

NAME :- Arshay



Gupta

CONTACT :- 9818116189

ORACLE

ORACLE

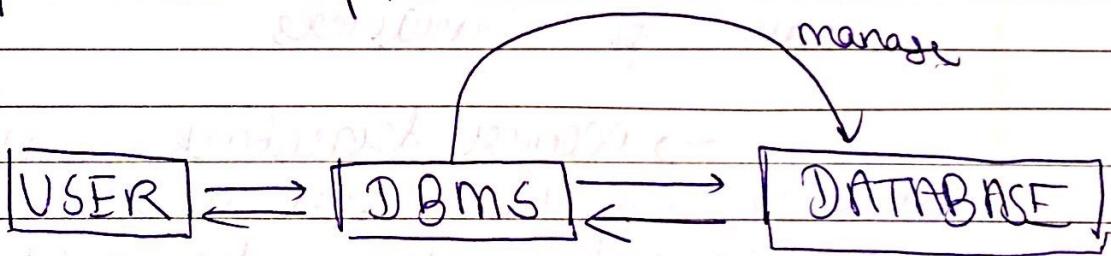
Data → whatever we entered into the computer is known as Data.

→ Data can also be called as raw fact
→ Data doesn't give any meaning.

Information :- Processed data can be called as Information.

→ It gives some meaning.

Database :- collection of information stored at particular place written in a predetermined manner belongs to particular topic.



→ In the above diagram if user need any information from database then he has to contact with Database, take the necessary information from database & give it to the user.

MODULES OF DATABASE

broadly → These modules are divided into following types :-

- i) FMS / FDDBMS
- ii) IMS / HIDBMS
- iii) NDBMS
- iv) RDBMS
- v) ORDBMS
- vi) OODBMS

FMS :- → This is the first module developed in late 1950's.

In this module data always storage with in the form of sequential manner or continuous stream of characters.

→ Major drawback of this module is, if the user need to find out any value then he need to start searching from beginning of file.

→ Eg :- COBOL, C

HMS :-

→ This module was developed by IBM at 1968, when they were developing

a project called IMS (information management system). This module was developed on an operating system called MSDOS.

→ In this module data always arranged in the form of hierarchical manner or in different levels. Top level can be called as root, second, third and fourth levels can be called as parent, child and siblings/ leaf respectively

→ Main advantage of this module is user can easily find out the value without wasting much time.

drawbacks :-

only one user can share the database at a time.

i) If you want to add one more extra level between the existing levels then user has to reconstruct the entire tree structure.

NORMS :-

→ This module was developed by IBM at 1969 when they were developing the project called IMS.

→ This module was developed on an operating system called MULTICS (Multiplex Information Computing System).

→ Major advantage of this module is more than one user can share database at a time.

drawbacks:-

- i) There is no security
- ii) It increases redundancy in database
- iii) It occupies lot of memory and leads to decrease the system performance & increase inconsistency.

RDBMS :-

→ This module was developed by the german scientist EF Codd at 1970. based on a mathematical concept names called relational algebra.

→ Here, relation can be defined as commonness between the various objects.

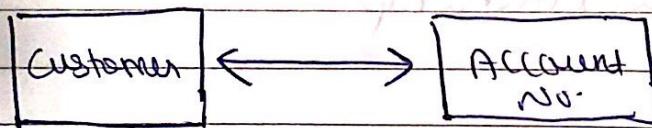
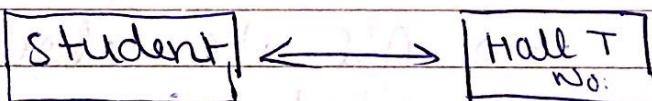
→ Relations are divided into three types :-

i) 1:1

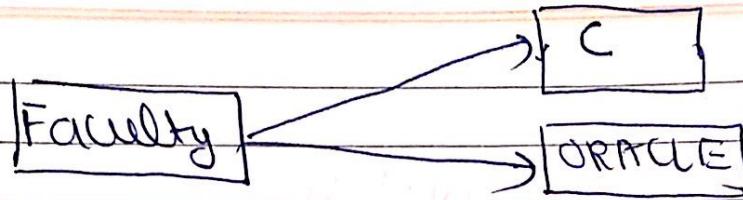
ii) 1:m / m:1

iii) m:m

→ 1:1 relationship :- In this relation, one object has the relationship with only one another object.

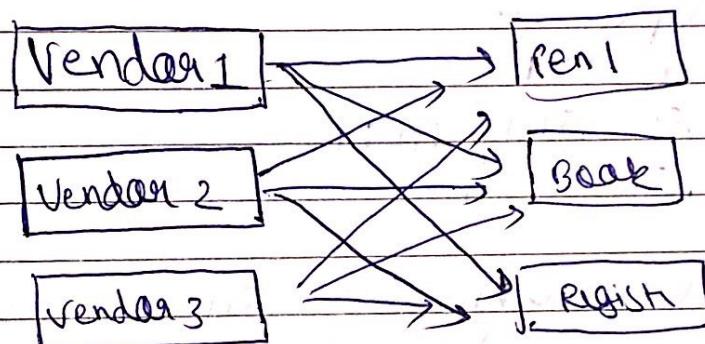


→ 1:m relationship :- In this relation, one object has relationship with many other objects.



M:m relationship :-

→ In this relation, many object has relationship with many other objects.



m:m

→ All the above relations can be called as degree of relationship.

Properties of RDBMS :-

i) Data should be stored in the form of Table. A table can be defined as collection of rows and columns.

→ The horizontal lines are known as Rows / Record / Tuple.

→ Vertical lines are known as column / field / Attribute.

→ Intersection of rows & column is called as cell. A cell is a place where we can store the data.

→ other name of row can be called as record / tuple.

→ other name of column can be called as field / Attribute.

→ other name of table can be called as object / entity.

column/field/ attribute	EMP		
	EID	Name	Salary
	101	RAT	10,000
	102	MAT	20,000
	103	SAT	30,000
	104	KAT	40,000

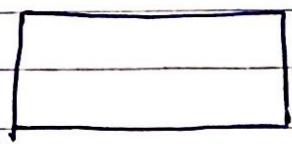
→ row / record / tuple
→ cell

- ii) Table should not contain duplicate columns.
- iii) While inserting the records in the table user no need to follow any order.
- iv) Database should not contain duplicate objects / table name.
- v) While we define columns in the table user no need to follow any order.

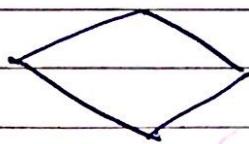
E-R diagrams

→ This concept was developed by Mr. Peter Cheng, it is the diagrammatic representation of manual database.

→ whenever we need to design ER-diagrams we need to use the following symbols.

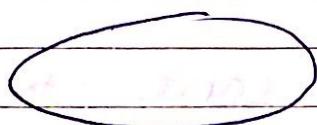


Entity



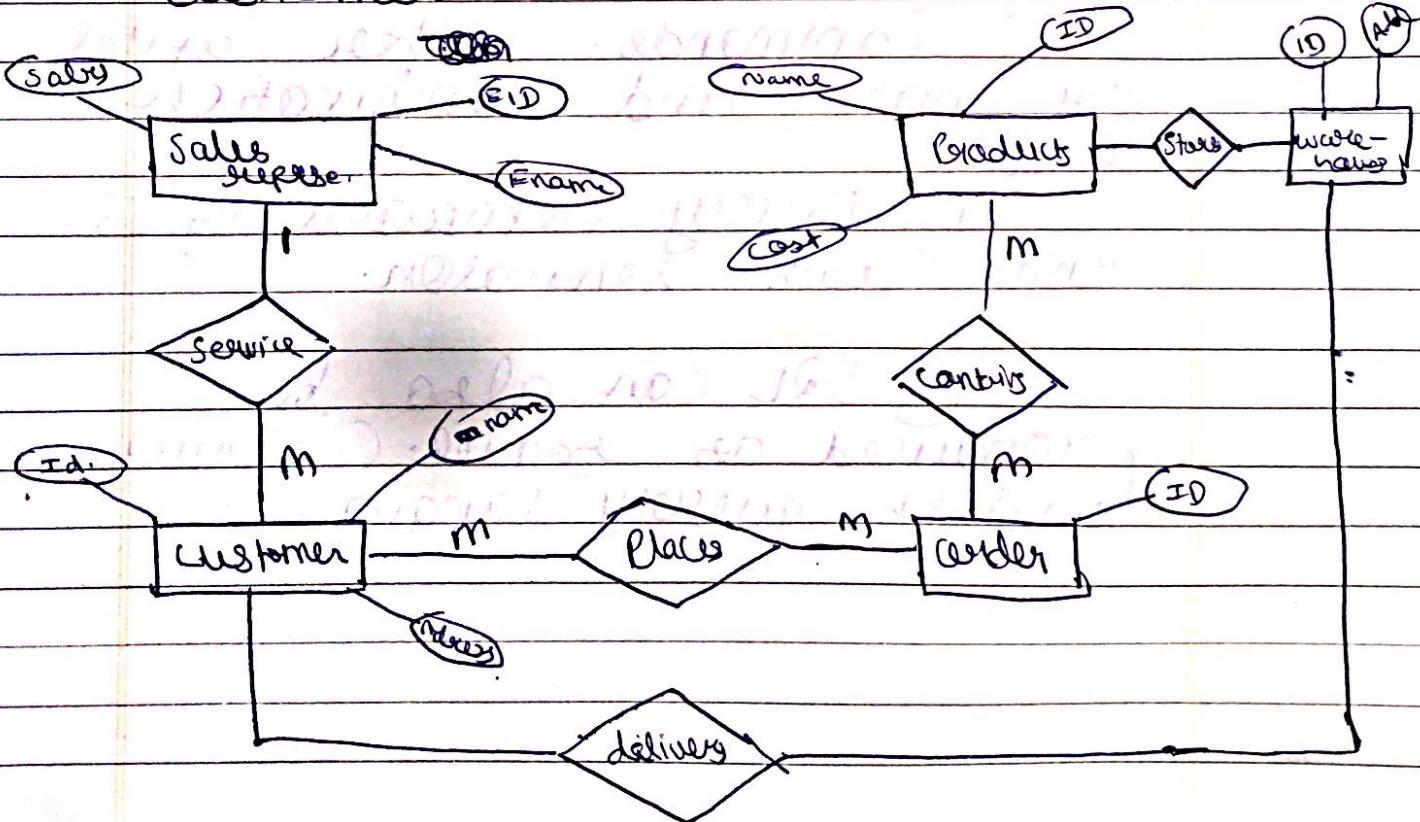
Name of the relationship

→ Connection



Attribute

Draw an E-R diagram which represents the relation between sales representative and the customer.



SQL

→ It is the processed language which is used to communicate with all types of databases such as oracle, SQL SERVER & MySQL etc.

→ This language was developed by german scientist Mr. EF Codd.

→ In 1960 AT&T appeared this language.

→ In 1972 SQL came into the market.

Features of SQL :-

i) It is not a case sensitive language, so we can write SQL commands either capital or small and combination also.

ii) Every command of SQL ends with semicolon.

iii) SQL can also be pronounced as sequel (structured english query language).

iv) SQL can be called as natural language interface, because every command of SQL is similar to English language.

v) SQL can be called as common language interface, because this is the only language to communicate with all types of data bases, such as Oracle, SQL server, DB2 etc.

→ SQL is subdivided into 4 sublanguages :-

i) Data Definition Language (DDL)
ii) Data manipulation language (DML)
iii) Transaction control language (TCL).

iv) Data control language (DCL).

- (i) creating DB
- (ii) inserting data
- (iii) reading data

Data Types of attack :-

1) Number :- This data type allows us to enter numeric values ie either integer numbers or decimal numbers.

→ It is further divided into two types :-
i) NUMBER (SIZE)
ii) Number (l, s).

NUMBER (SIZE) :-

→ This data type allows us to enter integer values ie either positive numbers or negative numbers such as student Roll Numbers, mobile Numbers, pin codes etc.

Eg:-

EID Number (3)

PIN Number (6)

M-NO Number (10)

→ The maximum size of this datatype is 1-38 digits.

① Number (P, S) :-

→ This data type allows us to enter decimal values such as student average marks, employee salaries etc.

→ Here P represents precision and S represents scale.

→ Precision means number of digits including decimal places.

Ex: Number (5, 2)

Rs 150.75

Size Number (6, 2)

Rs 9870.50

4 + 2

→ Maximum size of this data type is 1-38 digits.

② Alphabet :-

→ This data type allows us to enter character values such as student Name, employee Name etc.

→ It is further divided into two types :-

i) char (size)

ii) VARCHAR (size)

char (size) :-

→ This data type allows us to enter character values such as student name, employee name etc. & maximum size of this datatype is 1-2000 characters.

→ It always allocates the memory in a static fashion.

VARCHAR (size) :-

→ This datatype also allows us to enter character values such as student name, employee name etc. & the maximum size of this datatype is 1-4000 characters.

→ It always allocates the memory in dynamic fashion.

③ DATE :-

→ This datatype allows us to enter date values such as student joining date, employee hire date etc.

→ The default format of date datatype is DD-MM-YY

→ The range of the date type is 01-JAN-4712 BC to 31-DEC-9999 AD

Eg:- DATE DATE;

~~TIME STAMP~~

④ TIME STAMP :-

→ This data type is the combination of date + time.

→ The default format of time stamp is :-

DD-MM-YY HH:MM:SS

Student (1992) at 10:45 AM

Data Definition Language :-

→ This is the first sublanguage in SQL which is used to define database objects such as table, view, synonyms, sequence, index, trigger, stored procedure, function, package and trigger.

→ This language contains five commands :-

- I) Create
- II) Alter
- III) Rename
- IV) Truncate
- V) Drop

CREATE :-

→ This command is used to create database objects such as table, view etc

SYNTAX TO CREATE table :-

Q

Create table <Table Name>

(col 1 datatype (size);
 Col 2 datatype (size),
 Col n datatype (size));

Example :-

Create table emp
 (eid number (3),
 ename Varchar2 (15),
 sal number (6,2));

Table created.

→ clear, this command
 is used to clear the screen.

→ desc / describe, this command
 is used to show the structure
 of the table.

Syntax:-

desc <Table Name>

desc aft3

sp-helptext <stored procedure name>,

Rules for naming a table :-

- ① Table Name must starts with an alphabet.
it contains minimum 1 character and maximum 30 characters and it should not allow any spaces or special characters except - , \$, # and @ to 9 numbers.
- ② A table can have minimum 1 column and maximum 1000 columns.
- ③ table can have minimum 0 records and maximum n number of records.
- ④ Oracle keywords or reserved words should not be used as column names & table names.
- ⑤ column name must start with an alphabet
it contains minimum 1 character & maximum 30 characters & it should not allow any spaces or special characters except - , \$, # and @ to 9 numbers.

Steps to enter into ORACLE database

① Click on Start button → go to programs

→ ORACLE - ORATAB11G - HOME 1

→ Application Development → SQL plus

NOTE :-

→ clear is used to clear the screen.

→ desc/describe is used to show the structure of the table.

→ Syntax :-

desc <table Name>

Eg:- desc emp11

→ The above two commands are SQL plus commands, the difference between SQL commands and SQL plus commands are every SQL command ends with semicolon whereas SQL commands, no need to write semicolon in the end.

Syntax to make the table read only :-

alter table <table name> read only;

Eg:- alter table month II read only;

② ALTER :-

→ This command is used to modify the structure of the table, and we can make table read only.

→ using this command we can perform four different operations. This command contains four subcommands :-

① ALTER - MODIFY

② ALTER - ADD

③ ALTER - RENAME

④ ALTER - DROP

ALTER - MODIFY :-

→ This command is used to ~~remove~~ increase or decrease the size of the data type and also we can change the datatype from old datatype to new datatype.

Syntax :-

Alter table <table Name> modify

Column name datatype (size);

Syntax to make the table both read & write

alter table <table name> read write;

Eg:- alter table mca11 read write;

Eg:-

→ Alter table mca11 modify ename varchar(25);

→ If you want to change more than one column at a time.

Syntax :-

alter table <table Name> modify

(column 1 datatype(size),

column 2 datatype(size),

⋮
column n datatype(size));

alter table mca11 modify

(ename char(10),

cid number(5))

② Alter - Add :-

→ This command is used to add a new column to a existing table.

Syntax :-

alter table <table Name> add column
name datatype (size)

Eg:-
→ alter table man11 add dept no number(3)

→ If I want to alter more than one column.

alter table <table name> add
(col1 datatype (size),
col2 datatype (size),
coln datatype (size));

Eg:-
→ alter table man11 add
(matno number(10),
address varchar(25))

→ We cannot add a new column at required position of table, all the columns will add to the end of table only.
Also we cannot change the position of columns.

ALTER - RENAME :-

→ This command is used to change the column name from old column name to new column name.

Syntax :-

alter table <table Name> rename
Column <old column Name> to
<new column Name>;

Eg:-
alter table moonll rename column
sal to Salary

NOTE :- we cannot change more than one column name at a time.

Syntax to change table Name :-

alter table <table Name> rename to
< new table name>

Eg :-

alter table moonll rename to moonlc

ORA-04043 : Table <table name> doesn't exist



Alter Drop :-

→ This command is used to remove the column from existing table.

→ Using this command we can remove the column in two ways :-

- i) logical dropping
- ii) physical dropping

Logical dropping :-

→ If the table contains more number of columns we can use logical dropping.

→ Performance wise logical dropping is faster than physical dropping.

Syntax to drop the column logically

alter table <Table Name> set unused
column <column name>

Eg :-

alter table mchen12 set unused
column salary

Syntax to drop more than one column

alter table <Table Name> drop unused
(list of columns)

Eg:-

alter table main12 drop unused
(ename, deptno, mtno, address)

/

Syntax to know how many number of columns dropped logically :-

Select * from user_unused_col_tabs,
OR

Select * from all_unused_col_tabs;

Syntax to drop the column physically

alter table <Table Name> drop column
<column name>

Eg:-

alter table main12 drop column sal

/

Syntax to drop more than one column physically :-

alter table <table Name> drop
(list of columns)

Eg :-

alter table mounk drop
(ename, dept no)

/

③ RENAME :- This command is used
to change the table Name
from old table Name to New
table Name.

Syntax :-

rename <old Table Name> to
<new Table Name>;

Eg :-

rename man12 to man11

④ TRUNCATE :-

→ This command is used to delete all the records permanently from the existing table.

Syntax :-

truncate table <table Name>;

Eg:-

truncate table mcaen12;

delete

① DML

truncate

① DDL

ii) temporarily

① all records deleted permanently

iii) it deletes specific records!

iii) it deletes all the rows

⑤ DROP :-

Syntax to drop the table temporarily

drop table <Table Name>;

drop table mcaen11;

Syntax to get back the table from Recycle bin:

flashback table < table Name > to before drop;

flashback table main || to before drop;

Syntax to drop the table permanently from DB

drop table < table Name > purge;

drop table main || purge;

Syntax to drop the table permanently from recycle bin:-

purge table < table Name >;

purge table main ||;

Syntax to drop all tables at a time from recycle bin:

purge recycle bin;

Syntax to see all the list of tables:

select * from cat;

OR

select * from tab;

NOTE :- all DDL commands are auto committed

DM

[Data Manipulation Language]

→ This language is used to manipulate the data within database.
It contains following commands :-

- i) Insert
- ii) update
- iii) Delete
- iv) Select

Insert :- This command is used to insert the records into the table.

→ Using this command we can insert the records by two methods :-

i) Explicit Method :-

→ In this method user need to enter all the values into all the columns in the table without left any column data.

Syntax of explicit method :-

insert ~~to~~ into <Table Name> values
(Val 1 , Val 2 ... Val n);

Example :-

Insert into mean11 Values (101, 'Ray', 9956);

→ Insert the records using
insertion ~~operator~~ operator :-

Insert into <Table Name> Values

(& col 1 , & col 2 ... & col n)

Eg :-

insert into mean11 Values
(& eid, & ename, & sal);

→ We use this insertion operator
because in the above method
if we have to write 100 values/
records then again & again
we have to write insert into <Table
Name> - To avoid this again &
again writing we use insertion
operator - By using insertion operator
it will ~~ask~~ automatically ask to enter values

Implicit method :-

→ In this method user can enter the value at required places in the database table so that user can omit some column information.

→ If the user cannot enter the value in any column in the table then that column automatically allows null.

→ null doesn't mean 0, it means no value.

Syntax :-

```
insert into <Table Name> (col1, col2, coln)
values (Val1, Val2, -- Valn);
```

Eg :-

```
insert into mounil (eid, sal) values
(106, 9632);
```

→ Here we can also use \$ symbol so that we don't need to write code again & again. SQL automatically ask for the values to be enter. After entering the values type / so that it ask again for the values.

Syntax :-

insert into <table name> (col₁, col₂,
values
(& col₁, & col₂ -- & col_n);

Eg :-

insert into main11 (eid, sal) values
(& eid, & sal);

NOTE :- double ampersand
(&&) →

→ If we give double &&
before any column in the table
then whatever value we have
given for the column, same
value automatically applicable to
succeed record in the table.

Syntax :-

insert into main11 values
(& eid, 'ename', & sal);

→ But the question is can we apply it to all the column.

Answer is yes you can apply it to all the columns, doesn't matter more than one value can be same in the database table; until and unless we didn't apply any constraints.

UPDATE :-

→ This command is used to modify the data in the table.

→ Using this command we can modify all the records in the table and also we can modify specific records of the table (using where clause).

Syntax :-

update <tableName> set columnName = Value;

Eg :-

Update maven11 set Sal = 15000;

/ → It will update all salaries to 15000/-.

Syntax to change more than one column data :-

update <Table Name> set
col1 = value , col2 = value ...
coln = value ;

Eg :-

update mernll set
cid = 007 , ename = 'James Bond'

③ Delete :-

→ This command is used to delete ~~one~~ the records from existing table.

→ Using this command we can delete all the records & also we can delete the specific records from the table.

Eg :- using where clause.

Syntax :-

delete from <table name>;

Eg :- delete from mernll;

Truncate

i) It is a DDL command.

ii) It is permanent deletion.

~~Temporary~~

iii) We cannot delete the specific record.

Delete

i) It is a DML command.

ii) It is temporary deletion, we can get back through rollback command.

iii) We can delete the specific records.

④ Select :-

This command is used to retrieve the data from existing table. Using this command we can retrieve all the records in the table & also we can retrieve specific records in the table. (using where clause).

Syntax :-

Select * from <Table Name>;

~~select~~

Eg:- Select * from model11;

→ In above example * represent all columns.

ALIAS :-

→ Alias is a duplicate name or an alternate name for the original column name or table name or any expression name.

→ whenever we need to submit understandable or meaningful reports to the end user then we need to use alias.

→ we can provide alias name in three levels

- (i) column level
- (ii) expression level
- (iii) Table level.

Column level Alias :-

→ Providing the alias name for the column is known as column level alias.

Expression Level Alias :-

→ Providing the alias name for the expression is known as expression level alias.

Select emp employee id, ename employee name,
Sal salary from mounit where ^{and} Salary < 11000

→ Using select command we can retrieve data in 3 ways :-

- i) Projection
- ii) Selection
- iii) Joins

Projection :-

Retrieval of specific column in table is known as projection

Syntax :-

Select col1, col2 - coln from <Table Name>

Eg :-

Select empid, ename from mounit

Selection :-

→ Retrieve the data based on condition

Syntax :-

Select * from <Table name> where
condition;

Eg:-

Select * from销货员 where
e_id = 104

Joining :-

→ Retrieve the data from more than one table known as

Join

Syntax :-

Select * from tab1, tab2;

Eg:-

Select * from emp, dept;