

DML : Data Manipulation Lang.

- Select: Retrieve Data
- Insert: To add new rows (Tuples)
- Update: To update the values of existing rows
- delete: To del. rows of a Table

Select [distinct] <Columns exp>

→ Columns of output

From <Tables / Subqueries>

→ The Tables to select from

Where <Constraint1
and
Constraint2
and
... >

Group By <Columns>

Having <Group By Conditions>

Order by <Columns>
[Direction]

Limit [n]

Offset [m]

<Column exp> :

- The name of a Column
and address

• e.g., The name of the Departments

Select DeptName, Address
From Department

- To remove duplicate values
of the output: DISTINCT

• e.g., The Buildings of the Departments
addresses

Select DISTINCT address
from Department

- '*' : To select ALL Columns

• e.g., To see all information of
Departments:

Select *
From Department

- Inject math. Operations

• e.g., what the Professor Salaries
would be after a 10% raise

Select name, 1.1 * Salary
From Professor

- Aggregation Operations
↳ later

- Rename the Columns (Alias): AS

• e.g., Select Distinct Address AS Building
From Department

From <Tables>
where <Conditions>

- e.g., The Professors with
Salary at least \$5000

Select *
from Professor
where Salary \geq 5000

- e.g., The Professors of
the 'CS' Department

Select Professor.name
from Department, Professor
where
Department.ID = DepID
and
Department.name = 'CS'

output

name
A
C
E

Department			Professor			
ID	name	...	ID	name	DepID	...
1	CS		1	A	1	
2	EE		2	B	2	
			3	C	1	
			4	D	2	
			5	E	1	

Cartesian Product

Dep			Professor			
ID	name	...	ID	name	DepID	...
1	CS	...	1	A	1	...
2	EE	...	1	A	1	...
1	CS	...	2	B	2	...
2	EE	...	2	B	2	...
1	CS	...	3	C	1	...
2	EE	...	3	C	1	...
...

- Observation:
The Size of Cartesian
Product is $n \times m$

Select P.name, P.Id
from Course, Course AS P
where
Course.PreReqID = P.Id
and
Course.name = 'DB'

Example From the book:

The name of Courses Student 'A' took
in Fall 2025.

Select Course.Title

from Student, Takes, Section, Course ← chain of Tables

where

Takes.ID = Student.ID
and

Takes.Sec-ID = Section.Sec-ID

and

Section.Course-ID = Course.Course-ID
and

Student.name = 'A'

and

Section.Semester = 'Fall'

and

Section.Year = 2025

← The Foreign Key Constraints

← Additional Constraints