Learn SQL by Example

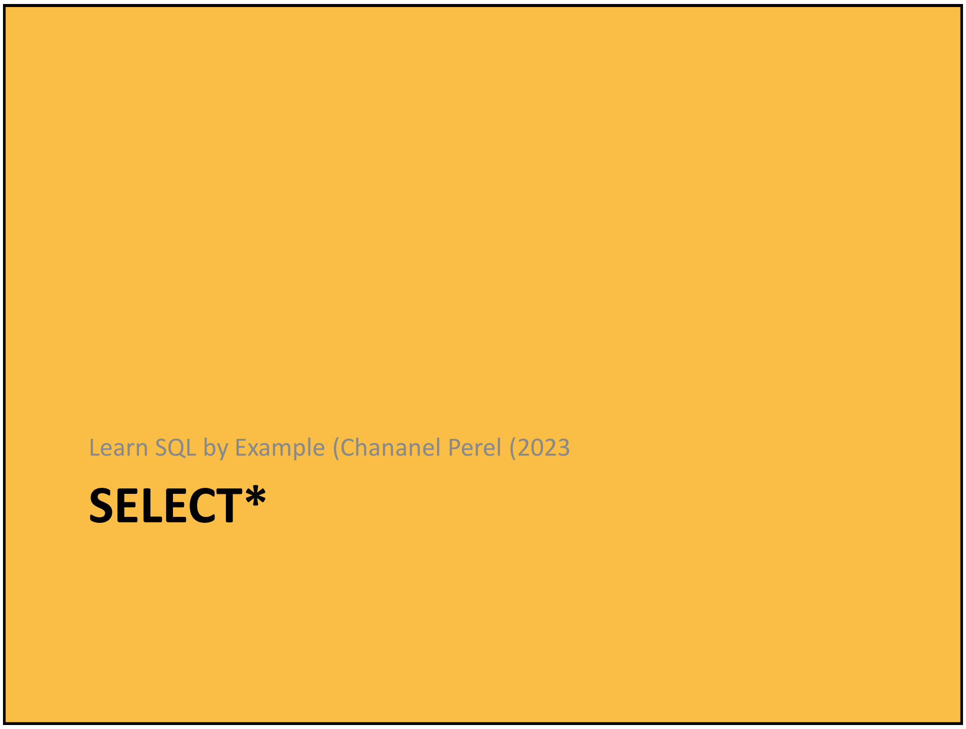
from Basic to Advanced

Chananel Perel

2023

Learn SQL by Example (Chananel Perel) 2024-04-03 12:55:52.638829

SQL READ 1

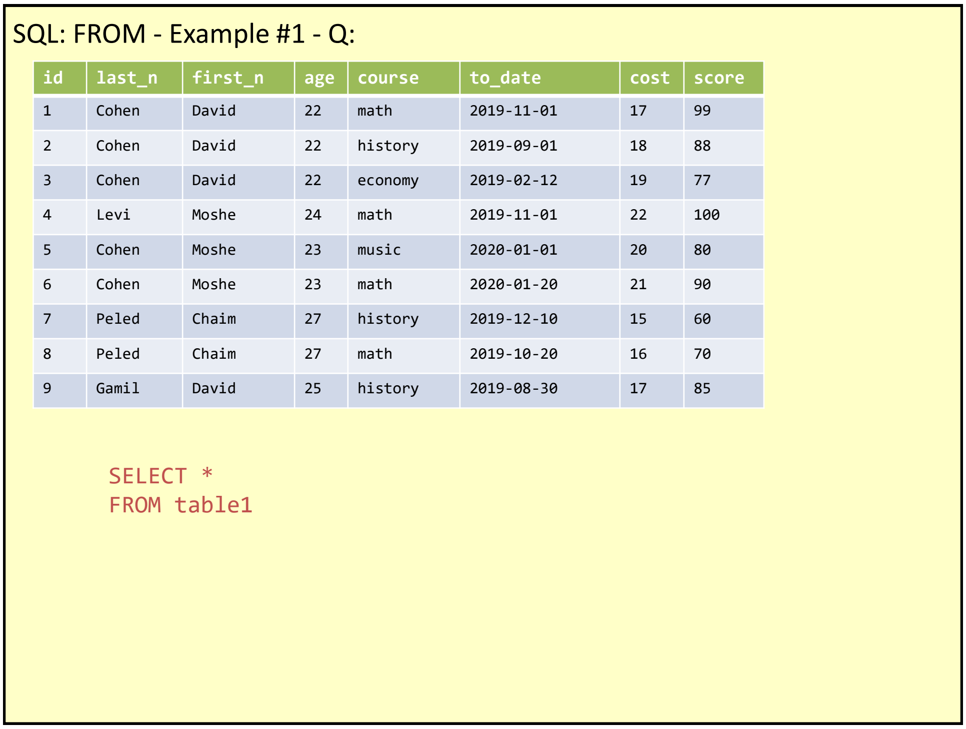


Learn SQL by Example (Chananel Perel (2023

SELECT*

Learn SQL by Example (Chananel Perel 2023)

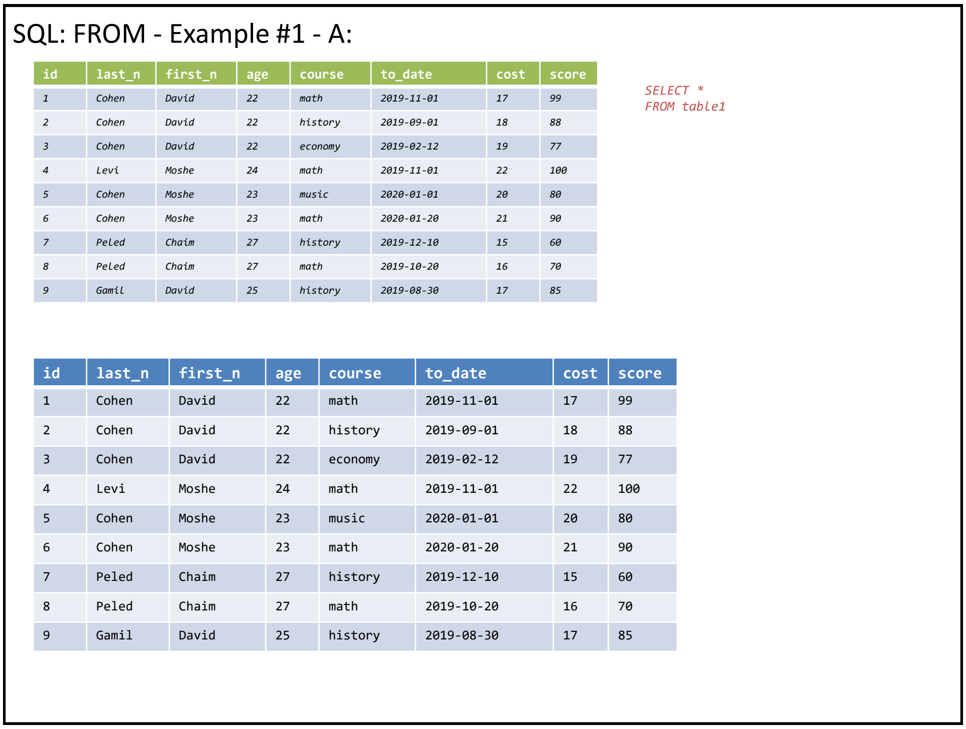
FROM



SQL: FROM - Example #1 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
```



SQL: FROM - Example #1 - A:

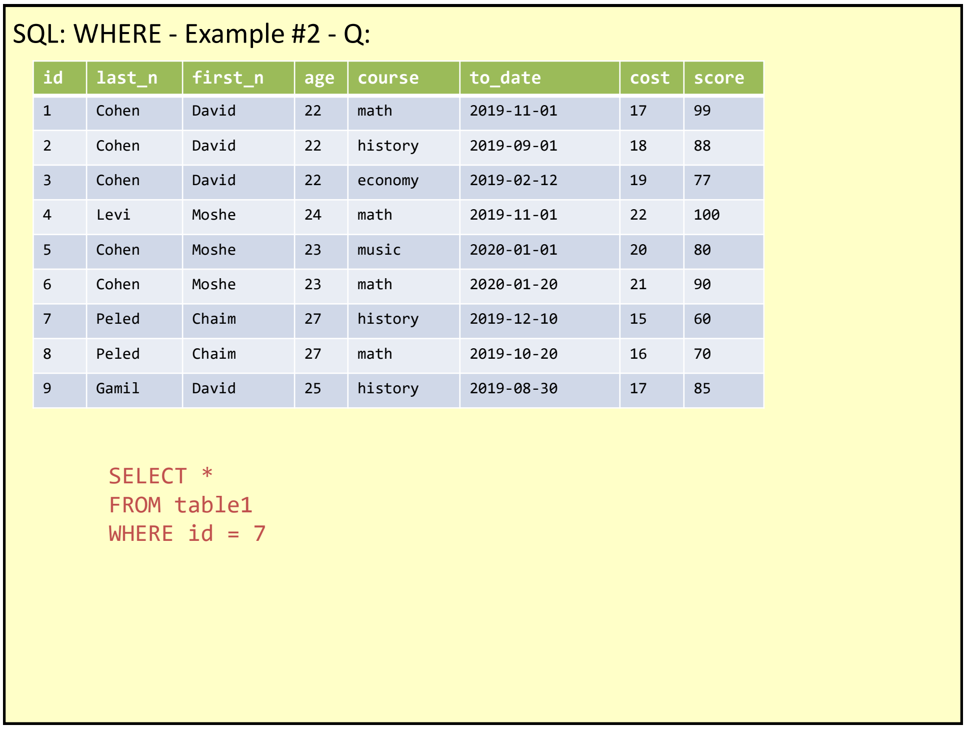
id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
```

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

Learn SQL by Example (Chananel Perel 2023)

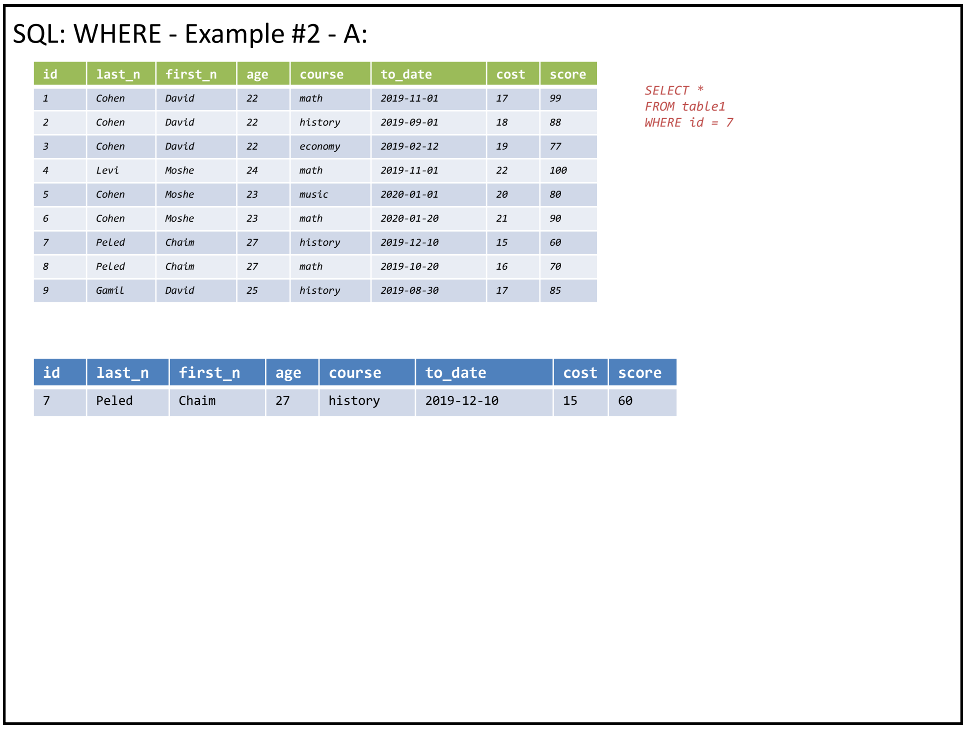
WHERE



SQL: WHERE - Example #2 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE id = 7
```

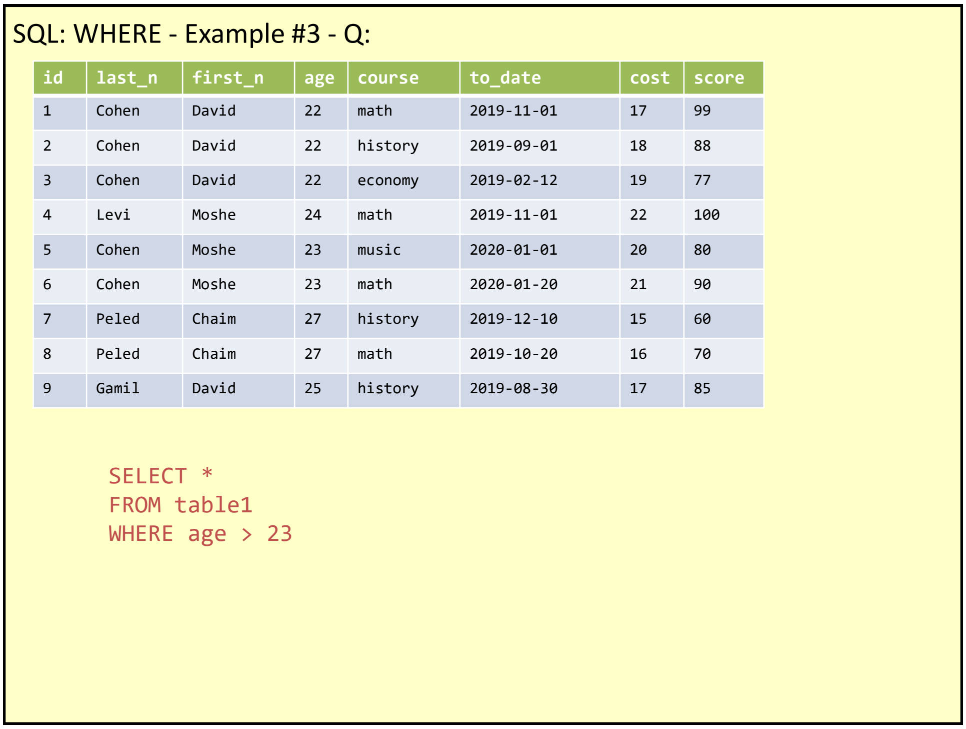


SQL: WHERE - Example #2 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE id = 7
```

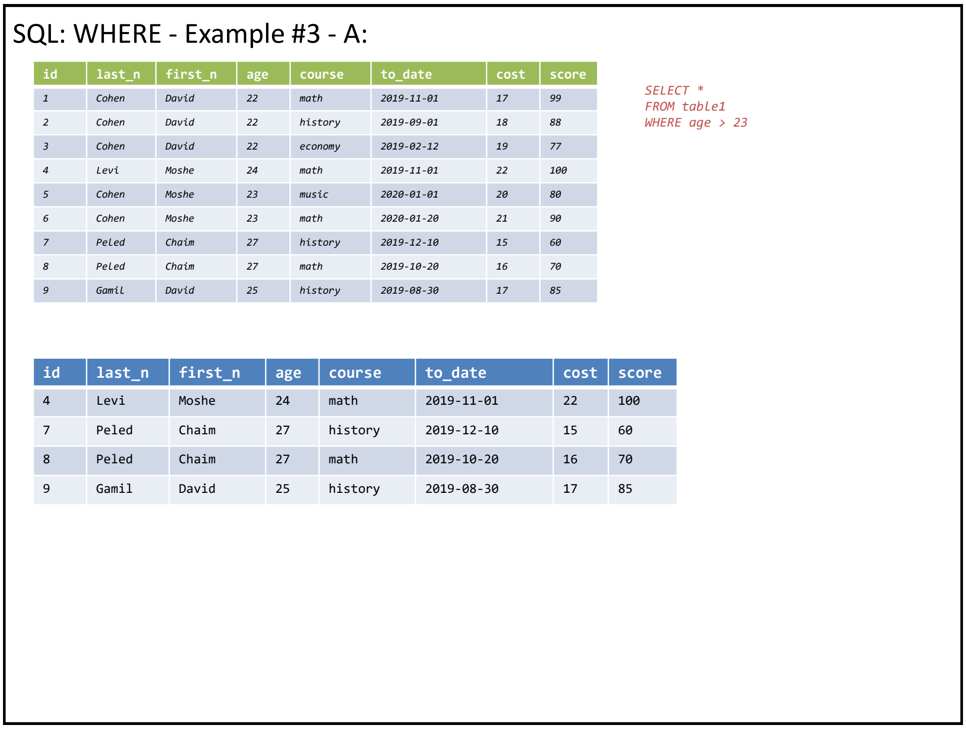
id	last_n	first_n	age	course	to_date	cost	score
7	Peled	Chaim	27	history	2019-12-10	15	60



SQL: WHERE - Example #3 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE age > 23
```

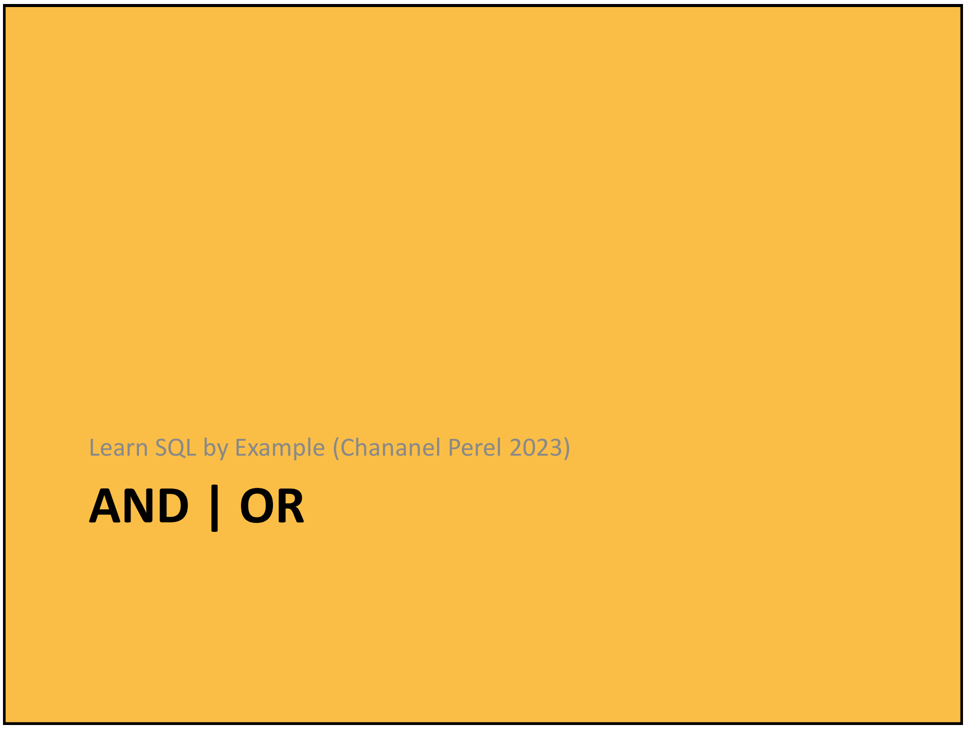


SQL: WHERE - Example #3 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

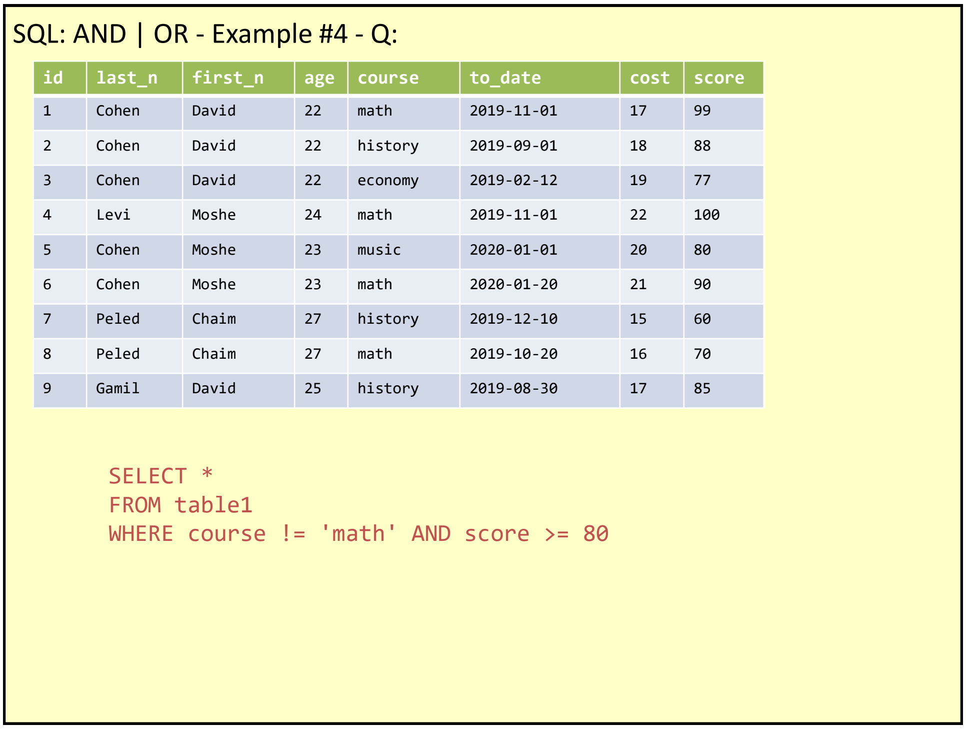
```
SELECT *
FROM table1
WHERE age > 23
```

id	last_n	first_n	age	course	to_date	cost	score
4	Levi	Moshe	24	math	2019-11-01	22	100
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85



Learn SQL by Example (Chananel Perel 2023)

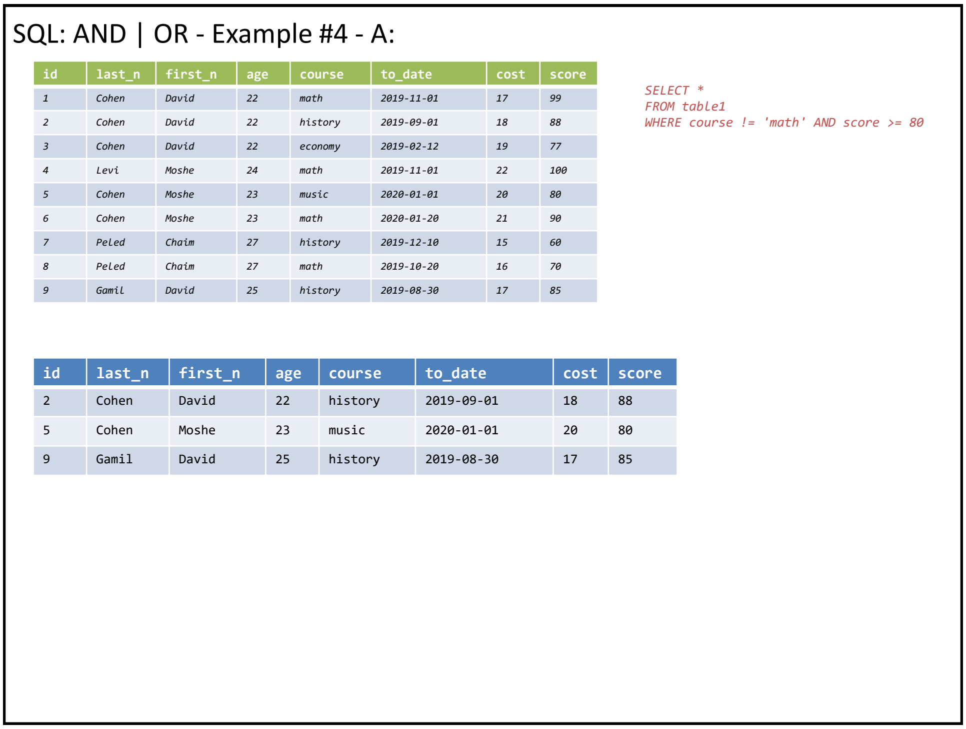
AND | OR



SQL: AND | OR - Example #4 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE course != 'math' AND score >= 80
```



SQL: AND | OR - Example #4 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE course != 'math' AND score >= 80
```

id	last_n	first_n	age	course	to_date	cost	score
2	Cohen	David	22	history	2019-09-01	18	88
5	Cohen	Moshe	23	music	2020-01-01	20	80
9	Gamil	David	25	history	2019-08-30	17	85

SQL: AND | OR - Example #5 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

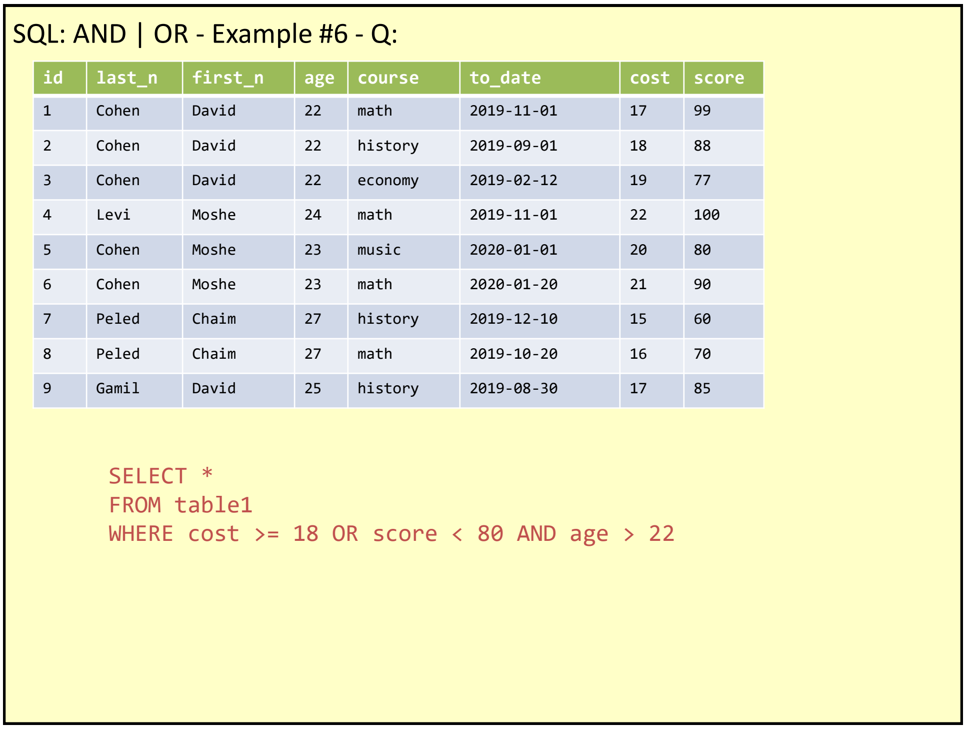
```
SELECT *
FROM table1
WHERE course != 'math' OR score >= 80
```

SQL: AND | OR - Example #5 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE course != 'math' OR score >= 80
```

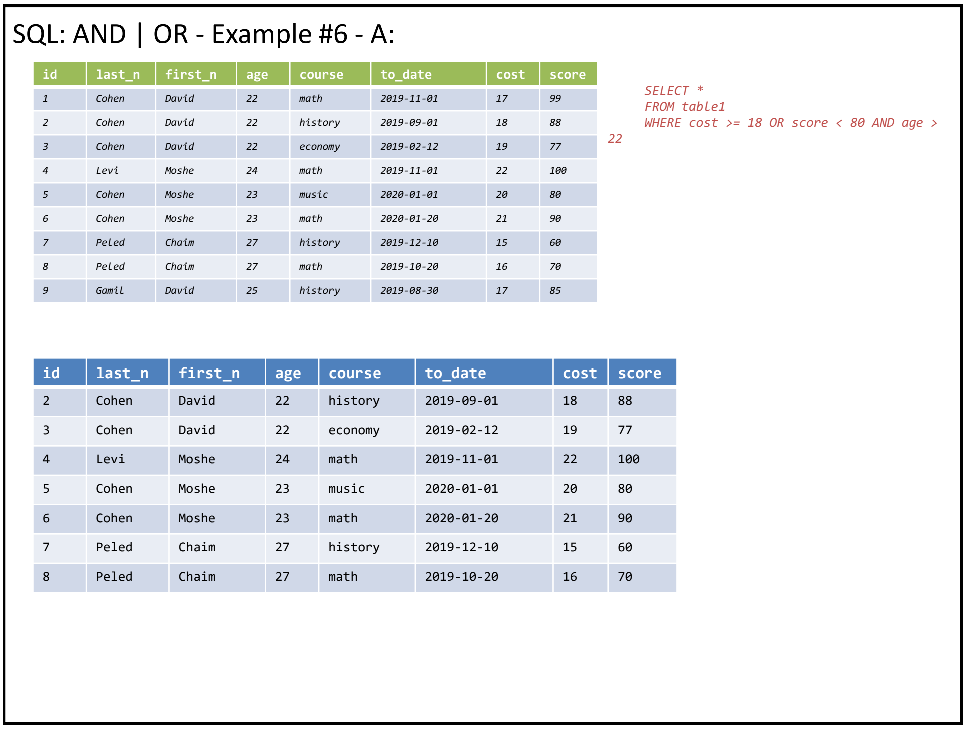
id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
9	Gamil	David	25	history	2019-08-30	17	85



SQL: AND | OR - Example #6 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE cost >= 18 OR score < 80 AND age > 22
```

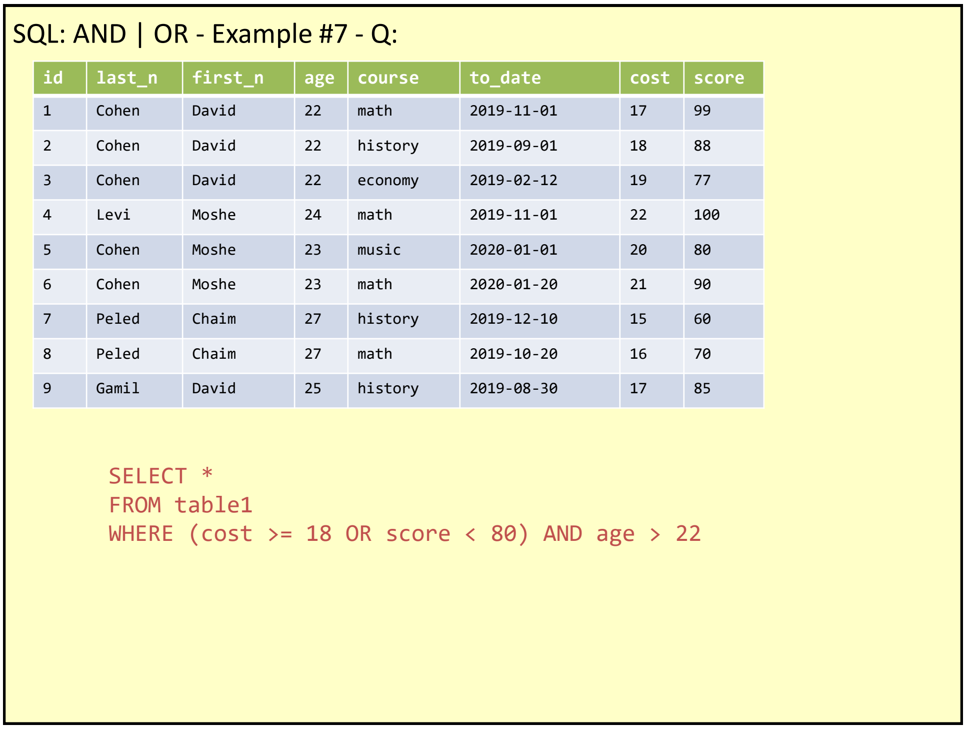


SQL: AND | OR - Example #6 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE cost >= 18 OR score < 80 AND age >
22
```

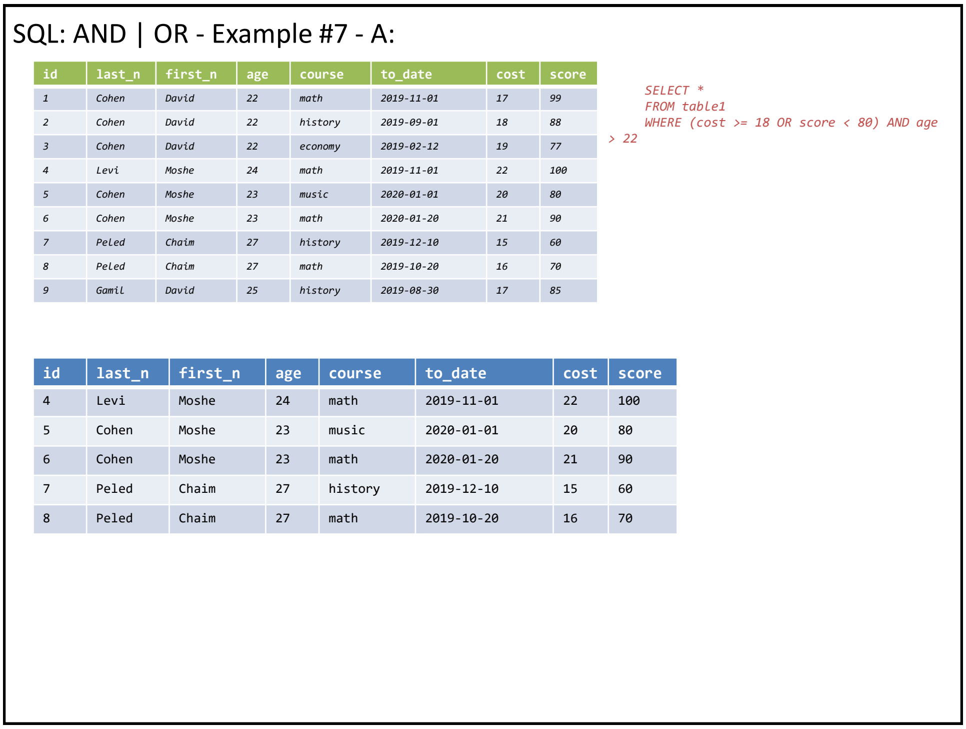
id	last_n	first_n	age	course	to_date	cost	score
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70



SQL: AND | OR - Example #7 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE (cost >= 18 OR score < 80) AND age > 22
```



SQL: AND | OR - Example #7 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
    SELECT *
    FROM table1
    WHERE (cost >= 18 OR score < 80) AND age
> 22
```

id	last_n	first_n	age	course	to_date	cost	score
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70

SQL: AND | OR - Example #8 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE cost >= 18 OR (score < 80 AND age > 22)
```

SQL: AND | OR - Example #8 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE cost >= 18 OR (score < 80 AND age
> 22)
```

id	last_n	first_n	age	course	to_date	cost	score
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70

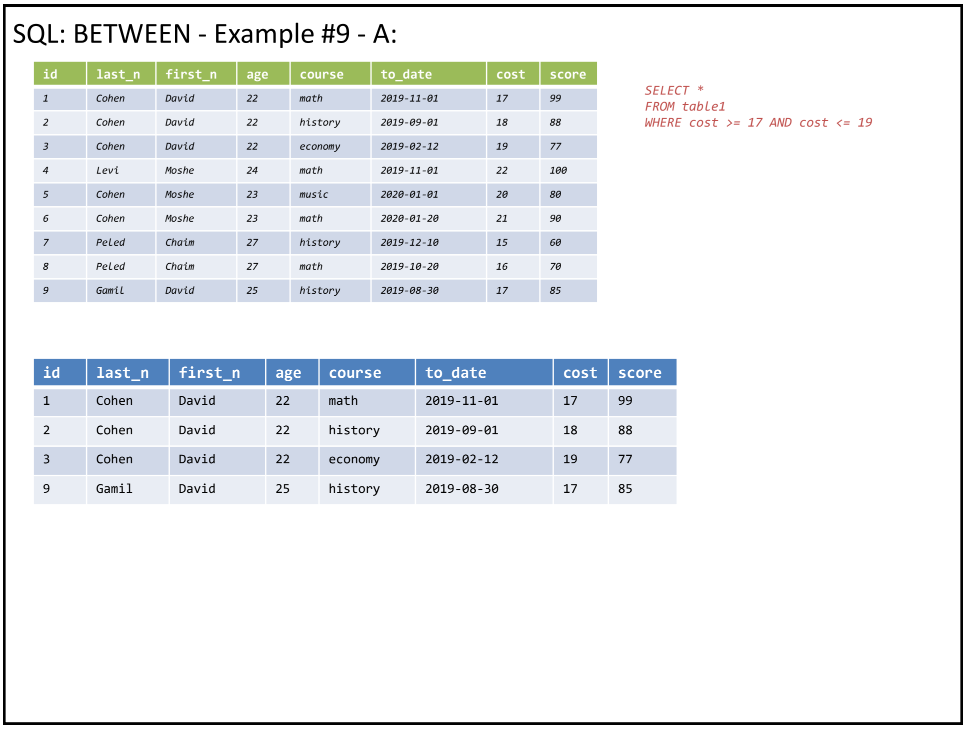
Learn SQL by Example (Chananel Perel 2023)

BETWEEN

SQL: BETWEEN - Example #9 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE cost >= 17 AND cost <= 19
```

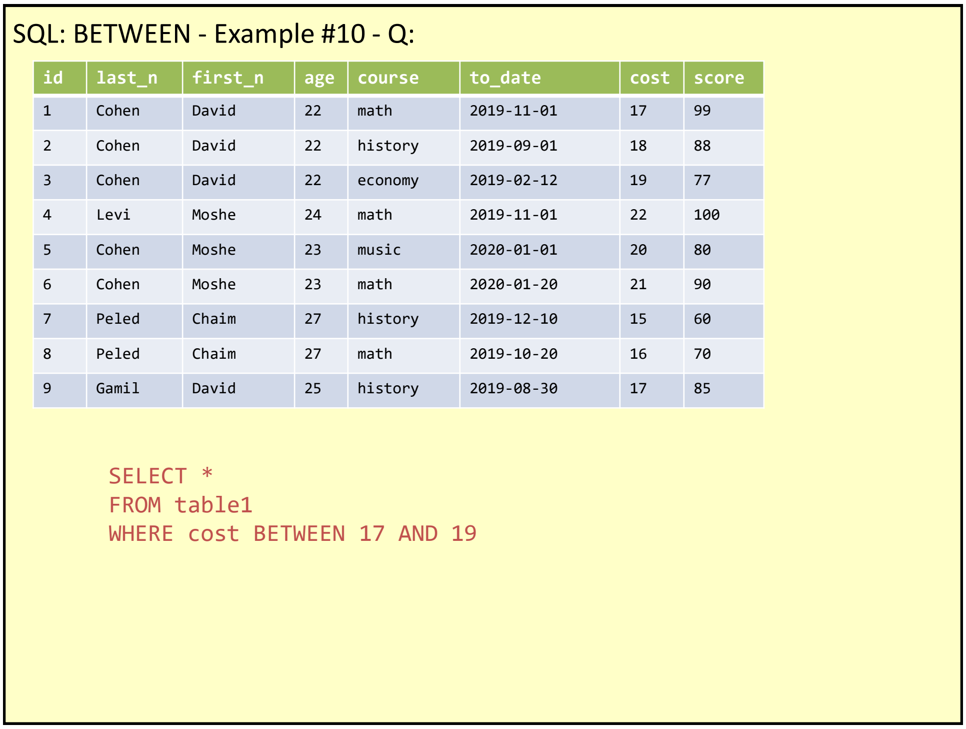


SQL: BETWEEN - Example #9 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	*Cohen*	*David*	*22*	*math*	*2019-11-01*	*17*	*99*
2	*Cohen*	*David*	*22*	*history*	*2019-09-01*	*18*	*88*
3	*Cohen*	*David*	*22*	*economy*	*2019-02-12*	*19*	*77*
4	*Levi*	*Moshe*	*24*	*math*	*2019-11-01*	*22*	*100*
5	*Cohen*	*Moshe*	*23*	*music*	*2020-01-01*	*20*	*80*
6	*Cohen*	*Moshe*	*23*	*math*	*2020-01-20*	*21*	*90*
7	*Peled*	*Chaim*	*27*	*history*	*2019-12-10*	*15*	*60*
8	*Peled*	*Chaim*	*27*	*math*	*2019-10-20*	*16*	*70*
9	*Gamil*	*David*	*25*	*history*	*2019-08-30*	*17*	*85*

```
SELECT *
FROM table1
WHERE cost >= 17 AND cost <= 19
```

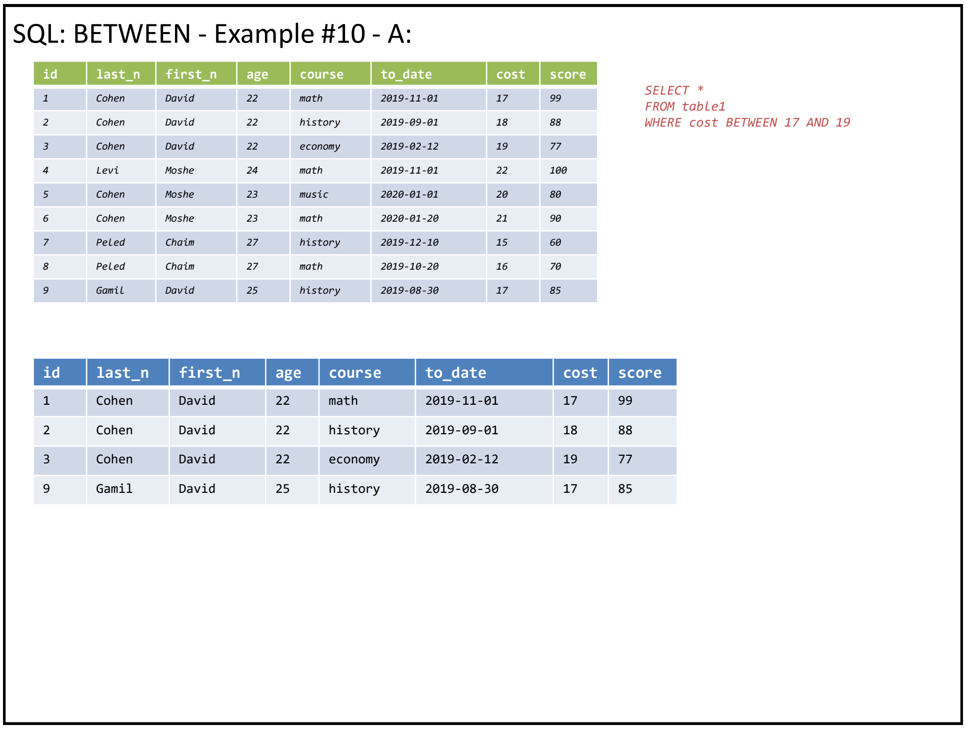
id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
9	Gamil	David	25	history	2019-08-30	17	85



SQL: BETWEEN - Example #10 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE cost BETWEEN 17 AND 19
```

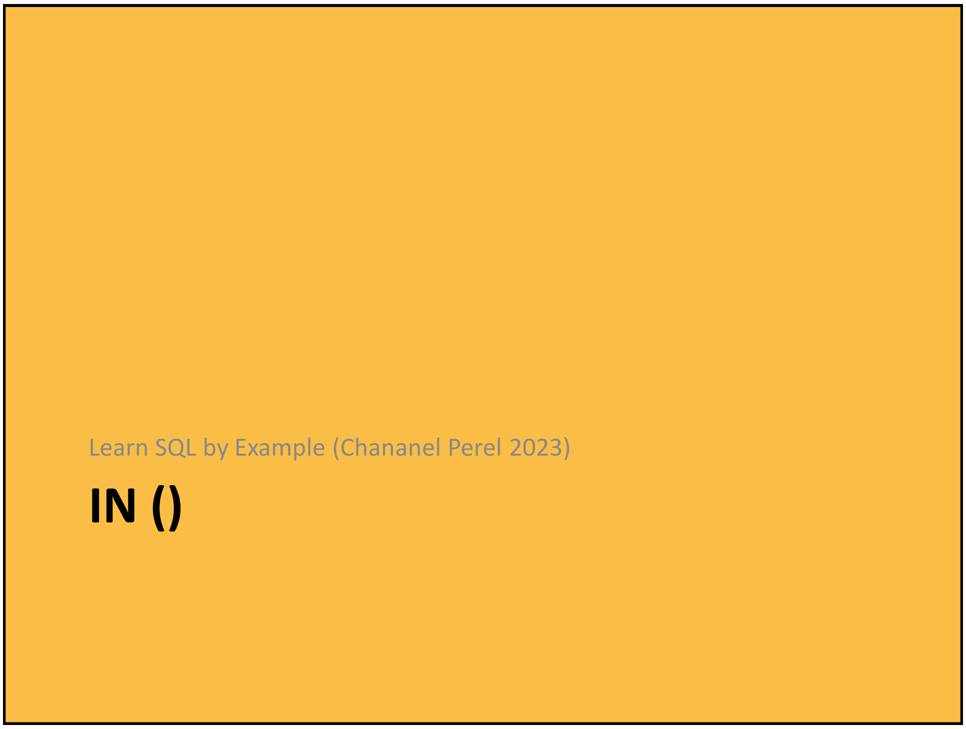


SQL: BETWEEN - Example #10 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

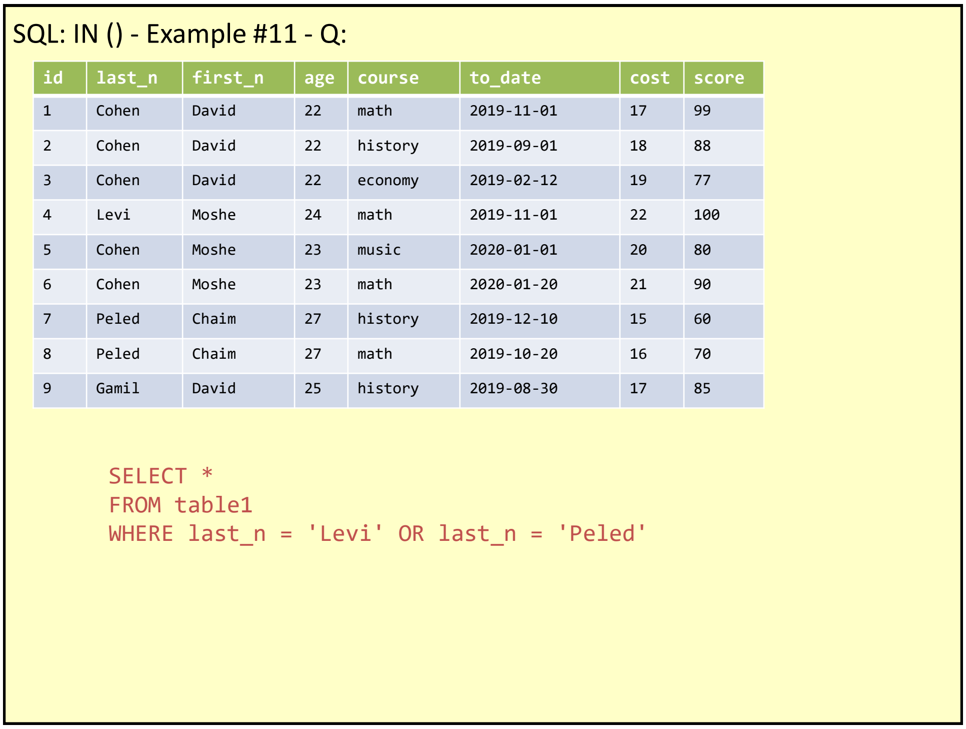
```
SELECT *
FROM table1
WHERE cost BETWEEN 17 AND 19
```

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
9	Gamil	David	25	history	2019-08-30	17	85



Learn SQL by Example (Chananel Perel 2023)

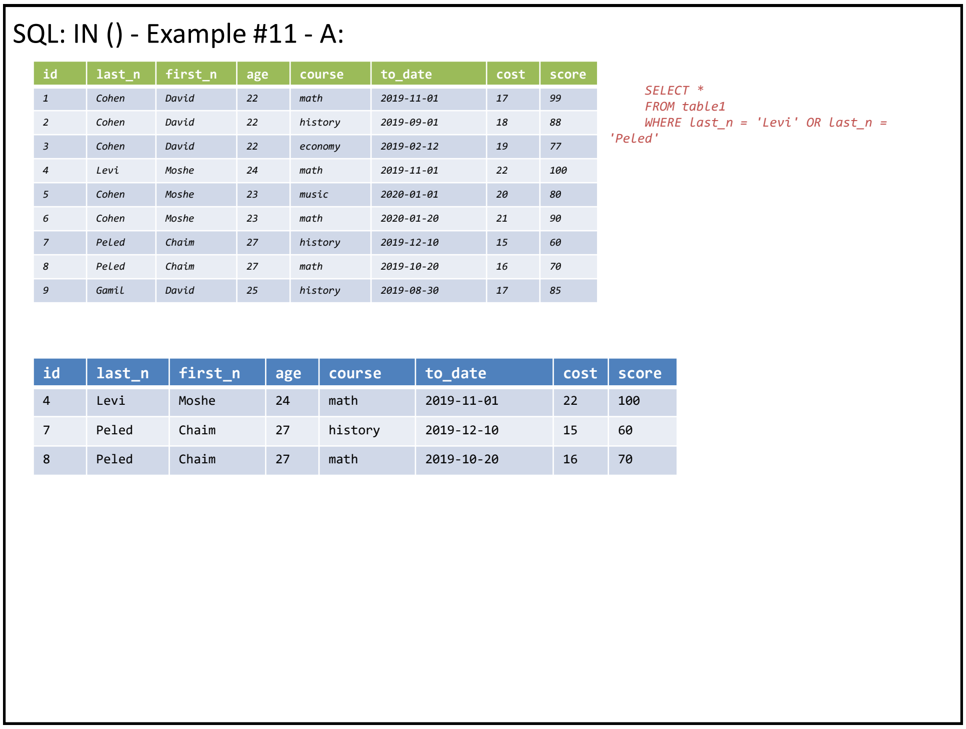
IN ()



SQL: IN () - Example #11 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE last_n = 'Levi' OR last_n = 'Peled'
```



SQL: IN () - Example #11 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
    SELECT *
    FROM table1
    WHERE last_n = 'Levi' OR last_n = 
'Peled'
```

id	last_n	first_n	age	course	to_date	cost	score
4	Levi	Moshe	24	math	2019-11-01	22	100
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70

SQL: IN () - Example #12 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

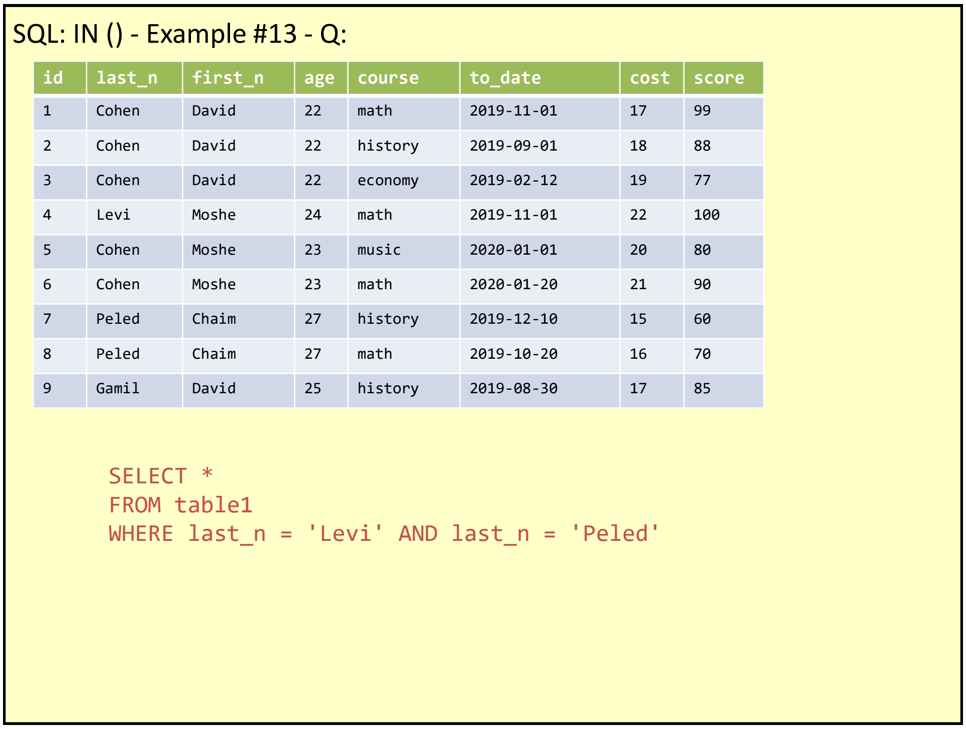
```
SELECT *
FROM table1
WHERE last_n IN ('Levi', 'Peled')
```

SQL: IN () - Example #12 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE last_n IN ('Levi', 'Peled')
```

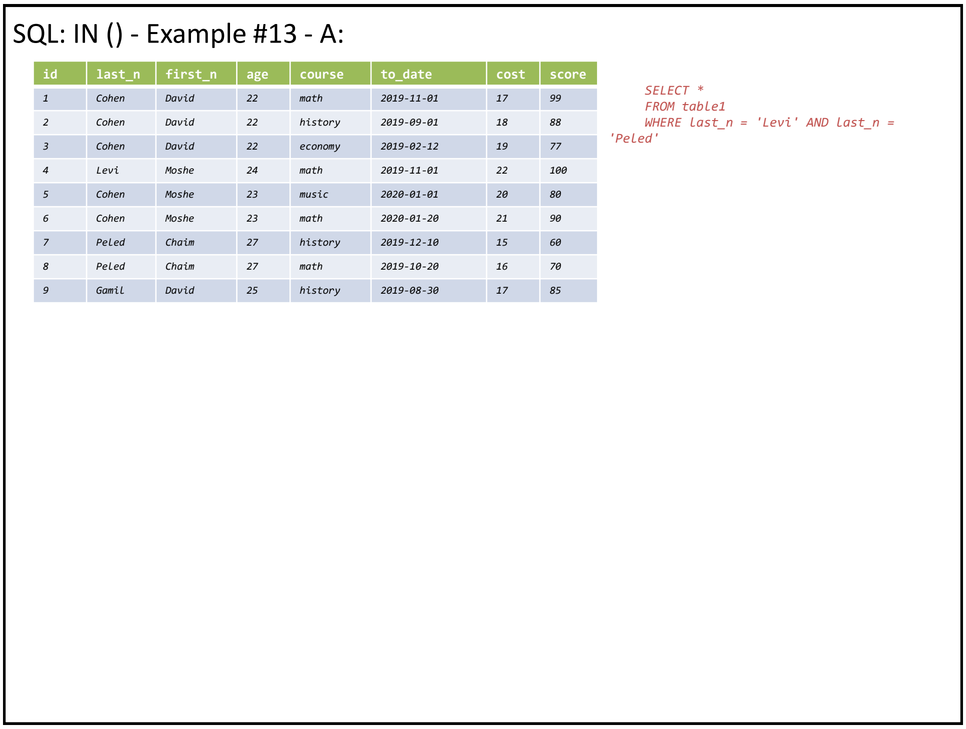
id	last_n	first_n	age	course	to_date	cost	score
4	Levi	Moshe	24	math	2019-11-01	22	100
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70



SQL: IN () - Example #13 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE last_n = 'Levi' AND last_n = 'Peled'
```



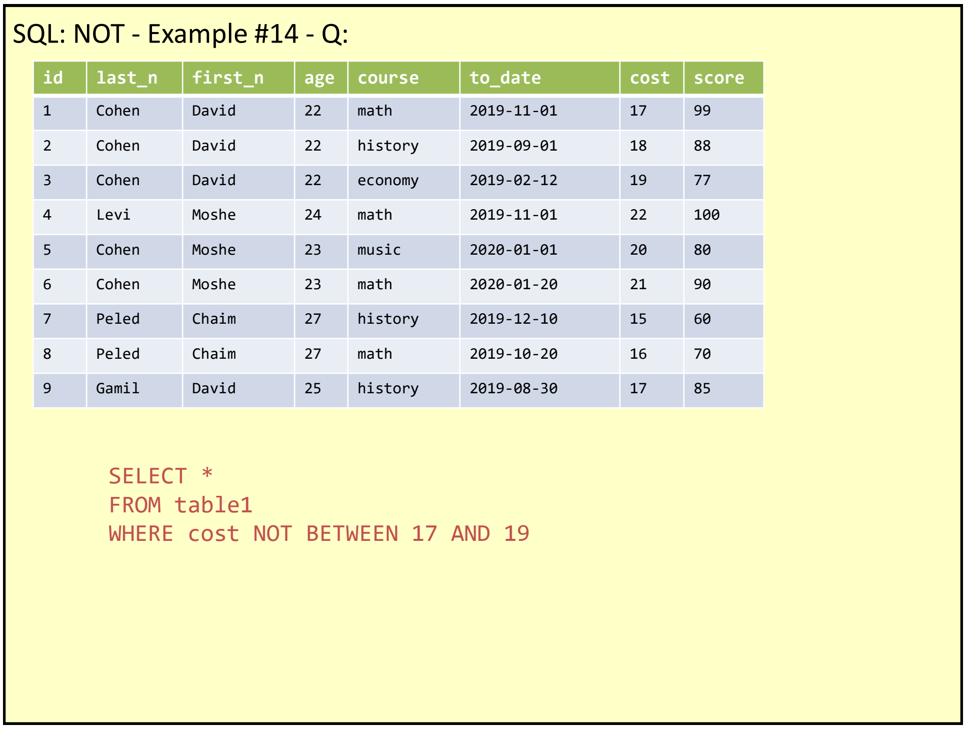
SQL: IN () - Example #13 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
    SELECT *
    FROM table1
    WHERE last_n = 'Levi' AND last_n =
'Peled'
```

Learn SQL by Example (Chananel Perel 2023)

NOT



SQL: NOT - Example #14 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE cost NOT BETWEEN 17 AND 19
```

SQL: NOT - Example #14 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

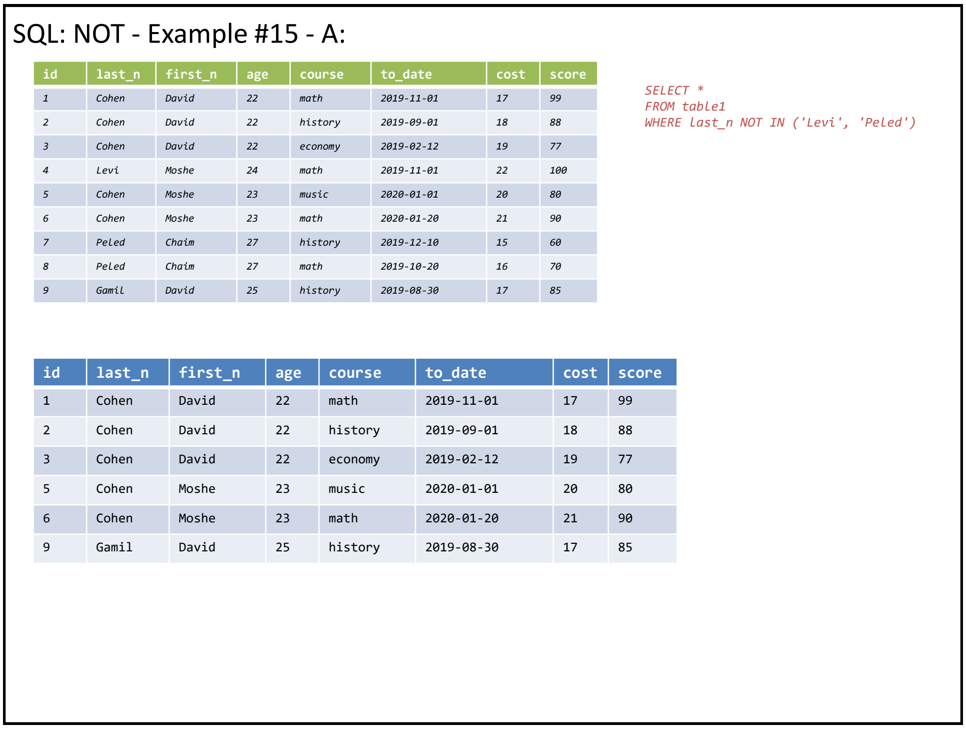
```
SELECT *
FROM table1
WHERE cost NOT BETWEEN 17 AND 19
```

id	last_n	first_n	age	course	to_date	cost	score
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70

SQL: NOT - Example #15 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE last_n NOT IN ('Levi', 'Peled')
```

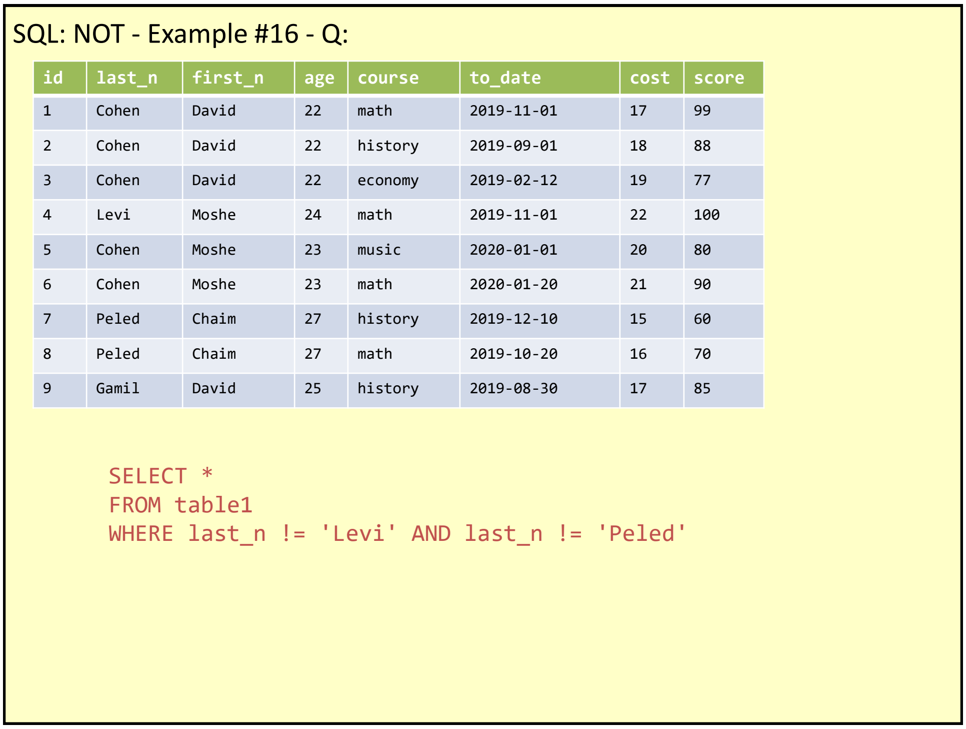


SQL: NOT - Example #15 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE last_n NOT IN ('Levi', 'Peled')
```

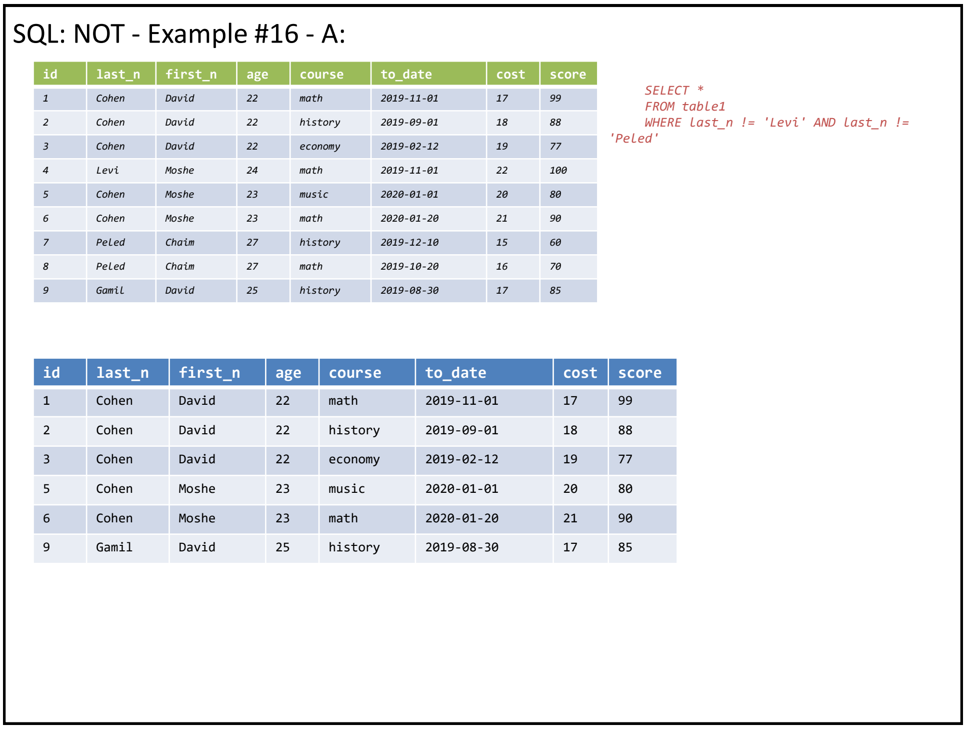
id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
9	Gamil	David	25	history	2019-08-30	17	85



SQL: NOT - Example #16 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE last_n != 'Levi' AND last_n != 'Peled'
```

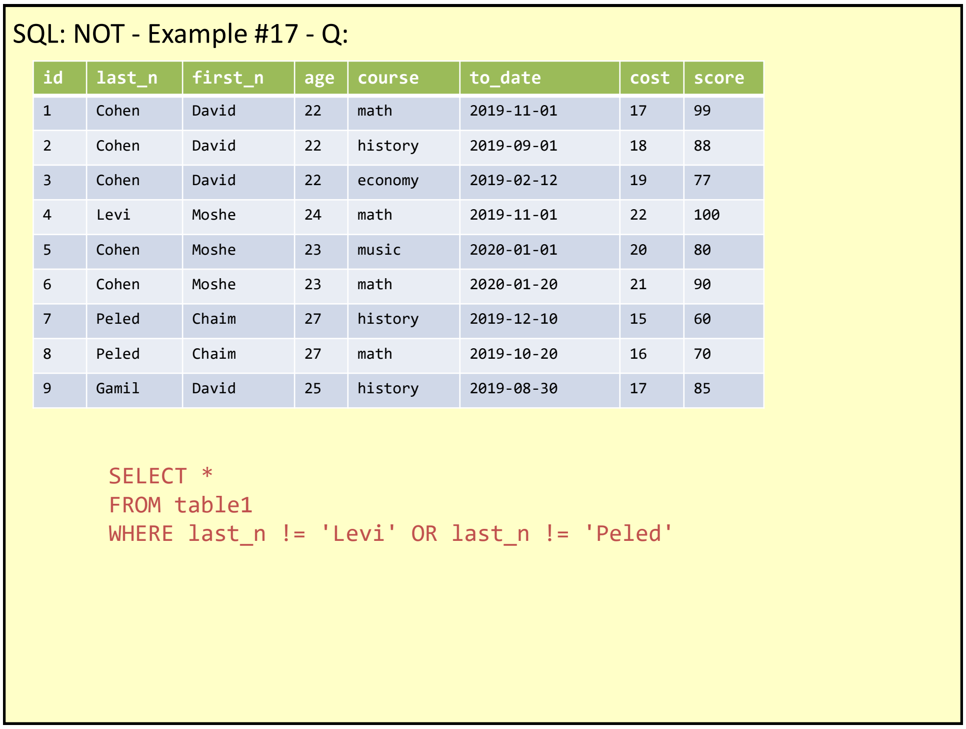


SQL: NOT - Example #16 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
    SELECT *
    FROM table1
    WHERE last_n != 'Levi' AND last_n !=
'Peled'
```

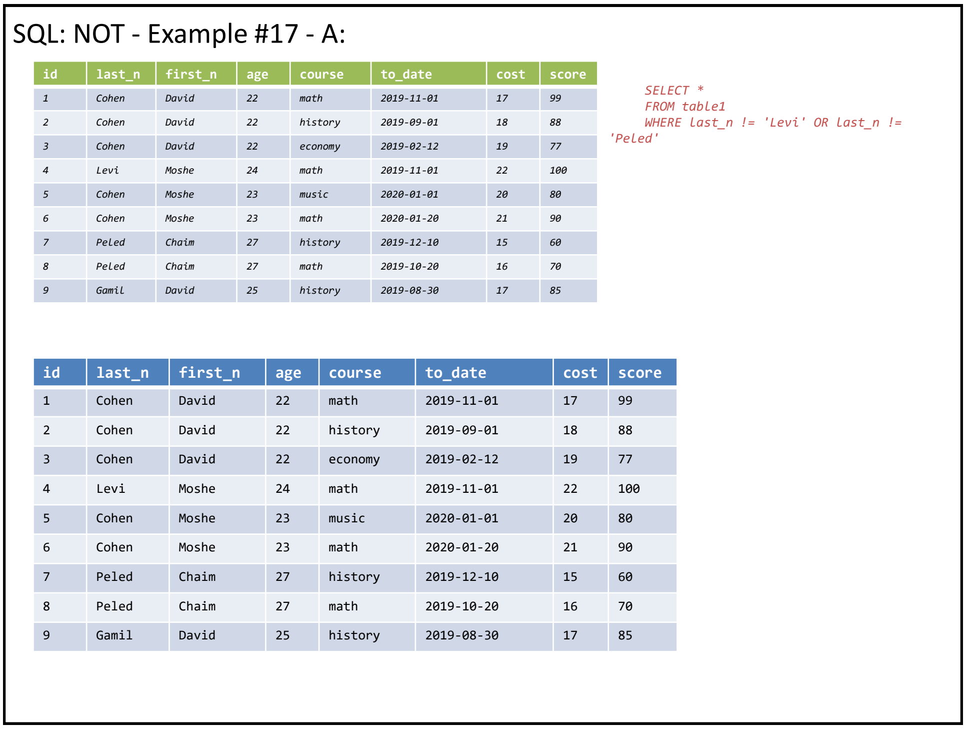
id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
9	Gamil	David	25	history	2019-08-30	17	85



SQL: NOT - Example #17 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE last_n != 'Levi' OR last_n != 'Peled'
```



SQL: NOT - Example #17 - A:

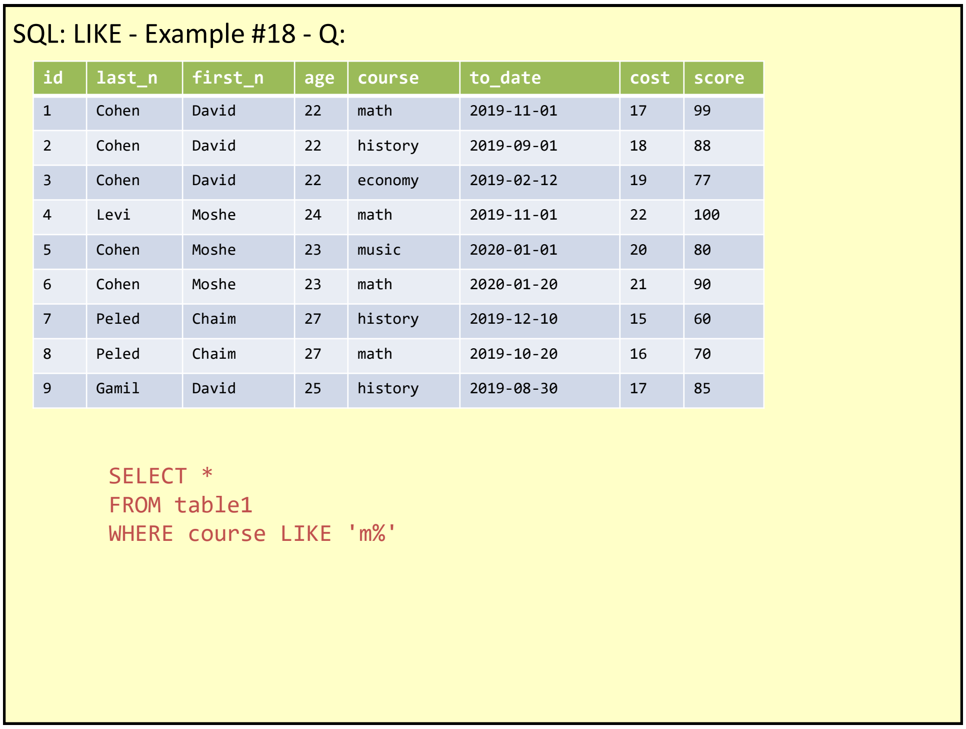
id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
    SELECT *
    FROM table1
    WHERE last_n != 'Levi' OR last_n != 
'Peled'
```

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

Learn SQL by Example (Chananel Perel 2023)

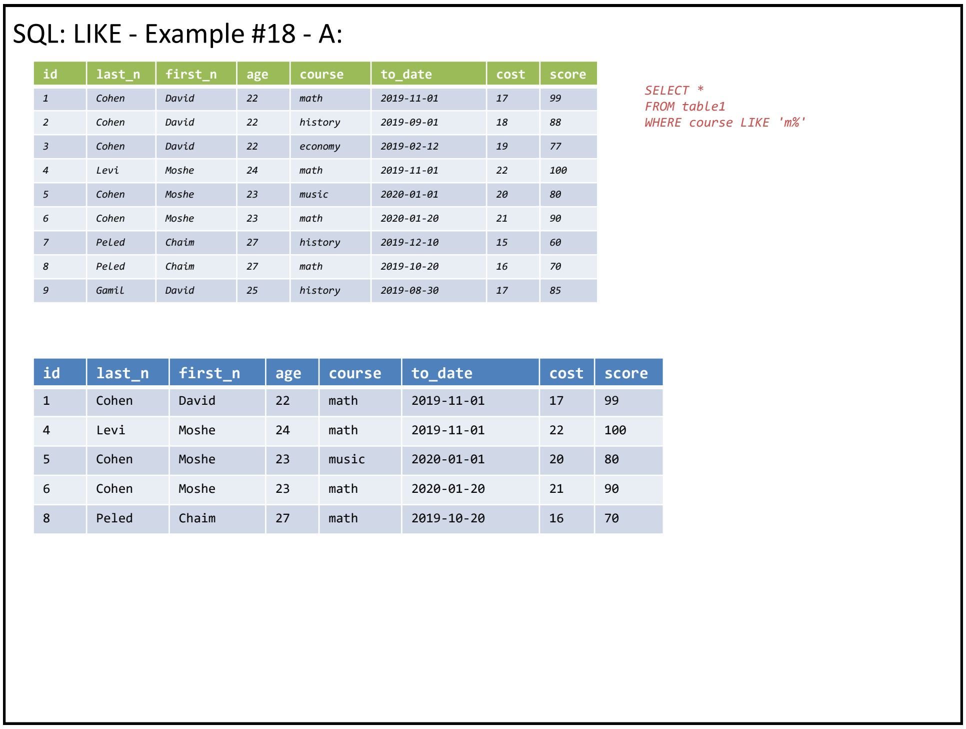
LIKE



SQL: LIKE - Example #18 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE course LIKE 'm%'
```



SQL: LIKE - Example #18 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE course LIKE 'm%'
```

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
8	Peled	Chaim	27	math	2019-10-20	16	70

SQL: LIKE - Example #19 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

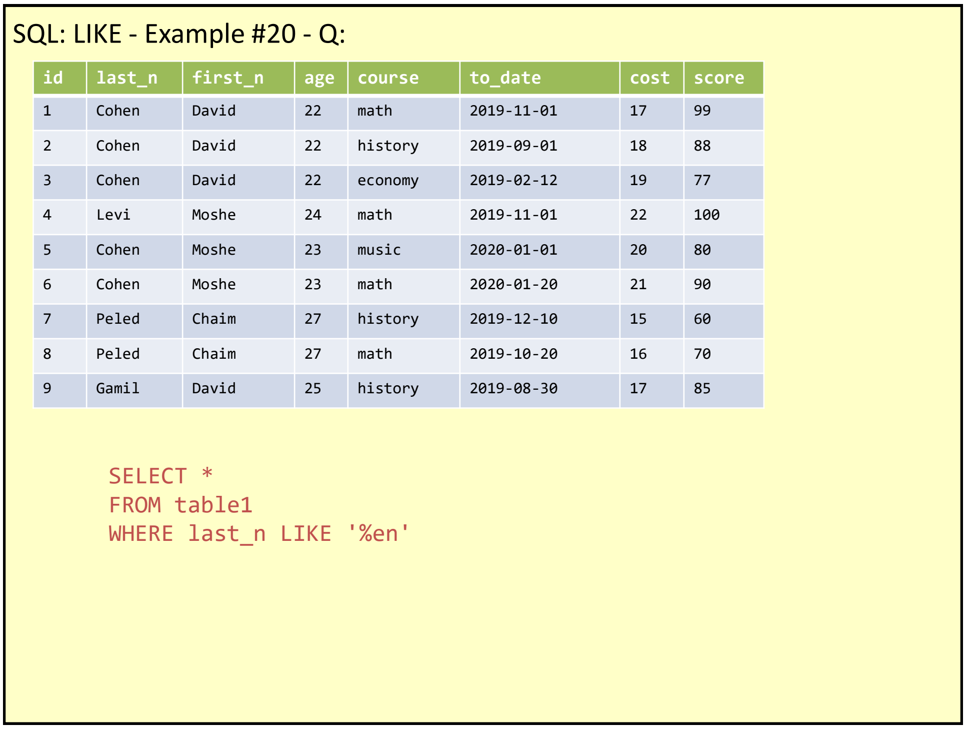
```
SELECT *
FROM table1
WHERE course LIKE 'm____'      -- (4*_ underscores)
```

SQL: LIKE - Example #19 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
     SELECT *
     FROM table1
     WHERE course LIKE 'm____'      -- (4*_
underscores)
```

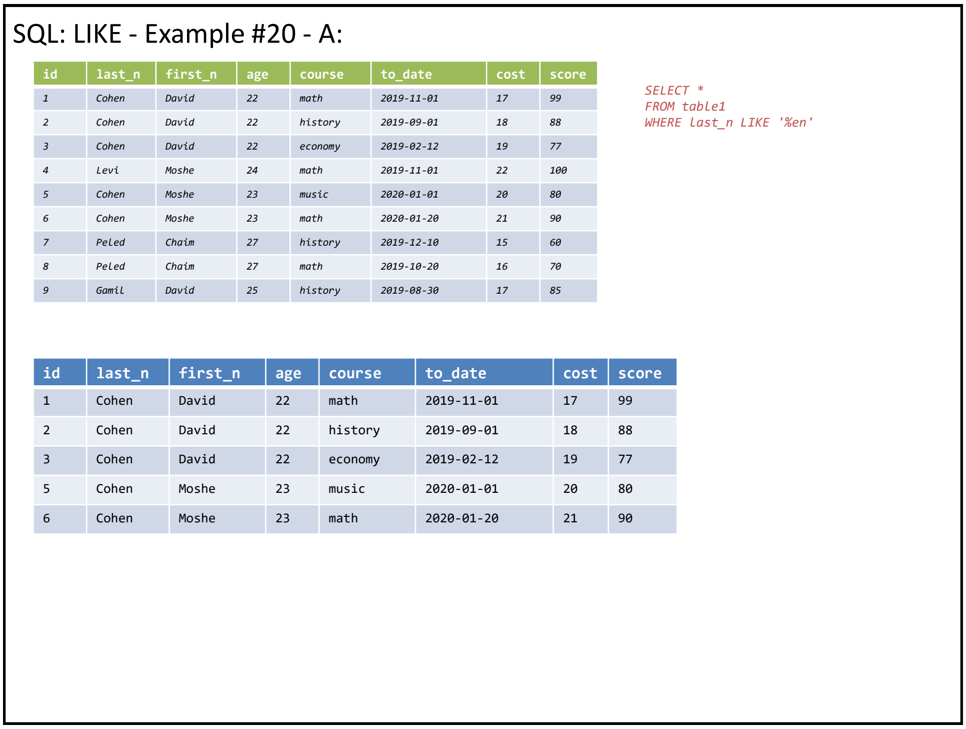
id	last_n	first_n	age	course	to_date	cost	score
5	Cohen	Moshe	23	music	2020-01-01	20	80



SQL: LIKE - Example #20 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE last_n LIKE '%en'
```

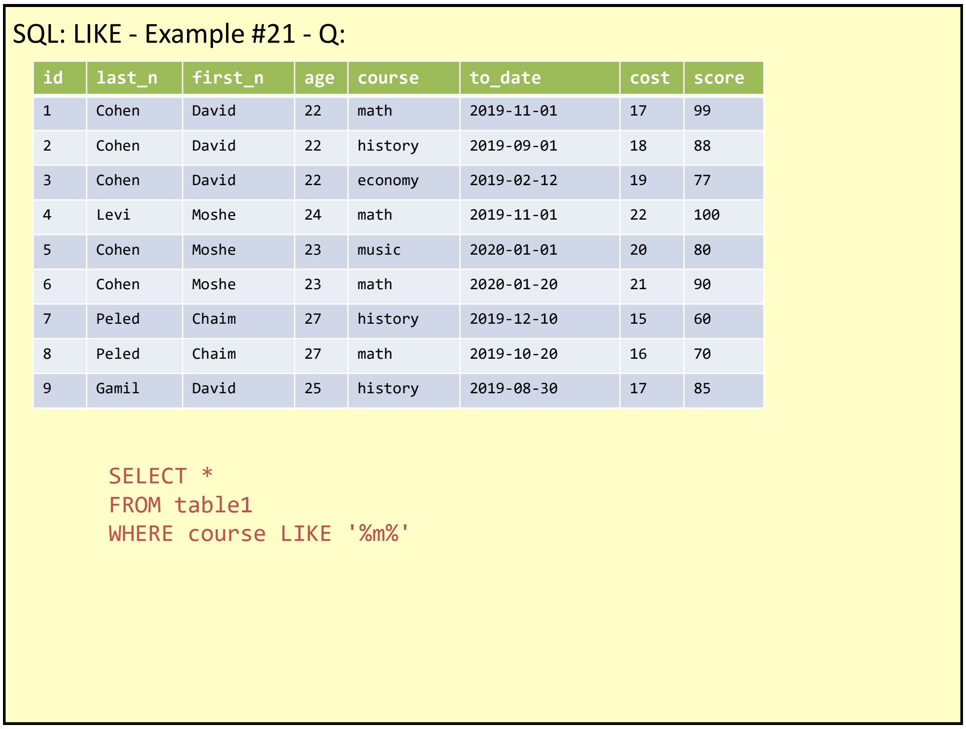


SQL: LIKE - Example #20 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE last_n LIKE '%en'
```

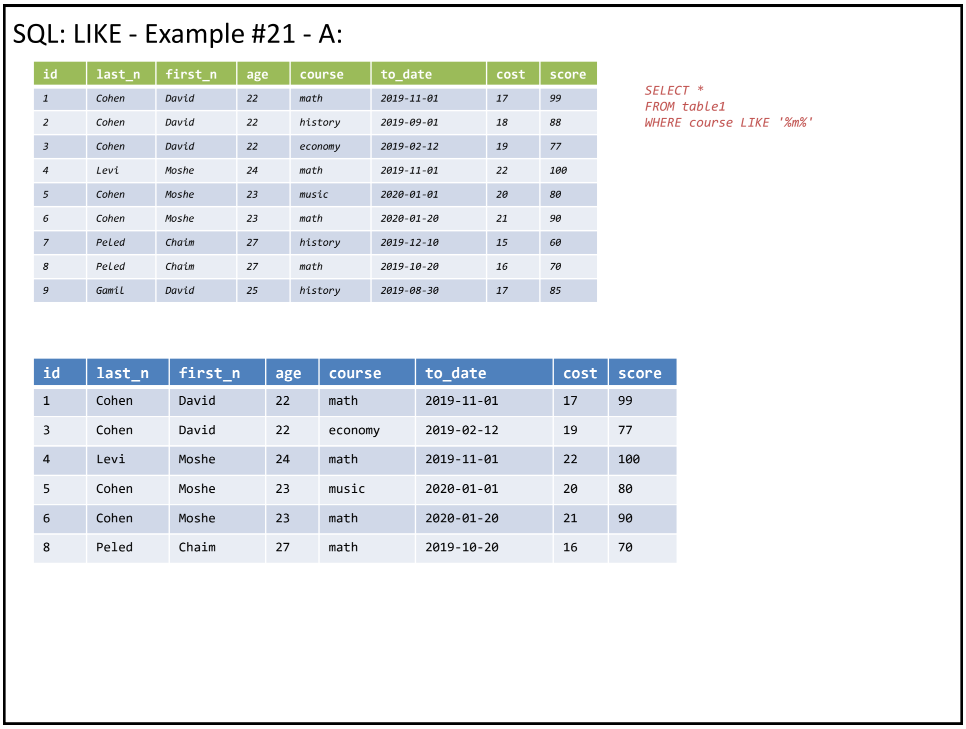
id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90



SQL: LIKE - Example #21 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
WHERE course LIKE '%m%'
```

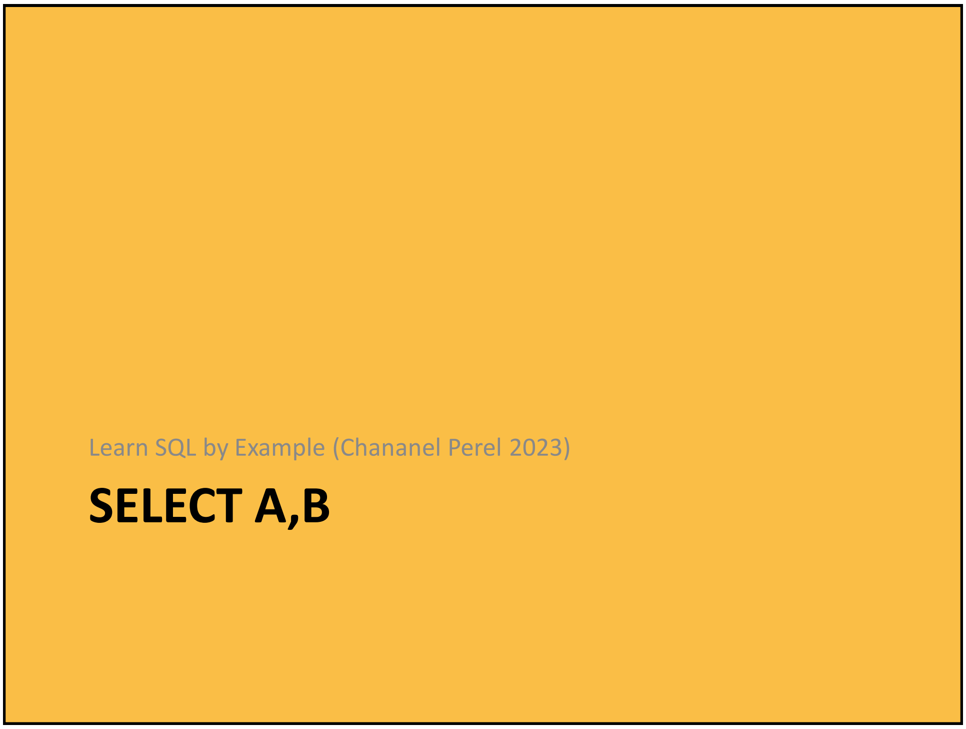


SQL: LIKE - Example #21 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

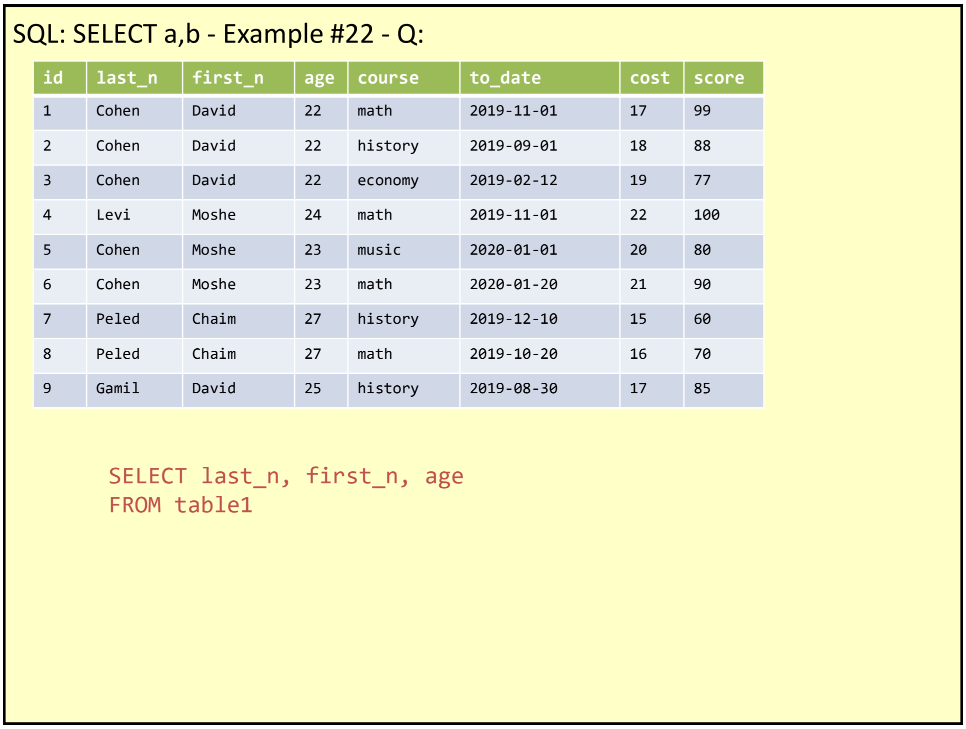
```
SELECT *
FROM table1
WHERE course LIKE '%m%'
```

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
8	Peled	Chaim	27	math	2019-10-20	16	70



Learn SQL by Example (Chananel Perel 2023)

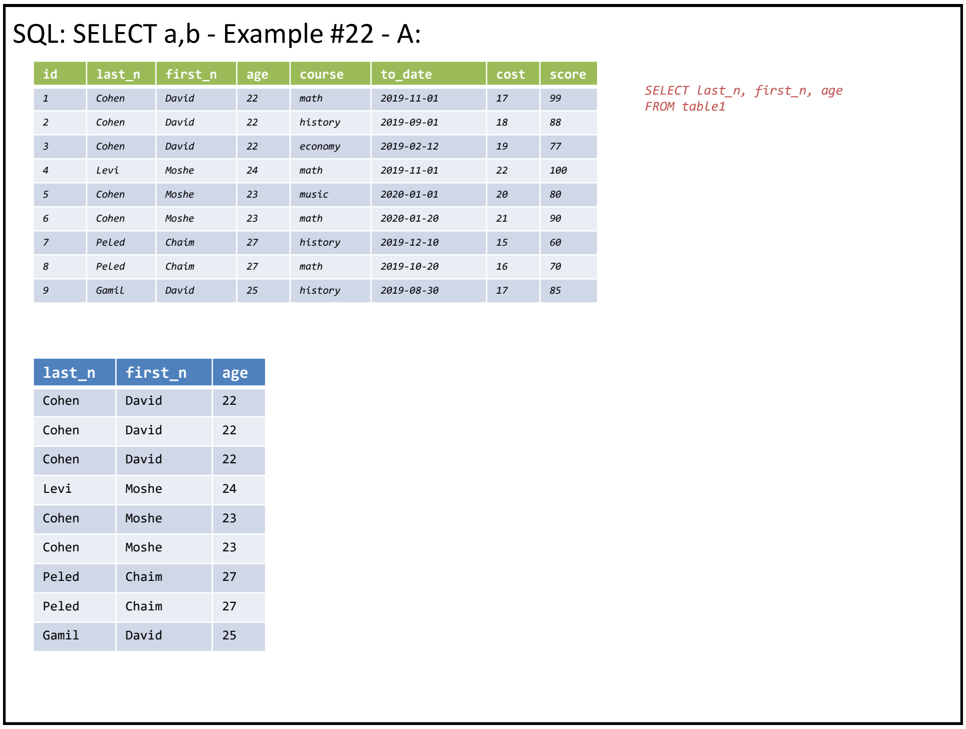
SELECT A,B



SQL: SELECT a,b - Example #22 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT last_n, first_n, age
FROM table1
```



SQL: SELECT a,b - Example #22 - A:

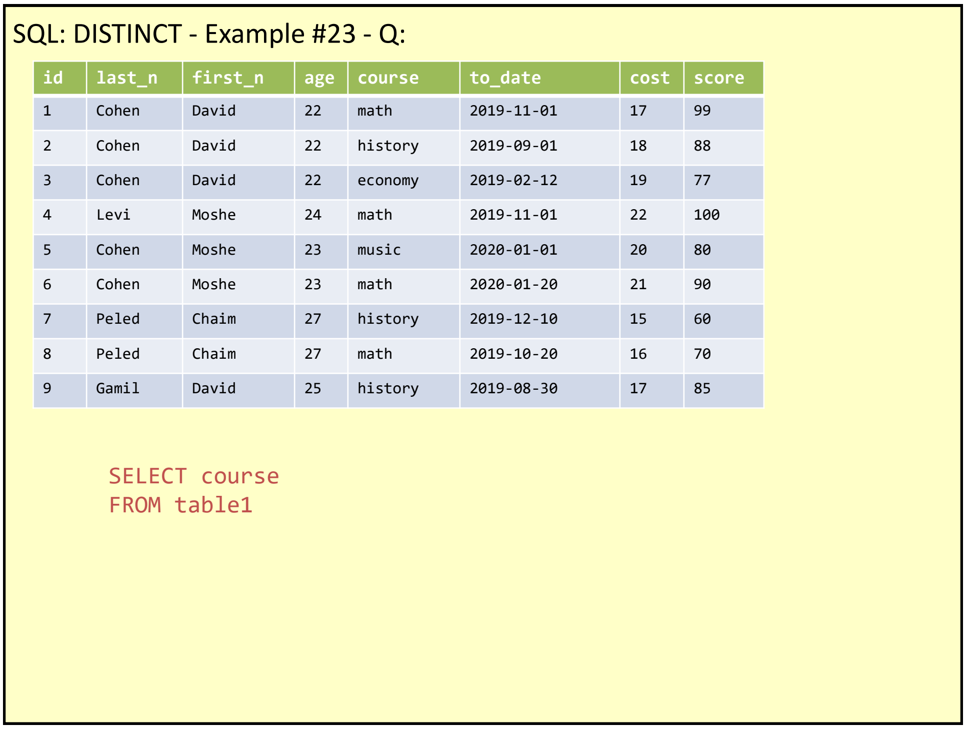
id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT last_n, first_n, age
FROM table1
```

last_n	first_n	age
Cohen	David	22
Cohen	David	22
Cohen	David	22
Levi	Moshe	24
Cohen	Moshe	23
Cohen	Moshe	23
Peled	Chaim	27
Peled	Chaim	27
Gamil	David	25

Learn SQL by Example (Chananel Perel 2023)

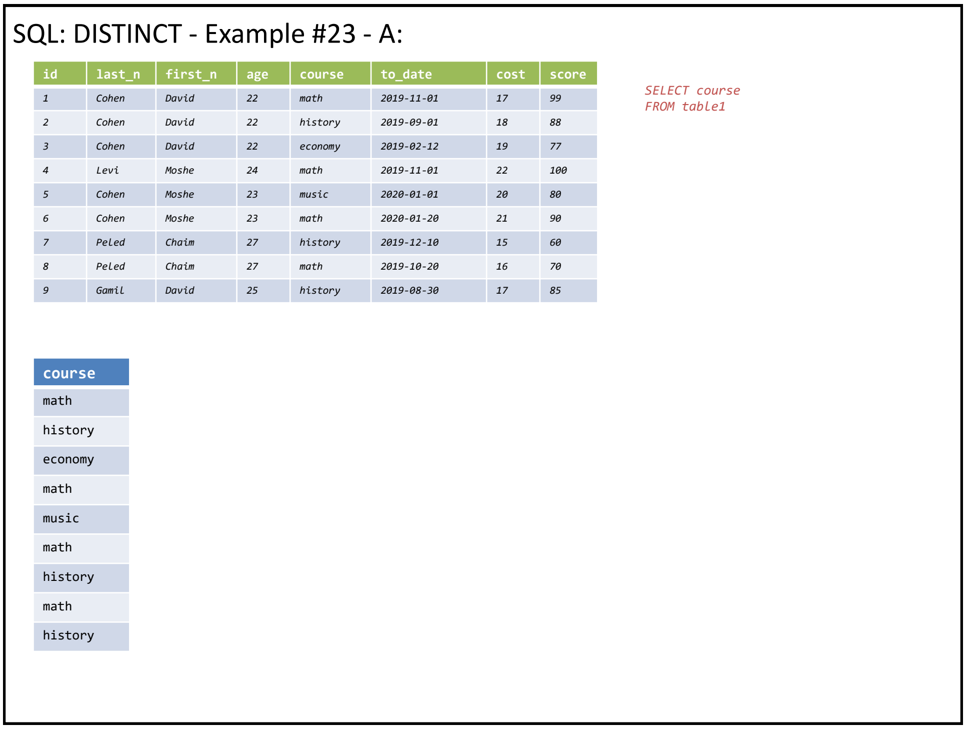
DISTINCT



SQL: DISTINCT - Example #23 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT course
FROM table1
```



SQL: DISTINCT - Example #23 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT course
FROM table1
```

course
math
history
economy
math
music
math
history
math
history

SQL: DISTINCT - Example #24 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

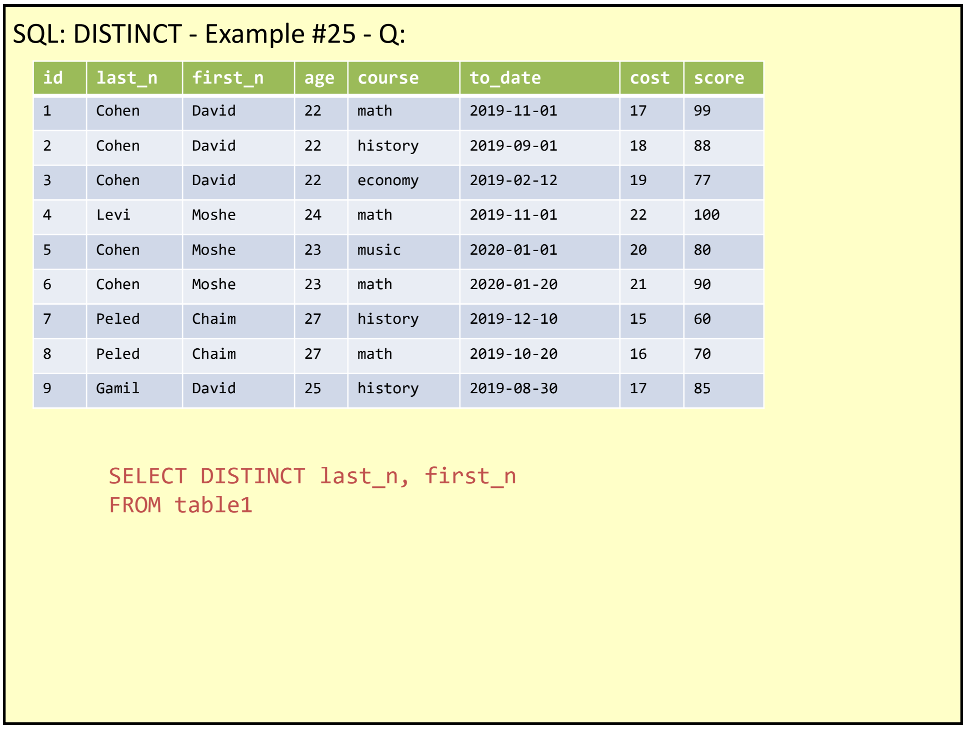
```
SELECT DISTINCT course
FROM table1
```

SQL: DISTINCT - Example #24 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	*Cohen*	*David*	*22*	*math*	*2019-11-01*	*17*	*99*
2	*Cohen*	*David*	*22*	*history*	*2019-09-01*	*18*	*88*
3	*Cohen*	*David*	*22*	*economy*	*2019-02-12*	*19*	*77*
4	*Levi*	*Moshe*	*24*	*math*	*2019-11-01*	*22*	*100*
5	*Cohen*	*Moshe*	*23*	*music*	*2020-01-01*	*20*	*80*
6	*Cohen*	*Moshe*	*23*	*math*	*2020-01-20*	*21*	*90*
7	*Peled*	*Chaim*	*27*	*history*	*2019-12-10*	*15*	*60*
8	*Peled*	*Chaim*	*27*	*math*	*2019-10-20*	*16*	*70*
9	*Gamil*	*David*	*25*	*history*	*2019-08-30*	*17*	*85*

```
SELECT DISTINCT course
FROM table1
```

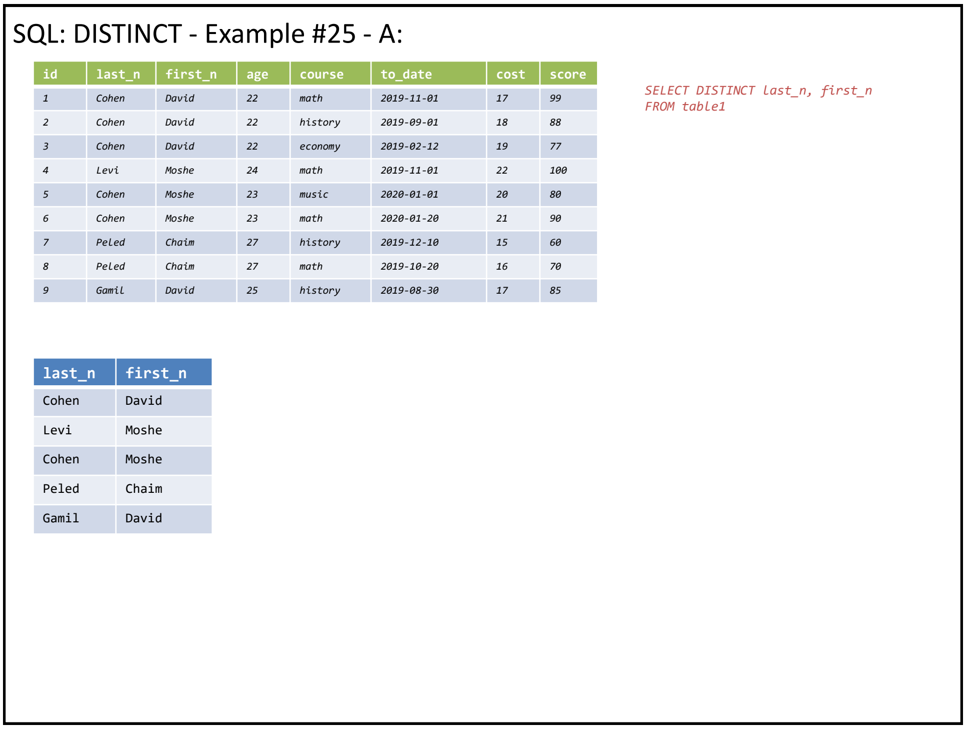
course
math
history
economy
music



SQL: DISTINCT - Example #25 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT DISTINCT last_n, first_n
FROM table1
```

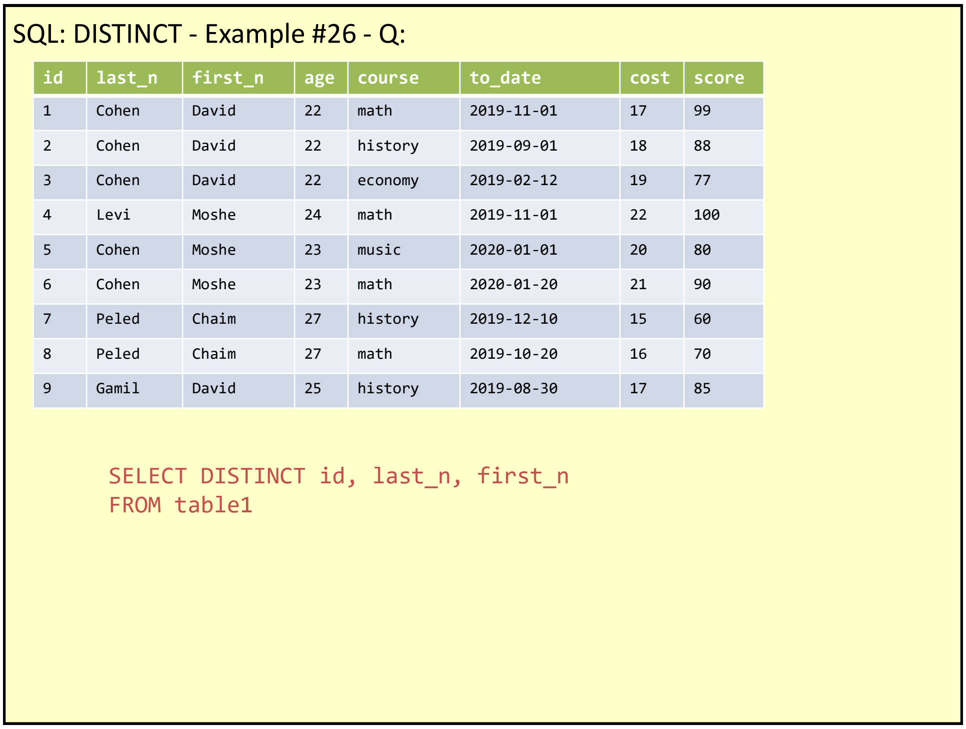


SQL: DISTINCT - Example #25 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT DISTINCT last_n, first_n
FROM table1
```

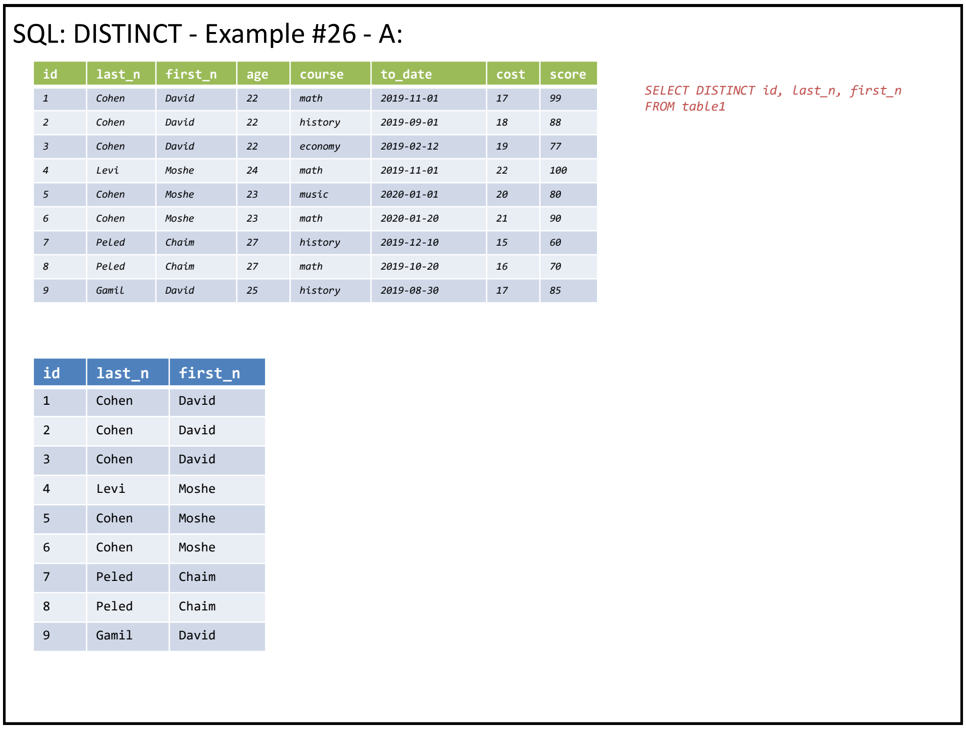
last_n	first_n
Cohen	David
Levi	Moshe
Cohen	Moshe
Peled	Chaim
Gamil	David



SQL: DISTINCT - Example #26 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT DISTINCT id, last_n, first_n
FROM table1
```



SQL: DISTINCT - Example #26 - A:

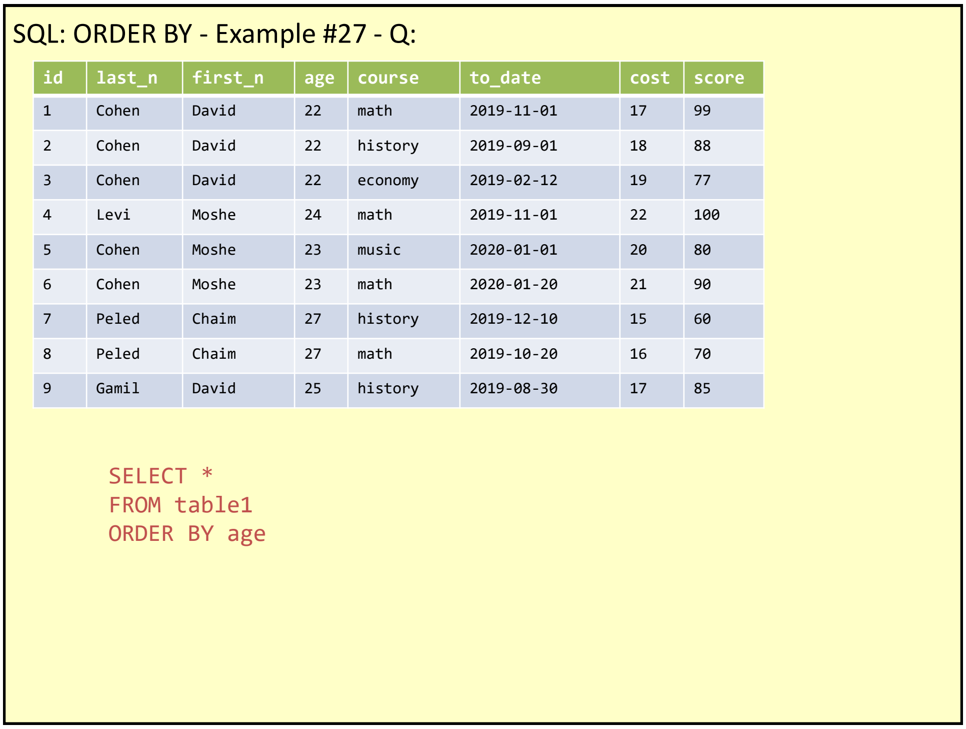
id	last_n	first_n	age	course	to_date	cost	score
1	*Cohen*	*David*	*22*	*math*	*2019-11-01*	*17*	*99*
2	*Cohen*	*David*	*22*	*history*	*2019-09-01*	*18*	*88*
3	*Cohen*	*David*	*22*	*economy*	*2019-02-12*	*19*	*77*
4	*Levi*	*Moshe*	*24*	*math*	*2019-11-01*	*22*	*100*
5	*Cohen*	*Moshe*	*23*	*music*	*2020-01-01*	*20*	*80*
6	*Cohen*	*Moshe*	*23*	*math*	*2020-01-20*	*21*	*90*
7	*Peled*	*Chaim*	*27*	*history*	*2019-12-10*	*15*	*60*
8	*Peled*	*Chaim*	*27*	*math*	*2019-10-20*	*16*	*70*
9	*Gamil*	*David*	*25*	*history*	*2019-08-30*	*17*	*85*

```
SELECT DISTINCT id, last_n, first_n
FROM table1
```

id	last_n	first_n
1	Cohen	David
2	Cohen	David
3	Cohen	David
4	Levi	Moshe
5	Cohen	Moshe
6	Cohen	Moshe
7	Peled	Chaim
8	Peled	Chaim
9	Gamil	David

Learn SQL by Example (Chananel Perel 2023)

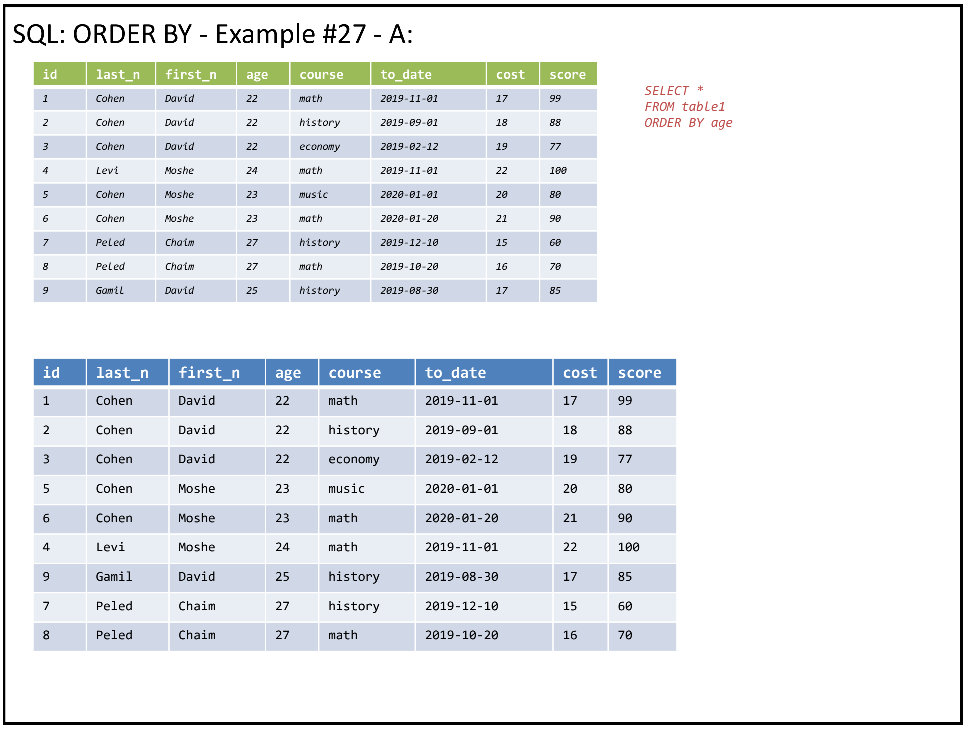
ORDER BY



SQL: ORDER BY - Example #27 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
ORDER BY age
```

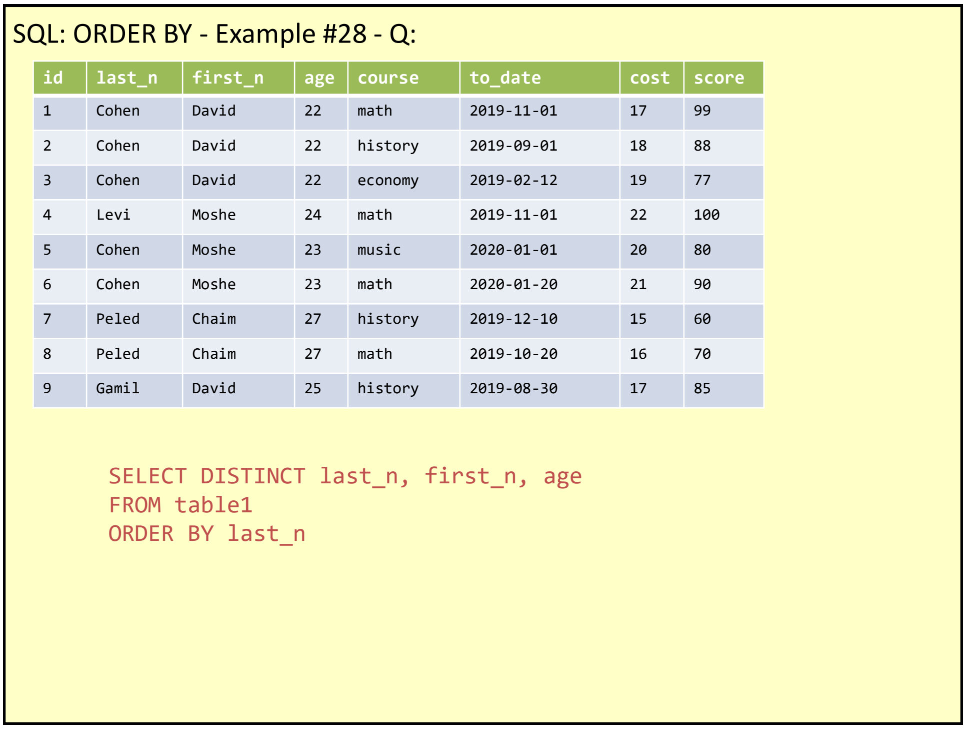


SQL: ORDER BY - Example #27 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	*Cohen*	*David*	*22*	*math*	*2019-11-01*	*17*	*99*
2	*Cohen*	*David*	*22*	*history*	*2019-09-01*	*18*	*88*
3	*Cohen*	*David*	*22*	*economy*	*2019-02-12*	*19*	*77*
4	*Levi*	*Moshe*	*24*	*math*	*2019-11-01*	*22*	*100*
5	*Cohen*	*Moshe*	*23*	*music*	*2020-01-01*	*20*	*80*
6	*Cohen*	*Moshe*	*23*	*math*	*2020-01-20*	*21*	*90*
7	*Peled*	*Chaim*	*27*	*history*	*2019-12-10*	*15*	*60*
8	*Peled*	*Chaim*	*27*	*math*	*2019-10-20*	*16*	*70*
9	*Gamil*	*David*	*25*	*history*	*2019-08-30*	*17*	*85*

```
SELECT *
FROM table1
ORDER BY age
```

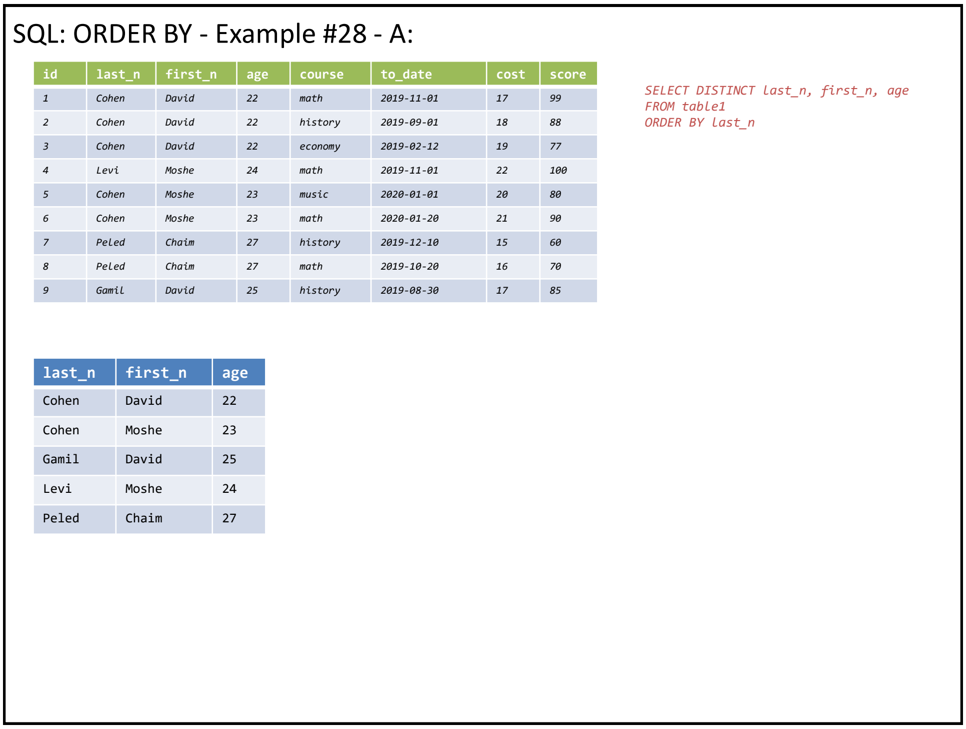
id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
4	Levi	Moshe	24	math	2019-11-01	22	100
9	Gamil	David	25	history	2019-08-30	17	85
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70



SQL: ORDER BY - Example #28 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT DISTINCT last_n, first_n, age
FROM table1
ORDER BY last_n
```

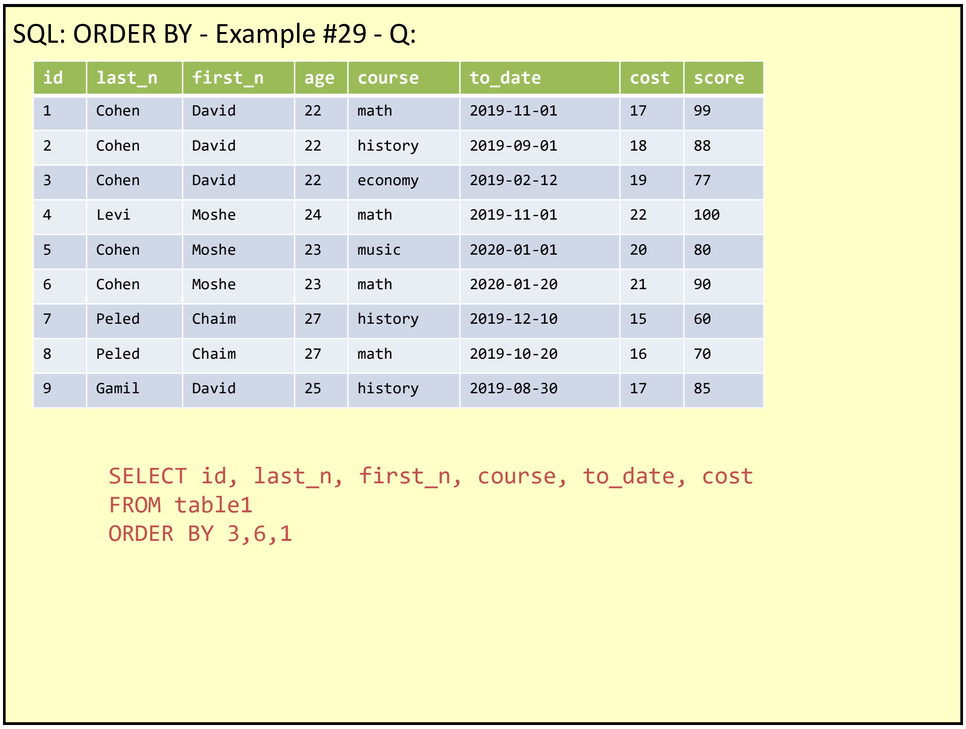


SQL: ORDER BY - Example #28 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT DISTINCT last_n, first_n, age
FROM table1
ORDER BY last_n
```

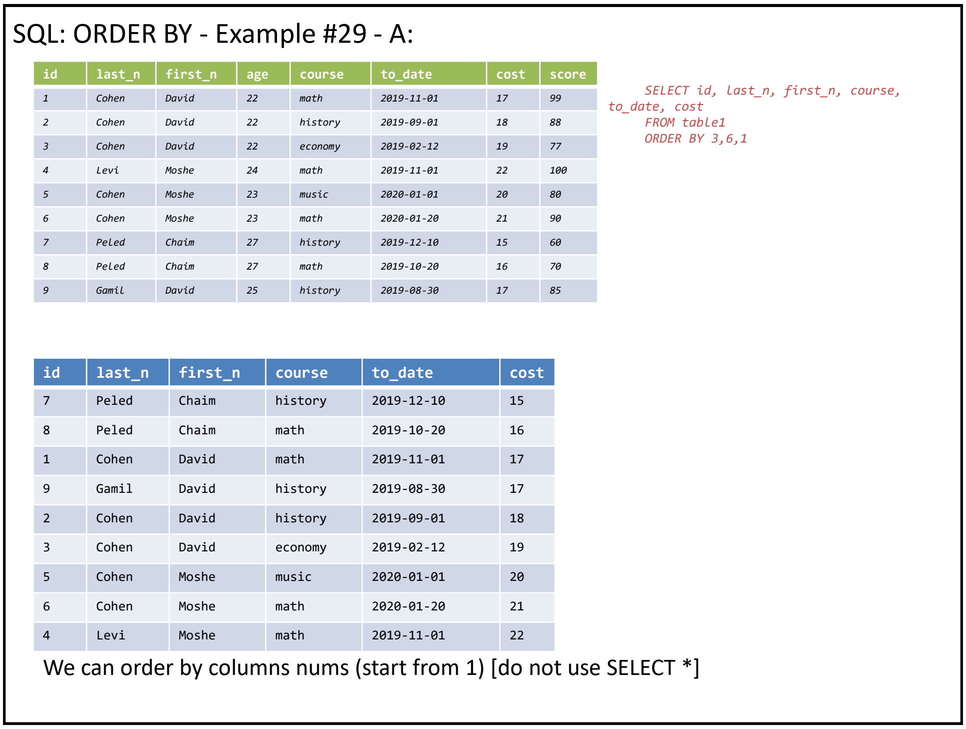
last_n	first_n	age
Cohen	David	22
Cohen	Moshe	23
Gamil	David	25
Levi	Moshe	24
Peled	Chaim	27



SQL: ORDER BY - Example #29 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT id, last_n, first_n, course, to_date, cost
FROM table1
ORDER BY 3,6,1
```



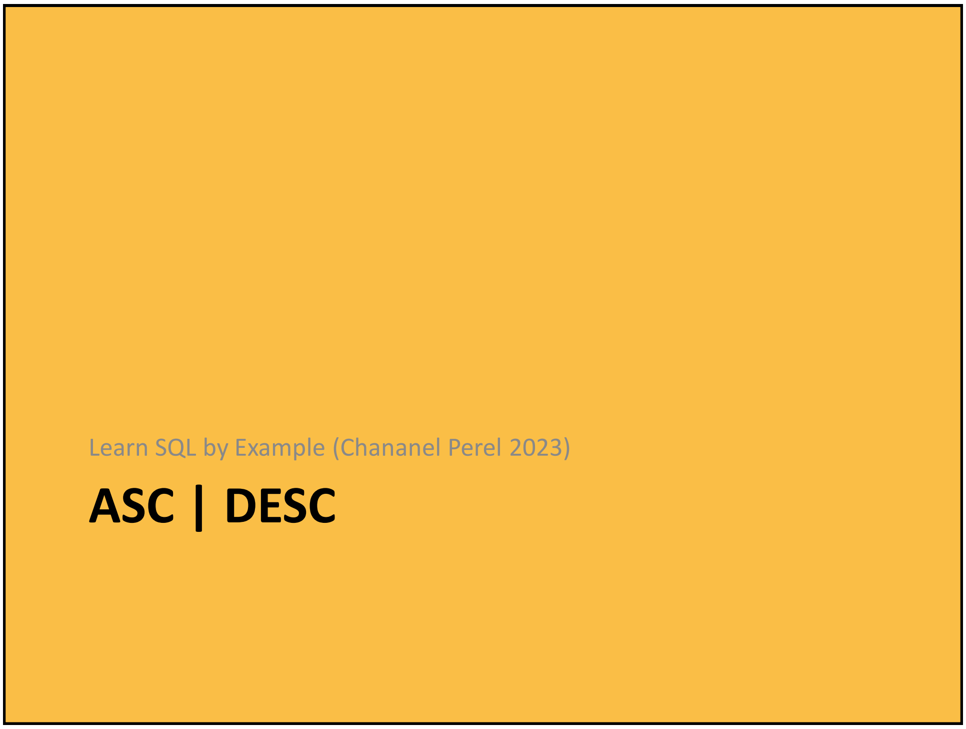
SQL: ORDER BY - Example #29 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
    SELECT id, last_n, first_n, course,
to_date, cost
    FROM table1
    ORDER BY 3,6,1
```

id	last_n	first_n	course	to_date	cost
7	Peled	Chaim	history	2019-12-10	15
8	Peled	Chaim	math	2019-10-20	16
1	Cohen	David	math	2019-11-01	17
9	Gamil	David	history	2019-08-30	17
2	Cohen	David	history	2019-09-01	18
3	Cohen	David	economy	2019-02-12	19
5	Cohen	Moshe	music	2020-01-01	20
6	Cohen	Moshe	math	2020-01-20	21
4	Levi	Moshe	math	2019-11-01	22

We can order by columns nums (start from 1) [do not use SELECT *]



Learn SQL by Example (Chananel Perel 2023)

ASC | DESC

SQL: ASC | DESC - Example #30 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

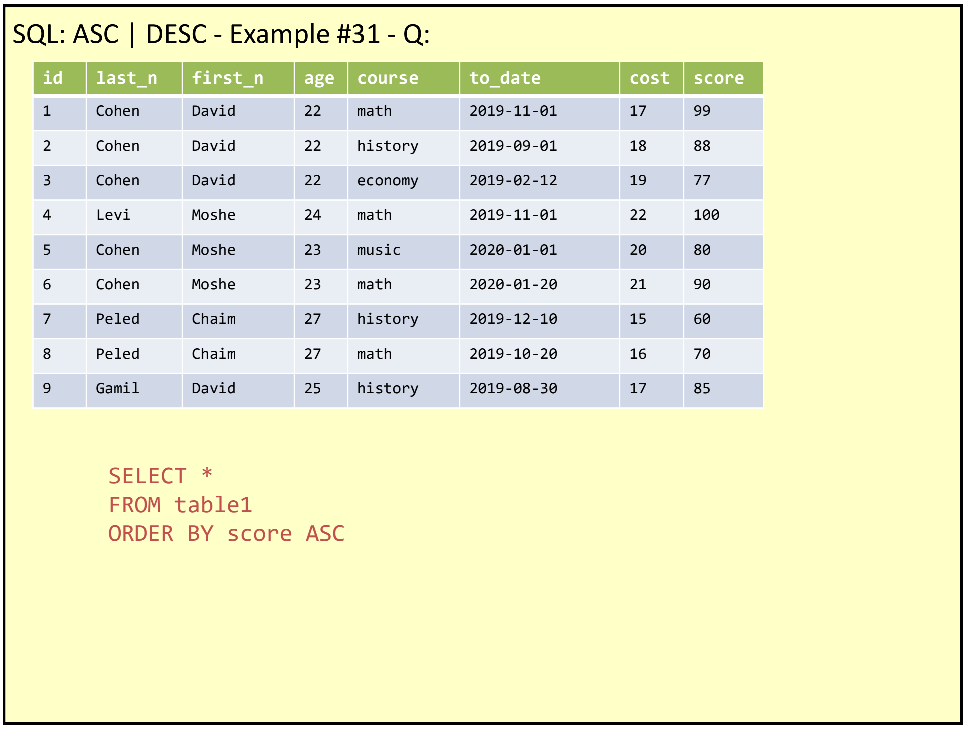
```
SELECT *
FROM table1
ORDER BY id DESC
```

SQL: ASC | DESC - Example #30 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
ORDER BY id DESC
```

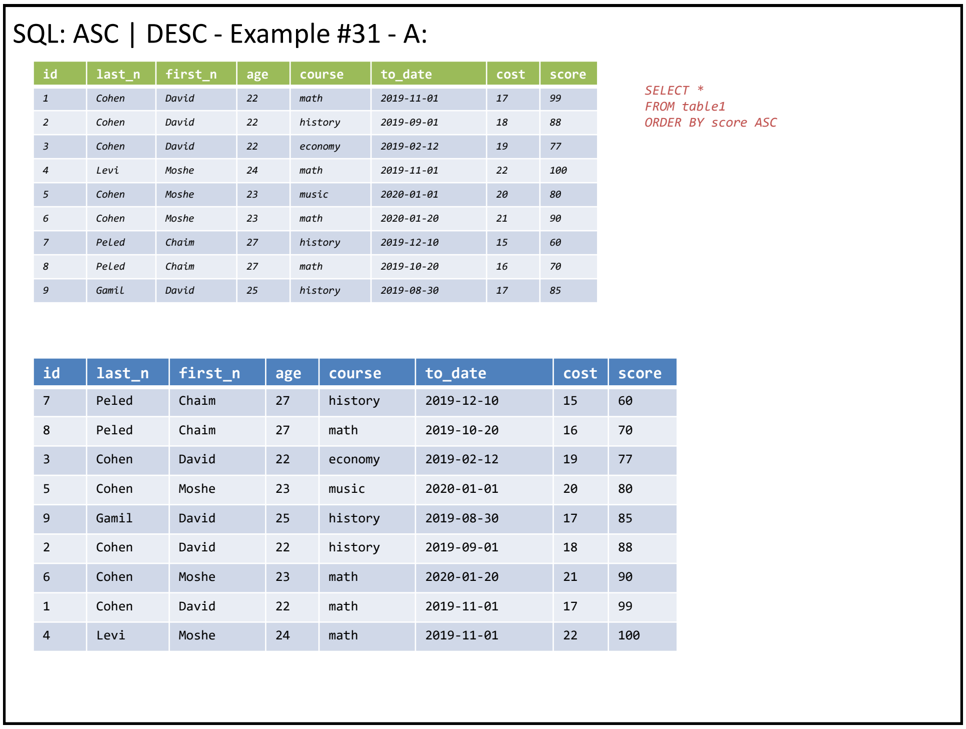
id	last_n	first_n	age	course	to_date	cost	score
9	Gamil	David	25	history	2019-08-30	17	85
8	Peled	Chaim	27	math	2019-10-20	16	70
7	Peled	Chaim	27	history	2019-12-10	15	60
6	Cohen	Moshe	23	math	2020-01-20	21	90
5	Cohen	Moshe	23	music	2020-01-01	20	80
4	Levi	Moshe	24	math	2019-11-01	22	100
3	Cohen	David	22	economy	2019-02-12	19	77
2	Cohen	David	22	history	2019-09-01	18	88
1	Cohen	David	22	math	2019-11-01	17	99



SQL: ASC | DESC - Example #31 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
ORDER BY score ASC
```

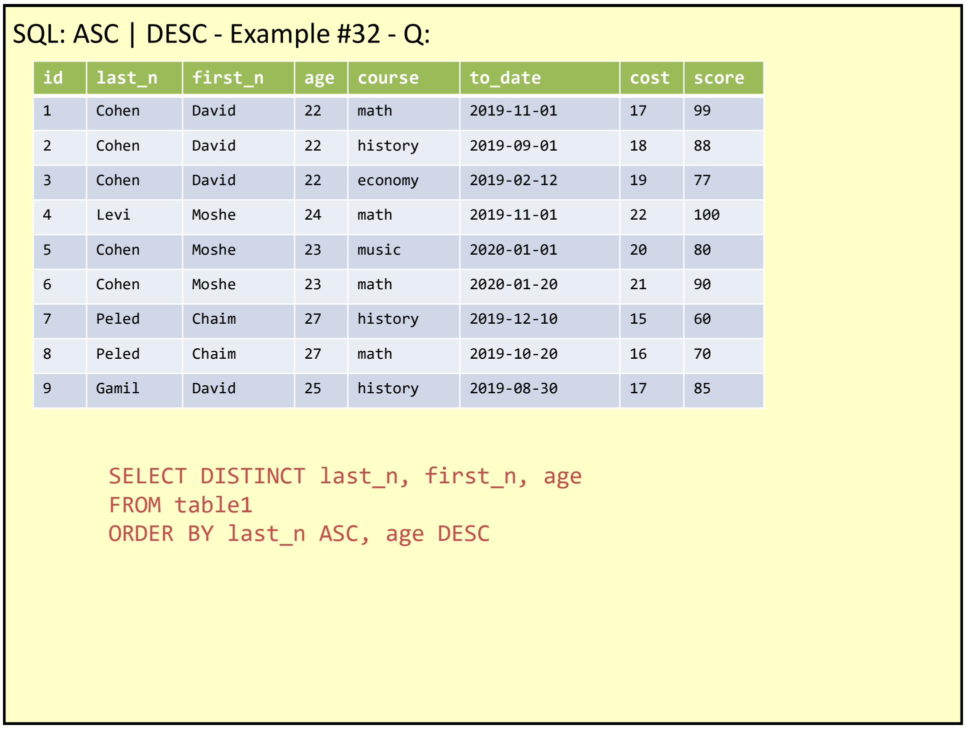


SQL: ASC | DESC - Example #31 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *
FROM table1
ORDER BY score ASC
```

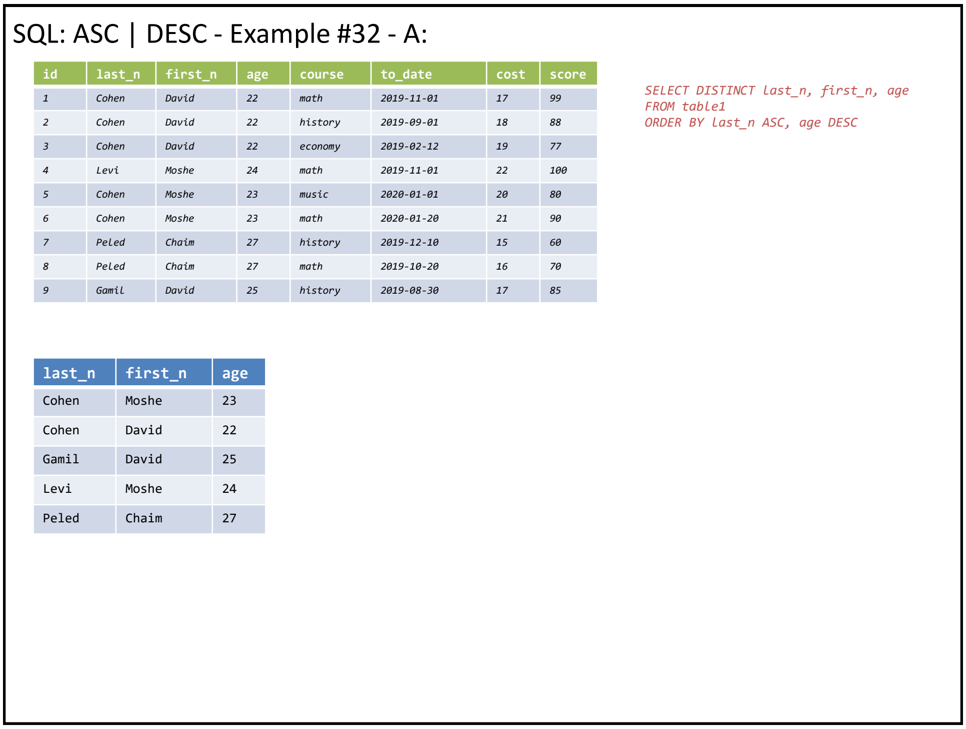
id	last_n	first_n	age	course	to_date	cost	score
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
3	Cohen	David	22	economy	2019-02-12	19	77
5	Cohen	Moshe	23	music	2020-01-01	20	80
9	Gamil	David	25	history	2019-08-30	17	85
2	Cohen	David	22	history	2019-09-01	18	88
6	Cohen	Moshe	23	math	2020-01-20	21	90
1	Cohen	David	22	math	2019-11-01	17	99
4	Levi	Moshe	24	math	2019-11-01	22	100



SQL: ASC | DESC - Example #32 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT DISTINCT last_n, first_n, age
FROM table1
ORDER BY last_n ASC, age DESC
```



SQL: ASC | DESC - Example #32 - A:

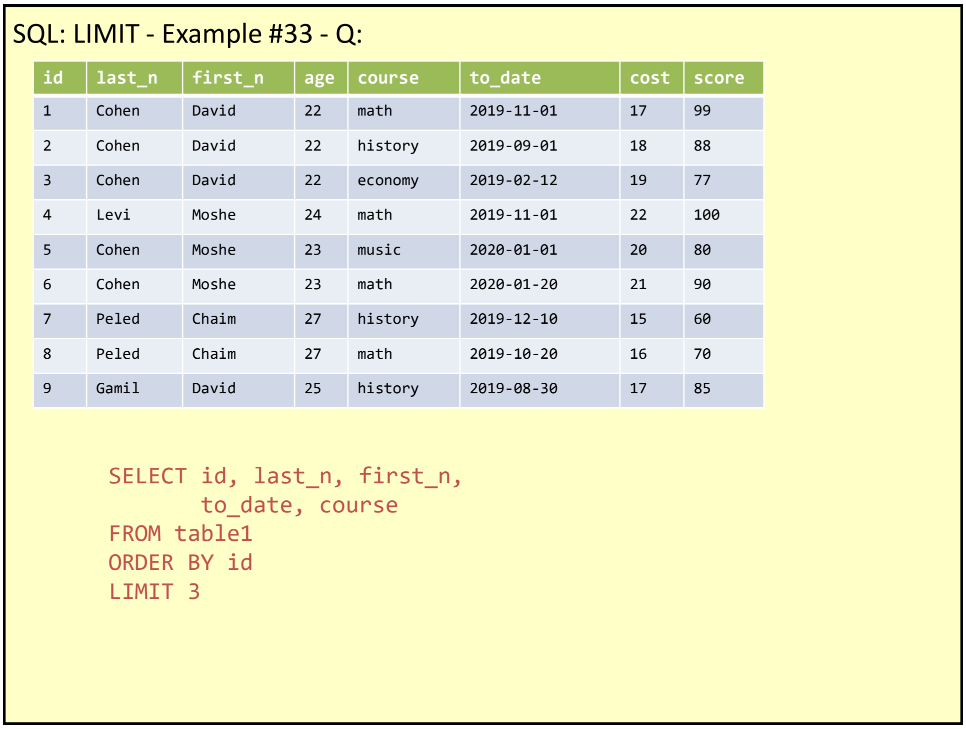
id	last_n	first_n	age	course	to_date	cost	score
1	*Cohen*	*David*	*22*	*math*	*2019-11-01*	*17*	*99*
2	*Cohen*	*David*	*22*	*history*	*2019-09-01*	*18*	*88*
3	*Cohen*	*David*	*22*	*economy*	*2019-02-12*	*19*	*77*
4	*Levi*	*Moshe*	*24*	*math*	*2019-11-01*	*22*	*100*
5	*Cohen*	*Moshe*	*23*	*music*	*2020-01-01*	*20*	*80*
6	*Cohen*	*Moshe*	*23*	*math*	*2020-01-20*	*21*	*90*
7	*Peled*	*Chaim*	*27*	*history*	*2019-12-10*	*15*	*60*
8	*Peled*	*Chaim*	*27*	*math*	*2019-10-20*	*16*	*70*
9	*Gamil*	*David*	*25*	*history*	*2019-08-30*	*17*	*85*

```
SELECT DISTINCT last_n, first_n, age
FROM table1
ORDER BY last_n ASC, age DESC
```

last_n	first_n	age
Cohen	Moshe	23
Cohen	David	22
Gamil	David	25
Levi	Moshe	24
Peled	Chaim	27

Learn SQL by Example (Chananel Perel 2023)

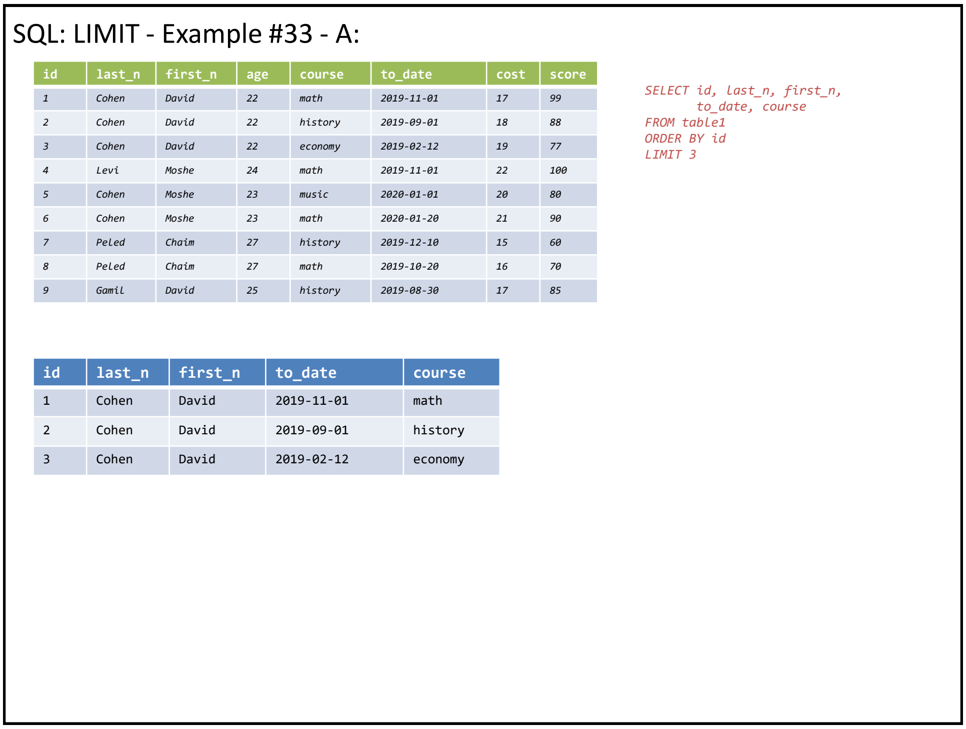
LIMIT



SQL: LIMIT - Example #33 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT id, last_n, first_n,
       to_date, course
FROM table1
ORDER BY id
LIMIT 3
```

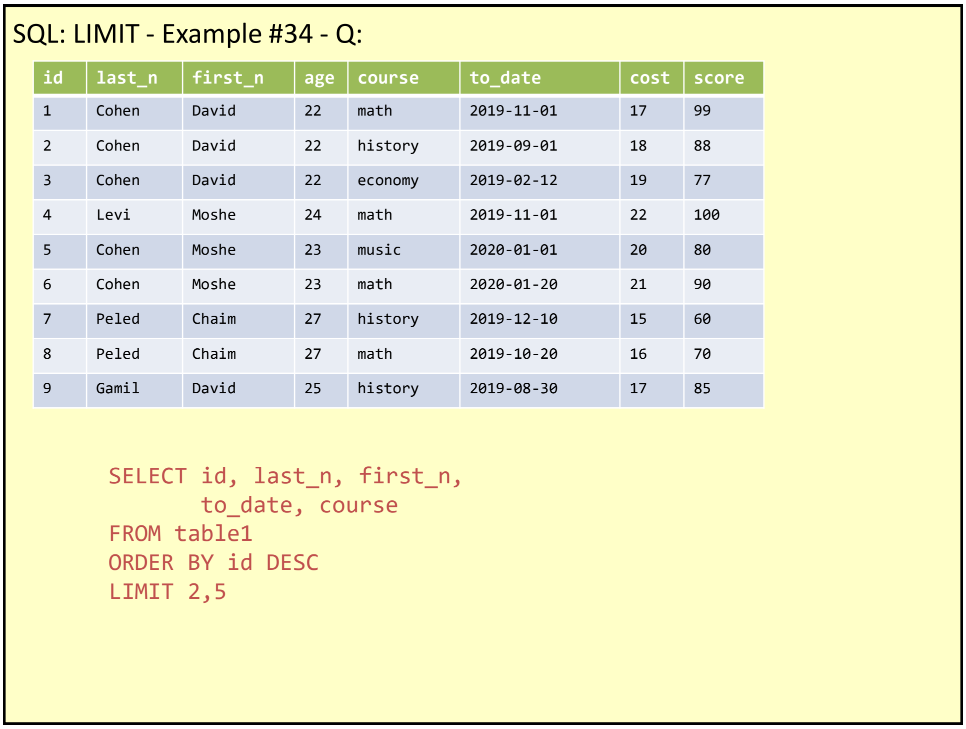


SQL: LIMIT - Example #33 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT id, last_n, first_n,
       to_date, course
FROM table1
ORDER BY id
LIMIT 3
```

id	last_n	first_n	to_date	course
1	Cohen	David	2019-11-01	math
2	Cohen	David	2019-09-01	history
3	Cohen	David	2019-02-12	economy



SQL: LIMIT - Example #34 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT id, last_n, first_n,
       to_date, course
FROM table1
ORDER BY id DESC
LIMIT 2,5
```

SQL: LIMIT - Example #34 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT id, last_n, first_n,
       to_date, course
FROM table1
ORDER BY id DESC
LIMIT 2,5
```

id	last_n	first_n	to_date	course
7	Peled	Chaim	2019-12-10	history
6	Cohen	Moshe	2020-01-20	math
5	Cohen	Moshe	2020-01-01	music
4	Levi	Moshe	2019-11-01	math
3	Cohen	David	2019-02-12	economy

Learn SQL by Example (Chananel Perel 2023)

OFFSET

SQL: OFFSET - Example #35 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT id, last_n, first_n,
       to_date, course
FROM table1
ORDER BY id
LIMIT 2 OFFSET 5
```

SQL: OFFSET - Example #35 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	*Cohen*	*David*	*22*	*math*	*2019-11-01*	*17*	*99*
2	*Cohen*	*David*	*22*	*history*	*2019-09-01*	*18*	*88*
3	*Cohen*	*David*	*22*	*economy*	*2019-02-12*	*19*	*77*
4	*Levi*	*Moshe*	*24*	*math*	*2019-11-01*	*22*	*100*
5	*Cohen*	*Moshe*	*23*	*music*	*2020-01-01*	*20*	*80*
6	*Cohen*	*Moshe*	*23*	*math*	*2020-01-20*	*21*	*90*
7	*Peled*	*Chaim*	*27*	*history*	*2019-12-10*	*15*	*60*
8	*Peled*	*Chaim*	*27*	*math*	*2019-10-20*	*16*	*70*
9	*Gamil*	*David*	*25*	*history*	*2019-08-30*	*17*	*85*

```
SELECT id, last_n, first_n,
       to_date, course
FROM table1
ORDER BY id
LIMIT 2 OFFSET 5
```

id	last_n	first_n	to_date	course
6	Cohen	Moshe	2020-01-20	math
7	Peled	Chaim	2019-12-10	history

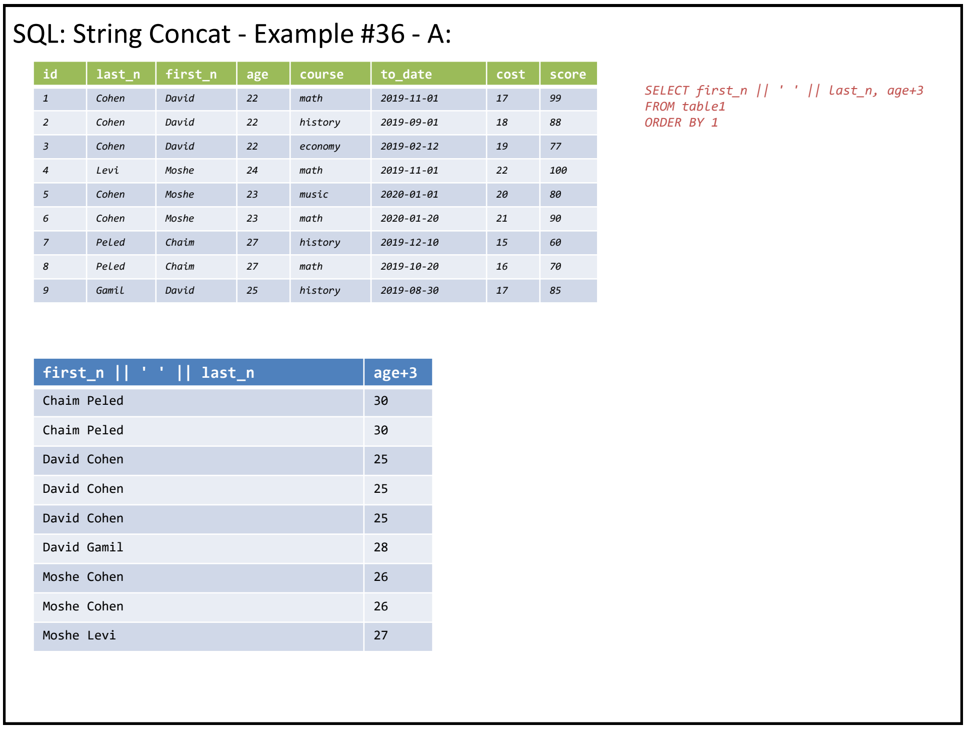
Learn SQL by Example (Chananel Perel 2023)

STRING CONCAT

SQL: String Concat - Example #36 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```sql
SELECT first_n || ' ' || last_n, age+3
FROM table1
ORDER BY 1
```

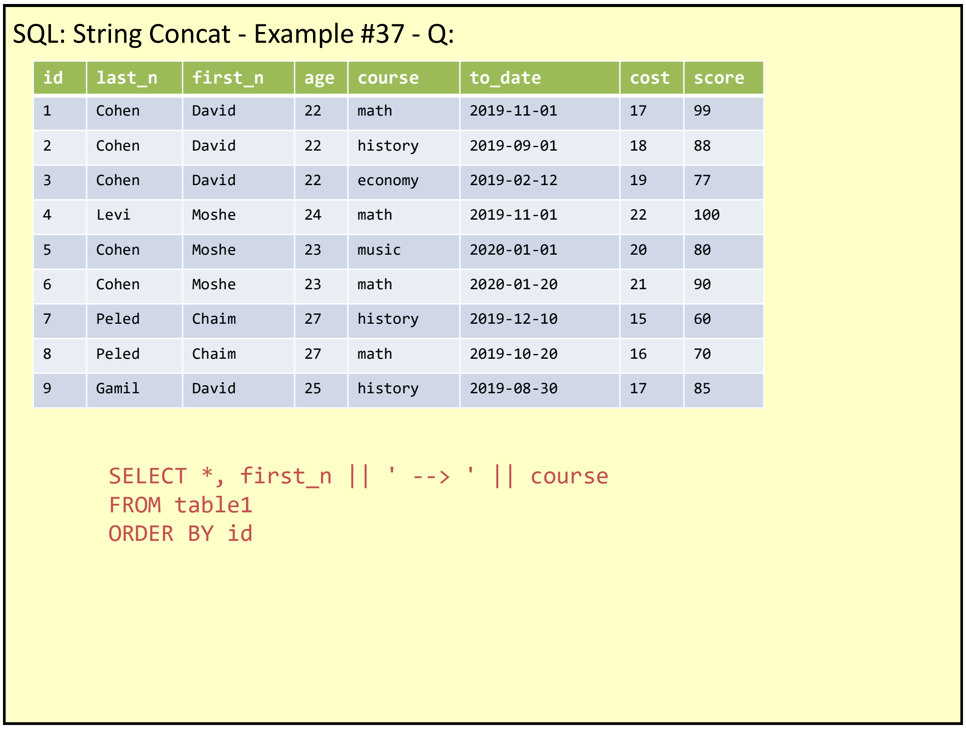


SQL: String Concat - Example #36 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT first_n || ' ' || last_n, age+3
FROM table1
ORDER BY 1
```

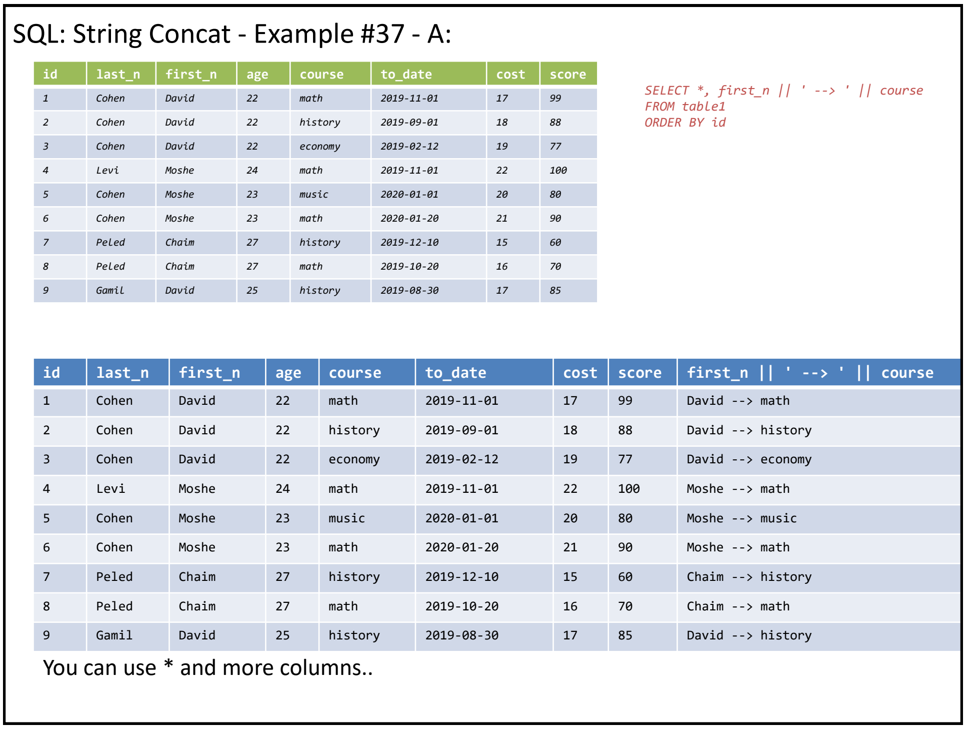
first_n || ' ' || last_n	age+3
Chaim Peled	30
Chaim Peled	30
David Cohen	25
David Cohen	25
David Cohen	25
David Gamil	28
Moshe Cohen	26
Moshe Cohen	26
Moshe Levi	27



SQL: String Concat - Example #37 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT *, first_n || ' --> ' || course
FROM table1
ORDER BY id
```



SQL: String Concat - Example #37 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

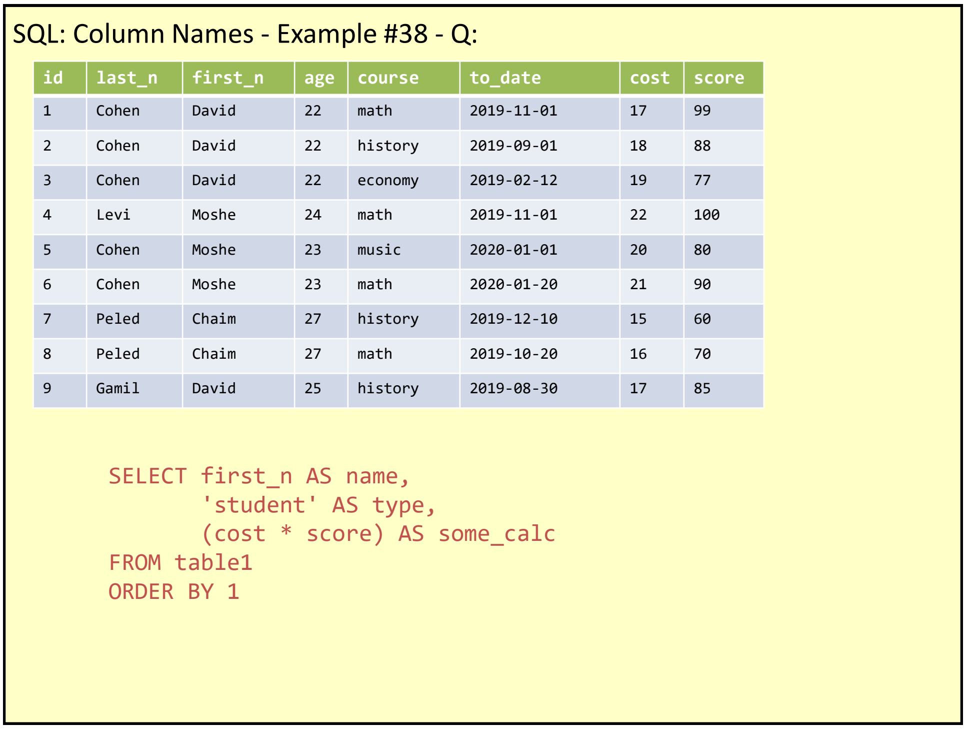
```
SELECT *, first_n || ' --> ' || course
FROM table1
ORDER BY id
```

id	last_n	first_n	age	course	to_date	cost	score	first_n || ' --> ' || course
1	Cohen	David	22	math	2019-11-01	17	99	David --> math
2	Cohen	David	22	history	2019-09-01	18	88	David --> history
3	Cohen	David	22	economy	2019-02-12	19	77	David --> economy
4	Levi	Moshe	24	math	2019-11-01	22	100	Moshe --> math
5	Cohen	Moshe	23	music	2020-01-01	20	80	Moshe --> music
6	Cohen	Moshe	23	math	2020-01-20	21	90	Moshe --> math
7	Peled	Chaim	27	history	2019-12-10	15	60	Chaim --> history
8	Peled	Chaim	27	math	2019-10-20	16	70	Chaim --> math
9	Gamil	David	25	history	2019-08-30	17	85	David --> history

You can use * and more columns..

Learn SQL by Example (Chananel Perel 2023)

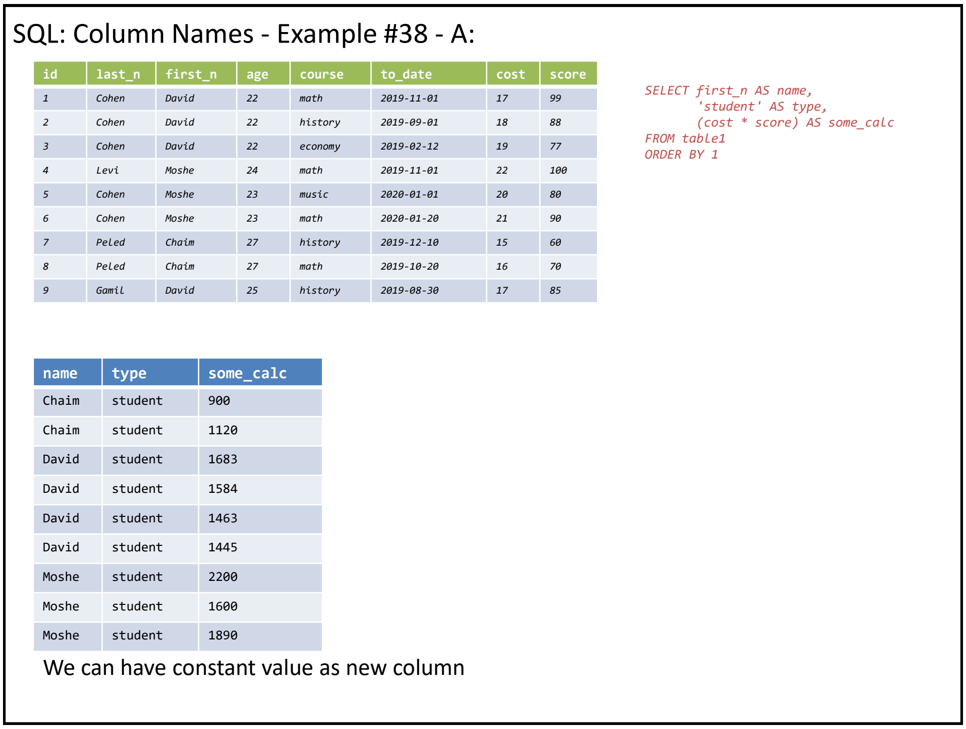
COLUMN NAMES



SQL: Column Names - Example #38 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT first_n AS name,
       'student' AS type,
       (cost * score) AS some_calc
FROM table1
ORDER BY 1
```



SQL: Column Names - Example #38 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT first_n AS name,
       'student' AS type,
       (cost * score) AS some_calc
FROM table1
ORDER BY 1
```

name	type	some_calc
Chaim	student	900
Chaim	student	1120
David	student	1683
David	student	1584
David	student	1463
David	student	1445
Moshe	student	2200
Moshe	student	1600
Moshe	student	1890

We can have constant value as new column

SQL: Column Names - Example #39 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT first_n name,
       17 points,
       (cost * 2) "double cost"
FROM table1
ORDER BY 1
```

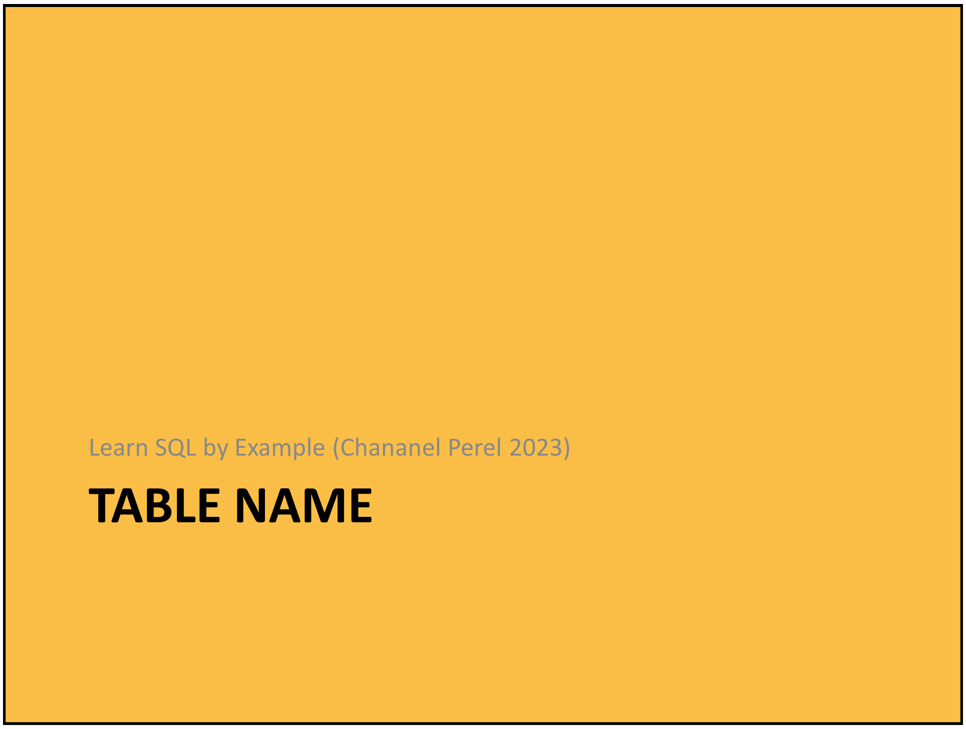
SQL: Column Names - Example #39 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT first_n name,
       17 points,
       (cost * 2) "double cost"
FROM table1
ORDER BY 1
```

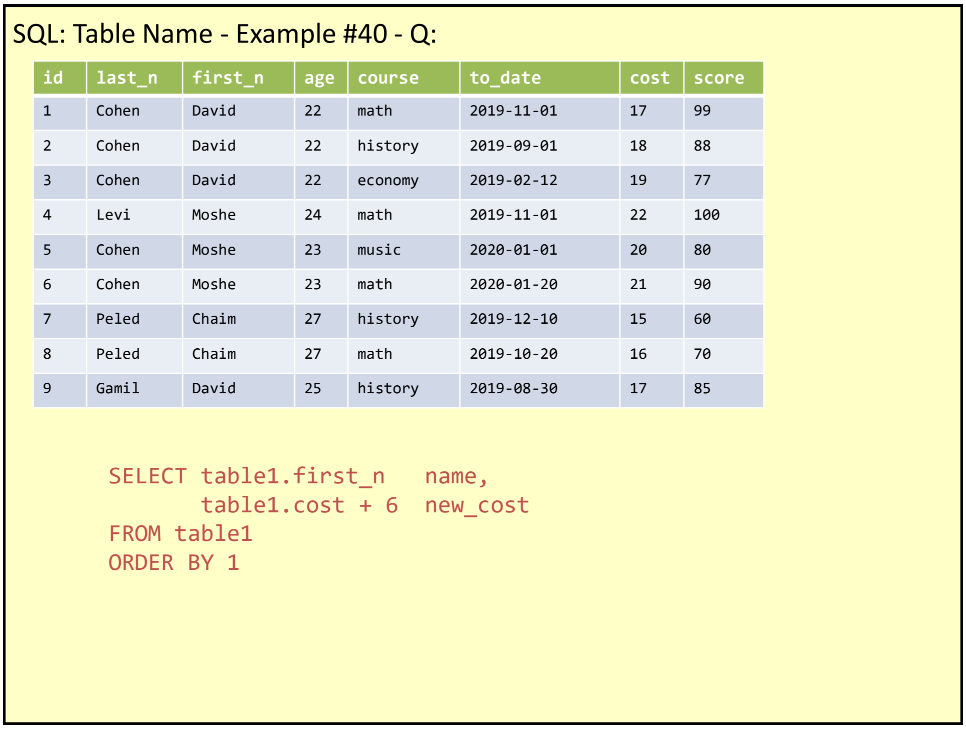
name	points	double cost
Chaim	17	30
Chaim	17	32
David	17	34
David	17	36
David	17	38
David	17	34
Moshe	17	44
Moshe	17	40
Moshe	17	42

We do not need the AS..
We can put space in column name, but do NOT do it!



Learn SQL by Example (Chananel Perel 2023)

TABLE NAME



SQL: Table Name - Example #40 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT table1.first_n   name,
       table1.cost + 6  new_cost
FROM table1
ORDER BY 1
```

SQL: Table Name - Example #40 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

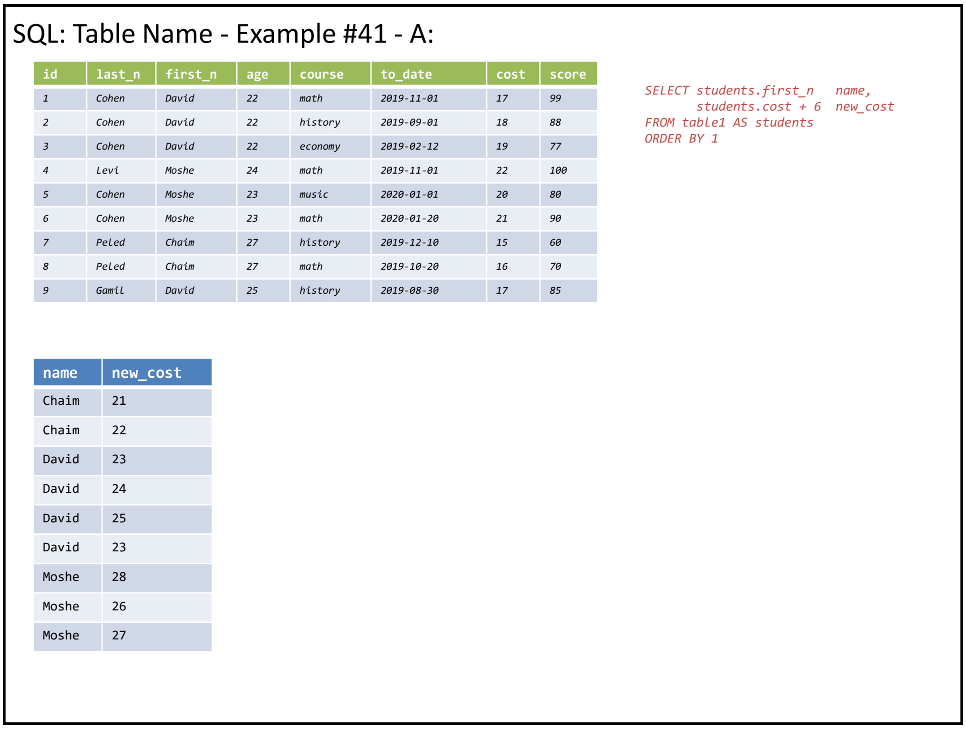
```
SELECT table1.first_n   name,
       table1.cost + 6  new_cost
FROM table1
ORDER BY 1
```

name	new_cost
Chaim	21
Chaim	22
David	23
David	24
David	25
David	23
Moshe	28
Moshe	26
Moshe	27

SQL: Table Name - Example #41 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT students.first_n   name,
       students.cost + 6  new_cost
FROM table1 AS students
ORDER BY 1
```

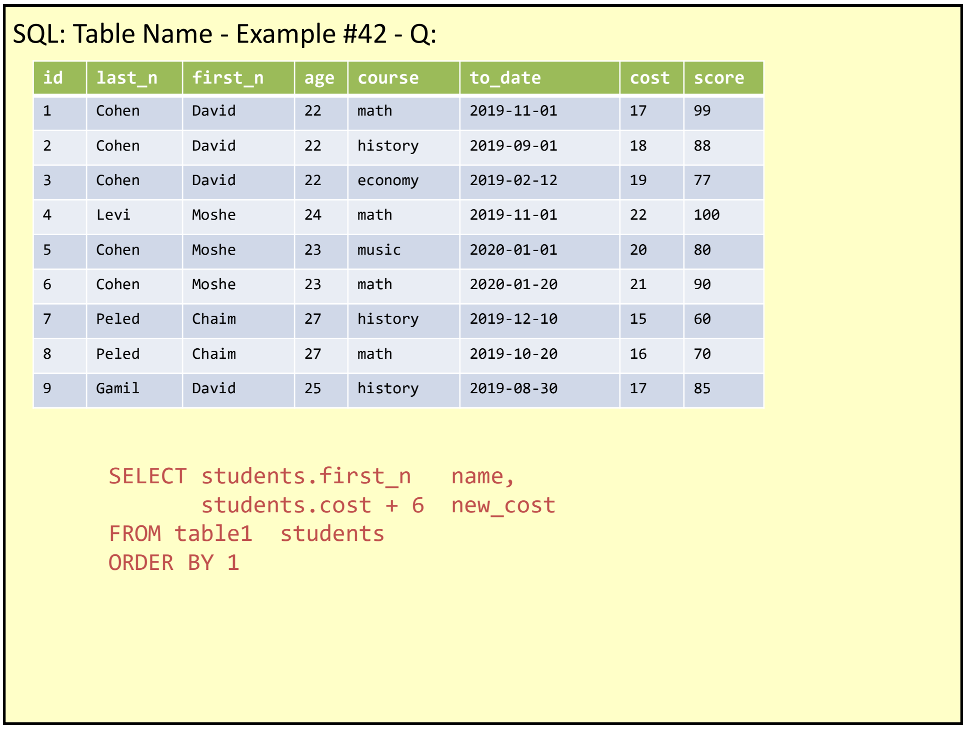


SQL: Table Name - Example #41 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT students.first_n   name,
       students.cost + 6  new_cost
FROM table1 AS students
ORDER BY 1
```

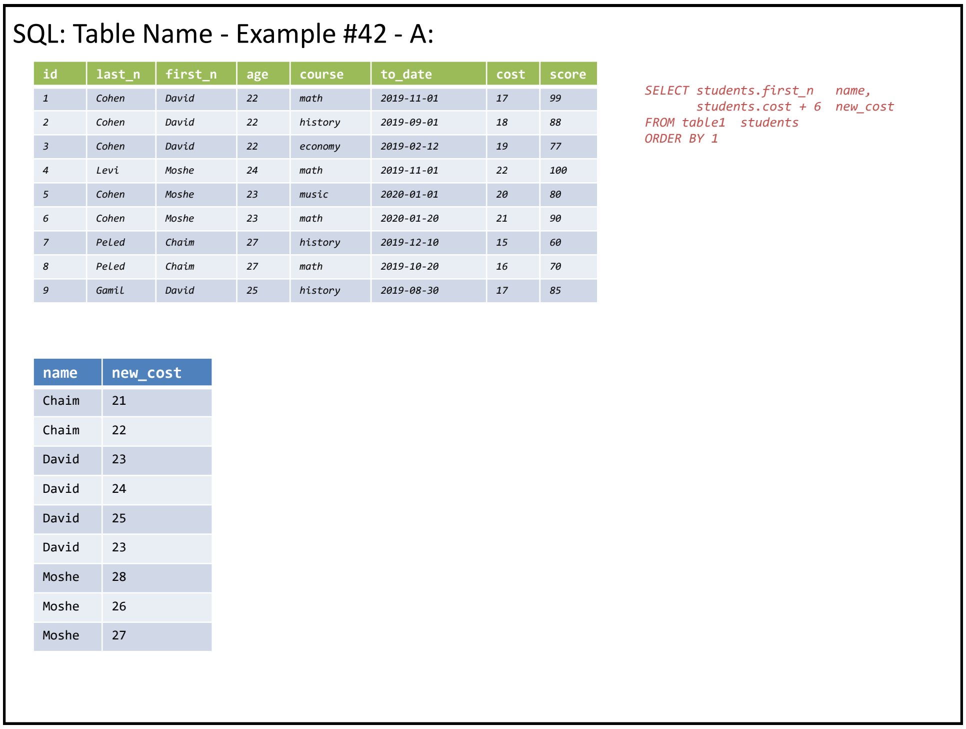
name	new_cost
Chaim	21
Chaim	22
David	23
David	24
David	25
David	23
Moshe	28
Moshe	26
Moshe	27



SQL: Table Name - Example #42 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT students.first_n   name,
       students.cost + 6  new_cost
FROM table1  students
ORDER BY 1
```



SQL: Table Name - Example #42 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT students.first_n   name,
       students.cost + 6  new_cost
FROM table1  students
ORDER BY 1
```

name	new_cost
Chaim	21
Chaim	22
David	23
David	24
David	25
David	23
Moshe	28
Moshe	26
Moshe	27

Learn SQL by Example (Chananel Perel 2023)

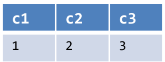
COMMENT

SQL: Comment - Example #43 - Q:

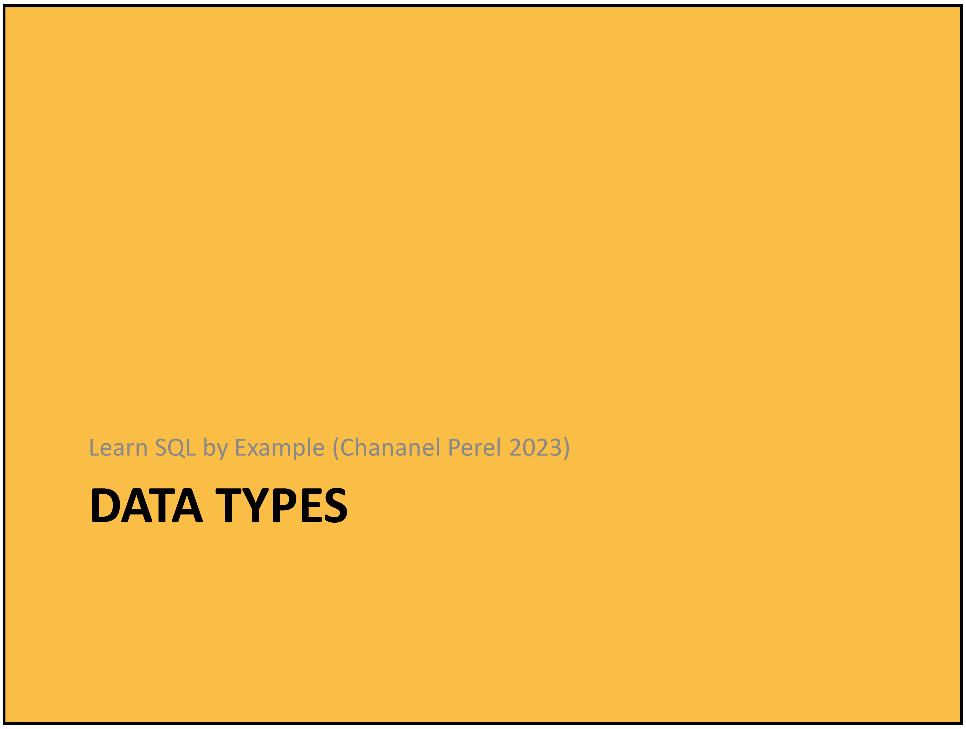
```
SELECT 1 c1, -- comment 1
       2 c2, -- comment 2
       3 c3  /* and also this
                is supported.. */
```

SQL: Comment - Example #43 - A:

```
SELECT 1 c1, -- comment 1
       2 c2, -- comment 2
       3 c3  /* and also this
                is supported.. */
```

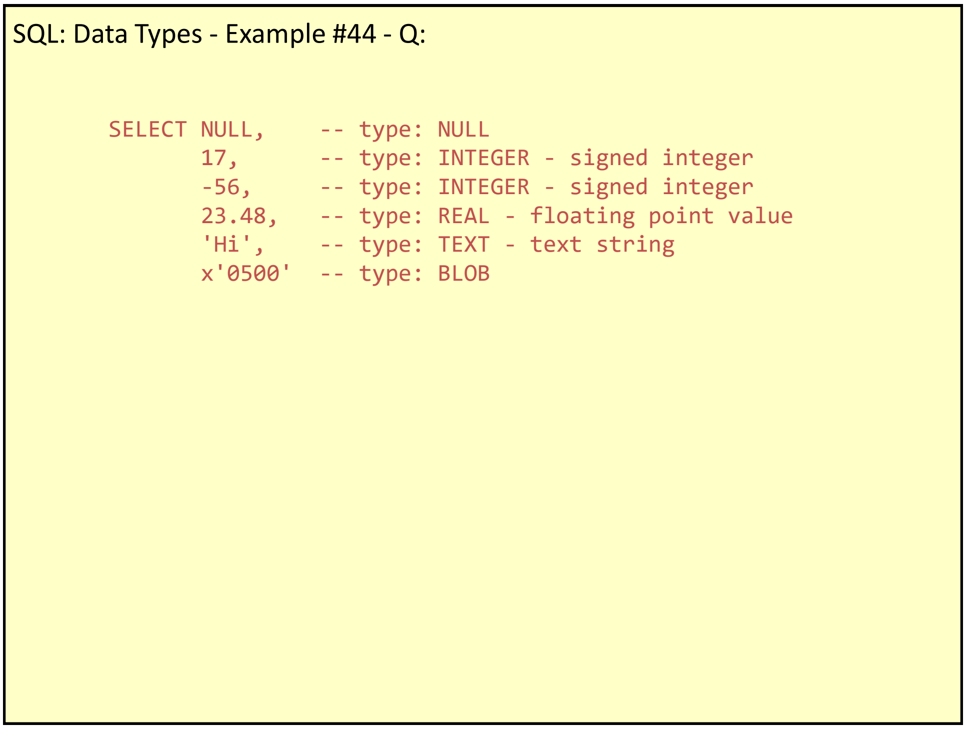


c1	c2	c3
1	2	3



Learn SQL by Example (Chananel Perel 2023)

DATA TYPES

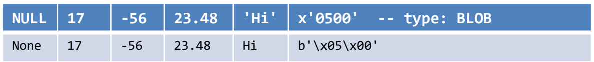


SQL: Data Types - Example #44 - Q:

```
SELECT NULL,    -- type: NULL
       17,      -- type: INTEGER - signed integer
       -56,     -- type: INTEGER - signed integer
       23.48,   -- type: REAL - floating point value
       'Hi',    -- type: TEXT - text string
       x'0500'  -- type: BLOB
```

SQL: Data Types - Example #44 - A:

```
SELECT NULL,    -- type: NULL
       17,      -- type: INTEGER - signed integer
       -56,     -- type: INTEGER - signed integer
       23.48,   -- type: REAL - floating point value
       'Hi',    -- type: TEXT - text string
       x'0500'  -- type: BLOB
```



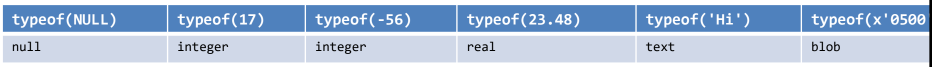
NULL	17	-56	23.48	'Hi'	x'0500' -- type: BLOB
None	17	-56	23.48	Hi	b'\x05\x00'

SQL: Data Types - Example #45 - Q:

```
SELECT typeof(NULL),
       typeof(17),
       typeof(-56),
       typeof(23.48),
       typeof('Hi'),
       typeof(x'0500')
```

SQL: Data Types - Example #45 - A:

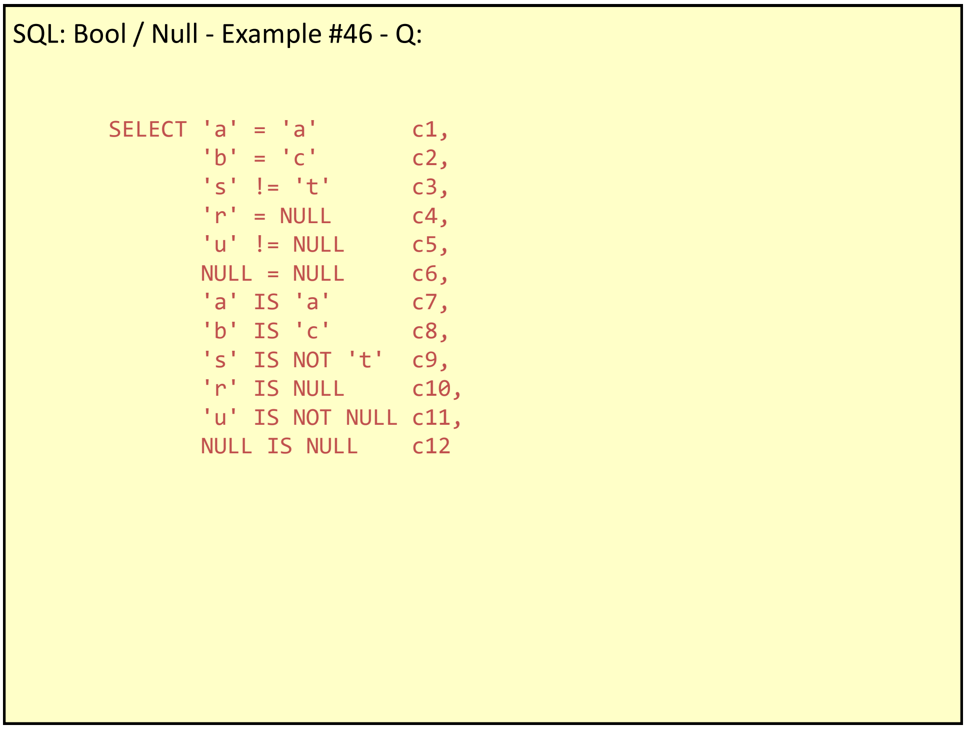
```
SELECT typeof(NULL),
       typeof(17),
       typeof(-56),
       typeof(23.48),
       typeof('Hi'),
       typeof(x'0500')
```



typeof(NULL)	typeof(17)	typeof(-56)	typeof(23.48)	typeof('Hi')	typeof(x'0500
null	integer	integer	real	text	blob

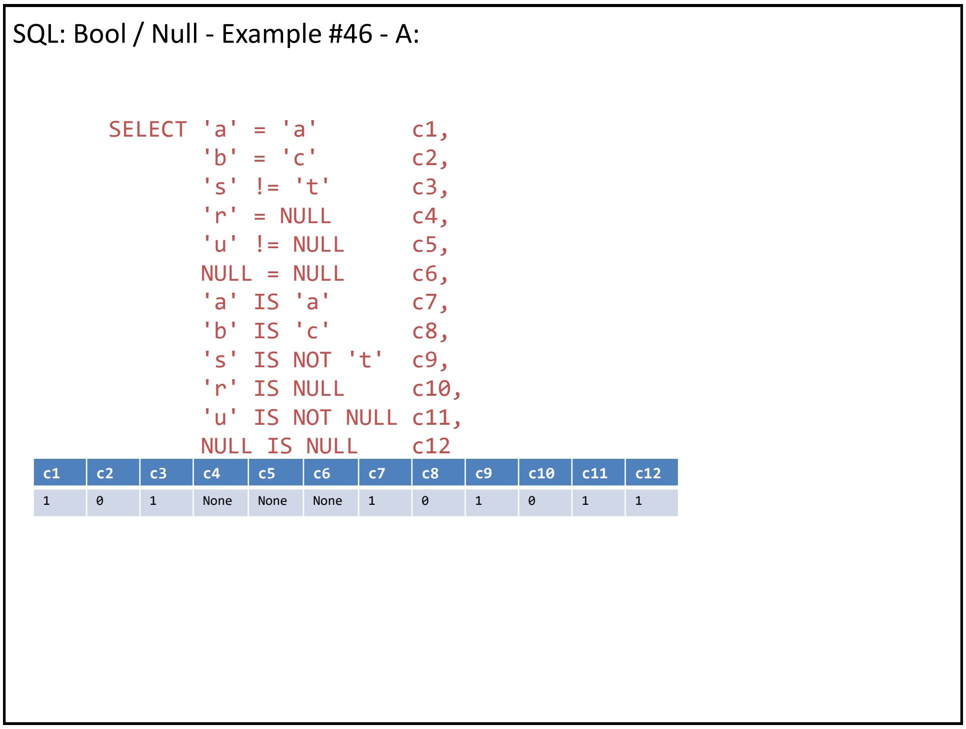
Learn SQL by Example (Chananel Perel 2023)

BOOL / NULL



SQL: Bool / Null - Example #46 - Q:

```
SELECT 'a' = 'a'       c1,
       'b' = 'c'       c2,
       's' != 't'      c3,
       'r' = NULL      c4,
       'u' != NULL     c5,
       NULL = NULL     c6,
       'a' IS 'a'      c7,
       'b' IS 'c'      c8,
       's' IS NOT 't'  c9,
       'r' IS NULL     c10,
       'u' IS NOT NULL c11,
       NULL IS NULL    c12
```



SQL: Bool / Null - Example #46 - A:

```
SELECT 'a' = 'a'       c1,
       'b' = 'c'       c2,
       's' != 't'      c3,
       'r' = NULL      c4,
       'u' != NULL     c5,
       NULL = NULL     c6,
       'a' IS 'a'      c7,
       'b' IS 'c'      c8,
       's' IS NOT 't'  c9,
       'r' IS NULL     c10,
       'u' IS NOT NULL c11,
       NULL IS NULL    c12
```

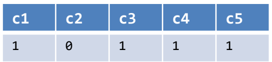
c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12
1	0	1	None	None	None	1	0	1	0	1	1

SQL: Bool / Null - Example #47 - Q:

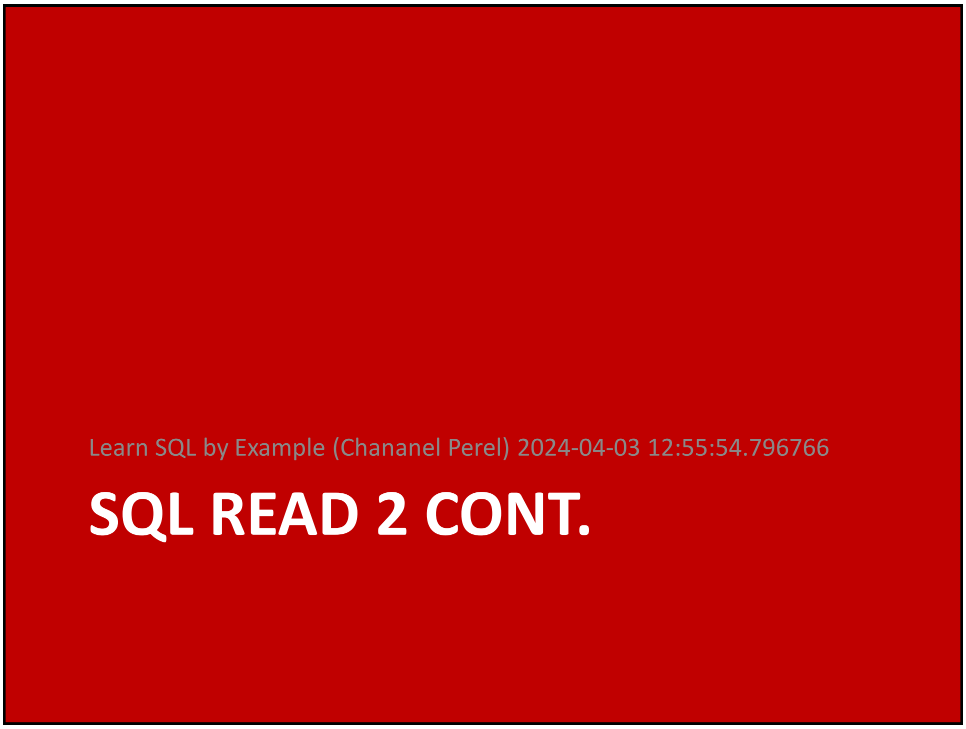
```
SELECT 1 = 1      c1,
       1 = 7      c2,
       1 != 7     c3,
       1 <> 7     c4,
       NOT 1 = 7  c5
```

SQL: Bool / Null - Example #47 - A:

```
SELECT 1 = 1      c1,
       1 = 7      c2,
       1 != 7     c3,
       1 <> 7     c4,
       NOT 1 = 7  c5
```



c1	c2	c3	c4	c5
1	0	1	1	1



Learn SQL by Example (Chananel Perel) 2024-04-03 12:55:54.796766

SQL READ 2 CONT.

Learn SQL by Example (Chananel Perel 2023)

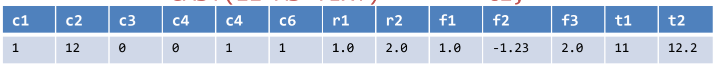
CAST AS

SQL: CAST AS - Example #48 - Q:

```
SELECT CAST('1' AS INTEGER)    c1,
       CAST('12ab' AS INTEGER) c2,
       CAST('cd34' AS INTEGER) c3,
       CAST('ef' AS INTEGER)   c4,
       CAST(1.12 AS INTEGER)   c4,
       CAST(1.99 AS INTEGER)   c6,
       ROUND(1.12)             r1,
       ROUND(1.99)             r2,
       CAST('1' AS REAL)       f1,
       CAST('-1.23' AS REAL)   f2,
       CAST(2 AS REAL)         f3,
       CAST(11 AS TEXT)        t1,
       CAST(12.2 AS TEXT)      t2
```

SQL: CAST AS - Example #48 - A:

```
SELECT CAST('1' AS INTEGER)    c1,
       CAST('12ab' AS INTEGER) c2,
       CAST('cd34' AS INTEGER) c3,
       CAST('ef' AS INTEGER)   c4,
       CAST(1.12 AS INTEGER)   c4,
       CAST(1.99 AS INTEGER)   c6,
       ROUND(1.12)             r1,
       ROUND(1.99)             r2,
       CAST('1' AS REAL)       f1,
       CAST('-1.23' AS REAL)   f2,
       CAST(2 AS REAL)         f3,
       CAST(11 AS TEXT)        t1,
```



c1	c2	c3	c4	c4	c6	r1	r2	f1	f2	f3	t1	t2
1	12	0	0	1	1	1.0	2.0	1.0	-1.23	2.0	11	12.2

Learn SQL by Example (Chananel Perel 2023)

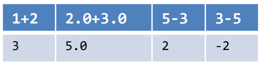
MATH OPERATORS

SQL: Math Operators - Example #49 - Q:

```
SELECT 1+2,
       2.0+3.0,
       5-3,
       3-5
```

SQL: Math Operators - Example #49 - A:

```
SELECT 1+2,
       2.0+3.0,
       5-3,
       3-5
```



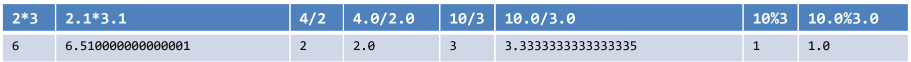
1+2	2.0+3.0	5-3	3-5
3	5.0	2	-2

SQL: Math Operators - Example #50 - Q:

```
SELECT 2*3,
       2.1*3.1,
       4/2,
       4.0/2.0,
       10/3,
       10.0/3.0,
       10%3,
       10.0%3.0
```

SQL: Math Operators - Example #50 - A:

```
SELECT 2*3,
       2.1*3.1,
       4/2,
       4.0/2.0,
       10/3,
       10.0/3.0,
       10%3,
       10.0%3.0
```



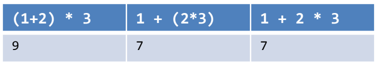
2*3	2.1*3.1	4/2	4.0/2.0	10/3	10.0/3.0	10%3	10.0%3.0
6	6.510000000000001	2	2.0	3	3.3333333333333335	1	1.0

SQL: Math Operators - Example #51 - Q:

```
SELECT (1+2) * 3,
       1 + (2*3),
       1 + 2 * 3
```

SQL: Math Operators - Example #51 - A:

```
SELECT (1+2) * 3,
       1 + (2*3),
       1 + 2 * 3
```



(1+2) * 3	1 + (2*3)	1 + 2 * 3
9	7	7

Better to always put () to make sure everyone understand what was done.

Learn SQL by Example (Chananel Perel 2023)

BUILT-IN FUNCTIONS

SQL: Built-In Functions - Example #52 - Q:

```
SELECT abs(-3)                        abs_,
       3/0                            div0_,
       hex('hello')                   hex_,
       random()                       rnd_,
       round(1.2938485, 3)            round_,
       coalesce(4/0,NULL,'ZZ','YY')   coal_
```

SQL: Built-In Functions - Example #52 - A:

```
SELECT abs(-3)                        abs_,
       3/0                            div0_,
       hex('hello')                   hex_,
       random()                       rnd_,
       round(1.2938485, 3)            round_,
       coalesce(4/0,NULL,'ZZ','YY')   coal_
```

abs_	div0_	hex_	rnd_	round_	coal_
3	None	68656C6C6F	8024669380428803438	1.294	ZZ

SQL: Built-In Functions - Example #53 - Q:

name	age	birth
David	22	2000
Sara	12	None
Moshe	None	2018
Chaim	None	None

```
SELECT name,
       coalesce(age, 2022 - birth, 'NA') as fixed_age
FROM children
```

SQL: Built-In Functions - Example #53 - A:

name	age	birth
David	*22*	*2000*
Sara	*12*	*None*
Moshe	*None*	*2018*
Chaim	*None*	*None*

```
SELECT name,
       coalesce(age, 2022 - birth, 'NA') as fixed_age
FROM children
```

name	fixed_age
David	22
Sara	12
Moshe	4
Chaim	NA

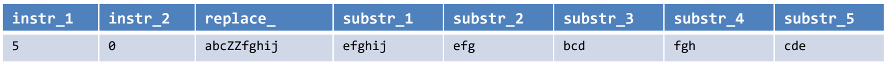
Example for using coalesce

SQL: Built-In Functions - Example #54 - Q:

```
SELECT instr('abcdefghij', 'ef')        instr_1,
       instr('abcdefghij', 'm')         instr_2,
       replace('abcdefghij','de','ZZ')  replace_,
       substr('abcdefghij', 5)          substr_1,
       substr('abcdefghij', 5,3)        substr_2,
       substr('abcdefghij', 5,-3)       substr_3,
       substr('abcdefghij',-5,3)        substr_4,
       substr('abcdefghij',-5,-3)       substr_5
```

SQL: Built-In Functions - Example #54 - A:

```
SELECT instr('abcdefghij', 'ef')        instr_1,
       instr('abcdefghij', 'm')         instr_2,
       replace('abcdefghij','de','ZZ')  replace_,
       substr('abcdefghij', 5)          substr_1,
       substr('abcdefghij', 5,3)        substr_2,
       substr('abcdefghij', 5,-3)       substr_3,
       substr('abcdefghij',-5,3)        substr_4,
       substr('abcdefghij',-5,-3)       substr_5
```



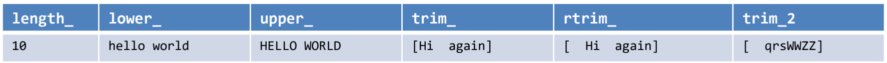
instr_1	instr_2	replace_	substr_1	substr_2	substr_3	substr_4	substr_5
5	0	abcZZfghij	efghij	efg	bcd	fgh	cde

SQL: Built-In Functions - Example #55 - Q:

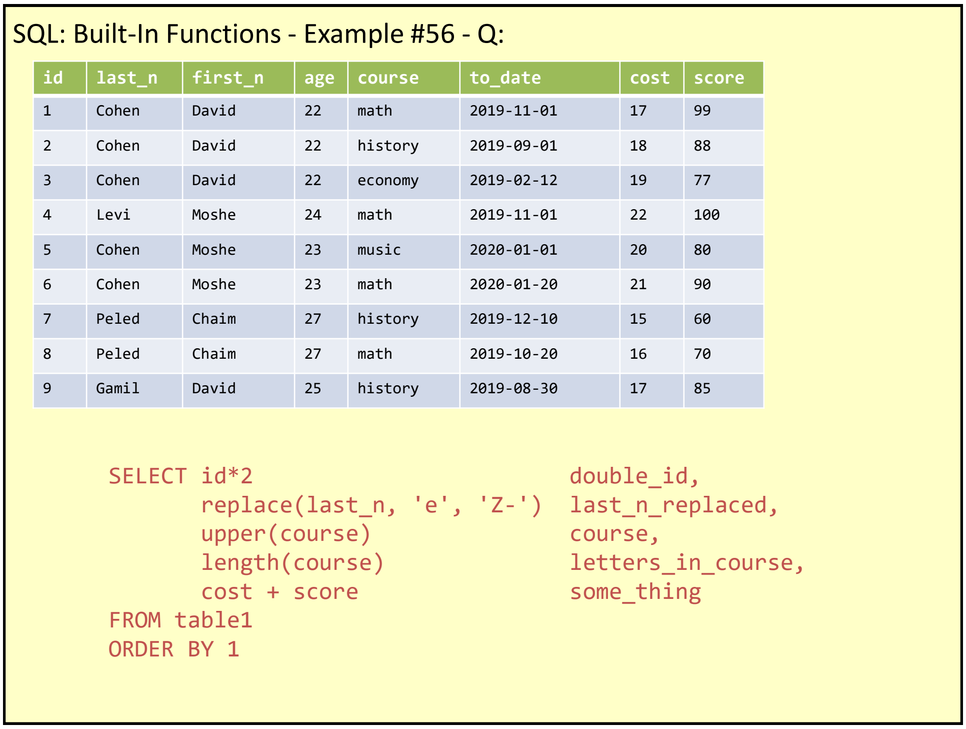
```
SELECT
   length('abcdefghij')                     length_,
   lower('Hello worLD')                     lower_,
   upper('Hello worLD')                     upper_,
   '['|| trim ('  Hi  again  ') ||']'       trim_,
   '['|| rtrim('  Hi  again  ') ||']'       rtrim_,
   '['|| trim ('  qrsWWZZqrs','qrs') ||']'  trim_2
```

SQL: Built-In Functions - Example #55 - A:

```
SELECT
   length('abcdefghij')                     length_,
   lower('Hello worLD')                     lower_,
   upper('Hello worLD')                     upper_,
   '['|| trim ('  Hi  again  ') ||']'       trim_,
   '['|| rtrim('  Hi  again  ') ||']'       rtrim_,
   '['|| trim ('  qrsWWZZqrs','qrs') ||']'  trim_2
```



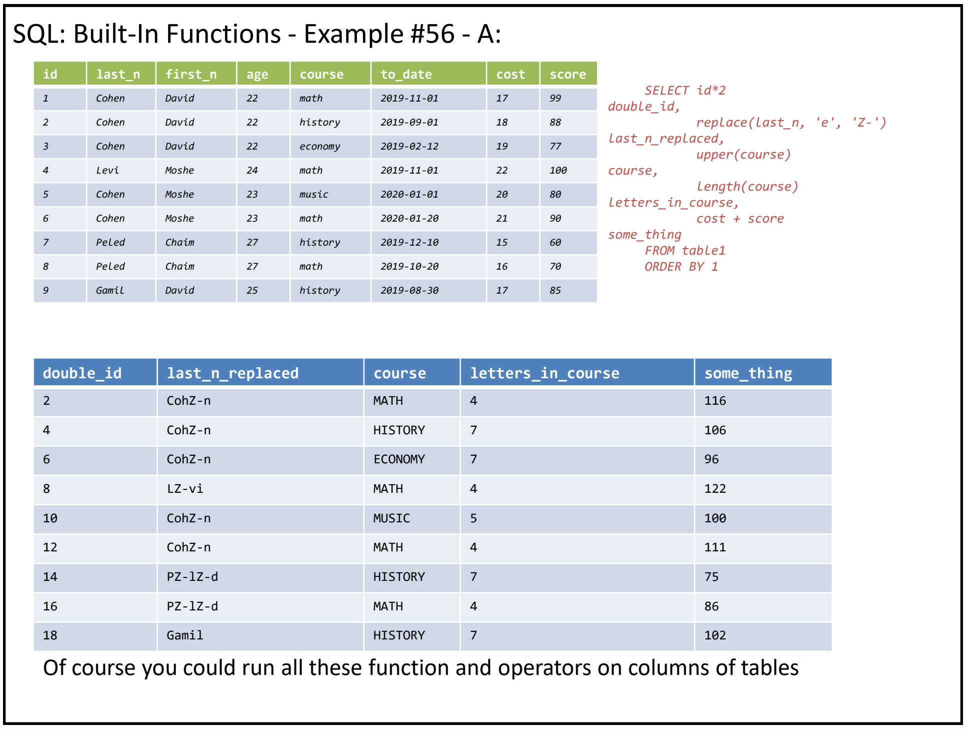
length_	lower_	upper_	trim_	rtrim_	trim_2
10	hello world	HELLO WORLD	[Hi again]	[Hi again]	[qrsWWZZ]



SQL: Built-In Functions - Example #56 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT id*2                            double_id,
       replace(last_n, 'e', 'Z-')      last_n_replaced,
       upper(course)                   course,
       length(course)                  letters_in_course,
       cost + score                    some_thing
FROM table1
ORDER BY 1
```



SQL: Built-In Functions - Example #56 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
      SELECT id*2
double_id,
             replace(last_n, 'e', 'Z-')
last_n_replaced,
             upper(course)
course,
             length(course)
letters_in_course,
             cost + score
some_thing
      FROM table1
      ORDER BY 1
```

double_id	last_n_replaced	course	letters_in_course	some_thing
2	CohZ-n	MATH	4	116
4	CohZ-n	HISTORY	7	106
6	CohZ-n	ECONOMY	7	96
8	LZ-vi	MATH	4	122
10	CohZ-n	MUSIC	5	100
12	CohZ-n	MATH	4	111
14	PZ-lZ-d	HISTORY	7	75
16	PZ-lZ-d	MATH	4	86
18	Gamil	HISTORY	7	102

Of course you could run all these function and operators on columns of tables

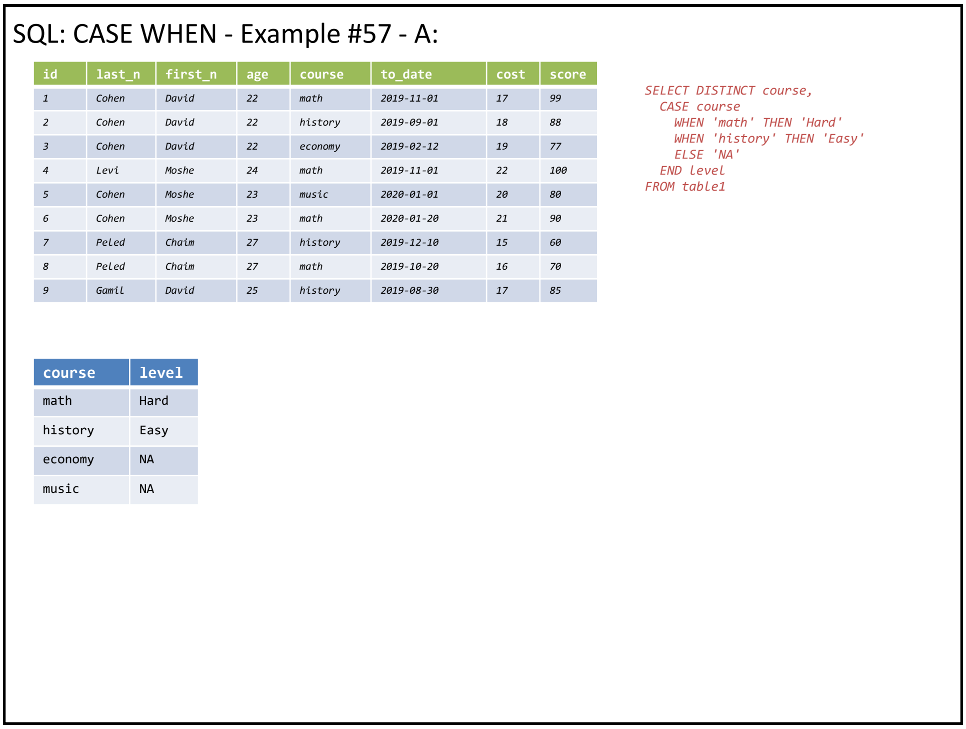
Learn SQL by Example (Chananel Perel 2023)

CASE WHEN

SQL: CASE WHEN - Example #57 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT DISTINCT course,
  CASE course
    WHEN 'math' THEN 'Hard'
    WHEN 'history' THEN 'Easy'
    ELSE 'NA'
  END level
FROM table1
```

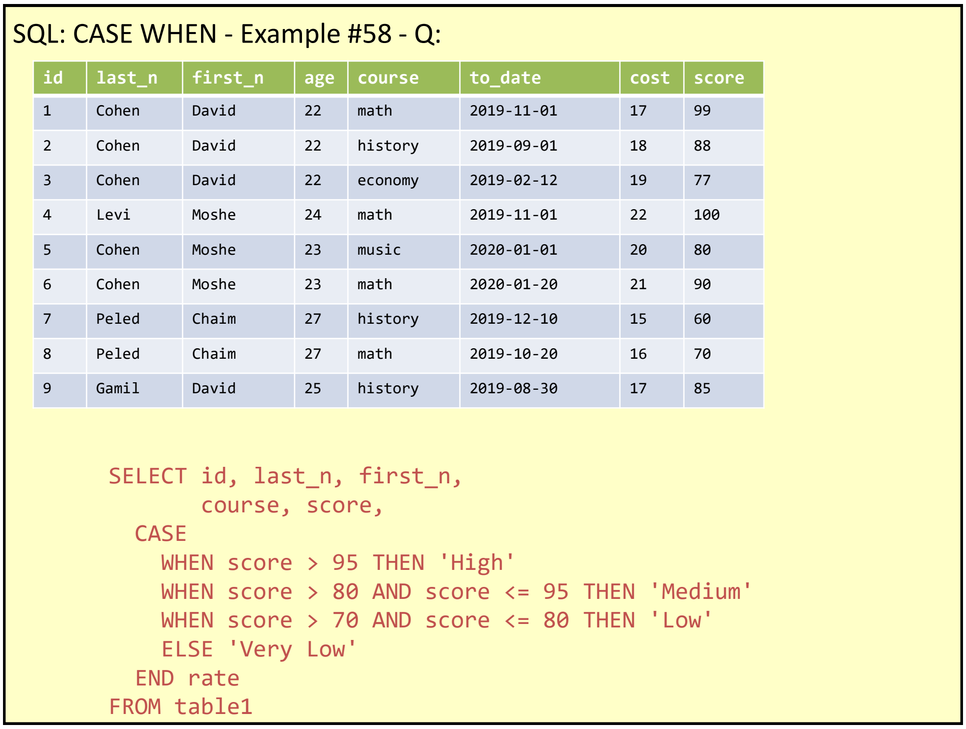


SQL: CASE WHEN - Example #57 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT DISTINCT course,
  CASE course
    WHEN 'math' THEN 'Hard'
    WHEN 'history' THEN 'Easy'
    ELSE 'NA'
  END level
FROM table1
```

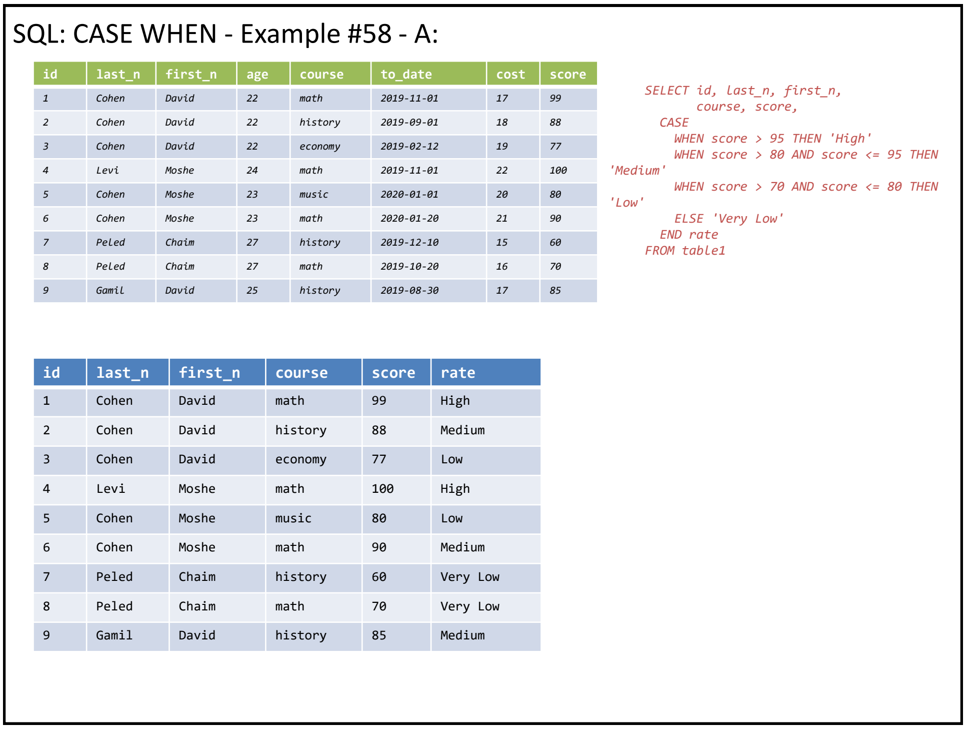
course	level
math	Hard
history	Easy
economy	NA
music	NA



SQL: CASE WHEN - Example #58 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
SELECT id, last_n, first_n,
       course, score,
  CASE
    WHEN score > 95 THEN 'High'
    WHEN score > 80 AND score <= 95 THEN 'Medium'
    WHEN score > 70 AND score <= 80 THEN 'Low'
    ELSE 'Very Low'
  END rate
FROM table1
```



SQL: CASE WHEN - Example #58 - A:

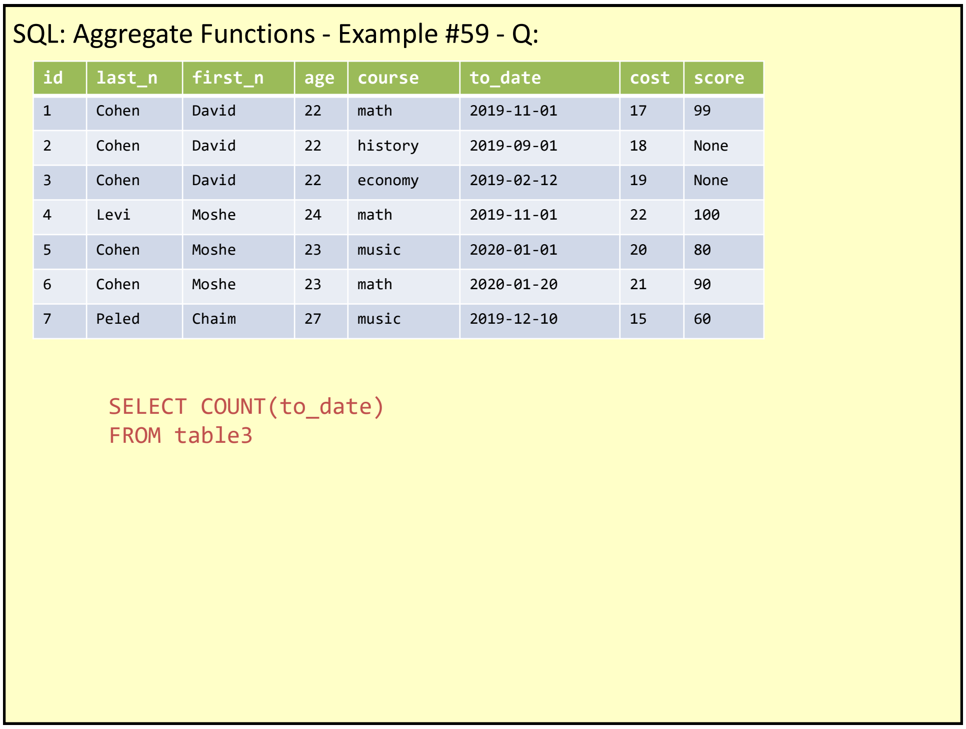
id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	88
3	Cohen	David	22	economy	2019-02-12	19	77
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	history	2019-12-10	15	60
8	Peled	Chaim	27	math	2019-10-20	16	70
9	Gamil	David	25	history	2019-08-30	17	85

```
    SELECT id, last_n, first_n,
           course, score,
      CASE
        WHEN score > 95 THEN 'High'
        WHEN score > 80 AND score <= 95 THEN
'Medium'
        WHEN score > 70 AND score <= 80 THEN
'Low'
        ELSE 'Very Low'
      END rate
    FROM table1
```

id	last_n	first_n	course	score	rate
1	Cohen	David	math	99	High
2	Cohen	David	history	88	Medium
3	Cohen	David	economy	77	Low
4	Levi	Moshe	math	100	High
5	Cohen	Moshe	music	80	Low
6	Cohen	Moshe	math	90	Medium
7	Peled	Chaim	history	60	Very Low
8	Peled	Chaim	math	70	Very Low
9	Gamil	David	history	85	Medium

Learn SQL by Example (Chananel Perel 2023)

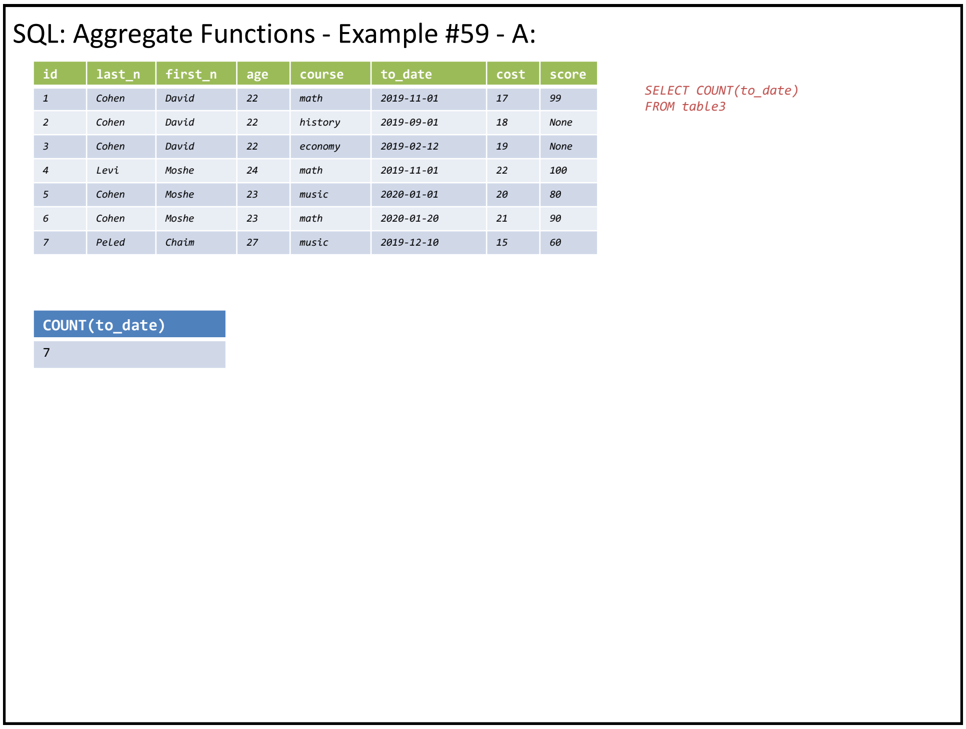
AGGREGATE FUNCTIONS



SQL: Aggregate Functions - Example #59 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT COUNT(to_date)
FROM table3
```



SQL: Aggregate Functions - Example #59 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT COUNT(to_date)
FROM table3
```

COUNT(to_date)
7

SQL: Aggregate Functions - Example #60 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

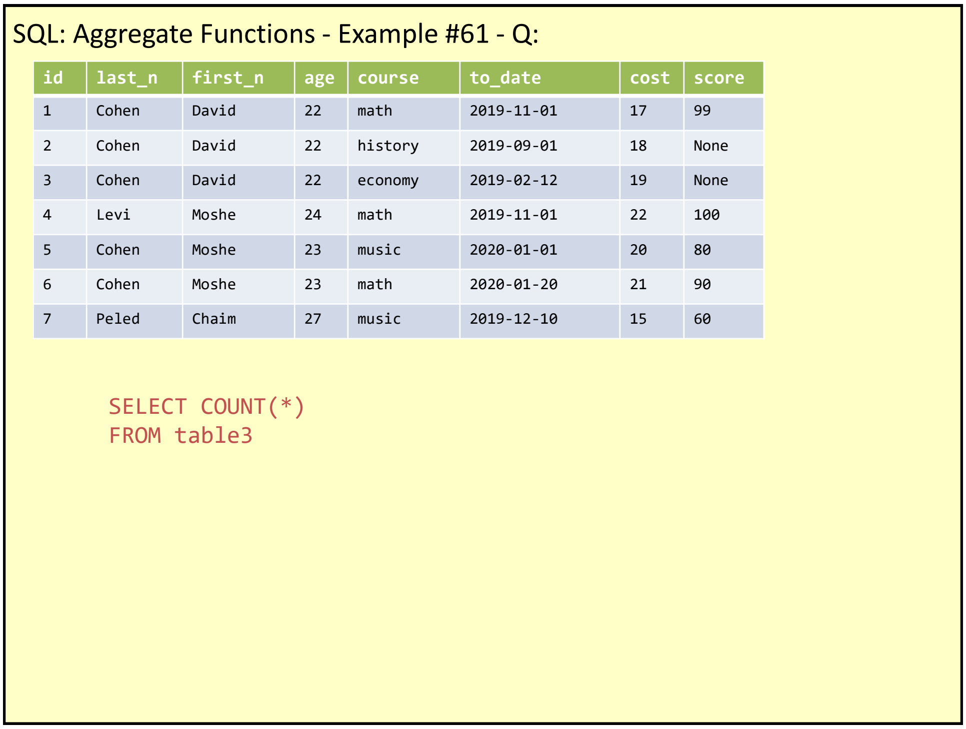
```
SELECT COUNT(score)
FROM table3
```

SQL: Aggregate Functions - Example #60 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT COUNT(score)
FROM table3
```

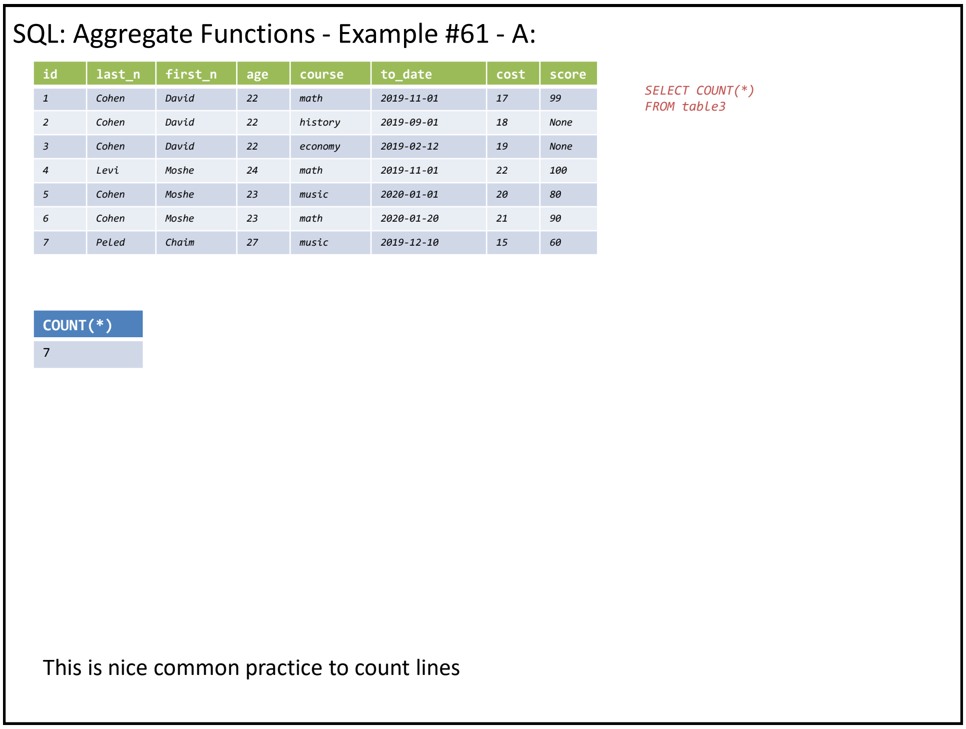
COUNT(score)
5



SQL: Aggregate Functions - Example #61 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT COUNT(*)
FROM table3
```



SQL: Aggregate Functions - Example #61 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT COUNT(*)
FROM table3
```

COUNT(*)
7

This is nice common practice to count lines

SQL: Aggregate Functions - Example #62 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT COUNT(1)
FROM table3
```

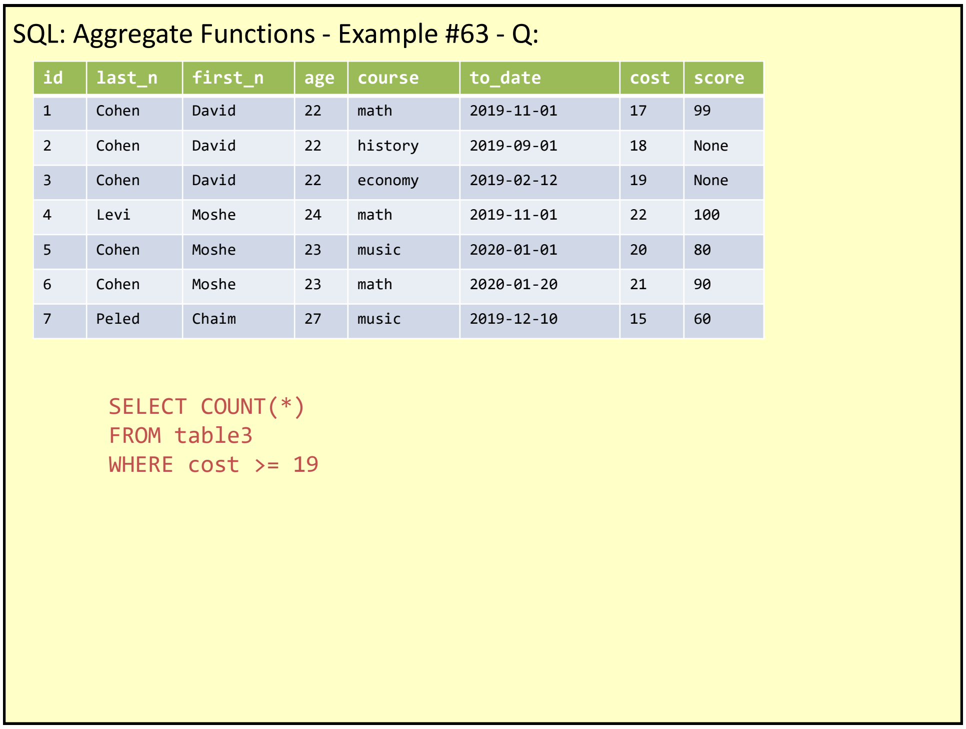
SQL: Aggregate Functions - Example #62 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT COUNT(1)
FROM table3
```

COUNT(1)
7

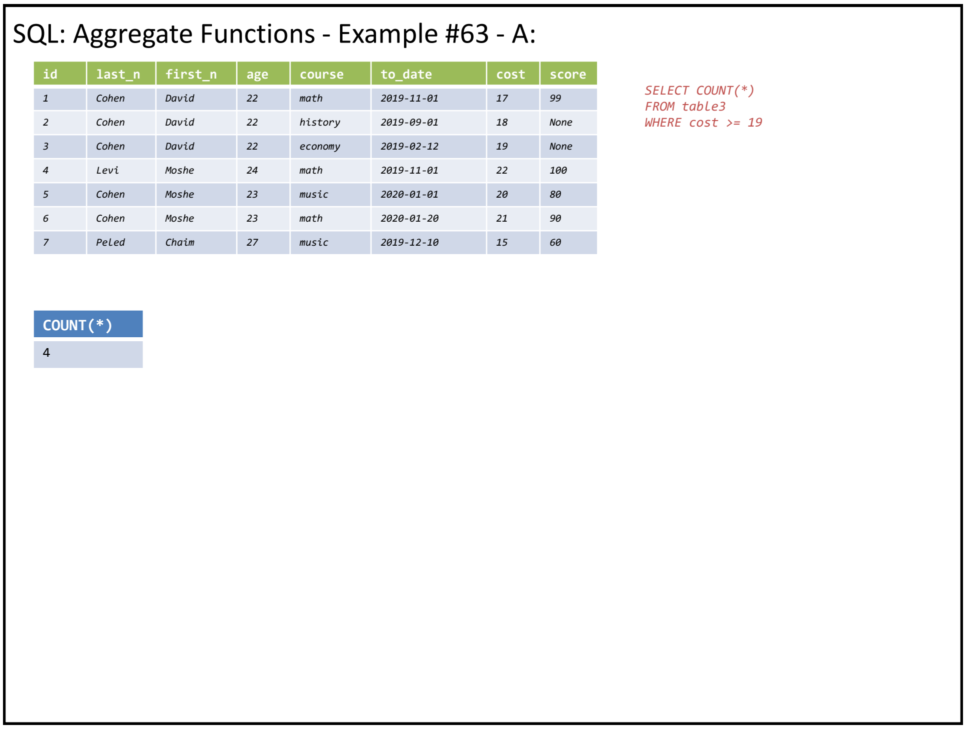
.. and also this (less nice)



SQL: Aggregate Functions - Example #63 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT COUNT(*)
FROM table3
WHERE cost >= 19
```



SQL: Aggregate Functions - Example #63 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT COUNT(*)
FROM table3
WHERE cost >= 19
```

COUNT(*)
4

SQL: Aggregate Functions - Example #64 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT MIN(score), AVG(score), MAX(score)
FROM table3
```

SQL: Aggregate Functions - Example #64 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	*Cohen*	*David*	*22*	*math*	*2019-11-01*	*17*	*99*
2	*Cohen*	*David*	*22*	*history*	*2019-09-01*	*18*	*None*
3	*Cohen*	*David*	*22*	*economy*	*2019-02-12*	*19*	*None*
4	*Levi*	*Moshe*	*24*	*math*	*2019-11-01*	*22*	*100*
5	*Cohen*	*Moshe*	*23*	*music*	*2020-01-01*	*20*	*80*
6	*Cohen*	*Moshe*	*23*	*math*	*2020-01-20*	*21*	*90*
7	*Peled*	*Chaim*	*27*	*music*	*2019-12-10*	*15*	*60*

```
    SELECT MIN(score), AVG(score),
MAX(score)
    FROM table3
```

MIN(score)	AVG(score)	MAX(score)
60	85.8	100

SQL: Aggregate Functions - Example #65 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

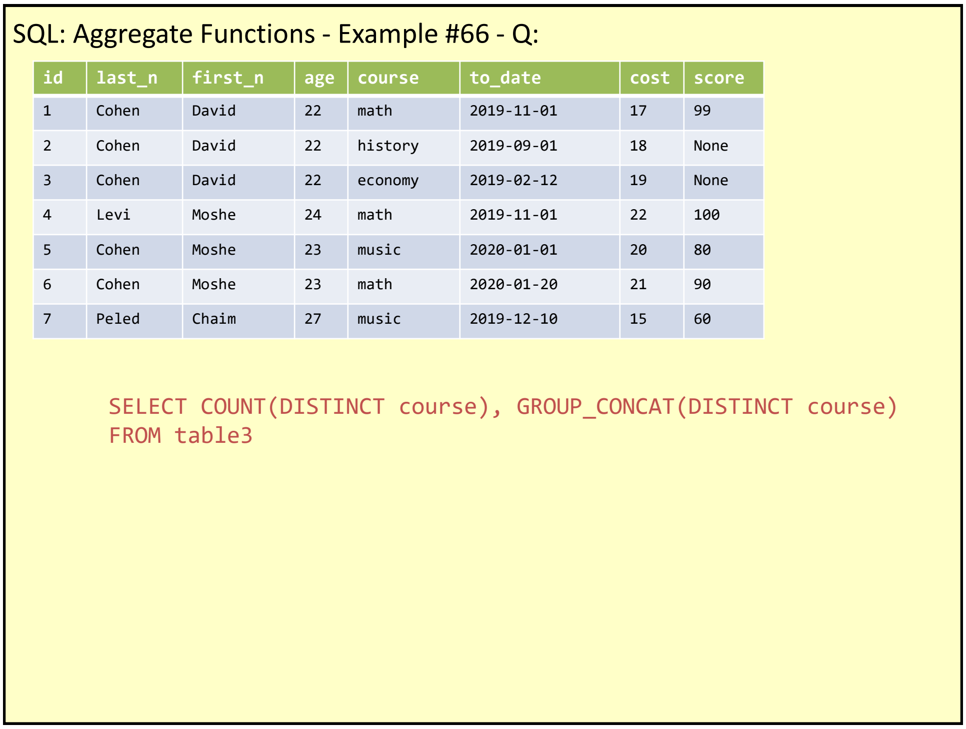
```
SELECT SUM(cost), GROUP_CONCAT(course)
FROM table3
```

SQL: Aggregate Functions - Example #65 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT SUM(cost), GROUP_CONCAT(course)
FROM table3
```

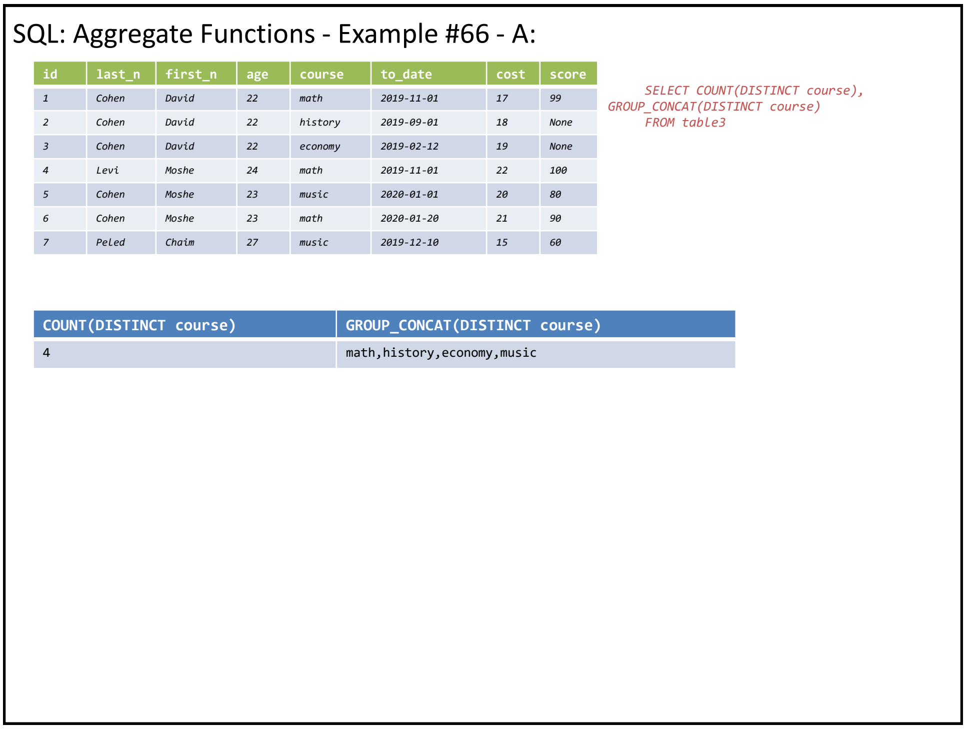
SUM(cost)	GROUP_CONCAT(course)
132	math,history,economy,math,music,math,music



SQL: Aggregate Functions - Example #66 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT COUNT(DISTINCT course), GROUP_CONCAT(DISTINCT course)
FROM table3
```



SQL: Aggregate Functions - Example #66 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
    SELECT COUNT(DISTINCT course),
GROUP_CONCAT(DISTINCT course)
    FROM table3
```

COUNT(DISTINCT course)	GROUP_CONCAT(DISTINCT course)
4	math,history,economy,music

Learn SQL by Example (Chananel Perel 2023)

SUB QUERY

SQL: Sub Query - Example #67 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT *
FROM table3
WHERE score > ( SELECT score
                FROM table3
                WHERE id = 6 )
```

SQL: Sub Query - Example #67 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

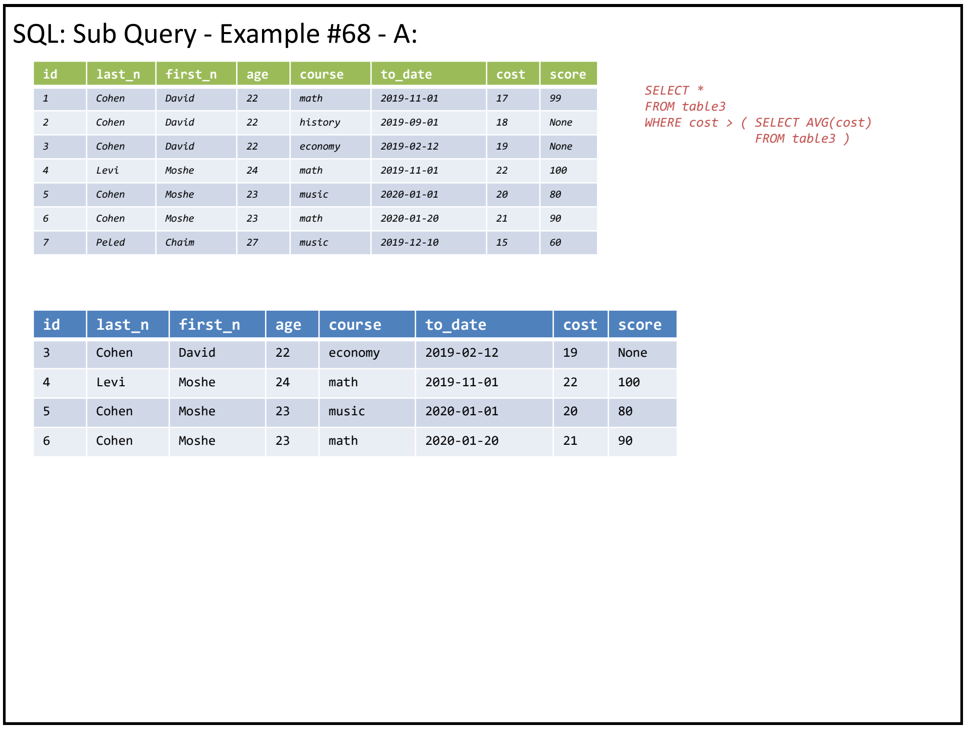
```
SELECT *
FROM table3
WHERE score > ( SELECT score
                FROM table3
                WHERE id = 6 )
```

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
4	Levi	Moshe	24	math	2019-11-01	22	100

SQL: Sub Query - Example #68 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT *
FROM table3
WHERE cost > ( SELECT AVG(cost)
               FROM table3 )
```

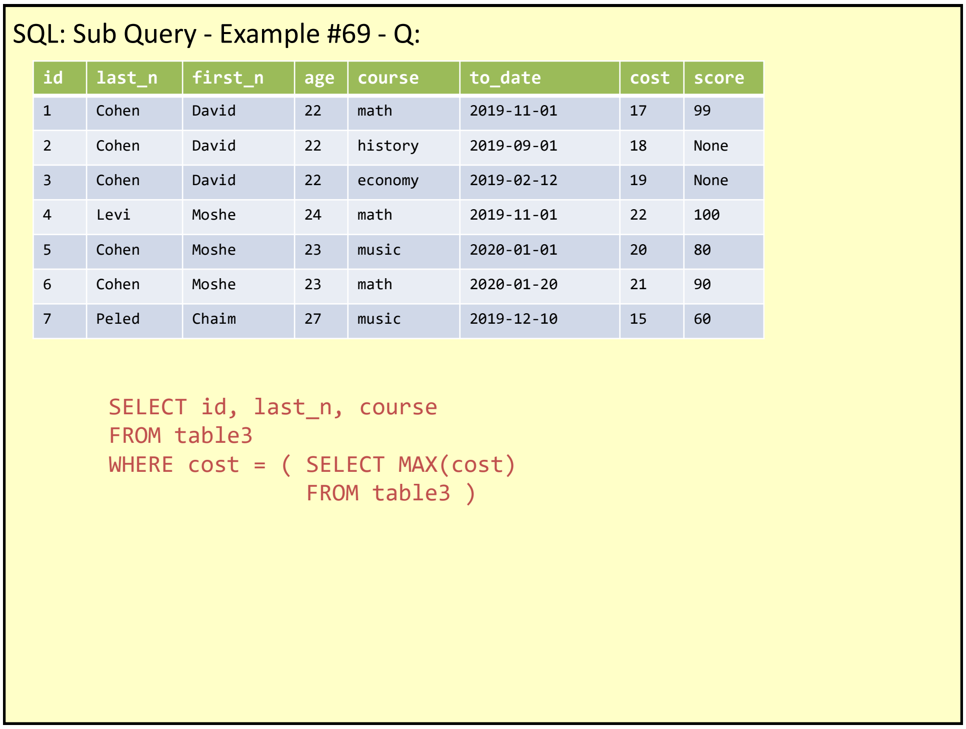


SQL: Sub Query - Example #68 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT *
FROM table3
WHERE cost > ( SELECT AVG(cost)
               FROM table3 )
```

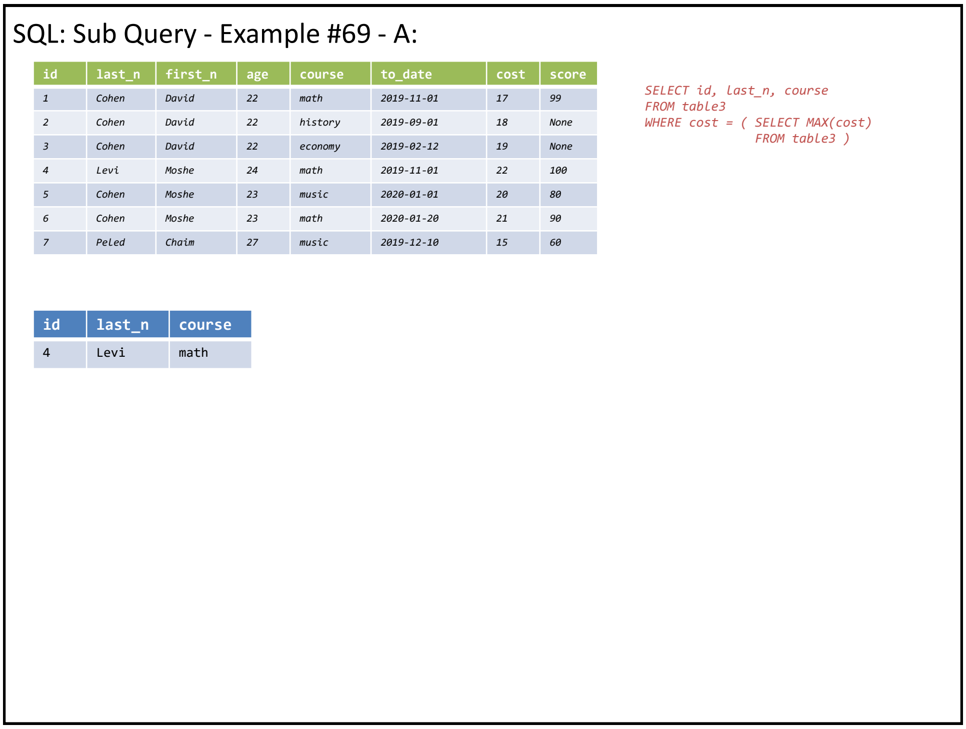
id	last_n	first_n	age	course	to_date	cost	score
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90



SQL: Sub Query - Example #69 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT id, last_n, course
FROM table3
WHERE cost = ( SELECT MAX(cost)
               FROM table3 )
```

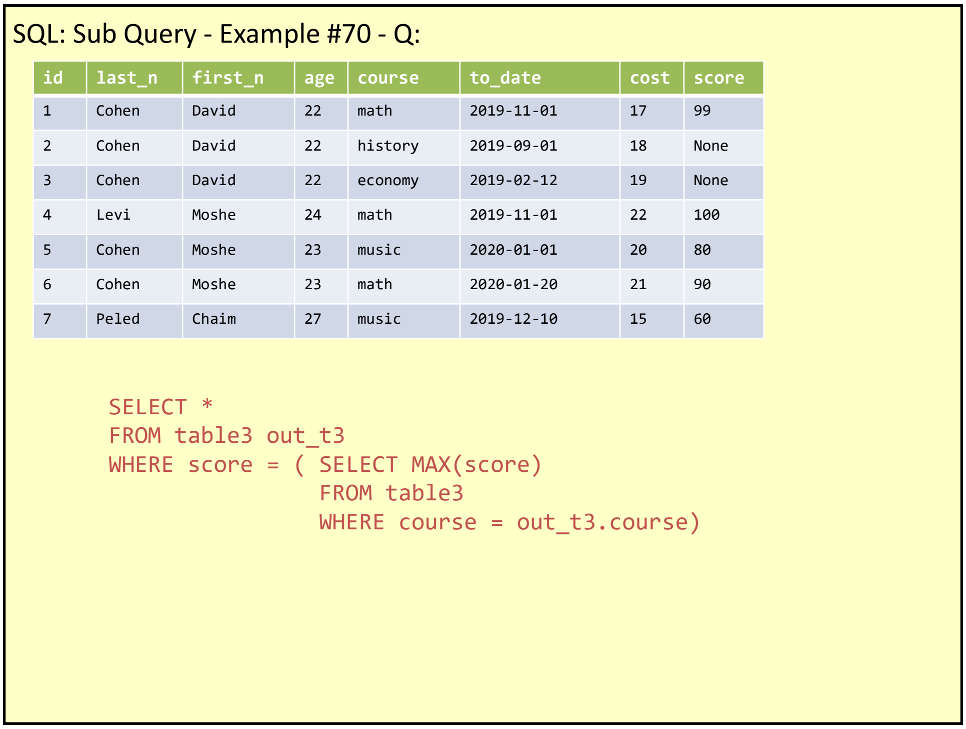


SQL: Sub Query - Example #69 - A:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT id, last_n, course
FROM table3
WHERE cost = ( SELECT MAX(cost)
               FROM table3 )
```

id	last_n	course
4	Levi	math



SQL: Sub Query - Example #70 - Q:

id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
SELECT *
FROM table3 out_t3
WHERE score = ( SELECT MAX(score)
                FROM table3
                WHERE course = out_t3.course)
```

SQL: Sub Query - Example #70 - A:

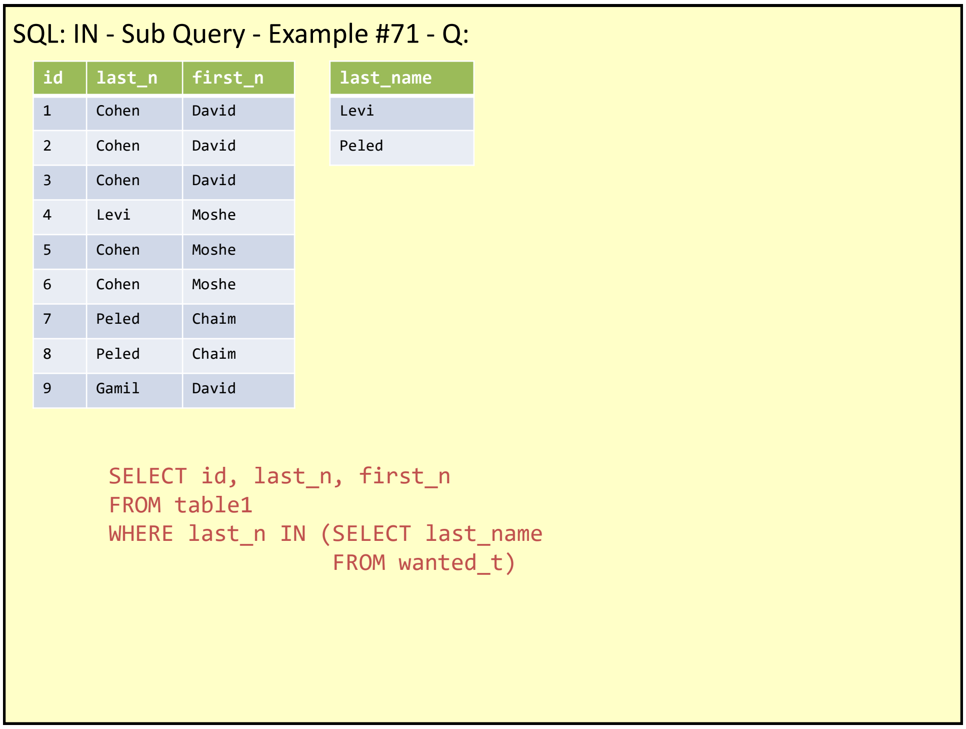
id	last_n	first_n	age	course	to_date	cost	score
1	Cohen	David	22	math	2019-11-01	17	99
2	Cohen	David	22	history	2019-09-01	18	None
3	Cohen	David	22	economy	2019-02-12	19	None
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80
6	Cohen	Moshe	23	math	2020-01-20	21	90
7	Peled	Chaim	27	music	2019-12-10	15	60

```
    SELECT *
    FROM table3 out_t3
    WHERE score = ( SELECT MAX(score)
                    FROM table3
                    WHERE course =
out_t3.course)
```

id	last_n	first_n	age	course	to_date	cost	score
4	Levi	Moshe	24	math	2019-11-01	22	100
5	Cohen	Moshe	23	music	2020-01-01	20	80

Learn SQL by Example (Chananel Perel 2023)

IN - SUB QUERY

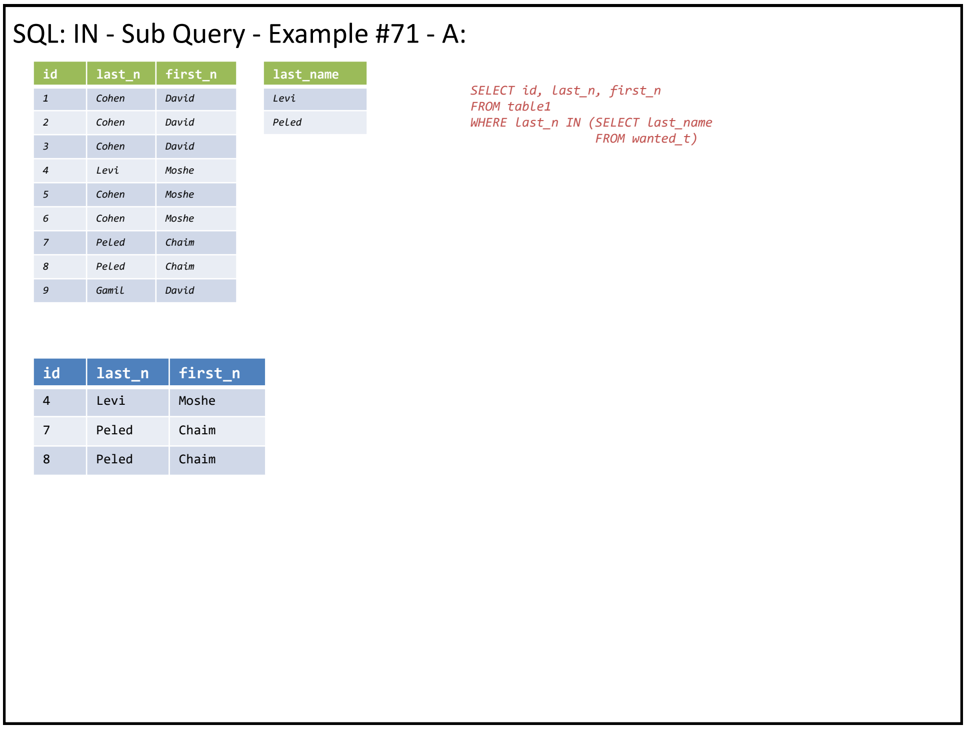


SQL: IN - Sub Query - Example #71 - Q:

id	last_n	first_n
1	Cohen	David
2	Cohen	David
3	Cohen	David
4	Levi	Moshe
5	Cohen	Moshe
6	Cohen	Moshe
7	Peled	Chaim
8	Peled	Chaim
9	Gamil	David

last_name
Levi
Peled

```
SELECT id, last_n, first_n
FROM table1
WHERE last_n IN (SELECT last_name
                 FROM wanted_t)
```



SQL: IN - Sub Query - Example #71 - A:

id	last_n	first_n
1	Cohen	David
2	Cohen	David
3	Cohen	David
4	Levi	Moshe
5	Cohen	Moshe
6	Cohen	Moshe
7	Peled	Chaim
8	Peled	Chaim
9	Gamil	David

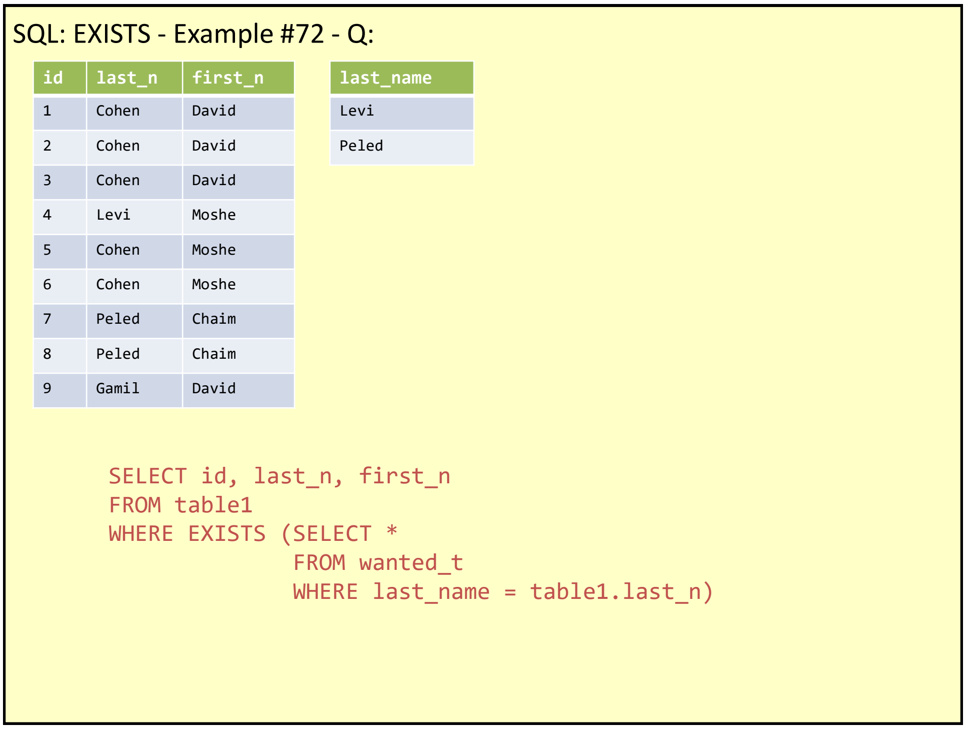
last_name
Levi
Peled

```
SELECT id, last_n, first_n
FROM table1
WHERE last_n IN (SELECT last_name
                 FROM wanted_t)
```

id	last_n	first_n
4	Levi	Moshe
7	Peled	Chaim
8	Peled	Chaim

Learn SQL by Example (Chananel Perel 2023)

EXISTS

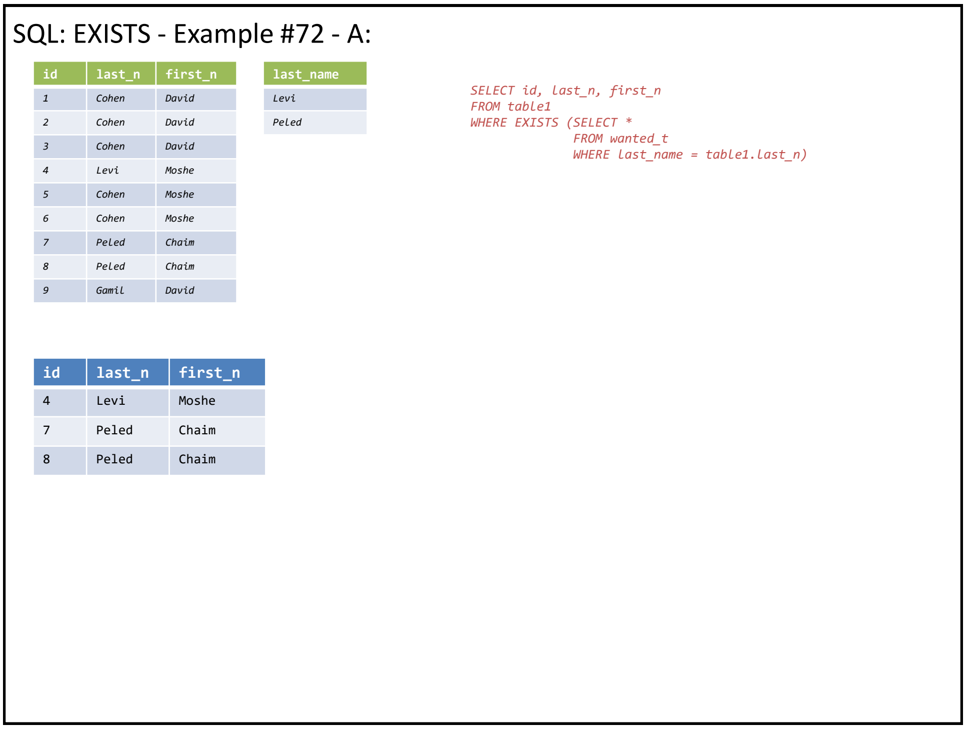


SQL: EXISTS - Example #72 - Q:

id	last_n	first_n
1	Cohen	David
2	Cohen	David
3	Cohen	David
4	Levi	Moshe
5	Cohen	Moshe
6	Cohen	Moshe
7	Peled	Chaim
8	Peled	Chaim
9	Gamil	David

last_name
Levi
Peled

```
SELECT id, last_n, first_n
FROM table1
WHERE EXISTS (SELECT *
              FROM wanted_t
              WHERE last_name = table1.last_n)
```



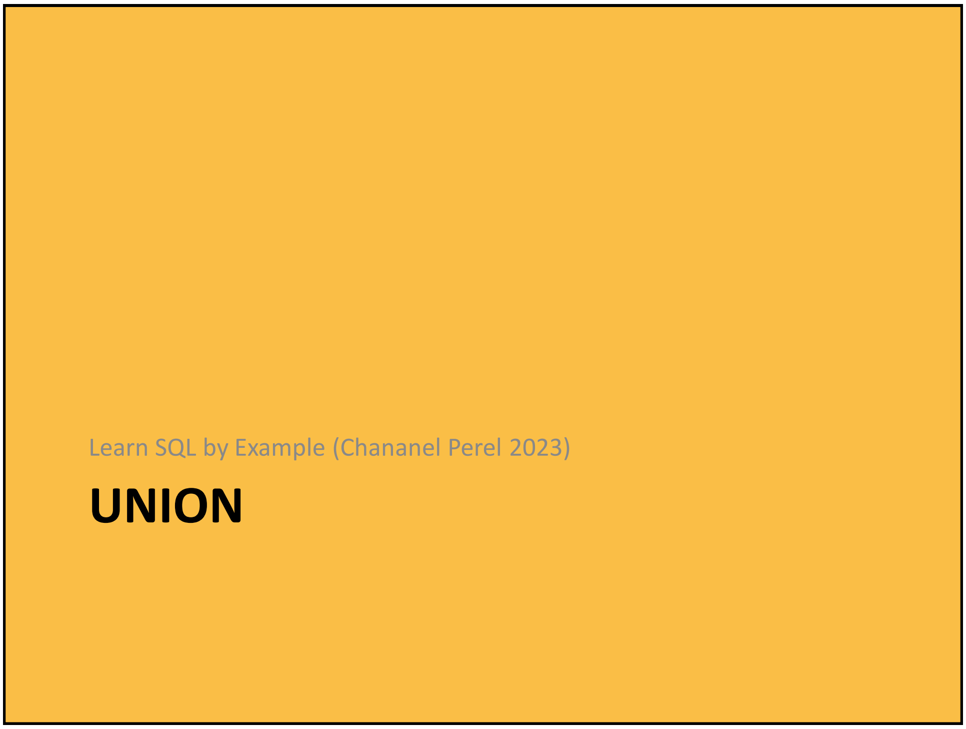
SQL: EXISTS - Example #72 - A:

id	last_n	first_n
1	Cohen	David
2	Cohen	David
3	Cohen	David
4	Levi	Moshe
5	Cohen	Moshe
6	Cohen	Moshe
7	Peled	Chaim
8	Peled	Chaim
9	Gamil	David

last_name
Levi
Peled

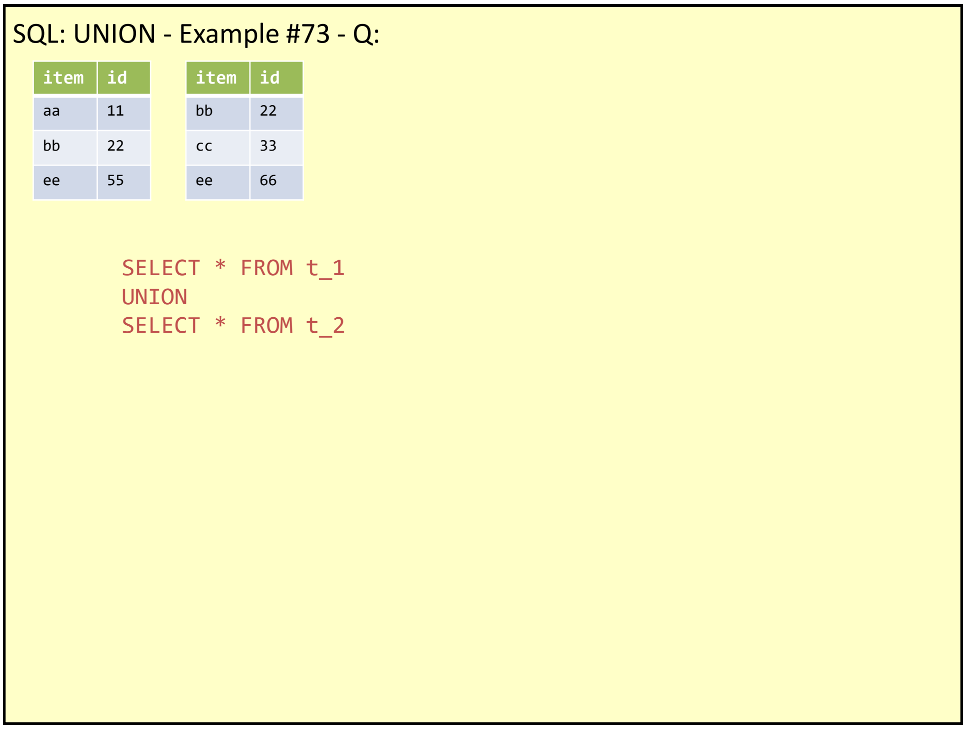
```
SELECT id, last_n, first_n
FROM table1
WHERE EXISTS (SELECT *
              FROM wanted_t
              WHERE last_name = table1.last_n)
```

id	last_n	first_n
4	Levi	Moshe
7	Peled	Chaim
8	Peled	Chaim



Learn SQL by Example (Chananel Perel 2023)

UNION

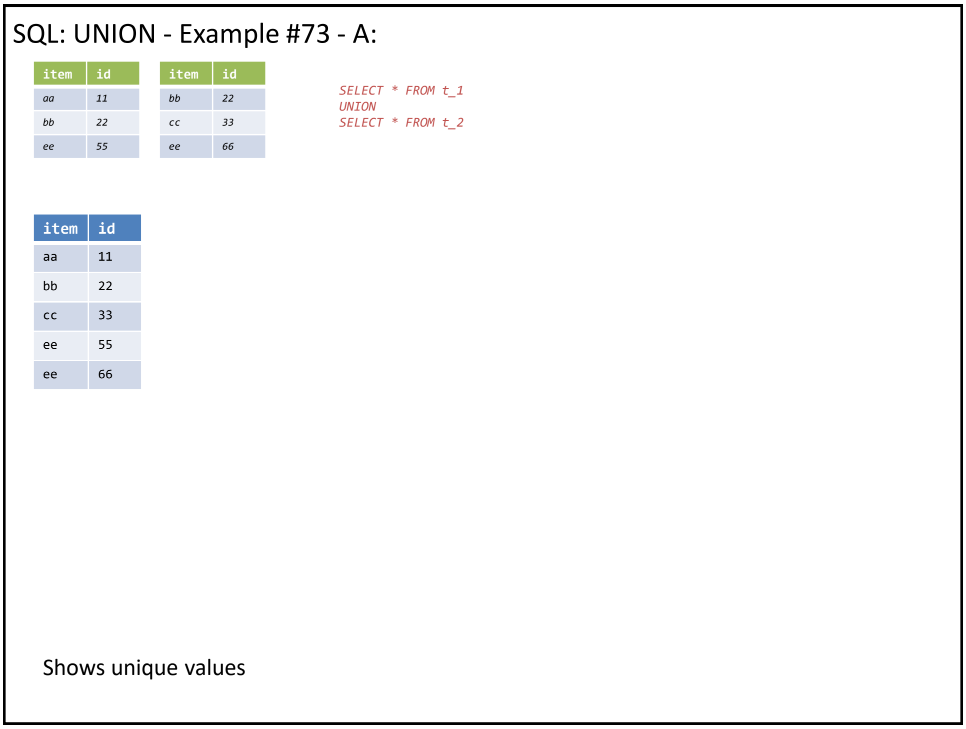


SQL: UNION - Example #73 - Q:

item	id
aa	11
bb	22
ee	55

item	id
bb	22
cc	33
ee	66

```
SELECT * FROM t_1
UNION
SELECT * FROM t_2
```



SQL: UNION - Example #73 - A:

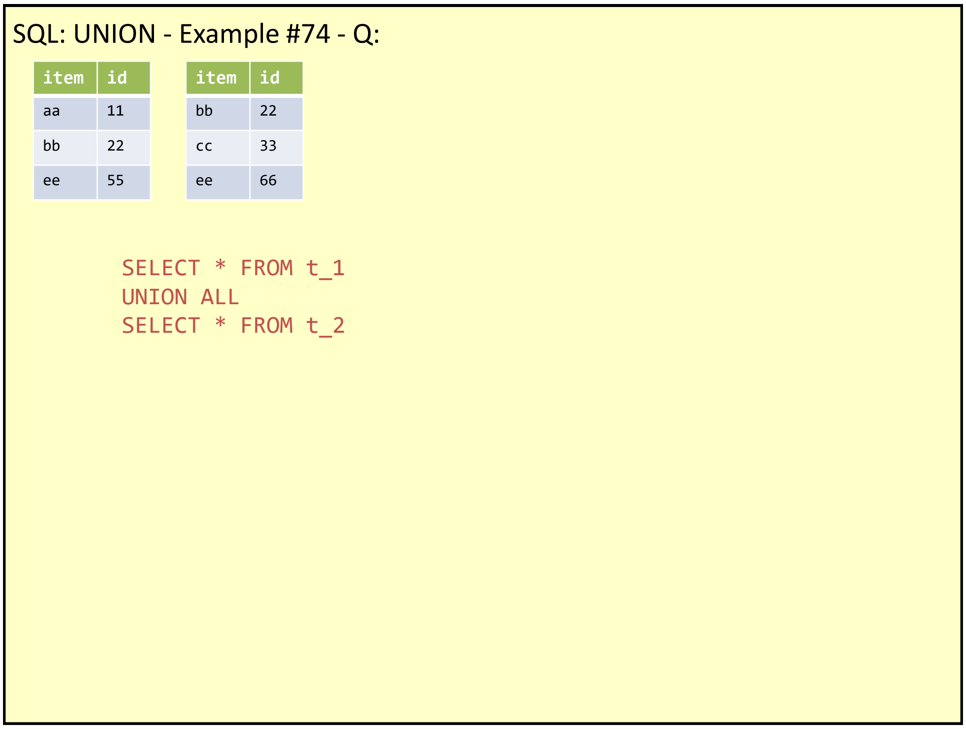
item	id
aa	11
bb	22
ee	55

item	id
bb	22
cc	33
ee	66

```
SELECT * FROM t_1
UNION
SELECT * FROM t_2
```

item	id
aa	11
bb	22
cc	33
ee	55
ee	66

Shows unique values

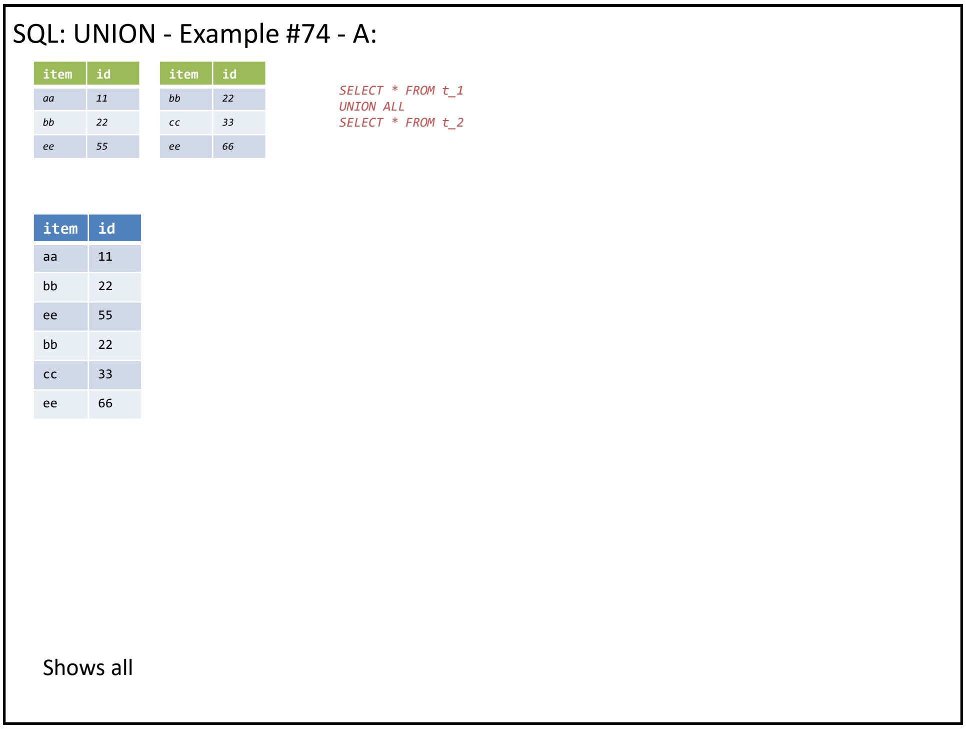


SQL: UNION - Example #74 - Q:

item	id
aa	11
bb	22
ee	55

item	id
bb	22
cc	33
ee	66

```
SELECT * FROM t_1
UNION ALL
SELECT * FROM t_2
```



SQL: UNION - Example #74 - A:

item	id
aa	11
bb	22
ee	55

item	id
bb	22
cc	33
ee	66

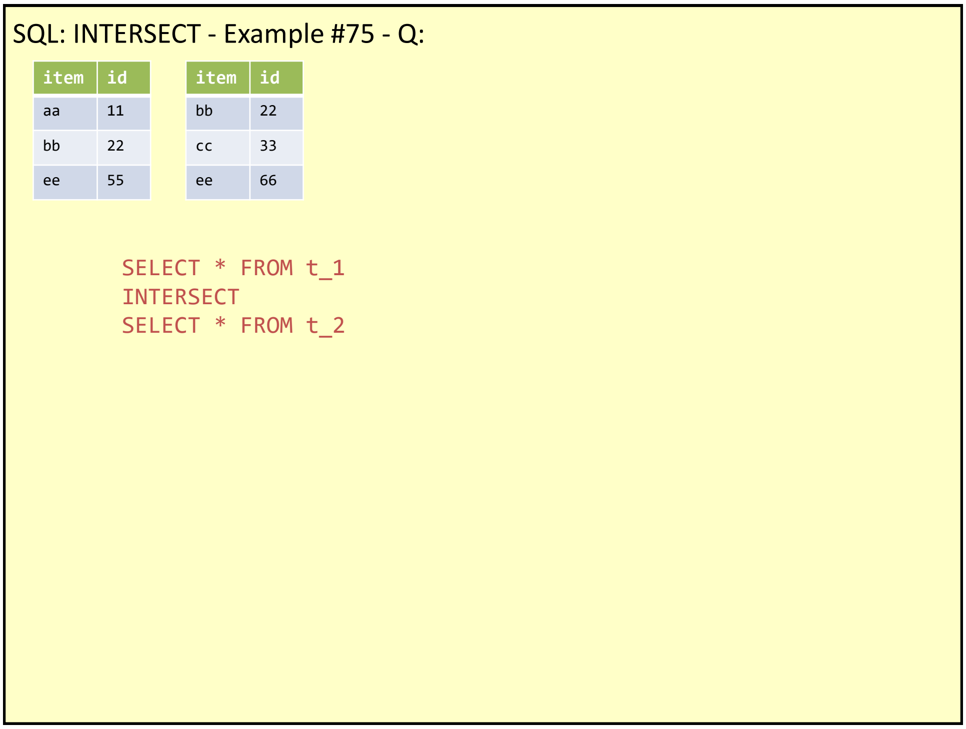
```
SELECT * FROM t_1
UNION ALL
SELECT * FROM t_2
```

item	id
aa	11
bb	22
ee	55
bb	22
cc	33
ee	66

Shows all

Learn SQL by Example (Chananel Perel 2023)

INTERSECT

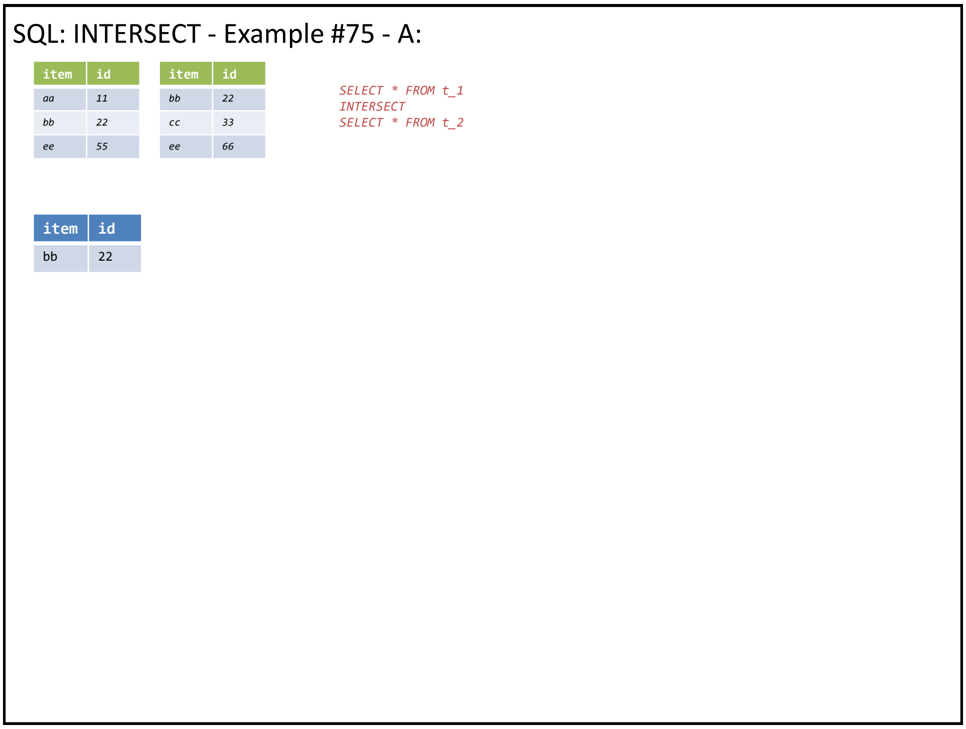


SQL: INTERSECT - Example #75 - Q:

item	id
aa	11
bb	22
ee	55

item	id
bb	22
cc	33
ee	66

```
SELECT * FROM t_1
INTERSECT
SELECT * FROM t_2
```



SQL: INTERSECT - Example #75 - A:

item	id
aa	11
bb	22
ee	55

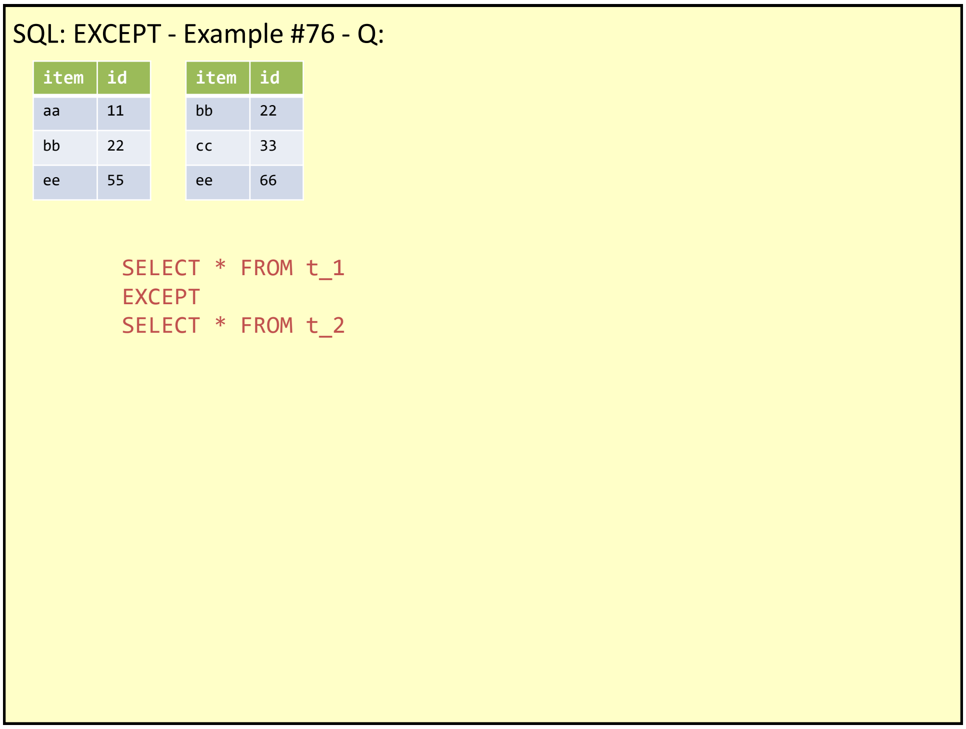
item	id
bb	22
cc	33
ee	66

```
SELECT * FROM t_1
INTERSECT
SELECT * FROM t_2
```

item	id
bb	22

Learn SQL by Example (Chananel Perel 2023)

EXCEPT

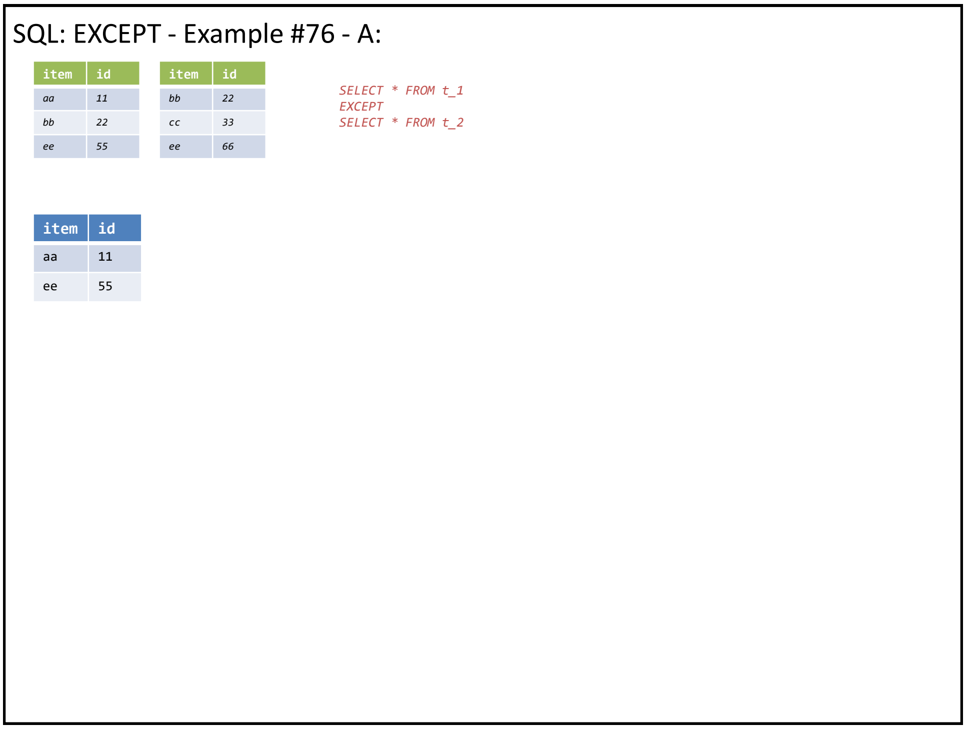


SQL: EXCEPT - Example #76 - Q:

item	id
aa	11
bb	22
ee	55

item	id
bb	22
cc	33
ee	66

```
SELECT * FROM t_1
EXCEPT
SELECT * FROM t_2
```



SQL: EXCEPT - Example #76 - A:

item	id
aa	11
bb	22
ee	55

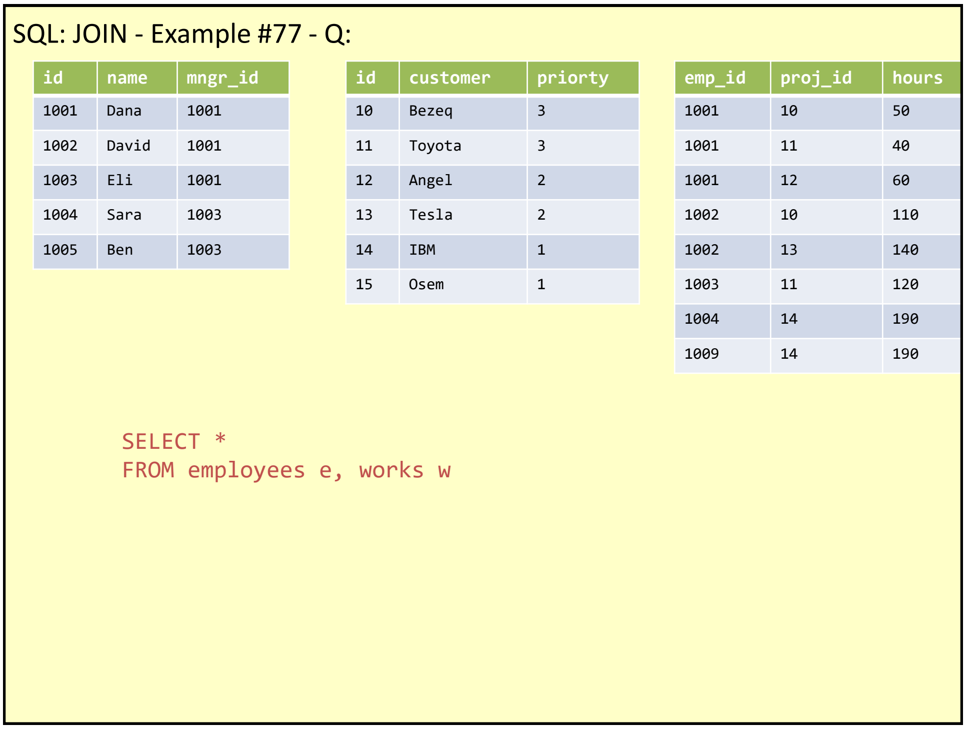
item	id
bb	22
cc	33
ee	66

```
SELECT * FROM t_1
EXCEPT
SELECT * FROM t_2
```

item	id
aa	11
ee	55

Learn SQL by Example (Chananel Perel 2023)

JOIN



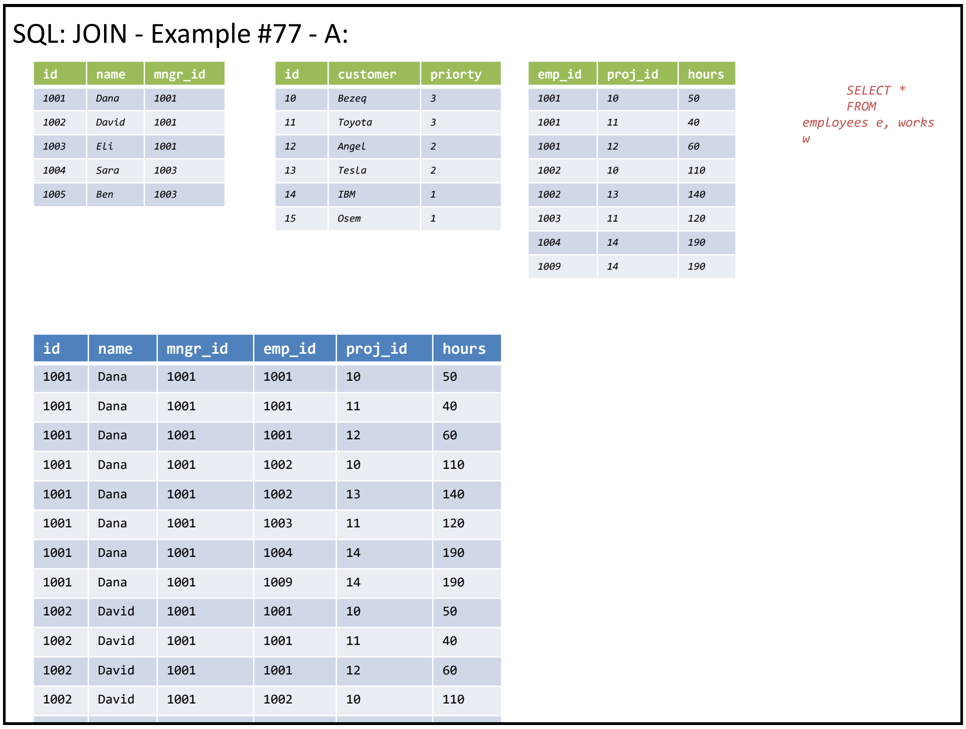
SQL: JOIN - Example #77 - Q:

id	name	mngr_id
1001	Dana	1001
1002	David	1001
1003	Eli	1001
1004	Sara	1003
1005	Ben	1003

id	customer	priorty
10	Bezeq	3
11	Toyota	3
12	Angel	2
13	Tesla	2
14	IBM	1
15	Osem	1

emp_id	proj_id	hours
1001	10	50
1001	11	40
1001	12	60
1002	10	110
1002	13	140
1003	11	120
1004	14	190
1009	14	190

```
SELECT *
FROM employees e, works w
```



SQL: JOIN - Example #77 - A:

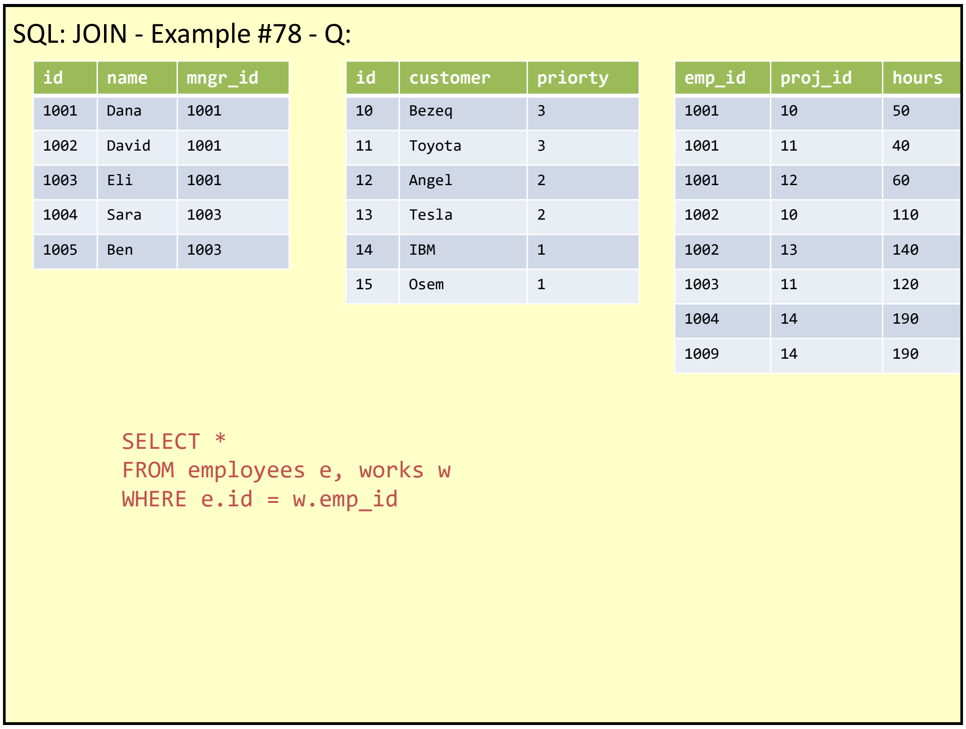
id	name	mngr_id
1001	Dana	1001
1002	David	1001
1003	Eli	1001
1004	Sara	1003
1005	Ben	1003

id	customer	priorty
10	Bezeq	3
11	Toyota	3
12	Angel	2
13	Tesla	2
14	IBM	1
15	Osem	1

emp_id	proj_id	hours
1001	10	50
1001	11	40
1001	12	60
1002	10	110
1002	13	140
1003	11	120
1004	14	190
1009	14	190

```
SELECT *
FROM
employees e, works
w
```

id	name	mngr_id	emp_id	proj_id	hours
1001	Dana	1001	1001	10	50
1001	Dana	1001	1001	11	40
1001	Dana	1001	1001	12	60
1001	Dana	1001	1002	10	110
1001	Dana	1001	1002	13	140
1001	Dana	1001	1003	11	120
1001	Dana	1001	1004	14	190
1001	Dana	1001	1009	14	190
1002	David	1001	1001	10	50
1002	David	1001	1001	11	40
1002	David	1001	1001	12	60
1002	David	1001	1002	10	110



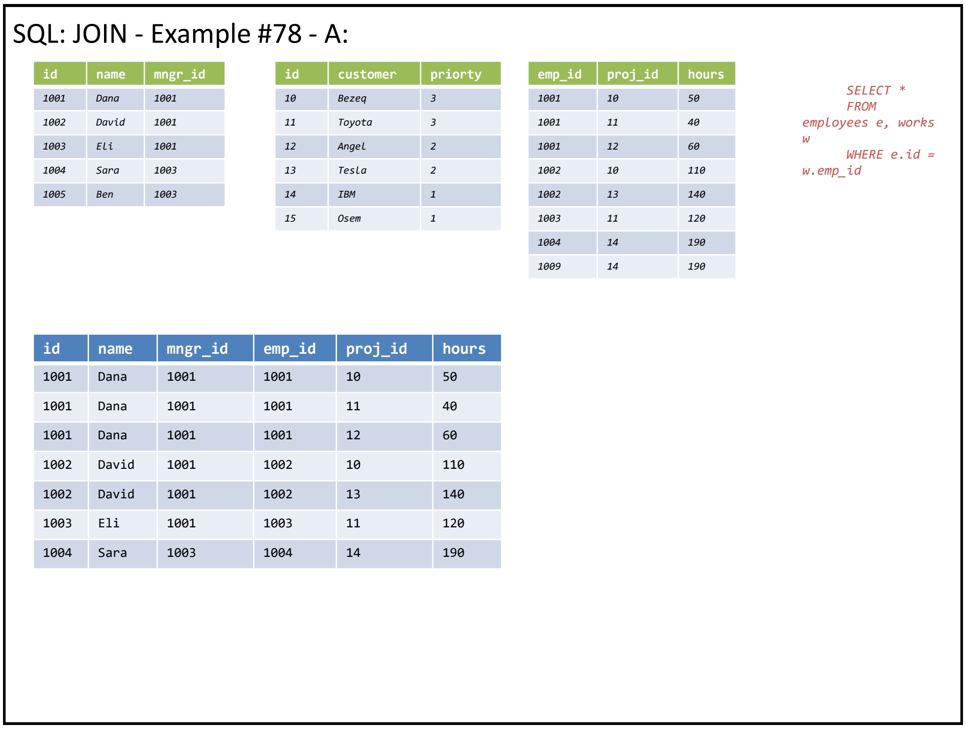
SQL: JOIN - Example #78 - Q:

id	name	mngr_id
1001	Dana	1001
1002	David	1001
1003	Eli	1001
1004	Sara	1003
1005	Ben	1003

id	customer	priorty
10	Bezeq	3
11	Toyota	3
12	Angel	2
13	Tesla	2
14	IBM	1
15	Osem	1

emp_id	proj_id	hours
1001	10	50
1001	11	40
1001	12	60
1002	10	110
1002	13	140
1003	11	120
1004	14	190
1009	14	190

```
SELECT *
FROM employees e, works w
WHERE e.id = w.emp_id
```



SQL: JOIN - Example #78 - A:

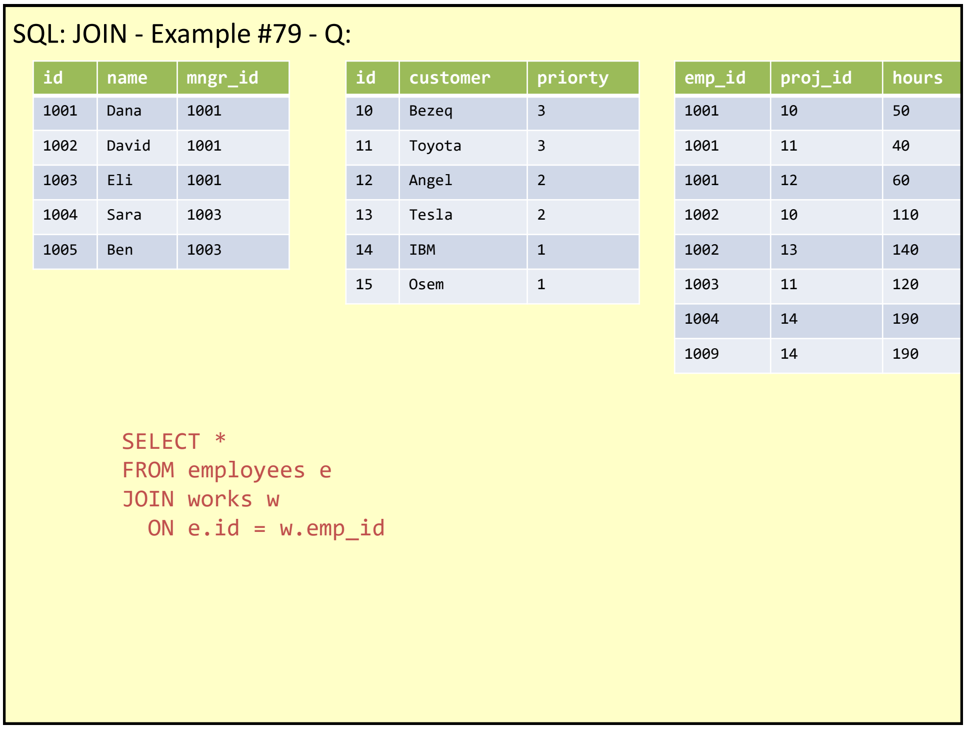
id	name	mngr_id
1001	Dana	1001
1002	David	1001
1003	Eli	1001
1004	Sara	1003
1005	Ben	1003

id	customer	priorty
10	Bezeq	3
11	Toyota	3
12	Angel	2
13	Tesla	2
14	IBM	1
15	Osem	1

emp_id	proj_id	hours
1001	10	50
1001	11	40
1001	12	60
1002	10	110
1002	13	140
1003	11	120
1004	14	190
1009	14	190

```
        SELECT *
        FROM
employees e, works
w
        WHERE e.id =
w.emp_id
```

id	name	mngr_id	emp_id	proj_id	hours
1001	Dana	1001	1001	10	50
1001	Dana	1001	1001	11	40
1001	Dana	1001	1001	12	60
1002	David	1001	1002	10	110
1002	David	1001	1002	13	140
1003	Eli	1001	1003	11	120
1004	Sara	1003	1004	14	190



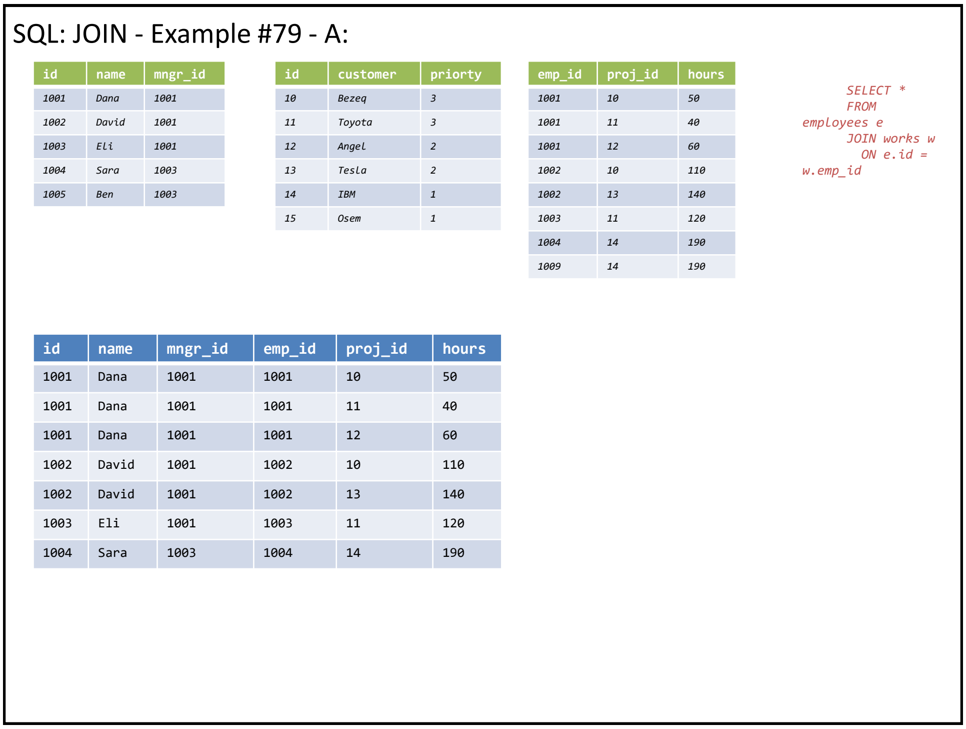
SQL: JOIN - Example #79 - Q:

id	name	mngr_id
1001	Dana	1001
1002	David	1001
1003	Eli	1001
1004	Sara	1003
1005	Ben	1003

id	customer	priorty
10	Bezeq	3
11	Toyota	3
12	Angel	2
13	Tesla	2
14	IBM	1
15	Osem	1

emp_id	proj_id	hours
1001	10	50
1001	11	40
1001	12	60
1002	10	110
1002	13	140
1003	11	120
1004	14	190
1009	14	190

```
SELECT *
FROM employees e
JOIN works w
  ON e.id = w.emp_id
```



SQL: JOIN - Example #79 - A:

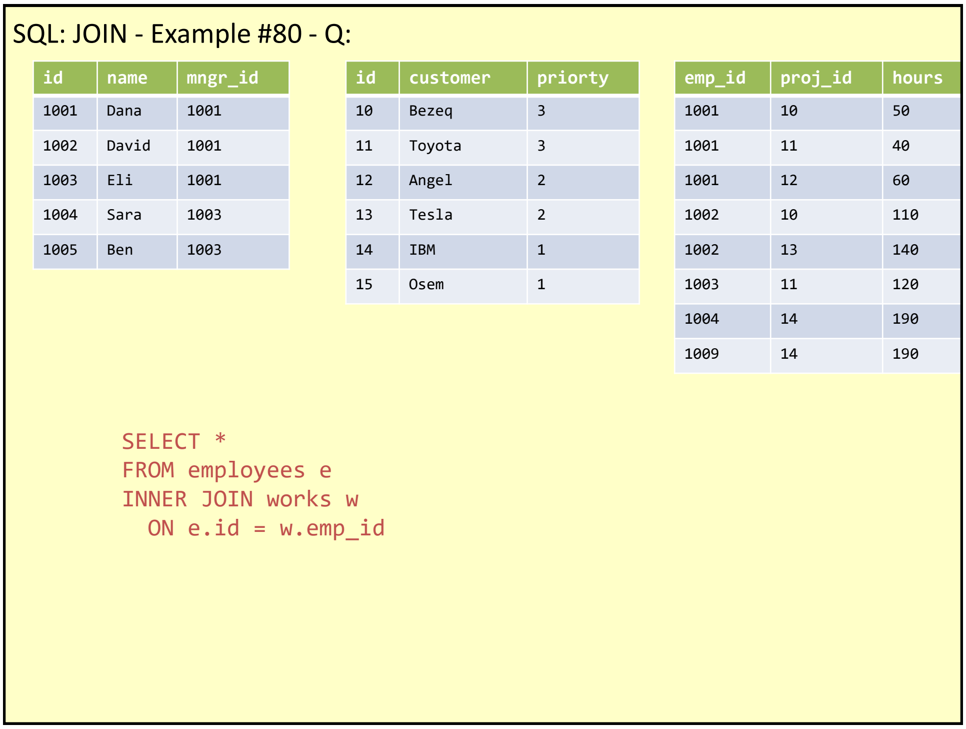
id	name	mngr_id
1001	Dana	1001
1002	David	1001
1003	Eli	1001
1004	Sara	1003
1005	Ben	1003

id	customer	priorty
10	Bezeq	3
11	Toyota	3
12	Angel	2
13	Tesla	2
14	IBM	1
15	Osem	1

emp_id	proj_id	hours
1001	10	50
1001	11	40
1001	12	60
1002	10	110
1002	13	140
1003	11	120
1004	14	190
1009	14	190

```
      SELECT *
      FROM
employees e
      JOIN works w
        ON e.id =
w.emp_id
```

id	name	mngr_id	emp_id	proj_id	hours
1001	Dana	1001	1001	10	50
1001	Dana	1001	1001	11	40
1001	Dana	1001	1001	12	60
1002	David	1001	1002	10	110
1002	David	1001	1002	13	140
1003	Eli	1001	1003	11	120
1004	Sara	1003	1004	14	190



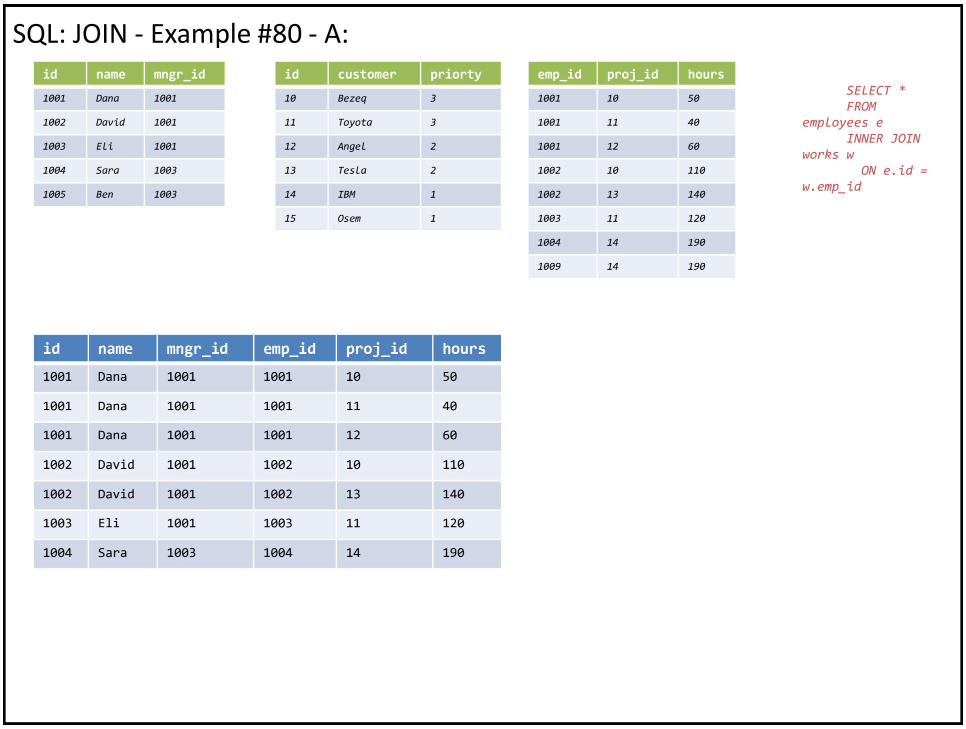
SQL: JOIN - Example #80 - Q:

id	name	mngr_id
1001	Dana	1001
1002	David	1001
1003	Eli	1001
1004	Sara	1003
1005	Ben	1003

id	customer	priorty
10	Bezeq	3
11	Toyota	3
12	Angel	2
13	Tesla	2
14	IBM	1
15	Osem	1

emp_id	proj_id	hours
1001	10	50
1001	11	40
1001	12	60
1002	10	110
1002	13	140
1003	11	120
1004	14	190
1009	14	190

```
SELECT *
FROM employees e
INNER JOIN works w
  ON e.id = w.emp_id
```



SQL: JOIN - Example #80 - A:

id	name	mngr_id
1001	Dana	1001
1002	David	1001
1003	Eli	1001
1004	Sara	1003
1005	Ben	1003

id	customer	priorty
10	Bezeq	3
11	Toyota	3
12	Angel	2
13	Tesla	2
14	IBM	1
15	Osem	1

emp_id	proj_id	hours
1001	10	50
1001	11	40
1001	12	60
1002	10	110
1002	13	140
1003	11	120
1004	14	190
1009	14	190

```
      SELECT *
      FROM
employees e
      INNER JOIN
works w
        ON e.id =
w.emp_id
```

id	name	mngr_id	emp_id	proj_id	hours
1001	Dana	1001	1001	10	50
1001	Dana	1001	1001	11	40
1001	Dana	1001	1001	12	60
1002	David	1001	1002	10	110
1002	David	1001	1002	13	140
1003	Eli	1001	1003	11	120
1004	Sara	1003	1004	14	190

Learn SQL by Example (Chananel Perel 2023)

LEFT JOIN

SQL: LEFT JOIN - Example #81 - Q:

id	name	mngr_id
1001	Dana	1001
1002	David	1001
1003	Eli	1001
1004	Sara	1003
1005	Ben	1003

emp_id	proj_id	hours
1001	10	50
1001	11	40
1001	12	60
1002	10	110
1002	13	140
1003	11	120
1004	14	190
1009	14	190

```
SELECT *
FROM employees e
LEFT JOIN works w
  ON e.id = w.emp_id
```

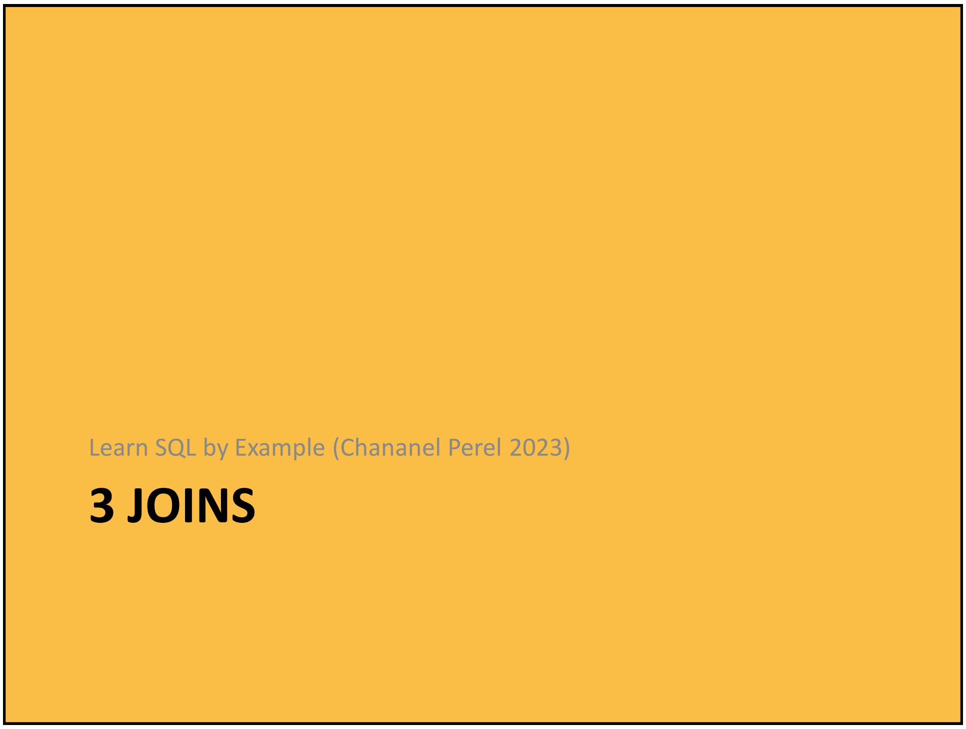
SQL: LEFT JOIN - Example #81 - A:

id	name	mngr_id
1001	Dana	1001
1002	David	1001
1003	Eli	1001
1004	Sara	1003
1005	Ben	1003

emp_id	proj_id	hours
1001	10	50
1001	11	40
1001	12	60
1002	10	110
1002	13	140
1003	11	120
1004	14	190
1009	14	190

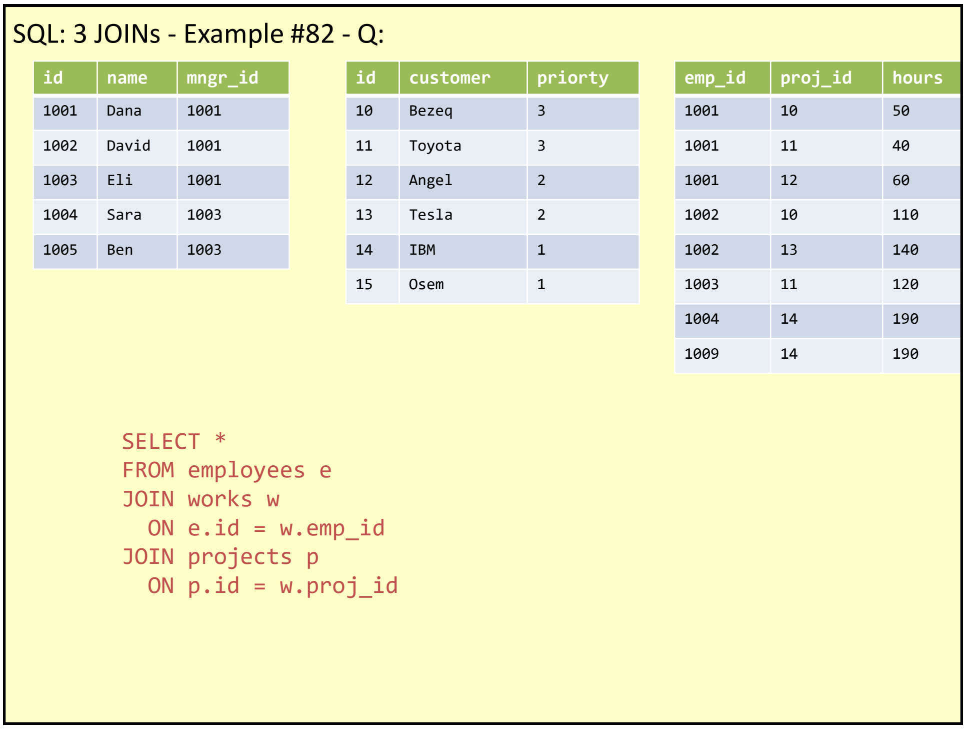
```
SELECT *
FROM employees e
LEFT JOIN works w
  ON e.id = w.emp_id
```

id	name	mngr_id	emp_id	proj_id	hours
1001	Dana	1001	1001	10	50
1001	Dana	1001	1001	11	40
1001	Dana	1001	1001	12	60
1002	David	1001	1002	10	110
1002	David	1001	1002	13	140
1003	Eli	1001	1003	11	120
1004	Sara	1003	1004	14	190
1005	Ben	1003	None	None	None



Learn SQL by Example (Chananel Perel 2023)

3 JOINS



SQL: 3 JOINs - Example #82 - Q:

id	name	mngr_id
1001	Dana	1001
1002	David	1001
1003	Eli	1001
1004	Sara	1003
1005	Ben	1003

id	customer	priorty
10	Bezeq	3
11	Toyota	3
12	Angel	2
13	Tesla	2
14	IBM	1
15	Osem	1

emp_id	proj_id	hours
1001	10	50
1001	11	40
1001	12	60
1002	10	110
1002	13	140
1003	11	120
1004	14	190
1009	14	190

```
SELECT *
FROM employees e
JOIN works w
  ON e.id = w.emp_id
JOIN projects p
  ON p.id = w.proj_id
```

SQL: 3 JOINs - Example #82 - A:

id	name	mngr_id
1001	Dana	1001
1002	David	1001
1003	Eli	1001
1004	Sara	1003
1005	Ben	1003

id	customer	priorty
10	Bezeq	3
11	Toyota	3
12	Angel	2
13	Tesla	2
14	IBM	1
15	Osem	1

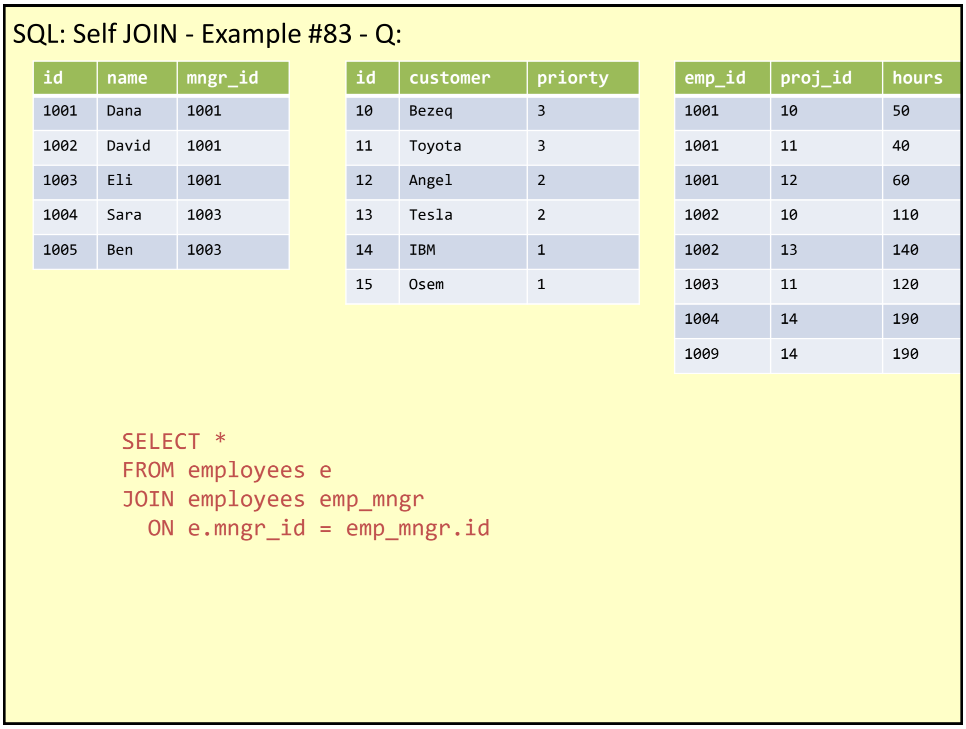
emp_id	proj_id	hours
1001	10	50
1001	11	40
1001	12	60
1002	10	110
1002	13	140
1003	11	120
1004	14	190
1009	14	190

```
      SELECT *
      FROM
employees e
      JOIN works w
        ON e.id =
w.emp_id
      JOIN
projects p
        ON p.id =
w.proj_id
```

id	name	mngr_id	emp_id	proj_id	hours	id	customer	priorty
1001	Dana	1001	1001	10	50	10	Bezeq	3
1001	Dana	1001	1001	11	40	11	Toyota	3
1001	Dana	1001	1001	12	60	12	Angel	2
1002	David	1001	1002	10	110	10	Bezeq	3
1002	David	1001	1002	13	140	13	Tesla	2
1003	Eli	1001	1003	11	120	11	Toyota	3
1004	Sara	1003	1004	14	190	14	IBM	1

Learn SQL by Example (Chananel Perel 2023)

SELF JOIN



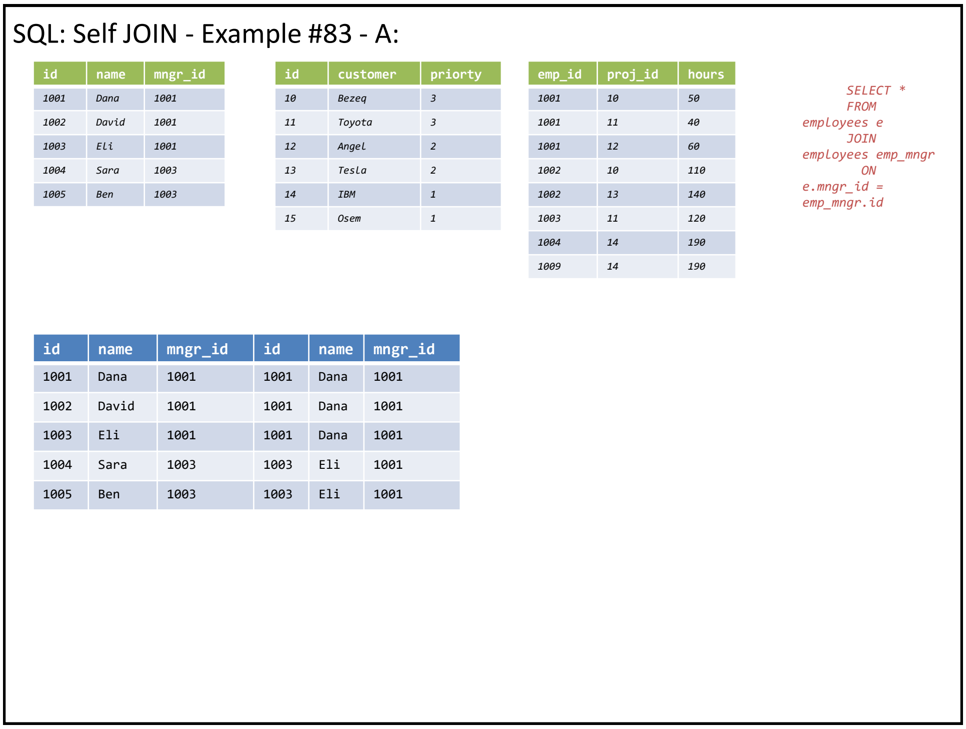
SQL: Self JOIN - Example #83 - Q:

id	name	mngr_id
1001	Dana	1001
1002	David	1001
1003	Eli	1001
1004	Sara	1003
1005	Ben	1003

id	customer	priorty
10	Bezeq	3
11	Toyota	3
12	Angel	2
13	Tesla	2
14	IBM	1
15	Osem	1

emp_id	proj_id	hours
1001	10	50
1001	11	40
1001	12	60
1002	10	110
1002	13	140
1003	11	120
1004	14	190
1009	14	190

```
SELECT *
FROM employees e
JOIN employees emp_mngr
  ON e.mngr_id = emp_mngr.id
```



SQL: Self JOIN - Example #83 - A:

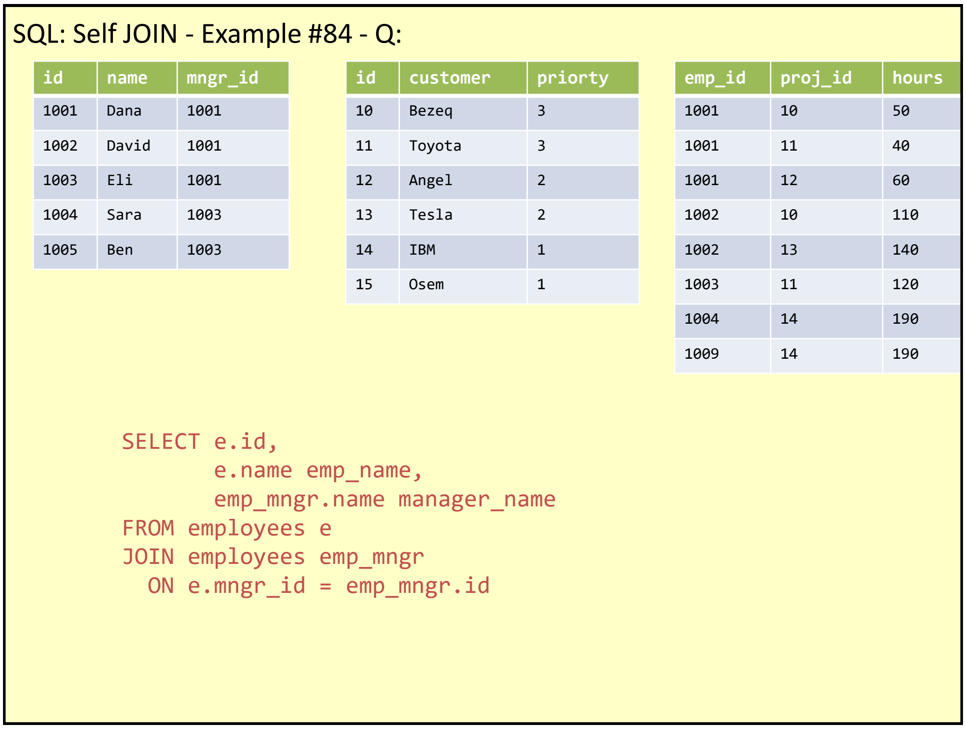
id	name	mngr_id
1001	Dana	1001
1002	David	1001
1003	Eli	1001
1004	Sara	1003
1005	Ben	1003

id	customer	priorty
10	Bezeq	3
11	Toyota	3
12	Angel	2
13	Tesla	2
14	IBM	1
15	Osem	1

emp_id	proj_id	hours
1001	10	50
1001	11	40
1001	12	60
1002	10	110
1002	13	140
1003	11	120
1004	14	190
1009	14	190

```
     SELECT *
    FROM
employees e
     JOIN
employees emp_mngr
      ON
e.mngr_id =
emp_mngr.id
```

id	name	mngr_id	id	name	mngr_id
1001	Dana	1001	1001	Dana	1001
1002	David	1001	1001	Dana	1001
1003	Eli	1001	1001	Dana	1001
1004	Sara	1003	1003	Eli	1001
1005	Ben	1003	1003	Eli	1001



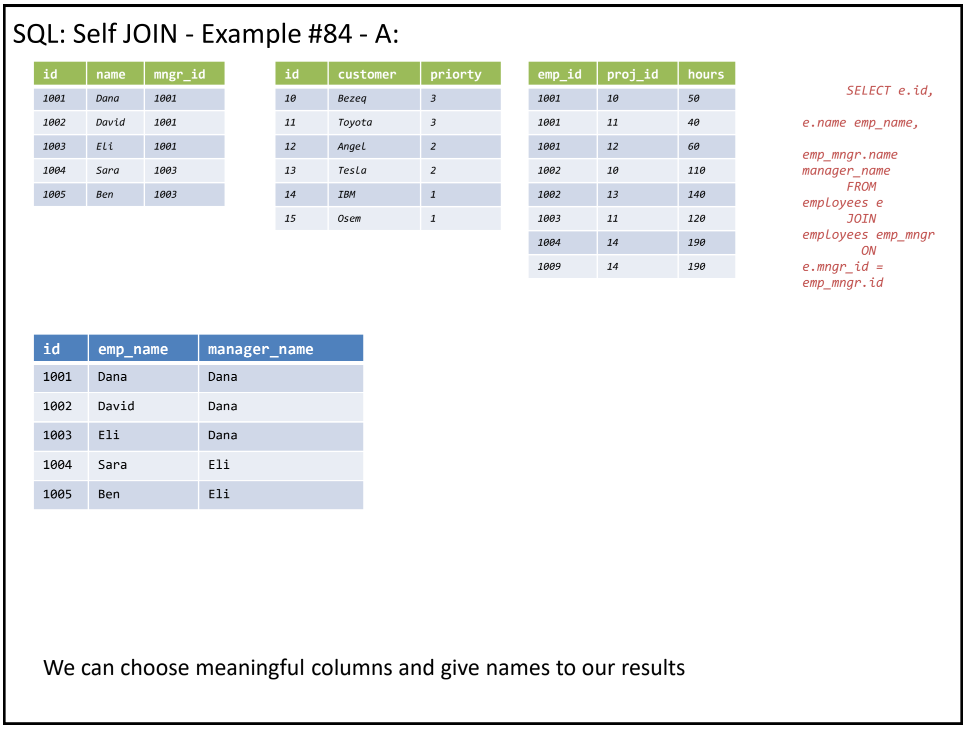
SQL: Self JOIN - Example #84 - Q:

id	name	mngr_id
1001	Dana	1001
1002	David	1001
1003	Eli	1001
1004	Sara	1003
1005	Ben	1003

id	customer	priorty
10	Bezeq	3
11	Toyota	3
12	Angel	2
13	Tesla	2
14	IBM	1
15	Osem	1

emp_id	proj_id	hours
1001	10	50
1001	11	40
1001	12	60
1002	10	110
1002	13	140
1003	11	120
1004	14	190
1009	14	190

```
SELECT e.id,
       e.name emp_name,
       emp_mngr.name manager_name
FROM employees e
JOIN employees emp_mngr
  ON e.mngr_id = emp_mngr.id
```



SQL: Self JOIN - Example #84 - A:

id	name	mngr_id
1001	Dana	1001
1002	David	1001
1003	Eli	1001
1004	Sara	1003
1005	Ben	1003

id	customer	priorty
10	Bezeq	3
11	Toyota	3
12	Angel	2
13	Tesla	2
14	IBM	1
15	Osem	1

emp_id	proj_id	hours
1001	10	50
1001	11	40
1001	12	60
1002	10	110
1002	13	140
1003	11	120
1004	14	190
1009	14	190

```
    SELECT e.id,
e.name emp_name,
emp_mngr.name
manager_name
      FROM
employees e
      JOIN
employees emp_mngr
       ON
e.mngr_id =
emp_mngr.id
```

id	emp_name	manager_name
1001	Dana	Dana
1002	David	Dana
1003	Eli	Dana
1004	Sara	Eli
1005	Ben	Eli

We can choose meaningful columns and give names to our results