**Лабораторная работа №1.** **Простейшие запросы, условия выбора**

1.1. Вывести список имен, офисов и дат приема на работу всех служащих.

SELECT name,rep\_office,hire\_data

FROM salesreps

1.2. Вывести список офисов с их плановыми и фактическими объемами продаж.

SELECT office, city, target, sales

FROM offices

1.3. Вывести список офисов, расположенных в восточном регионе с их плановыми и фактическими объемами продаж.

SELECT office, city, target, sales

FROM offices

WHERE region = 'EASTERN'

1.4. Вывести список офисов, расположенных в восточном регионе, в которых фактический объем продаж превысил плановый, отсортировать список в алфавитном порядке.

SELECT office, city, target, sales

FROM offices

WHERE region = 'EASTERN' AND sales>target

ORDER BY city

1.5. Как зовут служащего с идентификатором 107?

SELECT name

FROM salesreps

WHERE empl\_num = 107

1.6. Каков средний возраст служащих компании?

SELECT AVG(age) AS avgage

FROM salesreps

1.7. Выдать для каждого офиса список городов, регионов и сумм, на которые был перевыполнен/недовыполнен план по продажам.

SELECT city, region, sales-target AS "sales-target"

FROM offices

1.8. Показать общую стоимость по каждому товару на складе.

SELECT description, price\*qyy\_on\_had AS cost

FROM products

1.9. Что получится, если увеличить плановый объем продаж для каждого служащего на 3% от его фактического объема продаж.

SELECT quota+(sales\*0.03) AS chto

FROM salesreps

1.10. Вывести список объемов продаж для каждого города.

SELECT city,'has sales of',sales

FROM offices

1.11. Вывести список идентификаторов всех менеджеров офисов.

SELECT DISTINCT mgr

FROM offices

WHERE mgr IS NOT NULL

2.1. Найти всех служащих принятых до 1988 года.

SELECT name,age,title,hire\_data

FROM salesreps

WHERE hire\_data < '01.01.1988'

2.2. Вывести список офисов, фактические объемы продаж в которых составили менее 80% от плановых.

SELECT \*

FROM offices

WHERE sales<0.8\*target

2.3. Вывести список офисов, менеджером которых НЕ является служащий с идентификатором 108.

SELECT \*

FROM offices

WHERE mgr <> 108

2.4. Найти все заказы сделанные в последнем квартале 1989 года.

SELECT \*

FROM orders

WHERE order\_data BETWEEN '30.09.1989'AND '01.01.1990'

2.5. Найти заказы, стоимость которых попадает в диапазон от 20 000 до 30 000.

SELECT \*

FROM orders

WHERE amount::numeric BETWEEN 20000 AND 30000

2.6. Вывести список служащих, фактический объем продаж которых не попадает в диапазон от 80 до 120 процентов плана.

SELECT name

FROM salesreps

WHERE sales NOT BETWEEN quota\*0.8 AND quota\*1.2

2.7. Вывести список служащих, которые работают в Нью-Йорке, Атланте и Денвере (11,13,22).

SELECT name, city

FROM salesreps INNER JOIN offices

ON rep\_office = office

WHERE city = 'NEW YORK' OR city = 'ATLANTA' OR city = 'DENVER'

2.8. Найти служащего, который не закреплен за офисом.

SELECT name

FROM salesreps

WHERE rep\_office IS NULL

2.9. Вывести список служащих, которые уже закреплены за офисами.

SELECT name

FROM salesreps

WHERE rep\_office IS NOT NULL

2.10. Как вы думаете, почему нельзя просто проверить значение на равенство NULL?

Результатом сравнения чего-либо с NULL является NULL.

2.11. Найти служащих, у которых фактический объем продаж меньше планового, но больше 150 000.

SELECT \*

FROM salesreps

WHERE sales<quota AND sales::numeric >150000

12. Найти всех служащих, которые: работают в Денвере, Нью-Йорке или Чикаго, или не имеют менеджера или были приняты на работу после июня 1988 года, или превысили плановый объем продаж, но не достигли уровня в 600 000.

SELECT \*

FROM salesreps INNER JOIN offices

ON rep\_office = office

WHERE city = 'DENVER' OR city = 'NEW YORK' OR city = 'CHICAGO' OR manager IS NULL OR hire\_data>'30.06.1988' OR salesreps.sales>quota AND salesreps.sales::numeric <600000

13. Вывести список офисов, отсортированных по фактическим объемам продаж в порядке убывания.

SELECT \*

FROM offices

ORDER BY sales DESC

14. Вывести список всех офисов, отсортированных по разности между фактическими и плановыми объемами продаж в порядке убывания.

SELECT office, city, region, sales-target AS diff\_sales\_target

FROM offices

ORDER BY diff\_sales\_target DESC

15. Вывести список всех офисов, отсортированных в алфавитном порядке по названию региона, а в каждом регионе - по разности между фактическими и плановыми объемами продаж в порядке убывания.

SELECT office, city, region, sales-target

FROM offices

ORDER BY 3, 4 DESC

**Лабораторная работа №2.** **Итоговые запросы на выборку**

1. Каков средний процент выполнения плана в нашей компании?

SELECT AVG(sales::numeric/quota::numeric \* 100)

FROM salesreps

2. Какова сумма всех заказов, принятых служащим с идентификатором 104?

SELECT SUM(amount)

FROM orders

WHERE rep=104

3. Каков наибольший процент выполнения плана среди всех служащих?

SELECT MAX(sales::numeric/quota::numeric \* 100)

FROM salesreps

4. Сколько служащих перевыполнили план?

SELECT COUNT(\*)

FROM salesreps

WHERE sales>quota

5. Сколько имеется заказов стоимостью более 25 000?

SELECT COUNT(\*)

FROM orders

WHERE amount::numeric>25000

6. Сколько должностей существует в нашей компании?

SELECT COUNT(DISTINCT title)

FROM salesreps

7. Какова средняя стоимость заказа для каждого служащего?

SELECT rep, AVG(amount::numeric)

FROM orders

GROUP BY rep

8. Каков диапазон плановых объемов продаж для каждого офиса?

SELECT MIN(quota::numeric), MAX(quota::numeric)

FROM salesreps INNER JOIN offices

ON rep\_office = office

GROUP BY office

9. Сколько клиентов обслуживает каждый сотрудник?

SELECT cust\_rep, COUNT(cust\_num)

FROM customers

GROUP BY cust\_rep

10. Подсчитать общее количество заказов по каждому клиенту для каждого служащего.

SELECT rep, cust, COUNT(order\_num)

FROM orders

GROUP BY cust, rep

**Лабораторная работа №3.** **Подчиненные запросы на выборку**

1. Вывести список служащих, плановый объём продаж которых составляет менее 10% от планового объёма продаж всей компании.

SELECT name

FROM salesreps

WHERE quota < 0.1\* (SELECT SUM(target)

FROM offices)

2. Вывести список служащих, у которых плановый объём продаж равен или больше планового объёма продаж офиса, расположенного в Атланте.

SELECT name

FROM salesreps

WHERE quota >= (SELECT target

FROM offices

WHERE city='ATLANTA')

3. Вывести имена и данные о возрасте служащих, для которых плановый объём продаж выше среднего.

SELECT name, age

FROM salesreps

WHERE quota > (SELECT AVG(quota)

FROM salesreps)

4. Вывести список служащих тех офисов, где фактический объём продаж превышает плановый.

SELECT name

FROM salesreps

WHERE rep\_office IN (SELECT office

FROM offices

WHERE sales>target)

5. Вывести имена и данные о возрасте служащих, работающих в офисах западного региона.

SELECT name, age

FROM salesreps

WHERE rep\_office IN (SELECT office

FROM offices

WHERE region = 'WESTERN')

6. Вывести список служащих, не работающих в офисах, которыми руководит Ларри Фитч.

SELECT name

FROM salesreps

WHERE rep\_office NOT IN (SELECT office

FROM offices

WHERE mgr = (SELECT empl\_num

FROM salesreps

WHERE name = 'Larry Fitch'))

7. Вывести список клиентов, закрепленных за служащими, работающими в офисах восточного региона.

SELECT company

FROM customers

WHERE cust\_rep IN (SELECT empl\_num

FROM salesreps

WHERE rep\_office IN (SELECT office

FROM offices

WHERE region = 'EASTERN')

9. Вывести список офисов, где имеется служащий, чей план превышает 55 процентов от плана офиса.

SELECT city

FROM offices

WHERE EXISTS (SELECT \*

FROM salesreps

WHERE rep\_office = office

AND quota > (.55 \* target))

11. Вывести имена и данные о возрасте всех служащих, которые руководят офисами.

SELECT name, age

FROM salesreps

WHERE empl\_num = ANY (SELECT mgr

FROM offices)

12. Вывести имена и данные о возрасте всех служащих, которые не руководят офисами.

SELECT name, age

FROM salesreps

WHERE empl\_num <> ANY (SELECT mgr

FROM offices)

**Лабораторная работа №4. Многотабличные запросы**

1. Вывести список всех служащих, включая города и регионы, в которых они работают. Где таблица потомок, а где предок?

SELECT empl\_num, city, region

FROM salesreps, offices

WHERE rep\_office = office

Offices – таблица-предок, salesreps – таблица-потомок.

2. Вывести список офисов, включая имена и должности их руководителей

SELECT office, city, name, title

FROM salesreps, offices

WHERE mgr=empl\_num

3. Вывести список офисов, в которых план продаж превышает 600 000.

SELECT office

FROM offices

WHERE target::numeric>600000

4. Вывести список всех заказов, в том числе их стоимости и описания товаров, заказов стоимостью выше 25 000.

SELECT order\_num, order\_data, amount, description

FROM orders, products

WHERE product=product\_id AND mfr=mer\_id AND amount::numeric>25000

5. Вывести список заказов стоимостью выше 25 000, включая имена служащего, принявшего заказ, и имя клиента, сделавшего его.

SELECT order\_num, company, name, amount

FROM (orders INNER JOIN salesreps ON rep=empl\_num)

INNER JOIN customers

ON cust\_num=cust

WHERE amount::numeric>25000

6. Вывести список заказов стоимостью выше 25 000, включая имя клиента, сделавшего заказ, имя закрепленного за ним служащего и офис, в котором работает этот служащий.

SELECT order\_num, company, name, rep\_office

FROM orders, customers, salesreps

WHERE cust=cust\_num AND cust\_rep=empl\_num AND amount>'25000'

7. Для каждого офиса, в котором работает два или более человек, вычислить общий плановый и фактический объёмы продаж для всех служащих офиса.

SELECT city, SUM(quota),SUM(salesreps.sales)

FROM offices, salesreps

WHERE office=rep\_office

GROUP BY rep\_office, city

HAVING COUNT(rep\_office)>2

8. Показать цену, количество на складе и общее количество заказанных единиц для каждого наименования товара, если для него общее количество заказанных единиц превышает 75% от количества товара на складе.

SELECT product\_id, description, price,qyy\_on\_had, qty

FROM products, orders

WHERE product=product\_id AND qty>(0.75\*qyy\_on\_had)

9. Показать имя, офис и объем продаж каждого служащего.

SELECT name, city, salesreps.sales

FROM salesreps, offices

WHERE rep\_office=office

10. Вывести список имён служащих и их руководителей.

SELECT s.name employee , m.name manager

FROM salesreps s, salesreps m

where s.manager=m.empl\_num

11. Вывести список служащих, которые работают со своими руководителями в различных офисах, включая имена и офисы как служащих, так и руководителей.

SELECT s.name AS employee, s.rep\_office AS office\_emp, m.name AS manager, m.rep\_office AS office\_man

FROM salesreps s, salesreps m

WHERE s.manager = m.empl\_num AND s.rep\_office <> m.rep\_office

12. Вывести список служащих и городов, где они работают.

SELECT name, city

FROM salesreps INNER JOIN offices ON rep\_office = office

SELECT name, city

FROM salesreps FULL OUTER JOIN offices ON rep\_office = office

SELECT name, city

FROM salesreps LEFT OUTER JOIN offices ON rep\_office = office

SELECT name, city

FROM salesreps RIGHT OUTER JOIN offices ON rep\_office = office

**Лабораторная работа №5. Изменение данных**

1. Добавьте в БД информацию о новом служащем. А также о том, что он получил свой первый заказ от компании IntelCorp, нового клиента, которому был присвоен идентификатор 2126, это заказ №113069 на 20 изделий ACI-4104 общей стоимостью 2340.

INSERT INTO salesreps (empl\_num, name, rep\_office, title, hire\_data, quota, sales) VALUES (54321, 'Sergey', 13, 'Delivery boy', '12.07.2000', 11000, 1000)

INSERT INTO customers (cust\_num, company, cust\_rep) VALUES (2126, 'InterCorp', 54321)

INSERT INTO orders (order\_num, cust, rep, mfr, product, qty, amount) VALUES (113069, 2126, 54321, 'ACI', '41004', 20, 2340.00)

3. Провести анализ, что именно приобретают клиенты.

CREATE TABLE demand

INSERT INTO demand (amount, company, name, perf, mfr, product, qty)

SELECT amount, company, name, (sales - quota), mfr, product, qty

FROM orders, customers, salesreps

WHERE cust = cust\_num AND rep = empl\_num AND amount::numeric > 15000

4. Удалить информацию о добавленном новом служащем.

DELETE FROM salesreps

WHERE empl\_num = 54321

5. Удалить данные о всех служащих, принятых на работу до июля 1988 года и ещё не имеющих личного плана.

DELETE FROM salesreps

WHERE hire\_data < '01.07.1988' AND quota IS NULL

6. Удалить все заказы, принятые Сью Смит.

DELETE FROM orders

WHERE rep=102

7. Удалить данные о всех клиентах, обслуживаемых служащими, у которых фактический объём продаж меньше 80% их плана.

DELETE FROM customers

WHERE cust\_rep IN (SELECT empl\_num

FROM salesreps

WHERE (.8\*quota)>sales)

8. Увеличить лимит кредита для компании Acme Mfg до 60000 и закрепить её за Мэри Джонс (идентификатор 109)

UPDATE customers SET credit\_limit = 70000.00, cust\_rep = 109

WHERE company = 'Acme Mfg'

9. Перевести всех служащих из Чикагского офиса (12) в нью-йоркский (11) и понизить их личные планы на 10%.

UPDATE salesreps SET rep\_office = 11, quota = (0.9 \* quota)

WHERE rep\_office = 12

10. Перевести всех клиентов, обслуживаемых служащими с идентификаторами 105, 106, 107 к служащему с идентификатором 102.

UPDATE customers SET cust\_rep = 102

WHERE cust\_rep IN (105, 106, 107)

11. Увеличить на 5000 лимит кредита для всех клиентов, которые сделали заказ на сумму более 25000.

UPDATE customers SET credit\_limit = credit\_limit + 5000.00

WHERE cust\_num IN (SELECT DISTINCT cust

FROM orders

WHERE amount > 25000.00)

12. Закрепить за Биллом Адамсом (105) клиентов, обслуживаемых служащими, чей объём продаж меньше 80% их личного плана.

UPDATE customers SET cust\_rep = 105

WHERE cust\_rep IN (SELECT empl\_num

FROM salesreps

WHERE sales < (.8 \* quota))

13. Всех служащих, обслуживающих более трех клиентов, подчинить непосредственно Сэму Кларку (106).

UPDATE salesreps

SET manager = 106

WHERE 3 < (SELECT COUNT(\*)

FROM customers

WHERE cust\_rep = empl\_num)

**Лабораторная работа №6. Представления**

1. Создать представление, показывающее информацию о служащих восточного региона.

CREATE VIEW eastern AS

SELECT \*

FROM salesreps

WHERE rep\_office IN (11, 12, 13)

2. Создать представление, показывающее информацию о служащих западного региона.

CREATE VIEW western AS

SELECT \*

FROM salesreps

WHERE rep\_office NOT IN (11, 12, 13)

3. Создать представление, показывающее заказы, сделанные клиентами конкретного менеджера с идентификатором 102.

CREATE VIEW orders102 AS

SELECT \*

FROM Orders

WHERE rep = 102

4. Создать представление, показывающее только тех клиентов, которые в настоящий момент сделали заказы на сумму более 30000.

CREATE VIEW orderedover30 AS

SELECT cust

FROM orders

WHERE amount::numeric>30000

5. Создать представление таблицы Offices, которое включает в себя идентификатор офиса, название города и название региона.

CREATE VIEW otchet AS

SELECT office, city, region

FROM offices

6. Создать представление, включающее ид.клиента, имя клиента и лимит кредита для всех клиентов служащего с ид.102.

CREATE VIEW view6 AS

SELECT cust\_num, company, credit\_limit

FROM customers

WHERE cust\_rep = 102

7. Создать представление, включающее суммарные данные о заказах по каждому служащему (количество, суммарную стоимость, мин, мах, среднюю стоимость).

CREATE VIEW view7 AS

SELECT name, COUNT(order\_num) AS thiscount, SUM(amount::numeric) AS thissum, MIN(amount::numeric) AS thismin, MAX(amount::numeric) AS thismax, AVG(amount::numeric) AS thisavg

FROM orders INNER JOIN salesreps ON rep=empl\_num

GROUP BY name

8.Обращаясь к полученному представлению, отобразить на экран имя, число заказов, общую стоимость заказов и среднюю стоимость заказов по каждому служащему.

SELECT name, thissum, thissum, thisavg

FROM view7

9. Создать представление, модифицирующее таблицу заказов так, чтобы вместо идентификаторов служащих стояли имена.

CREATE VIEW view9 AS

SELECT name, order\_data, cust, rep, mfr, product, qty, amount

FROM orders, salesreps

WHERE rep= empl\_num

10. Вывести список крупных заказов (>20000), упорядоченных по стоимости.

CREATE VIEW view10 AS

SELECT \*

FROM orders

WHERE amount::numeric > 20000

ORDER BY amount

**Лабораторная работа №7. Оператор CASE. Индексы**

1. Разделить клиентов на три категории в соответствии с суммой их кредитного лимита. Категория A credit\_limit >60 000, категория B credit\_limit > 30 000, остальные относятся к С категории.

SELECT company, case

WHEN credit\_limit::numeric > 60000 THEN 'A'

WHEN credit\_limit::numeric > 30000 THEN 'B'

ELSE 'C'

END

FROM customers

2. Вывести суммы плановых объёмов продаж служащих по офисам. Если служащему ещё не назначен офис, его сумма должна быть включена в итоги по офису его руководителя.

SELECT city, SUM (salesreps.sales)

FROM offices, salesreps

WHERE office=

CASE WHEN (rep\_office IS NOT NULL) THEN rep\_office

ELSE (SELECT rep\_office

FROM salesreps AS mgrs

WHERE mgrs.empl\_num=manager)

END

GROUP BY city

3. Вывести список всех офисов компании с именами их руководителей и названиями городов и штатов, в которых они расположены.

SELECT name, city, CASE WHEN office=11 THEN 'New York'

WHEN office=12 THEN 'Illinois'

WHEN office=13 THEN 'Georgia'

WHEN office=21 THEN 'California'

WHEN office=22 THEN 'Colorado'

END

FRON offices, salesreps

WHERE mgr=empl\_num

4. Разделить служащих на три категории в соответствии с суммой их фактических объёмов продаж.

SELECT name, case

WHEN sales ::numeric > 300000 THEN 'A'

WHEN sales ::numeric > 100000 THEN 'B'

ELSE 'C'

END

FROM salesreps

5. Сделать отчёт о служащих и их плановых объёмах продаж. Если служащему ещё не назначен план, то в этой колонке нужно ввести его фактический объём продаж на текущий период. Если же, по каким-то причинам, значение столбца SALES также окажется NULL, тогда во второй колонке отчёт вывести ноль.

SELECT name, COALESCE (quota::numeric, sales::numeric, 0.00)

FROM salesreps

6. Создайте заданные индексы company и name на основании схемы.

CEATE INDEX indexCompany ON customers (company ASC)

CEATE INDEX indexName ON salesreps (name DESC)

**Лабораторная работа №8. Хранимые процедуры. Триггеры.**

1. Изменить процедуру ADD\_CUST новым условием: если предполагаемая сумма заказов клиента на первый год составляет не более 20000, эта сумма будет добавлена к плану служащего, а если более 20000, к плану будут добавлены фиксированные 20000.

CREATE OR REPLACE FUNCTION aad\_cust\_v\_2

(c\_name character varying,

c\_num integer,

cred\_lim numeric,

tgt\_sls numeric,

c\_rep integer,

c\_offc character varying

RETURNS void AS

$BODY$

BEGIN

INSERT INTO customers (cust\_num, company, cust\_rep, credit\_limit)

VALUES (c\_num,c\_name,c\_rep,cred\_lim);

IF tgt\_sls < 20000

THEN

UPDATE salesreps

SET quota =quota::numeric + tgt\_sls

WHERE empl\_num = c\_rep;

ELSE

UPDATE salesreps

SET quota =quota::numeric + 20000

WHERE empl\_num = c\_rep;

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN RAISE EXCEPTION 'error';

COMMIT;

END

$BODY$

LANGUAGE plpgsql VOLATILE

2. Когда в таблицу ORDERS добавляется новый заказ, в БД необходимо внести следующие два изменения: 1) значение в столбце SALES для служащего, принявшего заказ, должно быть увеличено на стоимость заказа; 2) текущее количество заказанного товара в столбце QTY\_ON\_HAND должно быть уменьшено на заказанное количество единиц товара.

CREATE OR REPLACE FUNCTION triggger\_orders\_before\_insert()

RETURNS trigger AS

$BODY$

BEGIN

UPDATE salesreps

SET sales= sales::numeric + new.amount::numeric

FROM salesreps , orders

WHERE salesreps.empl\_num=new.rep;

UPDATE products

SET qyy\_on\_had= qyy\_on\_had – new.qty

FROM products, orders

WHERE mer\_id = mfr AND product\_id = product;

RETURN new;

END

$BODY$

LANGUAGE plpgsql VOLATILE;

CREATE TRIGGER triggger\_orders\_before\_insert

BEFORE INSERT ON orders FOR EACH ROW

EXECUTE PROCEDURE triggger\_orders\_before\_insert();