

EQUI JOIN & NATURAL JOIN — Challenging Output Questions

Class XII Computer Science (083) | 13 questions with worked answers and the reasoning behind each

The three tables

TABLE : BOOK

CODE	BNAME	TYPE	PRICE
F101	The priest	Fiction	350
L102	Easy Python	Programming	520
C101	Juman Ji	Thriller	280
F102	Untold Story	Fiction	410
C102	War Stories	Comic	150

TABLE : MEMBER

MNO	MNAME	CODE	ISSUEDATE
M101	SNEH SINHA	L102	2022-10-13
M102	SARA KHAN	C101	2022-06-12
M103	SARTHAK	F102	2021-02-23
M104	RITU ARORA	L102	2022-01-05
M105	AMIT VERMA	NULL	2022-11-20

TABLE : PUBLISHER

CODE	PNAME	TYPE
F101	Rupa	Fiction
L102	BPB	Programming
C101	Diamond	Comic
C102	Rupa	Comic

- BOOK and MEMBER share ONE column : CODE.
- BOOK and PUBLISHER share TWO columns : CODE and TYPE. Watch what that does to a natural join.
- Three rows are designed to be dropped: C102 and F101 have no member, AMIT VERMA has NULL in CODE, and L102 is issued to two members.
- SQL does not promise any particular order of rows unless ORDER BY is written. The answers below follow the order of the first table.

PART A — EQUI JOIN

Q1. SELECT B.BNAME, M.MNAME FROM BOOK B, MEMBER M

WHERE B.CODE = M.CODE AND YEAR(ISSUEDATE) = 2022;

TABLE: BOOK

CODE	BNAME	TYPE	PRICE
F101	The priest	Fiction	350
L102	Easy Python	Programming	520
C101	Juman Ji	Thriller	280
F102	Untold Story	Fiction	410
C102	War Stories	Comic	150

TABLE: MEMBER

MNO	MNAME	CODE	ISSUEDATE
M101	SNEH SINHA	L102	2022-10-13
M102	SARA KHAN	C101	2022-06-12
M103	SARTHAK	F102	2021-02-23
M104	RITU ARORA	L102	2022-01-05
M105	AMIT VERMA	NULL	2022-11-20

OUTPUT — 3 rows

BNAME	MNAME
Easy Python	SNEH SINHA
Juman Ji	SARA KHAN
Easy Python	RITU ARORA

- SARTHAK is removed by YEAR(ISSUEDATE) = 2021, AMIT VERMA by the join itself. Easy Python appears twice because two members hold it.

Q2. SELECT B.TYPE, COUNT(*) FROM BOOK B, MEMBER M

WHERE B.CODE = M.CODE GROUP BY B.TYPE;

TABLE: BOOK

CODE	BNAME	TYPE	PRICE
F101	The priest	Fiction	350
L102	Easy Python	Programming	520
C101	Juman Ji	Thriller	280
F102	Untold Story	Fiction	410
C102	War Stories	Comic	150

TABLE: MEMBER

MNO	MNAME	CODE	ISSUEDATE
M101	SNEH SINHA	L102	2022-10-13
M102	SARA KHAN	C101	2022-06-12
M103	SARTHAK	F102	2021-02-23
M104	RITU ARORA	L102	2022-01-05
M105	AMIT VERMA	NULL	2022-11-20

OUTPUT — 3 rows

TYPE	COUNT(*)
Fiction	1
Programming	2
Thriller	1

- Comic never appears. Grouping happens after the join, and War Stories was already dropped because no member has issued it. Fiction shows 1, not 2, because The priest is also unissued.

Q3. Give the degree and cardinality of :

```
SELECT * FROM BOOK, MEMBER WHERE BOOK.CODE = MEMBER.CODE;
```

TABLE: BOOK

CODE	BNAME	TYPE	PRICE
F101	The priest	Fiction	350
L102	Easy Python	Programming	520
C101	Juman Ji	Thriller	280
F102	Untold Story	Fiction	410
C102	War Stories	Comic	150

TABLE: MEMBER

MNO	MNAME	CODE	ISSUEDATE
M101	SNEH SINHA	L102	2022-10-13
M102	SARA KHAN	C101	2022-06-12
M103	SARTHAK	F102	2021-02-23
M104	RITU ARORA	L102	2022-01-05
M105	AMIT VERMA	NULL	2022-11-20

OUTPUT

Degree	Cardinality
8	4

- Degree = 4 + 4 = 8, because an equi join keeps BOTH copies of CODE. Cardinality = 4, the number of matching pairs.

Q4. SELECT B.BNAME, P.PNAME FROM BOOK B, PUBLISHER P WHERE B.CODE = P.CODE;

TABLE: BOOK

CODE	BNAME	TYPE	PRICE
F101	The priest	Fiction	350
L102	Easy Python	Programming	520
C101	Juman Ji	Thriller	280
F102	Untold Story	Fiction	410
C102	War Stories	Comic	150

TABLE: PUBLISHER

CODE	PNAME	TYPE
F101	Rupa	Fiction
L102	BPB	Programming
C101	Diamond	Comic
C102	Rupa	Comic

OUTPUT — 4 rows

BNAME	PNAME
The priest	Rupa
Easy Python	BPB
Juman Ji	Diamond
War Stories	Rupa

- Juman Ji IS listed here. Keep this answer in mind — Q9 asks the same thing with a natural join and the result changes.

Q5. SELECT M.MNAME, B.PRICE FROM BOOK B, MEMBER M

```
WHERE B.CODE = M.CODE AND B.PRICE > 300 ORDER BY M.MNAME;
```

TABLE: BOOK

CODE	BNAME	TYPE	PRICE
F101	The priest	Fiction	350
L102	Easy Python	Programming	520
C101	Juman Ji	Thriller	280
F102	Untold Story	Fiction	410
C102	War Stories	Comic	150

TABLE: MEMBER

MNO	MNAME	CODE	ISSUEDATE
M101	SNEH SINHA	L102	2022-10-13
M102	SARA KHAN	C101	2022-06-12
M103	SARTHAK	F102	2021-02-23
M104	RITU ARORA	L102	2022-01-05
M105	AMIT VERMA	NULL	2022-11-20

OUTPUT — 3 rows

MNAME	PRICE
RITU ARORA	520
SARTHAK	410
SNEH SINHA	520

- SARA KHAN goes out because Juman Ji costs 280. ORDER BY sorts the names alphabetically, not the prices.

Q6. SELECT COUNT(*), COUNT(DISTINCT B.CODE) FROM BOOK B, MEMBER M

```
WHERE B.CODE = M.CODE;
```

TABLE: BOOK

CODE	BNAME	TYPE	PRICE
F101	The priest	Fiction	350
L102	Easy Python	Programming	520
C101	Juman Ji	Thriller	280
F102	Untold Story	Fiction	410
C102	War Stories	Comic	150

TABLE: MEMBER

MNO	MNAME	CODE	ISSUEDATE
M101	SNEH SINHA	L102	2022-10-13
M102	SARA KHAN	C101	2022-06-12
M103	SARTHAK	F102	2021-02-23
M104	RITU ARORA	L102	2022-01-05
M105	AMIT VERMA	NULL	2022-11-20

OUTPUT — 1 row

COUNT(*)	COUNT(DISTINCT B.CODE)
4	3

- Four rows survive the join, but only three different books are involved — L102 is counted once by DISTINCT and twice by COUNT(*)

Q7. SELECT B.BNAME, M.MNAME FROM BOOK B, MEMBER M WHERE B.TYPE = 'Fiction';

OUTPUT — 10 rows

BNAME	MNAME
The priest	SNEH SINHA
The priest	SARA KHAN
The priest	SARTHAK

TABLE: BOOK

CODE	BNAME	TYPE	PRICE
F101	The priest	Fiction	350
L102	Easy Python	Programming	520
C101	Juman Ji	Thriller	280
F102	Untold Story	Fiction	410
C102	War Stories	Comic	150

TABLE: MEMBER

MNO	MNAME	CODE	ISSUEDATE
M101	SNEH SINHA	L102	2022-10-13
M102	SARA KHAN	C101	2022-06-12
M103	SARTHAK	F102	2021-02-23
M104	RITU ARORA	L102	2022-01-05
M105	AMIT VERMA	NULL	2022-11-20

BNAME	MNAME
The priest	RITU ARORA
The priest	AMIT VERMA
Untold Story	SNEH SINHA
Untold Story	SARA KHAN
Untold Story	SARTHAK
Untold Story	RITU ARORA
Untold Story	AMIT VERMA

- There is no joining condition here, so this is a CARTESIAN PRODUCT of the 2 fiction books with all 5 members: $2 \times 5 = 10$. Even AMIT VERMA appears. A WHERE clause is a join only when it compares the common columns of the two tables.

PART B — NATURAL JOIN

Q8. SELECT * FROM BOOK NATURAL JOIN MEMBER;

OUTPUT — 4 rows, 7 columns

CODE	BNAME	TYPE	PRICE	MNO	MNAME	ISSUEDATE
L102	Easy Python	Programming	520	M101	SNEH SINHA	2022-10-13
L102	Easy Python	Programming	520	M104	RITU ARORA	2022-01-05
C101	Juman Ji	Thriller	280	M102	SARA KHAN	2022-06-12
F102	Untold Story	Fiction	410	M103	SARTHAK	2021-02-23

- CODE is written ONCE and comes FIRST. Degree = $4 + 4 - 1 = 7$. Compare with Q3, where the equi join gave 8 columns for the same 4 rows.
- A book issued to two members produces two rows — a natural join does not remove duplicate rows.

Q9. SELECT MNAME FROM BOOK NATURAL JOIN MEMBER;

TABLE: BOOK

CODE	BNAME	TYPE	PRICE
F101	The priest	Fiction	350
L102	Easy Python	Programming	520
C101	Juman Ji	Thriller	280
F102	Untold Story	Fiction	410
C102	War Stories	Comic	150

TABLE: MEMBER

MNO	MNAME	CODE	ISSUEDATE
M101	SNEH SINHA	L102	2022-10-13
M102	SARA KHAN	C101	2022-06-12
M103	SARTHAK	F102	2021-02-23
M104	RITU ARORA	L102	2022-01-05
M105	AMIT VERMA	NULL	2022-11-20

OUTPUT — 4 rows

MNAME
SNEH SINHA
RITU ARORA
SARA KHAN
SARTHAK

- AMIT VERMA is missing. His CODE is NULL, and NULL is never equal to anything — not even to another NULL. So his row can never match and is dropped by every join.

Q10. SELECT * FROM BOOK NATURAL JOIN PUBLISHER;

OUTPUT — 3 rows, 5 columns

CODE	TYPE	BNAME	PRICE	PNAME
F101	Fiction	The priest	350	Rupa
L102	Programming	Easy Python	520	BPB
C102	Comic	War Stories	150	Rupa

- This is the hardest idea in the topic. BOOK and PUBLISHER have TWO columns with the same name, so a natural join matches on CODE AND TYPE together.
- Both common columns are placed first, in the order they appear in BOOK — CODE, then TYPE. Degree = $4 + 3 - 2 = 5$.

Q11. C101 exists in both BOOK and PUBLISHER. Why is it missing from the answer of Q10, when Q4 did list Juman Ji?

ANSWER

TABLE: BOOK

CODE	BNAME	TYPE	PRICE
F101	The priest	Fiction	350
L102	Easy Python	Programming	520
C101	Juman Ji	Thriller	280
F102	Untold Story	Fiction	410
C102	War Stories	Comic	150

TABLE: PUBLISHER

CODE	PNAME	TYPE
F101	Rupa	Fiction
L102	BPB	Programming
C101	Diamond	Comic
C102	Rupa	Comic

Table	CODE	TYPE
BOOK	C101	Thriller
PUBLISHER	C101	Comic

- The natural join needs BOTH common columns to be equal. C101 matches on CODE but Thriller is not Comic, so the row is rejected.
- Q4 was an equi join written on CODE alone, so the mismatch in TYPE was never checked and Juman Ji appeared. Equi join = the condition you write. Natural join = every common column, whether you wanted it or not.

Q12. SELECT BNAME, MNAME, PRICE FROM BOOK NATURAL JOIN MEMBER

WHERE TYPE = 'Fiction' OR PRICE < 300;

OUTPUT — 2 rows

TABLE: BOOK

CODE	BNAME	TYPE	PRICE
F101	The priest	Fiction	350
L102	Easy Python	Programming	520
C101	Juman Ji	Thriller	280
F102	Untold Story	Fiction	410
C102	War Stories	Comic	150

TABLE: MEMBER

MNO	MNAME	CODE	ISSUEDATE
M101	SNEH SINHA	L102	2022-10-13
M102	SARA KHAN	C101	2022-06-12
M103	SARTHAK	F102	2021-02-23
M104	RITU ARORA	L102	2022-01-05
M105	AMIT VERMA	NULL	2022-11-20

BNAME	MNAME	PRICE
Juman Ji	SARA KHAN	280
Untold Story	SARTHAK	410

- War Stories costs 150 but has no member, so it is already gone before WHERE runs. TYPE and PRICE need no table name here because each exists in only one table.

Q13. SELECT * FROM BOOK NATURAL JOIN MEMBER

TABLE : BOOK

CODE	BNAME	TYPE
F101	The priest	Fiction
L102	Easy Python	Programming
C101	Juman Ji	Thriller
F102	Untold Story	Fiction
C102	War Stories	Comic

Table: MEMBER

MNO	MNAME	CODE	ISSUEDATE
M101	SNEH SINHA	L102	2022-10-13
M103	SARTHAK	F102	2021-02-23
M102	SARA KHAN	C101	2022-06-12

OUTPUT — 3 rows, 6 columns

CODE	BNAME	TYPE	MNO	MNAME	ISSUEDATE
L102	Easy Python	Programming	M101	SNEH SINHA	2022-10-13
C101	Juman Ji	Thriller	M102	SARA KHAN	2022-06-12
F102	Untold Story	Fiction	M103	SARTHAK	2021-02-23

- F101 and C102 are dropped because no member has issued them. Degree = $3 + 4 - 1 = 6$, Cardinality = 3.

Q14. Fill in the table below for the three queries.

ANSWER

Query	Degree	Cardinality
SELECT * FROM BOOK, MEMBER;	8	25
SELECT * FROM BOOK NATURAL JOIN MEMBER;	7	4
SELECT * FROM BOOK NATURAL JOIN PUBLISHER;	5	3

- Cartesian product : $5 \times 5 = 25$ rows and $4 + 4 = 8$ columns. The NULL row of AMIT VERMA is counted here — a cartesian product matches nothing, so nothing is rejected.

The six traps in this set

Trap	Where it appears	What to remember
WHERE without the common columns	Q7	it is a filter, not a join — the answer is a cartesian product
NULL in the joining column	Q9	NULL matches nothing, the row is always dropped
Two common columns	Q10, Q11	a natural join matches ALL of them, not just the key
Duplicate matches	Q6, Q8	one row on the left can produce several rows in the result
Grouping after a join	Q2	unmatched rows are gone before GROUP BY runs
Column count in SELECT *	Q3, Q8, Q13	equi join keeps both copies, natural join keeps one

