Федеральное государственное бюджетное образовательное учреждение высшего

образования

«Чувашский государственный университет им. И.Н. Ульянова»

Кафедра вычислительной техники

Лабораторная работа № 4

SQL-запросы

Вариант: Инвестор

Выполнил: Назаров Ярослав

студент группы ИВТ-42-23

Проверил: старший преподаватель   
Марков А.В.

Чебоксары, 2025

Описание запросов

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Имя запроса** | **Текст запроса** | **Запрос на SQL** |
| **1** | q\_Топ5ДорогихАктивов | Вывести 5 самых дорогих активов по цене за единицу. | SELECT TOP 5 Наименование, Тикер, ЦенаЗаЕдиницу FROM АКТИВ ORDER BY ЦенаЗаЕдиницу DESC; |
| **2** | q\_10ПоследнихТранзакций | Вывести 10% самых последних транзакций. | SELECT TOP 10 PERCENT \* FROM ТРАНЗАКЦИЯ ORDER BY ДатаВремя DESC; |
| **3** | q\_ТранзакцииЗаПериод | Вывести транзакции в указанном интервале дат (параметрический). | PARAMETERS [Начальная дата] DateTime, [Конечная дата] DateTime; SELECT \* FROM ТРАНЗАКЦИЯ WHERE ДатаВремя BETWEEN [Начальная дата] AND [Конечная дата]; |
| **4** | q\_СчетаВИСилиБрокерские | Вывести счета, которые являются "ИИС" или "Брокерский". | SELECT НомерСчета, ТипСчета, Статус FROM СЧЁТ WHERE ТипСчета IN ('ИИС', 'Брокерский'); |
| **5** | q\_АктивныеСчета | Вывести все активные счета инвесторов. | SELECT НомерСчета, ТипСчета, ДатаОткрытия FROM СЧЁТ WHERE Статус = 'Активен'; |
| **6** | q\_ИнвесторыБезСчетов | Вывести инвесторов, у которых нет открытых счетов. | SELECT ФИО FROM ИНВЕСТОР WHERE [#И] NOT IN (SELECT [#И(FK)] FROM СЧЁТ); |
| **7** | q\_СчетаПоФамилии | Вывести все счета инвестора по введенной фамилии (части ФИО). | PARAMETERS [Введите часть ФИО] Text(255); SELECT И.ФИО, С.НомерСчета, С.ТипСчета FROM ИНВЕСТОР AS И INNER JOIN СЧЁТ AS С ON И.[#И] = С.[#И(FK)] WHERE И.ФИО LIKE "\*" & [Введите часть ФИО] & "\*"; |
| **8** | q\_ИнвесторыРожденныеВ90х | Вывести инвесторов, родившихся в 1990-х годах. | SELECT ФИО, ДатаРождения FROM ИНВЕСТОР WHERE ДатаРождения BETWEEN #01/01/1990# AND #12/31/1999#; |
| **9** | q\_БрокерыБолее5счетов | Вывести брокеров, у которых открыто более 5 счетов. | SELECT Б.Наименование, COUNT(С.[#С]) AS [Количество счетов] FROM БРОКЕР AS Б INNER JOIN СЧЁТ AS С ON Б.[#БР] = С.[#БР(FK)] GROUP BY Б.Наименование HAVING COUNT(С.[#С]) > 5; |
| **10** | q\_КолвоСчетовПоТипам | Вывести количество счетов каждого типа. | SELECT ТипСчета, COUNT(НомерСчета) AS [Количество] FROM СЧЁТ GROUP BY ТипСчета; |
| **11** | q\_ПолнаяИнформацияТранзакции | Вывести полную информацию о транзакциях: кто, у какого брокера, по какому счету, какой актив и операция. | SELECT И.ФИО, Б.Наименование AS Брокер, С.НомерСчета, А.Тикер, Т.ТипОперации, Т.СуммаТранзакции FROM (((ИНВЕСТОР AS И INNER JOIN СЧЁТ AS С ON И.[#И] = С.[#И(FK)]) INNER JOIN БРОКЕР AS Б ON С.[#БР(FK)] = Б.[#БР]) INNER JOIN ТРАНЗАКЦИЯ AS Т ON С.[#С] = Т.[#С(FK)]) INNER JOIN АКТИВ AS А ON Т.[#А(FK)] = А.[#А]; |
| **12** | q\_ИнвесторыСАктивнымиСчетами | Вывести инвесторов, у которых есть хотя бы один активный счёт (с использованием EXISTS). | SELECT ФИО FROM ИНВЕСТОР AS И WHERE EXISTS (SELECT \* FROM СЧЁТ AS С WHERE С.[#И(FK)] = И.[#И] AND С.Статус = 'Активен'); |
| **13** | q\_ИнвесторыСПочтойGmail | Вывести инвесторов с электронной почтой на домене 'gmail.com'. | SELECT ФИО, ЭлектроннаяПочта FROM ИНВЕСТОР WHERE ЭлектроннаяПочта LIKE "\*@gmail.com"; |
| **14** | q\_ИнвесторыСТойЖеДатойРождения | Вывести пары инвесторов с одинаковой датой рождения (самообъединение). | SELECT А.ФИО, В.ФИО, А.ДатаРождения FROM ИНВЕСТОР AS А, ИНВЕСТОР AS В WHERE А.ДатаРождения = В.ДатаРождения AND А.[#И] < В.[#И]; |
| **15** | q\_ВсеБрокерыИСчета | Вывести список всех брокеров и количество их счетов, включая брокеров без счетов. | SELECT Б.Наименование, COUNT(С.[#БР(FK)]) AS [Кол-во счетов] FROM БРОКЕР AS Б LEFT JOIN СЧЁТ AS С ON Б.[#БР] = С.[#БР(FK)] GROUP BY Б.Наименование; |
| **16** | q\_ДобавитьИнвестора | Добавить новую запись в таблицу "ИНВЕСТОР". | INSERT INTO ИНВЕСТОР (ФИО, ДатаРождения, КонтактныйТелефон, ЭлектроннаяПочта) VALUES ('Иванов Иван Иванович', #10/15/1985#, '+7 (123) 456-78-90', 'ivanov.ii@email.com'); |
| **17** | q\_ОбновитьСтатусСчета | Обновить статус счёта на "Закрыт" для счёта с конкретным номером. | UPDATE СЧЁТ SET Статус = 'Закрыт' WHERE НомерСчета = '40817810000000001234'; |
| **18** | q\_СписокВсехКонтактов | Вывести единый список ФИО инвесторов и наименований брокеров с пометкой их типа. | SELECT ФИО AS Имя, 'Инвестор' AS Тип FROM ИНВЕСТОР UNION SELECT Наименование, 'Брокер' FROM БРОКЕР; |
| **19** | q\_СоздатьТаблицуАрхив | Создать таблицу "АрхивТранзакций\_2024" на основе транзакций за 2024 год. | SELECT \* INTO АрхивТранзакций\_2024 FROM ТРАНЗАКЦИЯ WHERE YEAR(ДатаВремя) = 2024; |
| **20** | q\_УдалитьТранзакции | Удалить все транзакции с суммой менее 100. | DELETE FROM ТРАНЗАКЦИЯ WHERE СуммаТранзакции < 100; |
| **21** | q\_СводкаПоТипамОпераций | Построить перекрестный запрос, показывающий общую сумму транзакций для каждого инвестора по типам операций. | TRANSFORM SUM(Т.СуммаТранзакции) AS Сумма SELECT И.ФИО FROM (ИНВЕСТОР AS И INNER JOIN СЧЁТ AS С ON И.[#И] = С.[#И(FK)]) INNER JOIN ТРАНЗАКЦИЯ AS Т ON С.[#С] = Т.[#С(FK)] GROUP BY И.ФИО PIVOT Т.ТипОперации; |

Виды запросов

по видам.

## Таблица запросов

| № | Имя запроса | Текст запроса | Запрос на SQL |
| --- | --- | --- | --- |
| 1 | q\_Топ5ДорогихАктивов | Вывести 5 самых дорогих активов по цене за единицу | SELECT name, ticker, unit\_price <br>FROM investor\_asset <br>ORDER BY unit\_price DESC <br>LIMIT 5; |
| 2 | q\_10ПоследнихТранзакций | Вывести 10 последних транзакций | sql<br>SELECT \* <br>FROM investor\_transaction <br>ORDER BY transaction\_datetime DESC <br>LIMIT 10;<br> |
| 3 | q\_ТранзакцииЗаПериод | Вывести транзакции в указанном интервале дат | sql<br>SELECT \* <br>FROM investor\_transaction <br>WHERE transaction\_datetime <br>BETWEEN $1 AND $2;<br> |
| 4 | q\_СчетаВИСилиБрокерские | Вывести счета, которые являются "ИИС" или "Брокерский" | sql<br>SELECT name, account\_type, status <br>FROM investor\_account <br>WHERE account\_type IN ('iis', 'broker');<br> |
| 5 | q\_АктивныеСчета | Вывести все активные счета инвесторов | sql<br>SELECT a.name, a.account\_type, <br> a.open\_date, i.name as investor\_name<br>FROM investor\_account a<br>JOIN investor\_investor i ON a.investor\_id = i.id<br>WHERE a.status = 'active';<br> |
| 6 | q\_ИнвесторыБезСчетов | Вывести инвесторов, у которых нет открытых счетов | sql<br>SELECT name <br>FROM investor\_investor i<br>WHERE NOT EXISTS (<br> SELECT 1 <br> FROM investor\_account a <br> WHERE a.investor\_id = i.id<br>);<br> |
| 7 | q\_СчетаПоФамилии | Вывести все счета инвестора по введенной фамилии | ```sql SELECT [i.name](https://i.name/) as investor\_name,  [a.name](https://a.name/) as account\_number,  a.account\_type FROM investor\_investor i JOIN investor\_account a ON [i.id](https://i.id/) = a.investor\_id WHERE [i.name](https://i.name/) ILIKE '%' |
| 8 | q\_ИнвесторыРожденныеВ90х | Вывести инвесторов, родившихся в 1990-х годах | sql<br>SELECT name, birth\_date <br>FROM investor\_investor <br>WHERE EXTRACT(YEAR FROM birth\_date) <br>BETWEEN 1990 AND 1999;<br> |
| 9 | q\_БрокерыБолее5счетов | Вывести брокеров, у которых открыто более 5 счетов | sql<br>SELECT b.name, COUNT(a.id) as account\_count<br>FROM investor\_broker b<br>JOIN investor\_account a ON b.id = a.broker\_id<br>GROUP BY b.id, b.name<br>HAVING COUNT(a.id) > 5;<br> |
| 10 | q\_КолвоСчетовПоТипам | Вывести количество счетов каждого типа | sql<br>SELECT account\_type, COUNT(\*) as count\_accounts<br>FROM investor\_account <br>GROUP BY account\_type;<br> |
| 11 | q\_ПолнаяИнформацияТранзакции | Вывести полную информацию о транзакциях | sql<br>SELECT i.name as investor\_name,<br> b.name as broker\_name,<br> a.name as account\_number,<br> ast.name as asset\_name,<br> t.operation\_type, t.amount,<br> t.transaction\_datetime<br>FROM investor\_transaction t<br>JOIN investor\_account a ON t.account\_id = a.id<br>JOIN investor\_investor i ON a.investor\_id = i.id<br>JOIN investor\_broker b ON a.broker\_id = b.id<br>LEFT JOIN investor\_asset ast ON t.asset\_id = ast.id<br>ORDER BY t.transaction\_datetime DESC;<br> |
| 12 | q\_ИнвесторыСАктивнымиСчетами | Вывести инвесторов с активными счетами | sql<br>SELECT name <br>FROM investor\_investor i<br>WHERE EXISTS (<br> SELECT 1 <br> FROM investor\_account a <br> WHERE a.investor\_id = i.id <br> AND a.status = 'active'<br>);<br> |
| 13 | q\_ИнвесторыСПочтойGmail | Вывести инвесторов с почтой на [gmail.com](https://gmail.com/) | sql<br>SELECT name, email <br>FROM investor\_investor <br>WHERE email LIKE '%@gmail.com';<br> |
| 14 | q\_ИнвесторыСТойЖеДатойРождения | Вывести пары инвесторов с одинаковой датой рождения | sql<br>SELECT i1.name as investor1,<br> i2.name as investor2,<br> i1.birth\_date<br>FROM investor\_investor i1<br>JOIN investor\_investor i2 ON <br> i1.birth\_date = i2.birth\_date <br> AND i1.id < i2.id;<br> |
| 15 | q\_ВсеБрокерыИСчета | Вывести всех брокеров и количество их счетов | sql<br>SELECT b.name as broker\_name,<br> COUNT(a.id) as account\_count<br>FROM investor\_broker b<br>LEFT JOIN investor\_account a ON b.id = a.broker\_id<br>GROUP BY b.id, b.name<br>ORDER BY account\_count DESC;<br> |
| 16 | q\_ДобавитьИнвестора | Добавить новую запись в таблицу инвесторов | sql<br>INSERT INTO investor\_investor <br>(name, birth\_date, phone, email) <br>VALUES ('Иванов Иван Иванович', <br> '1985-10-15', <br> '+7 (123) 456-78-90', <br> 'ivanov.ii@email.com');<br> |
| 17 | q\_ОбновитьСтатусСчета | Обновить статус счёта на "Закрыт" | sql<br>UPDATE investor\_account <br>SET status = 'closed' <br>WHERE name = '40817810000000001234';<br> |
| 18 | q\_СписокВсехКонтактов | Вывести единый список инвесторов и брокеров | sql<br>SELECT name, 'investor' as type <br>FROM investor\_investor<br>UNION ALL<br>SELECT name, 'broker' as type <br>FROM investor\_broker<br>ORDER BY type, name;<br> |
| 19 | q\_СоздатьТаблицуАрхив | Создать таблицу архивных транзакций | sql<br>CREATE TABLE archive\_transactions\_2024 AS <br>SELECT \* <br>FROM investor\_transaction <br>WHERE EXTRACT(YEAR FROM transaction\_datetime) = 2024;<br> |
| 20 | q\_УдалитьТранзакции | Удалить транзакции с суммой менее 100 | sql<br>DELETE FROM investor\_transaction <br>WHERE amount < 100;<br> |
| 21 | q\_СводкаПоТипамОпераций | Сводка операций по типам для инвесторов | sql<br>SELECT i.name as investor\_name,<br> SUM(CASE WHEN t.operation\_type='buy' THEN t.amount ELSE 0 END) as total\_buy,<br> SUM(CASE WHEN t.operation\_type='sell' THEN t.amount ELSE 0 END) as total\_sell<br>FROM investor\_investor i<br>LEFT JOIN investor\_account a ON i.id = a.investor\_id<br>LEFT JOIN investor\_transaction t ON a.id = t.account\_id<br>GROUP BY i.id, i.name;<br> |

## Таблица запросов

| № | Код на Python (Odoo) |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | python<br>@api.model<br>def get\_top\_expensive\_assets(self):<br> return self.env['investor.asset'].search(<br> [], order='unit\_price desc', limit=5<br> )<br> |  |  |  |  |
| 2 | python<br>@api.model<br>def get\_recent\_transactions(self):<br> return self.env['investor.transaction'].search(<br> [], limit=10, order='transaction\_datetime desc'<br> )<br> |  |  |  |  |
| 3 | python<br>@api.model<br>def get\_transactions\_by\_date\_range(self, start\_date, end\_date):<br> return self.env['investor.transaction'].search([<br> ('transaction\_datetime', '>=', start\_date),<br> ('transaction\_datetime', '<=', end\_date)<br> ])<br> |  |  |  |  |
| 4 | python<br>@api.model<br>def get\_iis\_broker\_accounts(self):<br> return self.env['investor.account'].search([<br> ('account\_type', 'in', ['iis', 'broker'])<br> ])<br> |  |  |  |  |
| 5 | python<br>@api.model<br>def get\_active\_accounts(self):<br> return self.env['investor.account'].search([<br> ('status', '=', 'active')<br> ])<br> |  |  |  |  |
| 6 | python<br>@api.model<br>def get\_investors\_without\_accounts(self):<br> all\_investors = self.env['investor.investor'].search([])<br> investors\_with\_accounts = self.env['investor.account'].search([]).mapped('investor\_id')<br> return all\_investors - investors\_with\_accounts<br> |  |  |  |  |
| 7 |  | $1 |  | '%'; ``` | python<br>@api.model<br>def get\_accounts\_by\_investor\_name(self, name\_part):<br> return self.env['investor.account'].search([<br> ('investor\_id.name', 'ilike', name\_part)<br> ])<br> |
| 8 | python<br>@api.model<br>def get\_investors\_born\_in\_90s(self):<br> return self.env['investor.investor'].search([<br> ('birth\_date', '>=', '1990-01-01'),<br> ('birth\_date', '<=', '1999-12-31')<br> ])<br> |  | | | |
| 9 | python<br>@api.model<br>def get\_brokers\_with\_many\_accounts(self):<br> brokers = self.env['investor.broker'].search([])<br> result = []<br> for broker in brokers:<br> account\_count = self.env['investor.account'].search\_count([<br> ('broker\_id', '=', broker.id)<br> ])<br> if account\_count > 5:<br> result.append(broker)<br> return result<br> |  | | | |
| 10 | python<br>@api.model<br>def get\_accounts\_by\_type(self):<br> accounts = self.env['investor.account'].read\_group(<br> [], ['account\_type'], ['account\_type']<br> )<br> return accounts<br> |  | | | |
| 11 | python<br>@api.model<br>def get\_transaction\_details(self):<br> return self.env['investor.transaction'].search\_read(<br> [], order='transaction\_datetime desc'<br> )<br> |  | | | |
| 12 | python<br>@api.model<br>def get\_investors\_with\_active\_accounts(self):<br> return self.env['investor.investor'].search([<br> ('account\_ids.status', '=', 'active')<br> ])<br> |  | | | |
| 13 | python<br>@api.model<br>def get\_gmail\_investors(self):<br> return self.env['investor.investor'].search([<br> ('email', 'ilike', '@gmail.com')<br> ])<br> |  | | | |
| 14 | python<br>@api.model<br>def get\_investors\_same\_birthdate(self):<br> investors = self.env['investor.investor'].search([])<br> result = []<br> processed = set()<br> for i1 in investors:<br> for i2 in investors:<br> if (i1.id != i2.id and <br> i1.birth\_date == i2.birth\_date and<br> (i2.id, i1.id) not in processed):<br> result.append((i1, i2))<br> processed.add((i1.id, i2.id))<br> return result<br> |  | | | |
| 15 | python<br>@api.model<br>def get\_brokers\_accounts\_count(self):<br> brokers = self.env['investor.broker'].search([])<br> result = []<br> for broker in brokers:<br> account\_count = len(broker.account\_ids)<br> result.append({<br> 'broker': broker.name,<br> 'account\_count': account\_count<br> })<br> return result<br> |  | | | |
| 16 | python<br>@api.model<br>def create\_investor(self, vals):<br> return self.env['investor.investor'].create(vals)<br># Использование:<br># self.create\_investor({<br># 'name': 'Иванов Иван Иванович',<br># 'birth\_date': '1985-10-15',<br># 'phone': '+7 (123) 456-78-90',<br># 'email': 'ivanov.ii@email.com'<br># })<br> |  | | | |
| 17 | python<br>@api.model<br>def close\_account(self, account\_number):<br> account = self.env['investor.account'].search([<br> ('name', '=', account\_number)<br> ])<br> if account:<br> account.status = 'closed'<br> |  | | | |
| 18 | python<br>@api.model<br>def get\_all\_contacts(self):<br> investors = self.env['investor.investor'].search([])<br> brokers = self.env['investor.broker'].search([])<br> return {<br> 'investors': investors.mapped('name'),<br> 'brokers': brokers.mapped('name')<br> }<br> |  | | | |
| 19 | python<br>@api.model<br>def archive\_2024\_transactions(self):<br> # В Odoo обычно используют отдельную модель для архива<br> transactions\_2024 = self.env['investor.transaction'].search([<br> ('transaction\_datetime', '>=', '2024-01-01'),<br> ('transaction\_datetime', '<=', '2024-12-31')<br> ])<br> # Архивирование через создание записей в архивной модели<br> # или пометку существующих записей<br> |  | | | |
| 20 | python<br>@api.model<br>def delete\_small\_transactions(self):<br> small\_transactions = self.env['investor.transaction'].search([<br> ('amount', '<', 100)<br> ])<br> small\_transactions.unlink()<br> |  | | | |
| 21 | python<br>@api.model<br>def get\_operations\_summary(self):<br> investors = self.env['investor.investor'].search([])<br> summary = []<br> for investor in investors:<br> accounts = investor.account\_ids<br> transactions = self.env['investor.transaction'].search([<br> ('account\_id', 'in', accounts.ids)<br> ])<br> buy\_total = sum(t.amount for t in transactions if t.operation\_type == 'buy')<br> sell\_total = sum(t.amount for t in transactions if t.operation\_type == 'sell')<br> summary.append({<br> 'investor': investor.name,<br> 'buy\_total': buy\_total,<br> 'sell\_total': sell\_total<br> })<br> return summary<br> |  | | | |

|  |  |  |
| --- | --- | --- |
| **№** | **Вид запроса** | **Шифр(ы)** |
| **1** | **Запросы с выражениями**, использующие операторы сравнения с образцом (Between, In, Like, Is Null), статистические функции, константы даты. | 3, 4, 5, 7, 8, 13 |
| **2** | **Использование логических операторов**, включающих в себя комбинацию «И», «ИЛИ» | 9, 12, 14 |
| **3** | **Запрос с вычисляемым полем** | 9, 10, 15, 18 |
| **4** | **Запрос с предикатом TOP** | 1, 2 |
| **5** | **Запрос с внутренним соединением по нескольким полям** | 11 |
| **6** | **Запрос с внутренним неэквивалентным соединением** |  |
| **7** | **Запрос с использованием внешнего соединения (LEFT, RIGHT)** | 15 |
| **8** | **Запрос с self-соединением (самообъединение)** | 14 |
| **9** | **Подзапросы** (вложенный, коррелированный) | 6, 12 |
| **10** | **Запрос с оператором UNION** | 18 |
| **11** | **Запрос с предикатом EXISTS** | 12 |
| **12** | **Запрос с сохраненным именованным запросом как источником исходных данных** |  |
| **13** | **Перекрестный запрос** | 21 |
| **14** | **Запросы с параметрами** | 3, 7 |
| **15** | **Запросы на группирование данных** (итоговые значения по всем записям и по группам) | 9, 10, 15, 21 |
| **16** | **Запросы на удаление записи** | 20 |
| **17** | **Запросы на обновление записи** | 17 |
| **18** | **Запросы на добавление записей** | 16 |
| **19** | **Запросы на создание таблицы** |  |

## Виды запросов

| № | Вид запроса | Шифр(ы) |
| --- | --- | --- |
| 1 | **Запросы с выражениями** (Between, In, Like, статистические функции) | 3, 4, 5, 7, 8, 13 |
| 2 | **Использование логических операторов** (И, ИЛИ) | 4, 5, 9, 12, 14 |
| 3 | **Запрос с вычисляемым полем** | 9, 10, 15, 18, 21 |
| 4 | **Запрос с предикатом LIMIT** | 1, 2 |
| 5 | **Запрос с внутренним соединением по нескольким полям** | 5, 11, 14 |
| 6 | **Запрос с внутренним неэквивалентным соединением** | - |
| 7 | **Запрос с использованием внешнего соединения (LEFT JOIN)** | 15, 21 |
| 8 | **Запрос с self-соединением (самообъединение)** | 14 |
| 9 | **Подзапросы** (вложенный, коррелированный) | 6, 12 |
| 10 | **Запрос с оператором UNION** | 18 |
| 11 | **Запрос с предикатом EXISTS** | 6, 12 |
| 12 | **Запрос с сохраненным именованным запросом как источником исходных данных** | - |
| 13 | **Перекрестный запрос** | 21 |
| 14 | **Запросы с параметрами** | 3, 7 |
| 15 | **Запросы на группирование данных** (итоговые значения по группам) | 9, 10, 15, 21 |
| 16 | **Запросы на удаление записи** | 20 |
| 17 | **Запросы на обновление записи** | 17 |
| 18 | **Запросы на добавление записей** | 16 |
| 19 | **Запросы на создание таблицы** | 19 |