

DATABASE MANAGEMENT SYSTEM

A database is an organized collection of data, generally stored and accessed electronically from a computer system.

A database management system (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database. A DBMS generally manipulates the data itself, the data format, field names, record structures and file structure. It also defines rules to validate and manipulate this data.

A DBMS retrieves users of framing programs for data maintenance. Fourth-generation query languages, such as SQL, are used along with the DBMS package to interact with a database. Some other DBMS examples include:

- * My SQL
- * Oracle
- * SQL server
- * Fox Pro

A database management system receives instruction from a database administrator (DBA) and accordingly instructs the system to make the necessary changes. These commands can be to load, retrieve or modify existing data from the system.

NORMALIZATION

In relational database design, the process of organizing data to minimize redundancy. Normalization usually involves dividing a large table into two or more tables and defining relationships between the tables. The objective is to isolate data so that additions, deletions, and modifications of a field can be made in one table and then propagated through the rest of the database via the defined relationships.

There are three main normal forms, each with increasing levels of normalization:

1. First Normal Form (1NF): Each field in a table must contain only one value, representing different information. For example, in an employee list, each table would contain one birthdate field.

2. Second Normal Form (2NF): Each field in a table must be a function of the primary key. A field that is not a determinant of the contents of another field must itself be a function of the other fields in the table.

3. Third Normal Form (3NF): No duplicate information is permitted. So, for example, if two tables both require a birthdate field, the birthdate information would be separate in each table, and the two other tables would then reference the birthdate information would be



entity \Rightarrow A real world thing either living or non-living that is easily recognizable and unambiguous. It is anything in the enterprise that is to be represented in our database. It may be a physical thing or simply a fact about the enterprise or any event that happened in the past.





relationship \Rightarrow It is nothing but associated among two or more entities.

- You work in the chemistry department. Entire part in relationship we can often identify relationships with verbs or verb phrases.

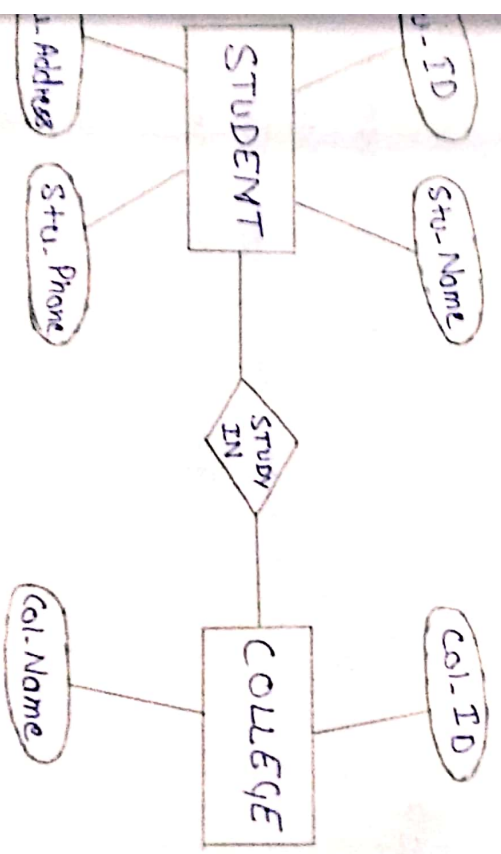
Attributes \Rightarrow It is a single valued property of either an entity-type or a relationship-type. For eg- a lecture might have attributes - time, date, duration, place etc. An attribute is represented by an Ellipse.

ER DIAGRAM NOTATIONS

Shape	Shape Name	Description
	Rectangle	This symbol represents entity types
	Ellipse	Symbol represents attributes

	Line	It links attributes to entity types and entity types with other relationship types.
	Diamond	This symbol represents relationship types.
	Double Ellipse	Multi-valued Attributes
	Double Diamond	Represent weak relationship types

Simple ER Diagram



index field in the birthdate table.
change to a birthdate would automatically
reflect in all tables that link to
birthdate table.

are additional normalization levels,
is Boyce Codd Normal Form (BCNF),
normal form (4NF) and fifth normal
5NF). While normalization make database
efficient to maintain, they can also
turn more complex because data is
spread into so many different tables.

ENTITY-RELATIONSHIP MODEL

ER model is a high-level conceptual data
diagram. Entity-relationship model is
on the notations of real-world entities and the
relationship between them.

modelling helps you to analyse data requirements
rationally to produce a well-designed database.
is considered a best practise to complete
modelling implementing your database. ER diagram
shows the relationships of entity set stored in
database.

Components of the ER Diagram

model is based on three basic concepts:

entities
relationships
attributes

Logical Operators

In this three logical operators are used

→ AND
→ OR
→ NOT

If we check both the conditions then apply

AND

If we check one condition then apply OR

select name from employee where salary > 4000
or salary < 10000;

NAME
Babulal
Ram Lal

select name from employee where salary > 4000
and salary < 6000;

NAME
Babulal

select name from employee where salary > 4000 and
not (salary = 5000);

NAME
Babulal
Ram Lal

select sum(salary) from employee

sum(salary)
113000

Write SQL queries for arithmetic and logical operators.

select 4+5 sum-of-two-no from dual;

SUM_OF_TWO_NO

9

to select all tables from login

select * from tab;

ITEMP
TEMPR
HOSPITAL

EMPLOYEE

<u>EMPNO</u>	<u>NAME</u>	<u>DOJ</u>	<u>SALARY</u>
11	Babulal	0	5500
12	Ramlal	0	6300

select name, salary, salary+500 new.sal from employee

<u>NAME</u>	<u>SALARY</u>	<u>NEW.SAL</u>
Babulal	5500	5500
Ramlal	6300	6800

These columns are only representing the name of the table only it does not replace the actual data.

update itemp set daj = '' where id = 104;
if now updated

select * from itemp

ID	NAME	DEPT	DOJ	CITY
01	Ayush	CSE	14-Mar-2015	Kanpur
02		ME		Kanpur
03	Amit		15-Mar-2015	Lucknow
04				Lucknow
05				Lucknow

Alter the datatype of variable
alter table nempt modify name varchar(12);
table altered

Alter the table name of the column

alter table nempt rename column drame
to dept;

table altered

Alter the name of the table

rename nempt to itemp;

table altered.

select * from itemp;

ID	NAME	DEPT	DOJ	CITY
101	Ayazh	CSE	14-Mar-2015	
102		ME		
103	Amit		15-Mar-2015	
104			17-Mar-2015	
105				

UPDATE

update itemp set doj = '29-Mar-2015'

where id = 102

1 row updated

update itemp set city = 'Lucknow';

5 rows updated

update itemp set city = 'Kanpur' where
dept = 'CSE' or dept = 'ME';

2 rows updated

insert into nemp (id, doj);
values (104, '17-Mar-2015');

1 row created

insert into nemp (id)
values (105);

1 row created

select * from nemp

id	name	dname	doj
101	Ayush	CSE	14-Mar-2015
102			
103	Amit	ME	15-Mar-2015
104			17-Mar-2015
105			

Alter table :-

alter table nemp add city varchar(10);

Table altered

desc nemp

Name	NULL?	Type
ID		NUMBER(5)
NAME		VARCHAR2(10)
DNAME		VARCHAR(10)
DOJ		DATE
CITY		VARCHAR2(10)

Create a table and then do alter, add modify, and rename a table

Create a table and then inserting the value as follows

id	name	dname	dof
101	Ayush	CSE	14-Mar-2015
102	—	ME	—
103	Amit	—	15-Mar-2015
104	—	—	17-Mar-2015
105	—	—	—

Syntax:-

create table nemp (id number (15), name varchar (10), dname varchar (10), dof date);

table created

insert into nemp (id, name, dname, dof)

values (101, 'Ayush', 'CSE', '14-Mar-2015');

I now created

insert into nemp (id, dname);

values (102, ME);

I now created

insert into nemp (id, name, dof);

values (103, 'Amit', '15-Mar-2015');

I now created

select * from student;

ROLL NO	NAME	YEAR	CITY
3215	Ayush	Second	Lucknow
4633	Amit	Second	Lucknow
6332	Khursheed	Second	Lucknow
9331	Divakar	Second	Delhi
6339	Rehan	Second	Mumbai
9334	Kaif	Second	Bangalore

Creating a student table and inserting values and displaying.

```
create table Student (Rollno int(4), Name varchar(10),  
Year varchar(10), city varchar(10));
```

table created

```
insert into student  
values ('3215', 'Ayush', 'Second', 'Lucknow');
```

row created

```
insert into student  
values ('4633', 'Amit', 'second', 'Lucknow');
```

row created

```
insert into student  
values ('6332', 'Khursheed', 'Second', 'Lucknow');
```

row created

```
insert into student  
values ('9331', 'Divakar', 'second', 'Delhi');
```

row created

```
insert into student  
values ('6339', 'Rehan', 'second', 'Mumbai');
```

row created

```
insert into student  
values ('9334', 'Kaif', 'second', 'Banglore');
```

CREATE TABLE

The create table statement is used to create new table in a database.

Syntax - create table table-name (column1 datatype, column2 datatype, ----- columnn datatype);

Example - create table Persons (PersonID int, LastName varchar(255), FirstName varchar(255), Address varchar(255), city varchar(255));

INSERT INTO

The insert into statement is used to insert new records in a table.

Syntax 1 - The first way specifies both the column names and the values to be inserted

insert into table-name (column1, column2,)
values (value1, value2, value3,);

Syntax 2 - If we are adding values for all the columns in the table.

insert into table-name
values (value1, value2, value3,);

INTRODUCTION TO SQL

SQL is a database computer language designed for the retrieval and management of data in a relational database. SQL stands for Structured Query Language. SQL is a language to operate databases; it includes database creation, deletion, fetching rows, modifying rows, etc. SQL is an ANSI standard language, but there are many different versions of the SQL language. SQL is the standard language for relational Database system. All RDBS like MySQL, MS Access, Oracle, Sybase, Informix and SQL server use SQL as their standard database language. SQL is widely popular because it offers the following advantages.

- * Allow users to access data in RDBS.
- * Allow users to describe the data.
- * Allow users to create and drop databases and tables.

DATA TYPES IN SQL

Exact Numeric Data Types

bigint	• decimal
int	• numeric
smallint	• money
tinyint	• smallmoney
bit	

Approximate Numeric Data Types

float
real

Date and Time Data Types

datetime	• date
smalldatetime	• time

Character Strings Data Types

• char	• varchar(max)
• varchar	• text

Binary Data Types

• binary	• varbinary(max)
• varbinary	• image