**Cross Join**

**Query:**

SELECT \* FROM CUSTOMER\_TABLE CROSS JOIN AGENTS;

**Explanation:** This query returns the Cartesian product of CUSTOMER\_TABLE and AGENTS. Every row from CUSTOMER\_TABLE is combined with every row from AGENTS.

**Query 1:**

SELECT C.CUST\_NAME, O.ORD\_NUM

FROM CUSTOMER\_TABLE C CROSS JOIN ORDERS O;

**Query 2:**

SELECT C.CUST\_NAME, A.AGENT\_NAME

FROM CUSTOMER\_TABLE C CROSS JOIN AGENTS A

WHERE A.COUNTRY = 'India';

**Query 3:**

SELECT \* FROM CUSTOMER\_TABLE CROSS JOIN AGENTS

WHERE CUSTOMER\_TABLE.CUST\_COUNTRY = AGENTS.COUNTRY;

**Query 4:**

SELECT C.CUST\_NAME, A.AGENT\_NAME

FROM CUSTOMER\_TABLE C CROSS JOIN AGENTS A

ORDER BY C.CUST\_NAME;

Query 5:

SELECT \* FROM CUSTOMER\_TABLE C CROSS JOIN AGENTS A

WHERE C.WORKING\_AREA = A.WORKING\_AREA;

**Equi Join**

**Query:**

SELECT C.CUST\_NAME, A.AGENT\_NAME

FROM CUSTOMER\_TABLE C, AGENTS A

WHERE C.AGENT\_CODE = A.AGENT\_CODE;

**Explanation:** This equi-join matches rows from CUSTOMER\_TABLE and AGENTS where AGENT\_CODE is equal in both tables.

**Equi Join**

**Query 1:**

SELECT O.ORD\_NUM, C.CUST\_NAME

FROM ORDERS O, CUSTOMER\_TABLE C

WHERE O.CUST\_CODE = C.CUST\_CODE;

**Q1:** What is this query fetching? **A1:** Order numbers with corresponding customer names.

**Query 2:**

SELECT A.AGENT\_NAME, O.ORD\_AMOUNT

FROM AGENTS A, ORDERS O

WHERE A.AGENT\_CODE = O.AGENT\_CODE;

**Q2:** What does this query retrieve? **A2:** Agent names with the corresponding order amounts.

**Query 3:**

SELECT C.CUST\_NAME, A.WORKING\_AREA

FROM CUSTOMER\_TABLE C, AGENTS A

WHERE C.AGENT\_CODE = A.AGENT\_CODE;

**Q3:** What is the significance of WHERE C.AGENT\_CODE = A.AGENT\_CODE? **A3:** It ensures only rows with matching AGENT\_CODE values from both tables are returned.

**Query 4:**

SELECT C.CUST\_NAME, O.ORD\_DATE

FROM CUSTOMER\_TABLE C, ORDERS O

WHERE C.CUST\_CODE = O.CUST\_CODE

AND O.ORD\_DATE > TO\_DATE('2008-07-01', 'YYYY-MM-DD');

**Q4:** What additional condition does this query impose? **A4:** It filters orders placed after July 1, 2008.

**Query 5:**

SELECT C.CUST\_NAME, A.COMMISSION

FROM CUSTOMER\_TABLE C, AGENTS A

WHERE C.AGENT\_CODE = A.AGENT\_CODE

AND A.COMMISSION > 0.10;

**Q5:** What does this query filter on? **A5:** It filters results to include only those agents with a commission greater than 10%.

**Inner Join**

**Query:**

SELECT C.CUST\_NAME, A.AGENT\_NAME

FROM CUSTOMER\_TABLE C

INNER JOIN AGENTS A

ON C.AGENT\_CODE = A.AGENT\_CODE;

**Explanation:** Similar to the equi-join, this query uses an INNER JOIN to retrieve customers and their respective agents.

**Inner Join**

**Query 1:**

SELECT O.ORD\_NUM, C.CUST\_NAME

FROM ORDERS O

INNER JOIN CUSTOMER\_TABLE C

ON O.CUST\_CODE = C.CUST\_CODE;

**Q1:** What will this query return? **A1:** Order numbers along with corresponding customer names.

**Query 2:**

SELECT C.CUST\_NAME, A.AGENT\_NAME

FROM CUSTOMER\_TABLE C

INNER JOIN AGENTS A

ON C.AGENT\_CODE = A.AGENT\_CODE

WHERE A.WORKING\_AREA = 'New York';

**Q2:** What does the WHERE clause do in this query? **A2:** It filters the results to include only agents working in New York.

**Query 3:**

SELECT A.AGENT\_NAME, O.ORD\_AMOUNT

FROM AGENTS A

INNER JOIN ORDERS O

ON A.AGENT\_CODE = O.AGENT\_CODE

WHERE O.ORD\_AMOUNT > 1000;

**Q3:** What is the purpose of O.ORD\_AMOUNT > 1000? **A3:** It limits the result to orders with amounts greater than 1000.

**Query 4:**

SELECT C.CUST\_NAME, O.ORD\_DESCRIPTION

FROM CUSTOMER\_TABLE C

INNER JOIN ORDERS O

ON C.CUST\_CODE = O.CUST\_CODE

WHERE O.ORD\_DESCRIPTION LIKE '%SOD%';

**Q4:** What does the LIKE '%SOD%' clause filter? **A4:** It filters orders where the description contains "SOD".

**Query 5:**

SELECT C.CUST\_NAME, A.COUNTRY

FROM CUSTOMER\_TABLE C

INNER JOIN AGENTS A

ON C.AGENT\_CODE = A.AGENT\_CODE

AND C.CUST\_COUNTRY = A.COUNTRY;

**Q5:** What additional join condition is applied here? **A5:** It requires that the customer’s country matches the agent’s country.

**Natural Join**

**Query:**

SELECT \*

FROM CUSTOMER\_TABLE NATURAL JOIN AGENTS;

**Explanation:** This NATURAL JOIN automatically joins tables based on columns with the same names and data types, in this case, AGENT\_CODE.

**Natural Join**

**Query 1:**

SELECT \*

FROM ORDERS NATURAL JOIN CUSTOMER\_TABLE;

**Q1:** What columns will the NATURAL JOIN be based on? **A1:** The join will be based on CUST\_CODE, which is common in both tables.

**Query 2:**

SELECT \*

FROM AGENTS NATURAL JOIN CUSTOMER\_TABLE

WHERE COUNTRY = 'USA';

**Q2:** What does this query return? **A2:** It returns customers and agents from the USA where their country is the same.

**Query 3:**

SELECT CUST\_NAME, WORKING\_AREA

FROM CUSTOMER\_TABLE NATURAL JOIN AGENTS;

**Q3:** What type of data will this query return? **A3:** Customer names along with their working areas, where the working area is the same in both tables.

**Query 4:**

SELECT \*

FROM CUSTOMER\_TABLE NATURAL JOIN AGENTS

WHERE WORKING\_AREA = 'London';

**Q4:** What is filtered by WHERE WORKING\_AREA = 'London'? **A4:** It filters the results to include only entries where the working area is London.

**Query 5:**

SELECT CUST\_NAME, PHONE\_NO

FROM CUSTOMER\_TABLE NATURAL JOIN AGENTS

WHERE COMMISSION > 0.05;

**Q5:** What does this query retrieve? **A5:** Customer names and phone numbers where the agent's commission is greater than 5%.

**Using Clause**

**Query:**

SELECT C.CUST\_NAME, A.AGENT\_NAME

FROM CUSTOMER\_TABLE C

JOIN AGENTS A USING (AGENT\_CODE);

**Explanation:** This query joins CUSTOMER\_TABLE and AGENTS using the AGENT\_CODE column, simplifying the syntax by avoiding explicit ON conditions.

**Using Clause**

**Query 1:**

SELECT C.CUST\_NAME, A.AGENT\_NAME

FROM CUSTOMER\_TABLE C

JOIN AGENTS A USING (AGENT\_CODE);

**Q1:** What does this query return? **A1:** Customer names along with their corresponding agent names using AGENT\_CODE.

**Query 2:**

SELECT ORD\_NUM, CUST\_NAME

FROM ORDERS JOIN CUSTOMER\_TABLE USING (CUST\_CODE);

**Q2:** What is the result of this query? **A2:** It returns order numbers and customer names for each order.

**Query 3:**

SELECT CUST\_NAME, COUNTRY

FROM CUSTOMER\_TABLE JOIN AGENTS USING (COUNTRY);

**Q3:** What does this query filter by? **A3:** It filters by the same COUNTRY in both tables.

**Query 4:**

SELECT CUST\_NAME, COMMISSION

FROM CUSTOMER\_TABLE JOIN AGENTS USING (AGENT\_CODE)

WHERE COMMISSION > 0.10;

**Q4:** What is the condition applied in this query? **A4:** It retrieves customers where the agent’s commission is greater than 10%.

**Query 5:**

SELECT CUST\_NAME, WORKING\_AREA

FROM CUSTOMER\_TABLE JOIN AGENTS USING (WORKING\_AREA);

**Q5:** What does this query return? **A5:** Customer names and working areas where both tables share the same working area.

**Outer Joins**

**Left Outer Join Query:**

SELECT C.CUST\_NAME, A.AGENT\_NAME

FROM CUSTOMER\_TABLE C

LEFT OUTER JOIN AGENTS A

ON C.AGENT\_CODE = A.AGENT\_CODE;

**Right Outer Join Query:**

SELECT C.CUST\_NAME, A.AGENT\_NAME

FROM CUSTOMER\_TABLE C

RIGHT OUTER JOIN AGENTS A

ON C.AGENT\_CODE = A.AGENT\_CODE;

**Full Outer Join Query:**

SELECT C.CUST\_NAME, A.AGENT\_NAME

FROM CUSTOMER\_TABLE C

FULL OUTER JOIN AGENTS A

ON C.AGENT\_CODE = A.AGENT\_CODE;

**Explanation:** These queries include all rows from one or both tables, even if there is no match.

**Self Join**

**Query:**

SELECT A.CUST\_NAME AS Customer1, B.CUST\_NAME AS Customer2

FROM CUSTOMER\_TABLE A, CUSTOMER\_TABLE B

WHERE A.WORKING\_AREA = B.WORKING\_AREA AND A.CUST\_CODE != B.CUST\_CODE;

**Explanation:** This self-join finds pairs of customers working in the same area.

**Outer Joins**

**Left Outer Join Query 1:**

SELECT C.CUST\_NAME, A.AGENT\_NAME

FROM CUSTOMER\_TABLE C

LEFT OUTER JOIN AGENTS A

ON C.AGENT\_CODE = A.AGENT\_CODE

WHERE A.AGENT\_NAME IS NULL;

**Q1:** What does WHERE A.AGENT\_NAME IS NULL do? **A1:** It filters customers who do not have matching agents.

**Right Outer Join Query 1:**

SELECT C.CUST\_NAME, A.AGENT\_NAME

FROM CUSTOMER\_TABLE C

RIGHT OUTER JOIN AGENTS A

ON C.AGENT\_CODE = A.AGENT\_CODE

WHERE C.CUST\_NAME IS NULL;

**Q2:** What is the purpose of this query? **A2:** It finds agents without any customers assigned.

**Full Outer Join Query 1:**

SELECT C.CUST\_NAME, A.AGENT\_NAME

FROM CUSTOMER\_TABLE C

FULL OUTER JOIN AGENTS A

ON C.AGENT\_CODE = A.AGENT\_CODE;

**Q3:** What will this query return? **A3:** It returns all customers and agents, including those without a match in the other table.

**Left Outer Join Query 2:**

SELECT C.CUST\_NAME, O.ORD\_NUM

FROM CUSTOMER\_TABLE C

LEFT JOIN ORDERS O

ON C.CUST\_CODE = O.CUST\_CODE;

**Q4:** What does this query retrieve? **A4:** All customers, including those without orders.

**Full Outer Join Query 2:**

SELECT CUST\_NAME, ORD\_NUM

FROM CUSTOMER\_TABLE C

FULL JOIN ORDERS O

ON C.CUST\_CODE = O.CUST\_CODE;

**Q5:** What is the output of this query? **A5:** All customers and orders, showing nulls where

there is no match.

**Self Join**

**Query 1:**

SELECT A.CUST\_NAME AS Customer1, B.CUST\_NAME AS Customer2

FROM CUSTOMER\_TABLE A, CUSTOMER\_TABLE B

WHERE A.AGENT\_CODE = B.AGENT\_CODE AND A.CUST\_CODE < B.CUST\_CODE;

**Q1:** What does this query return? **A1:** Pairs of customers with the same agent but different customer codes.

**Query 2:**

SELECT A.CUST\_NAME AS Customer1, B.CUST\_NAME AS Customer2

FROM CUSTOMER\_TABLE A, CUSTOMER\_TABLE B

WHERE A.COUNTRY = B.COUNTRY AND A.CUST\_CODE != B.CUST\_CODE;

**Q2:** What is the purpose of this query? **A2:** It finds pairs of customers from the same country with different customer codes.

**Query 3:**

SELECT A.CUST\_NAME, B.CUST\_NAME

FROM CUSTOMER\_TABLE A, CUSTOMER\_TABLE B

WHERE A.WORKING\_AREA = B.WORKING\_AREA AND A.CUST\_NAME < B.CUST\_NAME;

**Q3:** What does this query return? **A3:** Customer pairs working in the same area, ordered by name.

**Query 4:**

SELECT A.CUST\_NAME, B.CUST\_NAME

FROM CUSTOMER\_TABLE A

JOIN CUSTOMER\_TABLE B ON A.AGENT\_CODE = B.AGENT\_CODE

WHERE A.CUST\_CODE != B.CUST\_CODE;

**Q4:** What does this query find? **A4:** Customers handled by the same agent but with different customer codes.

**Query 5:**

SELECT A.CUST\_NAME AS Customer1, B.CUST\_NAME AS Customer2

FROM CUSTOMER\_TABLE A

JOIN CUSTOMER\_TABLE B ON A.WORKING\_AREA = B.WORKING\_AREA

WHERE A.CUST\_CODE < B.CUST\_CODE;

**Q5:** What is this query’s output? **A5:** Customer pairs from the same working area, with Customer1 having a lower customer code.

**Set Operations**

**Union Query:**

SELECT CUST\_NAME, WORKING\_AREA FROM CUSTOMER\_TABLE

UNION

SELECT AGENT\_NAME, WORKING\_AREA FROM AGENTS;

**Union All Query:**

SELECT CUST\_NAME, WORKING\_AREA FROM CUSTOMER\_TABLE

UNION ALL

SELECT AGENT\_NAME, WORKING\_AREA FROM AGENTS;

**Intersect Query:**

SELECT CUST\_NAME, WORKING\_AREA FROM CUSTOMER\_TABLE

INTERSECT

SELECT AGENT\_NAME, WORKING\_AREA FROM AGENTS;

**Minus Query:**

SELECT CUST\_NAME, WORKING\_AREA FROM CUSTOMER\_TABLE

MINUS

SELECT AGENT\_NAME, WORKING\_AREA FROM AGENTS;

**Set Operations**

**Union Query 1:**

SELECT CUST\_NAME FROM CUSTOMER\_TABLE

UNION

SELECT AGENT\_NAME FROM AGENTS;

**Q1:** What is returned by this query? **A1:** A combined list of unique customer and agent names.

**Union All Query 1:**

SELECT WORKING\_AREA FROM CUSTOMER\_TABLE

UNION ALL

SELECT WORKING\_AREA FROM AGENTS;

**Q2:** What is the difference between UNION and UNION ALL? **A2:** UNION removes duplicates, while UNION ALL includes all rows.

**Intersect Query 1:**

SELECT WORKING\_AREA FROM CUSTOMER\_TABLE

INTERSECT

SELECT WORKING\_AREA FROM AGENTS;

**Q3:** What does INTERSECT return? **A3:** Common working areas present in both tables.

**Minus Query 1:**

SELECT CUST\_NAME FROM CUSTOMER\_TABLE

MINUS

SELECT AGENT\_NAME FROM AGENTS;

**Q4:** What does this MINUS query do? **A4:** It returns customer names not listed as agent names.

**Union Query 2:**

SELECT CUST\_NAME, CUST\_COUNTRY FROM CUSTOMER\_TABLE

UNION

SELECT AGENT\_NAME, COUNTRY FROM AGENTS;

**Q5:** What does this query result in? **A5:** A unique list of names and countries from both customers and agents.

These queries explore various aspects of SQL joins and set operations, providing comprehensive practice and understanding.