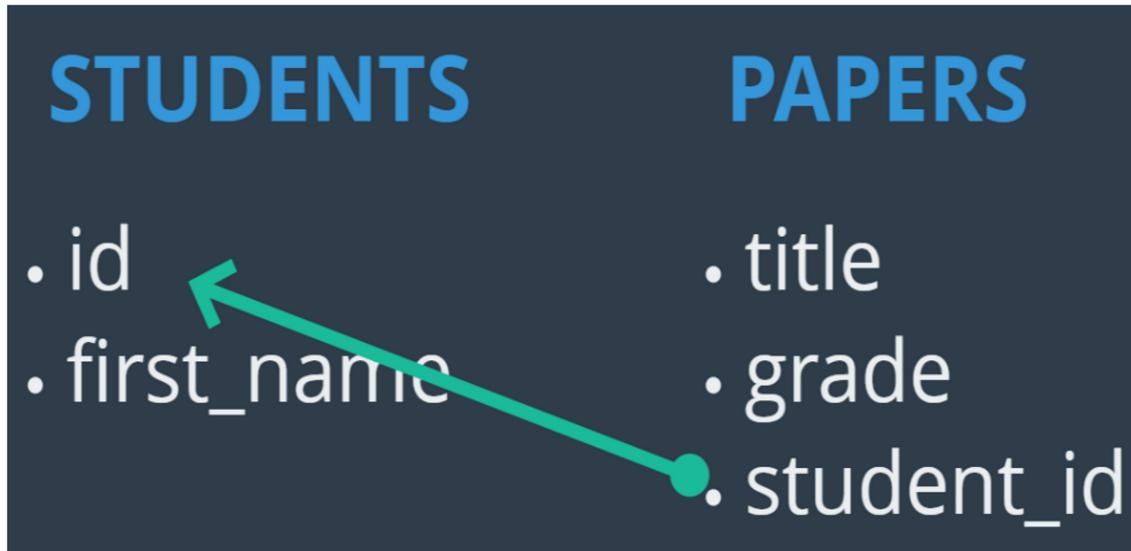


MySQL - Relationships & Joins - p2 - Lecture 23

Write this Schema



```
CREATE TABLE Students(
    id int auto_increment,
    first_name varchar(50),
    PRIMARY KEY(id)
);
```

```
CREATE TABLE Papers(
    title varchar(50),
    grade char(1),
    student_id int not null,
    foreign key(student_id) references Students(id)
);
```

```
mysql> CREATE TABLE Students(
->   id int auto_increment,
->   first_name varchar(50),
->   PRIMARY KEY(id)
-> );
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> DESC Students;
+-----+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra       |
+-----+-----+-----+-----+-----+-----+
| id         | int        | NO   | PRI | NULL    | auto_increment |
| first_name | varchar(50) | YES  |     | NULL    |               |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.02 sec)
```

```

mysql> CREATE TABLE Papers(
->   title varchar(50),
->   grade char(1),
->   student_id int not null,
->   foreign key(student_id) references Students(id)
-> );
Query OK, 0 rows affected (0.06 sec)

mysql> DESC Papers;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| title      | varchar(50) | YES  |     | NULL    |       |
| grade      | char(1)    | YES  |     | NULL    |       |
| student_id | int        | NO   | MUL | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)

```

```

INSERT INTO students (first_name) VALUES
('Caleb'), ('Samantha'), ('Raj'), ('Carlos'), ('Lisa');

INSERT INTO papers (student_id, title, grade ) VALUES
(1, 'My First Book Report', 60),
(1, 'My Second Book Report', 75),
(2, 'Russian Lit Through The Ages', 94),
(2, 'De Montaigne and The Art of The Essay', 98),
(4, 'Borges and Magical Realism', 89);

```

```

mysql> ALTER TABLE Papers MODIFY COLUMN grade varchar(3);
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0

```

```

mysql> INSERT INTO students (first_name) VALUES
-> ('Caleb'), ('Samantha'), ('Raj'), ('Carlos'), ('Lisa');
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

```

```

mysql> INSERT INTO papers (student_id, title, grade ) VALUES
-> (1, 'My First Book Report', 60),
-> (1, 'My Second Book Report', 75),
-> (2, 'Russian Lit Through The Ages', 94),
-> (2, 'De Montaigne and The Art of The Essay', 98),
-> (4, 'Borges and Magical Realism', 89);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

```

```

mysql> SELECT * FROM Students;
+---+-----+
| id | first_name |
+---+-----+
| 1 | Caleb
| 2 | Samantha
| 3 | Raj
| 4 | Carlos
| 5 | Lisa
+---+-----+
5 rows in set (0.00 sec)

mysql> SELECT * FROM Papers;
+-----+-----+-----+
| title | grade | student_id |
+-----+-----+-----+
| My First Book Report | 60 | 1 |
| My Second Book Report | 75 | 1 |
| Russian Lit Through The Ages | 94 | 2 |
| De Montaigne and The Art of The Essay | 98 | 2 |
| Borges and Magical Realism | 89 | 4 |
+-----+-----+-----+
5 rows in set (0.00 sec)

```

Challenge 1: Print this:

first_name	title	grade
Samantha	De Montaigne and The Art of The Essay	98
Samantha	Russian Lit Through The Ages	94
Carlos	Borges and Magical Realism	89
Caleb	My Second Book Report	75
Caleb	My First Book Report	60

```

mysql> SELECT first_name, title, grade
    -> FROM Students JOIN Papers
    -> ON Students.id = Papers.student_id
    -> ORDER BY GRADE DESC;
+-----+-----+-----+
| first_name | title | grade |
+-----+-----+-----+
| Samantha | De Montaigne and The Art of The Essay | 98 |
| Samantha | Russian Lit Through The Ages | 94 |
| Carlos | Borges and Magical Realism | 89 |
| Caleb | My Second Book Report | 75 |
| Caleb | My First Book Report | 60 |
+-----+-----+-----+
5 rows in set (0.00 sec)

```

Challenge 2: Print this:

first_name	title	grade
Caleb	My First Book Report	60
Caleb	My Second Book Report	75
Samantha	Russian Lit Through The Ages	94
Samantha	De Montaigne and The Art of The Essay	98
Raj	NULL	NULL
Carlos	Borges and Magical Realism	89
Lisa	NULL	NULL

Whenever you find NULL values on right side of your resultant table. You have to use "Left Join".

```
mysql> SELECT first_name, title, grade  
    -> FROM Students LEFT JOIN Papers  
    -> ON Students.id = Papers.student_id;
```

first_name	title	grade
Caleb	My First Book Report	60
Caleb	My Second Book Report	75
Samantha	Russian Lit Through The Ages	94
Samantha	De Montaigne and The Art of The Essay	98
Raj	NULL	NULL
Carlos	Borges and Magical Realism	89
Lisa	NULL	NULL

7 rows in set (0.00 sec)

Challenge 3: Print this:

first_name	title	grade
Caleb	My First Book Report	60
Caleb	My Second Book Report	75
Samantha	Russian Lit Through The Ages	94
Samantha	De Montaigne and The Art of The Essay	98
Raj	MISSING	0
Carlos	Borges and Magical Realism	89
Lisa	MISSING	0

`IFNULL(title,"Missing")`

```
mysql> SELECT first_name,
-> IFNULL(title,"MISSING") AS Title,
-> IFNULL(grade,"0") AS Grade
-> FROM Students LEFT JOIN Papers
-> ON Students.id = Papers.student_id;
+-----+-----+
| first_name | Title      | Grade |
+-----+-----+
| Caleb     | My First Book Report | 60   |
| Caleb     | My Second Book Report | 75   |
| Samantha  | Russian Lit Through The Ages | 94   |
| Samantha  | De Montaigne and The Art of The Essay | 98   |
| Raj       | MISSING        | 0    |
| Carlos    | Borges and Magical Realism | 89   |
| Lisa      | MISSING        | 0    |
+-----+-----+
7 rows in set (0.04 sec)
```

Challenge 4: Print this:

```
+-----+-----+
| first_name | average |
+-----+-----+
| Samantha  | 96.0000 |
| Carlos    | 89.0000 |
| Caleb     | 67.5000 |
| Raj       | 0         |
| Lisa      | 0         |
+-----+-----+
```

```
mysql> SELECT first_name,
-> AVG(IFNULL(grade,0)) AS average
-> FROM Students LEFT JOIN Papers
-> ON Students.id = Papers.student_id
-> GROUP BY first_name
-> ORDER BY average DESC;
+-----+-----+
| first_name | average |
+-----+-----+
| Samantha  | 96   |
| Carlos    | 89   |
| Caleb     | 67.5 |
| Raj       | 0    |
| Lisa      | 0    |
+-----+-----+
5 rows in set (0.01 sec)

mysql> SELECT first_name,
-> AVG(IFNULL(grade,0)) AS average
-> FROM Students LEFT JOIN Papers
-> ON Students.id = Papers.student_id
-> GROUP BY 1 ORDER BY 2 DESC;
+-----+-----+
| first_name | average |
+-----+-----+
| Samantha  | 96   |
| Carlos    | 89   |
| Caleb     | 67.5 |
| Raj       | 0    |
| Lisa      | 0    |
+-----+-----+
5 rows in set (0.01 sec)
```

Challenge 5: Print this:

first_name	average	passing_status
Samantha	96.0000	PASSING
Carlos	89.0000	PASSING
Caleb	67.5000	FAILING
Raj	0	FAILING
Lisa	0	FAILING

CASE:
When average > 75
[Pass]

```
mysql> SELECT first_name,
-> AVG(IFNULL(grade,0)) AS average,
-> CASE
-> WHEN AVG(IFNULL(grade,0)) > 75 THEN "Passing"
-> ELSE "Failing"
-> END AS passing_status
-> FROM Students LEFT JOIN Papers
-> ON Students.id = Papers.student_id
-> GROUP BY 1 ORDER BY 2 DESC;
+-----+-----+-----+
| first_name | average | passing_status |
+-----+-----+-----+
| Samantha | 96 | Passing
| Carlos | 89 | Passing
| Caleb | 67.5 | Failing
| Raj | 0 | Failing
| Lisa | 0 | Failing
+-----+-----+-----+
5 rows in set (0.02 sec)
```

```
SELECT first_name,
AVG(IFNULL(grade,0)) AS average,
CASE
    WHEN AVG(IFNULL(grade,0)) > 75 THEN "Passing"
    ELSE "Failing"
END AS passing_status
FROM Students LEFT JOIN Papers
ON Students.id = Papers.student_id
GROUP BY 1 ORDER BY 2 DESC;
```