

I N D E X

Sr. No.	Topic	Page No.	Date of Experiment	Date of Submission	Signature
1	Installing oracle/ MYSQL.		08/04/24	15/04/24	
2	Creating Entity-Relationship Diagram using case tools.		15/04/24	22/04/24	
3	Writing SQL statements Using ORACLE /MYSQL:		22/04/24	29/04/24	
4	Normalization.		29/04/24	06/05/24	
5	Creating cursor.		06/05/24	27/05/24	
6	Creating procedure and functions.		27/05/24	03/06/24	
7	Design and implementation of payroll processing system.		03/06/24	10/06/24	
8	Design and implementation of Library Information System		10/06/24	24/06/24	
9	Design and implementation of Student Information System.		24/06/24	24/06/24	
10	Creating packages and triggers.		24/06/24	24/06/24	
11	Automatic backup of files and recovery of files		24/06/24	24/06/24	

Experiment-2

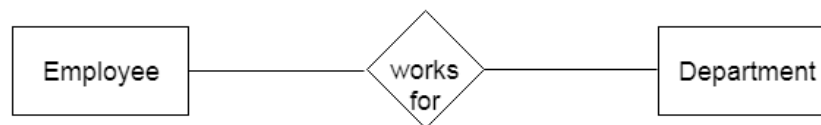
Objective: Creating Entity-Relationship diagram using case tools.

Tools Used:

1. Pen & Paper
2. Database
3. Computer
4. Designing Software

Theory:

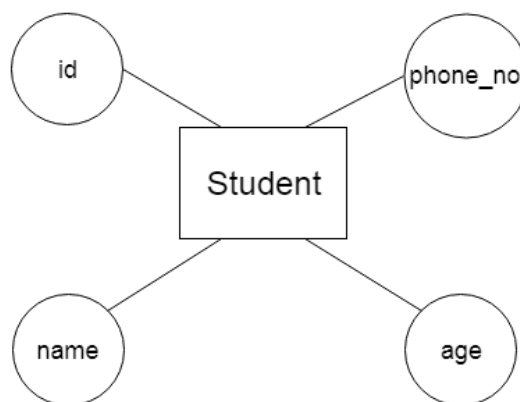
1. Entity: An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles. Consider an organization as an example- manager, product, employee, department etc. can be taken as an entity.



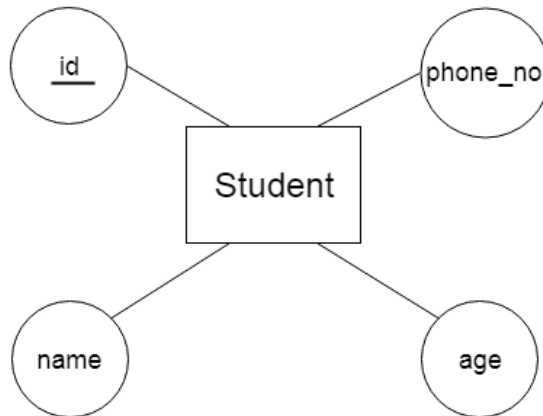
a. Weak Entity: An entity that depends on another entity called a weak entity. The weak entity doesn't contain any key attribute of its own. The weak entity is represented by a double rectangle



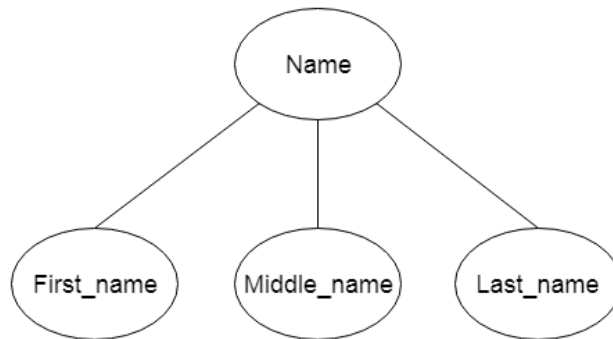
2. Attribute: The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute.



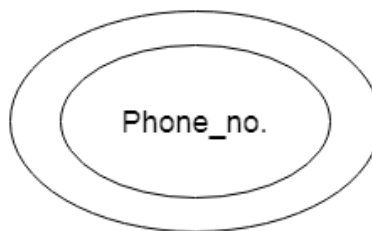
a. Key Attribute: The key attribute is used to represent the main characteristics of an entity. It represents a primary key. The key attribute is represented by an ellipse with the text underlined.



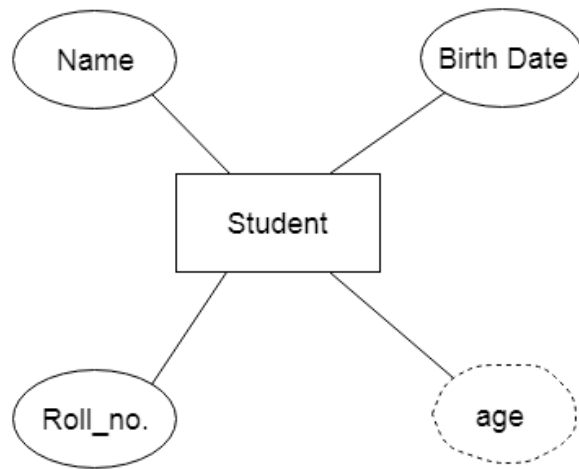
b. Composite Attribute: An attribute that is composed of many other attributes is known as a composite attribute. The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.



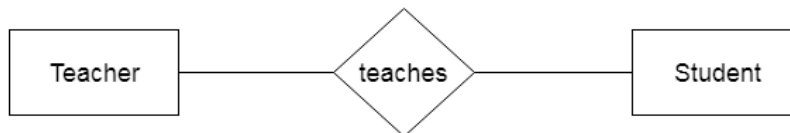
c. Multivalued Attribute: An attribute can have more than one value. These attributes are known as a multivalued attribute. The double oval is used to represent a multivalued attribute.



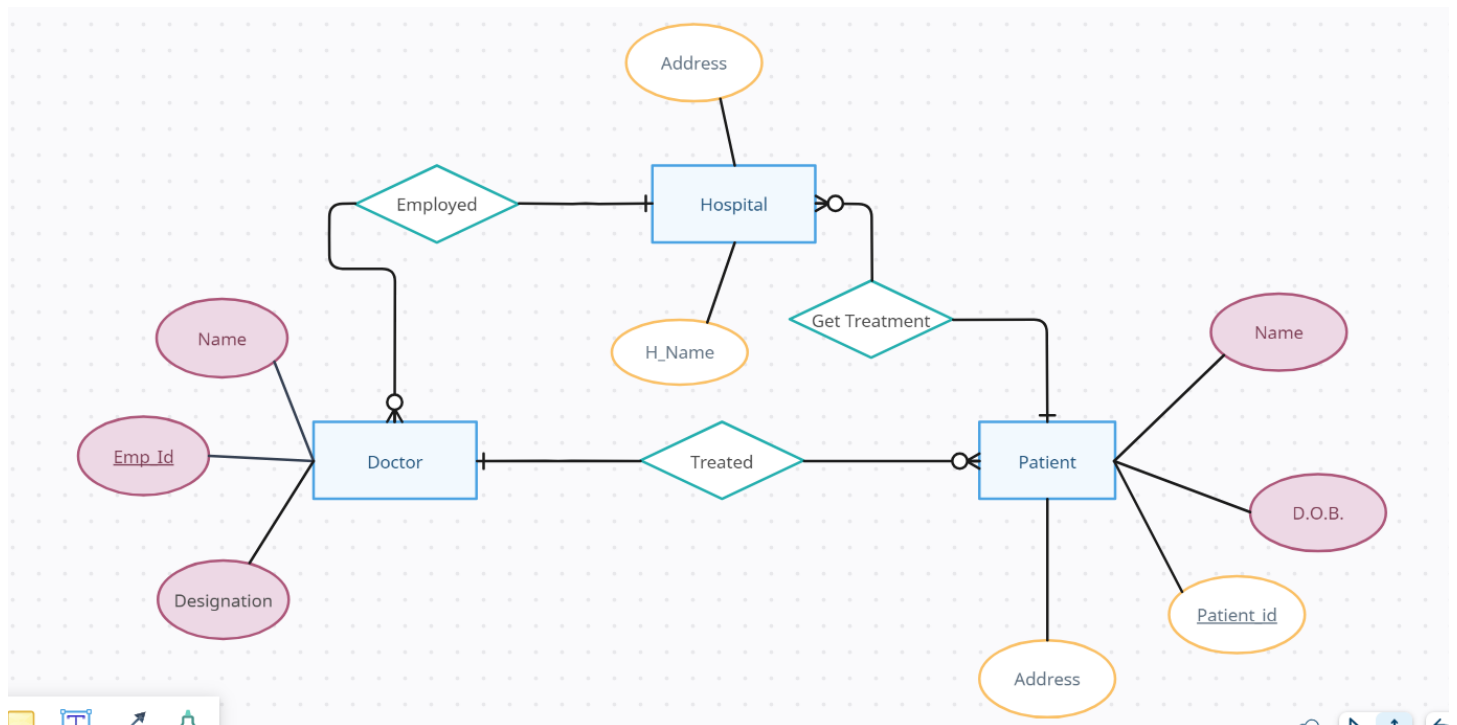
d. Derived Attribute: An attribute that can be derived from another attribute is known as a derived attribute. It can be represented by a dashed ellipse



3. Relationship: A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.



ER Diagram of Hospital: -



Experiment 3

Objective: Writing SQL statements using Oracle/MySQL:

- a) Creating and managing tables
- b) SQL Select statement
- c) Restricting and sorting data
- d) Aggregating data using group functions
- e) Manipulating data

Theory:

1. Creating table:

CREATE query is used to create a new database or table in SQL.

create table student (

Roll_no int(5) PRIMARY KEY,

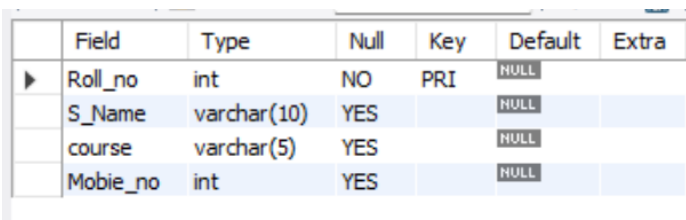
S_Name varchar(10),

course varchar(5),

Mobie_no int(10)

);

-- for integer of 10 or more digits we have to use BIGINT or VARCHAR



	Field	Type	Null	Key	Default	Extra
►	Roll_no	int	NO	PRI	NULL	
	S_Name	varchar(10)	YES		NULL	
	course	varchar(5)	YES		NULL	
	Mobie_no	int	YES		NULL	

Insert values into table:

INSERT query is used to insert values in a given table.

insert into Student **values** (101, 'Anurag', 'MCA', 892232134), (102, 'Diksha', 'MCA', 872612891), (103, 'Faizan', 'MCA', 721563217) ;

2. Select:

SELECT query is used to select all or particular records or field the table.

-- Selecting all the rows from a table

select * from Student;

	Roll_no	S_Name	course	Mobie_no
▶	101	Anurag	MCA	892323214
	102	Diksha	MCA	872362891
	103	Faizan	MCA	761563217
*	NULL	NULL	NULL	NULL

-- SELECT S_Name COLUMN

select S_Name from Student;

	S_Name
▶	Anurag
	Diksha
	Faizan

-- SELECT ROW BASED ON CONDITION

select * from Student where Roll_no=101;

	Roll_no	S_Name	course	Mobie_no
▶	101	Anurag	MCA	892323214
*	NULL	NULL	NULL	NULL

3. Restricting and Sorting:

Sorting table records is done by using ORDER BY query.

By default i.e by using only ORDER BY without giving specific order(asc or desc) table will be shown in asc(ascending) order.

-- SORTING table in descending order

select * from student order by CGPA desc;

	Roll_no	S_Name	course	Mobie_no	CGPA
▶	103	Faizan	MCA	761563217	9.8
	101	Anurag	MCA	892323214	9.6
	102	Diksha	MCA	872362891	9.0
*	NULL	NULL	NULL	NULL	NULL

4. Aggregate Functions: MIN (), MAX (), COUNT (), SUM (), AVG ()

Aggregate functions are the function that perform some calculations and gives results as per their

MAX()-> gives maximum value in a field.

select max(CGPA) from Student;

max(CGPA)
9.8

MIN()-> gives minimum value in a field.

select min(CGPA) from Student;

min(CGPA)
9.0

COUNT()-> gives count of values in a field.

select count(CGPA) as Total_Students from Student;

Total_Students
3

SUM()-> gives sum of values in a field.

select sum(CGPA) from Student;

sum(CGPA)
28.4

AVG()-> gives average of values in a field.

select avg(CGPA) as Average_CGPA from Student;

Average_CGPA
9.46667

OR

select max(CGPA) as Maximum_CGPA, min(CGPA) as Minimum_CGPA, count(CGPA) as No_of_CGPA, sum(CGPA) as Sum_of_CGPA, avg(CGPA) as Avearge_of_CGPA from Student;

Maximum_CGPA	Minimum_CGPA	No_of_CGPA	Sum_of_CGPA	Avearge_of_CGPA
9.8	9.0	3	28.4	9.46667

5. Manipulating:

For manipulating records and fields in a table we use UPDATE-SET, ALTER, ADD, DROP, etc queries.

-- UPDATING a record in a table

update Student set RNo=103 where RNo=112;

select * from Student;

	Roll_no	S_Name	course	Mobie_no	CGPA
▶	102	Diksha	MCA	872362891	9.0
	103	Faizan	MCA	761563217	9.8
	180	Anurag	MCA	892323214	9.6
*	NULL	NULL	NULL	NULL	NULL

-- UPDATING multiple field entries

update Student set CGPA=9.8 where Roll_no =103;

update Student set CGPA=9.6 where Roll_no =101;

update Student set CGPA=9 where Roll_no =102;

select * from Student;

	Roll_no	S_Name	course	Mobie_no	CGPA
▶	102	Diksha	MCA	872362891	9.0
	103	Faizan	MCA	761563217	9.8
	180	Anurag	MCA	892323214	9.6
*	NULL	NULL	NULL	NULL	NULL

-- ADDING a field into a table float(no_of_digits,
no_of_digits_after_decimal)

alter table student add CGPA float (2,1);

select * from Student;

	Roll_no	S_Name	course	Mobie_no	CGPA
▶	102	Diksha	MCA	872362891	NULL
	103	Faizan	MCA	761563217	NULL
	180	Anurag	MCA	892323214	NULL
*	NULL	NULL	NULL	NULL	NULL

-- REMOVE a field

alter table student drop column CGPA;

select * from Student;

	Roll_no	S_Name	course	Mobie_no
▶	102	Diksha	MCA	872362891
	103	Faizan	MCA	761563217
	180	Anurag	MCA	892323214
*	NULL	NULL	NULL	NULL

Experiment — 11

Aim: Automatic Backup of Files and Recovery of Files.

This topic describes how to back up files and filegroups in SQL Server by using SQL Server Management Studio, Transact-SQL. When the database size and performance requirements make a full database backup impractical, you can create a file backup instead. A file backup contains all the data in one or more files (or filegroups).

Using SQL Server Management Studio

1. After connecting to the appropriate instance of the SQL Server Database Engine, in Object Explorer, click the server name to expand the server tree.
2. Expand Databases, and, depending on the database, either select a user database or expand System Databases and select a system database.
3. Right-click the database, point to Tasks, and then click Back Up. The Back Up Database dialog box appears.
4. In the Database list, verify the database name. You can optionally select a different database from the list.
5. In the Backup type list, select Full or Differential.
6. For the Backup component option, click File and Filegroups.
7. In the Select Files and Filegroups dialog box, select the files and filegroups you want to back up. You can select one or more individual files or check the box for a filegroup to automatically select all the files in that filegroup.
8. Either accept the default backup set name suggested in the Name text box, or enter a different name for the backup set.
9. (optional) In the Description text box, enter a description of the backup set.
10. Specify when the backup set will expire:
 - TO have the backup set expire after a specific number of days, click After (the default option) and enter the number of days after set creation that the set will expire. This value can be from 0 to 99999 days; a value of 0 days means that the backup set will never expire.

The default value is set in the Default backup media retention (in days) option of the Server Properties dialog box (Database Settings page). To access this

option, right-click the server name in Object Explorer and select properties; then select the Database Settings page.

- To have the backup set expire on a specific date, click On, and enter the date on which the set will expire.
- II. Choose the type of backup destination by clicking Disk or Tape. To select the paths of up to 64 disk or tape drives that contain a single media set, click Add. The selected paths are displayed in the Backup to list.

Note

To remove a backup destination, select it and click Remove. To view the contents of a backup destination, select it and click Contents.

12. To view or select the advanced options, click Options in the Select a page pane.

13. Select an Overwrite Media option, by clicking one of the following:

- Back up to the existing media set

For this option, click either Append to the existing backup set or Overwrite all existing backup sets.

- (optional) Select Check media set name and backup set expiration to cause the backup operation to verify the date and time at which the media set and backup set expire.
- (optional) Enter a name in the Media set name text box. If no name is specified, a media set with a blank name is created. If you specify a media set name, the media (tape or disk) is checked to see whether the actual name matches the name that you enter here.

If you leave the media name blank and check the box to check it against the media, success will equal the media name on the media also being blank. • Back up to a new media set, and erase all existing backup sets

For this option, enter a name in the New media set name text box, and, optionally, describe the media set in the New media set description text box.

14. (optional) In the Reliability section, check:

- Verify backup when finished.

- Perform checksum before writing to media, and (optional) Continue on checksum error.
15. If you are backing up to a tape drive (as specified in the Destination section of the General page), the Unload the tape after backup option is active. Clicking this option enables the Rewind the tape before unloading option.

Note

The options in the Transaction log section are inactive unless you are backing up a transaction log (as specified in the Backup type section of the General page).

16. SQL Server 2008 (10.0.x) Enterprise and later versions support backup compression. By default, whether a backup is compressed depends on the value of the backupcompression default server configuration option. However, regardless of the current server-level default, you can compress a backup by checking Compress backup, and you can prevent compression by checking Do not compress backup.

Examples

The following examples back up one or more files of the secondary filegroups of the Sales database. This database uses the full recovery model and contains the following secondary filegroups:

- A filegroup named SalesGroup1 that has the files SGrp1Fi1 and SGrp1Fi2.
- A filegroup named SalesGroup2 that has the files SGrp2Fi1 and SGrp2Fi2.

A. Create a file backup of two files

The following example creates a differential file backup of only the SGrp1Fi2 file of the SalesGroup1 and the SGrp2Fi2 file of the SalesGroup2 filegroup.

--Backup the files in the SalesGroup1 secondary filegroup.

BACKUP DATABASE sales

FILE =

'SGrp1Fi2',

```
FILE =  
  
'SGrp2Fi2'  
  
TO DISK=          ServerBackups\Sales\SalesGroup1.bck';  
  
GO
```

B. Create a full file backup of the secondary filegroups

The following example creates a full file backup of every file in both of the secondary filegroups.

```
—Back up the files in SalesGroup1.  
  
BACKUP DATABASE Sales  
  
FILEGROUP =  
  
'SalesGroup1',  
  
FILEGROUP = 'SalesGroup2'  
  
TO DISK = 'C:\MySQLServer\Backups\Sales\SalesFiles.bck';  
  
GO
```

C. Create a differential file backup of the secondary filegroups

The following example creates a differential file backup of every file in both of the secondary filegroups.

```
—Back up the files in SalesGroup1.  
  
BACKUP DATABASE sales  
FILEGROUP = 'SalesGroup1',
```

FILEGROUP = 'SalesGroup2'

TO DISK = 'C:\MySQLServer\Backups\Sales\SalesFiles.bck'

WITH

DIFFERENTI

AL; GO