37          companies as comp, prescriptions as p
38     WHERE con.company_id = comp.id
39     AND pf.pharmacy_id = con.pharmacy_id
40     AND comp.name LIKE "%DRUGXO%"
41     AND pf.prescription_id = p.id
42     GROUP BY p.physician_id
43 )
44
45 SELECT physician_id, DENSE_RANK() OVER (

```

physician_id	drugxo_rank
118-66-5958	1
614-57-6885	2
156-28-1945	2

sql trigger.sql make-database.ipynb U queries.sql M ...
Results 2 Results 3 Results 4 Results 5 X Results 8 Results 9 Results 10 ...

```

sql-project > queries.sql
1 USE hospital;
2 -- 1.
3 -- Find the physicians (ssn) who have most prescribed drugs which can
4 -- (due to possible adverse interaction with a previously prescribed
5 WITH alert_rankings AS (
6     SELECT physician_id, DENSE_RANK() OVER(
7         ORDER BY COUNT(*) DESC
8     ) as alert_rank
9     FROM alerts
10    GROUP BY physician_id
11 )
12
13 SELECT physician_id
14 FROM alert_rankings
15 WHERE alert_rank = 1;
16
17 -- 2.
18 -- Find the physicians (ssn) who have prescribed two drugs to the same
19
20 SELECT DISTINCT p1.physician_id
21 FROM prescriptions as p1,
22      prescriptions as p2
23 WHERE p1.physician_id = p2.physician_id
24 AND p1.drug_name != p2.drug_name
25 AND p1.patient_id = p2.patient_id
26 AND p2.drug_name IN (
27     SELECT drug_name_2
28     FROM adverse_interactions
29     WHERE drug_name = p1.drug_name
30 );
31
32 -- 3.
33 -- Find the physicians who have prescribed most drugs supplied by con
34 WITH drugxo_fills AS (
35     SELECT physician_id, COUNT(*) as cnt
36     FROM pharmacy_fills as pf, contracts as con,
37          companies as comp, prescriptions as p
38     WHERE con.company_id = comp.id
39     AND pf.pharmacy_id = con.pharmacy_id
40     AND comp.name LIKE "%DRUGXO%"
41     AND pf.prescription_id = p.id
42     GROUP BY p.physician_id
43 )
44
45 SELECT physician_id, DENSE_RANK() OVER (

```

drug_name	price_per_unit	avg_price_per_unit
Glucozepam Amcipientin	1.8125	1.25625
Glucozepam Amcipientin	0.7	1.25625

main\* ↻ ⌂ 0 ⌃ 0 localhost hospital
Go Live

 data-1050

□ □ □ □

```
sql trigger.sql make-database.ipynb U queries.sql M ...
sql-project > queries.sql

1 USE hospital;
2 -- 1.
3 -- Find the physicians (ssn) who have most prescribed drugs which can
4 -- (due to possible adverse interaction with a previously prescribed
5 WITH alert_rankings AS (
6     SELECT physician_id, DENSE_RANK() OVER(
7         ORDER BY COUNT(*) DESC
8     ) as alert_rank
9     FROM alerts
10    GROUP BY physician_id
11 )
12
13 SELECT physician_id
14 FROM alert_rankings
15 WHERE alert_rank = 1;
16
17 -- 2.
18 -- Find the physicians (ssn) who have prescribed two drugs to the same
19
20 SELECT DISTINCT p1.physician_id
21 FROM prescriptions as p1,
22      prescriptions as p2
23 WHERE p1.physician_id = p2.physician_id
24 AND p1.drug_name != p2.drug_name
25 AND p1.patient_id = p2.patient_id
26 AND p2.drug_name IN (
27     SELECT drug_name_2
28     FROM adverse_interactions
29     WHERE drug_name = p1.drug_name
30 );
31
32 -- 3.
33 -- Find the physicians who have prescribed most drugs supplied by con
34 WITH drugxo_fills AS (
35     SELECT physician_id, COUNT(*) as cnt
36     FROM pharmacy_fills as pf, contracts as con,
37          companies as comp, prescriptions as p
38     WHERE con.company_id = comp.id
39     AND pf.pharmacy_id = con.pharmacy_id
40     AND comp.name LIKE "%DRUGXO%"
41     AND pf.prescription_id = p.id
42     GROUP BY p.physician_id
43 )
44
45 SELECT physician_id, DENSE_RANK() OVER (
```

pharmacy_id	drug_name	markup
4	Kanulin	9.1785714286



A screenshot of a Java code editor showing a class named `Game`. The code includes imports for `java.awt`, `java.awt.event`, and `java.util`. It defines a constructor, a `start` method, and a `main` method. The `start` method initializes a `Panel` and a `Frame`, setting the title to "Tic Tac Toe". The `main` method creates a `Game` object and calls its `start` method.

```
import java.awt.*;
import java.awt.event.*;
import java.util.*;

public class Game {
    public Game() {
        start();
    }

    public void start() {
        Panel panel = new Panel();
        Frame frame = new Frame("Tic Tac Toe");
        frame.add(panel);
        frame.pack();
        frame.setVisible(true);
    }

    public static void main(String[] args) {
        new Game();
    }
}
```

sql trigger.sql make-database.ipynb U queries.sql M ...
Results 2 Results 3 Results 4 Results 5 Results 8 Results 9 Results ...

```

sql-project > queries.sql
1 USE hospital;
2 -- 1.
3 -- Find the physicians (ssn) who have most prescribed drugs which can
4 -- (due to possible adverse interaction with a previously prescribed
5 WITH alert_rankings AS (
6     SELECT physician_id, DENSE_RANK() OVER(
7         ORDER BY COUNT(*) DESC
8     ) as alert_rank
9     FROM alerts
10    GROUP BY physician_id
11 )
12
13 SELECT physician_id
14 FROM alert_rankings
15 WHERE alert_rank = 1;
16
17 -- 2.
18 -- Find the physicians (ssn) who have prescribed two drugs to the same
19
20 SELECT DISTINCT p1.physician_id
21 FROM prescriptions as p1,
22      prescriptions as p2
23 WHERE p1.physician_id = p2.physician_id
24 AND p1.drug_name != p2.drug_name
25 AND p1.patient_id = p2.patient_id
26 AND p2.drug_name IN (
27     SELECT drug_name_2
28     FROM adverse_interactions
29     WHERE drug_name = p1.drug_name
30 );
31
32 -- 3.
33 -- Find the physicians who have prescribed most drugs supplied by con
34 WITH drugxo_fills AS (
35     SELECT physician_id, COUNT(*) as cnt
36     FROM pharmacy_fills as pf, contracts as con,
37          companies as comp, prescriptions as p
38     WHERE con.company_id = comp.id
39     AND pf.pharmacy_id = con.pharmacy_id
40     AND comp.name LIKE "%DRUGXO%"
41     AND pf.prescription_id = p.id
42     GROUP BY p.physician_id
43 )
44
45 SELECT physician_id, DENSE_RANK() OVER (

```

drug_name	avg_diff
Avafoxin	1.75
Cleotrana	1.3333
Primalovir	0
Glucozepam Amcipientin	1.5
Olanzanafine	2
Abnazole Toleluble	4
Dantopex Quixilinum	2
Quixiposide	2
Kanulin	0

main\* ↻ ⌂ 0 ⌃ 0 localhost hospital
Go Live

sql trigger.sql make-database.ipynb queries.sql ...

sql-project > queries.sql

```

1 USE hospital;
2 -- 1.
3 -- Find the physicians (ssn) who have most prescribed drugs which can
4 -- (due to possible adverse interaction with a previously prescribed
5 WITH alert_rankings AS (
6     SELECT physician_id, DENSE_RANK() OVER(
7         ORDER BY COUNT(*) DESC
8     ) as alert_rank
9     FROM alerts
10    GROUP BY physician_id
11 )
12
13 SELECT physician_id
14 FROM alert_rankings
15 WHERE alert_rank = 1;
16
17 -- 2.
18 -- Find the physicians (ssn) who have prescribed two drugs to the same
19
20 SELECT DISTINCT p1.physician_id
21 FROM prescriptions as p1,
22      prescriptions as p2
23 WHERE p1.physician_id = p2.physician_id
24 AND p1.drug_name != p2.drug_name
25 AND p1.patient_id = p2.patient_id
26 AND p2.drug_name IN (
27     SELECT drug_name_2
28     FROM adverse_interactions
29     WHERE drug_name = p1.drug_name
30 );
31
32 -- 3.
33 -- Find the physicians who have prescribed most drugs supplied by con
34 WITH drugxo_fills AS (
35     SELECT physician_id, COUNT(*) as cnt
36     FROM pharmacy_fills as pf, contracts as con,
37          companies as comp, prescriptions as p
38     WHERE con.company_id = comp.id
39     AND pf.pharmacy_id = con.pharmacy_id
40     AND comp.name LIKE "%DRUGXO%"
41     AND pf.prescription_id = p.id
42     GROUP BY p.physician_id
43 )
44
45 SELECT physician_id, DENSE_RANK() OVER (

```

Results 2 Results 3 Results 4 Results 5 Results 8 Results 9 Results 10

name	drug_name
Springfield Pharmacy	Cleotrana
Springfield Pharmacy	Primalovir
Peachtree Meds	Cleotrana
Lone Star Drugs	Glucozepam Amcipientin
Emerald City Pharmacy	Abnazole Toleluble
Emerald City Pharmacy	Avafoxin
Golden Gate Drugs	Avafoxin
Sunshine Pharmacy	Primalovir
Sunshine Pharmacy	Quixiposide
Liberty Meds	Avafoxin
Lakeside Drugs	Avafoxin
Desert Bloom Pharmacy	Olanzanafine
Bayside Pharmacy	Glucozepam Amcipientin
Capital Meds	Dantopex Quixilinum
Beantown Drugs	Cleotrana



main\* 0 0 localhost hospital Go Live