CREATE TABLE:

1. Create new database library and create table book\_dtl.

(Which will show book\_id, book\_name, author\_name, num\_pages, price)

2. Enter following data in above table.

102,Dasbodh,Ramdas Swami,504,2000

103,Agnipankh,APJ Abdul Kalam,244,200

104,Let Us C,Yashvant Kanitkar,250,300

105,Pointers in C,Yashvant Kanitkar,340,300

106,Unix SHell Scripting,Tashvant Kanitkar,250,200

107,Python,Guido Van Rossum,360,1000

108,Python Pandas,Guido Van Rossum,410,1230

109,Java,Oracle,324,230

110,Power BI,Microsoft Ltd,510,650

111,Mrutyunjay,Shivaji Sawant,388,600

112,Raja Shivchatrapati,Babasaheb Purandare,455,2000

3. Create table store\_dtl in retail database.

(Which will show store\_id, store\_city, store\_state, contact\_no)

4. Enter hard coded store details as

1, Satara, Maharashtra, 02162-234567

2, Pune, Maharashtra, 020-23456789

3, Aurangabad, Maharashtra, 0240-2345678

5. Create family\_dtl table consisting family details in family database.

(Which will show first\_name, middle\_name, last\_name, age, relation\_with\_yourself)

6. Create table student in school database.(Roll no,first name,last name,DOB)

7. Create table marks in school database.(Roll no,maths marks,English marks,science marks)

8. Create table ‘Wing a’showing details of flat holders in shrileela database.(flat No,owner name,mobile number)(max 10 flats)

9.Create table ‘wing B’showing details of flat holders in shrileela database.(flat number,owner name,mobile number)(max 10 flats)

10. Create table donor in blood\_bank database.(donor no,first name,last name,blood group,contact no,latest date of donation).

ALTER TABLE:

1. Add city column in family details table.

2. Add DOB column after last\_name column.

3. Delete age column.

4. Delete family\_details table.

5. Rename price column from book table as book\_price.

6. Create new column in student table for standard.

7. Create new columns geography\_% of marks in mark table.

8. Create new column in wing A and wing B showing number of family members.

9. Create new column no. of guests in wing A and wing B table.

10. Create new column donor address in donor table.

11. Rename standard column from student table to student\_std

12. Rename % of marks from marks table to percent\_marks.

13. Rename donor address column from donor table to permanent\_address.

14. Drop No. of guests column.

15.Drop permanent\_address column

16. Drop donor table

17. Drop shrileela database.

GENERATION OF REPORT:

1. display emp\_id, first name for employees on the report

2. Generate a report that shows all products all columns

3. Generate a report that shows product\_id, description, category for all products belonging to any two categories

4. Insert 3 products to product table with description field as empty

5. Generate a report that shows product\_id, description for all products.

6. Generate report showing all details of employees table.

7. Generate report showing all details of salaries table.

8. Generate report showing all details of departments table.

9. Generate report showing all details of dept\_emp table

10. Generate report showing all details of titles table.

11. Generate report showing member\_id,first\_name and last\_name from member table.

12. Generate report showing member\_id , reg\_date and reg\_store from member table.

13. Generate report showing product\_id , description,price from product table.

14. Generate report showing tran\_id , qty and amt from tran \_dtl table.

15. Generate report showing tran\_id, member\_id, store\_id and tran\_dt from tran\_hdr table.

ASIC\_AGGREGATION\_FUNCTIONS

1. Generate report showing total number of employees.

2. Generate report showing total number of transactions at store 2.

3. Generate report showing average price for baked goods category.

4. Generate report showing number of employees born in year 1960.

5. Generate report showing min price for beverages category.

6. Generate report showing max price for frozen category.

7. Generate report showing total amount of sales for transaction date 2019-08-12.

AGGREGATION WITH GROUP BY, ORDER BY

1. Generate report showing monthly sale for each month.

2. Generate report showing employees who have worked in more than one department.

3. Generate report showing total number of managers for each department.

4. Generate report showing total salary withdrawn from company for each employee.

5. Generate report showing only those employees from above report who have withdrawn total salary in between 400000 to 500000

and arrange them in descending order by salary withdrawn.

6. Genetaye report showing total number of transactions per member per month.

7. Generate report showing total number of transaction per month per store in ascending order.

8. Generate report showing number of members registered per month per store.

9. Generate report showing category wise number of products having price less than 1.

10. Generate report showing total number of transactions per member per store.

1. Generate report that shows number of transactions per member per store. (Show member name and store city).

2. Generate a report that shows total product sale by month. show product\_description, month and sale

3. Generate a report that shows total sale by member, product and month. (show member name, product description, month, total\_sale on the report.)

4. Generate a report that shows employee number, first name, title, department name.

5. Generate report that shows title wise number of employees in each department. (show department number and department name) \*\*\*\*(MODERATE TO DIFFICULT LEVEL)

1. Generate report showing all details of product having minimum price.

2. Generare report showing details of all products from category having minimum price product.

3. Generate report showing no of transactions per member having more than 5 different products purchased in single transaction.

4. Generate report showing no of transactions by member in store where he/she is registered.

5. Generate report showing number of transactions having total sales more than 50 per member.

1. Generate report showing tran\_id, member name, category, quantity and amount.

tran\_id ==> tran\_hdr(primary key) or tran\_dtl

member\_name ==> member table(member\_id primary key)

category ==> product table(product id primary key)

qty ==> tran\_dtl table

amt ==> tran\_dtl table

STEP-1

SELECT th.tran\_id, m.first\_name

FROM tran\_hdr th

JOIN member m

ON th.member\_id = m.member\_id;

STEP 2

SELECT th.tran\_id, m.first\_name, td.qty AS quantity, td.amt AS amount

FROM tran\_hdr th

JOIN member m

ON th.member\_id = m.member\_id

JOIN tran\_dtl td

ON th.tran\_id = td.tran\_id;

STEP 3

SELECT th.tran\_id, m.first\_name, p.category, td.qty AS quantity, td.amt AS amount

FROM tran\_hdr th

JOIN member m

ON th.member\_id = m.member\_id

JOIN tran\_dtl td

ON th.tran\_id = td.tran\_id

JOIN product p

ON td.product\_id = p.product\_id;

STEP 4

SELECT m.first\_name, p.category, sum(td.qty) AS total\_quantity, sum(td.amt) AS total\_amount

FROM tran\_hdr th

JOIN member m

ON th.member\_id = m.member\_id

JOIN tran\_dtl td

ON th.tran\_id = td.tran\_id

JOIN product p

ON td.product\_id = p.product\_id

GROUP BY m.first\_name, p.category;

1. display emp\_id, first name and years of experience on a report.

2. select member\_id, name and membership\_age (no of days since registration)

3. encoding: select all columns from employee and one additional column that classifies employess in 3 categories based on year of experience. (hire\_date)

4. encoding::select all columns from member table and one additional column that classifies members in 3 categories based on membership\_age

5. Transpose:: Represent three categories generated in assignment 4 along columns instead of rows. Three columns (basic, executive, premium each column will carry value 0 or 1)

6. Generate a report that shows sales and qty sold by store and month.

7. Generate a report that shows sales by category and year where total sale is more than 2000 dollars

8. Generate a report that shows employee name and manage name

9. Generate a report that shows avg price for every category

10. Generate a report that selects all columns from product table with one additional column price\_cat\_avg (this has three values "below\_avg", "avg", "above\_avg" computed using price and avg price for category)

11. Build query for transpose type 2 (convert columns into rows)

12.

ASSIGNMENT 1 - Simple Expressions

1. display emp\_id, first name for employees on the report

2. Generate a report that shows all products all columns

3. Generate a report that shows product\_id, description, category for all products belonging to any two categories

4. Insert 3 products to product table with description field as empty

5. Generate a report that shows product\_id, description for all products. Products where description field is empty should show "STANDARD PRODUCT" as description.

ASSIGNMENT 2. - Simple Expressions

1. Generate a report that shows member\_id, name, membership\_age (no of days since registration)

2. Generate a report that shows tran\_id, product\_id, quantity, sale and price (computed using quantity and sale values)

3. Generate a report that shows emp\_id, emp\_fullname (concatenate first and last name separated by space)

4. Generate a report that shows tran\_id, product\_id, quantity, sale, recency (no of days between current date and tran\_date)

5. Generate a report that shows tran\_id, product\_id, quantity, sale, day, month, quarter, year

ASSIGNMENT 3 - Aggregations

1. Generate a report that shows monthly sale by product

2. Generate a report that shows monthly sale by product, member order results in descending order of the sale value.

3. Generate a report that shows average monthly sale by a member order results by descending order of sale value.

4. Generate a report that shows total monthly sale by each product where total sale value is > 1000

5. Generate a report that shows month and no of products for which total monthly sale for that month is > 1000

ASSIGNMENT 4 - Joins - Aggregations

1. Generate a report that shows total product sale by month. show product\_description, month and sale

2. Generate a report that shows total sale by member, product and month. show member name, product description, month, total\_sale on the report.

3. Add 3 products in product table that don't have key in tran\_detail table. DO query 1 with left outer join insted of inner join.

4. Add one row in tran\_detail table where product id is not in product table. try left, right and full outer join queries.

5. Generate a report that shows emp\_id, emp\_name, manager\_id, manager\_name from employee table. (self join).

ASSIGNMENT 5 - Medium Complexity Transformations

1. Add 3 products to product table with little variation in category description field (some spelling difference).

Write a query to replace this new description to correct one (new column can be added). Save output of this query in new table.

2. Create a table emp\_latest\_dept (defination similar to emp\_dept) and write a query to save employee and his/her latest department in this new table.

3. encoding: select all columns from employee and one additional column that classifies employess in 3 categories based on year of experience. (hire\_date)

4. encoding::select all columns from member table and one additional column that classifies members in 3 categories based on membership\_age (new,loyal, very\_loyal)

5. Create table that stores tran\_id, member\_id, store\_id, total\_qty, total\_sale for the transaction, tran\_cat (this column should have values low, medium, high based on total sale value of the transaction.)

ASSIGNMENT 6: Complex Transformations

1. Generate a report that shows month, product\_description, category\_description, total\_sale\_product, total\_sale\_category, %\_of\_cat

2. Add additional column price\_cat on product table (low, medium, high based on price range). Generate a report that shows member\_id, name, month, price\_cat, total\_quantity, total\_sale by member, month.

3. Generate a report that shows month, product\_description, total\_product\_sale, Avg(sale) for all products in the category to which product belongs

4. Compute indirect contribution of a product.:: Select transactions in which a product is present.. get total sale value for those transactions and

generate a report that shows product\_id, month, total\_sale, total\_sale\_indirect (total\_sale value of transactions in which product is present)

5. Generate report that shows product combinations (size 2) and total qty and total sale value when that combination is present. (\*\*\* very complex)

ASSIGNMENT 7: Transpose

1. Columns in to rows :: Create a table with defination similar to product table. (need not have higher dimensions like category). New table will have product\_id, Description, level (this can be line\_item, category etc)

Write a query to store all hierarchy levels of product in one table (in came column) identified by level column.

2. Write a query to show quarteryly sale by product in 4 columns instead of rows. it should look as product\_id, description, q1\_sale, q2\_sale, q3\_sale,q4\_sale

3. Generate a report similar to previous one but have 12 columns (one for each month).

4. Generate a report that shows product\_id, total\_sale\_low\_value\_basket, total\_sale\_medium\_value\_basket, total\_sale\_high\_value\_basket

5. Generate a report that shows product\_id, new\_member\_sale, loyal\_member\_sale, very\_loyal\_member\_sale

ASSIGNMENT 8: Rank and Bucket

1. Generate a report that shows top 5 products by member ranked by total qty purchased.

2. Generate a report that shows top 10 products by store ranked by total sale value.

3. Generate a report that divides products purchased by members in 10 buckets and shows appropriate bucket no against product. member\_id, product\_id, total\_sale, bucket\_no as member\_bucket

4. Generate a report that divides products in 10 buckets (considering all data) and shows. product\_id, total\_sale, bucket\_no as global\_bucket

5. Generate a report that shows member\_id, product\_id, global\_bucket, member\_bucket.

ASSIGNMENT 9: Running total

1. Generate a report member, month, toal\_sale, total\_sale\_till\_month (running\_total)

2. Add data for one more year (do some small changes in data). Generate a report that shows member, year, month, sale, running total, previous year month sale, previous yr running total.

3. Geneate a similar report for quarterly sale.

4. Generate a report that shows employee, year, month, salary, total\_salary\_till\_date.

5. Generate report 2 at category level and report 4 at department level.

ASSIGNMENT 10: Moving window, lead and lag, nth value

1. Compute inter-purchase interval for every member :: member\_id, tran\_date, previous\_tran\_date, inter\_purchase\_interval (in days) -- compute avg ipi for every member.

2. Generate a report that displays previous 2 month and next 3 month data on columns. member\_id, product\_id, -2\_month\_sale, -1\_month\_sale, month\_sale,next\_month\_sale, +2\_month\_sale, +3\_month\_sale

3. Generate a report that shows member, product, month, sale, last\_3\_month\_sale

4. Generate a report that shows member, product, month, sale, 2nd\_max\_totalsale\_value, 3rd\_min\_totalsale\_value

5. Generate a report 4 for categories.