

CUBE & ROLLUP

```
GROUP BY
CUBE (column1, column2, column3)
```

```
GROUP BY
GROUPING SETS (
  (column1, column2, column3),
  (column1, column3),
  (column2, column3),
  (column1),
  (column1),
  (column2),
  (column3),
  ()
)
```

CUBE & ROLLUP

```
GROUP BY ROLLUP (column1, column2, column3)
```

```
GROUP BY
GROUPING SETS (
  (column1, column2, column3),
  (column1, column2),
  (column1),
  ()
)
```

Hierarchy

Challenge

Write a query that calculates a booking amount rollup for the hierarchy of quarter, month, week in month and day.

quarter numeric	month numeric	week_in_month text	day date	booking_amount numeric
2	6	3	2017-06-21	441900.00
2	6	3	[null]	441900.00
2	6	4	2017-06-22	775300.00
2	6	4	2017-06-23	1822000.00

Hint

Pattern for week in month is 'w'.



Referencing employee_id

employee_id integer	name character varying (50)	manager_id integer
1	Liam Smith	[null]
2	Oliver Brown	1
3	Elijah Jones	1
4	William Miller	1
5	James Davis	2
6	Olivia Hernandez	2



Referencing employee_id

_			
employee_id integer	name character varying (50)	manager_id integer	employee_id integer
1	Liam Smith	[null]	[null]
2	Oliver Brown	1	1
3	Elijah Jones	1	1
4	William Miller	1	1
5	James Davis	2	2
6	Olivia Hernandez	2	2

Referencing employee_id

employee_id integer	name character varying (50)	manager_id integer	employee_id integer	manager character varying (50)
1	Liam Smith	[null]	[null]	[null]
2	Oliver Brown	1	1	Liam Smith
3	Elijah Jones	1	1	Liam Smith
4	William Miller	1	1	Liam Smith
5	James Davis	2	2	Oliver Brown
6	Olivia Hernandez	2	2	Oliver Brown

Standard join with itself

```
SELECT
t1.column1,
t2.column1
[,...]
FROM table1 t1
LEFT JOIN table1 t2
ON t1.column1=t2.column1
```

employee_id integer	employee character varying (50)	manager character varying (50)	manager_id integer
1	Liam Smith	[null]	[null]
2	Oliver Brown	Liam Smith	1

```
SELECT
```

t1.column1,
t2.column1

 $[\ ,...]$

FROM employee t1

LEFT JOIN employee t2

ON t1.column1=t2.column1

employee_id integer	employee character varying (50)	manager character varying (50)	manager_id integer
1	Liam Smith	[null]	[null]
2	Oliver Brown	Liam Smith	1

```
SELECT
```

t1.column1,
t2.column1

 $[\ ,...]$

FROM employee emp

LEFT JOIN employee mng

ON t1.column1=t2.column1

employee_id integer	employee character varying (50)	manager character varying (50)	manager_id integer
1	Liam Smith	[null]	[null]
2	Oliver Brown	Liam Smith	1

```
SELECT
```

t1.column1,
t2.column1

[, ...]

FROM employee emp

LEFT JOIN employee mng

ON emp.manager_id=t2.column1

employee_id integer	employee character varying (50)	manager character varying (50)	manager_id integer
1	Liam Smith	[null]	[null]
2	Oliver Brown	Liam Smith	1

```
SELECT
```

t1.column1,
t2.column1

 $[\ ,...]$

FROM employee emp

LEFT JOIN employee mng

employee_id integer	employee character varying (50)	manager character varying (50)	manager_id integer
1	Liam Smith	[null]	[null]
2	Oliver Brown	Liam Smith	1

SELECT

emp.employee_id,
t2.column1

 $[\ ,...]$

FROM employee emp

LEFT JOIN employee mng

employee_id integer	employee character varying (50)	manager character varying (50)	manager_id integer
1	Liam Smith	[null]	[null]
2	Oliver Brown	Liam Smith	1

SELECT

emp.employee_id,
emp.name AS employee

FROM employee emp

LEFT JOIN employee mng

employee_id integer	employee character varying (50)	manager character varying (50)	manager_id integer
1	Liam Smith	[null]	[null]
2	Oliver Brown	Liam Smith	1

SELECT

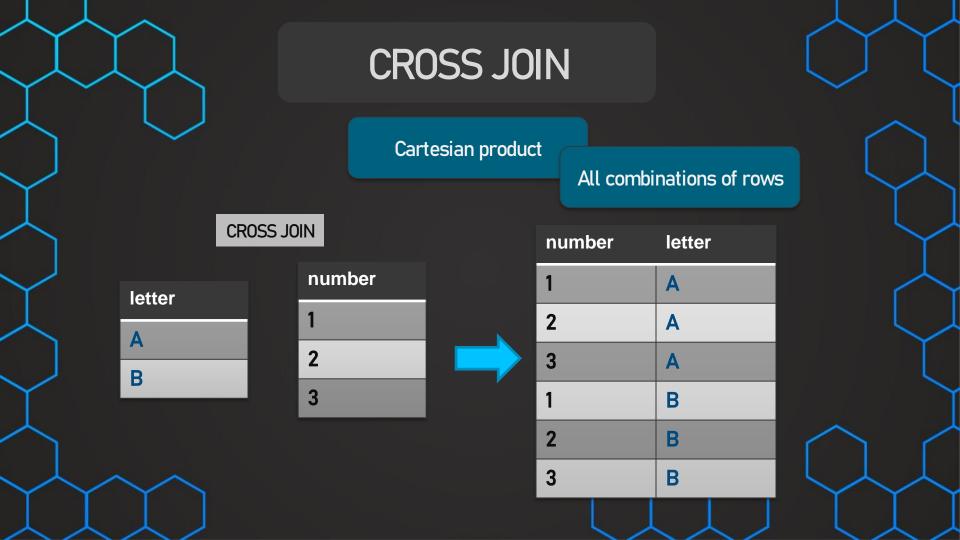
emp.employee_id,
emp.name AS employee,
mng.name AS manager
FROM employee emp
LEFT JOIN employee mng
ON emp.manager_id=mng.employee_id

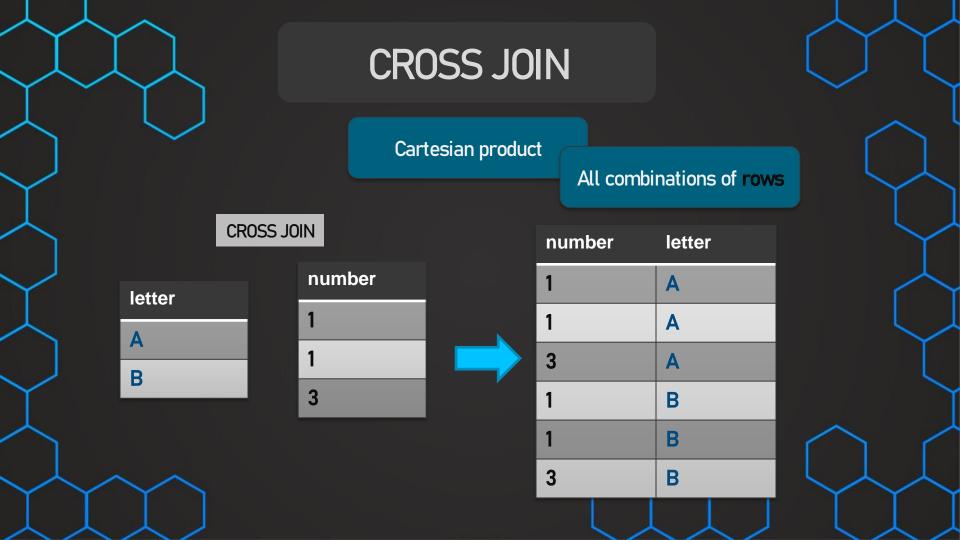
Challenge

Find all the pairs of films with the same length!

title text	title text	length smallin
MUSCLE BRIGHT	SOLDIERS EVOLUTION	185
MUSCLE BRIGHT	HOME PITY	185
MUSCLE BRIGHT	DARN FORRESTER	185
MUSCLE BRIGHT	GANGS PRIDE	185
MUSCLE BRIGHT	POND SEATTLE	185









CROSS JOIN

SELECT t1.column1, t2.column1 FROM table1 t1 CROSS JOIN table2 t2



NATURAL JOIN

Just like a normal JOIN

Automatically joins using columns with the <u>same column name</u>

SELECT

 \star

FROM payment
NATURAL LEFT JOIN customer



NATURAL JOIN

Just like a normal JOIN

Automatically joins using columns with the <u>same column name</u>

SELECT

 \star

FROM payment
NATURAL INNER JOIN customer

employee_id integer	employee character varying (50)	manager character varying (50)	manager_id integer
1	Liam Smith	[null]	[null]
2	Oliver Brown	Liam Smith	1

SELECT

emp.employee_id,
emp.name AS employee

FROM employee emp

LEFT JOIN employee mng

