Введение в базы данных

01 Для чего нужна база данных

Практические задания отсутствуют

02 СУБД — великий библиотекарь

Практические задания отсутствуют

03 Язык запросов SQL

Практические задания отсутствуют

04 Первый запрос

Задача 1 / 3

```
SELECT * FROM media_type
```

Задача 2 / 3

```
SELECT * FROM playlist
```

Задача 3 / 3

```
SELECT * FROM invoice
```

05 Стиль запроса

Практические задания отсутствуют

Срезы данных в SQL

Как ограничить выборку

Задача 1 / 3

```
SELECT

email, first_name, last_name

FROM client;
```

Задача 2 / 3

```
SELECT

invoice_id, customer_id, invoice_date,billing_address

FROM invoice

LIMIT 5;
```

Задача 3 / 3

```
SELECT

name, unit_price

FROM track

LIMIT 20;
```

Типы данных в PostgreSQL

Практические задания отсутствуют

Изменение типов данных

Задача 1 /3

```
SELECT CAST(milliseconds AS varchar),

CAST(bytes AS varchar)

FROM track;
```

Задача 2 / 3

```
SELECT CAST(total AS int)

FROM invoice;
```

Задача 3 / 3

```
SELECT CAST(birth_date AS date)

FROM staff;
```

Оператор WHERE. Операторы сравнения

Практические задания отсутствуют

Срез данных с помощью оператора WHERE

Задача 1 / 3

```
SELECT *

FROM invoice_line

WHERE unit_price > 0.99;
```

Задача 2 / 3

```
SELECT first_name, last_name, city

FROM client

WHERE country = 'Brazil';
```

Задача 3 / 3

```
SELECT billing_address, CAST(invoice_date AS date)

FROM invoice
```

```
WHERE total > 8;
```

Как работать с диаграммой и описанием базы данных

Практические задания отсутствуют

Логические операторы

Практические задания отсутствуют

Срез данных с помощью логических операторов

Задача 1 / 6

```
SELECT total, customer_id

FROM invoice

WHERE billing_city = 'Dublin' OR billing_city = 'London' OR billing_city = 'Paris';
```

Задача 2 / 6

```
SELECT total, customer_id

FROM invoice

WHERE total >= 5 AND (customer_id = 40 OR customer_id = 46);
```

Задача 3 / 6

```
SELECT total, customer_id

FROM invoice

WHERE (billing_city = 'Dublin' OR billing_city = 'London' OR billing_city = 'Paris')

AND (total >= 5 AND (customer_id = 40 OR customer_id = 46));
```

Задача 4 / 6

```
SELECT last_name, phone

FROM client

WHERE support_rep_id = 3 AND (country = 'USA' OR country = 'France');
```

Задача 5 / 6

```
SELECT title

FROM movie

WHERE rental_rate < 2

AND rental_duration > 6

AND rating <> 'PG'

AND rating <> 'PG-13';
```

Задача 6 / 6

```
SELECT billing_address, billing_city

FROM invoice
```

```
WHERE total > 2

AND invoice_date > '2009-08-31 00:00:00'

AND invoice_date < '2009-10-01 00:00:00'

AND billing_country <> 'USA'

AND billing_country <> 'Brazil';
```

Специальные операторы в условиях

Практические задания отсутствуют

Операторы IN, LIKE, BETWEEN

Задача 1 / 6

```
SELECT name

FROM playlist

WHERE name LIKE '%Classic%'
```

Задача 2 / 6

```
SELECT billing_address, billing_country

FROM invoice

WHERE billing_country IN ('USA','India','Canada','Argentina','France');
```

Задача 3 / 6

```
SELECT billing_address,

       billing_country

FROM invoice

WHERE billing_country IN ('USA',

                          'India',

                          'Canada',

                          'Argentina',

                          'France')

          AND billing_city NOT IN ('Redmond','Lyon','Delhi');
```

Задача 4 / 6

```
SELECT *

FROM invoice

WHERE (CAST(invoice_date AS date) BETWEEN '2009-03-04' AND '2012-02-09')

AND total < 5

AND billing_country NOT IN ('Canada','Brazil','Finland');
```

Задача 5 / 6

```
SELECT title

FROM movie
```

```
WHERE description LIKE '%Mexico' AND

(rental_rate < 2 OR rating NOT LIKE 'PG-13');
```

Задача 6 / 6

```
SELECT name

FROM track

WHERE (milliseconds>300000 AND composer LIKE '%Bono%' AND genre_id IN (7,8,9,10)) OR (bytes>1000000000);
```

Операторы и функции для работы с датой и временем

Задача 1 / 5

```
SELECT customer_id,

       invoice_date,

       total

FROM invoice

WHERE customer_id BETWEEN 20 AND 50;
```

Задача 2 / 5

```
SELECT customer_id,

           invoice_date,

           total,

           DATE_TRUNC('month', CAST(invoice_date AS timestamp)),

           EXTRACT(WEEK FROM CAST(invoice_date AS timestamp))

FROM invoice

WHERE customer_id BETWEEN 20 AND 50;
```

Задача 3 / 5

```
SELECT customer_id,

       invoice_date,

       total,

       DATE_TRUNC('month', CAST(invoice_date AS timestamp)),

       EXTRACT(WEEK FROM CAST(invoice_date AS timestamp))

FROM invoice

WHERE (customer_id BETWEEN 20 AND 50) AND (EXTRACT(WEEK FROM CAST(invoice_date AS timestamp)) IN (5,7,10,33,48));
```

Задача 4 / 5

```
SELECT *

FROM invoice

WHERE EXTRACT(DAY FROM CAST(invoice_date AS timestamp)) = 1;
```

Задача 5 / 5

```
SELECT email

FROM staff

WHERE EXTRACT(YEAR FROM CAST(hire_date AS timestamp)) = 2002 AND city LIKE 'Calgary';
```

Специальное значение NULL

Практические задания отсутствуют

Условная конструкция с оператором CASE

Задача 1 / 2

```
SELECT last_name, first_name, title,

      CASE

           WHEN title LIKE '%IT%' THEN 'разработка'

           WHEN title LIKE '%Manager%' THEN 'отдел продаж'

           WHEN title LIKE '%Support%' THEN 'поддержка'

       END

FROM staff;
```

Задача 2 / 2

```
SELECT title, rental_rate,

       CASE

           WHEN rental_rate < 1 THEN 'категория 1'

           WHEN rental_rate >= 1 AND rental_rate < 3 THEN 'категория 2'

           WHEN rental_rate >= 3 THEN 'категория 3'

       END

FROM movie;
```

Работа с пропусками

Задача 1 / 4

```
SELECT billing_city

FROM invoice

WHERE billing_postal_code IS NULL;
```

Задача 2 / 4

```
SELECT billing_city

FROM invoice

WHERE billing_postal_code IS NULL

AND billing_state IS NOT NULL

AND total > 15;
```

Задача 3 / 4

```
SELECT album_id

FROM track

WHERE milliseconds>250000

AND name LIKE '%Moon%'

AND composer IS NULL;
```

Задача 4 / 4

```
SELECT first_name, last_name, country

FROM client

WHERE company  IS NULL AND

state IS NULL AND

phone IS NULL AND

fax IS NULL;
```

Агрегирующие функции. Группировка и сортировкаю

---

01 Математические операции

Практические задания отсутствуют

---

02 Агрегирующие функции

Практические задания отсутствуют

---

03 Применение агрегирующих функций

Задача 1 / 6

```
SELECT *

FROM invoice

WHERE EXTRACT(MONTH FROM CAST(invoice_date AS timestamp)) = 9;
```

Задача 2 / 6

```
SELECT customer_id,

CAST(invoice_date AS date),

total

FROM invoice

WHERE EXTRACT(MONTH FROM CAST(invoice_date AS timestamp)) = 9

AND customer_id IN ('11','13','44','36','48','52','54','56');
```

Задача 3 / 6

```
SELECT MIN(total), MAX(total)

FROM invoice

WHERE EXTRACT(MONTH FROM CAST(invoice_date AS timestamp)) = 9

AND customer_id in (11, 13, 44, 36, 48, 52, 54, 56);
```

Задача 4 / 6

```
SELECT MIN(total),

       MAX(total),

       ROUND(AVG(total)),

       COUNT(DISTINCT(customer_id)),

       SUM(total)

FROM invoice

WHERE EXTRACT(MONTH FROM CAST(invoice_date AS timestamp)) = 9

AND customer_id in (11, 13, 44, 36, 48, 52, 54, 56);
```

Задача 5 / 6

```
SELECT (COUNT(client)-COUNT(fax))

FROM client;
```

Задача 6 / 6

```
SELECT AVG(total)

FROM invoice

WHERE EXTRACT(DOW FROM CAST(invoice_date AS timestamp)) = 1;
```

---

04 Группировка данных

Практические задания отсутствуют

---

05 Применение оператора GROUP BY

Задача 1 / 6

```
SELECT SUM(total)

FROM invoice

WHERE billing_country = 'USA';
```

Задача 2 / 6

```
SELECT billing_city, SUM(total), COUNT(total), AVG(total)

FROM invoice

WHERE billing_country = 'USA'

GROUP BY billing_city;
```

Задача 3 / 6

```
SELECT SUM(total),

COUNT(DISTINCT(customer_id)),

SUM(total)/COUNT(DISTINCT(customer_id))

FROM invoice

WHERE billing_country = 'USA';
```

Задача 4 / 6

```
SELECT DATE_TRUNC('week', CAST(invoice_date AS timestamp)),

SUM(total),

COUNT(DISTINCT customer_id),

SUM(total)/COUNT(DISTINCT customer_id)

FROM invoice

WHERE billing_country = 'USA'

GROUP BY DATE_TRUNC('week', CAST(invoice_date AS timestamp));
```

Задача 5 / 6

```
SELECT support_rep_id,

COUNT(customer_id)

FROM client

WHERE email LIKE '%yahoo%'

OR email LIKE '%gmail%'

GROUP BY support_rep_id;
```

Задача 6 / 6

```
SELECT

CASE

WHEN total <1 THEN 'low cost'

WHEN total >=1 THEN 'high cost'

END,

SUM(total)

FROM invoice

WHERE billing_postal_code IS NOT NULL

GROUP BY

CASE

WHEN total <1 THEN 'low cost'

WHEN total >=1 THEN 'high cost'

END;
```

---

06 Сортировка данных

Практические задания отсутствуют

07

Задача 1 / 3

```
SELECT *

FROM invoice

ORDER BY total DESC
```

```
LIMIT 5;
```

Задача 2 / 3

```
SELECT customer_id,

COUNT(customer_id)

FROM invoice

WHERE billing_country = 'USA' AND

CAST(invoice_date AS date)

BETWEEN '2011-05-25'

AND '2011-09-25'

GROUP BY customer_id

ORDER BY  COUNT(customer_id) DESC, customer_id ASC

LIMIT 5;
```

Задача 3 / 3

```
SELECT EXTRACT(YEAR FROM CAST(invoice_date AS timestamp)),

MAX (total),

MIN(total),

SUM(total),

COUNT(total),

ROUND(SUM(total)/COUNT(DISTINCT customer_id))

FROM invoice

WHERE billing_country IN ('USA','United Kingdom','Germany')

GROUP BY EXTRACT(YEAR FROM CAST(invoice_date AS timestamp))

ORDER BY EXTRACT(YEAR FROM CAST(invoice_date AS timestamp)) DESC;
```

---

08 Группировка и сортировка по нескольким полям

Практические задания отсутствуют

---

09 Оператор HAVING

Практические задания отсутствуют

---

10 Применение оператора HAVING

Задача 1 / 3

```
SELECT rating, AVG(rental_rate)

FROM movie

GROUP BY rating

HAVING AVG(rental_rate)>3;
```

Задача 2 / 3

```
SELECT CAST(invoice_date AS date), SUM(total)

FROM invoice

WHERE CAST(invoice_date AS date)

BETWEEN '2011-09-01'

AND '2011-09-30'

GROUP BY CAST(invoice_date AS date)

HAVING SUM(total) BETWEEN 1 AND 10;
```

Задача 3 / 3

```
SELECT billing_country,

(COUNT(invoice) - COUNT(billing_postal_code))

FROM invoice

WHERE billing_address NOT LIKE '%Street%'

AND billing_address NOT LIKE '%Way%'

AND billing_address NOT LIKE '%Road%'

AND billing_address NOT LIKE '%Drive%'

GROUP BY billing_country

HAVING (COUNT(invoice) - COUNT(billing_postal_code))>10;
```

---

11 Заключение

Практические задания отсутствуют

Взаимоотношения между таблицами. Типы объединения таблиц

01 Как связаны таблицы

Практические задания отсутствуют

---

02 Графическое отображение связей. ER-диаграммы

Практические задания отсутствуют

---

03 Переименование полей и таблиц. Псевдонимы

Задача 1 / 3

```
SELECT billing_country,

       COUNT(i.total) AS total_purchases,

       SUM(i.total) AS total_revenue,

       ROUND(SUM(i.total)/COUNT(i.total),2) AS average_revenue

FROM invoice AS i

GROUP BY billing_country

ORDER BY average_revenue DESC

LIMIT 10;
```

Задача 2 / 3

```
SELECT rating AS rating_of_epic,

       release_year AS year_of_epic,

       AVG(rental_duration) AS average_rental

FROM movie

WHERE description LIKE '%Epic%'

GROUP BY rating, release_year;
```

Задача 3 / 3

```
SELECT

CASE

WHEN rating LIKE 'G' THEN 'без ограничений'

WHEN rating NOT LIKE  'G' THEN 'с ограничениями'

END AS new_rating,

SUM(rental_rate)

FROM movie

GROUP BY new_rating;
```

---

04 Операторы JOIN. Типы объединения таблиц

Практические задания отсутствуют

---

05 Оператор INNER JOIN

Задача 1 / 6

```
SELECT t.name, SUM(i.quantity)

FROM track AS t

INNER JOIN invoice_line AS i ON t.track_id = i.track_id

GROUP BY t.name

LIMIT 20;
```

Задача 2 / 6

```
SELECT t.name, p.playlist_id, SUM(i.quantity)

FROM track AS t

INNER JOIN invoice_line AS i ON t.track_id=i.track_id

INNER JOIN playlist_track AS p ON t.track_id=p.track_id

GROUP BY t.name, p.playlist_id

LIMIT 20;
```

Задача 3 / 6

```
SELECT t.name,

       SUM(i.quantity),
       pt.playlist_id,

       pl.name

FROM track AS t

INNER JOIN invoice_line AS i ON t.track_id=i.track_id

INNER JOIN playlist_track AS pt ON t.track_id=pt.track_id

INNER JOIN playlist AS pl ON pt.playlist_id=pl.playlist_id

GROUP BY t.name, pt.playlist_id,pl.name

LIMIT 20;
```

Задача 4 / 6

```
SELECT

p.name AS playlist_name,

SUM(i.unit_price) AS total_revenue

FROM track AS t

INNER JOIN invoice_line AS i ON t.track_id=i.track_id

INNER JOIN playlist_track AS pl ON t.track_id=pl.track_id

INNER JOIN playlist AS p ON pl.playlist_id = p.playlist_id

GROUP BY p.name

ORDER BY total_revenue DESC;
```

Задача 5 / 6

```
SELECT

g.name,

COUNT(i.invoice_id)

FROM track AS t

INNER JOIN invoice_line AS il ON t.track_id=il.track_id

INNER JOIN invoice AS i ON il.invoice_id=i.invoice_id

INNER JOIN genre AS g ON t.genre_id=g.genre_id

GROUP BY g.name

ORDER BY COUNT(i.invoice_id) DESC;
```

Задача 6 / 6

```
SELECT category.name

FROM movie

INNER JOIN film_actor ON movie.film_id=film_actor.film_id

INNER JOIN actor ON film_actor.actor_id=actor.actor_id

INNER JOIN film_category ON film_actor.film_id=film_category.film_id

INNER JOIN category ON film_category.category_id=category.category_id

WHERE actor.first_name LIKE'%Emily%'

GROUP BY category.name;
```

06 Операторы LEFT OUTER JOIN и RIGHT OUTER JOIN

Задача 1 / 6

```
SELECT track.name, CAST(invoice.invoice_date AS date)

FROM track

LEFT JOIN invoice_line ON track.track_id = invoice_line.track_id

LEFT JOIN invoice ON invoice_line.invoice_id = invoice.invoice_id;
```

Задача 2 / 6

```
SELECT EXTRACT(YEAR FROM CAST(invoice_date AS date)) AS year, COUNT(DISTINCT t.name)

FROM track AS t

LEFT JOIN invoice_line AS il ON t.track_id = il.track_id

LEFT JOIN invoice AS i ON il.invoice_id = i.invoice_id

GROUP BY year

ORDER BY year;
```

Задача 3 / 6

```
SELECT staff.last_name AS employee_last_name, COUNT(client.customer_id) AS all_customers

FROM staff

LEFT JOIN client ON staff.employee_id = client.support_rep_id

GROUP BY staff.employee_id

ORDER BY all_customers DESC;
```

Задача 4 / 6

```
SELECT

staff.last_name AS employee_last_name,

s.last_name AS manager_last_name

FROM staff

LEFT JOIN staff AS s ON staff.reports_to=s.employee_id;
```

Задача 5 / 6

```
SELECT movie.title

FROM movie

LEFT OUTER JOIN film_actor ON movie.film_id = film_actor.film_id

GROUP BY movie.title

HAVING COUNT(film_actor.actor_id) = 0;
```

Задача 6 / 6

```
SELECT artist.name

FROM artist

LEFT OUTER JOIN album ON artist.artist_id = album.artist_id
```

```
GROUP BY artist.name

HAVING COUNT(album.artist_id) = 0;
```

---

07 Оператор FULL OUTER JOIN

Практические задания отсутствуют

---

08 Альтернативные варианты присоединения: UNION и UNION ALL

Практические задания отсутствуют

---

09 Заключение

Практические задания отсутствуют

Подзапросы и временные таблицы

01 Подзапросы в FROM

Задача 1 / 5

```
SELECT title, rental_rate, length, rating

FROM movie

WHERE rental_rate > 2

ORDER BY length DESC

LIMIT 40;
```

Задача 2 / 5

```
SELECT rating,

MIN(length) AS min_length,

MAX(length) AS max_length,

AVG(length) AS avg_length,

MIN(rental_rate) AS min_rental_rate,

MAX(rental_rate) AS max_rental_rate,

AVG(rental_rate) AS avg_rental_rate

FROM

(

SELECT *

FROM movie AS mov

WHERE rental_rate > 2

ORDER BY mov.length DESC

LIMIT 40)

AS lim

GROUP BY lim.rating

ORDER BY avg_length;
```

Задача 3 / 5

```
SELECT

AVG(min_length) AS avg_min_length,

AVG(max_length) AS avg_max_length



FROM (SELECT top.rating,

       MIN(top.length) AS min_length,

       MAX(top.length) AS max_length,

       AVG(top.length) AS avg_length,

       MIN(top.rental_rate) AS min_rental_rate,

       MAX(top.rental_rate) AS max_rental_rate,

       AVG(top.rental_rate) AS avg_rental_rate

FROM

  (SELECT title,

          rental_rate,

          length,

          rating

   FROM movie

   WHERE rental_rate > 2

   ORDER BY length DESC

   LIMIT 40) AS top

GROUP BY top.rating

ORDER BY avg_length) AS xyz

;
```

Задача 4 / 5

```
SELECT AVG(count)

FROM

(SELECT album.title,COUNT(track.name) AS count

FROM album

INNER JOIN track ON album.album_id = track.album_id

WHERE album.title LIKE '%Rock%'

GROUP BY album.title

HAVING COUNT(track.name)>=8) AS xyz;
```

Задача 5 / 5

```
SELECT xyz.country

FROM

(SELECT

billing_country AS country,

EXTRACT(YEAR FROM CAST(invoice_date AS date)) AS year, EXTRACT(MONTH FROM CAST  (invoice_date AS date)) AS month,
AVG(total) AS avg_total

FROM invoice

WHERE EXTRACT(YEAR FROM CAST(invoice_date AS date)) = 2009

GROUP BY billing_country, EXTRACT(YEAR FROM CAST(invoice_date AS date)), EXTRACT    (MONTH FROM CAST(invoice_date AS date))) AS xyz

WHERE xyz.month IN (2,5,7,10)

GROUP BY xyz.country

HAVING SUM(xyz.avg_total)>10;
```

---

02 Подзапросы в WHERE

Задача 1 / 6

```
SELECT invoice_id

FROM invoice_line

GROUP BY invoice_id

HAVING COUNT(invoice_id)>5;
```

Задача 2 / 6

```
SELECT AVG(unit_price)

FROM invoice_line;
```

Задача 3 / 6

```
SELECT billing_country,

MIN(total) AS min_total,

MAX(total) AS max_total,

AVG(total) AS avg_total

FROM invoice

WHERE invoice_id IN (

    SELECT invoice_id

    FROM invoice_line

    GROUP BY invoice_id

    HAVING COUNT(invoice_id)>5

    )

AND total > (

    SELECT AVG(unit_price)

    FROM invoice_line)

GROUP BY billing_country

ORDER BY avg_total DESC;
```

Задача 4 / 6

```
SELECT name

FROM genre
```

```
WHERE genre_id IN

(SELECT DISTINCT(genre_id)

FROM

(SELECT genre_id

FROM track

ORDER BY milliseconds

LIMIT 10) AS xyz);
```

Задача 5 / 6

```
SELECT billing_city

FROM invoice

WHERE total > (SELECT AVG(total)

FROM invoice

WHERE EXTRACT(YEAR FROM CAST(invoice_date AS date)) = 2009)

GROUP BY billing_city
```

Задача 6 / 6

```
SELECT category.name, AVG(movie.length)

FROM movie

INNER JOIN film_category ON movie.film_id = film_category.film_id

INNER JOIN category ON  film_category.category_id = category.category_id

WHERE rating IN (SELECT rating

FROM movie

GROUP BY rating

ORDER BY AVG(rental_rate) DESC

LIMIT 1)

GROUP BY category.name;
```

---

03 Как сочетать объединения и подзапросы

Задача 1 / 4

```
SELECT iM.invoice_month, i2011.year_2011, i2012.year_2012, i2013.year_2013

FROM

(SELECT EXTRACT(MONTH FROM CAST(invoice_date AS date)) AS invoice_month

FROM invoice

GROUP BY invoice_month

ORDER BY invoice_month) AS iM

LEFT JOIN

(SELECT

EXTRACT(MONTH FROM CAST(invoice_date AS date)) AS invoice_month, COUNT(invoice_id) AS   year_2011

FROM invoice
WHERE EXTRACT(YEAR FROM CAST(invoice_date AS date)) = 2011

GROUP BY invoice_month

) AS i2011

ON iM.invoice_month = i2011.invoice_month

LEFT JOIN

(SELECT

EXTRACT(MONTH FROM CAST(invoice_date AS date)) AS invoice_month, COUNT(invoice_id) AS   year_2012

FROM invoice

WHERE EXTRACT(YEAR FROM CAST(invoice_date AS date)) = 2012

GROUP BY invoice_month

) AS i2012

ON iM.invoice_month = i2012.invoice_month

LEFT JOIN

(SELECT

EXTRACT(MONTH FROM CAST(invoice_date AS date)) AS invoice_month, COUNT(invoice_id) AS   year_2013

FROM invoice

WHERE EXTRACT(YEAR FROM CAST(invoice_date AS date)) = 2013

GROUP BY invoice_month

) AS i2013

ON iM.invoice_month = i2013.invoice_month
```

Задача 2 / 4

```
SELECT  last_name

FROM client

WHERE customer_id IN

(SELECT notjanyary.customer_id FROM

(SELECT DISTINCT(customer_id)

FROM invoice

WHERE (CAST(invoice_date AS date) BETWEEN '2013-01-01' AND '2013-01-31'))

AS janyary

INNER JOIN

(SELECT DISTINCT(customer_id)

FROM invoice

WHERE (CAST(invoice_date AS date) BETWEEN '2013-02-01' AND '2013-12-31'))

AS notjanyary

ON janyary.customer_id = notjanyary.customer_id);
```

Задача 3 / 4

```
SELECT

category.name AS name_category,

COUNT(movie.film_id) AS total_films
```

```
FROM movie


INNER JOIN film_category

ON movie.film_id = film_category.film_id


INNER JOIN category

ON category.category_id = film_category.category_id


WHERE movie.film_id IN (SELECT movie.film_id

FROM movie


INNER JOIN film_actor

ON movie.film_id = film_actor.film_id


WHERE film_actor.actor_id IN


(SELECT actor_id

FROM film_actor

INNER JOIN movie

ON movie.film_id = film_actor.film_id

WHERE movie.release_year > 2013

GROUP BY actor_id

HAVING COUNT(actor_id)>7)


GROUP BY movie.film_id)


GROUP BY category.name

ORDER BY COUNT(movie.film_id) DESC, category.name
```

Задача 4 / 4

> Год с наибольшей выручкой в летний период:2011

```
SELECT EXTRACT(YEAR FROM CAST(invoice_date AS date)) AS year, SUM(total)

FROM invoice

WHERE EXTRACT(MONTH FROM CAST(invoice_date AS date)) IN (6,7,8)

GROUP BY EXTRACT(YEAR FROM CAST(invoice_date AS date))

ORDER BY SUM(total) DESC
```

```
SELECT one.country, one.total_invoice, two.total_customer

FROM

(SELECT invoice.billing_country AS country,

      COUNT(invoice.total) AS total_invoice

      FROM invoice

WHERE EXTRACT(YEAR FROM CAST(invoice_date AS date)) = 2011

GROUP BY country) AS one

LEFT JOIN

(SELECT country, COUNT(customer_id) AS total_customer

FROM client

GROUP BY country) AS two

ON one.country = two.country

ORDER BY total_invoice DESC, country
```

06 Временные таблицы

Задача 1 / 3

```
WITH

top40 AS (

SELECT film_id

FROM movie

WHERE rental_rate >2

ORDER BY length DESC

LIMIT 40

)


SELECT rating,

MIN(length) AS min_length,

MAX(length) AS max_length,

AVG(length) AS avg_length,


MIN(rental_rate) AS min_rental_rate,

MAX(rental_rate) AS max_rental_rate,

AVG(rental_rate) AS avg_rental_rate

FROM

movie INNER JOIN top40 ON

movie.film_id = top40.film_id


GROUP BY rating

ORDER BY AVG(length)
```

Задача 2 / 3

```
SELECT iM.invoice_month, i2011.year_2011, i2012.year_2012, i2013.year_2013

FROM

(SELECT EXTRACT(MONTH FROM CAST(invoice_date AS date)) AS invoice_month

FROM invoice

GROUP BY invoice_month

ORDER BY invoice_month) AS iM

LEFT JOIN

(SELECT

EXTRACT(MONTH FROM CAST(invoice_date AS date)) AS invoice_month, COUNT(invoice_id) AS   year_2011

FROM invoice

WHERE EXTRACT(YEAR FROM CAST(invoice_date AS date)) = 2011

GROUP BY invoice_month

) AS i2011

ON iM.invoice_month = i2011.invoice_month

LEFT JOIN

(SELECT

EXTRACT(MONTH FROM CAST(invoice_date AS date)) AS invoice_month, COUNT(invoice_id) AS   year_2012

FROM invoice

WHERE EXTRACT(YEAR FROM CAST(invoice_date AS date)) = 2012

GROUP BY invoice_month

) AS i2012

ON iM.invoice_month = i2012.invoice_month

LEFT JOIN

(SELECT

EXTRACT(MONTH FROM CAST(invoice_date AS date)) AS invoice_month, COUNT(invoice_id) AS   year_2013

FROM invoice

WHERE EXTRACT(YEAR FROM CAST(invoice_date AS date)) = 2013

GROUP BY invoice_month

) AS i2013

ON iM.invoice_month = i2013.invoice_month
```

Задача 3 / 3

```
WITH

y2012 AS

(SELECT EXTRACT(MONTH FROM CAST(invoice_date AS date)) AS month,

SUM(total)

FROM  invoice

WHERE EXTRACT(YEAR FROM CAST(invoice_date AS date)) = 2012

GROUP BY month

),
y2013 AS

(SELECT EXTRACT(MONTH FROM CAST(invoice_date AS date)) AS month,

SUM(total)

FROM  invoice

WHERE EXTRACT(YEAR FROM CAST(invoice_date AS date)) = 2013

GROUP BY month

)


SELECT

y2012.month AS month,

y2012.sum AS sum_total_2012,

y2013.sum AS sum_total_2013,

ROUND(100*(y2013.sum / y2012.sum),0) AS perc


FROM  y2012 INNER JOIN y2013 ON y2012.month = y2013.month

ORDER BY y2012.month;
```

---

0 Заключение

Практические задания отсутствуют

Проект

01

```
SELECT COUNT(status)

FROM company

WHERE status LIKE 'closed'
```

02

```
SELECT funding_total

FROM company

WHERE category_code LIKE 'news' AND country_code LIKE 'USA'

ORDER BY funding_total DESC
```

03

```
SELECT SUM(price_amount)

FROM acquisition

WHERE term_code LIKE 'cash'

AND EXTRACT(YEAR FROM CAST(acquired_at AS date)) BETWEEN 2011 AND 2013
```

04

```
SELECT first_name, last_name, twitter_username

FROM people

WHERE twitter_username LIKE 'Silver%'
```

05

```
SELECT *

FROM people

WHERE twitter_username LIKE '%money%'

AND last_name LIKE 'K%'
```

06

```
SELECT country_code, SUM(funding_total)

FROM company

GROUP BY country_code

ORDER BY SUM(funding_total) DESC
```

07

```
SELECT funded_at, MIN(raised_amount), MAX(raised_amount)

FROM funding_round

GROUP BY funded_at

HAVING MIN(raised_amount) <> 0 AND MIN(raised_amount) <> MAX(raised_amount)
```

08

```
SELECT *,

      CASE

           WHEN invested_companies < 20 THEN 'low_activity'

           WHEN invested_companies < 100 THEN 'middle_activity'

           WHEN invested_companies >= 100 THEN 'high_activity'

       END

FROM fund
```

09

```
SELECT

CASE

   WHEN invested_companies>=100 THEN 'high_activity'

   WHEN invested_companies>=20 THEN 'middle_activity'

   ELSE 'low_activity'

END AS activity,
ROUND(AVG(investment_rounds))

FROM fund

GROUP BY activity

ORDER BY ROUND(AVG(investment_rounds));
```

10

```
SELECT country_code,

MIN(invested_companies),

MAX(invested_companies),

AVG(invested_companies)

FROM fund

WHERE EXTRACT(YEAR FROM CAST(funded_at AS date)) BETWEEN 2010 AND 2012

GROUP BY country_code

HAVING MIN(invested_companies) > 0

ORDER BY AVG(invested_companies) DESC, country_code

LIMIT 10
```

11

```
SELECT first_name, last_name, instituition

FROM people

LEFT JOIN education ON people.id = education.person_id
```

12

```
SELECT company.name,

COUNT(DISTINCT(instituition))

FROM company

LEFT JOIN people ON company.id = people.company_id

LEFT JOIN education ON people.id = education.person_id

GROUP BY company.id

ORDER BY COUNT(DISTINCT(instituition)) DESC

LIMIT 5
```

13

```
SELECT DISTINCT(company.name)

FROM funding_round

LEFT JOIN company ON company.id = funding_round.company_id

WHERE is_first_round = 1 AND is_last_round = 1

AND company.status LIKE 'closed'
```

14

```
WITH

org AS

(

SELECT DISTINCT(company.id)

FROM funding_round

LEFT JOIN company ON company.id = funding_round.company_id

WHERE is_first_round = 1 AND is_last_round = 1

AND company.status LIKE 'closed'

)


SELECT DISTINCT(people.id)

FROM people

WHERE people.company_id IN (SELECT * FROM org)
```

15

```
WITH

org AS

(

SELECT DISTINCT(company.id)

FROM funding_round

LEFT JOIN company ON company.id = funding_round.company_id

WHERE is_first_round = 1 AND is_last_round = 1

AND company.status LIKE 'closed'

),



p_id AS

(

SELECT DISTINCT(people.id)

FROM people

WHERE people.company_id IN (SELECT * FROM org)

)



SELECT person_id, instituition

FROM education

WHERE person_id IN (SELECT * FROM p_id)



UNION



SELECT person_id, instituition
```

```
FROM education

WHERE person_id IN (SELECT * FROM p_id)
```

16

```
WITH

org AS

(

SELECT DISTINCT(company.id)

FROM funding_round

LEFT JOIN company ON company.id = funding_round.company_id

WHERE is_first_round = 1 AND is_last_round = 1

AND company.status LIKE 'closed'

),



p_id AS

(

SELECT DISTINCT(people.id)

FROM people

WHERE people.company_id IN (SELECT * FROM org)

)



SELECT person_id, COUNT(instituition)

FROM education

WHERE person_id IN (SELECT * FROM p_id)

GROUP BY person_id
```

17

```
WITH

org AS

(

SELECT DISTINCT(company.id)

FROM funding_round

LEFT JOIN company ON company.id = funding_round.company_id

WHERE is_first_round = 1 AND is_last_round = 1

AND company.status LIKE 'closed'

),



p_id AS

(
SELECT DISTINCT(people.id)

FROM people

WHERE people.company_id IN (SELECT * FROM org)

)



SELECT AVG(count)

FROM

(

SELECT person_id, COUNT(instituition)

FROM education

WHERE person_id IN (SELECT * FROM p_id)

GROUP BY person_id

) AS tab
```

18

```
WITH

org AS

(

SELECT DISTINCT(company.id)

FROM funding_round

LEFT JOIN company ON company.id = funding_round.company_id

WHERE company.name LIKE 'Facebook'

),



p_id AS

(

SELECT DISTINCT(people.id)

FROM people

WHERE people.company_id IN (SELECT * FROM org)

)



SELECT AVG(count)

FROM

(

SELECT person_id, COUNT(instituition)

FROM education

WHERE person_id IN (SELECT * FROM p_id)

GROUP BY person_id

) AS tab
```

19

```
SELECT

fund.name AS name_of_fund,

company.name AS name_of_company,

funding_round.raised_amount AS amount

FROM funding_round

INNER JOIN investment ON funding_round.id = investment.funding_round_id

INNER JOIN fund ON investment.fund_id = fund.id

INNER JOIN company ON funding_round.company_id = company.id

WHERE EXTRACT(YEAR FROM CAST(funding_round.funded_at AS date)) BETWEEN 2012 AND 2013

AND company.milestones > 6
```

20

```
SELECT c_acquiring.name, acquisition.price_amount, c_acquired.name,     c_acquired.funding_total, ROUND(acquisition.price_amount/c_acquire

FROM acquisition

LEFT JOIN company AS c_acquiring ON c_acquiring.id = acquisition.acquiring_company_id

LEFT JOIN company AS c_acquired ON c_acquired.id = acquisition.acquired_company_id

WHERE acquisition.price_amount > 0 AND c_acquired.funding_total > 0

ORDER BY acquisition.price_amount DESC, c_acquired.name

LIMIT 10
```

21

```
SELECT company.name, EXTRACT(MONTH FROM CAST(funding_round.funded_at AS date))

FROM company

LEFT JOIN funding_round ON  company.id = funding_round.company_id

WHERE company.category_code LIKE 'social'

AND EXTRACT(YEAR FROM CAST(funding_round.funded_at AS date)) BETWEEN 2010 AND 2013

AND funding_round.raised_amount > 0
```

22

```
WITH

fund AS

(SELECT EXTRACT(MONTH FROM CAST(fr.funded_at AS date)) AS month,

COUNT(DISTINCT(f.name)) AS funds

FROM funding_round AS fr

INNER JOIN investment AS i ON fr.id = i.funding_round_id

INNER JOIN fund AS f ON i.fund_id = f.id

WHERE f.country_code LIKE 'USA'

AND EXTRACT(YEAR FROM CAST(fr.funded_at AS date)) BETWEEN 2010 AND 2013
GROUP BY month),

aqui AS

(SELECT EXTRACT(MONTH FROM CAST(a.acquired_at AS date)) AS month,

COUNT(a.acquired_company_id) AS count,

SUM(a.price_amount) AS sum

FROM acquisition AS a

WHERE EXTRACT(YEAR FROM CAST(a.acquired_at AS date)) BETWEEN 2010 AND 2013

GROUP BY month)


SELECT a.month, a.funds, b.count, b.sum

FROM aqui AS b

JOIN fund AS a ON b.month = a.month
```

23

```
WITH

y2011 AS (SELECT country_code AS country, AVG(funding_total) AS avg2011

FROM company

WHERE EXTRACT(YEAR FROM CAST(founded_at AS date)) = 2011

GROUP BY country_code),



y2012 AS (SELECT country_code AS country, AVG(funding_total) AS avg2012

FROM company

WHERE EXTRACT(YEAR FROM CAST(founded_at AS date)) = 2012

GROUP BY country_code),



y2013 AS (SELECT country_code AS country, AVG(funding_total) AS avg2013

FROM company

WHERE EXTRACT(YEAR FROM CAST(founded_at AS date)) = 2013

GROUP BY country_code)



SELECT y2011.country, y2011.avg2011, y2012.avg2012, y2013.avg2013

FROM y2011

INNER JOIN y2012 ON y2011.country = y2012.country

INNER JOIN y2013 ON y2011.country = y2013.country

ORDER BY y2011.avg2011 DESC
```

WITH

y2012 AS

(SELECT EXTRACT(MONTH FROM CAST(invoice_date AS date)) AS month,

SUM(total)

FROM invoice

```
WHERE EXTRACT(YEAR FROM CAST(invoice_date AS date)) = 2012

GROUP BY month

),

y2013 AS

(SELECT EXTRACT(MONTH FROM CAST(invoice_date AS date)) AS month,

SUM(total)

FROM invoice

WHERE EXTRACT(YEAR FROM CAST(invoice_date AS date)) = 2013

GROUP BY month

)

SELECT

y2012.month AS month,

y2012.sum AS sumtotal2012,

y2013.sum AS sumtotal2013,

ROUND(100*(y2013.sum / y2012.sum),0) AS perc

FROM y2012 INNER JOIN y2013 ON y2012.month = y2013.month

ORDER BY y2012.month;
```

Назад к списку тем

Sincerelly Yours
A1ex Ma