Assignment 3
Class: SE Comp
Year: 2020-21
Performed by: Danyl Fernandes, 72

Consider the employee database:

employee (employee_name, street, city,date_of_joining)
works (employee_name, company_name, salary)
company (company_name, city)
manages (employee_name, manager_name)

Write SQL queries for the following statements:

1) Find all the employees who joined in the month of october:

Query:

```
select employee_name as EmployeesJoinedInOctober from employee
where date_of_joining like '%OCT%';
```

2) Modify the database so that ‘Anjali’ now lives in ‘Mumbai’:

Query:

```
update employee set city
     = 'Mumbai' where employee_name = 'Anjali';
```

3) List all the employees who live in the same cities as their managers:

Query:

```
select e1.employee_name as EmployeesWithManagersResidence from
employee as e1, employee as e2 where (e1.employee_name) in
(select m.employee_name from manages as m where e1.employee_name
= m.employee_name and e2.employee_name = m.manager_name and
e1.street = e2.street and e1.city = e2.city);
```

4) Find all employees who earn more than the average salary of all the employees of their company:

Query:

```
select w1.employee_name as EmployeesPaidHigherThanAverageSalary
from works w1 where w1.salary > (select avg(w2.salary) from
works w2 where w2.company_name = w1.company_name);
```

5) Give all the employees of ABC Corporation a 15 percent raise:

Query:

```
update works set salary = salary * 1.15 where (works.cid) in
(select cid from company where cname = 'ABC Corporation');
```

Book(book_id, title,author, cost)
Store(store_no, city, state, inventory_val)
Stock(store_no, book_id,quantity)

1) Modify the cost of DBMS books by 10%:

Query:

```
update Book set cost = cost * 1.1 where title = 'DBMS';
```

2) Find the author of the books which are available in Mumbai store:

Query:

```
select author as AuthorsOfBooksInMumbai from Book where (boo_id)
in (select book_id from Stock where (store_no) in (select
store_no from store where city = 'Mumbai'));
```

3) Find the title of the most expensive book:

Query:

```
select title as MostExpensiveBook from book where cost = (select
max(cost) from book);
```

4) Find the total quantity of books in each store:

Query:

```
select store_no as StoreNumber, sum(quantity) as TotalQuantity
from stock group by store_no;
```

5) Add a new record in Book (Assume values as per requirement):

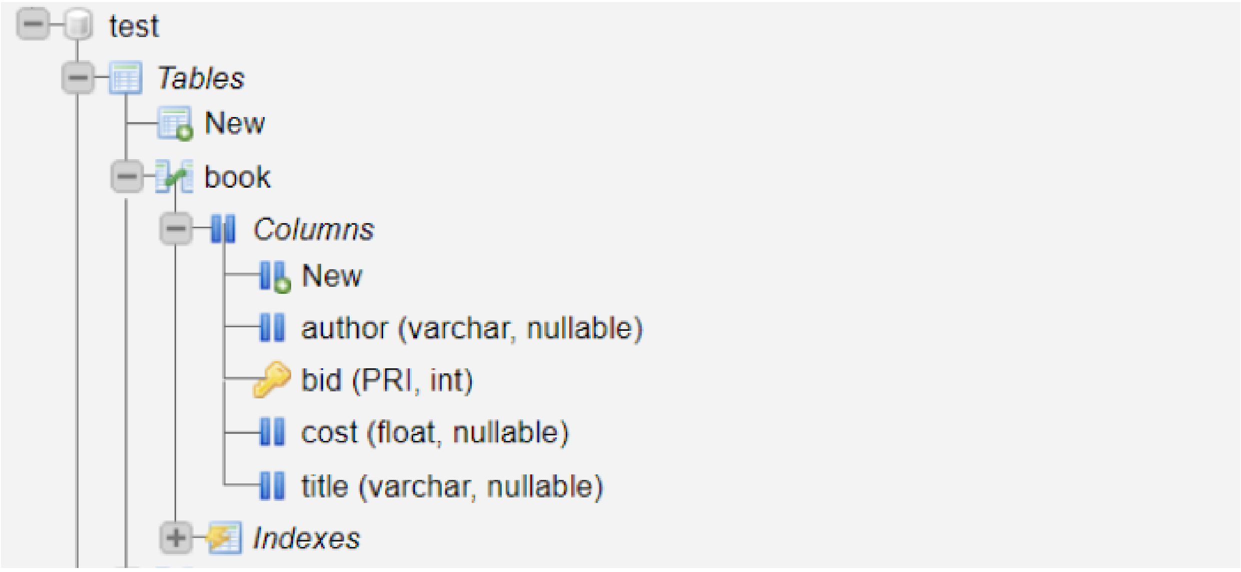
Query:

```
insert into Book values (101, 'Outliers', 'Malcolm Gladwell',
500);
```

3. What is the view in SQL, how it is defined? Analyze and find whether View exists if the table is dropped from the database also analyze the effect of updation, insertion and deletion on Views and base table and comment on the same.

View :

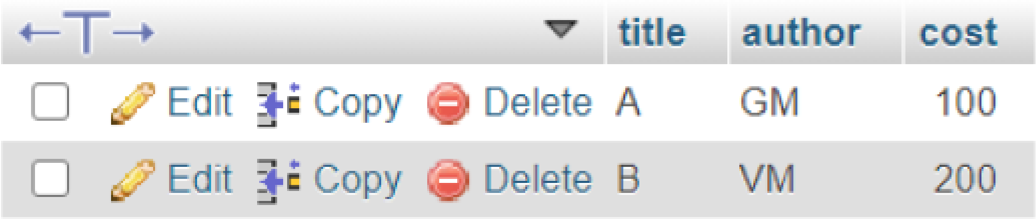
- A view is a database object that has no values.
- It's contents are based on the base table, meaning it contains rows and columns similar to an actual table.
- In MySQL, a View is a virtual table created by a query by joining one or more tables.
- It is operated similarly to the base table but does not contain any data of its own.
- A View and a Table have one main difference that the views are definitions built on top of other tables (or views).
- Analysis: Consider a database containing the following table:



test
Tables
New
book
Columns
New
author (varchar, nullable)
bid (PRI, int)
cost (float, nullable)
title (varchar, nullable)
Indexes

Defining a View:

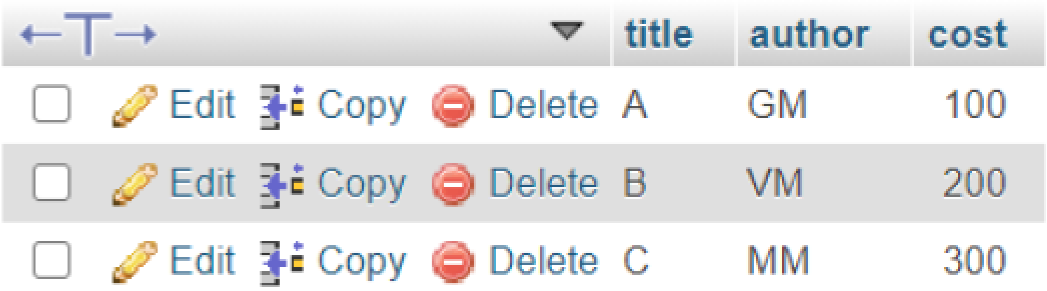
```
create view BookDetails as select title, author, cost from book;
select * from bookdetails;
```



←T→ ▼	title	author	cost
□ Edit Copy Delete	A	GM	100
□ Edit Copy Delete	B	VM	200

Effect on Insertion:

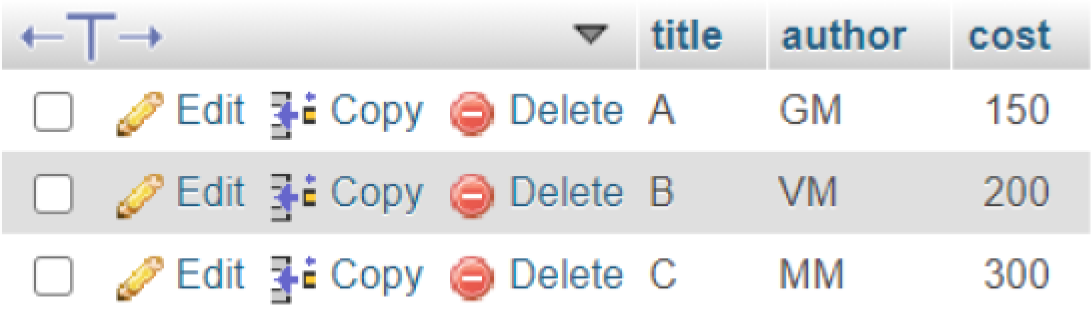
```
insert into book values (103, 'C', 'MM', 300);
select * from bookdetails;
```



←T→			▼	title	author	cost
☐	Edit	Copy	Delete	A	GM	100
☐	Edit	Copy	Delete	B	VM	200
☐	Edit	Copy	Delete	C	MM	300

Effect on Updation:

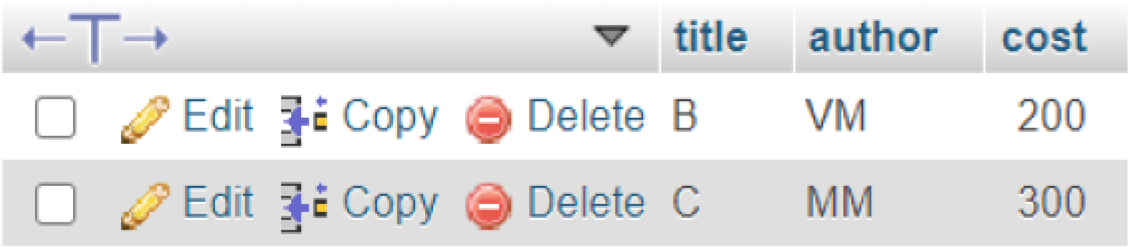
```
update book set cost = 150 where author = 'GM';
select * from bookdetails;
```



←T→			▼	title	author	cost
☐	Edit	Copy	Delete	A	GM	150
☐	Edit	Copy	Delete	B	VM	200
☐	Edit	Copy	Delete	C	MM	300

Effect on Deletion:

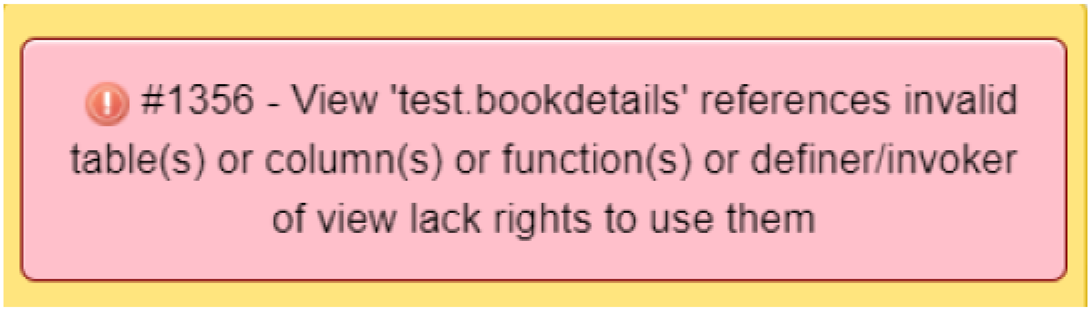
```
delete from book where author = 'GM';
select * from bookdetails;
```



←T→			▼	title	author	cost
☐	Edit	Copy	Delete	B	VM	200
☐	Edit	Copy	Delete	C	MM	300

Effect on Table Drop:

```
drop table book;
select * from bookdetails;
```



#1356 - View 'test.bookdetails' references invalid table(s) or column(s) or function(s) or definer/invoker of view lack rights to use them

- Hence, it can be concluded that if any changes occur in the underlying table, the same changes are reflected in the View.
- If the underlying table is dropped, the user loses rights to use the formerly created View.
- Significance:
  - Allows simplification of complex queries
  - Increases code reusability
  - Provides additional data security
  - Enables backward compatibility.

Person (driver_id, name , address)
Car (license, model, year)
Accident (report_no, adate, location)
Owns (driver_id, license)
Participated (driver_id, license, report_no, damage_amount)

1) Find the total number of people who owned cars that are involved in accidents in 2004:

Query:

```
select count(*) as NumberOfPeopleInvolvedIn2004CarAccidents from
Owns where (driver_id) in (select driver_id from Participated
where (report_no) in (select report_no from Accident where adate
like '%2004%'));
```

2) Find the number of accidents in which car belonging to 'John Smith' were involved:

Query:

```
select count(*) from Participated where (driver_id) in (select
driver_id from Person where name = 'John Smith');
```

3) Add a new accident to the database insert:

Query:

```
insert into Accident values (101, 'AUG 14, 2021', 'Mumbai');
```

4) Delete 'Santro' belonging to 'John Smith':

Query:

```
delete from car where model = 'Santro' and license = (select
from license from Owns where driver_id = (select driver_id from
Person where name = 'John Smith'));
```