**Relational Algebra Operations in SQL (Oracle) - Answer Key with Notations**

**Scenario:** You are managing a university database with two tables:

* STUDENT(SID, Name, DeptID)
* DEPARTMENT(DeptID, DeptName)

Some students may not be assigned to a department yet, and some departments may have no students.

**1. Table Creation and Sample Data (Run this in Oracle Live SQL)**

-- Drop tables if they already exist

DROP TABLE STUDENT;

DROP TABLE DEPARTMENT;

-- Create Department Table

CREATE TABLE DEPARTMENT (

DeptID NUMBER PRIMARY KEY,

DeptName VARCHAR2(50)

);

-- Create Student Table

CREATE TABLE STUDENT (

SID NUMBER PRIMARY KEY,

Name VARCHAR2(50),

DeptID NUMBER REFERENCES DEPARTMENT(DeptID)

);

-- Insert sample data

INSERT INTO DEPARTMENT VALUES (10, 'Computer Science');

INSERT INTO DEPARTMENT VALUES (20, 'Mathematics');

INSERT INTO DEPARTMENT VALUES (30, 'Physics');

INSERT INTO DEPARTMENT VALUES (40, 'Chemistry');

INSERT INTO STUDENT VALUES (1, 'Alice', 10);

INSERT INTO STUDENT VALUES (2, 'Bob', 20);

INSERT INTO STUDENT VALUES (3, 'Charlie', NULL);

INSERT INTO STUDENT VALUES (4, 'David', 10);

INSERT INTO STUDENT VALUES (5, 'Eva', 30);

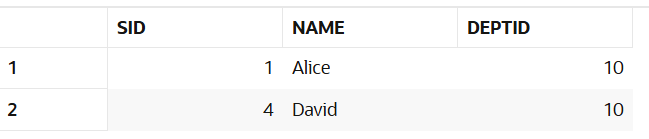
**2. Relational Algebra Operations - SQL Answers with Notations**

A. **Selection** (Select all students from the Computer Science department)

* **Relational Algebra**:



SELECT \* FROM STUDENT WHERE DeptID = 10;

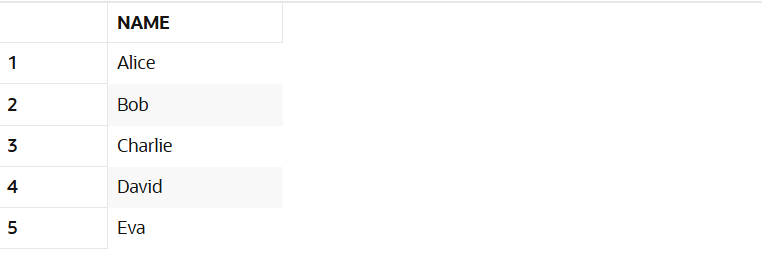


B. **Projection** (List only the names of students)



* **Relational Algebra**:

SELECT Name FROM STUDENT;



C. **Cartesian Product** (Show all combinations of students and departments)



* **Relational Algebra**:

SELECT \* FROM STUDENT, DEPARTMENT;

**WARNING AS IT IS TOO BIG**

D. **Left Outer Join** (Show all students and their departments, even those without departments)



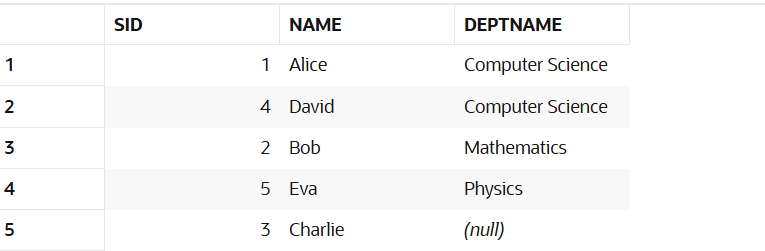
* **Relational Algebra**:

SELECT S.SID, S.Name, D.DeptName

FROM STUDENT S

LEFT OUTER JOIN DEPARTMENT D

ON S.DeptID = D.DeptID;



E. **Right Outer Join** (Show all departments and their students, even those without students)



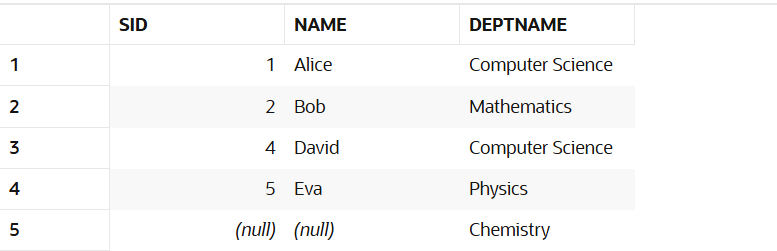
* **Relational Algebra**:

SELECT S.SID, S.Name, D.DeptName

FROM STUDENT S

RIGHT OUTER JOIN DEPARTMENT D

ON S.DeptID = D.DeptID;



F. **Full Outer Join** (Show all students and departments with possible matches)



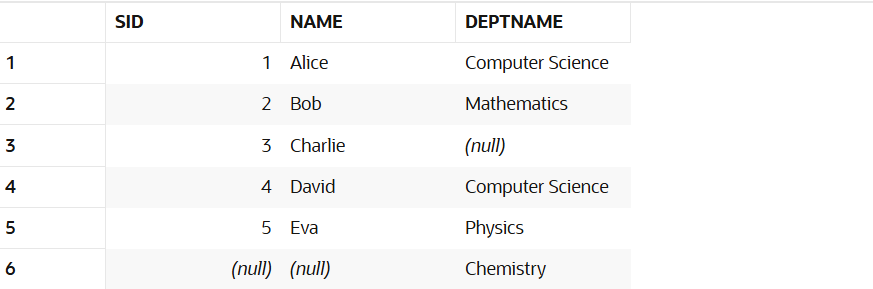
* **Relational Algebra**:

SELECT S.SID, S.Name, D.DeptName

FROM STUDENT S

FULL OUTER JOIN DEPARTMENT D

ON S.DeptID = D.DeptID;



-- Create Product Table

DROP TABLE PRODUCT;

DROP TABLE PRODUCT\_DETAIL;

CREATE TABLE PRODUCT (

    ProductID NUMBER PRIMARY KEY,

    ItemName VARCHAR2(100),

    Category VARCHAR2(50)

);

-- Create Product Detail Table

CREATE TABLE PRODUCT\_DETAIL (

    ProductID NUMBER PRIMARY KEY,

    Brand VARCHAR2(100),

    Weight VARCHAR2(20),

    ExpiryDate DATE,

    StockQuantity NUMBER,

    Price NUMBER(6,2),

    FOREIGN KEY (ProductID) REFERENCES PRODUCT(ProductID)

);

INSERT INTO PRODUCT VALUES (1, 'Milk', 'Dairy');

INSERT INTO PRODUCT VALUES (2, 'Apple', 'Fruit');

INSERT INTO PRODUCT VALUES (3, 'Rice', 'Grain');

INSERT INTO PRODUCT VALUES (4, 'Toothpaste', 'Personal Care');

INSERT INTO PRODUCT VALUES (5, 'Eggs', 'Poultry');

INSERT INTO PRODUCT\_DETAIL VALUES (1, 'Dutch Lady', '1L', TO\_DATE('2025-08-15', 'YYYY-MM-DD'), 100, 4.50);

INSERT INTO PRODUCT\_DETAIL VALUES (2, 'Envy', '200g', TO\_DATE('2025-06-30', 'YYYY-MM-DD'), 50, 1.20);

INSERT INTO PRODUCT\_DETAIL VALUES (3, 'Jasmine', '5kg', TO\_DATE('2026-01-01', 'YYYY-MM-DD'), 75, 28.90);

INSERT INTO PRODUCT\_DETAIL VALUES (4, 'Colgate', '150g', TO\_DATE('2026-05-10', 'YYYY-MM-DD'), 200, 6.80);

INSERT INTO PRODUCT\_DETAIL VALUES (5, 'Kampung Egg', '10 pcs', TO\_DATE('2025-07-01', 'YYYY-MM-DD'), 60, 7.50);

SELECT \* FROM Product;

A screenshot of a computer

AI-generated content may be incorrect.

SELECT \* FROM Product\_Detail;

A white background with black text

AI-generated content may be incorrect.

SELECT \* FROM PRODUCT WHERE Category = 'Fruit'; **\*SELECTION\***

A close up of a text

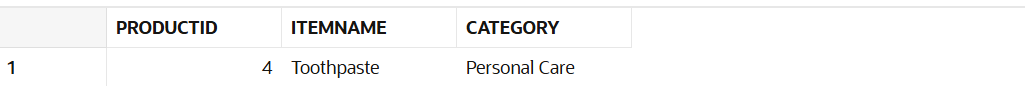
AI-generated content may be incorrect.

SELECT ITEMNAME FROM PRODUCT \***PROJECTION\***

A white rectangular object with black squares

AI-generated content may be incorrect.

SELECT \* FROM PRODUCT WHERE ProductID = 4; \***SELECTION\***



\***LEFT OUTER JOIN\***

SELECT P.ProductID, P.ItemName, D.Price

FROM Product P

LEFT OUTER JOIN Product\_Detail D

ON P.ProductID = D.ProductID;

A screenshot of a white background

AI-generated content may be incorrect.

\***RIGHT OUTER JOIN\***

SELECT P.ProductID, P.ItemName, D.Brand, D.Price

FROM Product P

RIGHT OUTER JOIN Product\_Detail D

ON P.ProductID = D.ProductID;

A screenshot of a computer

AI-generated content may be incorrect.

\***FULL OUTER JOIN\***

SELECT P.ProductID, P.ItemName, D.Brand, D.Price

FROM Product P

FULL OUTER JOIN Product\_Detail D

ON P.ProductID = D.ProductID;

A screenshot of a product list

AI-generated content may be incorrect.

In this case, both outer join will be same as everything is completed and no null value

ALTER TABLE PRODUCT RENAME COLUMN ItemName TO ProductName;

DESCRIBE Product;

A white background with black text

AI-generated content may be incorrect.