



Symbiosis Institute of Technology

Data Base Management System Lab Manual

Department of Computer Science & Information Technology

Lab Manual for Academic Session 2018-19

Programme Name: B.Tech TYCS/TYIT

Semester: V

Lab Course Details

Programme Name:- UG (TY B.Tech)

Course Name - Data Base Management Systems Lab

Course Code - 070121506 CS

Semester -V

Credit Points - 02

Practical Hrs/Week - 4

Term work - 50 marks

Assignment List

Assignment No.	Assignment Name
1	Concept of E-R Diagrams and Conversion to Relational Schema
2	Concept of EER Diagrams and Conversion to Relational Schema
3	Introduction to DDL and DML commands and its execution.
4	Implementation of SQL Functions
5	Query formation using nested update, insert delete statements and select with groupby and having clause.
6	Implementation of SQL Joins
7	Implementation of PL/SQL Functions
8	Implementation of PL/SQL Procedures
9	Implementation of PL/SQL Triggers
10	Introduction to Views.

DBMS Lab Assignment - 1

ER to Relational Model Conversion

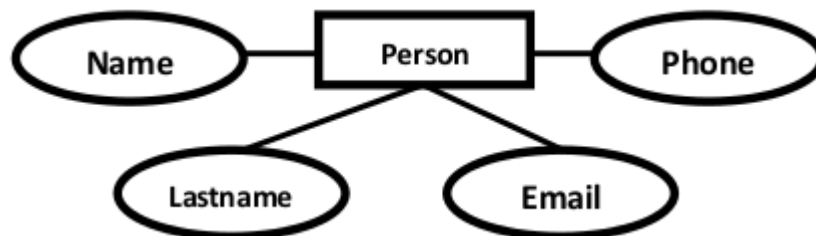
The ER Model is intended as a description of real-world entities. Although it is constructed in such a way as to allow easy translation to the relational schema model, this is not an entirely trivial process. The ER diagram represents the conceptual level of database design meanwhile the relational schema is the logical level for the database design. We will be following the simple rules:

1. Entities and Simple Attributes:

An entity type within ER diagram is turned into a table. You may preferably keep the same name for the entity or give it a sensible name but avoid DBMS reserved words as well as avoid the use of special characters. Each attribute turns into a column (attribute) in the table. The key attribute of the entity is the primary key of the table which is usually underlined. It can be composite if required but can never be null.

It is highly recommended that every table should start with its primary key attribute conventionally named as TableNameID.

Taking the following simple ER diagram:



The initial relational schema is expressed in the following format writing the table names with the attributes list inside a parentheses as shown below for

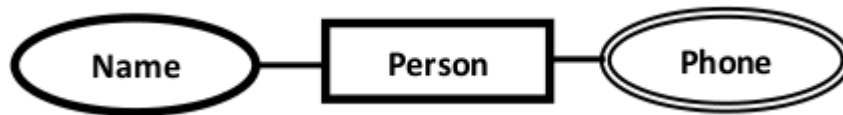
Persons(personid , name, lastname, email)

Persons and Phones are Tables. name, lastname, are Table Columns (Attributes).

personid is the primary key for the table : Person

2. Multi-Valued Attributes

A multi-valued attribute is usually represented with a double-line oval.



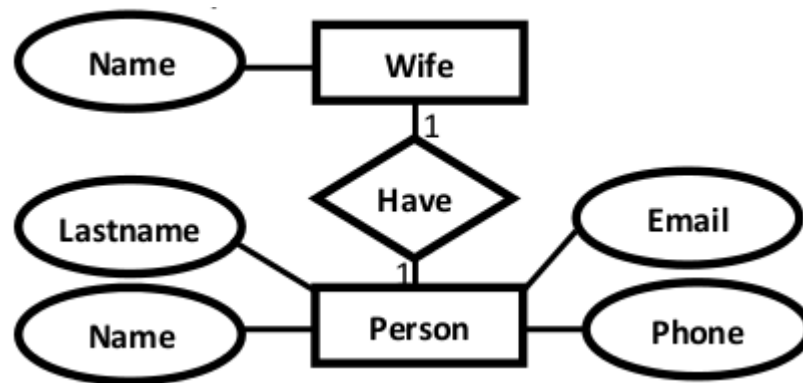
If you have a multi-valued attribute, take the attribute and turn it into a new entity or table of its own. Then make a 1:N relationship between the new entity and the existing one. In simple words. 1. Create a table for the attribute. 2. Add the primary (id) column of the parent entity as a foreign key within the new table as shown below:

Persons(personid , name, lastname, email)

Phones (phoneid , personid, phone)

personid within the table Phones is a foreign key referring to the personid of Persons

3. 1:1 Relationships



To keep it simple and even for better performances at data retrieval, I would personally recommend using attributes to represent such relationship. For instance, let us consider the case where the Person has or optionally has one wife. You can place the primary key of the wife within the table of the Persons which we call in this case Foreign key as shown below.

Persons(personid , name, lastname, email , wifeid)

Wife (wifeid , name)

Or vice versa to put the **personid** as a foreign key within the Wife table as shown below:

Persons(personid , name, lastname, email)

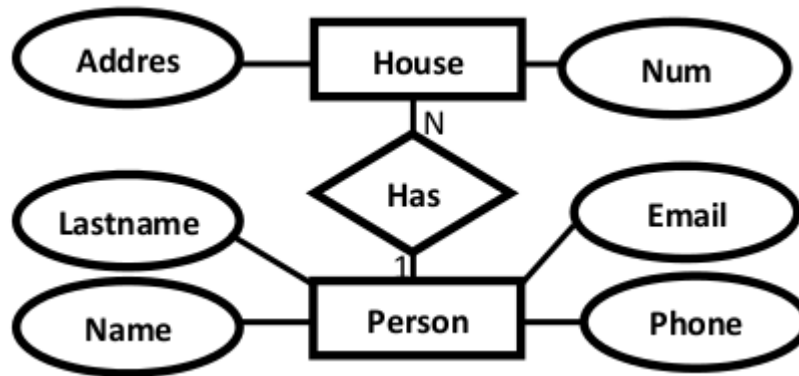
Wife (wifeid , name , personid)

For cases when the Person is not married i.e. has no wifeID, the attribute can set to NULL

4. 1:N Relationships

This is the tricky part ! For simplicity, use attributes in the same way as 1:1 relationship but we have only one choice as opposed to two choices. For instance, the Person can have a House

from zero to many , but a House can have only one Person. To represent such relationship the personid as the Parent node must be placed within the Child table as a foreign key but not the other way around as shown next:

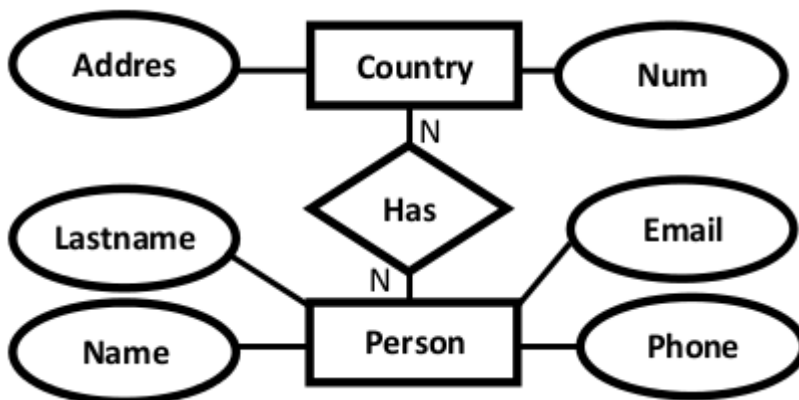


It should convert to :

Persons(personid , name, lastname, email)
House (houseid , num , address, *personid*)

5. N:N Relationships

We normally use tables to express such type of relationship. This is the same for N – ary relationship of ER diagrams. For instance, The Person can live or work in many countries. Also, a country can have many people. To express this relationship within a relational schema we use a separate table as shown below:



It should convert into :

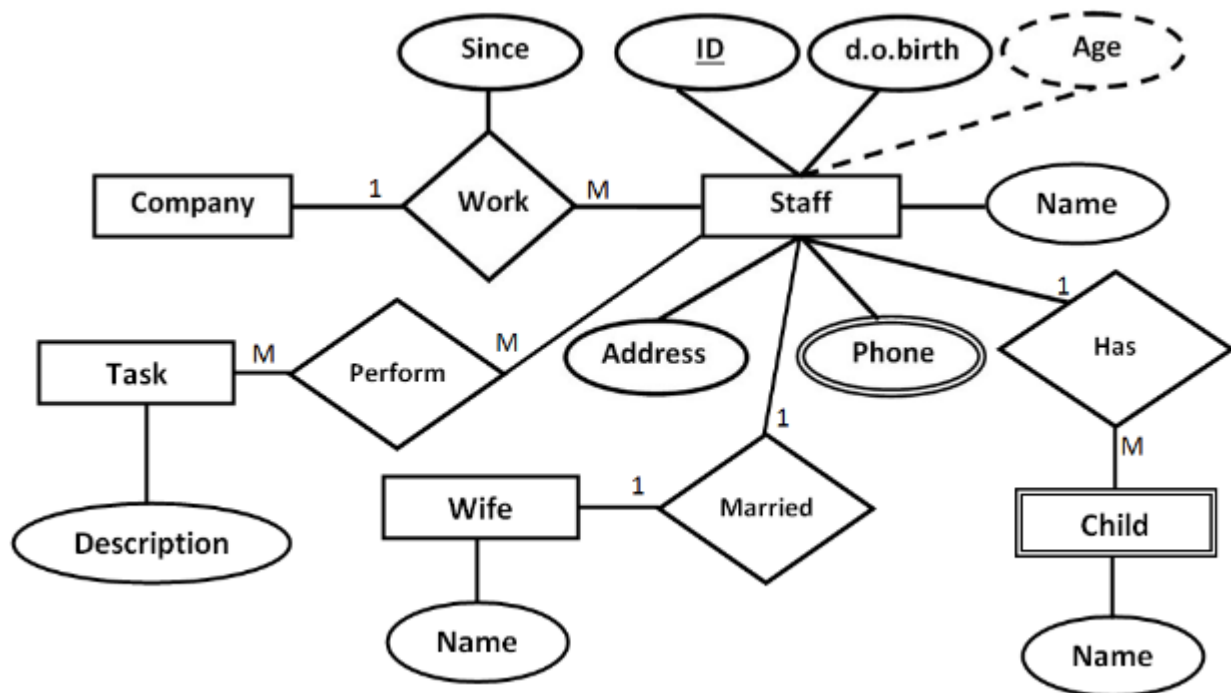
Persons(personid , name, lastname, email)
Countries (countryid , name, code)
HasRelat (hasrelatid , personid , countryid)

Relationship with attributes:

It is recommended to use table to represent them to keep the design tidy and clean **regardless of the cardinality** of the relationship.

Practice Example

Convert the following ER diagram into database relational schema:



The relational schema for the ER Diagram is given below as:

Company(CompanyID , name , address)
Staff(StaffID , dob , address , WifeID)
Child(ChildID , name , StaffID)
Wife (WifeID , name)
Phone(PhoneID , phoneNumber , StaffID)
Task (TaskID , description)
Work(WorkID , CompanyID , StaffID , since)
Perform(PerformID , StaffID , TaskID)

Assignment 1 Lab Questions

Theory Questions

1. What are ER Diagrams? What is their significance in DBMS?
2. Explain the conversion process from ER to Relational.

Implementation Question:- Draw an ER diagram for the following problem statements and then convert them into database relational schema

1) CAR Rental Management

- One customer can rent one car.
- Each car has one manufacturer.
- Each manufacturer can manufacture many types of cars.
- Each car has its own maintenance schedule.
- The revenue is generated based on number of days the car was rented.
- The company has branches in 5 major cities.
- The companies also have 5 employees per branch.
- Initial rental request are handed by employees.

2) SIU Library Management System.

- SIU library there can be individual institute libraries.
- Library has members.
- Library buys books of many publishers.
- Sellers sell books of many publishers.
- Library contains books
- Books are written by one/many authors.
- Authors' books are published by publishers.

3) Online Grocery Ordering System

- Customer must orders one or more items. Each customer is identified by his/her name. The postal address and email address of the customer are also required for the correct placement of order.
- Items have their name and price to be shown to the customers.
- Shopping carts are created by the orders of the customers.
- Shopping carts contain at least one item.
- An order must have at least one requested item.
- Not all items have customers.
- Distinct companies produce distinct items which are differentiated by the company's name.
- Payment must be made via credit cards. The payment process has to be done after the credit card's verification.

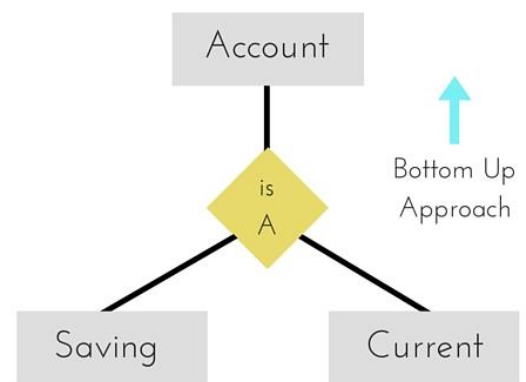
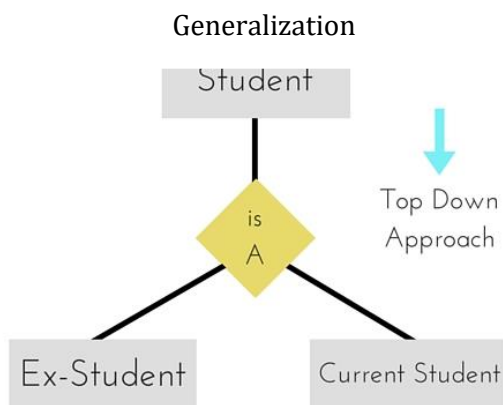
DBMS Lab Assignment -2

Conversion of EER Model into Relational Model

The Extended Entity-Relationship Model is a more complex and high-level model that extends an E-R diagram to include more types of abstraction, and to more clearly express constraints. All of the concepts contained within an E-R diagram are included in the EE-R model, along with additional concepts that cover more semantic information. These additional concepts include generalization, specialization, and aggregation.

Generalization is a bottom-up approach in which two lower level entities combine to form a higher level entity. In generalization, the higher level entity can also combine with other lower

level entity to make further higher level entity.



Specialization

Specialization is opposite to Generalization. It is a top-down approach in which one higher level entity can be broken down into two lower level entities. In specialization, some higher level entities may not have lower-level entity sets at all.

Conversion process for Generalization/ Specialization into Relational Schema:

- Create tables for all higher-level entities.

- Create tables for lower-level entities.
- Add primary keys of higher-level entities in the table of lower-level entities.
- In lower-level tables, add all other attributes of lower-level entities.
- Declare primary key of higher-level table and the primary key for lower-level table.
- Declare foreign key constraints.

Representing Generalization (Approach #1)

Create a table for each lower-level entity set only

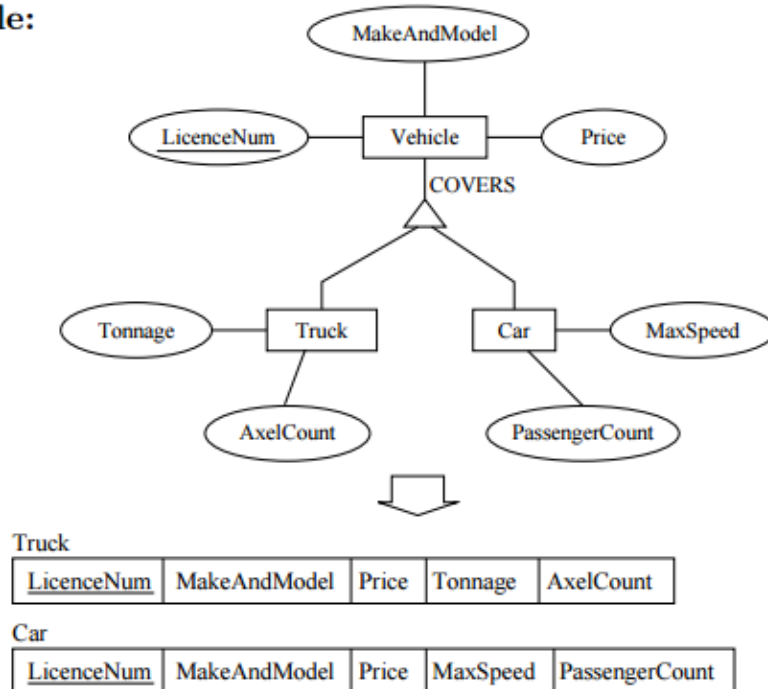
Columns of new tables should include

- Attributes of lower level entity set
- Attributes of the superset

The higher-level entity set can be defined as a view on the tables for the lower-level entity sets

Representing Generalization (Approach #1)

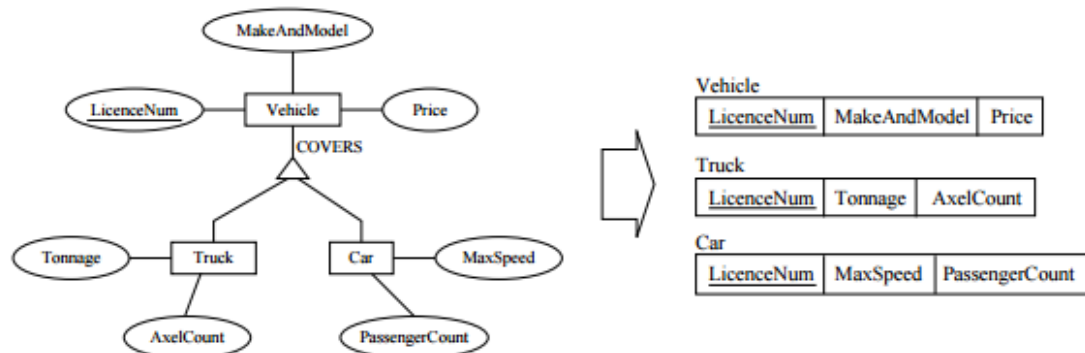
Example:



Representing Generalization (Approach #2)

Treat generalization the same as specialization.

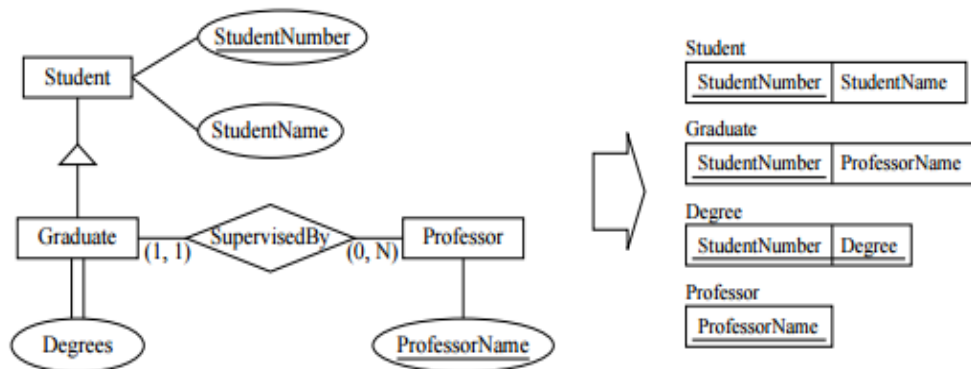
Example:



Representing Specialization

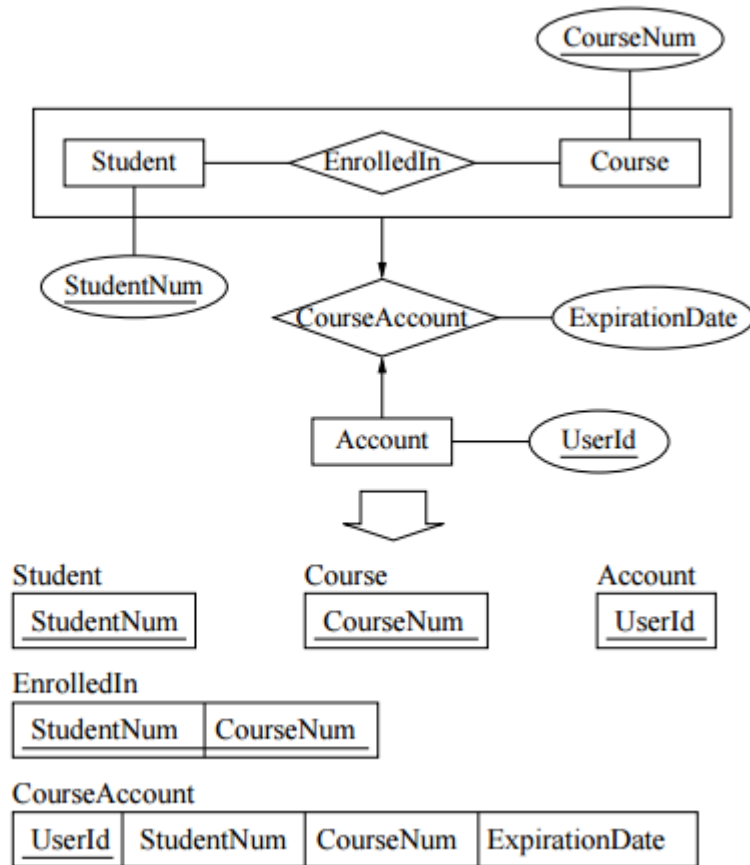
Create table for higher-level entity set, and treat specialized entity subsets like weak entity sets (without discriminators)

Example:



One limitation on ER model is that it cannot express relationships among relationships. **Aggregation** provides best solution for this purpose. Aggregation is a process when relation between two entities is treated as a single entity. To represent relationship set involving aggregation of R, treat the aggregation like an entity set whose primary key is the primary key of the table for R.

Example:



Assignment 2 Lab Questions

Theory Questions:-

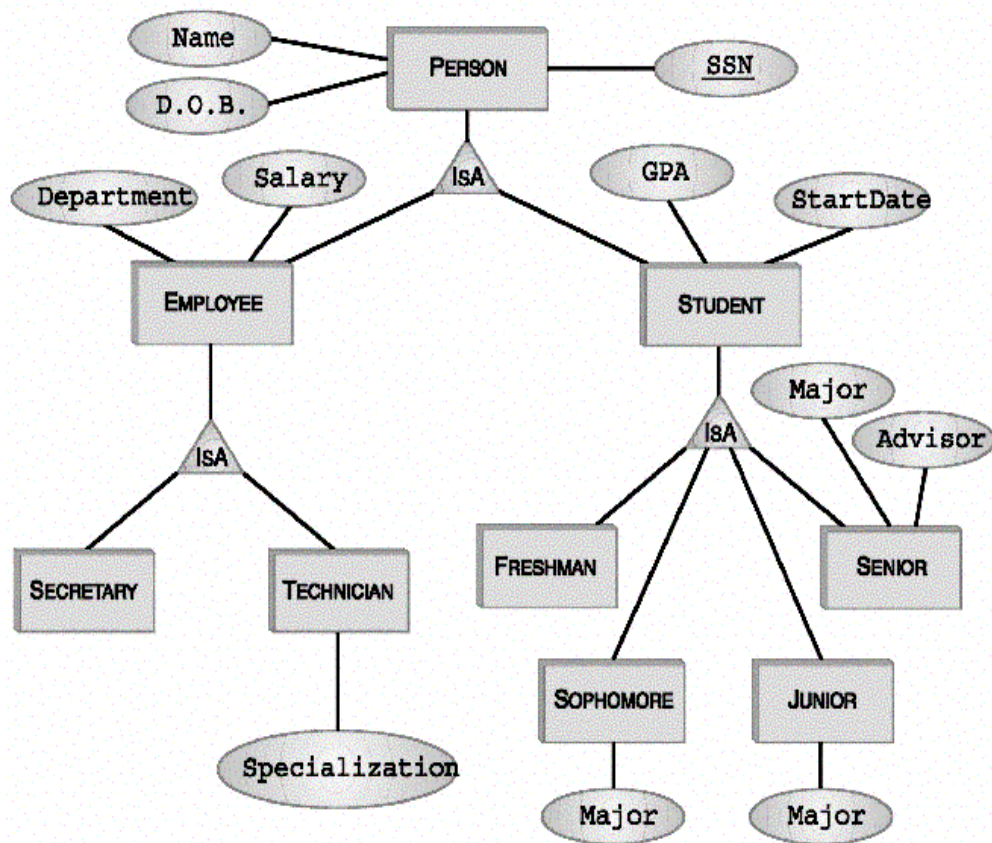
Q.1 Explain the need of EER diagrams

Q.2 Explain in detail Generalization, specialization and Aggregation and how do we convert them into relational model

Implementation Question:-

Q.1 Draw the EER for SIU library management and convert into relational model.

Q.2 Convert following EER to relational model



DBMS Lab Assignment - 3**Introduction to DDL and DML commands and its execution.**

Data definition language defines the schema for the database by specifying entities and the relationship among them. In addition to this, DDL even defines certain security constraints. The execution of DDL statements results in new tables which are stored in "system catalog" also called data dictionary or data directory.

Data Manipulation Language is a language that provides a set of operations to support the basic data manipulation operations on the data held in the databases. It allows users to insert, update, delete and retrieve data from the database. Data manipulations are applied at internal, conceptual and external levels of schemas. However, the level of complexity at each schema level varies from one another.

Data Control Language statements control access to data and the database using statements such as GRANT and REVOKE. A privilege can either be granted to a User with the help of GRANT statement. The privileges assigned can be SELECT, ALTER, DELETE, EXECUTE, INSERT, INDEX etc. In addition to granting of privileges, you can also revoke (taken back) it by using REVOKE command.

DDL : Data Definition Language

All DDL commands are auto-committed. That means it saves all the changes permanently in the database.

Command	Description
create	to create new table or database
alter	for alteration
truncate	delete data from table
drop	to drop a table
rename	to rename a table

DML : Data Manipulation Language

DML commands are not auto-committed. It means changes are not permanent to database, they can be rolled back.

Command	Description
insert	to insert a new row
update	to update existing row
delete	to delete a row
merge	merging two rows or two tables

DCL : Data Control Language

Data control language provides command to grant and take back authority.

Command	Description
grant	grant permission of right
revoke	take back permission.

Assignment 3 Lab Questions**Section- 1 DDL Commands**

Q-1 Explain DDL commands and their syntax

Q-2 Create tables for the following relational model of library management. Apply constraints on the columns and also alter the structure according to your requirements.

1. **SIULIBRARY** (Slid, lname, location, noofbranches)
2. **llibrary** (Lid, lname, city, area, slid)
3. **BOOKS** (Bid, Bname, Price, Lid)
4. **Noofcopies** (bnid, bid, blid)
5. **AUTHOR** (Aid, Aname, email, phoneno)
6. **Writes** (Bid, Aid, pid)
7. **PUBLISHER** (Pid, Pname)
8. **SELLER** (Sid, sname, city)
9. **DEPARTMENT** (Deptid, deptname, lname, lid)
10. **STUDENT** (Stuid, Sname, email, memid, deptid)
11. **STAFF** (Stid, stname, email, deptid, memid)
12. **PURCHASE** (prid, lid, sid, pid, bid, quantity, date, totalcost)
13. **ISSUE** (Issueid, memid, bid, lid, issuedate, returndate)
14. **SELLS** (sid, bid, pid)
15. **Employee** (eid, empname, email, salary, lid)

16.A_specialization(spec_id,spec_name,Aid)

17.Member(memid,lid)

Section- 2 DML Command execution

Q-1. Explain DML commands and their syntax.

Q-2. Insert 5 tuples in each of the created tables.

Q-3 Execute following queries on the library database

1. Which institute libraries are located in pune city?
2. To which institute CS department belongs to?
3. Find all the books whose price is between 800 to 12000?
4. Find out such employees who's salaries are not greater than 50,000/-
5. Find out such sellers who's name end with "ta"
6. Find out such institute libraries where their area information is missing.
7. Find out such staff members who's name doesn't starts with "A"
8. Find out such SIU libraries which have institute libraries located in Bangalore.
9. Which students belong to civil department?
10. Find out books which are written by "shruti" and published by McGraw hill.
11. Which members have issued books that belongs to E&TC department of SITM.
12. List the books of SIT library
13. Which books are written by " Ken Coel"
14. Find the staff members who's salaries are less than atleast one employee
15. Find the books who's price is greater than atleast one book in the library.
16. Find the employees who's salary is greater than all the employees of CS department.
17. Find such purchases who's total cost is greater than all the purchases which are done by "SIT library"

- 18.** Find name of such SIU library which is located in Nashik and has library named as "SIOM library"(solve using exists)
- 19.** Find such staff members who's name starts with "S" and do not work for "CS" department. (solve using not exists)
- 20.** Which books are issued by student Shivani?
- 21.** From which seller did SIT library purchases books of "Technical publications"?
- 22.** Which books are issued by staff of SSLA law department?
- 23.** Which sellers reside in the same city in which SCHC library resides?
- 24.** Find out the publishers whose books are provided by the seller "pragati books store" to SSBS library.
- 25.** Which books are published by "Shivam Kapoor" with the publishing house Wiley publications?
- 26.** What are the purchase details of the SIBM library for the month of December?
- 27.** Which institute does the member belong who have issued book number 453?
- 28.** Increase the price of all the books by 10%.
- 29.** Change the institute name to SSBS of "Biology department"
- 30.** Provide 25% Diwali bonus to SIT staff.
- 31.** Delete the purchase details which are happened in 2016.
- 32.** Delete such issues where books are already returned in 2017.

DBMS Lab Assignment - 4

SQL Functions and its types

SQL provides many built-in functions to perform operations on data. These functions are useful while performing mathematical calculations, string concatenations, sub-strings etc.

Four types of SQL functions:

Aggregate Functions

MySQL supports the following aggregate functions:

Function	Description
AVG()	Returns the average of the values in the selected column
COUNT()	Returns the number of rows returned for a selection
MAX()	Returns the maximum value for a column
MIN()	Returns the minimum value of a column
SUM()	Returns the sum of the values in a specified column

Date Functions

Function	Description
<u>NOW()</u>	Returns the current date and time
<u>CURDATE()</u>	Returns the current date
<u>CURTIME()</u>	Returns the current time
<u>DATE()</u>	Extracts the date part of a date or date/time expression
<u>EXTRACT()</u>	Returns a single part of a date/time
<u>DATE_ADD()</u>	Adds a specified time interval to a date
<u>DATE_SUB()</u>	Subtracts a specified time interval from a date
<u>DATEDIFF()</u>	Returns the number of days between two dates
<u>DATE_FORMAT()</u>	Displays date/time data in different formats

Numeric Functions

Command	Query	Output
Abs(n)	Select abs(-15) from dual;	15
Ceil(n)	Select ceil(55.67) from dual;	56
Exp(n)	Select exp(4) from dual;	54.59
Floor(n)	Select floor(100.2) from dual;	100
Power(m,n)	Select power(4,2) from dual;	16
Mod(m,n)	Select mod(10,3) from dual;	1
Round(m,n)	Select round(100.256,2) from dual;	100.26
Trunc(m,n)	Select trunc(100.256,2) from dual;	100.23
Sqrt(m,n)	Select sqrt(16) from dual;	4

Char Functions

lower (char); upper (char);	<i>select lower ('HELLO') from dual;</i> <i>select upper ('hello') from dual;</i>	hello HELLO
ltrim (char,[set]);	<i>select ltrim ('cseit', 'cse') from dual;</i>	it
rtrim (char,[set]);	<i>select rtrim ('cseit', 'it') from dual;</i>	cse
replace (char,search string, replace string);	<i>select replace ('jack and jue', 'j', 'bl') from dual;</i>	black and blue
substr (char,m,n);	<i>select substr ('information', 3, 4) from dual;</i>	form

Assignment 4 Lab Questions

Perform following queries using SQL functions:

1. Find the cheapest book of SIBM library.
2. Which library has the costliest book?
3. How many students from SIT issued the book?
4. What is the average cost of books in SITM library?
5. What is the total cost of purchase made by SSBS in the month of January?
6. How many books are written by “Shruti”
7. What is the costliest book published by “Pragati Book Store”
8. How many total copies of books do SIT has?
9. What is the average cost of books written by “Mr. Kumar”?
10. How many books are sold by seller living in Nagpur?
11. Print the student name in capital who belongs to SSBS
12. Add two months to the issue date of book written by “Ameya”
13. What was the last day of the month when Ayush issued the book?
14. How many books are issued from January to march 2016?
15. How many books have copies less than 5 available in the SIBM library?

DBMS Lab Assignment - 5

Group by, order by and having clause implementation for nested queries

Aggregate functions are performed on the columns values. But if we want to divide values stored in a single column in groups and then perform certain actions, then we have to use Group By clause.

GROUP BY is use to **group** values from a column, and, if you wish, performs calculations on that column. You can use COUNT, SUM, AVG, etc., functions on the grouped column.

Syntax

The syntax for the GROUP BY clause in MySQL is:

```
SELECT expression1, expression2, ... expression_n,  
       aggregate_function (expression)  
FROM tables  
[WHERE conditions]  
GROUP BY expression1, expression2, ... expression_n;
```

Parameters or Arguments

expression1, expression2, ... expression_n

The expressions that are not encapsulated within an aggregate function and must be included in the GROUP BY clause.

aggregate_function

A function such as SUM, COUNT, MIN, MAX, or AVG functions.

tables

The tables that you wish to retrieve records from. There must be at least one table listed in the FROM clause.

WHERE conditions

Optional. The conditions that must be met for the records to be selected.

This GROUP BY example uses the MAX function to return the name of each department and the maximum salary in the department.

```
SELECT department, MAX(salary) AS "Highest salary"  
FROM employees  
GROUP BY department;
```

Having Clause: - If we want to apply certain condition on which values should get included in the group, we have to use the "Having Clause".

Syntax

The syntax for the HAVING Clause in MySQL is:

```
SELECT expression1, expression2, ... expression_n,  
       aggregate_function (expression)  
FROM tables  
[WHERE conditions]  
GROUP BY expression1, expression2, ... expression_n  
HAVING condition;
```

Parameters or Arguments

aggregate_function

A function such as **SUM**, **COUNT**, **MIN**, **MAX**, or **AVG** functions.

expression1, expression2, ... expression_n

The expressions that are not encapsulated within an aggregate function and must be included in the GROUP BY clause.

WHERE conditions

Optional. These are the conditions for the records to be selected.

HAVING condition

This is a further condition applied only to the aggregated results to restrict the groups of returned rows. Only those groups whose condition evaluates to TRUE will be included in the result set.

Example - Using SUM function

Let's look at a MySQL HAVING clause example that uses the **SUM** function.

You could also use the [SUM function](#) to return the name of the product and the total quantity (for that product). The MySQL HAVING clause will filter the results so that only products with a total quantity greater than 10 will be returned.

```
SELECT product, SUM(quantity) AS "Total quantity"  
FROM order_details  
GROUP BY product  
HAVING SUM(quantity) > 10;
```

Order By:-

The MySQL ORDER BY clause is used to sort the records in your result set.

Syntax

The syntax for the ORDER BY clause in MySQL is:

```
SELECT expressions
FROM tables
[WHERE conditions]
ORDER BY expression [ ASC | DESC ];
```

Example:

```
SELECT last_name, first_name, city
FROM contacts
WHERE last_name = 'Johnson'
ORDER BY city DESC;
```

This MySQL ORDER BY example would return all records sorted by the *city* field in descending order.

We can also set a limit on the number of values which we want to be returned by the execution of order by query.

For example:

```
SELECT contact_id, last_name, first_name
FROM contacts
WHERE website = 'TechOnTheNet.com'
ORDER BY contact_id DESC
LIMIT 5;
```

This MySQL SELECT LIMIT example would select the first 5 records from the *contacts* table where the *website* is 'TechOnTheNet.com'. Note that the results are sorted by *contact_id* in descending order so this means that the 5 largest *contact_id* values will be returned by the SELECT LIMIT statement.

Assignment 5 Lab Questions

Perform following queries using Group by, having and Order By

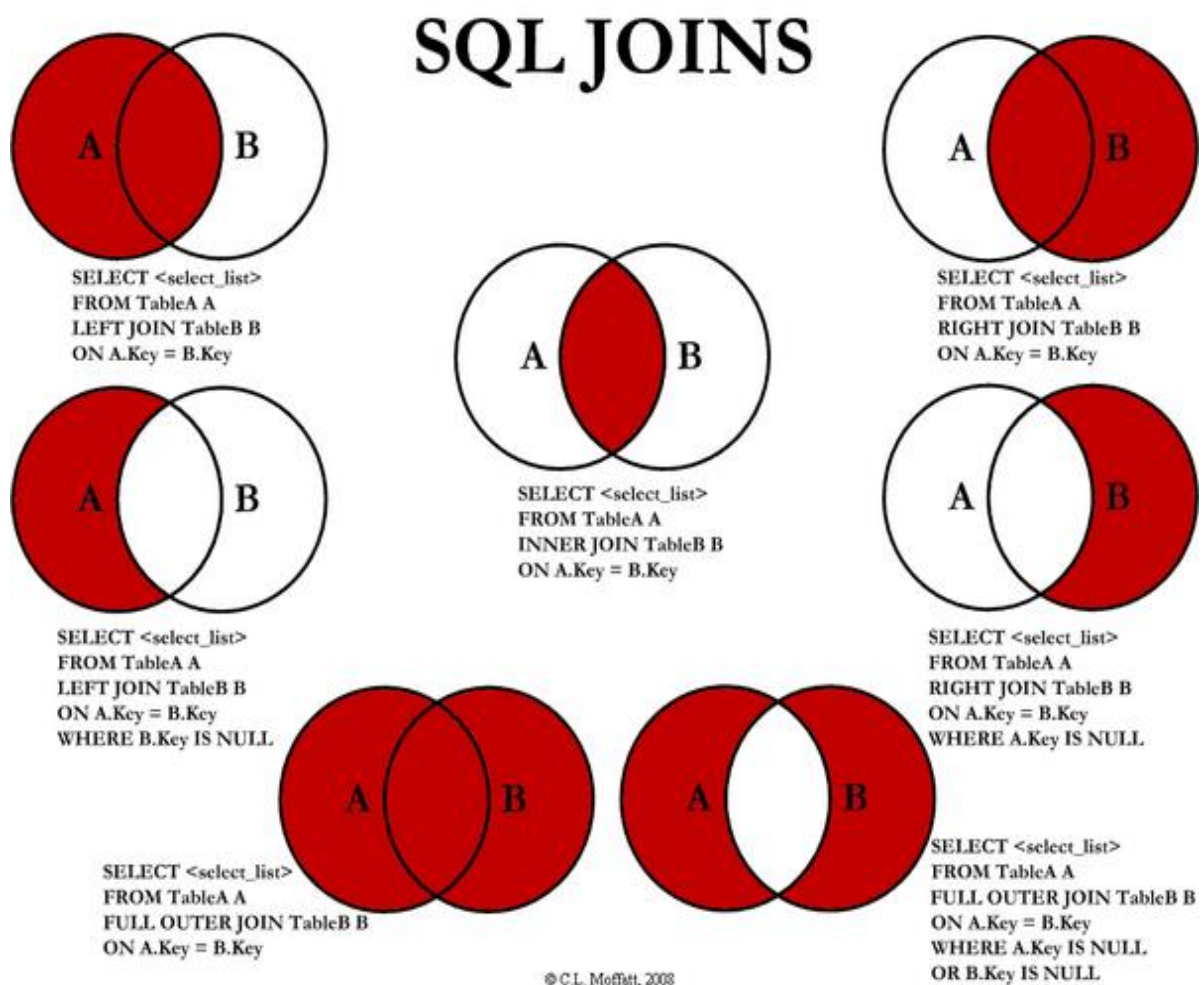
1. Give library wise book details.
2. Give bookwise total copies which are available.
3. Which library has total copies more than 100?
4. Give institute wise department details.
5. Give citywise seller details.
6. Give author wise book details that have authored more than 2 books.
7. Give book details library wise whose price is less than 1000
8. Give department wise staff details.
9. How many books are issued library wise
10. Give purchase details publisher wise.
11. Display books in a descending order of their cost.

SQL Joins

SQL JOINS are used to retrieve data from multiple tables. A SQL JOIN is performed whenever two or more tables are listed in a SQL statement.

There are 4 different types of SQL joins:

- SQL INNER JOIN (sometimes called simple join)
- SQL LEFT OUTER JOIN (sometimes called LEFT JOIN)
- SQL RIGHT OUTER JOIN (sometimes called RIGHT JOIN)
- SQL FULL OUTER JOIN (sometimes called FULL JOIN)



Assignment 6 Lab Questions

Using SQL joins, obtain the following information from SIU Library Database

1. Students in the IT department.
2. Books in SIT library.
3. Books with less than 3 copies in SIT Library.
4. Sellers in the same city as SIT.
5. Sellers selling books to SIT.
6. Books authored by Brian Kernighan, and published by Tata McGraw Hill.
7. Books authored by Ken Thompson.
8. Books issued by Mayank.
9. Books issued by SLS staff.
10. Publisher that provides books to SSBS through College Book Store.
11. Institutes whose staff and students have issued book with bid 4444.
12. Sellers selling to libraries in the same city.
13. Authors with books costing more than rs 500.

PL/SQL Functions

What is a Function in PL/SQL?

A function is a named PL/SQL Block which is similar to a procedure. The major difference between a procedure and a function is, a function must always return a value, but a procedure may or may not return a value.

Syntax:-

```
CREATE [OR REPLACE] FUNCTION function_name [(  
    parameter_1 [IN] [OUT] data_type,  
    parameter_2 [IN] [OUT] data_type,  
    parameter_N [IN] [OUT] data_type)  
    RETURN return_data_type IS  
    --the declaration statements  
BEGIN  
    -- the executable statements  
    return return_data_type;  
    EXCEPTION  
    -- the exception-handling statements  
END;  
/
```

- *Function-name* specifies the name of the function.
- [or replace] option allows the modification of an existing function.
- The optional parameter list contains name, mode and types of the parameters. In represents the value that will be passed from outside and out represents the parameter that will be used to return a value outside of the procedure.
- The function must contain a **return** statement.
- The *return* clause specifies the data type you are going to return from the function.
- *Function-body* contains the executable part.
- The as keyword is used instead of the is keyword for creating a standalone function.

A **subprogram** is a program unit/module that performs a particular task. These subprograms are combined to form larger programs. This is basically called the 'Modular design'. A subprogram can be invoked by another subprogram or program which is called the calling program.

A subprogram can be created:

- At schema level
- Inside a package

- Inside a PL/SQL block

A schema level subprogram is a **standalone subprogram**. It is created with the CREATE PROCEDURE or CREATE FUNCTION statement. It is stored in the database and can be deleted with the DROP PROCEDURE or DROP FUNCTION statement.

A subprogram created inside a package is a **packaged subprogram**. It is stored in the database and can be deleted only when the package is deleted with the DROP PACKAGE statement.

Assignment 7 Lab Questions

Implementation Questions:

1. Write a PL/SQL function to multiply 2 numbers
2. Write a PL/SQL function to find maximum of 2 numbers.
3. Write a PL/SQL function to define total number of books in SIU Library
4. Write a PL/SQL function to find average of 4 numbers.
5. Write a PL/SQL function to find factorial.
6. Write a PL/SQL function to find the name of the library which has the cheapest books.
7. Write a PL/SQL function to find total no. of books issued by SIT Staff.

DBMS Lab Assignment - 8

SQL Procedures

PL/SQL subprograms are named PL/SQL blocks that can be invoked with a set of parameters. PL/SQL provides two kinds of subprograms:

- **Functions:** these subprograms return a single value, mainly used to compute and return a value.
- **Procedures:** these subprograms do not return a value directly, mainly used to perform an action.

This chapter is going to cover important aspects of a **PL/SQL procedure** and we will cover **PL/SQL function** in next chapter.

Parts of a PL/SQL Subprogram

Each PL/SQL subprogram has a name, and may have a parameter list. Like anonymous PL/SQL blocks and, the named blocks a subprograms will also have following three parts:

Sr.No.	Parts & Description
1	Declarative Part:- It is an optional part. However, the declarative part for a subprogram does not start with the DECLARE keyword. It contains declarations of types, cursors, constants, variables, exceptions, and nested subprograms. These items are local to the subprogram and cease to exist when the subprogram completes execution.
2	Executable Part: - This is a mandatory part and contains statements that perform the designated action.
3	Exception-handling :- This is again an optional part. It contains the code that handles run-time errors.

Creating a Procedure

A procedure is created with the CREATE OR REPLACE PROCEDURE statement. The simplified syntax for the CREATE OR REPLACE PROCEDURE statement is as follows:

```
CREATE [OR REPLACE] PROCEDURE procedure_name
[(parameter_name [IN | OUT | IN OUT] type [, ...])]
{IS | AS}
BEGIN
< procedure_body >
```

```
END procedure_name;
```

Where,

- *procedure-name* specifies the name of the procedure.
- [OR REPLACE] option allows modifying an existing procedure.
- The optional parameter list contains name, mode and types of the parameters. IN represents that value will be passed from outside and OUT represents that this parameter will be used to return a value outside of the procedure.
- *procedure-body* contains the executable part.
- The AS keyword is used instead of the IS keyword for creating a standalone procedure.

S.N.	Parameter Mode & Description
1	IN :- An IN parameter lets you pass a value to the subprogram. It is a read-only parameter. Inside the subprogram, an IN parameter acts like a constant. It cannot be assigned a value. You can pass a constant, literal, initialized variable, or expression as an IN parameter. You can also initialize it to a default value; however, in that case, it is omitted from the subprogram call. It is the default mode of parameter passing. Parameters are passed by reference.
2	OUT:- An OUT parameter returns a value to the calling program. Inside the subprogram, an OUT parameter acts like a variable. You can change its value and reference the value after assigning it. The actual parameter must be variable and it is passed by value.
3	IN OUT:- An IN OUT parameter passes an initial value to a subprogram and returns an updated value to the caller. It can be assigned a value and its value can be read. The actual parameter corresponding to an IN OUT formal parameter must be a variable, not a constant or an expression. Formal parameter must be assigned a value. Actual parameter is passed by value.

Assignment 8 Lab Questions

Implementation Questions:

1. Write a procedure to find details of books of a given library;
2. Write a procedure to find out total no. of books written by a particular author.
3. Write a procedure to increase price of books by 10% which are sold by a particular seller to a particular library.
4. Write a procedure to count total number of publishers who are providing books to a particular library.
5. Write a procedure to find the costliest and the cheapest books that a particular library has.
6. Write a procedure to update the price of books by 25%, which are written by a particular author.
7. Write a procedure to provide total money spent by a particular library in purchasing books in a particular month.

DBMS Lab Assignment - 9
SQL Triggers

A SQL trigger is a set of SQL statements stored in the database catalog. A SQL trigger is executed or fired whenever an event associated with a table occurs e.g., insert, update or delete. A SQL trigger is a special type of [stored procedure](#). It is special because it is not called directly like a stored procedure. The main difference between a trigger and a stored procedure is that a trigger is called automatically when a data modification event is made against a table whereas a stored procedure must be called explicitly.

```
CREATE [OR REPLACE] TRIGGER trigger_name
  BEFORE | AFTER
  [INSERT, UPDATE, DELETE [COLUMN NAME..]]
  ON table_name

  Referencing [ OLD AS OLD | NEW AS NEW ]
  FOR EACH ROW | FOR EACH STATEMENT [ WHEN Condition ]

DECLARE
  [declaration_section
    variable declarations;
    constant declarations;
  ]

BEGIN
  [executable_section
    PL/SQL execute/subprogram body
  ]

EXCEPTION
  [exception_section
    PL/SQL Exception block
  ]

END;
```

CREATE [OR REPLACE] TRIGGER trigger_name : Create a trigger with the given name. If already have overwrite the existing trigger with defined same name.

BEFORE | AFTER : Indicates when the trigger get fire. BEFORE trigger execute before when statement execute before. AFTER trigger execute after the statement execute.

INSERT, UPDATE, DELETE [COLUMN NAME..] : Determines the performing trigger event. You can define more then one triggering event separated by OR keyword.

ON table_name : Define the table name to performing trigger event.

Referencing [OLD AS OLD | NEW AS NEW] : This allows you to refer new and old values for various DML statements, such as INSERT, UPDATE, and DELETE.

For each row | for each statement: Trigger must fire when each row gets Affected (ROW Trigger). and fire only once when the entire sql statement is execute (STATEMENT Trigger).

WHEN Condition : Optional. This provides a condition for rows for which the trigger would fire. This clause is valid only for row-level triggers. Trigger fire when specified condition is satisfied.

Assignment 9 Lab Questions

Implementation Questions:

- 1) Implement After update trigger on the Books Table.
- 2) Implement before insert trigger on the Booksby Table.
- 3) Implement After update trigger on the Purchase Table.

SQL Views

In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from one single table.

CREATE VIEW Syntax

```
CREATE VIEW view_name AS  
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

Note: A view always shows up-to-date data! The database engine recreates the data, using the view's SQL statement, every time a user queries a view.

Assignment 10 Lab Questions

Perform a study assignment on SQL Views which should include following details:

- 1) Need of SQL Views
- 2) How to implement and use views in MYSQL.
- 3) Write a view to select all books of SIT library whose cost is less than 1000/-