

# DBMS Lab 7-8

Ridhima Kohli  
B19CSE071

## Insertion commands for data

```
INSERT INTO `book_purchase_details`(`Book_ID`, `Purchase_Dt`, `Copies`, `Price`, `Purchase_Price`, `Book_Value`) VALUES
('Da001_Sel','2021-09-07', 5, 50 , NULL , NULL);
INSERT INTO `book_purchase_details`(`Book_ID`, `Purchase_Dt`, `Copies`, `Price`, `Purchase_Price`, `Book_Value`) VALUES
('Mi009_Emo','2021-09-02', 2, 50 , NULL , NULL);
INSERT INTO `book_purchase_details`(`Book_ID`, `Purchase_Dt`, `Copies`, `Price`, `Purchase_Price`, `Book_Value`) VALUES
('Mi009_Soc','2021-09-01', 2, 50 , NULL , NULL);
INSERT INTO `book_purchase_details`(`Book_ID`, `Purchase_Dt`, `Copies`, `Price`, `Purchase_Price`, `Book_Value`) VALUES
('Ra001_Pha','2021-09-02', 3, 50 , NULL , NULL);
INSERT INTO `book_purchase_details`(`Book_ID`, `Purchase_Dt`, `Copies`, `Price`, `Purchase_Price`, `Book_Value`) VALUES
('Ro015_Fan','2021-09-01', 12, 50 , NULL , NULL);
INSERT INTO `book_purchase_details`(`Book_ID`, `Purchase_Dt`, `Copies`, `Price`, `Purchase_Price`, `Book_Value`) VALUES
('Ro015_Fan','2021-09-09', 3, 50 , NULL , NULL);
INSERT INTO `book_purchase_details`(`Book_ID`, `Purchase_Dt`, `Copies`, `Price`, `Purchase_Price`, `Book_Value`) VALUES
('Da001_Sel','2021-09-07', 3, 50 , NULL , NULL);
```

Author\_Details:

Author_ID	Author_Name
Da_001	Damasio
Mi_009	Minsky
Ra_001	Ramachandran
Ro_015	Rowling
Ru_021	Russel
Sa_001	Safina
Ta_001	Tagore
To_015	Tolkien
Wo_015	Wodehouse

## Book\_Details:

Author_ID	Book_ID	Book
Da_001	Mi_009_Sel	Self Comes to Mind
Mi_009	Mi009_Emo	Emotion Machine
Mi_009	Mi009_Soc	Society of Mind
Ra_001	Ra001_Pha	Phantoms in the Brain
Ro_015	Ro015_Pri	Prisoner of Azkaban_Harry Potter
Ro_015	Ro015_Phi	Philosopher's Stone_Harry Potter
Ro_015	Ro015_Gob	Goblet of Fire_Harry Potter
Ro_015	Ro015_Fan	Fantastic Beasts and Where to Find Them
Sa_001	Sa001_Voy	Voyage of the Turtle
Sa_001	Sa001_Wha	What Animals Think
To_015	To015_Fel	Fellowship of the Rings_Lord of the Rings
Wo_015	Wo015_Wod	Wodehouse at the Wicket

## Book\_Purchase\_Details:

	Book_ID	Purchase_Dt	Copies	Price	Purchase_Price	Book_Value
a	Da001_Sel	2021-09-01	1	50	NULL	NULL
a	Da001_Sel	2021-09-07	5	50	NULL	NULL
a	Mi009_Emo	2021-09-02	2	50	NULL	NULL
a	Mi009_Soc	2021-09-01	2	50	NULL	NULL
a	Ra001_Pha	2021-09-02	2	50	NULL	NULL
a	Ro015_Fan	2021-09-01	3	50	NULL	NULL
a	Ro015_Fan	2021-09-09	12	35	NULL	NULL
a	Ro015_Gob	2021-09-01	3	50	NULL	NULL
a	Ro015_Phi	2021-09-01	3	50	NULL	NULL
a	Ro015_Phi	2021-09-10	20	75	NULL	NULL
a	Ro015_Pri	2021-09-01	3	50	NULL	NULL
a	Sa001_Voy	2021-09-02	2	50	NULL	NULL
a	Sa001_Wha	2021-09-02	2	50	NULL	NULL
a	To015_Fel	2021-09-01	3	50	NULL	NULL
a	To015_Fel	2021-09-12	9	55	NULL	NULL
a	Wo015_Wod	2021-09-05	1	50	NULL	NULL

Create a view comprising Author\_name, Book\_Name and Copies; Group by Book\_Name

```
CREATE VIEW parta AS
SELECT A.Author_Name , B.Book , SUM(P.Copies)
FROM author_details A ,book_details B , book_purchase_details P
WHERE A.Author_ID=B.Author_ID AND B.Book_ID=P.Book_ID
GROUP BY B.Book
```



Author_Name	Book	SUM(P.Copies)
Minsky	Emotion Machine	2
Rowling	Fantastic Beasts and Where to Find Them	15
Tolkien	Fellowship of the Rings_Lord of the Rings	12
Rowling	Goblet of Fire_Harry Potter	3
Ramachandran	Phantoms in the Brain	2
Rowling	Philosopher's Stone_Harry Potter	23
Rowling	Prisoner of Azkaban_Harry Potter	3
Damasio	Self Comes to Mind	6
Minsky	Society of Mind	2
Safina	Voyage of the Turtle	2
Safina	What Animals Think	2
Wodehouse	Wodehouse at the Wicket	1

## **Observation through mistake !**

During data entry I had entered the book\_id for self comes to mind wrong in Book\_ID table , due to which the view didn't display the book

After discovering the mistake , I corrected the entry and the view was updated automatically.

- c) Can you insert entries into the above view?
- d) Can you delete entries from the above view?

we cannot add or delete entries in the view


The view we have created here is read only , which is **not updatable**

Since here the view we have created has where and group by clause , hence we cannot insert/delete from view

### Error

SQL query: [Copy](#)

```
INSERT INTO `parta` (`Author_Name`, `Book`, `Total_Copies`) VALUES ('Ri_007','Nam_003',6);
```


MySQL said: 

#1471 - The target table parta of the INSERT is not insertable-into

### Error

SQL query: [Copy](#)

```
DELETE FROM `parta` WHERE Total_Copies = 12;
```

MySQL said: 

#1288 - The target table parta of the DELETE is not updatable



**Test the difference in retrieval time of the data in (a) post execution of the SQL and that from the created view. Note any observations made.**

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0132 seconds.)

```
CREATE VIEW parta AS SELECT A.Author_Name , B.Book , SUM(P.Copies) as Total_Copies FROM author_details A ,book_details B , book_purchase_details P WHERE A.Author_ID=B.Author_ID AND B.Book_ID=P.Book_ID GROUP BY B.Book;
```

✓ Showing rows 0 - 11 (12 total, Query took 0.0015 seconds.)

```
SELECT * FROM parta;
```

✓ Showing rows 0 - 11 (12 total, Query took 0.0035 seconds.)

```
SELECT A.Author_Name , B.Book , SUM(P.Copies) FROM author_details A ,book_details B , book_purchase_details P WHERE A.Author_ID=B.Author_ID AND B.Book_ID=P.Book_ID GROUP BY B.Book;
```

The select query takes significantly lesser time than VIEW creation+view retrieval , however as observed , VIEWS can get updated on their own so for multiple updates views can be better for execution

If we need only temporary results for few times then simple select query should be used but for long run , once a view is made , its retrieval time is less than simple select as it acts like cache

# What happens to the view if you insert or delete tuples from the main database?

On inserting or deleting tuples from main database the view gets updated accordingly if the updated tuples match the view criteria  
As shown :

Author_Name	Book	Total_Copies
Minsky	Emotion Machine	2
Rowling	Fantastic Beasts and Where to Find Them	15
Tolkien	Fellowship of the Rings_Lord of the Rings	12
Rowling	Goblet of Fire_Harry Potter	3
Ramachandran	Phantoms in the Brain	2
Rowling	Philosopher's Stone_Harry Potter	23
Rowling	Prisoner of Azkaban_Harry Potter	3
Damasio	Self Comes to Mind (new inserted)	8
Minsky	Society of Mind	2
Safina	Voyage of the Turtle	2
Safina	What Animals Think	2
Wodehouse	Wodehouse at the Wicket	1



Author_Name	Book	Total_Copies
Rowling	Philosopher's Stone_Harry Potter	23
Rowling	Fantastic Beasts and Where to Find Them	15
Tolkien	Fellowship of the Rings_Lord of the Rings	12
Rowling	Prisoner of Azkaban_Harry Potter	3
Rowling	Goblet of Fire_Harry Potter	3
Safina	What Animals Think	2
Safina	Voyage of the Turtle	2
Minsky	Society of Mind	2
Ramachandran	Phantoms in the Brain	2
Minsky	Emotion Machine	2
Wodehouse	Wodehouse at the Wicket	1

View updated after self comes to mind was deleted from book\_details

View updated after inserting self comes to mind (new insertion) to book\_details


## Can you update the view?

As mentioned earlier , the given view we have created is not updatable

### Error

SQL query: [Copy](#)

```
UPDATE `parta` SET `Book`='Society of Mind2' WHERE `Author_Name`='Minsky';
```

MySQL said: 

#1288 - The target table parta of the UPDATE is not updatable

However if we create a different view from a single table with **no group by or where conditions** then we can update it as shown

**Book ID Ro015\_Hal is not in base table but is inserted into view2 View**

Author_ID	Book_ID
Da_001	Da001_Sel
Mi_009	Mi009_Emo
Mi_009	Mi009_Soc
Ra_001	Ra001_Pha
Ro_015	Ro015_Fan
Ro_015	Ro015_Gob
Ro_015	Ro015_Phi
Ro_015	Ro015_Pri
Sa_001	Sa001_Voy
Sa_001	Sa001_Wha
To_015	To015_Fel
Wo_015	Wo015_Wod

✓ 1 row inserted. (Query took 0.0077 seconds.)

```
INSERT INTO `view2`(`Author_ID`, `Book_ID`) VALUES ('Ro_015','Ro015_Hal');
```

[Edit inline](#) ] [\[ Edit \]](#) [\[ Create PHP code \]](#)

Author_ID	Book_ID
Da_001	Da001_Sel
Mi_009	Mi009_Emo
Mi_009	Mi009_Soc
Ra_001	Ra001_Pha
Ro_015	Ro015_Fan
Ro_015	Ro015_Gob
Ro_015	Ro015_Hal
Ro_015	Ro015_Phi
Ro_015	Ro015_Pri
Sa_001	Sa001_Voy
Sa_001	Sa001_Wha
To_015	To015_Fel
Wo_015	Wo015_Wod

Create a trigger to calculate the Purchase\_Price of the books on the date of purchase

```
CREATE TRIGGER purchTrig BEFORE INSERT ON
book_purchase_details
FOR EACH ROW
SET NEW.Purchase_Price = NEW.Price*New.Copies;
```

✔ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0148 seconds.)

```
CREATE TRIGGER purchTrig BEFORE INSERT ON book_purchase_details FOR EACH ROW SET NEW.Purchase_Price = NEW.Price*New.Copies;
```

Now on new insertion :

```
INSERT INTO `book_purchase_details`(`Book_ID`, `Purchase_Dt`, `Copies`, `Price`,
`Purchase_Price`, `Book_Value`) VALUES ('Wo015_Wod','2021-09-06', '3', '50',NULL,NULL)
```

Book ID	Purchase Dt	Copies	Price	Purchase Price	Book Value
Wo015_Wod	2021-09-06	1	50	NULL	NULL
Wo015_Wod	2021-09-08	3	50	150	NULL

The trigger calculates purchase price of new entry

View is updated automatically too

Tolkien	Fellowship of the Rings_Lord of the Rings	12
Damasio	Self Comes to Mind (new inserted)	6
Wodehouse	Wodehouse at the Wicket	4
Rowling	Prisoner of Azkaban_Harry Potter	3

**h) Is the above an example of a DDL or a DML trigger**

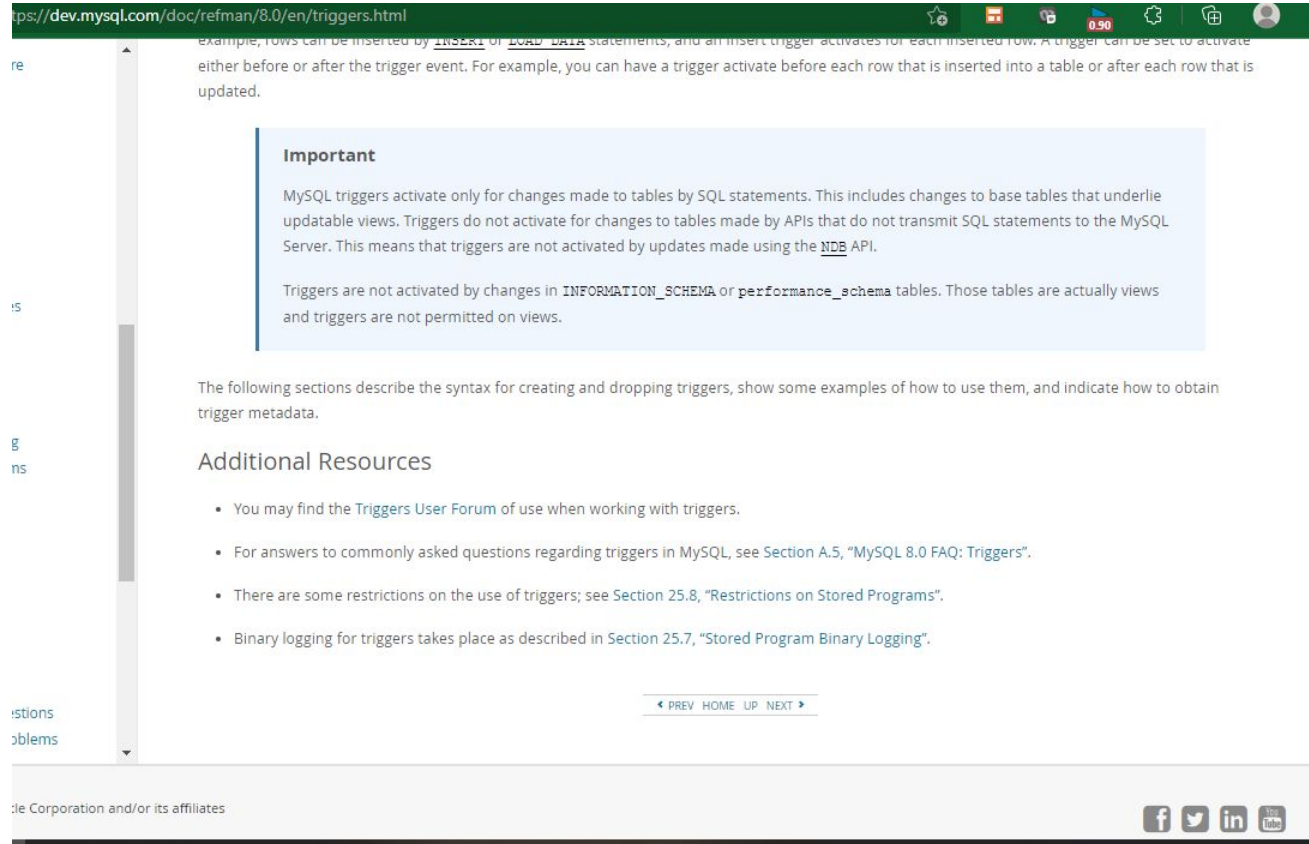
Since the data of table is being manipulated here or we can say that data is being updated after INSERT ( a DML command ) ,instead of change in table structure , hence the above is an example of DML trigger

i) Can we create triggers on views? If yes, frame an example using the given scenario, if not justify your answer

Yes , triggers on views can be created using INSTEAD OF keyword in dbms

However , MySQL doesn't have INSTEAD OF and it doesn't support triggers on views

As mentioned in the documentation : [MySQL :: MySQL 8.0 Reference Manual :: 25.3 Using Triggers](#)



The screenshot shows the MySQL 8.0 documentation page for triggers. The URL in the browser is <https://dev.mysql.com/doc/refman/8.0/en/triggers.html>. The page content includes an introduction to triggers, an 'Important' section, and a list of additional resources.

example, rows can be inserted by `INSERT` or `LOAD DATA` statements, and an insert trigger activates for each inserted row. A trigger can be set to activate either before or after the trigger event. For example, you can have a trigger activate before each row that is inserted into a table or after each row that is updated.

**Important**

MySQL triggers activate only for changes made to tables by SQL statements. This includes changes to base tables that underlie updatable views. Triggers do not activate for changes to tables made by APIs that do not transmit SQL statements to the MySQL Server. This means that triggers are not activated by updates made using the [NDB](#) API.

Triggers are not activated by changes in `INFORMATION_SCHEMA` or `performance_schema` tables. Those tables are actually views and triggers are not permitted on views.

The following sections describe the syntax for creating and dropping triggers, show some examples of how to use them, and indicate how to obtain trigger metadata.

**Additional Resources**

- You may find the [Triggers User Forum](#) of use when working with triggers.
- For answers to commonly asked questions regarding triggers in MySQL, see [Section A.5, "MySQL 8.0 FAQ: Triggers"](#).
- There are some restrictions on the use of triggers; see [Section 25.8, "Restrictions on Stored Programs"](#).
- Binary logging for triggers takes place as described in [Section 25.7, "Stored Program Binary Logging"](#).

Navigation links: [PREV](#) [HOME](#) [UP](#) [NEXT](#)

© 2019 MySQL Corporation and/or its affiliates

Social media icons: Facebook, Twitter, LinkedIn, YouTube



## j) What are DDL triggers?

DDL triggers - Data Definition Language Triggers are fired when there is a change in data definition of the table/database , like a CREATE or DROP command

## k) Can you think of examples of DDL triggers for the given scenario?

Example:

In case we want to keep the field constraints fixed , that is , the data fields' constraints shouldn't be changed , then we can use DDL triggers

For example , if someone tries to set Purchase\_Price to NOT NULL constraint , we can create a DDL trigger which fires on ALTER command on Book\_Purchase\_Details table and restricts the alter as well as displays a warning message : Table alteration is not allowed

```
CREATE TRIGGER stopAlterTrig
ON Book_Purchase_Details
FOR ALTER_TABLE AS
BEGIN
    PRINT 'Table alteration is not allowed'
    ROLLBACK TRANSACTION
END
```

I) Create a view/trigger comprising Author\_name, Book\_Name, Book\_Value, where Book\_Value derives from the Purchase\_Price as on Purchase\_Dt and Copies purchased

**Assumption made by me for Calculating book value :**

Here we are considering that the book is being rented for 10% profit on purchase price (which is itself calculated from a trigger ) and hence 10% of Purchase Price is added

```
CREATE TRIGGER bookValTrig BEFORE INSERT ON  
book_purchase_details  
FOR EACH ROW  
SET new.Book_Value = new.Purchase_Price +  
0.1*(new.Purchase_Price)
```

Book Value  
added on  
insert due to  
trigger



e	Wo015_Wod	2021-09-12	3	20	60	66
---	-----------	------------	---	----	----	----

m) Calculate the total money the library has spent for purchasing all books using triggers. What type of trigger are you going to use for this ?

We create a table `total_all_table` with only one column and one row - total , which will keep on adding purchase values on trigger . We use **after insert trigger** here too which is a **DML trigger**

✓ 1 row inserted. (Query took 0.0076 seconds.)

```
INSERT INTO `book_purchase_details` (`Book_ID`, `Purchase_Dt`, `Copies`, `Price`, `Purchase_Price`, `Book_Value`) VALUES ('Mi009_Soc', '2021-10-04', 5, 200, 0, 0);
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

```
DELIMITER #
CREATE TRIGGER TotPriceTrig
AFTER INSERT ON book_purchase_details
FOR EACH ROW
BEGIN
    UPDATE total_all_table
    SET Total = Total + New.Purchase_Price;
END#
```



n) Calculate the total money the library has spent for purchasing each book using triggers.  
What type of trigger are you going to use for this ?

Similar to previous , we create a table and update the book wise prices  
We use **AFTER trigger** here which is a **DML trigger**

```
DELIMITER #  
CREATE TRIGGER bookPriceTrig  
AFTER INSERT ON book_purchase_details  
FOR EACH ROW  
BEGIN  
    UPDATE purchase_values_by_books_table  
    SET Price_Tot = Price_Tot + New.Purchase_Price  
    WHERE purchase_values_by_books_table.Book_ID =  
    New.Book_ID;  
END#
```



Book_ID	Price_Tot
Da001_Sel	320
Mi009_Emo	0
Mi009_Soc	0
Ra001_Pha	0
Ro015_Fan	0
Ro015_Gob	0
Ro015_Phi	0
Ro015_Pri	0
Sa001_Voy	0
Sa001_Wha	0
To015_Fel	0
Wo015_Wod	710

Book_ID	Price_Tot
Da001_Sel	420
Mi009_Emo	0
Mi009_Soc	0
Ra001_Pha	0
Ro015_Fan	0
Ro015_Gob	0
Ro015_Phi	0
Ro015_Pri	0
Sa001_Voy	0
Sa001_Wha	0
To015_Fel	0
Wo015_Wod	710

Trigger updates new table with insertion

✓ 1 row inserted. (Query took 0.0066 seconds.)

```
INSERT INTO `book_purchase_details`(`Book_ID`, `Purchase_Dt`, `Copies`, `Price`, `Purchase_Price`, `Book_Value`) VALUES ('Da001_Sel','2021-10-2','4','25',0,0);
```

[\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Create PHP code \]](#)

o) Can you use both 'After' and 'Before' triggers together? Show with the help of an example derived from the given scenario.

Yes , both after and before triggers can be used

we have already done this in the previous parts  
We calculated purchase price and book value  
using **before triggers** and applied **after trigger**  
on same values for calculating total prices

