**Реляционная модель базы данных – PC\_club:**

* computer\_clubs (club\_id, room\_rental\_agreement\_number, region, city, street, building, job\_status, office\_number)
* employees (employee\_id, contract\_number, position, human\_id)
* clients (client\_id, human\_id, account\_score)
* humans (human\_id, name, surname, otchestvo, phone\_number, email, age)
* subscriptions (subscription\_id, unique\_subscription\_number, type, cost, validity\_period, registration\_date, status, computer\_id, employee\_id, client\_id, comp\_club\_id)
* computers (computer\_id, serial\_number, processor\_model, ram\_capacity, video\_card\_model, hard\_disk\_capacity, status, comp\_club\_id)

**Модель сущностей с полями в проекте:**

* Client (clientId, accountScore)
* Computer(computerId, serialNumber, processorModel, ramCapacity, videoCardModel, hardDiskCapacity, status)
* ComputerClub(clubId, roomRentalAgreementNumber, region, city, building, jobStatus, officeNumber)
* Employee(employeeId, contractNumber, position )
* Human(humanId, name, surname, otchestvo, phoneNumber, email, age )
* Subscription(subscriptionId, uniqueSubscriptionNumber, type, cost, validityPeriod, registrationDate, status)

**Запросы на поиск:**

1. Запрос на поиск информации из таблицы "Computer Club" для поиска клубов в городе «Москва»:  
SQL:

SELECT \* FROM computer\_clubs WHERE city = 'Москва';

2. Запрос на поиск клиентов и абонементов, которые оформили клиенты с ID от 1 до 8:

SQL:

SELECT c.\*, s.\*

FROM clients c

JOIN subscriptions s ON c.client\_id = s.client\_id

WHERE c.client\_id BETWEEN 1 AND 8;

4. Вывод записей компьютерных клубов с записями об id оформленных там аренд и Id сотрудника оформившего аренду с должностью продавец:

SQL:

SELECT cc.\*, s.subscription\_id, e.employee\_id

FROM computer\_clubs cc

JOIN subscriptions s ON cc.club\_id = s.comp\_club\_id

JOIN employees e ON s.employee\_id = e.employee\_id

WHERE e.position = 'Продавец';

5. Запрос на поиск записи аренды subscriptions (subscription\_id, unique\_subscription\_number), а также связанные данные о клиенте (client), компьютерном клубе (computer\_club) и сотруднике (employee).

SQL:

SELECT s.subscription\_id, s.unique\_subscription\_number, c.\*, cc.\*, e.\*

FROM subscriptions s

JOIN clients c ON s.client\_id = c.client\_id

JOIN computer\_clubs cc ON s.comp\_club\_id = cc.club\_id

JOIN employees e ON s.employee\_id = e.employee\_id;

**На изменение, удаление, добавление и каскадное удаление.**

1. Изменение почты у Human с id = 2.

SQL:

UPDATE humans

SET email = 'новая\_почта@example.com'

WHERE human\_id = 2;

HQL:

2. Добавление новой точки компьютерного клуба.

SQL:

INSERT INTO computer\_clubs (club\_id, room\_rental\_agreement\_number, region, city, street, building, job\_status, office\_number)

VALUES (новый\_club\_id, 'новый\_номер\_договора', 'регион', 'город', 'улица', 'здание', 'статус\_работы', 'номер\_офиса');

HQL:

3. Удаление компьютера id = 2.

SQL:

DELETE FROM computers

WHERE serial\_number = '76000002';

HQL:

**Три примера запросов с использованием функций агрегирования и двух таблиц:**

1. Запрос на подсчет общего количества аренд для каждого клиента:

SQL:  
SELECT c.client\_id, COUNT(s.subscription\_id) AS total\_rentals

FROM clients c

JOIN subscriptions s ON c.client\_id = s.client\_id

GROUP BY c.client\_id;

2. Запрос на вычисление средней стоимости абонемента для каждого сотрудника с точностью средней стоимости 2 числа после запятой:

SQL:  
SELECT e.employee\_id, ROUND(AVG(s.cost), 2) AS average\_cost

FROM employees e

JOIN subscriptions s ON e.employee\_id = s.employee\_id

GROUP BY e.employee\_id;

3. Запрос на поиск клиентов с максимальной общей стоимостью абонемента.

SQL:  
SELECT c.client\_id, SUM(s.cost) AS total\_cost

FROM clients c

JOIN subscriptions s ON c.client\_id = s.client\_id

GROUP BY c.client\_id

ORDER BY total\_cost DESC;

**Запросы с использованием IN и EXISTS:**

1. Запрос с подзапросом на поиск всех записи из таблицы computers. Подзапрос выбирает Computer\_ID из таблицы subsriptions, где cost больше 4000:

SQL:

SELECT \*

FROM computers

WHERE computer\_id IN (

SELECT computer\_id

FROM subscriptions

WHERE cost > 4000

);

2. Запрос на поиск, где выбираются все записи из таблицы subscriptions, для которых существует запись в таблице employees, подходящее под условие, что employee\_id совпадает и position равна 'Продавец':

SQL:

SELECT \*

FROM subscriptions

WHERE EXISTS (

SELECT \*

FROM employees

WHERE subscriptions.employee\_id = employees.employee\_id

AND employees.position = 'Продавец'

);