**CREATE**

1. STUDENT TABLE:

2. COURSE TABLE:

3. INSTRUCTOR TABLE:

4. ENROLLMENT TABLE:

**INSERT 30 VALUES TO ALL THE TABLES**

**PERFORM THESE QUERIES**

1. unique enrollment count i.e how many users purchased my course

2. retrieve the course name not enrolled by student

3. courseId,course\_name,course\_branch,instructorId,first name

4. student\_details with courseId

5. display the student based on no max course they purchased

6. rank for that student based on course fees

**EXPLANATIONS:-**

STUDENT TABLE:

studentId

first name

last name

email

yoj

COURSE TABLE:

courseId

courseName

branches

course fees

INSTRUCTOR table:

instructorId

first name

last name

email

branches

ENROLLMENT table:

enrollmentId

studentId - from student table

courseId - from COURSE table

enrollmentDate

truncate table tableName; -- delete the table data without logs

revoke all privileges , grant option from 'root'@'localhost';

start transaction;

insert - if insertion is wrong

rollback;

commit;

**queries need to be performed for STORE DATA:**

STORE DATA needs to be imported from this EXCEL:

1. select the data who is making profit

2. select the data who is making profit with discount >= 0.5

3. select unique customerID

4. Take the categories & subcategories in where this unique customerID present

5. select most profit making city

6. create a new table duplicate TABLE

7. In new table delete the rows whos discount is < 0.3

8. find what category is saled most

9. find which ship mode made most profit

10. GET which subcategories quantity is high

11. select the rank of 21-31 most sales record

12. create a new column combining category & subcategories ex category = 'Office Supplies' subcategories = 'Binders' newColumn = 'Office Supplies-Binders'

13. select the data ordered after 8/3/2015 and before 1/10/2017

14. find the most used customerID

15. create a new column having customer\_name\_lenght

16. how many unique orders created

17. what orderID has the most sales

18. Rank the orderID based on the sales, grouped on city

19. find the window function-sum based on partition by date

20. productID sales

**For DataSet kindly refer to the csv file attached in drive.**