```
SQL>
SQL> -- Question 1
SQL> UPDATE agent SET country = 'India' WHERE agent_code = 'Ac002';

1 row updated.

SQL> UPDATE agent SET country = 'London' WHERE agent_code = 'Ac004';

1 row updated.

SQL> ALTER TABLE agent ADD PRIMARY KEY (agent_code);

Table altered.

SQL> UPDATE orders SET agent_code = 'Ac002' WHERE ord_num = '010';

1 row updated.

SQL> UPDATE orders SET agent_code = 'Ac004' WHERE ord_num = '009';

1 row updated.

SQL> ALTER TABLE orders ADD FOREIGN KEY (agent_code) REFERENCES agent(agent_code);

Table altered.

SQL>
SQL> -- 1.a: Find ord_num, ord_amount, ord_date, cust_code and agent_code lives in same country or working area is same.
SQL> SELECT ord_num, ord_amount, ord_date, cust_code, agent_code FROM orders
  2  WHERE agent_code IN (
```

```
  3      SELECT a1.agent_code FROM agent a1, agent a2
  4      WHERE a1.agent_code <> a2.agent_code AND
  5      (a1.working_area = a2.working_area OR a1.country = a2.country)
  6  );

ORD ORD_AMOUNT ORD_DATE             CUST_ AGENT_CODE
--- ---------- -------------------- ----- ----------
004        200 15-aug-2020          C004  Ac001
007        600 17-sept-2020         C006  Ac003
008        700 19-feb-2019          C007  Ac005
009      10000 21-march-2010        C009  Ac004
010         20 21-april-2012        C006  Ac002

SQL>
SQL> -- 1.b: Retrive ord_num, ord_amount, cust_code and agent_code from the table orders where the
SQL> -- agent_code of orders table must be the same agent_code of agents table and agent_name of
SQL> -- agents table have atleast one ‘a’ having different working_area.
SQL> SELECT ord_num, ord_amount, cust_code, agent_code FROM orders
  2  WHERE agent_code IN (
  3      SELECT agent_code FROM agent WHERE agent_name LIKE '%a%'
  4      AND working_area IN (
  5          SELECT working_area FROM agent
  6          GROUP BY working_area HAVING COUNT(agent_code) = 1
  7      )
  8  );

ORD ORD_AMOUNT CUST_ AGENT_CODE
--- ---------- ----- ----------
008        700 C007  Ac005
009      10000 C009  Ac004
```

```
SQL>
SQL>
SQL> -- Question 2
SQL> ALTER TABLE employees ADD PRIMARY KEY (employee_id);

Table altered.

SQL>
SQL> -- 2.a: Display the employee_id, manager_id, first_name and last_name of those employees
SQL> -- who manage other employees having individual salary less than average salary of
SQL> -- person whose last_name starts with ‘p’
SQL> SELECT employee_id, manager_id, first_name, last_name FROM employees
  2  WHERE manager_id IN (
  3      SELECT employee_id FROM employees
  4      WHERE salary < (SELECT AVG(salary) FROM employees WHERE last_name LIKE 'P%')
  5  );

no rows selected

SQL>
SQL>
SQL> -- Question 3
SQL> UPDATE salesman SET city = 'Chennai' WHERE salesman_id = 'si123@19';

1 row updated.

SQL> UPDATE salesman SET city = 'Kolkata' WHERE salesman_id = 'si123@67';

1 row updated.
```

```
SQL> ALTER TABLE salesman ADD PRIMARY KEY (salesman_id);

Table altered.

SQL> ALTER TABLE sales_orders ADD CONSTRAINT sales_fk FOREIGN KEY (salesman_id) REFERENCES salesman(salesman_id);

Table altered.

SQL>
SQL> -- 3.a: Display all the orders for the salesman who belongs to the same city and the individual
SQL> -- commission of salesman is greater than the average commission of city.
SQL> SELECT * FROM sales_orders
  2  WHERE salesman_id IN (
  3      SELECT s1.salesman_id FROM salesman s1, salesman s2
  4      WHERE s1.salesman_id != s2.salesman_id AND
  5      s1.city = s2.city AND
  6      s1.commission > (SELECT AVG(commission) FROM salesman WHERE city = s1.city)
  7  );

    ORD_NO  PURCH_AMT ORD_DATE             CUSTO SALESMAN_I
---------- ---------- -------------------- ----- ----------
       123        600 20-aug-2010          003cd si123@19
       576        750 20-feb-2018          004cd si123@19
       700        745 26-jan-2021          007cd si123@09

SQL>
SQL> -- 3.b: Delete the salesman_id from table salesman whose commisson is greater than 0.2 and
SQL> -- set NA for the values not available in table orders.
SQL> ALTER TABLE sales_orders DISABLE CONSTRAINT sales_fk;
```

```
Table altered.

SQL> DELETE FROM salesman WHERE commission > 0.2;

6 rows deleted.

SQL> UPDATE sales_orders SET salesman_id = 'NA'
  2  WHERE salesman_id NOT IN (SELECT salesman_id FROM salesman);

4 rows updated.

SQL>
SQL> SELECT * FROM salesman;

SALESMAN_I NAME                 CITY            COMMISSION
---------- -------------------- --------------- ----------
si123@26   Paul Adam            London                  .1

SQL> SELECT * FROM sales_orders;

    ORD_NO  PURCH_AMT ORD_DATE            CUSTO SALESMAN_I
---------- ---------- -------------------- ----- ----------
       123        600 20-aug-2010          003cd NA
       576        750 20-feb-2018          004cd NA
       579        800 20-may-2012          004cd si123@26
       600      60000 20-jan-2021          006cd NA
       700        745 26-jan-2021          007cd NA
       800        860 29-jan-2019          007cd si123@26
```