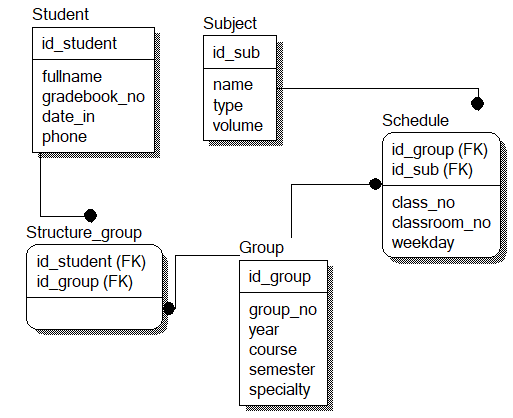
**Вариант 2.**

**Задача 1:**

1. Задана группа таблиц на рисунке ниже. Необходимо на языке SQL написать команды по созданию таблиц (типы столбцов указаны ниже) и заполнить каждую из таблиц 3 записями, удовлетворяющими ограничениям целостности, описанным ниже.



1. Первичными ключами считаются следующие атрибуты (тип – некоторое целое число): id\_student, id\_group\_list, id\_sub в таблицах Student, Group, Subject соответственно.

Внешними ключами считаются атрибуты: id\_student и id\_group в таблице Structure\_group; id\_group и id\_sub в таблице Schedule.

1. Перечисленные далее атрибуты не могут быть пустыми: fullname (строка), gradebook\_no (целое число), date\_in (дата, таб. Student), name (строка, таб. Subject), class\_no (число), classroom\_no, weekday (строки, таб. Schedule), group\_no (строка), course (число), year (число).
2. Атрибут type (таб. Subject) может принимать только значения из списка – “лк” или “лб”.

Атрибуты class\_no и course числа от 1 до 10.

Для оставшихся атрибутов считать тип – строки.

CREATE TABLE Student (

id\_student INT PRIMARY KEY,

fullname VARCHAR(255) NOT NULL,

gradebook\_no INT NOT NULL,

date\_in DATE NOT NULL,

phone VARCHAR(10)

);

CREATE TABLE Subject (

id\_sub INT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

type VARCHAR(10) CHECK (type IN ('лк', 'лб')),

volume INT

);

CREATE TABLE Group\_ (

id\_group INT PRIMARY KEY,

group\_no VARCHAR(255) NOT NULL,

year INT NOT NULL,

course INT CHECK (course BETWEEN 1 AND 10) NOT NULL,

semester VARCHAR(255),

specialty VARCHAR(255)

);

CREATE TABLE Structure\_group (

id\_student INT NOT NULL,

id\_group INT NOT NULL,

FOREIGN KEY (id\_student) REFERENCES Student(id\_student),

FOREIGN KEY (id\_group) REFERENCES Group\_(id\_group),

PRIMARY KEY (id\_student, id\_group)

);

CREATE TABLE Schedule (

id\_group INT NOT NULL,

id\_sub INT NOT NULL,

class\_no INT CHECK (class\_no BETWEEN 1 AND 10) NOT NULL,

classroom\_no VARCHAR(255) NOT NULL,

weekday VARCHAR(255) NOT NULL,

FOREIGN KEY (id\_group) REFERENCES Group\_(id\_group),

FOREIGN KEY (id\_sub) REFERENCES Subject(id\_sub),

PRIMARY KEY (id\_group, id\_sub, class\_no)

);

-- Insert data into tables

INSERT INTO Student (id\_student, fullname, gradebook\_no, date\_in, phone) VALUES

(1, 'Умный стулент', 12345, TO\_DATE('2024-12-12', 'YYYY-MM-DD'), '123-45-67');

INSERT INTO Student (id\_student, fullname, gradebook\_no, date\_in, phone) VALUES

(2, 'Вася Пупкин', 67891, TO\_DATE('2024-12-13', 'YYYY-MM-DD'), '765-43-21');

INSERT INTO Student (id\_student, fullname, gradebook\_no, date\_in, phone) VALUES

(3, 'Дядя Толик', 23456, TO\_DATE('2024-12-14', 'YYYY-MM-DD'), '111-22-33');

INSERT INTO Subject (id\_sub, name, type, volume) VALUES

(1, 'Математика', 'лк', 100);

INSERT INTO Subject (id\_sub, name, type, volume) VALUES

(2, 'Физика', 'лб', 110);

INSERT INTO Subject (id\_sub, name, type, volume) VALUES

(3, 'Информатика', 'лк', 120);

INSERT INTO Group\_ (id\_group, group\_no, year, course, semester, specialty) VALUES

(1, '101', 2024, 1, 'Первый', 'Математики');

INSERT INTO Group\_ (id\_group, group\_no, year, course, semester, specialty) VALUES

(2, '102', 2024, 2, 'Первый', 'Физики');

INSERT INTO Group\_ (id\_group, group\_no, year, course, semester, specialty) VALUES

(3, '103', 2024, 1, 'Второй', 'Программисты');

INSERT INTO Structure\_group (id\_student, id\_group) VALUES

(1, 1);

INSERT INTO Structure\_group (id\_student, id\_group) VALUES

(2, 2);

INSERT INTO Structure\_group (id\_student, id\_group)VALUES

(3, 3);

INSERT INTO Schedule (id\_group, id\_sub, class\_no, classroom\_no, weekday) VALUES

(1, 1, 1, '101', 'Пн');

INSERT INTO Schedule (id\_group, id\_sub, class\_no, classroom\_no, weekday) VALUES

(1, 2, 1, '102', 'Вт');

INSERT INTO Schedule (id\_group, id\_sub, class\_no, classroom\_no, weekday) VALUES

(2, 1, 2, '201', 'Ср');

-- Display the Student table

SELECT \* FROM Student;

-- Display the Subject table

SELECT \* FROM Subject;

-- Display the Group\_ table

SELECT \* FROM Group\_;

-- Display the Structure\_group table

SELECT \* FROM Structure\_group;

-- Display the Schedule table

SELECT \* FROM Schedule;

--DROP TABLE Schedule CASCADE CONSTRAINTS;

--DROP TABLE Structure\_group CASCADE CONSTRAINTS;

--DROP TABLE Group\_ CASCADE CONSTRAINTS;

--DROP TABLE Subject CASCADE CONSTRAINTS;

--DROP TABLE Student CASCADE CONSTRAINTS;

**Задача 2:**

Необходимо на языке SQL написать запросы к таблицам, расположенным в схеме HR:

1. Вывести имя (first\_name), фамилию (last\_name), дату приема на работу (hire\_date), зарплату (salary) и комиссионные (commission\_pct) для сотрудников, нанятых после 22.10.2005. Отсортировать полученные результаты по дате приема на работу.

SELECT first\_name, last\_name, hire\_date, salary, commission\_pct

FROM employees

WHERE hire\_date > TO\_DATE('2005-10-22', 'YYYY-MM-DD')

ORDER BY hire\_date;

1. Вывести фамилии (last\_name), наименование их должности (job\_title), наименование департамента (department\_name) для программистов (Programmer), работающих в Саутлейк (Southlake).

SELECT e.last\_name, j.job\_title, d.department\_name

FROM employees e

JOIN jobs j ON e.job\_id = j.job\_id

JOIN departments d ON e.department\_id = d.department\_id

WHERE j.job\_title = 'Programmer' AND d.location\_id IN (

SELECT location\_id FROM locations WHERE city = 'Southlake'

);

1. Для каждого города определить количество департаментов, расположенных в этом городе. Необходимо вывести наименование города (city) и количество департаментов, при этом отбросить города, в которых расположен единственный департамент.

SELECT l.city, COUNT(d.department\_id) AS department\_count

FROM locations l

JOIN departments d ON l.location\_id = d.location\_id

GROUP BY l.city

HAVING COUNT(d.department\_id) > 1;

1. Вывести полную информацию по сотрудникам (все столбцы таблицы emploeeys), являющиеся руководителями некоторых других сотрудников. При этом данные руководители должны быть наняты позднее некоторых из своих подчиненных

SELECT e.\*

FROM employees e

WHERE e.employee\_id IN (

SELECT DISTINCT manager\_id

FROM employees

WHERE manager\_id IS NOT NULL

) AND e.hire\_date > (

SELECT MIN(e2.hire\_date)

FROM employees e2

WHERE e2.manager\_id = e.employee\_id

);

**Задача 3(текст и рисунок на следующей странице):**

Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание

Написать функцию на языке PL/SQL, рассчитывающую общую сумму по всем заказам за указанный период для некоторого клиента, заданного кодом. В сумму заказов могут входить только заказы в статусе “Исполнен”. Период описывается двумя датами – датой начала периода и датой окончания.

-- Создание таблицы КЛИЕНТЫ

CREATE TABLE КЛИЕНТЫ (

Код\_клиента NUMBER PRIMARY KEY,

ФИО VARCHAR2(100),

Номер\_паспорта VARCHAR2(200),

Серия\_паспорта VARCHAR2(100),

Телефон VARCHAR2(150),

Адрес VARCHAR2(255),

E\_mail VARCHAR2(100),

Пароль VARCHAR2(100)

);

-- Создание таблицы СОТРУДНИКИ

CREATE TABLE СОТРУДНИКИ (

Код\_сотрудника NUMBER PRIMARY KEY,

Серия\_паспорта VARCHAR2(100),

ФИО VARCHAR2(100),

Номер\_паспорта VARCHAR2(200),

Фото BLOB,

Дата\_рождения DATE,

Телефон VARCHAR2(150),

Должность VARCHAR2(500),

Адрес VARCHAR2(255),

ЗП NUMBER

);

-- Создание таблицы ЗАКАЗЫ

CREATE TABLE ЗАКАЗЫ (

Номер\_заказа NUMBER PRIMARY KEY,

Код\_клиента NUMBER,

Общая\_стоимость NUMBER,

Адрес\_доставки VARCHAR2(255),

Статус VARCHAR2(50),

Способ\_оплаты VARCHAR2(500),

Код\_сотрудника NUMBER,

FOREIGN KEY (Код\_клиента) REFERENCES КЛИЕНТЫ(Код\_клиента),

FOREIGN KEY (Код\_сотрудника) REFERENCES СОТРУДНИКИ(Код\_сотрудника)

);

-- Создание таблицы ТОВАРЫ

CREATE TABLE ТОВАРЫ (

Артикул\_товара NUMBER PRIMARY KEY,

Фото BLOB,

Стоимость NUMBER,

Описание VARCHAR2(255)

);

-- Создание таблицы ДЕТАЛИЗАЦИЯ ЗАКАЗА

CREATE TABLE ДЕТАЛИЗАЦИЯ\_ЗАКАЗА (

Номер\_детализации NUMBER PRIMARY KEY,

Артикул\_товара NUMBER,

Номер\_заказа NUMBER,

Количество NUMBER,

Стоимость NUMBER,

FOREIGN KEY (Артикул\_товара) REFERENCES ТОВАРЫ(Артикул\_товара),

FOREIGN KEY (Номер\_заказа) REFERENCES ЗАКАЗЫ(Номер\_заказа)

);

-- Создание таблицы ИСТОРИЯ ЗАКАЗОВ

CREATE TABLE ИСТОРИЯ\_ЗАКАЗОВ (

Номер\_истории NUMBER PRIMARY KEY,

Номер\_заказа NUMBER,

Дата\_и\_время\_формирования TIMESTAMP,

Код\_клиента NUMBER,

Адрес VARCHAR2(255),

Статус VARCHAR2(500),

FOREIGN KEY (Номер\_заказа) REFERENCES ЗАКАЗЫ(Номер\_заказа),

FOREIGN KEY (Код\_клиента) REFERENCES КЛИЕНТЫ(Код\_клиента)

);

INSERT INTO КЛИЕНТЫ (Код\_клиента, ФИО, Номер\_паспорта, Серия\_паспорта, Телефон, Адрес, E\_mail, Пароль)

VALUES (1, 'Иван Иванов', '1234567890', '1234', '+79991112233', 'ул. Ленина, 1', 'ivan.ivanov@mail.ru', 'password123');

INSERT INTO КЛИЕНТЫ (Код\_клиента, ФИО, Номер\_паспорта, Серия\_паспорта, Телефон, Адрес, E\_mail, Пароль)

VALUES (2, 'Петр Петров', '9876543210', '5678', '+79994445566', 'пр. Мира, 10', 'petr.petrov@mail.ru', 'securepass');

INSERT INTO КЛИЕНТЫ (Код\_клиента, ФИО, Номер\_паспорта, Серия\_паспорта, Телефон, Адрес, E\_mail, Пароль)

VALUES (3, 'Сидор Сидоров', '0000000000', '0000', '+79997778899', 'ул. Октябрьская, 25', 'sidr.sidorov@mail.ru', 'qwerty');

INSERT INTO СОТРУДНИКИ (Код\_сотрудника, Серия\_паспорта, ФИО, Номер\_паспорта, Фото, Дата\_рождения, Телефон, Должность, Адрес, ЗП)

VALUES (101, '1111', 'Андрей Андреев', '1111222222', NULL, TO\_DATE('1985-03-15', 'YYYY-MM-DD'), '+79991234567', 'Менеджер', 'ул. Пушкина, 1', 50000);

INSERT INTO СОТРУДНИКИ (Код\_сотрудника, Серия\_паспорта, ФИО, Номер\_паспорта, Фото, Дата\_рождения, Телефон, Должность, Адрес, ЗП)

VALUES (102, '2222', 'Елена Еленова', '2222333333', NULL, TO\_DATE('1990-08-20', 'YYYY-MM-DD'), '+79997654321', 'Курьер', 'ул. Лермонтова, 5', 35000);

INSERT INTO СОТРУДНИКИ (Код\_сотрудника, Серия\_паспорта, ФИО, Номер\_паспорта, Фото, Дата\_рождения, Телефон, Должность, Адрес, ЗП)

VALUES (103, '3333', 'Сергей Сергеев', '3333444444', NULL, TO\_DATE('1988-11-10', 'YYYY-MM-DD'), '+79991010101', 'Администратор', 'ул. Толстого, 12', 60000);

INSERT INTO ЗАКАЗЫ (Номер\_заказа, Код\_клиента, Общая\_стоимость, Адрес\_доставки, Статус, Способ\_оплаты, Код\_сотрудника)

VALUES (1, 1, 1500, 'ул. Ленина, 1', 'Обработка', 'Карта', 101);

INSERT INTO ЗАКАЗЫ (Номер\_заказа, Код\_клиента, Общая\_стоимость, Адрес\_доставки, Статус, Способ\_оплаты, Код\_сотрудника)

VALUES (2, 2, 2000, 'пр. Мира, 10', 'Доставка', 'Наличные', 102);

INSERT INTO ЗАКАЗЫ (Номер\_заказа, Код\_клиента, Общая\_стоимость, Адрес\_доставки, Статус, Способ\_оплаты, Код\_сотрудника)

VALUES (3, 3, 2500, 'ул. Октябрьская, 25', 'Выполнен', 'Безналичный', 103);

INSERT INTO ТОВАРЫ (Артикул\_товара, Фото, Стоимость, Описание)

VALUES (1001, NULL, 500, 'Товар А');

INSERT INTO ТОВАРЫ (Артикул\_товара, Фото, Стоимость, Описание)

VALUES (1002, NULL, 1000, 'Товар Б');

INSERT INTO ТОВАРЫ (Артикул\_товара, Фото, Стоимость, Описание)

VALUES (1003, NULL, 1500, 'Товар В');

INSERT INTO ДЕТАЛИЗАЦИЯ\_ЗАКАЗА (Номер\_детализации, Артикул\_товара, Номер\_заказа, Количество, Стоимость)

VALUES (1, 1001, 1, 1, 500);

INSERT INTO ДЕТАЛИЗАЦИЯ\_ЗАКАЗА (Номер\_детализации, Артикул\_товара, Номер\_заказа, Количество, Стоимость)

VALUES (2, 1002, 2, 1, 1000);

INSERT INTO ДЕТАЛИЗАЦИЯ\_ЗАКАЗА (Номер\_детализации, Артикул\_товара, Номер\_заказа, Количество, Стоимость)

VALUES (3, 1003, 3, 1, 1500);

INSERT INTO ИСТОРИЯ\_ЗАКАЗОВ (Номер\_истории, Номер\_заказа, Дата\_и\_время\_формирования, Код\_клиента, Адрес, Статус)

VALUES (1, 1, CURRENT\_TIMESTAMP, 1, 'ул. Ленина, 1', 'Создан');

INSERT INTO ИСТОРИЯ\_ЗАКАЗОВ (Номер\_истории, Номер\_заказа, Дата\_и\_время\_формирования, Код\_клиента, Адрес, Статус)

VALUES (2, 2, CURRENT\_TIMESTAMP, 2, 'пр. Мира, 10', 'Обработан');

INSERT INTO ИСТОРИЯ\_ЗАКАЗОВ (Номер\_истории, Номер\_заказа, Дата\_и\_время\_формирования, Код\_клиента, Адрес, Статус)

VALUES (3, 3, CURRENT\_TIMESTAMP, 3, 'ул. Октябрьская, 25', 'Доставлен');

-- Вывод данных из таблицы КЛИЕНТЫ

SELECT \* FROM КЛИЕНТЫ;

-- Вывод данных из таблицы СОТРУДНИКИ

SELECT \* FROM СОТРУДНИКИ;

-- Вывод данных из таблицы ЗАКАЗЫ

SELECT \* FROM ЗАКАЗЫ;

-- Вывод данных из таблицы ТОВАРЫ

SELECT \* FROM ТОВАРЫ;

-- Вывод данных из таблицы ДЕТАЛИЗАЦИЯ ЗАКАЗА

SELECT \* FROM ДЕТАЛИЗАЦИЯ\_ЗАКАЗА;

-- Вывод данных из таблицы ИСТОРИЯ ЗАКАЗОВ

SELECT \* FROM ИСТОРИЯ\_ЗАКАЗОВ;

-- Удаление таблицы ИСТОРИЯ ЗАКАЗОВ

DROP TABLE ИСТОРИЯ\_ЗАКАЗОВ CASCADE CONSTRAINTS;

-- Удаление таблицы ДЕТАЛИЗАЦИЯ ЗАКАЗА

DROP TABLE ДЕТАЛИЗАЦИЯ\_ЗАКАЗА CASCADE CONSTRAINTS;

-- Удаление таблицы ЗАКАЗЫ

DROP TABLE ЗАКАЗЫ CASCADE CONSTRAINTS;

-- Удаление таблицы СОТРУДНИКИ

DROP TABLE СОТРУДНИКИ CASCADE CONSTRAINTS;

-- Удаление таблицы ТОВАРЫ

DROP TABLE ТОВАРЫ CASCADE CONSTRAINTS;

-- Удаление таблицы КЛИЕНТЫ

DROP TABLE КЛИЕНТЫ CASCADE CONSTRAINTS;

CREATE OR REPLACE FUNCTION общая\_сумма\_заказов\_клиента (

p\_код\_клиента КЛИЕНТЫ.Код\_клиента%TYPE,

p\_дата\_начала DATE,

p\_дата\_окончания DATE

) RETURN NUMBER

IS

v\_общая\_сумма NUMBER := 0;

BEGIN

SELECT SUM(О.Общая\_стоимость)

INTO v\_общая\_сумма

FROM ЗАКАЗЫ О

JOIN ИСТОРИЯ\_ЗАКАЗОВ И ON О.Номер\_заказа = И.Номер\_заказа

WHERE И.Код\_клиента = p\_код\_клиента

AND И.Дата\_и\_время\_формирования BETWEEN p\_дата\_начала AND p\_дата\_окончания

AND О.Статус = 'Выполнен';

RETURN v\_общая\_сумма;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 0;

WHEN OTHERS THEN

RAISE;

END;

/

SELECT общая\_сумма\_заказов\_клиента(3, TO\_DATE('01.01.2023', 'DD.MM.RRRR'), TO\_DATE('31.12.2024', 'DD.MM.RRRR')) FROM dual;